UNIVERSITÄT DER BUNDESWEHR
FAKULTÄT FÜR LUFT- UND RAUMFAHRTTECHNIK
INSTITUT FÜR FLUGSYSTEME

On synthetic datasets for development of computer vision algorithms in airborne reconnaissance applications

Georg Hummel

Vollständiger Abdruck der bei der
Fakultät für Luft- und Raumfahrttechnik
der Universität der Bundeswehr München
zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Gutachter/Gutachterin:

1. Univ.-Prof. Dr.-Ing. Peter Stütz
2. Prof. Dr. Paolo Remagnino

Diese Dissertation wurde am 05.10.2016 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 12.10.2016 angenommen. Die mündliche Prüfung fand am 07.04.2017 statt.

Abstract

The complexity of flight raises the acquisition cost for aerial datasets to prototype, test or evaluate airborne computer vision algorithms. A possible surrogate is the usage of virtual environments. However, it is unclear how results acquired in such environments transfer to real world situations. This thesis presents a general concept to identify performance differences of computer vision algorithm on synthetic and natural data. Further, it correlates these difference to image content differences to find causal relations. Lastly, different ways to parametrize the virtual environment are evaluated to identify rendering and modelling techniques reducing the algorithms performance difference. The results are eventually formulated as recommendations for modelling engineers and programmers to optimize their simulation environment.

Keywords: computer vision; evaluation; synthetic environment; validation; feature detector; repeatability; Image content difference; regression analysis; MPEG7; UAV; airborne; remote sensing; aerial reconnaissance; computer graphics imagery; synthetic data; design recommendations;

Kurzfassung

Die Komplexität des Fliegens erhöht die Beschaffungskosten von Bilddaten zur Untersuchung und Evaluierung von luftgestützten Bildverarbeitungsalgorithmen. Virtuelle Umgebungen können hier als möglicher Ersatz dienen. Allerdings ist nicht geklärt wie übertragbar die resultierenden Ergebnisse sind. Diese Doktorarbeit präsentiert ein allgemeines Konzept zur Ermittlung der Leistungsdifferenzen von Bildverarbeitungsalgorithmen operierend auf natürlichen Aufnahmen oder virtuellen Screenshots. Des Weiteren werden kausale Zusammenhänge zwischen diesen Differenzen und bildinhaltlichen Unterschieden gesucht. Zuletzt werden verschiedene Einstellungen der virtuellen Umgebung getestet um Rendering- und Modellierungstechniken zu ermitteln, die genannte Leistungsdifferenzen reduzieren. Die Ergebnisse werden schließlich genutzt um Gestaltungsrichtlinien für Programmierer und Ingenieure zur Optimierung ihrer Simulationsumgebung zu formulieren.

Schlagworte: Bildverarbeitung; Auswertung; virtuelle Umgebungen; Validierung; Feature Detektor; Reproduzierbarkeit; bildinhaltliche Unterschiede; Regressionsanalyse; MPEG7; UAV; luftgestützt; Fernerkundung; Luftaufklärung; Computergraphik; Gestaltungsrichtlinien;

Acknowledgements

First, I would like to thank Prof. Stütz for giving me the opportunity to write this thesis and allowing me to freely select my research focus. I am deeply thankful for all the advice, support and help I received as well as all the trust and freedom I was given. I truly learned a lot these last years, both scientifically and otherwise. I could not have wished for somebody else being my “Doktorvater”. I wrote this in German since this word has a much deeper and truer meaning than its English complement.

I would also like to thank Prof. Remagnino for being my second reviewer and coming all the way to Germany to attend the defense of my thesis. A big thank you also to Prof. Schulte for the given advice and support in every topic.

Special thanks to my seniors Diana Donath, Florian Böhm, Martin Russ, Jens Kleinhempel and Michael Strohal for their guidance, backing, ready ear and their company over the last years.

I also want to thank all my colleagues and close friends from the Institute of Flight Systems (IFS). The list is too long to name all, thank you everyone! I want to honorably mention my friends from the laboratory and office: Christian Hellert, Marc Schmitt, Sten Morawietz, Denis Smirnov, Markus Kaiser, Nikolaus Theißing and Alexander Schelle.

Further I am indebted to Alexander Gebhardt, Marc Schmitt, Martin Russ, Melanie Finze for providing me with an excellent experimental platform and heavily supporting me during the data acquisition flights. I would also like to thank all students that assisted me, especially Benjamin Hanke, Michael Ritter, Eric Ertl and Florian Haase.

Special thanks goes to my wife Lissy, who kept me free from obligation in every possible way while accepting my numerous prolonged workdays; especially in the final stage of this thesis. I would also like to thank my parents Werner and Gertraud for supporting me and my carrier decisions. I am also deeply grateful for my siblings Werner and Karin for always being there.

Finally, I would like to thank everyone, who helped me get this far and all my friends, who still talk to me after I made myself scarce to finish this thesis.

To my wife Lissy

and my son David.

Table of Content

Abstract ..... **III**

Kurzfassung ..... **III**

Acknowledgements ..... **V**

Table of Content ..... **IX**

List of abbreviations ..... **XII**

1 Introduction **1**

1.1 Problem description 2
1.1.1 Deficiencies of existing datasets 3
1.1.2 Constraints for the airborne acquisition of datasets 4
1.1.3 Synthetic datasets as possible surrogate 5
1.2 Scientific question 11

2 State of the art and related work **12**

2.1 Computer vision algorithm development and evaluation 12
2.1.1 General approach 13
2.1.2 Evaluation of CV-algorithms 13
2.2 Datasets for CV- algorithms 16
2.2.1 Characteristics of datasets 18
2.2.2 Generation of natural datasets 18
2.2.3 Generation of synthetic datasets 21
2.2.4 Conclusion 24
2.3 Photographic vs. computer graphic imagery 24
2.4 Realism and fidelity in computer graphics 30
2.5 Image comparison 33
2.5.1 Image quality measures 34
2.5.2 Application driven measures 39
2.5.3 Content driven measures 41
2.5.4 Similarity and distance measures 49
2.6 Concepts investigating transferability of synthetically acquired results 51

3 Concept 55

3.1 General concept 56
3.2 Applied concept 59
3.3 Test object 63
3.4 Object performance evaluation 66
3.5 Image comparison algorithms 71
3.6 Image content distance measures 75
3.7 Influence factor analysis method 80

4 Implementation 89

4.1 Unmanned aerial system 90
4.1.1 Unmanned aerial vehicle / sensor platform 92
4.1.2 Software implementation 95
4.2 Synthetic environment 98
4.2.1 Software implementation 100
4.2.2 Generation of terrain database 103
4.3 Concept demonstrator 110

5 Preliminary experiments and experimental datasets 113

5.1 Preliminary experiments 113
5.1.1 Validation of database accuracy 113
5.1.2 Validation of auto-generated ground truth concept 115
5.1.3 Validation of automatic ground truth computation implementation 120
5.1.4 Validation of image content distance measures 123
5.2 Experimental datasets 126
5.2.1 Description of scenes 127
5.2.2 Synthetic environment configurations of test datasets 130

6 Principle experiments and results 139

6.1 Baseline experiments 139
6.1.1 Object performance 140
6.1.2 Image content distances 149
6.1.3 Influence factor analysis 151
6.1.4 Summary of baseline experiment results 165
6.2 Configuration set experiments 168
6.2.1 Configuration set “Illumination” 171
6.2.2 Configuration set “Texture” 178
6.2.3 Configuration set “Edge” 187
6.2.4 Configuration set “3D-Objects” 196

6.2.5 Configuration set “Camera model” 203

7 Discussion and design recommendations 211

7.1 Discussion 211
7.2 Design recommendations 215

8 Summary 221

9 Prospects 223

References 224

Appendix 237

A First questionnaire on computer graphic engines 237

B Telemetry-based homography estimation 245

C Regression models 247

C.1 Regression models with categorical predictor 247
C.2 General model fitting quality R^2 249
C.3 Regression model coefficients 250

List of abbreviations

3D	Three Dimensional
AA	Antialiasing
ADAS	Advanced Driver Assistance System
AF	Anisotropic Filtering
AHRS	Attitude and Heading Reference System
AI	Artificial Intelligence
AICD	Aerial Image Change Detection
ANOVA	Analysis of Variance
API	Application Programming Interface
ARMA III	Armed Assault 3
ART	Angular Radial Transformation
ASI	Application Scripting Interface
AToC	Alpha-To-Coverage
BISim	Bohemia Interactive Simulations
BRDF	Bidirectional Reflectance and Distribution Function
CAD	Computer Aided Design
CBIR	Content Based Image Retrieval
CEP	Circular Error Probable
CI	Condition Index
CLD	Colour Layout Descriptor
CLIF	Columbus Large Image Format
COTS	Commercial Of The Shelf
CSD	Colour Structure Descriptor
CSM	Cascaded Shadow Maps
CSS	Curvature Scale Space
CSV	Comma Separated Value
CV	Computer Vision
DCD	Dominant Colour Descriptor
DCT	Direct Cosine Transformation
DDS	Data Distribution Service
DEM	Digital Elevation Model
DIN	Deutsche Industrie Norm
DSM	Digitized Surface Model
DTM	Digitized Terrain Model
DU	Distortion Unaware
EHD	Edge Histogram Descriptor
EO	Electro-Optical
ESRI	Environmental Systems Research Institute
FAST	Features from Accelerated Segment Test
FDP	Functional Difference Predictors
FFT	Fast Fourier Transformation
FMS	Flight Management System
FOV	Field of View
FSIM	Feature Similarity Index Measure
FXAA	Fast Approximate Antialiasing
GCS	Ground Control Station
GIS	Geographic Information System
GPS	Global Positioning System
GSD	Ground Sample Distance
GT	Ground Truth
GUI	Graphic User Interface
HDR	High Dynamic Range
HMMD	Hue Max Min Diff-Colour Space
HQ	High Quality
HQ_NB	High Quality; No Buildings
HSV	Hue, Saturation, Value Colour Space
HTD	Homogenous Texture Descriptor
ID	Identifier
IEC	International Electro-technical Commission
IFS	Institute of Flight Systems
IHE	Image-based Homography Estimation
IQA	Image Quality Assessment
IQM	Image Quality Metrics
ISO	International Standards Organization
ISPRS	International Society for Photogrammetry and Remote Sensing
ITEM	Integrated Test bed for Experimentation on Mission Sensors
IW-SSIM	Information content Weighted Structural Similarity Measure
JND	Just Noticeable Differences
JPEG	Joint Photographic Experts Group
LIDAR	Light Detection and Ranging
LiPo	Lithium-Polymer
LOD	Level of Detail
LQ	Low Quality
LTS	Long Term Support
MAD	Most Apparent Distortion
MGT	Manual Ground Truth
MPEG	Motion Picture Expert Group
MQ	Medium Quality
MSAA	Multi-Sampling Antialiasing
MSE	Mean Square Error
MSER	Maximally Stable Extremal Regions
MSL	Mean Sea Level
MSSIM	Mean Structural Similarity Index Measure

MS-SSIM	Multi Scale Structural Similarity Measure
MTF	Modulation Transfer Function
MTOW	Maximum Take-Off Weight
MVG	Multi-Variate Gaussian model
NGSIM	Next Generation Simulation programme
NIQE	Natural Image Quality Evaluator
NR	No Reference
NSS	Natural Scene Statistics
OGC	Open Geospatial Consortium
OME	Offline Mission Editor
OMG	Object Management Group
OSG	Open Scene Graph
OU	Opinion Unaware
OVVV	Object Video Virtual Video
PBO	Packed Bohemia Object
PCA	Principle Component Analysis
PCF	Percentage Closer Filtering
PLCC	Pearson Linear Correlation Coefficient
PSNR	Peak-Signal-to-Noise Ratio
PTZ	Pan-Tilt-Zoom
RAM	Random Access Memory
RANSAC	Random Sample Consensus
RGB	Red-, Green-, Blue-Colour Space
RMS	Root Mean Square
SAR	Synthetic Aperture RADAR
SCD	Scalable Colour Descriptor
SDK	Software Development Kit
SE	Standard Error
SIFT	Scale Invariant Feature Transform
SMAA	Subpixel Morphological Antialiasing
SRI	Stanford Research Institute
SR-SIM	Spectral Residual Similarity Measure
SSAA	Super Sampling Antialiasing
SSAO	Screen Space Ambient Occlusion
SSD	Sum of Squared Distances
SSIM	Structural Similarity Index Measure
STAR	Solenoidal Tracker at RHIC
SURF	Speeded Up Robust Features
THE	Telemetry-based Homography Estimation
TI	Thermal Infrared
TV	Television
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UNIBWM	University of the Bundeswehr Munich
USAF	United States Air Force
USB	Universal Serial Bus
UTM	Universal Mercator Transformation
VBS2	Virtual Battle Space 2
VBS3	Virtual Battle Space 3
VDP	Variance Decomposition Proportions
VIS	Visual range of the electromagnetic spectrum
VIVID	Video Verification of Identity
VSM	Variance Shadow Maps
WGS	World Geodetic System
WIFI	Wireless Local Area Network
WLAN	Wireless Local Area Network
XM	Experimental Model
XML	Extended Mark-up Language
YCbCr	Lumina, Chroma Colour Space

1 Introduction

Since the advent of lighter-then-air balloons in the late 17th century and heavier-then-air aircrafts in the early 19th century a great variety of applications for aerial platforms had been found. While without doubt transportation of persons and goods always played the predominant role, using aerial platforms for earthbound observation was always of great interest. New perspectives and insights were gained exploiting the unobscured view and enhanced visual range from higher altitudes.

Military forces were among the first ones to utilize these new means of information gathering. Up to date knowledge about location, strength, armament and movement of enemy forces was and still is crucial to achieve specific objectives on the battlefield. With that the term of airborne *intelligence*, *reconnaissance* and *surveillance* (ISR) was founded (North Atlantic Treaty Organization, 2005). In parallel respective civil applications emerged either in the field of public security (e.g. traffic and infrastructure surveillance, search and rescue, border control) (Murphy & Cycon, 1999) or earth sciences such as geology, geography, ecology or hydrology, where respective data are acquired through remote sensing methods (Lillesand, Kiefer, & Chipman, 2015a).

While in early days observations were conducted using only human eyesight, soon cameras were mounted on aircrafts and used to enhance and document the achieved results. Today, depending on the application, we see a variety of different mono-, multi- and hyperspectral-imaging as well as range-measuring sensors being flown as *payload* on dedicated aircraft types (e.g. aircraft depicted in Figure 1-1 equipped with SAR, one IR- and two EO-cameras).

Figure 1-1: Unmanned aerial reconnaissance vehicle IAI Heron. The yellow fairing hides a synthetic aperture radar (SAR). Under the nose, the optical reconnaissance system is located. (Source: USAF)

In the field of assessing the gathered information meanwhile various methods for automated sensor data processing are applied to handle the vast amount of data sampled with today's systems. They derive results more objectively or speed up the data exploitation process as such. In this context and enabled through suitable processing components, we currently witness the migration from off-line data assessment (done after mission completion) on ground to highly automated (near) real-time sensor data processing on board the aircraft. This becomes even more important when looking at the demand for more autonomous mission execution in the field of unmanned aerial systems (UAS). Here, data gathered from airborne sensors are not only relayed to the ground but are also exploited for intelligent machine decision making on-board (Russ & Stütz, 2012).

The deployment of computer vision on airborne platforms is currently emerging. Respective technologies can be seen in the already mentioned fields of *remote sensing* and *ISR* but also for *guidance & control* tasks. Mission related applications for *ISR / remote sensing* purposes have been demonstrated in several areas. Some example applications are disaster management (Quaritsch et al., 2010), landslide investigations (Niethammer, Rothmund, Schwaderer, Zeman, & Joswig, 2011), photogrammetry (Gini et al., 2013), fire detection (Ollero et al., 2005) or moving target detection and tracking (Rudol & Doherty, 2008), (Nejadasl, Gorte, & Hoogendoorn, 2006), (Breckon, Barnes, Eichner, & Wahren, 2009), to only name a few. Examples for *aircraft guidance* are attitude computation (Demonceaux, Vasseur, & Pègard, 2007), collision detection / estimation (B. Cohen & Byrne, 2009), road-following (Frew et al., 2004), guidance in GPS denied environments (Granlund et al., 2000) or feature based guidance (Garratt & Chahl, 2008).

In any case visual processing on board airborne platforms puts great demand on hard- and software and brings along specific requirements on aircraft design due to limitations on weight, size and power.

1.1 Problem description

Imaging sensors, whether operating in the visible or infrared domain still constitute the most wide spread airborne sensor type for ISR and remote sensing purposes. To increase the level of automation in airborne remote sensing systems sampled data needs to be processed with *photogrammetric* and *computer vision* methods. The development of respective high-level data processing algorithms relies heavily on the availability of considerable amounts of

exemplary sensor data in various forms depending on concept and development approach. In literature such example data are commonly referred to as *datasets* (Szeliski, 2011), (Daniel Scharstein et al., 2014):

- **Prototyping Datasets** are used to test algorithms during development stage. They may be highly abstract to test limits of the mathematical model or contain a real example scene.
- **Learning Datasets** provide image cuttings used to train self-learning algorithms. These contain the object of interest on which, the algorithm shall be trained for (*positive samples*) as well as background only examples (*negative samples*).
- **Noisy Datasets** contain scenes of interest with increasing levels of noise to determine the robustness of the algorithm.
- **Test Datasets** are used to test the learning quality. These datasets often are a subset of the used learning dataset. Therefore, the contained images have already been used to train the algorithm and the success rate should be close or equal to 100%.
- **Evaluation datasets** are used to evaluate the performance characteristics of the designed algorithm. These datasets shall demonstrate real world capability and therefore need to be from several sources, heterogeneous in content and of high mount. Evaluation data should consist of similar test cases, not used in any dataset listed before. Alternatively, the final evaluation of algorithm can be conducted in conjunction with the operational sensor system on-board the target aircraft as part of the *qualification* test flights for the end-customer.

This list underlines the need for comprehensive versatile datasets in order to produce high-performance machine perception systems. Here, the problem arises that for airborne application, acquisition of such datasets is difficult due to reasons presented in the following subchapters.

1.1.1 Deficiencies of existing datasets

Generally, computer vision (CV) methods are tested against known datasets generated for specific application problems. A prominent example is face recognition, e.g. (P.J. Phillips, Hyeonjoon Moon, Rizvi, & Rauss, 2000).

Unfortunately, airborne image datasets for airborne applications are sparse. Currently a only few are available in the public domain e.g. VIVID (Collins, Zhou, & Teh, 2005), VIRAT (Oh et al., 2011) or CLIF (AFRL, 2007). These datasets are providing aerial images including persons and vehicles in a number of events or scenarios. VIVID and VIRAT additionally provide images in the visual (VIS) and thermal infrared (TI) spectrum. Yet, the variances of these datasets concerning vegetation, environment, weather, time and actors are small. Several papers (Thacker et al., 2008), (Bowyer & Phillips, 1998) or (Kondermann, 2013) criticize that evaluation against specific datasets does not indicate the generalization or universality of the algorithm and instead a wide range of different datasets should be used. This range of datasets needs to differ in their source, quality and scene content.

Aforementioned datasets typically provide interesting problem cases for specific computer vision problems, but are limited in terms of optical, scenic and terrain-based effects that have to be regarded during real world operations. Therefore, currently the total amount of available data is often insufficient to claim that tested performance will come close to operational performance. This lack of data is becoming even more severe, when learning algorithms are involved. Here, data for training, testing and finally evaluation should be separated to prove a functional algorithm and generalization of the learned model.

1.1.2 Constraints for the airborne acquisition of datasets

For reasons pointed out above, algorithm developers for sensor systems deployed on flying platforms regularly have to resort to datasets specifically acquired for their projects. This leads to additional challenges:

To begin with appropriate fixed or rotary wing aircraft have to be employed equipped with sensor systems similar to those intended for operational use (Hoogendoorn & Schreuder, 2005). Dedicated workstations, high precision navigational equipment and specific IT components need to be installed supporting and executing the sampling process. Besides these fixed costs, recurring ones have to be estimated covering aircraft operation and personal costs. Depending on nature of data and quality requirements the resulting total cost results typically in a 4 digit US$ amount per flight hour (Hranac, 2004). With the advent of unmanned aerial systems, costs can be expected to go down. However evidence for this is currently hard to find, particularly as operational regulations are still limiting the usability of these platforms (Bundesministerium für Verkehr Bau und Stadtentwicklung, 2012).

Once the technical system is set up, data acquisition planning has to be performed. Such needs to concern the following requirements:

Weather & atmospheric conditions: As mentioned above, robust algorithmic results demand considerable diversity in this respect within the datasets. So a considerable number of flights need to be conducted even to cover the most important situations. Specific effects of interest may be seasonal (e.g. snow coverage), hard to predict (e.g. haze, fog) or even contradicting to safe flight operations (e.g. heavy rain and hail).

Geographical area: Sensor systems and algorithms may need to be developed and qualified for use in dedicated geographical regions according to customer requirements. However, equivalent or identical geographical settings may not be readily accessible for the test data provider due to distance or safety concerns e.g. in case of military activities.

Scenario settings: Algorithms may aim to spot changes in topography and infrastructure (“change detection”) or detect and track static and moving objects (“target detection & tracking”). Such scenarios and object behaviour, if not existing per se, must be staged in variations, which can be complex and costly. Whenever humans are involved as actors, their consent must be sought and data security ensured.

Trying to comply with all these different requirements eventually leads to considerable cost, extends development time and often discards specific dataset content and conditions.

1.1.3 Synthetic datasets as possible surrogate

Aforementioned chapters provided several reasons why test flight data available is not sufficient to a degree to best train or test computer vision algorithms. When searching for methods to compensate this shortage in suitable datasets, the interest quickly focuses on computer-generated imagery. In the aerospace domain, the use of virtual simulation techniques to render out-of-the cockpit views is very common and established for pilot training simulators. Here, where false decisions can quickly lead to devastating results simulation provides quick, relatively cheap and most importantly safe alternative to real flights.

However, can this approach be expanded to the world of computer vision? Here, synthetic images are already used to test the underlying algorithmic paradigms and to identify their limitations, since its accompanied reference data allows efficient testing without error prone

manual image annotation. Nevertheless, when it comes to application evaluation, synthetic data are commonly criticized[1]. Arguments against these approaches are generally questioning the transferability of measured results to real world performance. This thesis shall investigate how data acquired by sensors differs from computer-generated imagery and analyses this disparity concerning computer vision performance.

Sensor data or aerial imagery in the context of this thesis stands for images depicting terrain from an aerial viewpoint in the visual spectral range recorded using a capture device (e.g. a camera). These images captured in the physical world will be named *photographs*, *real world images* or *natural images* from this point onward[2].

Before diving any deeper in the subject, first the term *synthetic data* needs to be specified, (McGraw-Hill, 2002) defines synthetic data as *"any production data applicable to a given situation that are not obtained by direct measurement"*. This however could also mean processed data, which was originally measured directly. Therefore, in the domain of computer vision the term synthetic data means, a sequence of computer generated imagery for testing or evaluation purposes. As synonyms to *synthetic data*, *computer-generated images* or *rendered imagery* will be used from this point onward.

Within this specification, synthetic data can have considerable differences in appearance. From completely abstract images to scenes of higher complexity displaying common daily life scenes and/or objects (McCane, Novins, Crannitch, & Galvin, 2001). Some are modelled to only appear realistic (Taylor, Chosak, & Brewer, 2007), while others are computed according to physical models (Longhurst, Ledda, & Chalmers, 2003) or (Vedaldi, Ling, & Soatto, 2010).

Previous research investigating realism in rendered images (Herzog et al., 2012), (James A. Ferwerda, Ramanarayanan, Walter, & Bala, 2008) or (Longhurst et al., 2003) mainly focused on the use of physical model based rendering, such as ray-tracing for highest levels of visual appearance. Using best available techniques to compare captured to synthetic datasets may

[1] Online Discussion, Stephan Irgenfried, *Synthetic datasets vs . real images for computer vision algorithm evaluation?*, ResearchGate, http://www.researchgate.net/post/Synthetic_datasets_vs_real_images_for_computer_vision_algorithm_evaluation2, [Last Accessed: 08.07.2015]

[2] Based on the definition of natural scenes in (Sheikh, Bovik, & de Veciana, 2005)

introduces an overhead of effort since provided level of quality may not even be needed to investigate CV-Algorithms.

Therefore, this thesis deliberately selects only commercial-of-the-shelf real-time rendering engines, which provides a medium level of visual appearance (concerning pre-rendered, ray traced images) to determine the minimum level of quality necessary to serve as a synthetic datasets providing valid results (e.g. ARMA3 from Bohemia Interactive in Figure 1-2). Using these engines, it is intended to identify the specific images properties discerning natural from synthetic images and minimize them with the lowest necessary effort. Results acquired with synthetic data will only be accepted when they are demonstrated to be transferable to natural data.

Figure 1-2: In-Game Screenshot of the military simulation game ARMA III by bohemia interactive

However, it should not be forgotten that the general idea of using synthetic data in algorithm development is to reduce the necessary amount of necessary real world images and not to fully replace it. Recent publications indeed identified synthetic data as an efficient tool for validation and evaluation of new algorithms (S. N. R. Meister, 2014), (Gschwandtner, Kwitt, Uhl, & Pree, 2011) or (Daniel J Butler, Wulff, Stanley, & Black, 2012). This allows the assumption that synthetic data has its place in development of computer vision algorithms.

So in summary, using image renderings instead of captured images in the early development stages should lead to a number of new possibilities and positive effects:

- **Amount of datasets:** When using computer-generated imagery to compile datasets the largest effort needs to be committed to the generation of the terrain database (modelling of terrain, roads, housing and vegetation). Having accomplished this,

numerous datasets can be created by using scenario editors, hemispherical lighting models (attitude of the sun), sensor perturbation models or weather effects. The limiting factors then are reduced to IT-equipment and human resources.

- **Scenario diversity:** As it was pointed out, diverse scenarios would enable developers to test the robustness of algorithms. Once the terrain database has been developed, depending on the rendering engine, scenarios can be put quickly together by scripting, editors or programming. (Hendriks, Tideman, Pelders, Bours, & Liu, 2010) propose a respective system as a development tool for advanced driver assistance systems (ADAS). The effort to record datasets of varying scenarios (actors moving according to a storyboard) is low concerning recordings of live staged scenarios, where complex scenarios are usually recorded once to limit the financial expenses.

- **Budget:** Synthetic data allows lowering the number of test flights and thus natural datasets, so results already achieved using synthetic data only need to be validated to prove the transferability. The efforts to create a realistic synthetic environment only need to be invested once in the beginning of the project. “Flights” in this environment then can be repeated efficiently and at very low cost.

- **Safety:** The lowered number of actual necessary real test flights reduces the risk of vehicle loss and accidents.

- **Determinism:** One of the major advantages in using synthetic data, is the possibility to conduct repetitive in-the-loop tests of a scenario, which provide deterministic results allowing reliable identification of errors.

- **In-the-loop simulation:** The use of real-time rendering engines enable software- and / or hardware-in-the-loop simulations. This allows testing of systems components, which shall react dynamically to simulated sensor outputs without risks for man and machine. The main advantages lie in the reduction of system complexity and testing time while increasing software quality (Nabi, Balike, Allen, & Rzemien, 2004).

- **Availability of reference data:** Reference data (*ground truth*) allows verification of information acquired via sensors and the quantification of algorithmic performance. (Lillesand, Kiefer, & Chipman, 2015b). When setting up datasets from physical sensors, ground truth needs to be created either by additional different sensors (automatic) or via (manual/semi-automatic) annotations. Ground truth acquired by

sensors is expensive since it needs far higher accuracy compared to the tested system (usually 10x). Manual annotation is also costly due to the extensive labour involved. also quality tends to degrade when scenes become complex and sub-pixel accuracy is demanded (Kondermann, 2013). Here, synthetic environments provide a feasible alternative since the 3D- to 2D-transformation process is known and ground truth can be retrieved automatically.

The aforementioned points highlighted the potential advantages provided by synthetic images to the development process of computer vision algorithms. Naturally, the usage of synthetic data also is accompanied by shortcomings that need to be considered:

- **Simplifications in 3D rendering:** In synthetic environments the number and detail of 3D-objects is limited compared to their real counterparts (e.g. vehicles, houses or trees). Further, direction and intensity of lighting is simplified to shorten processing times, while maintaining high visual quality. (Daniel Scharstein & Szeliski, 2001) criticize the low complexity of geometries and textures in synthetic datasets. This statement might be outdated, since nowadays the complexity of synthetic datasets is scalable. For example (Martull, Peris, & Fukui, 2012) remodelled the famous *Head and Lamp* dataset from (Nakamura, Matsuura, Satoh, & Ohta, 1996) in full detail.
- **Clear image composition:** According to (Vaudrey, Rabe, Klette, & Milburn, 2008) rendered images show more distinct outlines of objects than captured images. These provide high intensity gradients upon which many computer vision algorithms rely on. This effect could be compensated by *motion blur*, *depth-of-field* emulation or *antialiasing*. (Daniel Scharstein & Szeliski, 2001) also identified synthetic dataset as too “clean”, which is a texture modelling problem and has been identified and investigated by (Longhurst et al., 2003) in psychophysical experiments.
- **Missing optical effects:** (Daniel Scharstein & Szeliski, 2001) argue that camera distortions are seldom modelled in synthetic environments. This fact however changed in the recent years (Grapinet, De Souza, Smal, & Blosseville, 2012), (Hummel & Stütz, 2011), (Nentwig, Miegler, & Stamminger, 2012) or (Taylor et al., 2007).
- **Diversity of textures:** Textures in captured images are often highly diverse even when identical objects are present in the same image. For instance, rooftop tiles of

old buildings are highly heterogeneous due to environmental wear. In computer engines, textures are reused to increase performance leading to visual repetition.

- **Modelling efforts vs. degree of realism:** The effort necessary to create a virtual representation of a location existing in the real world depends on the requirements set on its realism. For example, humans can easily interpret the content of very abstract images. Similarly, CV-applications may not need physically realistic visual quality to perform successfully. However, the necessary content to provide a 'realistic' synthetic datasets is yet to be defined. Currently, modelling costs (work force, computation time) are the main limiting factor. Knowledge on the degree of modelling effort needed would help to create datasets on point, thus saving money.
- **Acceptance, Transferability:** The use of synthetic data to evaluate computer vision algorithms is still very controversial as mentioned in (S. Meister & Kondermann, 2011). Studies demonstrating the performance of specific algorithms on real and synthetic data exist, e.g. (Nentwig et al., 2012), (S. N. R. Meister, 2014) or (Wood et al., 2015), but yet have to identify the fundamental image properties influencing the performance.
- **Accurate modelling:** (Ellis, 2002) states that *"…accurate modelling of all the many factors to simulate a 'realistic' sequence still presents major problems in the field of computer graphics and animation."* This is true, but as long as the necessary factors and requirements to stimulate tracking algorithms are not or only fuzzy described, accurate modelling cannot exist.

This chapter presented advantages and drawbacks of using synthetic data for evaluation of computer vision algorithms. It has been shown that identifying image properties on which computer vision algorithms are sensitive to, is of major importance to create synthetic test datasets efficiently that produce results comparable to real data.

1.2 Scientific question

In the previous chapter the general need for airborne datasets, their specific requirements and problems with existing ones have been described and the use of synthetic datasets has been proposed. Therefore, this work aims to investigate on the following general question:

How can synthetic datasets for development of computer vision algorithms be designed and generated to achieve performance results transferable to real world conditions?

More specifically this question can be separated into three objectives. This thesis shall draft, implement and execute appropriate experiments to

Objective 1) quantitatively asses **performance differences** of selected CV-algorithms on synthetic and natural datasets,

Objective 2) identify inducing **image and rendering properties** and eventually

Objective 3) formulate **design recommendations**, which support database modelling engineers and simulation system manufacturers in providing suitable synthetic datasets.

Each of these objectives formulate capabilities necessary to identify the underlying reasons for synthetic imagery to perform differently to natural imagery. If the concept presents methods and metrics to successfully demonstrate and / or measure these capabilities, the objectives are fulfilled.

2 State of the art and related work

The topic of this thesis touches domains such as computer vision, computer graphics, remote sensing, image retrieval and aerospace engineering in an interdisciplinary way. The necessary background is provided in the following chapters together with the state-of-the-art analysis. The first subchapter presents an introduction into computer vision development and best practices for the evaluation of CV-algorithms. In chapter 2.2 the characteristics and generation methods of natural and synthetic computer vision datasets are presented. The next chapter presents the image acquisition pipeline and the differences of natural and artificial image generation. Current efforts discussing realism and fidelity in synthetic images are presented in chapter 2.4. The following chapter presents published image comparison metrics categorised into image quality-, application- and scene-based metrics. Additionally, the mathematical similarity and distance measures employed by these metrics are presented. Most researchers often do not acknowledge scientific results solely acquired using synthetic data due to reasons presented in chapter 1.1.3. After identifying current concerns, existing research addressing the transferability of synthetically acquired results on to the real world is discussed in chapter 2.6. The last subchapter of the state of the art analysis presents published procedures for performance characterization of airborne mission sensors.

2.1 Computer vision algorithm development and evaluation

Airborne *Intelligence, Surveillance and Reconnaissance* (ISR) applications address the detection, tracking or identification of physical objects. To automate this process Computer Vision (CV) methods can be used. CV as defined by (McGraw-Hill, 2002) is a technical field focusing on acquisition, extraction, characterization or interpretation of information in digital imagery of a three dimensional world. While humans are able to perform aforementioned tasks (e.g. detection of persons in images), performance of computer vision algorithms is not yet up to par. CV methods reconstruct information from images the provide insufficient information of the depicted 3D-world by designing explicit solutions based on probabilistic, physics-based or mathematical models. These models used in CV are usually developed in physics or computer graphics, e.g. computation of light scattering and reflection, position and movement of objects, or camera lens distortion (Szeliski, 2011). This inverse relationship between computer vision and computer graphics will be exploited in the concept of this thesis to evaluate CV-algorithms. (Berger, Levine, Nonato, Taubin, & Silva, 2013) for instance also

uses both worlds to evaluate their algorithm. Before diving into specific evaluation methods of algorithms the general approaches of development and evaluation are discussed.

2.1.1 General approach

According to (Szeliski, 2011) current CV-algorithm development processes can be categorized into three high level approaches:

- **The scientific approach** is based on the understanding of physical principles necessary for image formation. These are analysed and modelled in order to invert the model to obtain the desired scene description from acquired pictures.
- **Within the engineering approach,** the problem needs to be defined and the validity of basic assumptions and goals needs to be questioned. Then alternative solutions are implemented and tested based on defined metrics. Afterwards tests in real-world conditions lead to the most promising concept. This approach is focussing on testing during the development phase, and therefore in need of relevant test cases and large datasets.
- **The statistical approach** uses large training datasets to learn probabilistic models coping the worlds and the image formation process' uncertainty. After learning, these models allow estimation of results and quantification of their uncertainty.

These approaches coexist, since depending on the problem to solve each approach has its advantages and disadvantages. The reference process used in this thesis has been developed along the engineering approach. This thesis focuses on the evaluation of the algorithms detection qualities. Performances such as speed or usability are not investigated, since these are independent to the nature of used testing data. The following chapter will discuss the general concepts of CV-algorithm evaluation and presents the role of synthetic data.

2.1.2 Evaluation of CV-algorithms

According to (Bowyer & Phillips, 1998) no methods for empirical evaluation of computer vision algorithms were commonly accepted until the mid-nineties. The authors' state, "Evaluating algorithms lets researchers know the strengths and weaknesses of a particular approach and identifies aspects of a problem where further research is needed". They divide possible approaches in three categories to introduce standardized evaluations allowing comparison of algorithms:

- **Independently administered evaluations:** Here, an external group set up test datasets, designed the evaluation method, and provided these to the algorithm developers. The measured results are then sent back to the group for evaluation. This approach provides objective results since dataset and methods are published and tested by independent persons. The drawback is the evaluation group's high workload.
- **Externally conducted evaluations:** An external evaluation group collects and evaluates all algorithms of interest on their own. When original implementations are not available, algorithms are implemented based on literature. This produces a baseline where state-of-the-art algorithms can be compared. Often implementations are not available, raising the effort for the evaluation group.
- **Evaluation concepts for non-self-evident ground truth:** Here, "a major part of the evaluation process is to develop a method of obtaining the ground truth" (Bowyer & Phillips, 1998), followed by an evaluation against the newly defined metrics.

Today, depending on the algorithm the most suitable approach is selected. For instance algorithms of common problems, demonstrate their increased performance against well-known datasets as suggested in the first category. Prominent examples are the *Hamburg Taxi Sequence* (Dreschler & Nagel, 1982) for optical flow or the *Middlebury* datasets for stereo matching (D. Scharstein & Szeliski, 2003). Due to this focus on publicly available datasets, it may happen that algorithms are tuned towards high performance in these scenes while disregarding their general performance ("overfitting"). Therefore, (Szeliski, 2011) presents an evaluation strategy based on three levels:

1. Evaluation on clean synthetic data with known ground truth.
2. Evaluation on noisy synthetic data with known ground truth.
3. Evaluation on an extensive amount of real-world images from a wide variety of sources (locations, cameras, lighting conditions).

This approach specifically asks for synthetic data to enable the comparison with known ground truth data. The use of synthetic data is however questioned because the transferability to the real world is unknown (Bowyer & Phillips, 1998). Therefore level two (for robustness against noise) and level three (real-world transferability) complete this strategy.

In (Barron, Fleet, & Beauchemin, 1994) the methods for optical flow estimation are evaluated using an externally conducted approach. Ten different techniques are compared using four

synthetic and four real datasets. This corresponds to level one and three of Szeliski's evaluation strategy. According to the author, synthetic datasets have been used since motion fields and scene properties can be methodically controlled and tested. This method was criticized by (McCane et al., 2001) due to selection of abstract synthetic image sequences. Thus, the authors created synthetic datasets with several levels of complexity with ground truth as synthetic and real datasets. The implementations of seven tested algorithms were collected from prior research (Barron et al., 1994) or self-implemented. These showed consistent behaviour on synthetic and real datasets, leading the authors to conclude the validity of their test approach. Absolute results on real datasets presented low performance for almost every algorithm. The existing performance differences between synthetic and real data are not discussed. To improve the evaluation approach the authors suggest that future algorithm developers should publish their code, evaluate on existing standard benchmarks (datasets with ground truth) using standard metrics and enter their results in a public central database. The paper emphasizes the usefulness of synthetic datasets in evaluation due to available ground truth and scalability of synthetic scenes.

The literature survey of (Thacker et al., 2008) analyses the use of evaluation and validation techniques in the computer vision domain. It first traces back the frustration of system designers about the unreliability of computer vision algorithms to deficits in evaluation. The authors acknowledge the efforts of the last 20 years, which improved the quality of algorithms by dataset sharing, code sharing and comparative testing. However, they still see potential for improvement by exploiting the techniques of probability theory and quantitative statistics.

To characterize algorithm performance (P. Jonathon Phillips, Martin, Wilson, & Przybocki, 2000) distinguish between two types of evaluation:

- *Technology evaluation* regards the characteristics against changing conditions of the image acquisition such as noise or contrast.
- *Scenario evaluation* concern the behaviour of the system with regard to specific use cases and application (e.g. recognition).

The authors then review the CV-domain by selecting specific well-known topics to discuss current testing methods, available ground-truth datasets and suggest possibilities for improvement. They conclude that algorithm performance cannot be extrapolated on unseen datasets and scenario evaluation should be conducted to estimate the usability. They believe that single isolated datasets cannot provide a general means of algorithm performance

evaluation. Thus, simulation of imaging system and environment is proposed to identify the key factors of data variability and the expected response of the algorithm. This simulation can be conducted empirical or analytically but should be statistically calibrated using test datasets.

(Thacker et al., 2008) provide a comprehensive view on specific computer vision domains and review the evaluation from a statistical point of view. They encourage computer vision researchers to deepen their statistical knowledge to increase objective evaluation leading to more stable algorithms. The authors conclude that apparently the statistical nature of vision problems is often unknown to the researcher and has not been considered. The more holistic algorithm characterization desired by the authors could be conducted by introducing a sensor and environment simulation when calibrated using natural test data.

This review on algorithm evaluation paradigms show that it is still an energetically discussed, emerging field. Most importantly, algorithms should be considered to work only on the applications they have been created for until experiments or demonstrations suggest otherwise. In addition, the comparison of novel algorithms against existing publicly available evaluation datasets and the publication of the source code are considered as current best practices for well-known algorithm categories (e.g. stereo vision) (Thacker et al., 2008). Additionally, the evaluation should also include engineered dataset (real data with synthetic perturbations) to demonstrate its reliability against technical issues such as noise or lighting conditions (Szeliski, 2011). The last step is to test the method against a variety of data from different sources to measure its robustness against the diversity of the real world (scenario evaluation). Adhering to these recommendations leads to high testing efforts for developers. Therefore, researchers often contemplate with synthetic environments to demonstrate mathematical validity of the method and show its real world capability on one specific standardized dataset. (Thacker et al., 2008) introduced the idea of a sensor and scenario simulation to acquire the general performance on algorithms by varying both. The simulation shall be calibrated against real datasets to replicate their statistical properties.

2.2 Datasets for CV- algorithms

The previous chapters presented the importance of datasets for CV-algorithm development. This chapter discusses the characteristics of reference *datasets* necessary to evaluate, test or train CV-algorithms for desired use cases. Further methods to generate natural and synthetic datasets are presented.

According to (Kondermann, 2013) reference datasets can be categorised into:

- *Reference data without ground truth*: Data without any information about results of interest. Performance estimation is conducted subjectively via human interpretation.
- *Reference data with weak ground truth*: Data including desired results in insufficient accuracy (measurements accuracy is less than an order of magnitude more accurate than the algorithm). These results can be acquired automatically using other sensors (e.g. inertial measurement units) or manually via annotation.
- *Reference data with ground truth*: Data and results in sufficient accuracy. These sets can be acquired by high precision sensors (e.g. LIDAR) or by creating a synthetic dataset, which allows direct extraction of the ground truth.

The Author then discusses these categories concerning benefit, effort and quality. For instance, it is reasoned that *reference data without ground truth* can be acquired cost effectively; however, it cannot be used to evaluate the reliability of an algorithm. This task needs available ground truth, which always requires a certain amount of time and money. The estimated costs of (Kondermann, 2013) are subjective and provided in incomparable units, but still help to identify the most suitable method for individual application. The potential of using computer-generated graphics to create reference data is highlighted. The problem to transfer these results to real world is also mentioned. In (S. Meister & Kondermann, 2011) a comparative test conducted with a simple synthetic scene in a controllable environment using optical flow shows promising results. In (Baker et al., 2010) the results of the famous *Middlebury evaluation dataset*[3] for optical flow algorithms are presented. The dataset consists of twelve sets for training and evaluation of various kinds, e.g. real images of non-rigid moving scenes, real images of rigid scenes, realistic synthetic imagery, etc. The results of 24 optical flow algorithms are evaluated by using new and common error metrics. Having different types of datasets allows improved insights on the characteristics of algorithms. The provided statistics furthermore allow identification of the reasons for the resulting behaviours. Evaluations such as these help to identify the most suitable algorithm for specific applications for engineers and identify areas of potential improvement for researchers.

In the last 20 years, many datasets emerged for most of the topics in computer vision. Airborne datasets have also been generated in limited amount (e.g. ISPRS Benchmark

[3] Website providing datasets and results: http://vision.middlebury.edu/flow/ [Accessed: 07.09.2015]

(Nex et al., 2015)). These often cannot fulfil the best practices for evaluation as presented in chapter 2.1.2 (such as public availability, diversity, designed for the desired use case). For example, the VIRAT dataset consists of a 4h single take video stream of one location on one day (Oh et al., 2011). This shortage of data leads to slow research of airborne cv-algorithms compared to ground-based methods, since acquisition of airborne reference data is still very demanding and complicated. The following subchapter discusses the necessary features of a high quality evaluation dataset followed by subchapters about generation of airborne datasets. The last subchapter closes the dataset discussion with a short conclusion.

2.2.1 Characteristics of datasets

Certain characteristics of a dataset for evaluation purposes are fundamental for correct results. For instance, when a dataset with blurred edges (defocused) is used to evaluate an edge detector, the results will only be valid for exactly this type of images. Thus, (Thacker et al., 2008) proposes a statistical evaluation of reference data to identify the effect the algorithm has by comparing the probability distribution of input and output. (Haeusler & Klette, 2010) analysed nine different datasets for stereo matching by using common known feature detection and matching algorithms to identify the difference of these datasets. Testing against datasets with varying complexities is interesting since it shows the algorithms performance behaviour (e.g. covariance between performance and complexity). (McCane et al., 2001) also provided datasets of different complexity (synthetic and real) to benchmark algorithms. A more enhanced test was introduced by (Baker et al., 2010), incorporating many of current best practices for CV-algorithms evaluation. Benchmarking using diverse datasets accompanied by evaluation statistics improves the characterization and helps to identify algorithmic limitations. This shows that dataset generation should not only create reference data for evaluation but also provide its probabilistic statistics and should be of varying complexity, with and without perturbations.

2.2.2 Generation of natural datasets

Before creating a dataset, use case and purpose of investigation need to be defined. The simplest type of dataset consists of images without ground truth, (Kondermann, 2013) suggests this type of data to be useful for a proof of concept of a novel category of algorithms. Datasets without ground truth can easily be created in high amounts and subjective analysis by the researcher then identifies whether the algorithms works or fails in the recorded situation. This approach should be followed by further analysis using referenced data. In

contrast, the generation of natural datasets with ground truth can get expensive in terms of time and budget. Of course, each ground truth acquisition method is strongly dependent on the algorithm and subsequently on the data that needs to be acquired. In the following three different state of the art approaches to generate natural reference data are presented using specific examples:

- The *Middlebury optical flow datasets* (Baker et al., 2010) depict compact scenes in a staged laboratory environment populated with multiple objects (e.g. toy figures, plants). These objects are painted with ultra-violet paint and are moved by mechanical actuators to change the scene in small steps. The paint references the scene allowing the generation of ground truth in sub-pixel accuracy.
- For stereo images of static scenes the approach of (Daniel Scharstein et al., 2014) allows acquisition of high resolution images with ground truth by projecting structured light onto the scene to label correspondences in the two views. This and the previous approach are limited to a laboratory environment, which cannot be used to generate airborne image content.
- (Mikolajczyk et al., 2005) used an efficient method working in natural environments, where the datasets are recorded using a digital camera on a tripod. The (weak) ground truth is computed by assuming a homographic relationship between images, which allows depiction of plane surfaces only or limits the camera motion to rotation and zoom.

Aerial imagery differs strongly in regard to perspective, distance, movement or lighting from ground-based datasets. These differences need to be considered in design and generation of test and evaluation datasets. The ISPRS Benchmark (International Society for Photogrammetry and Remote Sensing) of (Nex et al., 2015) has collected data for remote sensing applications. Two buildings have been photographed and measured with multiple sensors and from several perspectives. The dataset consists of professional airborne images shot by a manned aircraft with a highly specialized camera; UAV based aerial images using a small lightweight camera and terrestrial images. Terrestrial and airborne laser scanners together with geo-coordinate measurements of several hundred locations are used to generate the Reference Data. The dataset is currently only partly published and provides ground truth for dense image matching and image orientation. This dataset allows algorithm tests in a natural urban environment. Since, the purpose of the dataset is to provide evaluation means

for remote sensing issues all images have been recorded at sunny weather. Therefore, environmental effects (e.g. lighting, fog) or sensor artefacts (e.g. noise, distortion) are not covered in this dataset, disabling its use in measuring the robustness of algorithms.

During the *next generation simulation program* (NGSIM) (Alexiadis, Colyar, Halkias, Hranac, & McHale, 2004) near ground aerial datasets of several highways in the United States have been recorded using stationary pole mounted cameras. The focus of the project is to acquire driver behaviour to establish and improve traffic simulation. The use of these datasets for evaluation of airborne computer vision algorithms is limited due to the missing ego-motion of the sensor and its focus on streets. Moving vehicle detectors and trackers may use this data due to the provided vehicle trajectories. Also in this dataset, environmental perturbations are not considered. During the project existing means of data acquisition for driver models where evaluated, presenting methods and necessary financial efforts (Hranac, 2004). For instance, the use of helicopters has been considered, but the spatio-temporal limitation of the platform disabled the possibility of holistic vehicle trajectory survey.

The *VIRAT Dataset*[4], a benchmark for object detection, object tracking and event recognition in surveillance videos using COTS surveillance equipment is presented in (Oh et al., 2011). The dataset provides terrestrial and airborne image data of several scenes depicting different events. The aerial dataset is a video stream of four hours length and depicts the view of an aerial platform circling above warehouses and military vehicles in the visible and thermal infrared spectrum. Ground truth is provided as manual annotation of persons, vehicles and events. The video stream includes also perturbations such as changing viewpoints, illumination and zooming and stabilization issues to test the robustness of computer vision algorithms. On the other hand, data was record on one day with good weather conditions eliminating evaluation of weather-induced perturbations. This benchmark categorizes imagery in training-, test- and evaluation-data, where latter consists of scenes not contained in the other two to reduce the possibility of overfitting.

During the *Video Verification of Identity* (VIVID) program a benchmark for tracking algorithms was published (Collins et al., 2005). The benchmark consists of eight aerial scenes in the visible or thermal infrared spectrum. The datasets provide a range of images that differ in resolution, contrast, and occlusion. Again, weather or sensor perturbation are not

[4] Website of the VIRAT dataset: http://www.viratdata.org/ [Last Accessed: 10.09.2015]

considered. The ground truth consists of bounding boxes around the objects to track and their detailed contours both acquired using human annotation. An evaluation program and example baseline algorithms accompany the referenced dataset[5].

Designing and recording data for benchmarks requires significant efforts and human labour. For airborne applications, this is increased by the need of an aerial sensor platform. Another significant component is the acquisition of reference data for evaluation purposes, which introduces additional efforts for natural airborne datasets where automatic annotation methods are difficult to apply. Numbers for complete costs of datasets are rarely presented, but (Kondermann, 2013) provided some estimates of known datasets (e.g. *Middlebury optical flow dataset)*. Further, he estimates the costs for labelling one image from ten to forty dollars. In (Vondrick, Ramanan, & Patterson, 2013) the cost and quality of manual image annotation is also investigated. They conclude that cheap purely manual annotation is error prone and needs to be assisted by intelligent tools reducing the workload of the workers. Generating ground truth for airborne imagery is especially difficult since usual methods applied in lab environments cannot be deployed.

2.2.3 Generation of synthetic datasets

Synthetic datasets in their simplest fashion have a long-standing tradition in computer vision. Usually these purely abstract images are used to validate the mathematical model of algorithms against their mathematical concept (e.g. grayscale sinusoidal images) and test their limits. Later on, more realistic synthetic datasets where generated to create test data more closely to the desired use cases. One of the first of this kind is the *Yosemite* dataset produced by Lynn Quam at the *Stanford Research Institute* (SRI), which became popular due its usage in the paper of (Barron et al., 1994). It was produced to test optical flow algorithms and consisted of sixteen grayscale images with corresponding ground truth. Computer graphics and computer vision are inversely related. Combining these two fields provides the possibility to validate the implementation against a known depicted scene (ground truth available). In recent years computer graphics have reached the quality that allows realistic impression of still life scenes (Szeliski, 2011), while the representation of living beings are still perceived as artificial (see the *uncanny valley* (Mori, MacDorman, & Kageki, 2012)).

[5] Website of the VIVID dataset: http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html [Last Accessed : 10.09.2015]

One of the first airborne synthetic datasets was *RADIUS* (Thornton et al., 1994), which has not been computer rendered but consists of model-boards depicting a scaled down landscape for airborne 3D-reconstruction, image feature extraction and classification. These boards, enriched by landscape, streets and polyhedral 3D-buildings are photographed using a digital camera. Therefore, only the scene and not the image acquisition is synthetic. The ground truth consists of manual annotations classifying the objects present in the scene. Using this reference data, several statistics are compiled to identify and tune to be applied edge detectors.

The *Middlebury optical flow evaluation dataset* (Baker et al., 2010) includes the *Yosemite* scene into their dataset collection, containing six more synthetic scenes. These scenes also depict urban environments and close ups of vegetation. According to the authors, these scenes are designed to provide complex test cases with significant occlusion and to have full control over the image generation process (e.g. lighting). The main advantage of synthetic scenes is obviously the simple extraction of ground truth. In this case, a custom shader extracts the motion between two images. Even though the authors speak of realistic synthetic datasets, the term realistic is neither discussed nor defined.

The previously presented datasets are of limited size (small number of images). With the generation of the *SINTEL dataset for optical flow* (Daniel J Butler et al., 2012) an extensive and purely synthetic set has been published. It consists of 35 selected clips of the open source short film *Sintel*. These clips add up to 1628 referenced frames separated in test and training. Reference data are provided via two dimensional flow fields. The clips have been selected to address all major topics of optical flow and to provide a varying dataset. Therefore, scenes of various difficulty with a sequence length of 50 frames have been taken. The designers of this database compared synthetic scenes with respect to content lookalike ‘natural’ scenes from movies or TV shows. By using image and flow statistics as evaluation criteria, they conclude that their dataset is more complex and closer to reality than the aforementioned *Middlebury dataset*. Additionally, it is stated that it is *“sufficiently rich to be a useful challenge for the community and that algorithms that are successful on Sintel are likely to be useful on a relatively rich class of natural movies”* (Daniel J Butler et al., 2012).

The Tool *Object Video Virtual Video* (OVVV) presented by (Taylor et al., 2007) is a simulation test bed for public surveillance scenarios. It is based on the game engine *Source* from *Valve Software* and is able to provide rendered imagery of stationary active Pan-Tilt-

Zoom (PTZ)-Cameras. The imagery can be perturbed with noise, ghosting, distortion and camera jitter. S*uper sampling* (SSAA) is employed to reduce aliasing. Physical cameras have no aliasing due to the smoothing effect of imperfect camera optics (optical resolution of optics; see *modulation transfer function* (MTF)) or *optical low pass filters* mounted in front of the image sensor. The presented framework also provides ground truth for persons in images in form of bounding boxes and segmented foreground images. Scenarios are created using the tools of the engine itself, these can be scripted or AI controlled. The authors also show the possibility to create natural terrains using satellite imagery and *digital terrain model*s (DTM). They compare their framework against existing natural scenarios of (Toyama, Krumm, Brumitt, & Meyers, 1999) by loosely reconstructing the same indoor scenes. The resulting differences in performance between these two scenarios are justified by differences in implementation and image content. The authors state that *"the performance of algorithms across different scenarios and relative performance in the same scenario is generally preserved using real and synthetic sequences"* (Taylor et al., 2007). The possibility to simulate an UAV camera footage is mentioned but not detailed. Essentially the used game engine limits the size of scenarios, provides no dynamic flight model and restricts the resolution of the satellite image based ground texture. The total map size is limited to 2^{15} Units[6] in each direction with 52.48 units being one meter[7] leading to a maximum terrain size of 600 by 600 meters, which constrains its application for airborne scenarios.

The *Aerial Imagery Change Detection* (AICD) dataset of (Bourdis, Marraud, & Sahbi, 2011) dataset has been created using the serious game engine *Virtual Battle Space 2* (VBS2) from Bohemia Interactive Simulations (BISim)[8]. The engine allows terrain sizes of up to 20 by 20 km. AICD has been made for evaluation of algorithms detecting local changes in the terrain by comparing images shot at different times. The dataset consists of one hundred scenes depicted in five different viewpoints with and without illumination changes. Hard shadows and occlusions, both prominent problems for change detection, are omitted. Ground truth is provided as a mask image depicting the location of the change in white on black unchanged background. The method to generate ground truth is not detailed, but the description suggests manual annotation. Even though the authors state the dataset is "realistic", no comparisons to

[6] As defined in forge game data (FGD) definition file base.cfg: http://therazzerapp.de/fgd/base.fgd [Last Accessed: 16.09.2015]

[7] 1 Unit is defined as 1ft: https://developer.valvesoftware.com/wiki/Dimensions [Last Accessed: 16.09.2015]

[8] A serious game development company https://bisimulations.com/company/news [Last Accessed: 16.09.2015]

real data confirm this statement. The presented work produced a dataset tackling the issue of change detection especially focussing on parallax effects.

2.2.4 Conclusion

In the beginning, it was identified that reference data with ground truth is necessary to evaluate the performance of computer vision algorithms. This data needs to meet the necessary requirements of the use case, leading to the need of specific benchmarks for individual research problems. The reference data itself should also be analysed to identify the range of possible situations covered. The limiting factor for design and generation of new reference dataset are the involved costs in staging scenarios that provide ground truth and represent the actual intended use case. In case of airborne applications, the costs are amplified even further. Natural datasets provide photographs most closely resembling later use cases. Generating airborne natural datasets is expensive. Thus, usually these dataset are limited in amount, versatility, covered area and complexity. Additionally, complex technical solutions or high manual effort is necessary to generate the necessary ground truth.

Synthetic datasets have already a long tradition of being used in computer vision. However, they are still mostly of abstract nature. Today’s rendering technology now allows the generation of sophisticated and complex scenes. Synthetic dataset are cheaper to produce while the versatility and amount of data is higher compared to natural data. Direct availability of ground truth further increase the efficiency of synthetic data. However, results acquired via rendered imagery may differ to those from natural data concerning absolute values. Therefore, even though synthetic data could be a promising means for efficient evaluation of (airborne) CV-algorithms, it needs to be identified whether the results can be transferred to the real word and which image differences exactly affect the tested algorithm.

2.3 Photographic vs. computer graphic imagery

After having discussed the availability and building process of natural and synthetic datasets, it is of interest to identify the cause of similarities and differences between these image types. Comparing the real world with a rendered image or a photograph is difficult due to the presentation systems involved. Equal comparison providing the same conditions needs to be conducted on the same system towards the receiver, which can be a human or a computer vision algorithm. Therefore, the comparison of images needs to be based on a format allowing

comparison of photographs and rendering engines. A possible format are 2D representations (images) of real world and synthetic 3D environment. The necessary toolchain to acquire images can be simplified to following entities: Scene, imaging system, presentation system and receiver as depicted in Figure 2-1. The Scene is a 3D space in which position, orientation and scale of all objects and the surface are defined. The imaging system converts this 3D space into a 2D representation from a specific point of view. The imaging system also digitizes the image in a *rasterization* process into a pixel-based matrix. This digital image then can be displayed using a presentation system (in this example a monitor) to convert it into a perceivable format. In case of a human observer, this would be radiance. While this explanation of the generic imaging workflow only provides a high-level idea, the whole process is much more complex. Each element of the flow can be different and this example is focussed on the first elements scene and imaging system, because here differences between the two image types are introduced. For instance when comparing synthetic and natural images the observer can be an objective evaluation method (e.g. *Peak-Signal-to-Noise Ratio* (PSNR)). The presentation system in this case is a digital image format. Thus, differences introduced by imaging system and scene are measured. These differences result from specific physical or mathematical characteristics of these two components. The imaging pipelines for photographic and computer graphic imagery are detailed in the following. Since the observer is not a part of the image generation process, it will not be considered in these descriptions.

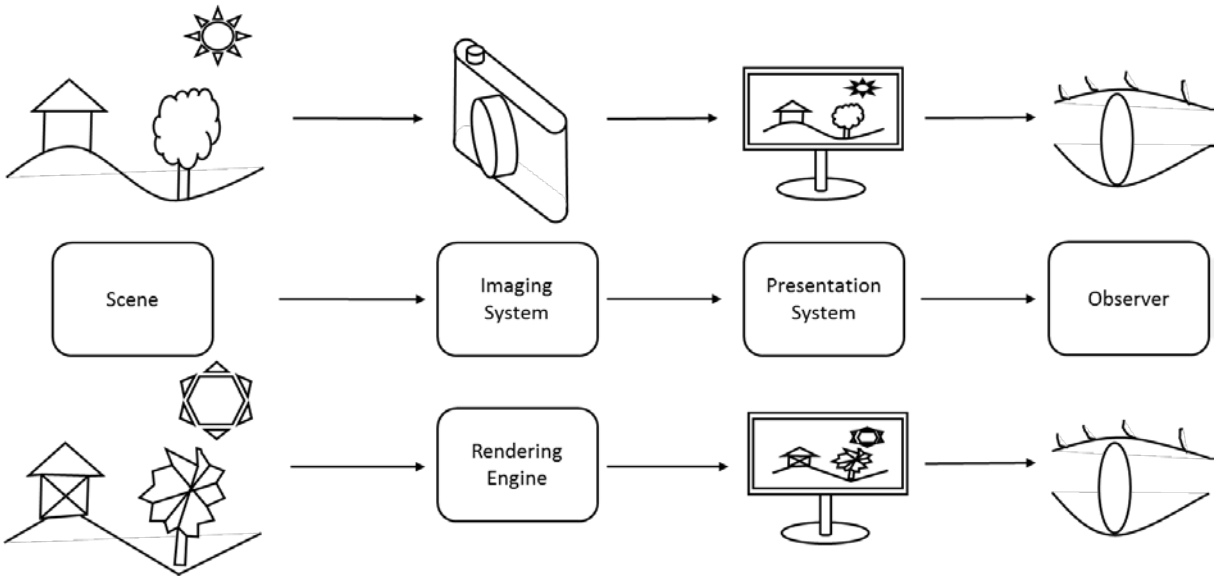


Figure 2-1: The generic imaging workflow (middle) and two examples: Digital photography (upper) and computer generated imagery (lower). Both examples focus on the first section of the workflow and thus use common a presentation system (monitor) and observer (human).

Imaging sensors are passive, capturing radiance of the ambient electromagnetic spectrum. Thus, lighting characteristics directly influence the resulting image. Spectrum, orientation (e.g. spotlight), position and intensity of all light sources within the scene establish the framework for the image. Every object is only presented by a reflection, diffusion, transmission or absorption of the light sources radiances. For instance, a green apple illuminated by a full spectrum light source appears green since this section of the spectrum is reflected while the rest is absorbed. In Figure 2-2 the first half of the imaging pipeline for digital photography is presented. Here, a light source illuminates a scene consisting of several objects. The reflection properties of each object are defined by its form, material (see bi-directional reflectance distribution function (Nicodemus, 1965)) and size. Further, objects are positioned on terrain, which can be an empty field. A part of the light reflected by objects and terrain enters the camera systems optics, which focuses the received radiance on the imaging sensor. Optics define the maximum angle of light also called field of view (FOV) that can be redirected onto the sensor, the amount of light being transmitted to the sensor (aperture) and the distance where objects are sharply reprojected (focus). On the other hand, optics can also introduce imaging errors and side effects such as blur (scattering the light of the object too much), chromatic aberration (scattering the light depending on its wavelength) and optical distortions.

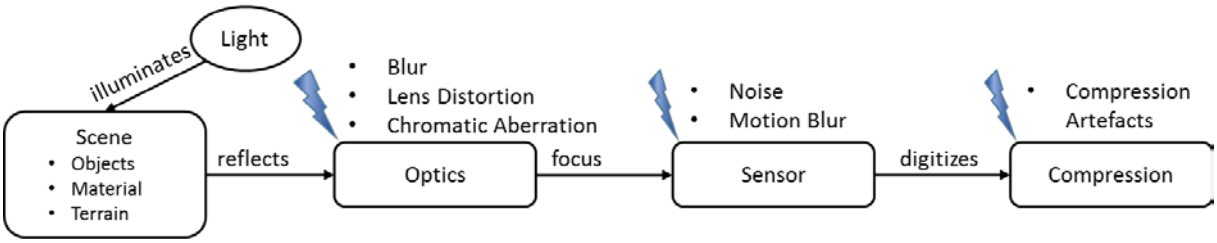


Figure 2-2: Imaging pipeline for digital photography.

The scene is projected through the optics onto the actual sensor, which transforms the incoming radiance into a discrete picture element (in short pixel) value. This transformation also introduces unwanted distortions such as noise for instance that is a statistical variation of intensities during the quantification process resulting from various physical effects (Farooque & Rohankar, 2013; STEMMER IMAGING GmbH, 2013). Another imaging error arises when an object within the scene has a significant speed concerning the exposure time of the camera. It will then appear blurred, which is called *motion-blur*. The last step, compression, is not mandatory and is used to save memory space or bandwidth. Depending on compression quality and method, image information gets lost and further artefacts are added. The nature of

such artefacts depends on the used compression method. The resulting digital image is eventually saved in a shareable image format.

The workflow of the synthetic image generation process has the same structure but differs in terms of implementation compared to the natural image acquisition. The scene also depicts objects and terrain but these are modelled with 3D points (vertices), which are connected to two other vertices forming a triangle plane called polygon. The modelling artist uses these polygons to form the desired objects (wired mesh). These are then coloured and / or textured using UV mapping, which maps a 2D image texture on the 3D mesh to give the object more detail. For each surface, the light reflection is defined depending on the represented material. This combination of mesh (geometry), texture (colour and details) and material (reflection) forms an object. All objects are archived in the *model database*. Surrounding terrain is defined by a 3D mesh. Often satellite imagery is mapped on this height grid to texture the terrain. Due to memory limitations, the terrain texture is often limited in resolution. This leads to low resolution terrain textures when the camera is close to the ground. Thus, sometimes procedural detail textures are employed to detail the surface indicated in the satellite image (e.g. grass or tarmac) (Roupé & Johansson, 2009). Detail textures, satellite imagery and height are encoded into a *terrain database*, which is then populated with environmental objects (e.g. houses or trees). These objects are 3D models stored in the *model database* and positioned via references defining position, orientation and scale. Both databases are the foundation on which scenarios are defined in computer graphic engines as depicted in Figure 2-3. 3D models are simplified representation of their natural pendants leading to differences in geometry, material and texture. In Figure 2-3, lightning bolts highlight possible errors that might be introduced during the image rendering process. For instance when recreating a natural scene possible sources of error are differences in scale and placement of objects. After defining the scene, the coordinates are transformed into the camera coordinate system. In this camera space, the camera location defines the origin. Thus, inaccuracies in replicating a referenced scene are possible errors. The scene is then lighted using a *global* or *local illumination method*. Global illumination methods are often based on physically correct illumination models (e.g. ray tracing), which can produce photorealistic images at the cost of increased computation efforts. A key interest of this thesis is the degree of graphical detail necessary to achieve functional realism for machine vision algorithm for reactive scenes, thus requiring real-time rendering. Therefore, the more computational efficient *local illumination methods* are of interest such as the *Phong reflectance model* (Phong, 1975). This model comprises three components:

- *Diffusive lighting* computes the reflection of light sources using the orientation of polygon normal vectors.
- *Ambient lighting* defines a constant amount of light that is applied to all polygons independent to their orientation.
- *Specular highlights* simulate specular highlights on surface that are computed using the orientation of polygons and material properties.

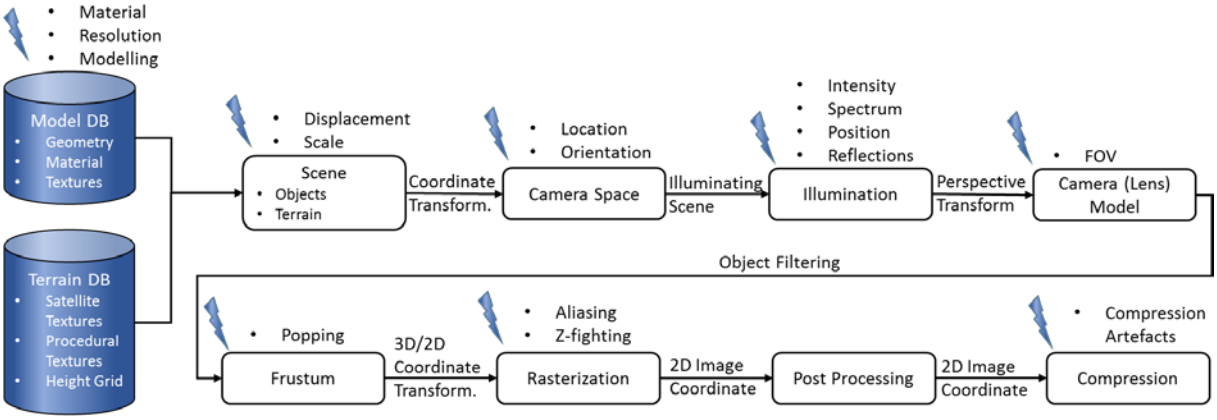


Figure 2-3: The generic graphics-rendering pipeline using *local illumination*. Lightning bolts indicate possible sources of error.

Here error sources are the simulated spectrum and intensity of used light sources as well as their location. Additionally, reflections of the environment will be physically incorrect when using *local illumination*. Afterwards the whole scene is transformed using a camera lens model that determines the optics *field of view* (FOV). The (camera) frustum limits the space that will be rendered using the maximum and minimum rendering distance or the viewing angle defined by the focal length. Objects outside of this frustum will not be rendered to reduce the computation effort (clipping). This technique may lead to suddenly appearing objects. A similar effect called *popping* arises when objects are modelled in several *Levels Of Detail* (LOD). LOD are used to reduce the computation effort by reducing the fidelity of objects proportional to the camera distance. The transmission between LOD states can appear as sudden content changes depending on the implementation of the rendering engine.

Rasterization summarizes several rendering steps (e.g. clipping, culling, viewport transformation, rasterization and hidden surface removal (Z-buffer)) to simplify the presentation of the rendering pipeline. More details on this subject can be found in (Bender & Brill, 2003) or (Watt, 2000). This process transforms the vector-based representation to a grid of quadratic picture elements called *pixels*. Manifold rasterization methods exist (Bender &

Brill, 2003). In this process, a pixel grid overlies the camera view. For example the *midpoint algorithm* (Pitteway, 1967; Van Aken & Novak, 1985) computes the mid of each grid element (pixel) border and if a line crosses the border before the middle of the line it is considered part of the line and thus coloured. The resulting rasterized edges have a width of one pixel and appear jagged. This *aliasing* effect produces unnaturally high spatial frequencies and optical artefacts in the image, especially when animated. Thus, this effect is a source of error and influences the quality of images. Another rendering based error is known as *Z-fighting*. Due to limited depth-buffer resolution and rounding effects, the order of polygons in same or similar distance to the camera cannot be robustly determined. These polygons then are 'fighting' for visibility resulting in flickering and alternating textures creating a noisy look (Vasilakis & Fudos, 2013).

After the pixel image has been generated, post-processing effects are deployed to improve the visual quality. Possible examples are anti-aliasing methods or cinematic effects such as film grain or colour filters. Identical to the digital photography pipeline the digital image can now be presented or stored. Compression is not mandatory but usual when storing images or video introducing compression artefacts depending on the compression method.

In summary, while the general components of image generation between digital photography and computer-generated imagery are similar, workflows and error sources differ significantly. The main errors for each pipeline can be extracted from Figure 2-4. These errors show the main differences between these two image types that need to be investigated in order to identify, which errors have an impact on the performance of computer vision algorithms.

	System induced Errors	Replication induced Errors
Digital Photography	• Blur • Lens Distortion • Chromatic Aberration • Noise • Motion Blur • Compression Artefacts	
Computer Generated Imagery	• Modelling Detail • Popping • Aliasing • Z-fighting • Compression Artefacts	• Object Material • Texture Resolution • Inaccuracies in Position, Orientation and Scale of Objects and Camera • Inaccuracies in Light Modelling (Spectrum, Intensity & Reflections)

Figure 2-4: System-related and reproduction errors in digital photography and computer generated imagery pipelines.

2.4 Realism and fidelity in computer graphics

Though synthetic datasets are often described as realistic and of high fidelity, commenters usually do not further explain what these buzzwords mean in the context of rendered computer graphics imagery as for instance can be seen in (Bourdis et al., 2011) or in chapter 2.2.3. The term realism is often used as an equivalent of the term ‘lookalike’. When observing datasets called *realistic*, they mainly depict real-world scenes, with natural and man-made objects using fitted textures of suitable resolution. Light effects are added to increase the visual representation. Thus, in this chapter current existing definitions of *realism* are presented, followed by the selection of the most suitable category of *realism* for investigations like this.

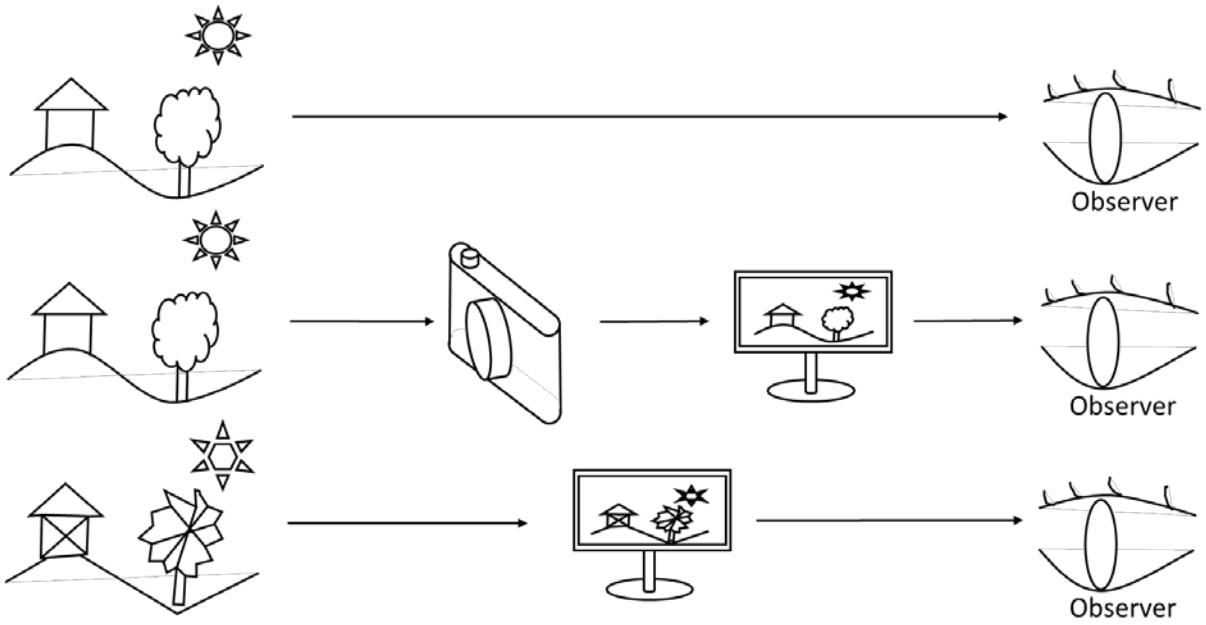


Figure 2-5: Three different courses of reception: Real world, real world via camera, virtual world.

(Blinn, Greenberg, Hagen, Feiner, & Mackinlay, 1988) define *photo-realism* or *photographic realism* as the quality level where computer rendered imagery is indistinguishable to photographs. Since realism in fact compares the visual quality of images to the real world, the term describes more a subjective impression closely related to the human vision system. Thus, the observer (e.g. human or machine vision system) of the image and his characteristics and limitations need to be considered. For example, Figure 2-5 depicts the reception of a scene by a human via three different courses. The first course depicts the direct physical perception while being personally at the scene. In the second course, the scene is perceived via a photograph displayed on a monitor. The last course presents a computer-generated representation of the scene displayed on the monitor and again received by the human vision

system. Each step involved in the recording and display process can introduce artefact and thus alter the visual experience. For this thesis, the interest focuses on the first steps (scene and image acquisition) while the observer is being replaced by a machine vision process.

The fuzziness in definition of the word realism is discussed in (James A. Ferwerda, 2003). Since realism of an image is dependent to its application, the author proposes three standards:

- **Physical Realism:** *"Here the criterion for realism is that the image has to provide the same **visual stimulation** as the scene. […], this means that the image has to be an accurate point-by-point representation of the spectral irradiance values at a particular viewpoint in the scene. This places strict demands on the image generation process. First, the model must contain accurate descriptions of the shapes, materials, and illumination properties of the scene. Next, the renderer must be able to accurately simulate the spectral and intensive properties of the light energy arriving at the observer's viewpoint. Finally, the display device must be able to accurately reproduce these energies. Although physically-based image synthesis methods can achieve the first two goals, conventional displays cannot, in general, reproduce the rendered light energies, so creating physically realistic images, is currently impossible except under restricted conditions."* (James A. Ferwerda, 2003).
  Physical realistic images are generated using physical models simulating the natural image generation process. This degree of realism is for instance suitable for applications interested in the physics of light scattering and reflections. Drawbacks of this method are the computational demand and the deficiencies of current display technologies to present the results physically correctly.

- **Photo-Realism:** This standard is directed to images that are indistinguishable from photographs of the same scene. The author defines this level of realism as *"the need of the image to produce the same **visual response** as the scene even though the physical energy coming off the image may be different in the scene"*(James A. Ferwerda, 2003). More simply phrased the rendered and photographed image shall be a *look-a-like* from the perspective of an observer. This kind of realism is used every day. For instance by exploiting the nature of the human eye, which integrates the light of small red, green and blue transistors to any colour in the visible spectrum. The author concludes that *"it is unclear that photo-realism is necessary or even desirable in a wide range of graphics applications, and second, adopting photo-realism as a standard for visual*

realism in computer graphics, classifies most renderings as failures, yet says nothing about their obvious utility in many application domains."(James A. Ferwerda, 2003).

- **Functional Realism:** The criterion for this category of realism is to provide the same ***visual information*** as the depicted scene (James A. Ferwerda, 2003): *"Information here means knowledge about meaningful properties of objects in a scene, such as their shapes, sizes, positions, motions and materials that allows an observer [receiver] to make reliable visual judgments and to perform useful visual tasks"*. He suggests, *"if an image lets you do the task you need to do, and allows you to perform the task as well as you could in the real world, then for that task, the image is realistic"* (James A. Ferwerda, 2003). He further provides an example about computer graphics in flight simulators. *"Typically, they [images rendered for flight simulation] are not physically accurate simulations, nor are the photo-realistic renderings, but they are functionally realistic in that they provide the observer [receiver] with much of the same visual information that they would receive if they were flying a real plane. The proof of the realism of these images is that they allow the observer to learn skills that then transfer into the real world*" (James A. Ferwerda, 2003). Functional realistic images provide the necessary information to successfully perform a task, but remove unnecessary details, using simpler rendering methods enabling real-time computation and interaction.

(James A. Ferwerda, 2003) then introduces the metrics *accuracy* and *fidelity* as measurements of functional realism:

- ***"… Accuracy** is the correctness of the image in respect to some physically measurable property of the scene such as radiance."*(James A. Ferwerda, 2003). More formally (Gross, 1999) describes it as *"the degree to which a parameter or variable or set of parameters or variables within a model or simulation conform exactly to reality or to some chosen standard or referent."*
- **Fidelity** can be measured by the degree an observer is able to perform a visual task in (James A. Ferwerda, 2003) and (Gross, 1999) provides the following definition: *"The degree to which a model or simulation reproduces the state and behaviour of a real world object or the perception of a real world object, feature, condition, or chosen standard in a measurable or perceivable manner; a measure of the realism of a model or simulation; faithfulness."*

This means *accuracy* can be directly measured, but *fidelity* needs to be acquired by the difference of success rates in task execution with given visual cues. (James A. Ferwerda, 2003) then presents a model based on *probabilistic inference* to quantify fidelity. Additionally, the *functional difference predictor* (FDP) is presented, which shall determine whether a rendered picture provides a functional difference to a picture rendered with physical-realism.

Conclusion

In this thesis, a machine vision algorithm is replacing the human observer. However, most definitions of realism consider the receiver to be human and thus consider and utilize the limitations of the human vision system. Therefore, a more generic definition of Ferwerda's *photo-realism* is proposed:

> **Perceptual Realism** is a more generalized form of the standard photo-realism defining a look-a-like to the natural scene as perceived by the observer. The image shall yield an equivalent ***visual stimulation***, while reducing the quality of visual cues not perceivable by the receiver. The perceptual realism is tuned towards the capabilities and limitations of the receiver.

Current *local illumination* rendering engines may be able to provide this level of realism. Actually, *functional realism* would suite the desired rendering quality in this thesis very neatly, but the actual implementation of a functional realistic rendering engine is dependant of a-priori knowledge. This means the visual cues on which the machine vision system is sensitive to need to be known beforehand, in order to develop or configure a rendering system for the desired application. To identify the visual cues of interest a test bed rendering system capable of a higher level of realism is necessary. Thus, by configuring the rendering system the necessary *visual information* for *functional realism* can be extracted. For machine vision systems to be evaluated using synthetic imagery this means influencing visual cues need to be available in sufficient (functional) realism, while visual cues that do not affect the performance of visual algorithms, can be neglected.

2.5 Image comparison

To identify the differences between photographs and computer graphics imagery objective quantifiable measures are necessary. These measures need to quantify the differences based on

different image properties to allow identification of essential properties via empirical experimentation. Image properties of digital imagery can be categorized in various forms (see for example (Hoogs & Hackett, 1995)). In this thesis, measures of various image properties are grouped into their application domains, since they share similar concepts and / or goals:

- **Image quality measures** originate from the desire to measure the quality difference of two with respect to content equal images. These are for example used to measure the efficiency of compression algorithms.
- **Application driven measures** use the actual use case as evaluation criteria and the output difference due to usage of different datasets.
- **Scene driven measures** extract the scene represented in an image and save it into a meta-format. This format is then compared to identify whether two images depict a similar scene.

In the following subchapters, all measures are presented with their original purpose. Additionally each measure is evaluated against following questions:

- Can the measure **distinguish** between **natural** and **synthetic** imagery of the same scene?
- Can the measure **identify the impact** exchanging the type of imagery has on the deployed **computer-vision algorithm**?
- Can the measure **identify the underlying reason** (structural difference) leading to its optical or performance difference?
- Has it limitations (e.g. human visual system, very-specific design)?

The last category presented in this chapter contains general **similarity and distance measures** necessary to compare arbitrary feature vectors. These are deployed to a certain extend in all previous mentioned categories and are thus granted a separate category for structural purposes.

2.5.1 Image quality measures

These measures are intended to compare images directly based on their pixel values (not considering the depicted scene). This chapter presents common *image quality metrics* (IQM). Usually these are applied to identify quality deficiencies in regard to noise, blur or compression artefacts to evaluate compression techniques. Prominent and simple examples

are *mean square error* (MSE) or *peak-signal-to-noise ratio* (*PSNR*). The discussed IQM measures are evaluated against their usability to evaluate the given scientific question (see Table 2-1). The overview shows that no literature could be found using these measures as an evaluation tool to distinguish between natural and synthetic data. Additionally, most methods are tuned towards the human vision system due to their main application as quality measure for video compression algorithms. Thus, none of the investigated IQM algorithms seem to fit the given task perfectly. However, they should be further investigated, since their suitability is still unknown.

Table 2-1: All IQM discussed measures and their suitability to evaluate the formulated scientific question.

Measures	Distinguish natural vs. synthetic?	Identify impact on CV-perf.?	Identify underlying reason?	Has Limitations?
MSE described in (Horé & Ziou, 2010)	Unknown	Unknown	No	No
PSNR described in (Horé & Ziou, 2010)	Unknown	Unknown	No	No
SSIM (Z. Wang, Bovik, Sheikh, & Simoncelli, 2004)	Unknown	Unknown	No	Yes
MS-SSIM (Z. Wang, Simoncelli, & Bovik, 2003)	Unknown	Unknown	No	Yes
IW-SSIM (Z. Wang & Li, 2011)	Unknown	Unknown	No	Yes
SR-SIM (Lin Zhang & Li, 2012)	Unknown	Unknown	No	Yes
FSIM (Lin Zhang, Zhang, Mou, & Zhang, 2011)	Unknown	Unknown	No	Yes
MAD (Larson & Chandler, 2010)	Unknown	Unknown	No	Yes
Visual Difference Predictor (Daly, 1992)	Unknown	No	No	Yes
JND (Lubin & Fibush, 1997)	Unknown	No	No	Yes
NSS (Sheikh, Bovik, & Cormack, 2005)	Unknown	No	No	Yes
Contrast (Ke, Tang, & Jing, 2006)	Unknown	No	Yes	Yes
Blur (Tong, Li, Zhang, & Zhang, 2004)	Unknown	No	Yes	Yes
Hue Count (Ke et al., 2006)	Unknown	No	Yes	Yes
Edge Distribution (Ke et al., 2006)	Unknown	No	Yes	Yes
Focus (Ke et al., 2006)	Unknown	No	Yes	Yes

MSE sums the mean square distances between reference image *r* and test image *t* with a resolution of *M* by *N* and divides the result through the multiplication of the resolution (Horé & Ziou, 2010):

$$MSE(r,t) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(r_{ij} - t_{ij}\right)^2 \quad (1)$$

PSNR divides through *MSE* and adds a logarithmic scale leading to higher values indicating higher image quality (Horé & Ziou, 2010):

$$PSNR(r,t) = 10\log_{10}\left(\frac{255^2}{MSE(r,t)}\right) \quad (2)$$

These metrics are simple to compute but still show good performance for measuring the difference to a reference. For instance according to the statistical analysis of (Avcibaş, Sankur, & Sayood, 2002), out of 26 evaluated image quality metrics *MSE* still provides the best measurement when assessing noise in images. However, when measuring blur or compression artefacts other metrics are more suitable. Further, *full-reference measures* (see Figure 2-6) such as *MSE* need the distortion-free reference image for comparison (Delepoulle, Bigand, & Renaud, 2012). Such approaches cannot be used characterize specific image properties, because of their holistic view of the image.

(Oelbaum, 2008) discusses all three major visual quality measure concepts: *full reference*, *reduced reference* and *no reference*. The *reduced reference* measures compare distorted images against parameters extracted from the reference to save bandwidth. Possible parameters are wavelet transformation coefficients. *No reference* measures directly process the distorted image and rate to the contained distortion. The difficulty for *no reference* measures is to distinct whether specific image content is supposed to look this way or if distortion reduced its quality. A standard approach in using these measures is the combination of several parameter measurements into one rating. Generally, they try to measure which artefacts are usually not present in natural images. (Oelbaum, 2008)

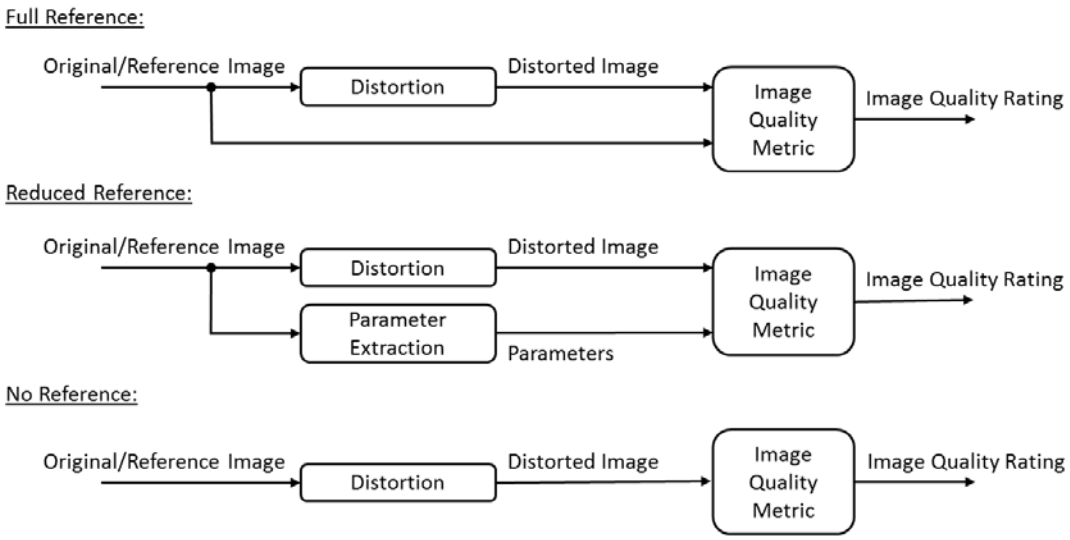


Figure 2-6: The three image quality measure concepts. Summarized from (Oelbaum, 2008).

In (Z. Wang et al., 2004) the *structural similarity* (*SSIM*) measure is presented. (Lin Zhang, Zhang, Mou, & Zhang, 2012) consider this algorithm a milestone in the development of novel

full-reference image quality measure. Structural information in an image is defined in (Z. Wang et al., 2004) as *"those attributes that represent the structure of objects in the scene, independent of the average (local) luminance and contrast."* The concept consists of three basic measurements of luminance, contrast and structure. The image is normalized using the acquired measures of luminance and contrast and the structural similarity is computed. Afterwards the three measurements are combined by a weighed multiplication. This measure shows higher performance compared to *MSE* or *PSNR* for the surveyed distortions. Further improved versions of this approach are *multi scale-SSIM* (*MS-SSIM)* (Z. Wang et al., 2003) and *information content weighted-SSIM (IW-SSIM)* (Z. Wang & Li, 2011). According to (Lin Zhang et al., 2012) *IW-SSIM* provides *"pleasing"* results while *SSIM* is still the fastest of all *"modern"* IQM's. (Kundu & Evans, 2015a) recently tested all common modern metrics on synthetic images concerning interpolation, blur, additive noise, JPEG compression artefacts and *fast fading*. The authors identified that other modern IQM's such as the *spectral residual* based similarity measure (*SR-SIM*) (Lin Zhang & Li, 2012), *feature similarity index* (*FSIM*) (Lin Zhang et al., 2011) or the *most apparent distortion (MAD)* (Larson & Chandler, 2010) outperform the structure based metrics on synthetic data. However, all of these metrics understand image quality only related to the presence of sensor or compression artefacts. (Kudelka, 2012) uses these IQMs to determine the quality of textures. It is commonly identified that the need for greyscale images as input is a major drawback of most IQMs.

Image quality measures that consider the perceptual limitations of the observer have also been designed. Currently, however these are solely tuned towards the human visual system. A very prominent *full-reference* example is the *visual difference predictor (VDP)* (Daly, 1992). The threshold model marks areas where difference in quality exists and is visible to a human observer. The amount of difference cannot be extracted. The measure calculates a probability of error detection for every image pixel. The model does not consider colour. A similar measure providing the same output is *just noticeable differences (JND)* (Lubin & Fibush, 1997), which additionally considers colour. (Boulenguez, Airieau, Larabi, & Meneveaux, 2012) conduct a psychophysical experiment to identify significant criteria on perceived quality of computer graphic imagery. The investigation identified that *"contrast, noise and shadows have a major effect on the overall [perceived] quality"* in comparison to colour bleeding and aliasing. The conclusion of the authors applies to human observers only and computer vision algorithms may weigh the importance of features differently, but the general approach can also be applied to machine vision after the perceptual limitations of the investigated algorithm have been modelled or measured.

A *no-reference metric* to measure the quality of images based on *natural scene statistics (NSS)* is presented in (Sheikh, Bovik, & Cormack, 2005). These statistics are acquired by a wavelet (spatial frequency) based image evaluation. The resulting image decomposition is used to construct histograms depicting the relationship of image frequency magnitudes at different decomposition scales. These histograms are used to feed a normalized (mean and variance) probability model. (Sheikh, Bovik, & Cormack, 2005) use *NSS* to identify that increased *JPEG2000* compression (which is also based on wavelets) degrade the natural wavelet structure of images. This finding is used to implement a measure of image quality assessment for compression quality. The performance of *NSS* based metrics on synthetic data is evaluated in (Kundu & Evans, 2015b) showing that distortion in synthetic images also change the scene statistics and are therefore detectable. Numerical results attribute *NSS* metrics better results than aforementioned full reference metrics *PSNR* or *SSIM*. However, most models of non-reference *IQM*'s need distortion less images to train their models, making the comparison of natural with synthetic images difficult. Additionally, only the overall quality difference is quantified without identifying the cause. However, specialized metrics exist for blur (Marziliano, Dufaux, Winkler, & Ebrahimi, 2002), noise and blocking artefacts (Z. Wang, Bovik, & Evan, 2000) as pointed out by (Oelbaum, 2008).

In contrast to above image quality measures (Ke et al., 2006) present photo-quality features and metrics for specific image properties. The goal of the authors is to learn a classifier that robustly differentiates between professional photos and snapshots. In the process of this work, the authors formulate several interesting measures of image properties:

- **Spatial distribution of edges:** The distribution of edges indicate the location of the subject and whether the background is cluttered. In general, this feature describes the spatial composition of the image. The metrics compare the frequency domain image against the mean spatial frequency distributions of high- and low-quality images derived from training datasets.

  A second metric uses edges as features to compute the area of a bounding box enclosing 96% of highest energy edges (sharpest). Since professional photos focus on the subject, a small bounding box is expected.

- **Colour distribution:** A histogram for each channel is calculated to create a three dimensional space histogram and is compared to the closest related histograms computed from the training dataset.

- **Blur:** In high quality photos, at least one segment of the image should be sharp. The authors present a metrics based on *Fast Fourier Transformation* (FFT) of the image.
- **Contrast:** Contrast is determined by acquiring the width of the additive grey level histogram of all three colours encapsulating 98% of all pixel values.

(Ke et al., 2006) identified the blur detector to be most discriminative feature followed by edge spatial distribution and contrast to classify between low- and high-quality photographs. However, the derived features may generally help in describing the quality and structure of an image. Unfortunately, the presented approach depends on training data, which diminishes its value since the effort to produce general valid training data is high (large natural database, different sources, manual annotation).

2.5.2 Application driven measures

Measures presented in this chapter evaluate images in the context of their use case. Thus, images are assessed whether they provide all necessary visual information necessary to perform a specific task (chapter 2.4). The previously presented image quality measures compare images by measuring differences between pixels or by computing the detection probability of differences for a specific observer. (J. A. Ferwerda & Pellacini, 2003) states that *"images are functionally equivalent with respect to a task"* when the task can be performed while the image has visible perceptual differences. This statement is tested via a psychophysical experiment using humans as observers and judge. The subjects had to perform tasks on the presented rendered images, which also contained visible errors. The author concludes that *"subjects are either totally affected or totally unaffected by the errors, depending on the type of error [distortion] introduced."* Additionally, it has been identified that even though the test subjects were aware of errors in the image, these did not affect their task execution performances. (J. A. Ferwerda & Pellacini, 2003) demonstrates a novel concept to base the quality of images on their influence on tasks performed by the receiver. Rudimentary psychophysical experiments are conducted validating aforementioned hypothesis. While the hypothesis itself may sound obvious, it carries the message that image quality needs to be set into context. Therefore, depending on the task, different kinds of errors affect the task performance in an image. For instance, the task of counting vehicles in an image may be robust to noise while the task to identify the colours of such could be affected.

Aforementioned investigation regards the human as image observer and thus measures the perceptual quality of images in respect to humans. Similarly, application driven quality measures are also beneficial with machine vision algorithms as observer. *Advanced Driver Assistance Systems* (ADAS) for instance use computer vision algorithms depending on sensor information to assist the driver in manifold ways (e.g. automatic cruise control, lane detection assistant, etc.). (Nentwig & Stamminger, 2011) evaluate applications for vehicle detection and lane detection on natural and synthetic (reconstructed scene) data to identify whether the provided visual information of both data types is equal for these two use cases. For vehicle detection, an algorithm to predict the distance of a vehicle in front is used as evaluation criteria and the result deviation between the two datatypes on the same scene are presented in percentage. The average deviation between results from natural and synthetic data is 5 to 10% depending on the evaluated scene. The lane detection algorithm is evaluated using lane view distance computations. Here depending on the evaluated scene the average deviations range from 3.2% to 18.8%. These deviations are attributed to inaccuracies in the simulation environment (e.g. weather model, scenario configuration). In this empirical evaluation, synthetic imagery still does not contain the same visual information (necessary for the vision task) compared to natural imagery. However, small average deviations show that in future this goal could be met after the necessary visual cues have been identified, which however is not possible using this approach. Comparing the method to the image quality measures (chapter 2.5.1) it qualifies as *full-reference measure* (Figure 2-7). In terms of realism levels discussed in section 2.3, this approach identifies the functional realism of computer-generated images. The similarity measure quantifies the functional realism between one (full) and zero (no).

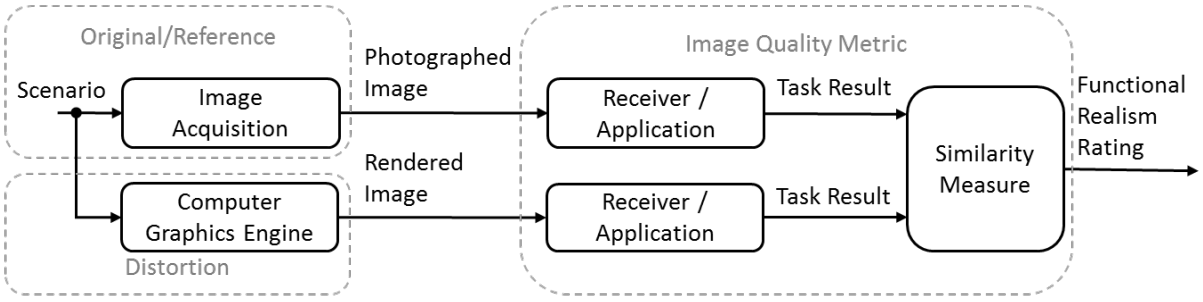


Figure 2-7: The empirical application driven image quality evaluation of (Nentwig & Stamminger, 2011) put into the context of image quality assessment.

In (Nentwig et al., 2012) the influence of lighting and camera model on the applications results is discussed. Three shadow generation techniques were evaluated against reference photographs by comparing the luminance and magnitude of gradients. The comparison was

conducted in a manual and qualitative way. The deployed rendering engine applies an atmospheric model (e.g. (Hoffman & Preetham, 2002)) to calculate the luminance in the image and different reflection values for road, vehicle or other entities using the BRDF-function (Bidirectional reflectance and distribution function[9]). These parameters are manually calibrated to minimize the difference to the reference image in respect to structural, geometric and photometric difference. Additionally, physical camera effects such as depth of field, noise or motion blur are added to the simulation. The vehicle classifier of (Nentwig & Stamminger, 2011) is used to evaluate the optimized synthetic imagery by computing the rates of true and false hypotheses and categorize them in regard to the depicted object. The results show, synthetic images produce a similar true to false ratio of hypotheses, however the absolute number differs by a factor of three. The statistical analysis identifies significant differences between the images depicting vegetation or vehicle. These deviances are concluded to be caused by differences in modelling or shading. In conclusion, the mentioned papers present image quality or image similarity metrics based on the performance differences and statistics of computer vision algorithms. Such approach can be categorized as application driven image quality measure, since the deviance to the reference (photograph) is caused by a different image structure reducing the capabilities of the computer vision algorithm. In such case, the synthetic image generation is not fully functional realistic, but the degree of difference can be calculated.

2.5.3 Content driven measures

This last type of measures consider the content or scene in the image. Table 2-2 provides an overview of all measures discusses in this chapter and their suitability to answer the scientific question according to the question formulated in chapter 2.5.The overview shows that all measures presented in this chapter have the possibility to identify a fundamental cause of image differences, because they are designed to evaluate very specific image properties. Most of the measures defined by (Tang, Luo, & Wang, 2013) are focussed towards grading the artistic degree of imagery which reduces their applicability as a more general quality measure. The measures defined by the MPEG7 standard show high potential, however their capability to distinguish between natural and synthetic imagery hasn’t been evaluated yet. Thus, none of

[9] The BRDF (Bidirectional reflectance and distribution) function defined by Fred Nicodemus (Nicodemus, 1965). This function describes the reflectance of light on solid not transparent surfaces depending on the incoming light direction, the normal vector of the surface, incoming radiance and outgoing radiance. Thus the reflection of a light by an object depending on the location of the viewer and the pose and position of the object.

the measure fit the given task perfectly and further investigation is necessary. The chapter now presents each measure in detailed.

Table 2-2: All discussed content driven measures and their suitability to evaluate the scientific question.

Measures	Distinguish natural vs. synthetic?	Identify impact on CV-perf.?	Identify underlying reason?	Has Limitations?
Hue Comp (Tang et al., 2013)	Unknown	Unknown	Yes	Yes
Scene Comp (Tang et al., 2013)	Unknown	Unknown	Yes	Yes
Dark Channel (Tang et al., 2013)	Unknown	Unknown	Yes	No
Complexity Feature (Tang et al., 2013)	Unknown	Unknown	Yes	Yes
MPEG7 DCD (Ohm et al., 2002)	Unknown	Unknown	Yes	No
MPEG7 SCD (Ohm et al., 2002)	Unknown	Unknown	Yes	No
MPEG7 CSD (Ohm et al., 2002)	Unknown	Unknown	Yes	No
MPEG7 CLD (Ohm et al., 2002)	Unknown	Unknown	Yes	No
MPEG7 EHD (Choi, Won, Ro, & Manjunath, 2002)	Unknown	Unknown	Yes	No
MPEG7 HTD (Choi et al., 2002)	Unknown	Unknown	Yes	No

In (Luo, Wang, & Tang, 2011) and their follow-up paper (Tang et al., 2013) the authors assess the quality of photographs based on regional and global features estimating the images composition. The authors assume that for different photographic content (for example "landscape" or "portrait/human") the number and type of relevant features differs vastly. They state *"for landscape photos, well balanced spatial structure, professional hue composition, and proper lighting are considered as traits of professional photography"* (Luo et al., 2011). The following features are listed:

- **Global Features** use the image as a whole for computation.
  - **Hue Composition** measures the colour composition scheme of an image
  - **Scene Composition** detects long continues edges often providing semantic meaning and characterizes these.
- **Regional Features** measure specific properties of the image subject, which is extracted using region detection algorithms.
  - **Dark Channel** measures blur, colour saturation and colour composition.
  - **Complexity** compares the complexity of the subject against the background.

Specifically the results of category "landscape" are of interest for this thesis as it comes closest to aerial images. The *dark channel* feature (measuring clarity and colourfulness) discerns between high and low quality images, followed by the hue composition feature. The

features show good capabilities in describing the difference of image content instead of global measurements based on image distortions (e.g. *IQMs*).

Other methods analysing the content of images for comparison can be found in the *Content-Based Image Retrieval* (CBIR) domain, which uses the image content to find content-related images. Here, the image content of a "search" image is compared to prior derived and stored content property descriptions of available images called *image description database*. The distance between the feature descriptions of the "search" image and database are computed using distance and similarity measures presented in section 2.5.4 and the most similar images are determined. Many features to describe an image have been formulated. The most prominent features are *colour* (Swain & Ballard, 1991), (Stricker, Stricker, Orengo, & Orengo, 1995), *texture* (Manjunath, Ohm, Vasudevan, & Yamada, 2001), (Haralick, Shanmugan, & Dinstein, 1973) and *shape* features (Loncaric, 1998), (Safar, Shahabi, & Sun, 2000).

The motion picture expert group (MPEG) standardized in ISO/IEC 15938 "Multimedia content description interface" a content description scheme for video, audio and text, abbreviated known as *MPEG7* (Sikora, 2001). The standard mainly defines interfaces, semantics and syntax of descriptors and the descriptions themselves in suitable ways. To keep the standard flexible the actual concept of*"…, how similarity between images or video is defined is left to the specific applications requirements"* (Sikora, 2001). Visual MPEG7 image descriptors are categorized in colour, texture, shape, human face or motion descriptors. A C-implementation of these descriptors can be found in the experimental model (XM) (Motion Picture Expert Group, 2003). The concepts of the descriptors are detailed in (Cieplinski, Kim, Ohm, Pickering, & Yamada, 2000). The following introduction was first published in (Hummel & Stütz, 2014)[10].

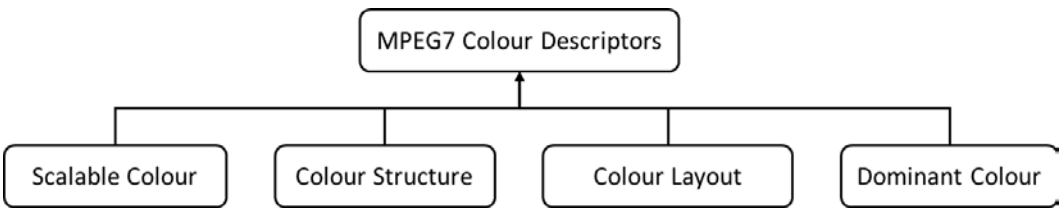
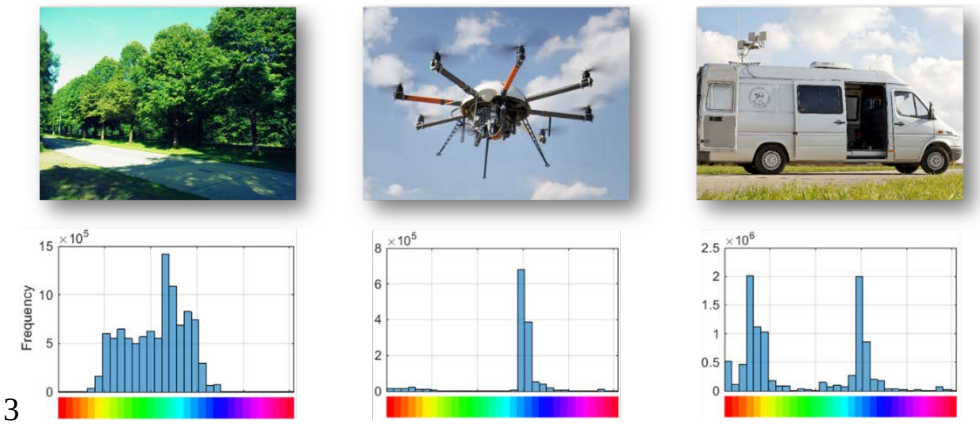


Figure 2-8: The MPEG7 Colour descriptors. Image based on (Manjunath et al., 2001).

[10] The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-13823-7

The first presented group of *MPEG7* content descriptors is *colour* oriented (Figure 2-8). The *Scalable Colour Descriptor* (SCD) is a global 256-bin colour histogram in the HSV colour space. The bin values are non-uniformly quantized to reduce the size of the descriptor. Haar-Transformation is used to reduce the amount of data even further (Manjunath et al., 2001). The distance between two colour histograms is calculated by matching the Haar-coefficients with the Manhattan distance (L_1). Thus, SCD evaluates the global difference in colour composition between two images. The example in Figure 2-9 the colour distribution of three images based on this feature the left and the middle image show higher similarity.



3

Figure 2-9: Three example images and their colour distributions.

The *Colour Structure Descriptor* (CSD) expresses the global colour features as well as the local colour structure by using an 8x8-pixel structure element (kernel) to count pixels of every present colour as depicted in Figure 2-10. The kernel is sweeping over the image describing the local colour structure at 64 uniformly distributed locations. For images with a resolution greater than 640x480 pixels, subsampling is used to cover the image uniformly (Buturovic, 2005). This way, CSD can even discern images that globally have the same amount of (e.g. as depicted in colour histograms) but different spatial distribution of colour among the image. Thus, CSD serves as a spatial distribution measure of colour content in the image.

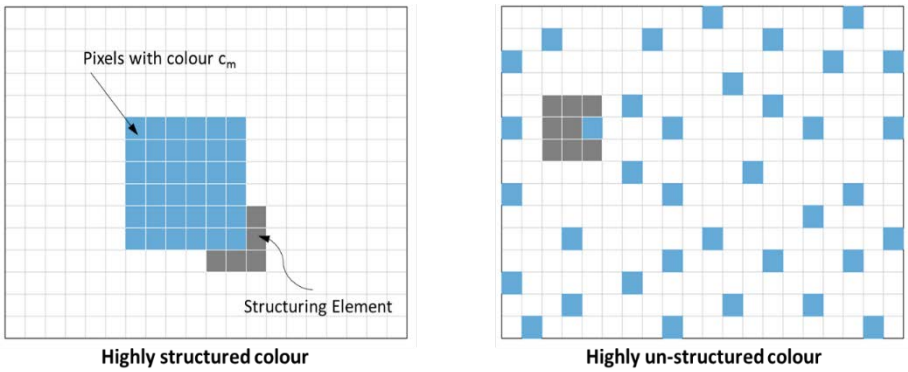


Figure 2-10: Example for structured and unstructured colour distribution (Cieplinski et al., 2000)

The resolution invariant *Colour Layout Descriptor* (CLD) describes the spatial distribution of colour in the YCbCr colour space. As depicted in Figure 2-11 the descriptor separates an image into 64 blocks (8x8) and computes the average colour for each block as its representative colour. Encoding of the resulting image is performed by zigzag scanning the image and applying a discrete cosine transformation. (Spyrou, Tolias, Mylonas, & Avrithis, 2009) concludes that CLD to be an especially effective descriptor for sketch-based image retrieval, content filtering and visualization. This descriptor allows the spatial comparison of dominant colours. Thus, this measure can be used to measure unneglectable misalignment of object or camera position or orientation between two images.

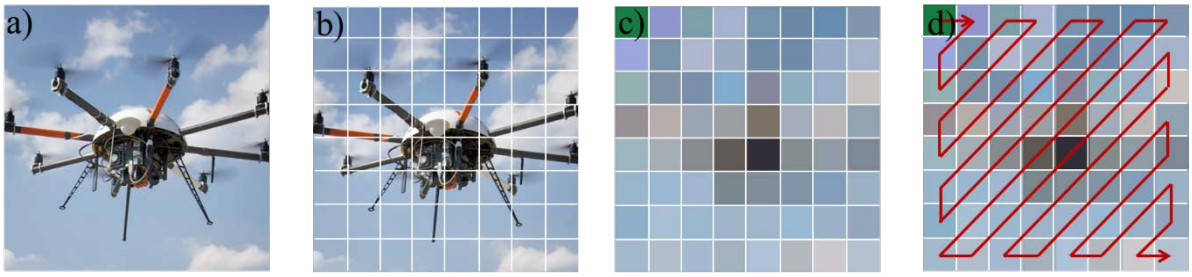


Figure 2-11: Steps of the Colour layout descriptor: a) Original image (J. Z. Wang, Li, & Wiederhold, 2001) b) subdivided image c) compute average d) zigzag scanning.

The *Dominant Colour Descriptor* (DCD) is a very compact descriptor describing up to eight colours dominating the images colour composition, the covered area per dominant colour in percentage and variance as well as the spatial coherency of dominant colours (Spyrou et al., 2009). The distance is calculated via the colour distance measure presented in (Ma, Deng, & Manjunath, 1997), which computes the Euclidian distance (L_2) between the colours identifying the closest related colour and multiplying the result with the difference in areal coverage for each colour. The DCD allows non-location based comparison of dominant colours and their amount of appearance.

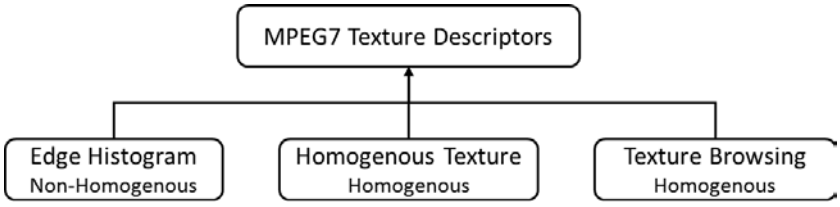


Figure 2-12: Texture Descriptors defined in the MPEG7 standard and their selective property

In CBIR the term texture is defined as a visual pattern with possibly homogenous properties that result from multiple colours and intensities in an image (Sikora, 2001). These peculiar patterns provide powerful means for similarity matching. In the MPEG-7 standard three

texture descriptors are defined (see Figure 2-12), the *Edge Histogram Descriptor* (EHD), the *Homogenous Texture Descriptor* (HTD) and *Texture Browsing Descriptor*. The scale invariant *Edge Histogram Descriptor* also known as non-homogenous texture descriptor captures the spatial distribution of edges similar to the CLD. The image is divided in 16 (4x4) equal blocks and edge orientation is calculated for five different categories as depicted in Figure 2-13. The resulting 80 bins (5x16) are useful to estimate the similarity of images containing non-homogeneous textures such as objects or non-repeating structures. This descriptor allows spatial comparison of contrast gradients (edges) distributed among the image. The distance between two EHD feature vectors is computed using the L_1-Distance measure.

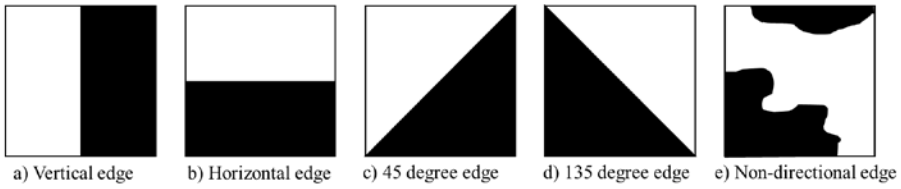


Figure 2-13: The five types of edges extracted from the edge histogram detector (Cieplinski et al., 2000).

The *Homogenous Texture Descriptor* (HTD) expresses the amount of structure inside an image by directionality, coarseness, regularity of patterns, etc. Since the descriptor focuses on image structure it is well suited for similarity matching in texture databases to identify corresponding repetitive patterns. The descriptor is calculated by converting the image into the frequency domain and filtering it by orientation and scale into 30 different channels as depicted in Figure 2-14. For each channel, the energy and energy deviation are calculated. The conversion in the frequency domain introduces the requirement of an image to have at least a size of 128 x 128 pixel for being able to compute the HTD. The distance between two HTD feature vectors is computed using the Mahalanobis distance. Further details about the descriptors math is detailed in (Ro & Yoo, 1999).

The *Texture Browsing Descriptor* is a very compact descriptor (12Bits length) that describes the regularity, directionality and bumpiness of a texture. Since textures can have more than one dominant direction and scale (bumpiness), the specification allows a maximum of two different values for each property (Manjunath et al., 2001). The regularity value defines the degree of uniformity of a periodic pattern with zero being random and three being clear in direction and bumpiness. The directionality of a value provides the dominant directions in six values ranging from zero to 150° in 30° steps. The bumpiness is provided with four values ranging from zero (fine grained) to three (rough-grained). The computation of the texture

browsing descriptor presented in (Manjunath, Wu, Newsam, & Shin, 2000) is closely related to HTD. The distance between two feature vectors is computed using the Manhattan distance (L_1) normed by the standard deviation of each feature component is used. (Manjunath et al., 2000)

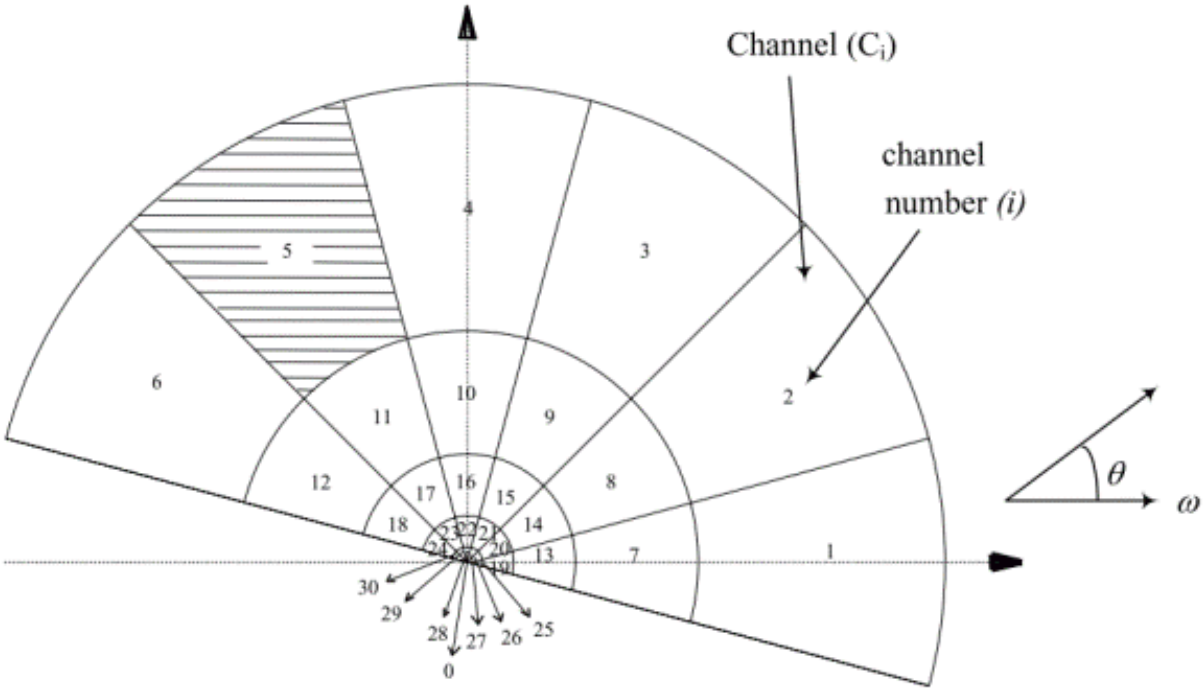


Figure 2-14: Segmentation of the frequency domain according to displayed layout for feature extraction (Cieplinski et al., 2000)

Another possibility to measure the similarity between images is shape extraction and comparison of presented objects. Out of all MPEG7 descriptors, only shape features allow object identification. These descriptors (see Figure 2-15) are presented in (Bober, 2001). In general, shape can be identified by the filled shape of the object (*region based)* and by their actual *contour*. Requirements to shape descriptors are compactness, fast computation and being invariant to scaling, rotation, translation and many shape distortions (e.g. perspective transform, segmentation errors).

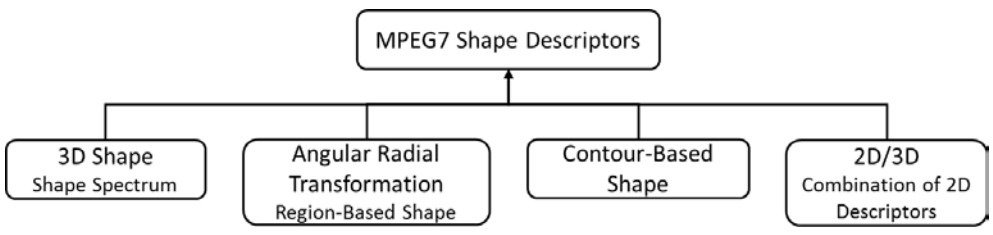


Figure 2-15: Shape descriptors as defined by the MPEG7 standard.

The *3D shape descriptor* or *shape spectrum descriptor* computes the minimum and maximum curvatures at each vertex point of a 3D-mesh and provides it using a histogram. The descriptor is designed for description 3D shapes and thus has no role in 3D image description. More information can be found in (Bober, 2001), (Grana & Cucchiara, 2006) or (Lisha Zhang, da Fonseca, & Ferreira, 2007).

The *Angular Radial Transformation (ART) region based shape descriptor* computes transformation invariant region-based moments. These are acquired through angular radial transformations on a unit disk in polar coordinates. This 2D descriptor is compact and robust to segmentation noise (Sikora, 2001).

The 2D *contour based shape descriptor* describes object shapes by their contour. It uses the *Curvature Scale-Space* (CSS) contour representation and includes eccentricity and circularity values of original and filtered contours (Sikora, 2001). In depth knowledge to CSS can be acquired in (Bober, 2001). The main advantages of the descriptor is its ability to distinguish between shapes of similar *region* but varying *contours*. This descriptor is robust to non-rigid deformations and perspective transforms (Bober, 2001).

The *2D/3D shape descriptor* can use the aforementioned 2D descriptors do define a 3D-object using multiple 2D snapshots. This 3D shape descriptor provides good results when the camera is rotating around the object but for aerial images this descriptor is of lower importance.

The MPEG7 standard also defines descriptors for motion and face identification and recognition. These are not addressed due to their limited benefit in the per image comparison of aerial images. For more information refer to (Sikora, 2001) or (Cieplinski et al., 2000).

In conclusion, MPEG7 provides four colour, two texture and two shape descriptors that may help to identify the similarity or differences of images on a content-based level. Texture Browsing and the two 3D shape descriptors have been dropped since their intended duty differs strongly from the desired use case.

In general, high-level description of properties provide a powerful tool to identify the similarity of images based on specific image attributes. However, it must be highlighted that high differences do not correlate to high quality differences. Whether these descriptors are helpful to identify image differences between natural and synthetic data needs to be investigated.

2.5.4 Similarity and distance measures

To identify the similarity or difference of images they need to be compared. Chapter 2.5.1 and 2.5.2 presented measures allowing direct comparison. The previous chapter presented methods to describe the appearance and content in various ways. Still all measures need the ability to compare some form of image description whether it be the direct image pixel or an abstract feature vector. This can be conducted using existing distance and similarity measures. The MPEG7 standard for instance provides a distance measure for each defined descriptor.

The most common deployed distance metric is the *Manhattan Distance* (L_1). It is derived from the more general *Minkowski Distance* L_p. The dissimilarity of feature vectors or images directly can be calculated using L_p, where n is the number of features, x_i the feature value of image X, y_i the feature value of image Y and p the order:

$$L_p = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{3}$$

Another well-known derivate of this metric is the Euler Distance ($p = 2$). The presented measures are called *distance measures* since they indicate the dissimilarity between two feature vectors. The greater the value, the larger the difference of the measured features. Two feature vectors are completely identical when the *distance* zero (Backhaus, Erichson, Plinke, & Weiber, 2006). *Similarity measures* on the other hand are greater, the larger the similarity between feature vectors is. In (Goshtasby, 2012) it is stated that *similarity/dissimilarity measures* do not need to be metric in order to provide effective measures. Comparable values are of metric, nominal, binary or probabilistic scale. Metric data are scaled by intervals and can be compared by differences of similar structured variables and vectors. A nominal scale is used once two variables or vectors contain absolute frequencies of occurrences. In Figure 2-16, a selection of measures is provided.

A very detailed list of binary measures can be found in (Seung-Seok et al., 2010), while nominal measurements are further described in (Sulc, 2014), (Michel, 2000). Interval based measures are the most commonly known measures and are detailed in (L. Wang, Zhang, & Feng, 2005), (Goshtasby, 2012) and (Brosius, 2013). Probabilistic measures are less common, since distributions can be transformed into nominal or interval scales. More about probabilistic measures or transformation of such can be found in (Cha, 2007), (Goshtasby, 2012), (Rahman, Bhattacharya, & Desai, 2005) and (Itoh & Shishido, 2008).

In (Goshtasby, 2012) distance metrics are evaluated regarding their performance of measuring the similarity of images using a template matching example. This rudimentary comparison presents characteristics of used metrics based on raw values, normalized values, ranks of values and joint probabilities of corresponding values (Goshtasby, 2012).

	Scale of Variables			
	Metric (interval)	Nominal (categorical)	Binary (0/1)	Probabilistic
Similarity Measure	• Cosine similarity • Pearson correlation	• Jaccard coefficient • Dice • Cosine similartiy • Overlap coefficient	• Jaccard coefficient • Dice, Czekanowski • Russel and Tao • Rogers and Tanimoto • Kulcysinki • Simple matching	• Fidelity
Distance Measure	• City Block, Manhattan L_1 • (Squared) Euclidian L_2 • Minkowski L_P • Chebyshev, Maximum L_∞	• Chi-square • Phi-square	• Binary (squared) euclidian • Size difference • Pattern difference • Variance • Lance and Williams	• Mahalanobis • Bhattacharyya • Hellinger • Bayes

Figure 2-16: Selection of *similarity* and *distance measures* for different variable types, based on (Backhaus et al., 2006)**, (Cha, 2007), (Seung-Seok, Sung-Hyuk, & Tappert, 2010), (Brosius, 2013) and (Michel, 2000).**

The *Mahalanobis distance* is a special metric that is able to compute the distance between a feature vector and a distribution. The distance is given in the number of standard deviations the vector deviates from the mean (the centre) of the distribution. A benefit of this distance is its ability to handle directional distributions (e.g. the Euclidean metric assumes an equal distribution for all given dimensions of the feature space). More about this measure can be found in (Mahalanobis, 1936) or (De Maesschalck, Jouan-Rimbaud, & Massart, 2000).

In CBIR systems, the correct choice of distance measures can improve their retrieval capabilities. Thus, efforts to improve the metrics in the MPEG7 standard are conducted (Eidenberger, 2003b). Most distance measures used in MPEG7 are based on geometric assumption since two similar images are expected to be in the same region of the feature space of the descriptor. The author compares the deployed metric-based distance measures to binary distance measure since these fit better with human perception according to (Tversky, 1977). It has been identified that distance measures selected in MPEG7 perform well but in some cases the measures *Meehl index* and *pattern difference measure* perform better. The reason for *pattern difference* better performance is that differences weight stronger than similarities (Eidenberger, 2003a).

For instance the *dominant colour descriptor* (DCD) was initially proposed using a quadratic dissimilarity measure to compute the distance between dominant colours of different images (Deng, Manjunath, Kenney, Moore, & Shin, 2001). In (Yang, Chang, Kuo, & Li, 2008) a similarity measure considering the difference of the dominant colours and the difference of percentages (amount the colour covers the image) is presented and experimental results show its improved performance on visually similar images.

2.6 Concepts investigating transferability of synthetically acquired results

As can been seen in chapter 2.2.3 synthetic data are already used for evaluation of CV-algorithms. However, these datasets are seldom investigated whether their acquired results can be transferred to the real world, which lead to criticism presented in chapter 1.1.3. This controversy presents the necessity to validate synthetic data against natural examples. This chapter presents approaches emerged in the last years to measure or investigate transferability. The metrics used for comparison are based on the questions provided in chapter 2.5. Table 2-3 provides a brief overview of all discussed concepts. Afterwards each approach is shortly discussed.

Table 2-3: All discussed concepts and their suitability to answer the given scientific question.

Approaches	Distinguish natural vs. synthetic?	Identify impact on CV-perf.?	Identify underlying reason?	Has Limitations?
Visual Task Performance				
Advanced Driver Assistance Systems (Nentwig et al., 2012)	Yes	Yes	No	Yes
Object Tracker (Hummel, Kovács, Stütz, & Szirányi, 2012)	Yes	Yes	No	Yes
Face localization (Sagonas, Tzimiropoulos, Zafeiriou, & Pantic, 2013)	Yes	Yes	No	Yes
Face-recognition (P.J. Phillips et al., 2000)	Yes	Yes	No	Yes
Eye lid localization (Wood et al., 2015)	Yes	Yes	No	Yes
CV-Component Performance				
Optical flow estimation (S. Meister & Kondermann, 2011)	Yes	Yes	No	Yes
Image segmentation (Irgenfried, Dittrich, & Wörn, 2014)	Yes	Yes	No	Yes
Image Statistics				
MPI-SINTEL (Daniel J Butler et al., 2012)	Yes	Yes	No	No
(Kundu & Evans, 2014)	Yes	No	No	No

To successfully answer the scientific question given in chapter 1.2, it is necessary to be able to distinguish between natural and synthetic imagery, identify the impact synthetic or natural data has on the performance of computer vision algorithms and to determine the cause of this

resulting performance difference. Further, the approach should be generally applicable. Table 2-3 shows that none of the discussed approaches provides these capabilities.

Visual task performance based approaches: The most common approach is to determine the transferability by remodelling the scene of a natural dataset and validate the results concerning a specific visual task. For instance in (Nentwig et al., 2012) road vehicles had to be detected from the drivers perspective to prototype *Advanced Driver Assistance Systems* (ADAS). Thus, a test drive has been remodelled and restaged in a synthetic environment to generate a scene correlating synthetic dataset. The performance of both datasets then provided insight about the transferability of synthetically acquired results towards the real world for this exact use case and scenario. Testing different configurations of the synthetic environment showed that specific settings increased the correlation of synthetic and natural datasets leading to improved results of the tested computer vision application. Similarly (Hummel, Kovács, Stütz, & Szirányi, 2012) replicated an aerial record of a populated street in an synthetic environment to demonstrate the transferability of results. The application, an airborne object tracker, was evaluated and results showed that the tracker developed on natural data performed better on synthetic data, which was explained by the lack of simulated camera or environment distortions. Depending on the visual task, specialized datasets may be necessary. Such exist for instance for *face localization* (Sagonas, Tzimiropoulos, Zafeiriou, & Pantic, 2013) or *face-recognition* (P.J. Phillips et al., 2000). In these special applications, only the investigated subject needs to be reproduced. For example in (Wood et al., 2015) a synthetic dataset of images depicting a human eye has been generated and compared against the publicly available dataset of (Sagonas et al., 2013) for the tasks eye-lid localization and appearance-based gaze estimation. They identified that removing eyelid motion and using only one lighting condition reduced the performance of the synthetic dataset proofing these step important for replication.

CV-component performance based approaches: The aforementioned papers used complex computer vision applications consisting of several processing steps to investigate the usability of synthetic data. However, performance differences on natural and synthetic data have also been investigated for isolated computer vision algorithms. In (S. Meister & Kondermann, 2011) a simple engineered scene of a wooden block on a turntable was used to generate a natural dataset for optical flow estimation along the method of (Zach, Pock, & Bischof, 2007). The same scene was reproduced synthetically in different degrees of detail and the resulting optical flow error was measured. The synthetic dataset with least details showed the smallest

errors, confirming the statement that algorithms perform best on clean images due to missing optical effects. Adding error sources such as shadow, specularity and deploying a more complex shader increased the transferability of synthetic results. Still, the errors measured in the natural dataset are larger. The remaining differences in error are concluded to origin in *"suboptimal reproduction of surface and material properties"*. The overall optical flow on synthetic images is stated to be *"too smooth and 'well behaved' as the finer structure found in real world images is missing"*. The authors used a local illumination model to generate their synthetic data without using normal or specular maps to create microstructure on the surfaces of the wooden block. The authors conclude that geometry and texture quality are the driving issues of replicating real scenes since lens distortion and camera noise did result in no significant results.

The effects of synthetic data on segmentation algorithms are evaluated in (Irgenfried, Dittrich, & Wörn, 2014). Here, an engineered scene has been replicated using the CAD models of objects together with their light reflection and distribution properties (BRDF (Nicodemus, 1965)). The scene is depicted using a global-illumination rendering engine. The datasets are then compared by measuring the segmentation error. Resulting segmentation differences between the natural and synthetic dataset show great dependencies on the used segmentation method and scene. Segmentation differences between datasets are concluded to result mainly from deficiencies in the light source modelling. The results show that the error produced by the synthetic dataset to deviate by ~3% from the natural dataset for simple scenes and ~20% for complex scenes.

Image statistics based approach: These methods investigate the transferability of synthetic results to the natural domain by direct comparison of image statistics computed for both dataset types. These statistics are formulated considering the later used cv-algorithm, but no algorithm applied to validate the dataset. Thus, they demonstrate similar statistical properties even though the appearance may differ. For example, the MPI-SINTEL dataset (Daniel J Butler et al., 2012) is a large scale dataset for optical flow benchmarking consisting entirely of extracted scenes from the animated movie SINTEL (Blender Institute, 2010). Here, transferability of results is demonstrated by comparing seven image and motion statistics against documented *natural scene statistics*. Additionally, the dataset is accompanied by similar natural image sequences extracted from movies and videos called *lookalikes*. The statistics of synthetic, lookalike and natural data are then compared, showing small statistical differences between the three sets. This comparison is followed by an evaluation using several

optical flow algorithms (equal to the CV-component based approaches above), which perform worse on the synthetic dataset compared to the natural datasets. The authors explain this by greater complexity of their synthetic dataset.

A similar approach has been investigated in (Kundu & Evans, 2014) where the structural correlation between pixels and their neighbours are calculated using the luminance channel of the image. The results are then modelled using a distribution function and the coefficients of this model are then compared between natural and synthetic imagery. The author concludes *"that in the spatial domain, for pristine images, synthetic scene statistics can also be modelled in a fashion similar to natural scene statistics."*(Kundu & Evans, 2014). This suggests that synthetic and natural images follow the same underlying statistical principles and thus this approach can be used to characterize their differences.

3 Concept

The previous chapter presented current investigations about computer vision algorithm evaluation, natural and synthetic dataset comparison, computer graphic realism, image comparison methods and concepts measuring transferability of synthetically acquired results. As specified in chapter 1.2, these foundations now serve to define an experimental concept that allows to

- investigate the usability of synthetic datasets for computer vision algorithm developments and evaluation. (Objective 1)
- identify the underlying reasons for algorithm performance differences when being applied on natural and synthetic datasets by correlating them to image content properties. (Objective 2)
- identify rendering techniques influencing these properties. The results shall help to formulate recommendations for modelling artists and computer graphics programmers. (Objective 3)

Already existing approaches investigating the possibility to transfer results acquired from synthetic data to the real world are presented in chapter 2.6. While some show promising results, the scope of these approaches does not extend to the identification of underlying principles. For example, (Longhurst et al., 2003) investigated visual artefacts that raise the perceived realism for a human observer (chapter 2.4). (Nentwig et al., 2012) on the other hand analysed the performance of a computer vision algorithm on natural and synthetic data and investigated which rendering techniques positively influenced its performance. However, the underlying alteration in performance was not identified (chapter 2.5.2 and 2.6). Still this is of major importance if a synthetic environment shall be designed for benchmarking of computer vision algorithms. Insight, which image properties need to be modelled accurately and which can be neglected should not only boost the quality (transferability) of the results and increase the acceptance due to objective measures, but also reduce the benchmarking costs.

Considering conventional and new emerging evaluation methods (see chapter 2.6), a generic concept measuring the transferability of results and pinpointing the remaining performance differences of an evaluated CV-algorithm to the causing image properties is presented in the following subchapter.

3.1 General concept

The evaluation procedure of computer vision algorithms greatly depends on the type of algorithm, the intended *visual task* (e.g. vehicle detection) as well as the deployment- or test-scenario. Similarly, concepts identifying the validity of using synthetic data for cv-algorithm evaluation are depending on the same factors. These factors are considered in the general evaluation concept, which is intended to allow investigation of essential image properties and influencing rendering technologies to identify a trade-off between modelling detail and algorithm performance in relation to natural images. Further, it is intended to provide recommendations towards a benchmark simulation system that indicates to what degree the results from synthetic data are transferable to the real world. The concept presented in this subchapter was first published in (Hummel & Stütz, 2014)[11], refined in (Hummel & Stütz, 2015)[12] and is summarised and updated here.

The multi-level concept consists of four layers as depicted in Figure 3-1. The first layer comprises of *reference* and *test datasets*. *Reference data* are natural images depicting a scene to be processed by a specified CV-algorithm type. The corresponding *test data* are generated from a synthetic environment and depicts a remodelled variant of the reference data scene. It is necessary to generate the *test datasets* in the synthetic environment using a sensor model representing the deployed real sensor (resolution, FOV, distortion, noise, etc.). In this environment, the scene is replicated within the limits imposed by hard- and software.

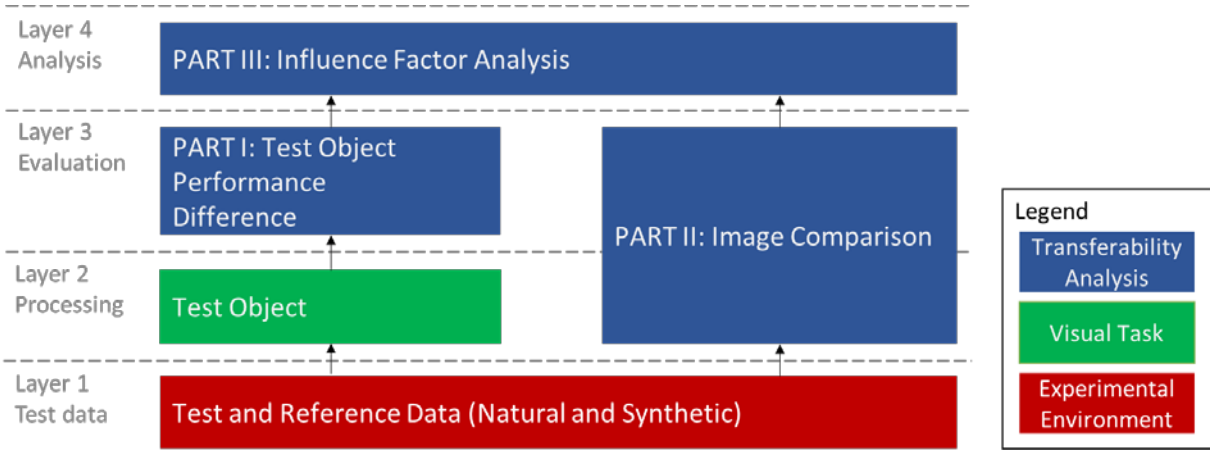


Figure 3-1: Simplified general multi-level concept.

[11] The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-13823-7

[12] The final publication is available at CSREA PRESS via ISBN: 1-60132-404-9

A *visual task* here is defined as an assignment that can be fulfilled using optically acquired data. A simple example would be to count the number of cars in an image. This can be realised technically using CV-algorithms. For instance in (Nentwig & Stamminger, 2010) the *visual task* to detect preceding vehicles with a dashboard mounted camera was implemented using a learned classifier based on Haar-wavelets (Viola & Jones, 2001). The *test object* is the actual algorithm selected to perform the *visual task*. The algorithm is located in level two together with the *image content description*.

After the *test object* is applied in parallel on consecutive reference and test images the algorithms performance on both image types is evaluated by comparing the calculated results to available ground truth. This is the first part of the *transferability analysis* and named *test object performance difference*. The performance on the reference dataset is selected as reference performance. If the synthetic performance is identical to the reference performance, then desired *functional realism* (see chapter 2.4) is considered to be achieved. The datasets are then identical for the *test object* in regard to the *visual task* and the scene. If the *test object* performs better on synthetic data, significant perturbations existing in natural images have not been modelled. On the other hand, if the test object performs worse on synthetic data, image details necessary for the test object are missing or the rendering process introduces perturbations not existing in natural data. In the proposed concept, the resulting *performance difference* is forwarded towards the *influence factor analysis*.

In the second part of the analysis, the *image comparison* the properties (e.g. colour, edges, etc.) of each dataset image are computed and saved as image descriptions. This allows quantification of image content for impartial comparison of image appearances. These described image properties are then used to compare reference and test dataset. Thus, the objective difference in content between these two dataset types, the *image content differences* can be numerically acquired. The results are then also forwarded toward the last layer.

On this last layer and part of the analysis, the *influence factor analysis,* the results from the previous steps are used to identify the individual impact investigated image properties have on the measured *performance difference.* This is achieved by analysing the behaviours of the algorithms *performance difference* concerning *image content differences.* Thus, revealing which image properties influence the *test object's* performance. Further, synthetic datasets of varying quality or enhanced by additional effects are also investigated to identify their effect.

Lastly, the *influence factor analysis* also helps to identify image properties not or insufficiently available in synthetic data.

In Figure 3-2, the general concept is further detailed. Level 1 consists of two different datasets and their ground truth:

- The *natural (reference) dataset* contains camera captured natural images of the scene. It is accompanied by its *ground truth*, which is necessary to evaluate the performance of a *test object*. A more detailed description of ground truth can be found in chapter 2.1.2.
- The *synthetic (test) datasets* are generated using a virtual environment depicting the same scene as the *reference dataset*. These *synthetic datasets* differ in rendering and modelling quality to identify influences of the rendering configuration. Synthetic datasets are also accompanied by ground truth.

The method of dataset generation is presented in chapter 4 and the resulting datasets are presented in detail in chapter 5.2.

The *test object* consists two identical instances of a CV-algorithm working on natural and synthetic images. Processing results are provided to the *object performance difference* module, which hosts a generally accepted *evaluation method* for this class of algorithm. The resulting performance values of natural and synthetic data are subtracted to acquire the performance difference.

Both datasets are also used by several different *image comparison* algorithms (each analysing one or more image properties) to provide a holistic quantitative description of each image. These descriptions are then received by a *distance measure* that calculates the distance between two images based on their quantified descriptions. While one total value is calculated, also the distances of each property are forwarded to the final analysis layer.

The *influence factor analysis* uses a method to identify factors significantly influencing *performance*, *image content* properties and *rendering configurations*. Here, changes in performance differences (due to different synthetic datasets) can be correlated to differences in image properties, thus allowing the identification of image properties significantly influencing the performance of the CV-algorithm. Furthermore, identified influencing image properties can be correlated to configuration parameters of the graphics engine by using the

results of synthetic datasets of different configurations. The resulting model will lead to insight which elements of the render pipeline are significant to design a synthetic benchmark for tested task and scenario.

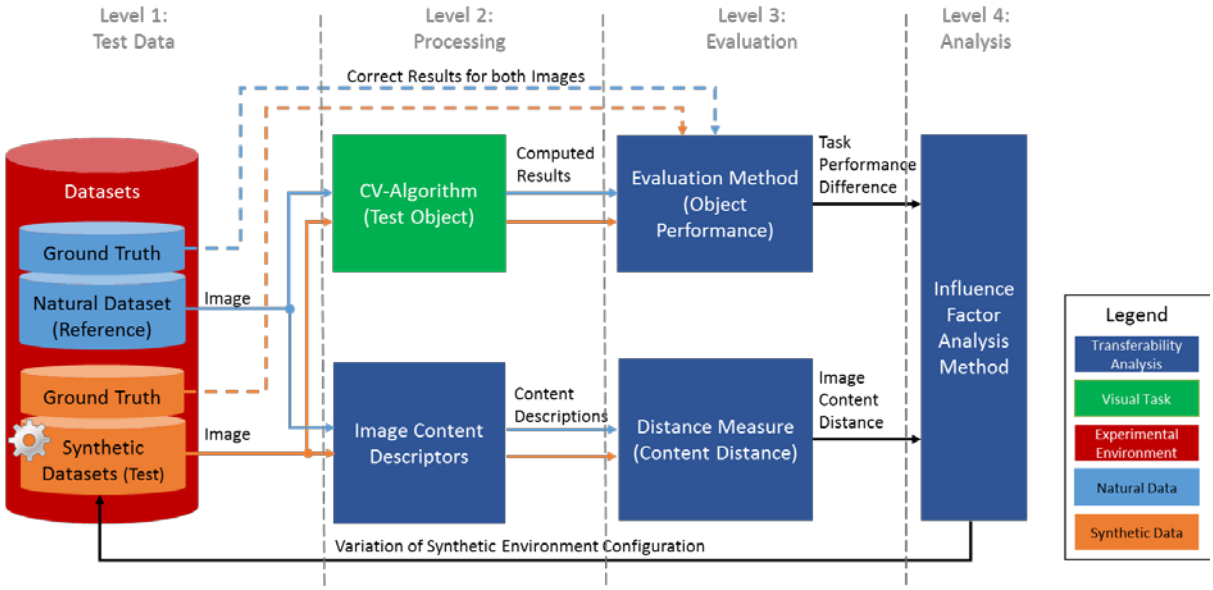


Figure 3-2: Detailed general multi-level concept separated into methods and data flow

This generic concept serves as a framework that can be applied to any scenario and task, whenever the discussed elements are filled with specific corresponding methods. The following chapters now describe an implementation variant derived from this concept as a demonstrator including scenario, task and the actual methods used for all abstract defined modules defined above.

3.2 Applied concept

In this section, the constraints of this work put on the general concept and the resulting applied concept are presented.

Constraints

A major motivation (see chapter 1.1) is to reduce the number of test flights necessary to prototype computer vision algorithms for airborne reconnaissance applications. Thus, the concept will be demonstrated in the airborne domain, while in general it is not limited to this domain. The natural and synthetic images will depict aerial photographs showing terrain

underneath the aircraft in flight. To reduce the complexity of the investigation the following constraints shall be observed:

- **Platform:** The camera is mounted to an aerial platform in a body fixed configuration, so that only the movement of the aircraft changes position and orientation of the camera.
- **Optical spectrum:** Camera images and synthetic images depict exclusively only the electromagnetic range visible to the visible eye (380nm – 780 nm (Schröder & Treiber, 2002)). This spectral range will be called electro-optical (EO) in the further course of this thesis. All images described are colour images, if not indicated otherwise.
- **Performance Measures:** All measures investigated in this work are related to quality, robustness or accuracy of the *test object.* Timely constraints are not considered, since these do not influence image quality (e.g. computational speed).
- **Real-time simulation engine:** Only real-time virtual simulation engines are considered for synthetic image generation. Such engines employ simple local illumination and show significant differences to the real-world allowing investigation of *functional realism*. Furthermore, these are usually used in the domain of flight simulators or tactical mission simulators, presenting the state of the art in the training & simulation industry.
- **Environmental Conditions:** All natural datasets used in this thesis are acquired during the day at sunny weather and acceptable wind speeds. This allows safe acquisition of image data without additional perturbations for a first proof of concept. Further investigations should address the impact of different weather conditions on the performance and its transferability to these real world conditions.
- **Scenario:** To demonstrate the proof of concept all objects except the airborne sensor platform and its camera are static to reduce the complexity of the experiments. Future work could investigate for instance vehicle trackers or other algorithms depending on object movement.

Resulting applied concept

After having the general concept outlined in section 3.1, its building blocks now have to be instantiated and detailed along the previously provided application constrains. The following table as well as Figure 3-3 summarize the results of this step. Readers interested in the rationales why certain methods have been chosen, will find this information in the next chapters.

Test Object	The CV-algorithms selected for evaluation are the commonly known and used *feature detectors SIFT* (Lowe, 2004), *SURF* (Bay, Tuytelaars, & Van Gool, 2006) and *MSER* (Matas, Chum, Urban, & Pajdla, 2002). *Feature detectors* are widely used to extract distinctive image features on which subsequent processing are based on. Therefore, these algorithms are often the first step in complex algorithms and are thus well suited for this evaluation. More details can be found in section 3.3.
Evaluation Method	With respect to the selected *test object* (Mikolajczyk & Schmid, 2002) and (Mikolajczyk et al., 2005) presented a *feature detector* evaluation method based on computed ground truth. This effective method allows to measure the *relative* and *absolute repeatability* of interest point detectors and has been used to describe the performance of the most known *feature detectors* (Lowe, 2004), (Bay et al., 2006). The workings of this method are presented in section 3.4.
Image Content Descriptors	A subset from the image comparison methods presented in chapter 2.5 has been selected in section 3.5. Well-known *image quality assessment* methods (MSE, PSNR) shall be applied to evaluate their ability to discern image properties. Further, the *image content descriptors* proposed in the MPEG7 standard (Sikora, 2001) are used to describe the image with simplified feature vectors. These descriptors are then tested against synthetic and natural datasets concerning their ability to discriminate image properties. The most promising will be used in the *influence factor analysis* (level four)

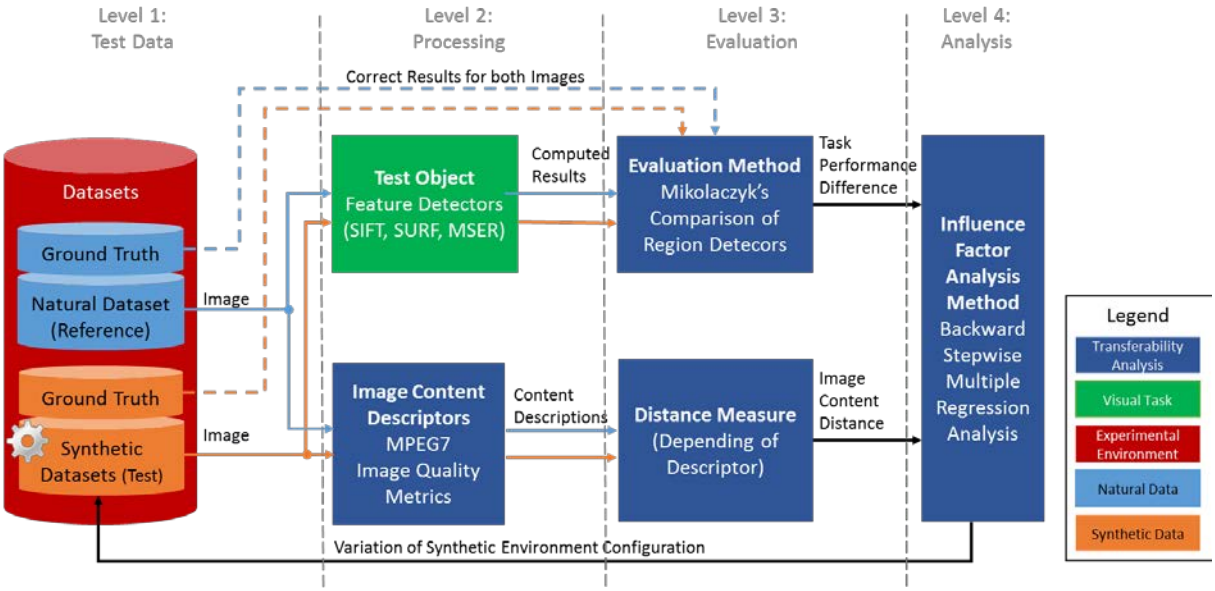


Figure 3-3: Applied concept used in this thesis to investigate the transferability of synthetically acquired algorithm performance results to the real world environment.

Distance Measures

While the MPEG7 *descriptors* come along with well-defined distances measures (chapter 2.5.4), measures without explicitly defined distance measures will employ the most suitable distance measures. The reason for selecting specific measures and their working principles are outlined in section 3.6.

Influence Factor Analyses

The relationship of performance differences and image content differences will be elaborated in layer four based upon a *multiple regression analysis* (explained in 3.7). This method reveals the effect of relationships (between performance and specific image properties) and their significance. Specifically, the *backward stepwise multiple regression analysis* variant is used, where all variables (image properties) are initially fed into the model and the least significant are stepwise removed from the regression model until only significant relations remain. The resulting model allows identification of the image properties the CV-algorithm is sensitive to, together with their amount of influence. It should be noted that this method allows the identification of multiple influencing image properties simultaneously and their degree of effect.

3.3 Test object

The selection of an appropriate test object is crucial to provide comprehensible test results to the reader.

Most CV-algorithms on application level consist of several basic methods chained together. Testing these chains would produce non-generalizable results only valid for this specific algorithm composition. Thus, each fundamental CV-algorithm should be tested separately. Typical airborne remote sensing applications (e.g. object detection, image registration or mapping) are based on detected image features. For instance the visual detection and tracking system of moving targets presented in (Siam & ElHelw, 2012) uses Harris corner features to identify possible foreground objects. While the analysis of density and motion of crowds in (Sirmacek & Reinartz, 2011) is based on FAST features. In (Brook & Ben-Dor, 2011) an image registration method is presented that uses features based on SURF as landmarks. In these examples, the extracted features are used for further processing making them the interface between the image (domain) and the subsequent processing components. Therefore, the performance of all following components can be assumed to depend on the behaviour of the feature detector. This makes them the ideal *test object* for a proof of concept. Ideally, a relevant *test object* should be common, well understood and publicly available to allow the reader insight into the evaluation process and results (not obscuring the results by using complex proprietary unknown algorithms).

Many algorithms have been developed in the past to detect image features. Commonly known representatives are *SIFT* (Lowe, 2004), *SURF* (Bay et al., 2006), *FAST* (Rosten & Drummond, 2005) or *Harris Corners* (Harris & Stephens, 1988).

The general idea of feature detectors is to identify easily detectable local gradient extremes that can be recovered in a subsequent image. These features are in general points due to the aperture problem, which describes that all pixels of an edge have similar properties and thus the actual location of an edge pixel cannot be recovered. Points on the other hand produce unique stand-alone gradients more robustly detectable in subsequent images. The *Harris corner detector* is a common feature detector, which uses the eigenvalues of a covariance matrix calculated from the spatial derivations of a local window in all directions (Harris & Stephens, 1988). This method is rotation- but not scale-invariant, which limits its use on airborne imagery with varying altitude or zoom.

Thus (Lowe, 2004) came up with the *scale invariant feature transform (SIFT)* detector. In this approach the goal is to detect distinctive key features in an image invariant to scale and orientation and robust to noise, affine distortions and change in illumination. This was achieved by using a three-step approach for local extremes detection. Step one uses pyramidal scaling of the image by factor two to determine extremes at different scales. This technique is called *scale space* and was first introduced by (Witkin, 1983). In each scale step (*octave*), the image is blurred in three intensities (different sizes of the Gaussian function) and the resulting images are subtracted from the first image of the scale. This approach helps in identification of stable local extremes robust to noise at multiple scale levels. The extremes are found by comparing the resulting difference of Gaussian value of the pixel against its spatial neighbours of the current and neighbouring scale within an *octave* and the highest ranked extreme is selected for further examination. Now, if the contrast to neighbouring pixels is too low the feature point is discarded. The location of the remaining points is now accurately pinpointed by interpolating derivatives of the neighbouring pixels, thus identifying the pixel closest to the local extreme. Since edges which are weak feature points, will also be detected by this approach the ratio of eigenvalues of the covariance is determined as in (Harris & Stephens, 1988) enabling the removal of features along edges, because they have large eigenvalue perpendicular and low eigenvalue horizontal to the edge. In a last step, the Gaussian smoothed images from the first step are used to determine the orientation of the feature point, thus a feature points is given by its location, scale and orientation. (Lowe, 2004) demonstrates the robustness to noise and their distinctive repeatability in a use case of object detection. This approach has been parametrized to be a trade-off between the number of possible key points and speed to allow near-real time computation as explained by the author.

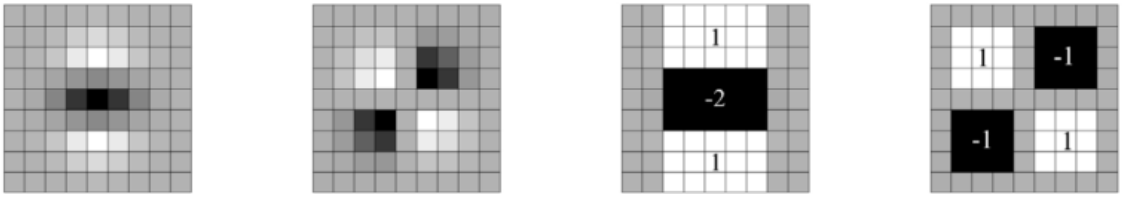


Figure 3-4: Visualization from (Bay et al., 2006) presenting Gaussian second order partial derivatives in *y*- and *xy*-direction as used by SIFT (Lowe, 2004) on the left and box-filter approximations as used in SURF in same dimension and direction on the right.

The *speeded up robust feature (SURF)* detector of (Bay et al., 2006) provides a simplified and thus faster feature detection method in regard to (Lowe, 2004). The author states that the importance of Gaussians in scale-space analysis has been overrated and thus provides a

simplified solution. Similar to *SIFT* it uses a Hessian matrix to identify feature points, which are invariant to scale and orientation. The scale space of *SURF* was highly simplified and thus speeded up. A combination of box filters (see Figure 3-4) and integral images replaces the Gaussian second order derivatives. Integral images $I_{\sum}(p)$ are images where the pixel value of $p(x,y)$ is the sum of all pixels between the origin $p(0,0)$ and $p(x,y)$:

$$I_{\sum}(p) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j) \tag{4}$$

The lowest level box filter (with a dimension of 9x9 pixel) approximates to Gaussian with a standard deviation of $\sigma=1.2$. The different scales are then achieved by applying box filters in various sizes (9x9, 15x15, 21x21, etc.). From here on the maximum is roughly located by searching the neighbouring pixels and neighbouring scales and the accurate maximum is then localized by applying the hessian matrix similar to *SIFT*. The author's state that this method outperforms *SIFT* in accuracy and speed.

The final feature detector algorithm considered in this thesis is *maximally stable extremal regions (MSER)* (Matas et al., 2002). Here, an extremal is a region of highest or lowest luminance intensity in relation to its surrounding pixels. These regions can be detected by thresholding the image at multiple different intensity values. Maximally stable refers to regions with minimal areal changes during the thresholding sweep of the image. Since this method is not location-based, small and large regions can be detected. However, limits for the maximum and minimum size of a region are set as well as limits for unstable regions (area changes too much). The method showed stability to scale-, viewpoint- and illumination-changes and was originally developed to identify wide-baseline correspondences using epi-polar geometry. In the study of (Mikolajczyk et al., 2005) the repeatability of *MSER* among other *Harris*- and *Hessian-based* approaches was evaluated. Presenting *MSER* and *Hessian-Affine* detectors with the best repeatability score followed by the *Harris-based detector* on most cases. However, the authors' state that the performance of the detectors is highly dependent on the scene and no single detector exists that outperforms others. Still years after this investigation, *MSER* is often used in today's computer vision applications. Especially in images depicting homogenous areas with clearly visible boundaries *MSER* outperforms other detectors. (Matas et al., 2002) state that *Hessian-Affine* and *MSER* define different areas as relevant features, thus the combination of both would provide best results at the expense of computation power.

Other feature detectors not evaluated in (Mikolajczyk et al., 2005) are rising in popularity due to their efficiency (e.g. *FAST* (Rosten & Drummond, 2005, 2006)) or performance in specific applications (e.g. STAR (Agrawal, Konolige, & Blas, 2008) for image registration). Still, even today *SIFT, FAST* and *MSER* are commonly used and remain popular and therefore shall serve as *test objects* in this study. This will help readers to more generally interpret the results and identify the chances and drawbacks when applying CV-algorithms on computer graphics imagery.

3.4 Object performance evaluation

The *object performance evaluation* is closely related to the selected *test object*, since the evaluation method is dependent of the algorithms output type and its fundamental purpose. In the survey of performance characterization in computer vision of (Thacker et al., 2008) the practices of evaluating feature detectors have been discussed. The first developed feature detectors such as (Harris & Stephens, 1988) demonstrated their functionality on a small set of images and by integration into robotic applications. The authors then discuss current methods and propose a probability based approach similar to (Ramesh, 1995). As described there, this approach assumes that ideal input data leads to ideal output data otherwise the algorithm is flawed. This input data is then modified using a probability based perturbation model, which is modelled for both error types, *false positives* and *false negatives*. The approach proposed in (Ramesh, 1995) however only considers noise as perturbation using a mean zero additive Gaussian as model. When considering a complex scene multiple disturbance source exist, which increases the complexity of the model and reduces the efficiency of the approach. Additionally, all existing perturbations need to be known before the performance of an algorithm can be evaluated. (Thacker et al., 2008) also discusses the work of (Mikolajczyk & Schmid, 2002), (Mikolajczyk et al., 2005), (Mikolajczyk & Schmid, 2005) who proposed the evaluation of feature detectors and descriptors based on geometric correlations. For instance, if a feature in image *A* is present in image *B* and both images depict the same static scene from different viewing angles, then the feature can be described by $f_B = M * f_A$ with *M* being the geometric transformation from view *A* to *B*. Now, if the transformation *M* between *A* and *B* is known the location of the feature is known, *M* becomes the ground truth. (Mikolajczyk et al., 2005) used this principle to evaluate performance characteristics of interest point detectors using natural imagery. The authors depicted only planar surfaces in their viewpoint evaluations enabling the use of *homography* as geometric transformation. In this respect, if

any of the following conditions are met, the geometric difference between two images can be described:

- Both images depict the same plane from a different viewpoint. Detected feature regions need to reside on this plane.
- Both images are taken from the same position at different viewing angles (e.g. image stitching).

Since this approach provides the possibility to (semi-)automatically produce ground truth when the location of the camera is known for every image, the approach of (Mikolajczyk et al., 2005) has been selected as *object performance evaluation* method. However, *homography* can only be used under the assumption that all detected features reside on a plane when the camera is moving. Due to a high enough altitude of the aircraft, the top-down perspective of the mounted camera and a suitable selection of the test area (only small objects present) this condition is adhered (and thoroughly evaluated in chapter 5.1.2). The used geometric transformation handles scaling, rotation and additionally translation when the depicted scene is planar. Features in (Mikolajczyk et al., 2005) are described as regions R on the image described by location and radius. The homography matrix H^{IJ} describes the geometric transformation from Image J to Image I and thus allows reprojection of the features R:

$$ {}^{I}R_J = (H^{IJ})^T \, {}^{J}R_J \tag{5}$$

The notation ${}^{I}R_J$ indicates regions detected in reference image J have been transformed into the image coordinate system of image I. The reprojected feature regions ${}^{I}R_J$ can then be compared against feature regions ${}^{I}R_I$ detected in image I (see Figure 3-5).

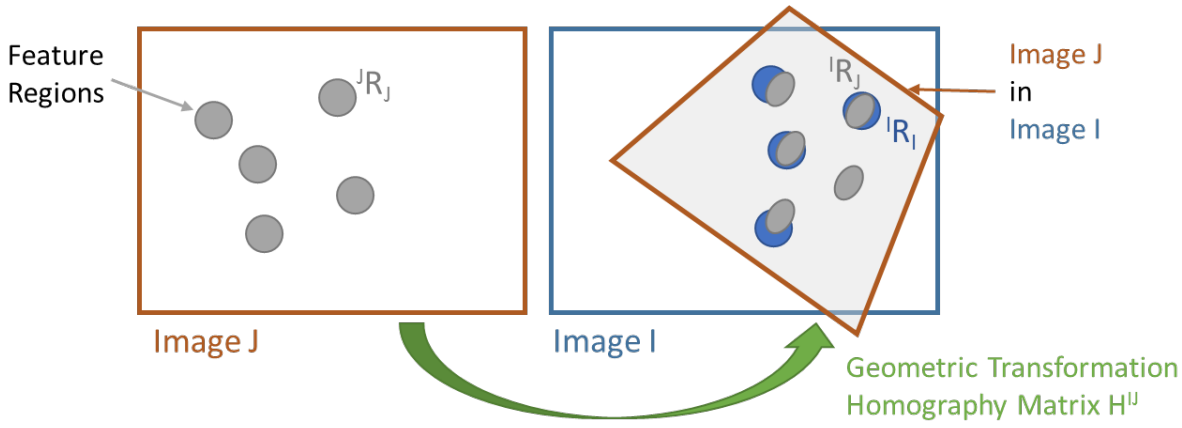


Figure 3-5: Feature regions of image J projected into image I using the homography matrix as geometric transformation.

The process to compute the ground truth (Homography Matrix H^{IJ}) is presented later in this chapter. Concerning measures of performance for feature detectors (Mikolajczyk et al., 2005) suggests to use:

> *"**Repeatability**, i.e., the average number of corresponding regions detected in images under different geometric and photometric transformations, both in **absolute** and **relative** terms (i.e., percentage-wise)."*

In this context, *relative repeatability* measures the amount of visible features that will be repeatedly detected in subsequent images and thus represents a measure of robustness. The authors introduced such repeatability measure based on the overlap of detected and transformed feature region:

$$1 - \frac{{}^{I}R_I \cap {}^{I}R_J}{{}^{I}R_I \cup {}^{I}R_J} < \epsilon_O \tag{6}$$

This means that a feature pair is corresponding when the *overlap error* threshold ϵ_O is bigger than the intersection of ${}^{I}R_I$ and ${}^{I}R_J$ divided by its union. The whole measure is subtracted from one to create an error or distance measure. The authors set the error threshold to 0.4 ≙ 40%. Only features present in both pictures are used to compute the relative repeatability. This performance measure is scale and orientation variant. The total number of valid corresponding pairs against the number of possible corresponding pairs (present in both images) is known as the *relative repeatability* measure, while the *absolute repeatability measure* simply counts the *number of valid corresponding pairs*. The latter is of importance since most applications need a minimum amount of feature pairs to perform correctly. On the other hand, too many feature pairs can result in long computation times and reduced repeatability due to detection of less robust interest points (Lowe, 2004). Therefore, most feature detectors have a configuration parameter to allow the user to configure the algorithm towards his needs. (Bay et al., 2006) also estimate *relative repeatability* as the most valuable property of a feature detector and therefore use it to demonstrate the performance of their SURF detector. Further (Lowe, 2004) in his work presenting the SIFT detector also applies *relative* and *absolute repeatability* as performance measure.

In Figure 3-6, two performance characterization examples for feature detectors are presented. In general, the *relative repeatability* is compared to the parameter that shall be evaluated. For instance, the left image displays the rank and performance differences of detectors concerning

the viewpoint angle of dataset “Graffiti”. The right image compares the *relative repeatability* of a detector with its *absolute repeatability* (number of detected corresponding regions) to show their characteristics in regard to the density of features.

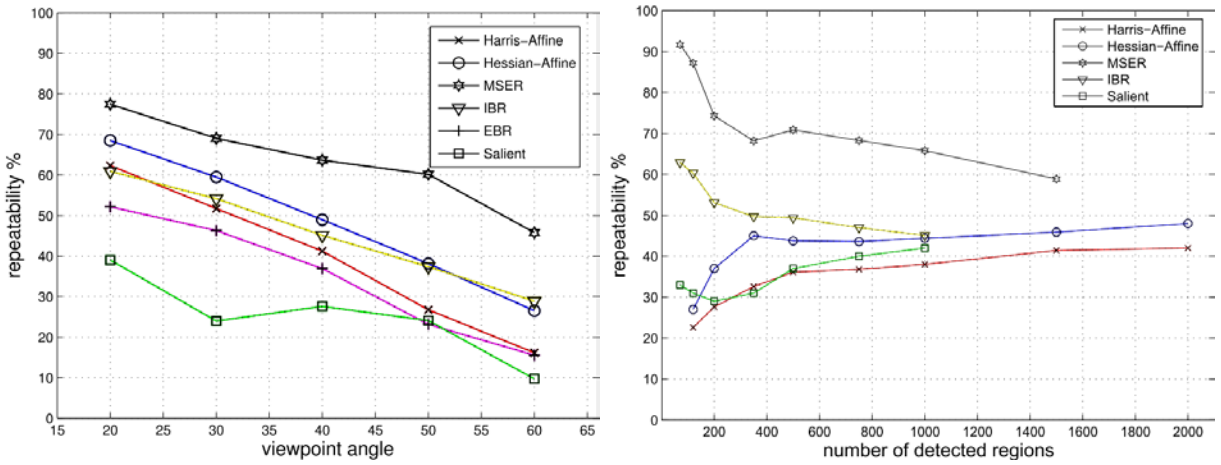


Figure 3-6: Example performance characterization of feature detectors using dataset *Graffiti*. Left: *Rel. repeatability* vs. viewpoint angle. Right: *Rel. repeatability* vs. *abs. repeatability*. (Mikolajczyk et al., 2005)

Ground Truth

The computation of bot metrics, *relative* and *absolute repeatability*, depend on the availability and accuracy of *ground truth*. For this thesis, the most relevant idea of Mikolajczyk’s performance evaluation is the computation of *ground truth* based on a reference image using the geometric correlation between the images (*homography*). In the authors two step approach they first compute an *approximate homography* is computed by using a small number of manually selected correspondences between reference image I and sequential image J. J is then warped using the acquired homography matrix to roughly align with image I. In a next step an automatic feature detection and matching mechanism *(brute force sum of squared distance (SSD)* matching) is used to find hundreds of correspondence pairs which are fed into the *RANSAC* algorithm (Fischler & Bolles, 1981) to compute the remaining homography (called *residual*). This algorithm iterates (up to 2000 times) to best fit the detected correspondence pairs into a plane, while identifying and excluding possible outliers. Afterwards the resulting *residual* (using *RANSAC*) and the *approximate homography* (using manually selected features) are combined to the final *homography* matrix. The resulting ground truth is then used to compare the detection performance of six affine region detectors under changes of viewpoint, zoom, rotation, image blur, JPEG compression and luminance. The results demonstrate in general that all detection algorithms perform similar to introduced

changes but with varying magnitudes, thus showing more robustness to perturbations then others.

Thus, in this thesis the ground truth computation is based on the just presented approach of (Mikolajczyk et al., 2005). However, the *approximate homography* is directly computed by finding correspondences using SURF feature detection and description (Bay et al., 2006). Finally the *SSD brute force matching* finds corresponding image pairs and the best fitting ones are used to initiate the RANSAC algorithm, which optimizes the *homography* based on the detected feature points. In this work, the manual process will be replaced by an automatic matching, to increase the number of images available for this evaluation. The impact of this modification is investigated in chapter 5.1.3.

Experimental Design

To test the effect of the rendering engines configuration several different datasets are used in which all configuration parameters are kept static except for one (*one-factor-at-a-time*). Other experimental designs such as *full factorial* or *fractional factorial,* which allow less test runs and knowledge about the relationship between parameters have been investigated (Box, Hunter, & Hunter, 2005). However, these demand the general behaviour of the system and the influence order of variables to be known. Since this is not the case, the more standard *one-factor-at-a-time* design is employed. However, these techniques are of high interest for further studies after the general order of variables has been identified. The performance relationship between real and synthetic imagery is measured in two steps as follows:

In step one, the feature detectors are employed on the natural dataset (reference) *photo* and the synthetic dataset “baseline” both depicting same scene and terrain. The performance difference is the acquired by simply subtracting the *relative repeatability* values of both datasets. The *absolute repeatability* results are normed before subtraction to improve the comparability among different feature detectors. This resulting performance difference $\Delta performance$ provides insight on the performance difference between the two datasets and which dataset produces higher *repeatability* values (when synthetic datasets perform higher the value is negative).

In the second step, the performance of the selected feature detector is measured for different rendering pipeline configurations by changing parameter *x*. For each pipeline configuration, a new synthetic dataset is generated and performance results are compared to the natural dataset

performance of the algorithm. The result $\Delta performance._x$ is given in percentage. *Performance* is a placeholder for the employed performance measures *relative* and *absolute repeatability*, thus it is computed for both measures. The wildcard *x* will be replaced by the respective configuration parameters of the rendering engine.

This approach shall identify parameters, which increase the usability of synthetic datasets by behaving more closely to their natural counterpart. On the other hand, this test will also identify database or rendering parameters having no observable influence.

3.5 Image comparison algorithms

Image comparison as described in chapter 2.5 shall be used to identify image characteristics relevant to the *test objects* performance. Therefore, in this chapter the rational to short-list promising candidates is given. Working principles of candidates are presented in chapter 2.5.

To select a suitable comparison metric it is helpful to discuss the nature of image content first. Thus, it is suggested to differentiate in *content* and *appearance*. While *content* actually describes *what* is depicted, *appearance* refers to *how* it is depicted. An image thus can depict the same scene but due to different sensors or simply a different camera configuration, *appearance* of the scene can be quite different. On the other hand, images can have similar appearance but depict different scenes. Both, appearance and content need to be compared to measure the difference between to images. *Appearance measures* are *global* (they evaluate pixels independent to their neighbours), while *content measures* are *local* (evaluate local changes considering the values of neighbouring pixels).

Further, in this chapter, *image properties* are grouped in the categories *frame* and *data properties*. *Frame properties* describe technical properties usually kept constant during the evaluation (e.g. resolution, bit depth, colour channels) and thus are not investigated. D*ata properties* describe image appearance and content (e.g. brightness, contrast, frequency, colour, etc.). Such *data properties* are the subject of interest for *image comparison*. Roughly, with regard to digital images *data properties* can be categorized in the following groups:

- **Luminance** is the brightness value of a pixel strongly correlated to the irradiance received by the sensor at that pixels location.

- **Colour** stands for information present in the three colour channels. It describes globally and locally objects or other content. Often specific colour spaces are used to extract the hue value and saturation information.
- **Frequency** stands for gradients present in the image, typically extracted using Fourier or wavelet transformation.
- **Shape** stands for vector-based descriptions of shapes that can be numerically extracted and put into semantic context. However, shapes are not directly relevant to feature detectors and will not be considered further in this work.

The *data properties* categorization scheme is used to correlate these to the common camera distortions and effects (*image characteristics*) presented in chapter 2.3. Afterwards the *image comparison* methods presented in chapter 2.5 are also flagged when they affect the given *data properties*. The main goal is to combine these to results, to acquire possible relationships between *image comparison* methods and *image characteristics*. This allows identification, whether these are not covered by given methods, redundant methods and subjective usefulness of these methods when deployed. This procedure is a tool to reduce the number of *image comparison* algorithms while still covering all *image property categories*.

Table 3-1: Image characteristics (see chapter 2.3) vs. data property categories. X marks a properties fitting to the specific property category.

	Appearance (global)			Content (local)		
Image characteristics	Luminance	Colour	Frequency	Luminance	Colour	Frequency
Blur (1/Clarity)	X		X			
Noise	X	X	X			
Chromatic Aberration		X	X			
Motion Blur				X		X
Compression Artefacts	X		X			
Geometric Lens Distortion	X		X			
Modelling Detail				X	X	X
Modelling Errors				X	X	X
Aliasing			X			X
Aperture	X	X				
Light Effects (e.g. HDR, Bloom)	X	X				
Texture Quality			X			X
Positional Accuracy (Location, Orientation, Scale) of Objects and Camera				X	X	
Colour Distribution	X	X				
Shadow				X	X	X

Beginning with the common distortions, their subjective disaggregation shows that these affect at least one of the groups resulting from the categorization of image properties. In Table 3-1, an X marks *image characteristics* fitting to the previously mentioned categories. This grouping shall help to identify, which *image characteristics* influence specific *data properties*.

Image comparison measures introduced in chapter 2.5 can now be correlated to the proposed *data property* groups in Table 3-2. The goal is to select a range of image property measures to cover all local and global data property categories.

Table 3-2: Common image descriptors vs. data property categories. Descriptors sensitive to specific property categories are marked (x). Yellow highlighting indicates selection for further use in this thesis. Dark grey highlighting indicates interesting candidates for further investigations.

	Appearance (global)			Content (local)		
Image Descriptors	Luminance	Colour	Frequency	Luminance	Colour	Frequency
MSE described in (Horé & Ziou, 2010)	X		X			
PSNR described in (Horé & Ziou, 2010)	X		X			
SSIM (Z. Wang et al., 2004)			X			
MS-SSIM (Z. Wang et al., 2003)			X			
IW-SSIM (Z. Wang & Li, 2011)			X			
SR-SIM (Lin Zhang & Li, 2012)	X		X	X		X
FSIM (Lin Zhang et al., 2011)	X	X	X			
MAD (Larson & Chandler, 2010)	X		X			
Visual Difference Predictor (Daly, 1992)	X		X			
JND (Lubin & Fibush, 1997)		X	X			
NSS (Sheikh, Bovik, & Cormack, 2005)			X	X		X
NIQE (Mittal, Soundararajan, & Bovik, 2013)	X	X	X			
Contrast (Ke et al., 2006)	X					
Brightness (Ke et al., 2006)	X					
Blur (Tong et al., 2004)	X		X			
Hue Count (Ke et al., 2006)		X				
Edge Distribution (Ke et al., 2006)						X
Focus (Ke et al., 2006)						X
Hue Comp (Tang et al., 2013)		X	X			
Scene Comp (Tang et al., 2013)			X			
Dark Channel (Tang et al., 2013)					X	X
Complexity Feature (Tang et al., 2013)				X		X
MPEG7 DCD (Ohm et al., 2002)		X			X	
MPEG7 SCD (Ohm et al., 2002)		X				
MPEG7 CSD (Ohm et al., 2002)					X	X
MPEG7 CLD (Ohm et al., 2002)				X	X	
MPEG7 EHD (Choi et al., 2002)			X			X
MPEG7 HTD (Choi et al., 2002)			X			

Image quality measures

Though algorithms based on *full-reference* image comparison are expected to be less suitable, as they grade the image quality independent to the type of perturbation, this hypothesis needs to be confirmed. Thus, the common the quality measures *mean square error (MSE), peak-signal noise ratio (PSNR) and the structural similarity index (SSIM)* have been selected to represent the classical *full-reference image quality measures*.

(Sheikh, Bovik, & Cormack, 2005) introduced a quality measure based on common statistics available in natural images, the *natural scene statistics (NSS)*. This algorithm needs reference images to learn the NSS model. The measure *natural image quality evaluator (NIQE)* (Mittal et al., 2013) is based on a "quality aware" collection of statistical features based on NSS and is distortion and opinion unaware. Therefore, the measure was selected to represent *no-reference quality measures* and NSS-based quality measures.

Application driven measures

These measures have been excluded from this selection, since these demand full applications proprietary to the use case. The most closely related evaluation to this category is the performance evaluation (part one of the analysis concept).

Content driven measures

The image descriptors defined in the MPEG7 standard cover almost the whole range of image data property categories. Due to their abstract description of images, they provide a good basis to measure the distance or similarity of synthetic and photographic images. Thus, all colour descriptors (DCD, SCD, CSD and CLD) and all edge descriptors (EHD and HTD) have been selected. Algorithms based on shape description are not used, since feature detectors do net benefit from shapes.

Possible relationships between image characteristics and image descriptors

The previous tables (Table 3-1 and Table 3-2) are used to correlate image characteristics with image descriptors in Table 3-3. Whenever a certain characteristic affects the same *data property category* as an *image descriptor*, it is potentially described by the descriptor (marked with X). The correlation aids the selection of image descriptors, by removing those not assigned to any image characteristic. Further, if a characteristic is not assigned to any

descriptor, it is potentially not covered by the evaluation and should not be further investigated. Thus, this step shall help to reduce the evaluation complexity and show, the range of characteristics covered by the selected image descriptors. The later presented experiments shall identify the actual relationships that are currently just assumed in Table 3-3.

Table 3-3: Image characteristics related to image descriptors based on image data property categories.

Image Characteristics \ Image Descriptors	MSE	PSNR	SSIM	NIQE	MPEG7 DCD	MPEG7 SCD	MPEG7 CSD	MPEG7 CLD	MPEG7 EHD	MPEG7 HTD
Blur (1/Clarity)	X	X	X	X					X	X
Noise	X	X	X	X		X	X	X	X	X
Motion Blur	X	X	X	X					X	X
Geom. Lens Distortion	X	X	X	X					X	X
Modelling Detail					X		X	X	X	
Modelling Errors					X		X	X	X	
Aliasing	X	X	X	X					X	X
Aperture	X	X		X	X	X				
Light Effects	X	X		X	X	X				
Texture Quality	X	X	X	X			X		X	X
Positional Accuracy					X		X	X		
Colour Distribution				X	X	X		X		
Shadow					X		X	X	X	

3.6 Image content distance measures

Distance or similarity measure are methods to quantify the difference or similarity between two values or descriptions. In the applied concept (chapter 3.2), the need for distance measures has been formulated to compare the given content descriptions of images. The content descriptors are given in chapter 2.5.3. While comparing single values is simple, the topic becomes more demanding when multi-dimensional image content descriptions need to be compared. In this chapter, for each selected image descriptor a fitting distance measure is added. For more detail please refer to chapter 2.5.4, where most measures have already been introduced.

The appropriate distance measure for image descriptors defined in the MPEG7 standard have been defined already in most cases. In literature, often deviates (weighted, normalized) of *Manhattan*- (L_1) or *Euler-Distance* (L_2) have been used. In (Eidenberger, 2003b) the performance of image retrieval based MPEG7 visual descriptors with various distance

measures has been evaluated showing that the most appropriate distance measure for EHD, HTD and SCD is the *Pattern Difference* (Eidenberger, 2003b) after (Sint, 1975), for CLD the *Meehl Index* (Meehl, 1997) and for DCD the *Divergence Coefficient* (Clark, 1952). However, for all descriptors the measures originally defined in the standard also perform sufficiently. Thus, in this thesis those standard measures are employed, since they are more widely used and last level optimization is no focus of this thesis. Such investigation could be conducted in future to evaluate the influence of different distance measures on the correlation of these coefficients to the *synthetic configuration parameters* or *test object* performance. The following paragraphs present the selected distance measure for each employed image descriptor.

EHD: In (Choi et al., 2002) the matching method for the *edge histogram descriptor* is presented. The 80 bins of the local-edge histogram are used to compute a 5-bin global edge histogram and a 65-bin ($13 * 5$ bins) semi-global edge histogram. The distance $D_{EHD}(EHD_A, EHD_B)$ between the EHD descriptors of image *A* and image *B* is then computed using equation (7) by adding up the *Manhattan-Distance* (L_1) of each histogram.

$$D_{EHD}(EHD_A, EHD_B) = \sum_{i=0}^{79} |h_A(i) - h_B(i)| + 5 \times \sum_{i=0}^{4} \left|h_A^G(i) - h_B^G(i)\right| + \sum_{i=0}^{64} \left|h_A^S(i) - h_B^S(i)\right| \tag{7}$$

Where $h_A(i)$ and $h_B(i)$ are the individual bin values of the normalised local edge histogram of image *A* and image *B*. Correspondingly the bin values of the global edge histogram are represented by $h_A^G(i)$ and $h_B^G(i)$ and the semi-global histogram values by $h_A^S(i)$ and $h_B^S(i)$. To compensate the small values of the global histogram are small in comparison to the other two a weighting factor of 5 is introduced (Choi et al., 2002).

HTD: The distance measure $D_{HTD}(A, B)$ of the *homogenous texture descriptor* is also based on the L_1-measure. The HTD-descriptor contains the mean f_{mean} and standard deviation f_{stdev} of the images frequencies and the mean energy e_i and energy deviation d_i for each of the 30 image frequency channels presented in Figure 2-14.

$$HTD = [f_{mean}, f_{stdev}, e_1, \cdots, e_{30}, d_1, \cdots, d_{30}] \tag{8}$$

$$D_{HTD}(HTD_A, HTD_B) = \sum_{i=0}^{61} \left|\frac{HTD_A(i) - HTD_B(i)}{\alpha(i)}\right| \tag{9}$$

Equation (9) shows the distance measure $D_{HTD}(A,B)$ as defined by (Choi et al., 2002) and the MPEG7 standard. The measure is normalized by the weighting variable $\alpha(i)$ which should be the standard deviation of $HTD_B(i)$. In the experimental model of the MPEG7 standard (Yamada et al., 2001) fixed values between 0.22 and 1 have been used.

DCD: The *dominant colour descriptor* reasons for up to seven dominant colours (RGB) together with their areal presence. The original distance measure presented in (Ohm et al., 2002) and implemented in the experimental model (XM) of the MPEG7 standard (Yamada et al., 2001) uses a deviation of the Euler distance L_2 enhanced by the similarity coefficient $a_{Ai,Bj}$. p_{Ai} and p_{Bj} represent the areal presence of a dominant colour in image A and B. The similarity coefficient $a_{k,l}$ adds a colour based similarity measure to the distances measure to only measure the distance between two closely corresponding dominant colours.

$$D^2_{DCD\_ORG}(DCD_A, DCD_B) = \sum_{i=1}^{N_A} p_{Ai}^2 + \sum_{j=1}^{N_B} p_{Bj}^2 - \sum_{i=1}^{N_A}\sum_{j=1}^{N_B} 2a_{Ai,Bj}p_{Ai}p_{Bj} \tag{10}$$

This measure is criticized by (Yang et al., 2008) because it may lead to incorrect ranks for images with similar colour distribution. Therefore, they propose their own distance measure

$$D^2_{DCD\_YANG}(DCD_A, DCD_B) = 1 - \sum_{i=1}^{N_A}\sum_{j=1}^{N_B} a_{i,j}S_{i,j} \tag{11}$$

where $a_{i,j}$ is the colour similarity coefficient based on the L_2-measure

$$a_{i,j} = \sqrt{\left(r_{Ai} - r_{Bj}\right)^2 + \left(g_{Ai} - g_{Bj}\right)^2 + \left(b_{Ai} - b_{Bj}\right)^2} \tag{12}$$

and $S_{i,j}$ the similarity score between two areal percentages of dominant colours

$$S_{i,j} = \left(1 - \left|p_{Ai} - p_{Bj}\right|\right) \times min\left(p_{Ai}, p_{Bj}\right) \tag{13}$$

This distance measure copes with the drawbacks of $D^2_{DCD\_ORG}(DCD_A, DCD_B)$ and puts more emphasis on the colour similarity. Therefore $D^2_{DCD\_YANG}(DCD_A, DCD_B)$ is selected as distance measure for DCD in this thesis.

SCD: The *scalable colour descriptor* is in general a colour histogram in the HSV colour space, encoded using Haar-transform and linear / non-linear quantization (Ohm et al., 2002).

The distance $D_{SCD}(SCD_A, SCD_B)$ between two descriptors is computed using the Manhattan distance of each Haar-coefficient c_X

$$D_{SCD}(SCD_A, SCD_B) = \sum_i \left| c_A\ (i) - c_B\ (i) \right| \tag{14}$$

CSD: The *colour structure descriptor* is a colour histogram where each bin represents the normalized number of appearance of a colour in an 8x8 search window. The colours are described in the HMMD colour space defined in the MPEG7 standard (Ohm et al., 2002). The distance $D_{CSD}(CSD_A, CSD_B)$ between two resulting encoded histograms (colour structure descriptors) is computed using the L_1-measure.

$$D_{CSD}(CSD_A, CSD_B) = \sum_i \left| h_A\ (i) - h_B\ (i) \right| \tag{15}$$

CLD: The *colour layout descriptor* describes the spatial distribution of colours. The extracted spatial colour information is encoded using discrete cosine transformation (DCT) to 64 coefficients for each channel of the YCbCr colour space. The distance $D_{CLD}(CLD_A, CLD_B)$ is calculated by summing the weighted *Euler distances* (L_2) for all colour channel coefficients y_x, cb_x, cr_x of image *A* and *B*. Each coefficient is weighted individually. Lower frequency components are given larger weights w_x (Ohm et al., 2002).

$$D_{CLD}(CLD_A, CLD_B) = \sqrt{\sum_{i=0}^{63} w_{yi}(y_{Ai} - y_{Bi})^2} + \sqrt{\sum_{i=0}^{63} w_{cbi}(cb_{Ai} - cb_{Bi})^2} + \sqrt{\sum_{i=0}^{63} w_{cri}(cr_{Ai} - cr_{Bi})^2} \tag{16}$$

MSE: The *mean square error measure* introduced in chapter 2.5.1 combines description and distance calculation in one measure, calculating the normalized L_2-distance of luminance pixel values for image *A* and *B* (Horé & Ziou, 2010) The Eq. (1) is presented on page 35.

PSNR: The *peak signal to noise ratio* is in fact a derivation of MSE on a logarithmic scale. Equation (2) describes the measure mathematically on page 36.

MSSIM: The *Structural Similarity Index (SSIM)* (Z. Wang et al., 2004) detailed in chapter 2.5.1 is a combination of contrast, luminance and structure. Each property is compared using

the harmonic mean measure (Cha, 2007). The resulting three values are then multiplied leading to the final form of the SSIM quality difference measure(Z. Wang et al., 2004):

$$SSIM(A,B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)} \tag{17}$$

Where μ_A and μ_B (measure for global luminance) are the mean intensities μ_X for each image and σ_A and σ_B their standard deviations σ_X (estimate of signal contrast). They are defined as following (N is the number of pixels) (Z. Wang et al., 2004):

$$\mu_X = \frac{1}{N_X}\sum_{i=1}^{N_X} p_i \qquad \sigma_X = \left(\frac{1}{N_X - 1}\sum_{i=1}^{N_X}(p_i - \mu_x)^2\right)^{1/2} \tag{18}$$

Here, p_i represents the luminance value of a single pixel (grey value). The author states that this measure is best applied locally rather than globally, meaning that the local statistics of μ_X, σ_X, are computed using a local search window also called kernel, which is i.e. an 8x8 pixel sized 2D-filter that is convoluted with the image (calculating the result for each pixel location). This local measure is then combined to the final global measure called *mean-SSIM* or **MSSIM** by computing the mean result of all local measures.

NIQE: In (Mittal et al., 2013) the no-reference measure *natural image quality evaluator (NIQE)* is presented, which compares extracted statistical features of an image against a natural scene statistic (NSS) model:

"Our new NR OU-DU IQA [no-reference opinion-unaware distortion-unaware image quality assessment] model is based on constructing a collection of 'quality aware' features and fitting them to a multivariate Gaussian (MVG) model. The quality aware features are derived from a simple but highly regular natural scene statistic (NSS) model. The quality of a given test image is then expressed as the distance between a multivariate Gaussian (MVG) fit of the NSS features extracted from the test image, and a MVG model of the quality aware features extracted from the corpus of natural images." (Mittal et al., 2013)

The coefficients of the model are stated to follow a Gaussian distribution when computing natural images of low distortion. Is distortion added or are images synthetically generated the trend of the distribution changes. Thus, as a distance measure the mean vectors ν_A and ν_M (of *image A* and *MVG model M*) and covariance matrices of the natural image based MVG model

Cov_M and the computed test image MVG model Cov_A are compared using D_{NIQE}. The distance between two NIQE measurements is then computed using the *Euler distance*.

$$D_{NIQE}(v_A, v_M, Cov_A, Cov_M) = \sqrt{\left((v_A - v_M)^T \left(\frac{Cov_A + Cov_M}{2}\right)^{-1} (v_A - v_M)\right)} \quad (19)$$

The aforementioned paragraphs listed all distance measures used to compute the difference of two images based on the image content descriptors selected in chapter 3.5. When no distance measure is specifically defined by the author of the descriptor a derivate of the *Minkowski distance* is employed in this thesis.

Before the actual experiments, the individual measures are validated for their capability to determine image content differences. Here unsuitable descriptors are ruled out. The validation is presented in chapter 5.1.4. The actual experiment is structured in two parts:

The **first test** computes the distance $D_{Measure_{baseline}}$ between the default synthetic dataset *baseline* and the natural dataset *photo* for each image descriptor to identify the differing image properties between these two image types.

The **second test** computes the distances $D_{Measure_{parameter}}$ between a synthetic dataset with modified environment parameters and the natural dataset *photo* for each image descriptor to identify settings reducing the distance between these two image types.

3.7 Influence factor analysis method

The *influence factor analysis* shall relate the results acquired in the *object performance evaluation* (chapter 3.4) and *image content comparison* (chapter 3.5 and 3.6). As laid out before, the *object performance evaluation* provides the performance of the *test object* in regard to the (graphical) configuration of the rendering engine. These configuration settings affect appearance and content of the images, which is demonstrated by the *image content comparison*. Thus, combining these two evaluations based on the configuration of the synthetic environment allows isolating specific image properties influencing the performance of the test object and quantifying the amount of influence. This connection provides a fundamental understanding for the synthetic image composition and thus characterize synthetic data and its difference to natural data.

Studies investigating similar topics

In chapter 2.6, several approaches of other authors characterizing the difference between synthetic and natural data are presented. For example, (D. J. Butler, Wulff, Stanley, & Black, 2012) used image statistics based on histograms for luminance, spatial and temporal derivatives, gradient magnitude and the power spectra, which are averaged over a number of images to present the statistical difference of synthetic images to lookalike ("similar scene") screenshots taken from cinema and television movies. A measure to quantify the distance between luminance histograms is the *Kullback-Leibler divergence*:

$$D_{KL}(H_A, H_B) = \sum_{x=0}^{255} H_A(x) \cdot \log \frac{H_A(x)}{H_B(x)} \tag{20}$$

The numeric value to describe the derivate-based measures is simply the kurtosis of the (assumed) normal distribution. The power spectrum measure compares the slope of power to the image frequency. For this thesis, the presented approach is not sufficient because the originating source causing the algorithms performance difference cannot be extracted.

In (Avcibaş et al., 2002) image quality measures are evaluated in regard to four different image distortions using an *ANOVA (Analysis of variance)* test. This test considers variances of a dependent (output) variable to estimate whether the observed difference (of introduced changes) is due to chance or systematic. ANOVA can evaluate several different independent variables (groups) towards a specific hypothesis. The null hypothesis is valid when the mean values of all variables are equal (no influence of manipulated values) and thus shows that no effect has been found. The alternative hypothesis is becoming probable as soon as one group is significantly different from all the other means. This indicates that even though the result shows the alternative hypothesis to be significant, it is unknown which group is responsible for this difference. (Avcibaş et al., 2002) combined only different magnitudes of image distortions of the same kind within one ANOVA test, which allows them to detect whether an enlarged distortion has an effect on the tested measure. The authors used the F-ratio as measure for the variance explained by the ANOVA results against the remaining unexplained variance. The ratio is also a measure for the likelihood that such value is possible while the null hypothesis is valid. In this case, lower value yields a higher possibility. Therefore, the employed method can be considered suitable for the performed study. However, two drawbacks exist when this method is to be used for the *influence factor analysis*: Firstly, in the concept of this theses each synthetic environment parameter is represented by only one

dataset (not multiple increasing magnitudes), which means the variable responsible for failing the null hypothesis cannot be identified and further analysis would be necessary. Secondly, it is difficult to rank the input variables according to their weight on the output variable. Though, according to (Field, 2009) *t-test* and *ANOVA* are typical tests for experimental research (manipulation of one dataset and comparison to a reference, baseline or control dataset), the author also indicates that *regression* can be used to acquire cause-and-effect results (solving the rank problematic).

(Oelbaum, 2008) for instance uses *regression* in a two-step approach to measure the influence added image distortions have on image quality measures. First, the author describes the variance of all distortions performing a *principle component analysis (PCA)*. This step strongly reduces the computational effort while retaining all effects, but the relationship to the original variables (before grouping) is lost. Then the resulting principle component variables are analysed using the *partial least square regression*. While regression itself is a powerful method to identify the influence of multiple input variables on the outcome, a method without PCA is needed to allow identification of relevant input variables.

Selection of analysis method

The aforementioned studies support the idea that *multiple regression analysis* (several input variables, one output variable) is most suitable to evaluate the effect of image properties on CV-algorithm performance. Essentially, in regression analysis the output (outcome, dependent) variable and the input (predictor, independent) variables are used to fit a linear model describing the relationship predictors have on the outcome variable. This means regression estimates the amount of variance in the outcome caused by a specific predictor variable and its weight (compared to other predictors). If the resulting model is generally valid, it could be used to predict outcomes solely based on predictor values. The model is fitted by applying the method of least squares to identify the regression coefficients b_n of the model equation Y_i (Field, 2009):

$$Y_i = (b_0 + b_1 X_{i1} + b_2 X_{i2} + \cdots + b_n X_n) + \varepsilon_i \quad i = \{1, 2, \ldots, N\} \tag{21}$$

where i is the current sample, N the number of samples, n the number of predictor variables (input), X_{in} the value of these variables and ε_i the remaining difference between model value and the outcome Y_i. The regression coefficient b_0 is the static offset also called *intercept* of the regression model and can be deduced from the diagram of Figure 3-7 where the predictors

cross the vertical axis (all predictors = 0). This parameter allows the model to be different from zero in case all x values are zero. The orange and blue line stand for predictor examples. In this example b_1 is indicated by the orange line showing a strictly monotonic decreasing relationship with the outcome variable (b_2, the blue line is strictly monotonic increasing). The coefficients b_1 and b_2 describe the gradient of the lines depicted. The sum of predictor values is the result of the regression model and ε_i indicates the differences to the actual value of Y_i.

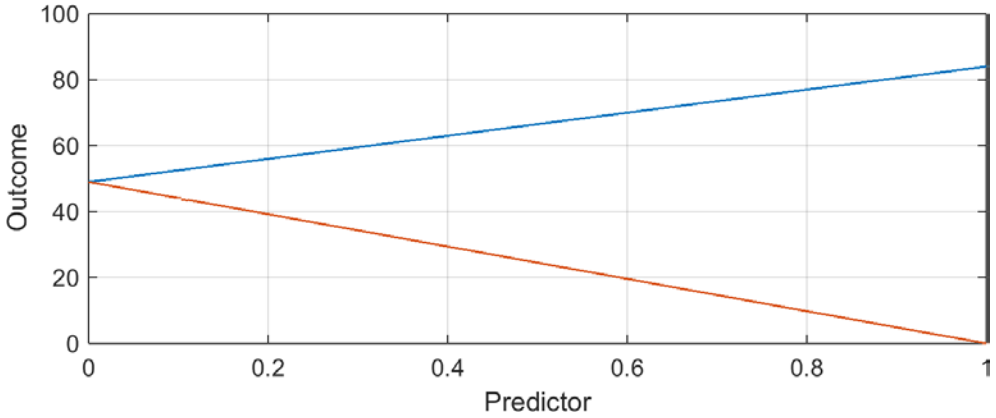


Figure 3-7: Example of the relationship of two predictors and the outcome variable. (Field, 2009)

Different *regression* methods exist, which vary by the concept on how variables are entered into the model. In general the most important or effective predictor should be added first into the model. In *hierarchical regression,* the predictors are selected based on findings in previous research, while *forced entry regression* adds all variables simultaneously at no specific order. In this thesis, *stepwise regression* is used because it allows adding predictors gradually and displays the benefit of the added predictor. More specifically the *backward stepwise regression analysis* is selected for this thesis. This methods starts with all predictors added to the model (just like *forced entry regression*), but then calculates the benefit of each predictor and removes the least contributing from the model until only significantly contributing predictors remain (Field, 2009).

Assumptions to be checked before analysis

Before performing *regression analysis* the image properties need to be analysed for relationships between these assumingly independent variables. If two or more predictors (these are the image descriptors my case) are correlating highly with each other, the resulting coefficient estimates will be inaccurate. This makes it difficult to assess the importance of a predictor. A well-known diagnostic measure for this effect called *multicollinearity* is the *condition index* (CI). This measure is acquired by dividing the largest eigenvalue λ_{max} of the

predictor matrix (measurements by descriptors) by the eigenvalue of the current dimension λ_x (Belsley, 1991):

$$CI_x = \frac{\lambda_{max}}{\lambda_x} \tag{22}$$

The higher the value, the more collinearity exists. During the analysis, a CI value for each dimension of the matrix is computed. The main interest is directed towards the dimension with the largest CI values. A value of ten to 30 is considered small and might be further investigated. Is 30<CI<100 then there are collinearity issues that should be investigated. When CI>100 the effect is severe and is sure to influence the *regression analysis*. The results of the investigation are best presented by table or table plot (Friendly & Kwan, 2009) (see chapter 6.1.3). If collinear variables are detected several possibilities exist depending on the goal of the analysis. For instance the model fit is unaffected by these effects and in this case the results can be neglected. If the contributions of the predictor variables are of interest as in this case, the effects of multicollinearity needs to be reduced by removing one of the responsible predictors (image measures). Removing an image measure sounds severe, however remember that a very similar image measure remains and thus only a repetitive measure is removed. For example, PSNR and MSE have the same measurement principle except that PSNR is of logarithmic scale. The intercept, included into the model to allow outcome values different from zero when all predictors are zero is not considered during multicollinearity evaluation as it is not an investigated parameter and does not hold interpretable meaning (Freund & Littell, 2000). This could be changed by centring the data (subtract the mean) but is advised against by (Belsley, 1991) and thus not conducted in this concept.

Execution of Analysis

After having identified the proper analysis method, now *relative-* and *absolute-repeatability* (chapter 3.4) and the *image content distances* (chapter 3.5 and 3.6) shall be used to populate the *regression* model. Since the goal is to explain the *feature detector performance differences* on natural and synthetic imagery through *image content differences*, the regression outcome variable *y* is defined by

$$\mathrm{y} \equiv \Delta Repeatability = Repeatability_{synthetic} - Repeatability_{photo} \tag{23}$$

where y is a vector of size (N-1) with N being the number of images in a dataset. The predictors are populated by the corresponding image content description distances. In Figure 3-8, the correlated measures are depicted. *Repeatability* is computed using two consecutive images of one configuration (image type) until the end of the dataset is reached (N-1 measurements). When the *repeatability* measures of two configurations are subtracted, four images are used to compute the dependent variable *y*. Thus, the same images need to be applied for the *image content distance* computation. This is achieved by computing the mean of two consecutive image distance results. The descriptors measure the *image content distances* $D_{Between,i}$ ($1 \times p$ -vector with p being the number of image descriptors) between the photograph and synthetic image depicting the same view for $i = 1, \dots, N$. To ensure comparability the mean results of each descriptor d_j (for $j \in$) for $D_{Between,i}$ and $D_{Between,i+1}$ are computed:

$$X := \frac{(D_{Between,i} + D_{Between,i+1})_{i=1,\dots,N}}{2} \triangleq \frac{(d1_{ij} + d2_{ij})_{i=1,\dots,N;j=1,\dots,p}}{2} \tag{24}$$

The resulting matrix X of size $(k \times p)$ matrix with $k = N - 1$ is now directly comparable to the performance difference y as it is based on the same data.

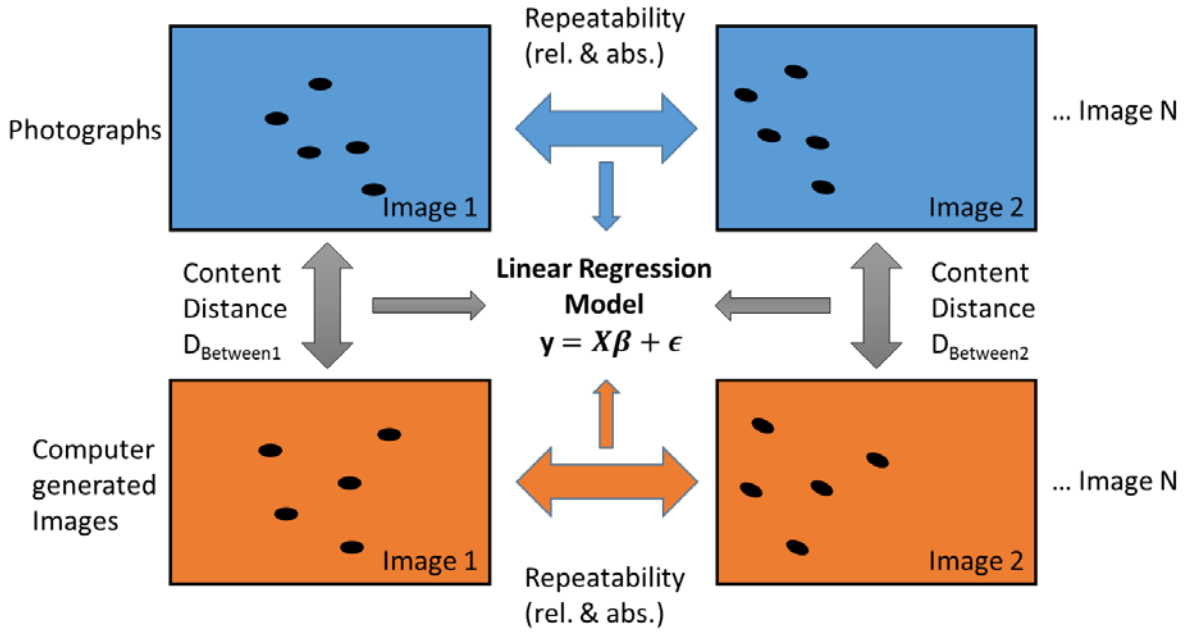


Figure 3-8: Evaluation principle of the influence factor analysis. *Repeatability* measures *y* and *image content distance* measures *X* are combined to populate the *linear regression model*.

After removing highly correlating image content descriptors due to multicollinearity, the regression fit can be conducted and the resulting model evaluated. There are several measures

to identify the fitting quality of the model. *R²* computes the amount of variation covered by the model. Thus if R² = 0.80, 80% of existing variation is covered by the resulting model. According to (Field, 2009) It is computed as follows:

$$R^2 = \frac{model\ sum\ of\ squares}{total\ sum\ of\ squares} = \frac{SS_M}{SS_T} \tag{25}$$

Where SS_M is the squared sum of residuals for the found model and SS_T is the squared sum of residuals using the mean $\bar{y}$ as model.

The square root of *R²* reveals the *Pearson Correlation Coefficient* (PLCC) *r* between model and data, the higher the value the better the fit. The coefficient can be described by the covariance cov_{xy} of two variables divided by the multiplication of their standard deviations s_x and s_y:

$$PLCC = \frac{cov_{xy}}{s_x s_y} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{(N-1)s_x s_y} \tag{26}$$

Applied to our problem the PLCC sums the difference for each result i and the mean of each dataset (x, y). The sum is then normed by their standard deviations. Thus, when the residuals of the different datasets behave similar a high correlation is measured. The significances of the correlations are calculated using Student's t-distribution. The measure p indicates the probability that the null hypothesis (the assumption the correlation is zero) is true. There are several threshold used in statistics with 5% probability being the most common one, meaning when the probability of the result being part of the population (null hypothesis being true) is greater than 5% the result is not significant. Other common thresholds are $p < .01$ (99% confidence) and $p < .001$ (99.9% confidence). When the resulting probability is less than the mentioned thresholds the alternative hypothesis is significant (the assumption a correlation exists). Thus, while PLCC presents the amount of correlation between x and y, p indicates the chance that the found correlation does not exist even though it is measured due to unfortunate sampling. It is important to mention that the identified correlation may be dependent on a third undiscovered variable. To measure the amount of shared variance, the coefficient of determination $R^2 = PLCC^2$ can be used. Additionally, a correlation test cannot identify the variable causing the correlation.

Another existing measure is the *F-Ratio*, which describes *"how much the model has improved the prediction [in regard to the mean (null hypothesis)] of the outcome compared to the level of inaccuracy of the model"* (Field, 2009). Thus, it can be used as a measure to rate the quality of the model. This measure results from dividing the mean squares of the model by the mean squares of the remaining differences (Field, 2009):

$$F = \frac{systematic\ variance}{unsystematic\ variance} = \frac{mean\ squares\ of\ model}{mean\ squares\ of\ residuals} = \frac{MS_M}{MS_R} \tag{27}$$

This measure presents the prediction quality of the model against its inaccuracy. Thus a large *F-ratio* (F<1) indicates a good fit. *p* indicates the *probability* of the computed F-ratio being a result of chance (see chapter 3.4 for more detail). A value of 0.05 for *p* means the probability of this *F-ratio* happening by chance is 5%. The F-Ratio can also be computed from the R^2 with the number of samples and the number of predictors (input variables):

$$F = \frac{(N-k-1)R^2}{k(1-R^2)} \tag{28}$$

In reverse, the number of necessary samples is dependent on the size of the effect that shall be detected. The expected R can be calculated as follows:

$$N = \frac{k}{R} + 1 \tag{29}$$

Thus for the three different effect sizes of R according to (Jacob Cohen, 1992) and 10 predictors the resulting minimum number of necessary samples is presented in Table 3-4.

Table 3-4: Sample numbers necessary to detected specific effect classes.

Expected *R*	Sample Size *N*
0.1 (weak effect)	101
0.3 (medium effect)	35
0.5 (strong effect)	21

The individual assessment of predictor variables is conducted using *t-tests* since *regression coefficients* close to zero can be interpreted as having no effect, therefore being equivalent to the null hypothesis of the *t-test*. The result of the test is computed by dividing the acquired regression coefficient b by the *standard error* of possible regression coefficients SE_b (by sampling the data multiple times and compute the resulting b values):

$$t = \frac{b}{SE_b} \tag{30}$$

After acquiring the degree of freedom ($df = N - k - 1$) the significance of the result can be computed, with 0.05 indicating that computed regression coefficient b is significantly different from zero (Field, 2009).

When incorporating the performances of feature detectors on different settings of the synthetic environment an additional *categorical* predictor needs to be added to the model. This is explained in appendix C.1 and in depth theoretical background can be found in (Field, 2009).

4 Implementation

This chapter describes the implementation of the applied concept presented previously. Specifically implementation aspects of following three components necessary to demonstrate the concept will be explained:

- Unmanned aerial system
- Synthetic environment
- Concept demonstrator

An unmanned aerial platform was deployed to capture natural imagery used to create the reference dataset *photo*. Then, the positional information of the recorded flight trajectory was fed into the synthetic environment to generate multiple synthetic datasets with varying parametrization. Lastly, the concept demonstrator to perform the comparison and evaluation steps explained in the concept chapter above was applied to the acquired datasets.

Before going into detail the procedure to achieve synchronized datasets depicted in Figure 4-1 needs to be presented.

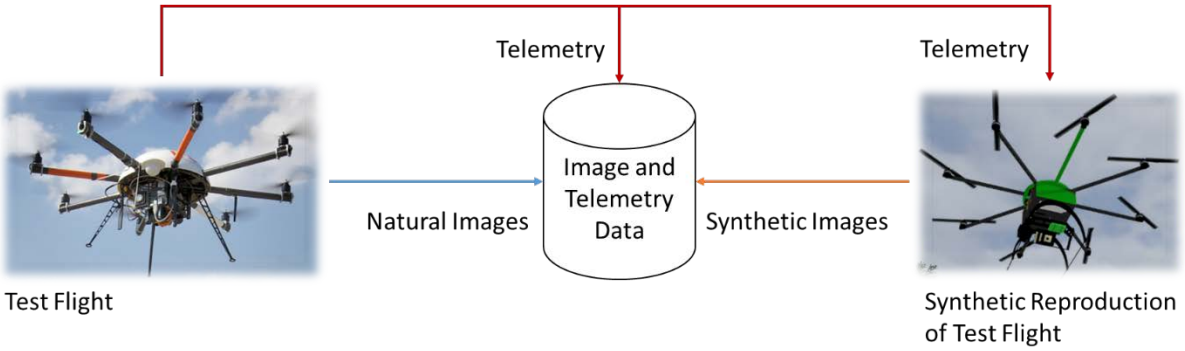


Figure 4-1: Dataset acquisition and generation approach

First, the test flight was conducted, where an aerial platform equipped with mission sensors (electro-optical camera and inertial measurement unit) flew above the target area and recorded natural images as well as telemetry data. This data contained all information necessary to monitor the current flight status of an aircraft (e.g. position, altitude, attitude and speed), the term itself origins from the wireless transmission necessary to deliver the data to the ground control station (monitoring and commanding system of the aircraft). It was provided among other sensors by an inertial measurement unit (AHRS) on-board the aircraft and was directly recorded on the aircraft. Telemetry data provided all data necessary to pinpoint the three

dimensional position and orientation of the aircraft at any time during the test flight, needed to reconstruct the actual viewpoint of the sensor for every recorded image During the test flight, telemetry and natural images were recorded together with their data acquisition timestamp, allowing the test flight to be replayed and used to feed the synthetic environment. The telemetry data was replayed to position the camera in the 3D virtual world. Each time a natural image has been captured a synthetic sensor image was recorded with the identical frame number.

4.1 Unmanned aerial system

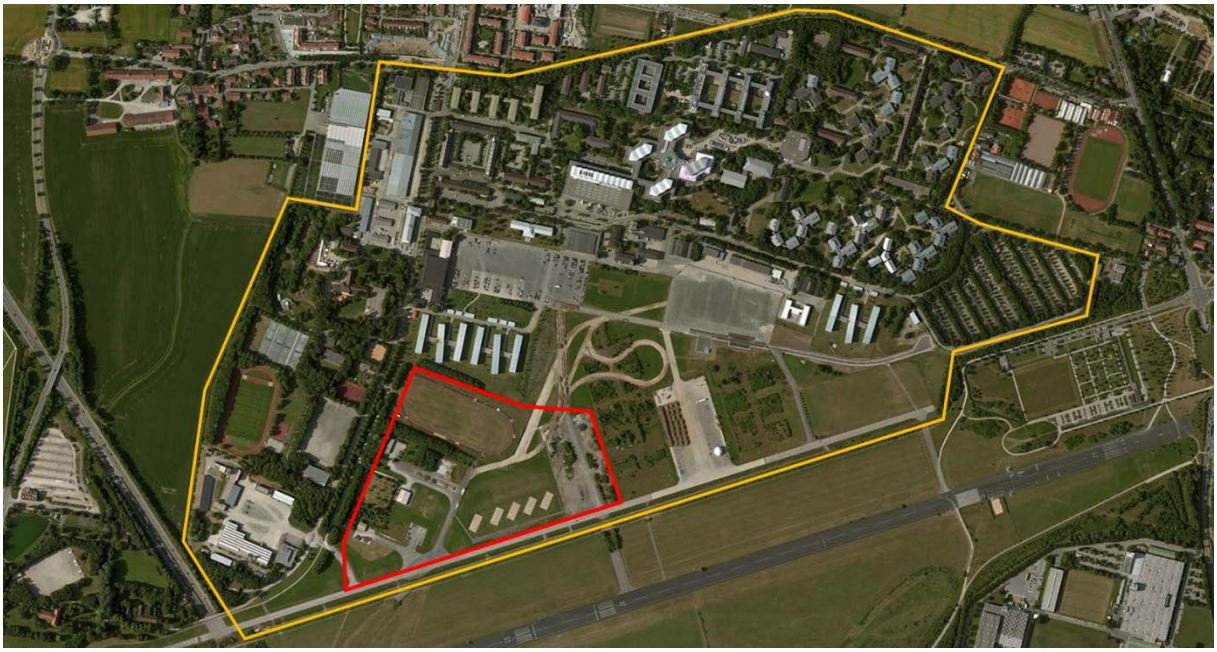


Figure 4-2: Campus of the University of the Bundeswehr Munich (yellow) and test flight area (red).

The natural image dataset was acquired by flying over the testing area and capture the necessary data. All aerial test flights conducted during this research were performed on the test flight area (highlighted in red in Figure 4-2) of the University of the Bundeswehr Munich (yellow). The area has been selected as it was easy to access, secured from unauthorized access, allowed small aircraft operation and had varying terrain surface (e.g. fields, forest, buildings, roads, etc.). To control the deployed unmanned aircraft a set of technical components was necessary. Usually grouped under the term *Unmanned Aerial System* (UAS), it is *" considered to be the system, whose components comprise the necessary equipment, network, and personnel to control an unmanned aircraft [vehicle] (UA[V])"* according to (Plöger, 2010) and comprises besides the airborne platform the control equipment on ground

(*Ground Control Station*, GCS) as well as the wireless telecommand and telemetry data link in between.

To capture the sensor image stream in sync with the current flight status (orientation, position, altitude of the aircraft) to geo-reference every image the experimental UAV *Okto XL* of the Institute of Flight Systems had been selected. This aircraft provided the necessary adaptability, while being compact, fulfilled all given requirements and was readily available to the institute. The aircraft is detailed further in chapter 4.1.1.

Figure 4-3: Left: The interior (left) and exterior (right) of the deployed ground control station. Note the antenna array located on the roof.

The UAV was controlled via the ground control station (GCS) depicted in Figure 4-3, consisting of four COTS 19” rack computers mounted into a box-type van hosting a human machine interface to control and monitor the unmanned aircraft. Connection to the UAV was established via a WLAN access point with multiple antennas for near field (omnidirectional) and more distant (two orientable directional antennas) WIFI reception. The antennas were automatically aligned towards the aircraft by computing the relative direction between location of the GCS and Aircraft (acquired from telemetry data). This, solution enabled the system to maintain WIFI connection throughout the test-area.

Further, as depicted in Figure 4-4 a safety pilot monitored the aircraft during the experiment to take control in case of unexpected aircraft behaviour via regular TC communication link.

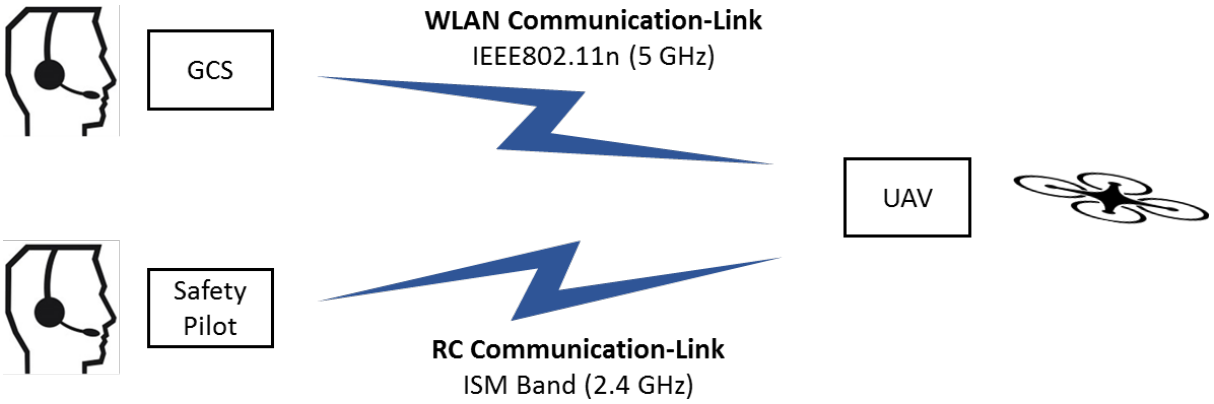


Figure 4-4: The Unmanned aerial system used to capture the natural image data.

In the following sections, hard- and software of the components used in this investigation are presented. A more detailed view about the general toolset, which is used to conduct flight experiments at the IFS is presented in (Schmitt, Rudnick, Stütz, & Schulte, 2015).

4.1.1 Unmanned aerial vehicle / sensor platform

Figure 4-5: The unmanned Aerial Vehicle *Octo XL* used in the experiments for this thesis. In this image, a fully equipped payload can be seen (Russ, Schmitt, Hellert, & Stuetz, 2013).

The *Octo XL* Multicopter has been developed as an experimental aircraft at the institute of flight systems. It is based on the assembly kit *MK Okto XL* from *HiSystems*[13]*;* however, the

[13] Website of manufacturer: http://www.mikrokopter.de/de/startseite [Last Accessed: 2016-01-11]

airframe has been specifically redesigned (Figure 4-5). The aircraft has been used previously in other experiments as an airborne sensor platform (Russ et al., 2013) and proved to be a stable platform also for this investigation.

The UAV is equipped with eight brushless electric motors MK3638 each generating a maximum thrust of 2.2 kg and consuming up to 350W. The motors drive eight 13” propellers. The necessary power is supplied by four Lithium-Polymer (LiPo) accumulators each providing 7Ah with a maximum voltage of 7.4V. The Multicopter has about a diameter of 1m and a height of 40cm. The MTOW is 6kg, allowing the payload to weigh up to 2.2kg. The aircraft can achieve speeds up to 60km/h and yields a flight endurance of about 15 minutes when fully equipped (MTOW).

A custom payload bay had been designed for the *Octo XL Multicopter* (Rönnfeldt, 2013). This experimental mission payload system depicted in Figure 4-6 has an adjustable centre plate that can be positioned as necessary depending on the spatial requirements payload.

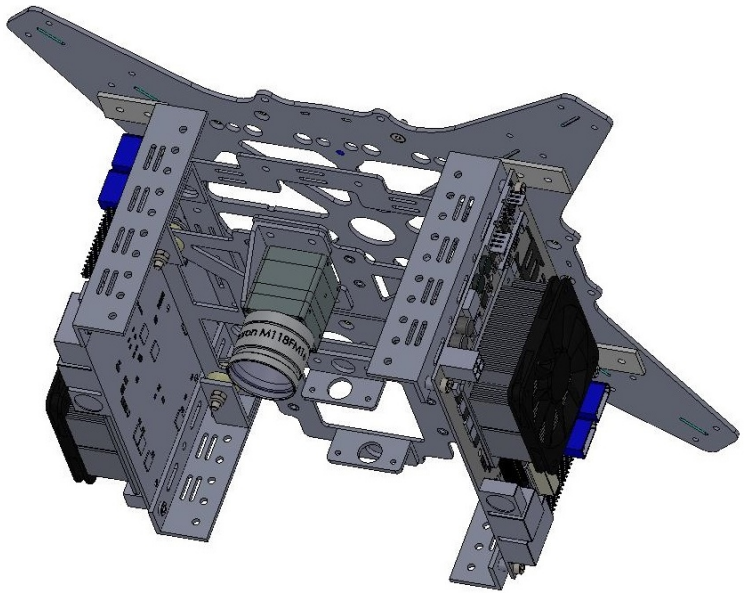


Figure 4-6: Experimental mission payload system. In this configuration, it is equipped with one camera and two processing boards.

The outside of the payload system has been designed to fit two 3.5”embedded mainboards providing the UAV the computational power to perform on-board data processing. The deployed configuration consisted of two processing boards, one image sensor and one attitude and heading reference system (AHRS). The installed COMMELL LS-37B processing boards are equipped with 9GB of RAM and i7-3840QM processors (2.8 GHz). As storage solid-state disks with capacities of 256GB and 480GB. As wireless interface, the SR-71 adapter from

Ubiquiti supporting the 802.11n standard in 2x2 MiMo antenna configuration is deployed and configured to 5GHz. The boards are connect with each other via wired Ethernet. The MQ042CG-CM from XIMEA has been chosen as camera system. This camera provides images with a resolution of 2048 by 2048 pixel at a maximum framerate of 90Hz via USB 3.0 to one of the processing boards. It was selected due to its performance at the provided form factor (2.6cm x 2.6cm x 3cm), weight of 32g and the possibility to directly access the camera via the provided C++ API.

For this investigation, a Myutron HS2514V lens with a focal length of 25mm (equal to 26.9° FOV) and manually configurable focus and aperture was attached to the camera. A MTi-G 700 from XSENS was used as GPS enhanced AHRS, providing position (usually 4-6Hz) and velocity to the 3D orientation at the sampling rate of the Kalman filter (configured to 400Hz) as listed in Table 4-1. This, sensor delivered the necessary telemetry data for every recorded image. For roll and pitch axis, the dynamic error of 1σ RMS is 0.3°, for yaw it is 1.0°. The positional accuracy has standard deviation of 1.0m in horizontal direction and 2.0m in vertical direction. The MTi-G 700 was interfaced through its USB 2.0 interface using a C++ API.

Table 4-1: Accuracy of AHRS according to the manufacturer (Xsens Technologies B.V., 2014)

Sensor Properties		
Telemetry Parameter	Accuracy	Update Rate
Position horizontal, AHRS, SBAS	1.0m STD	4 Hz
Position vertical , AHRS, SBAS	1.0m STD	4 Hz
Altitude (AGL) , AHRS, SBAS	2.0m STD	4 Hz
Yaw (Euler), typical, 1σ RMS	1.0°	400 Hz
Pitch (Euler), typical, 1σ RMS (static)	0.3°	400 Hz
Bank (Euler), typical, 1σ RMS (static)	0.3°	400 Hz

In Figure 4-7, the configured experimental mission payload system is shown. Below the red camera mount, camera and Myutron lens are visible. The camera system is mounted decentralized on the centre plate to provide vision unobscured from the landing gear. On this centre plate (hidden behind the blue cables), the AHRS is attached. The centre plate has been lowered to distance the AHRS from the magnetic field of the power controllers cooling fan. Both sensors (camera and AHRS) have been connected to the processing boards, which are also connected to the navigation control of the UAV to allow remote route transmission.

Figure 4-7: Experimental mission payload system as used for the test flights of this thesis.

4.1.2 Software implementation

All software components of the UAS have been implemented in C++ using the *Qt* application framework and *boost* C++ libraries as an extension to the basic functions provided by standard libraries. The computing hardware of UAV and GCS are using Ubuntu 14.04 LTS (64 bit) as operating system. All SW applications communicate using the *AnyCom* inter-process communication toolkit presented in chapter 0.

The Flowchart diagram in Figure 4-8 according to DIN 66001 details the functional steps of the four necessary logical components to record the natural image datasets. These components are *ground control station* (GCS), *UAV flight management system* (FMS), *UAV Sensors* and *UAV Recording*.

The *GCS* provides all necessary means to control the aircraft and its payload. The *UAV FMS* wraps the system specific communication interface of the autopilot to provide simplified means to communicate with the flight controls of the UAV (Clauss & Schulte, 2014). Additionally, it connects the flight controls to the inter-process communication. The *UAV Sensors* module similarly connects the specific interfaces of the sensors to the inter-process communication and provides sensor data and sensor control capabilities. Further, the *UAV Recording* module records the information of interest (image and telemetry data) in their original sample-rate to file.

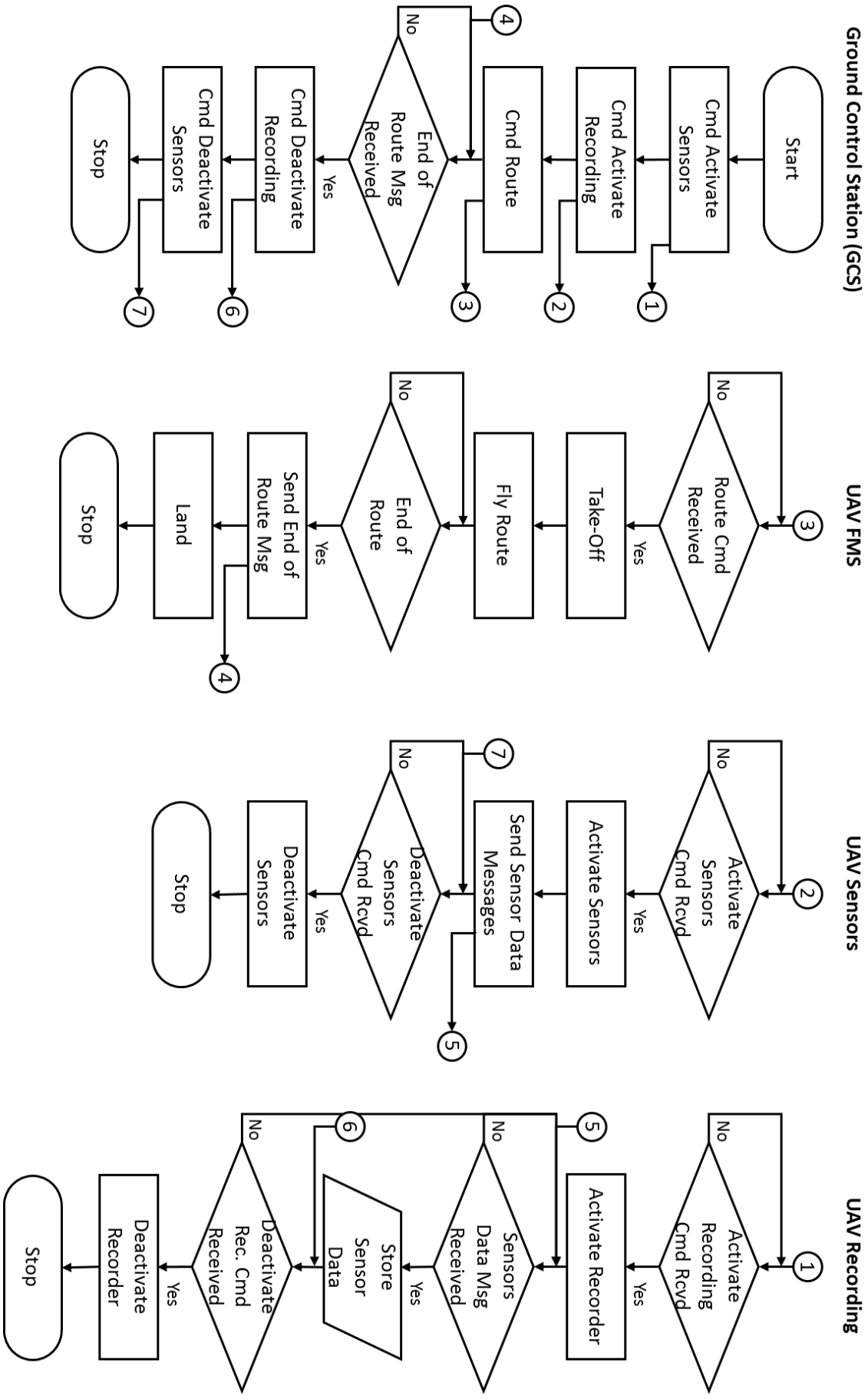


Figure 4-8: Flowchart diagram of the natural data capture system implementation.

As the flowchart in Figure 4-8 depicts, the ground control stations is the main control module triggering the start of the experiment and forwarding the planned route of flight to the FMS and activates the sensors and data recording. Thus, sensors provide their data to the inter-process communication framework where recording is subscribed to image and telemetry data, saving it to file. This continues until the end of the commanded route has been reached. When the *end of route* signal is provided to the *GCS*, it triggers the deactivation commands for the sensor and recording modules. The landing of the aircraft then concludes the experiment and the data acquisition.

The information flow and the distribution of processes between UAV and GCS are depicted in Figure 4-9. All arrows indicate communication between processes. The arrows with description indicate inter-process communication while blank arrows indicate specific proprietary communication interfaces.

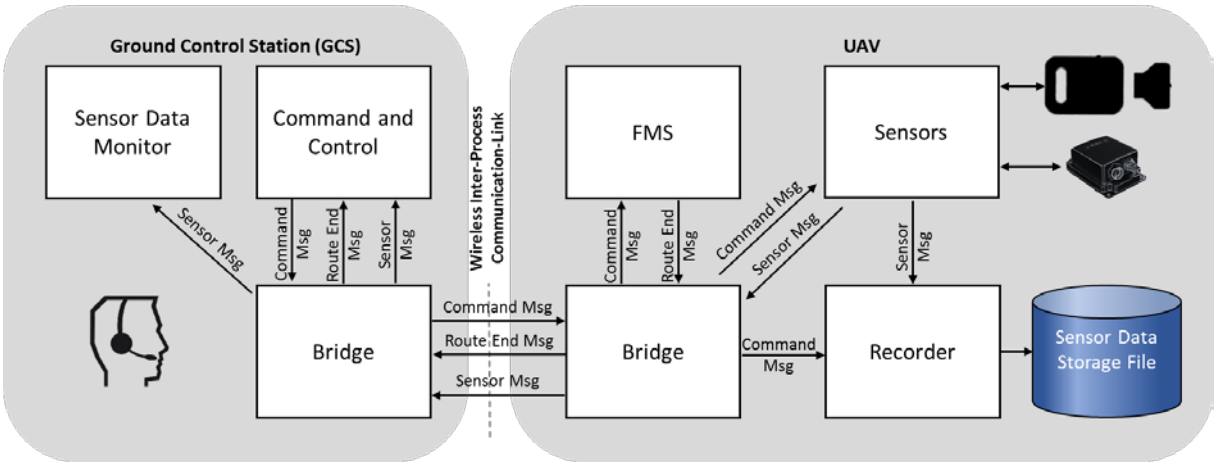


Figure 4-9: Distribution and communication of processes among the physical components. All labelled connections are inter-process communications. Others are communicated via C++ interfaces.

After the experiment, raw data images and telemetry were fused on ground to provide sensor and telemetry in one data container to ensure synchronisation and simplify replay and handling. Therefore, each image (10Hz) was combined with the last received telemetry data package (400Hz) and saved into a stream container format. Thus, the image was provided with telemetry data with a maximum time difference error of 2.5ms.

4.2 Synthetic environment

To prepare the experimental setup, first the synthetic environment used to produce synthetic (or computer graphic) imagery had to be selected. Before listing the requirements for this selection a short background is given that helps to curtail available choices. The Institute of Flight Systems investigates new concepts of guidance and automation for unmanned aerial vehicles. These concepts contain also paradigms for sensor guidance, sensor management and perception management where automation should adapt during the mission to the current tactical situation and environmental conditions. These paradigms include automatic image processing of mission sensor information and semantic data extraction of the depicted scene. However, due to the complexity of these investigations on *mission level*, staging complex operational conditions in real live environments becomes hardly achievable and regularly include experiments conducted in synthetic simulation environments including the simulation of coherent sensor data[14]. Specifically the presence of human-in-the-loop elements in such investigations bring along the necessity of (soft) real-time performance. This requirement a-priori removes high quality rendering engines such as the Octane[15] or Indigo Renderer[16] based on global-illumination (e.g. path-tracing) as a possibility.

Another possibility is the use of low-level graphic toolkits such as OpenSceneGraph[17] (OSG). While OSG is versatile and allows direct access to the rendering pipeline, this low-level access also increases the efforts to create a synthetic simulation environment for geo-referenced test flights and the main benefit of reducing the costs by simulation is diminished or even lost. Therefore, commercial of the shelf (COTS) products providing the complete toolchain for simulation, modelling of terrain, scenario staging and a model database are of greater interest and to be considered in the following. During the initial phase of this work three products where considered:

[14] More about the research conducted at the Institute of Flight Systems can be found on https://www.unibw.de/lrt13 [Last Accessed: 04.05.2016]

[15] According to the manufacturer, Octane is an unbiased, physically based renderer for photo-realistic results: https://home.otoy.com/render/octane-render/ [Last Accessed: 29.12.2015]

[16] According to the manufacturer, Indigo Renderer is an unbiased, physically based and photo-realistic renderer: http://indigorenderer.com/ [Last Accessed: 29.12.2015]

[17] According to the manufacturer, OpenSceneGraph is an open-source real-time graphics toolkit based on C++. http://www.openscenegraph.org/ [Last Accessed: 29.12.2015]

- **CryEngine 2 SDK** (*Crytek*) provides the highest near-ground image-quality of the three selected engines and includes a physics engine, AI, Scenario Scripting, a C++ interface and a sandbox editor. 3D-models have to be created using external tools (e.g. *3DS Max*) and are afterwards converted using the *CryExporter*. A medium amount of objects (from the Game *Crysis*) is already provided with the SDK. The game engine has been designed and optimized for ground entities leading to drawbacks when simulating aerial vehicles (e.g. small terrain size, no geo-referencing, import of GIS data, etc.)
- The **Modelling & Simulation Toolchain** from *Presagis* is targeted at developers for professional training simulators and covers most tools necessary for their development. For instance, the rendering engine *Vega Prime* allows considerable terrain sizes and high view distances but also provided less detailed terrain surfaces compared to its competitors at the time of selection. For scenario scripting and simulation multiple tools are available. An AI is also provided via *AI.Implant*. The necessary tools to create new 3D-models or terrain databases are also available. Unfortunately, the complete toolkit is quite expensive and the provided 3D-model database is small. Needed 3D-models can either be modelled or bought additionally increasing the financial effort.
- **Virtual Battlespace 3 (VBS3)** from Bohemia Simulations is a modified game engine (from ARMA) and recently gained popularity in the professional simulation market due to its holistic toolchain, modern computer graphics and attractive pricing. The provided complete toolchain consisting of rendering engine from, scripting language, dynamic simulation models, AI, large object database and tools to generate new content. The graphical quality provided can be ranked between CryEngine 2 and Presagis' Vega Prime. The high level of integration between renderer, scenario Editor and simulation allows quick scenario generation with the drawback that access to single components is highly limited. For external interaction with the engine, a C++ interface exists.

Selection

Out of these three COTS software kits, *VBS3* has been selected as rendering engine for the experiments (see Table 4-2). The main points where the possibility to create geo-referenced terrain and quickly generate small scenarios while limiting the necessary financial efforts.

Additionally, the results of this thesis may help a larger community when using VBS3 due to current wide spread use in the industry. Further, the provided large repository of 3D-models and content development tools provided all means to generate the terrain database (more in chapter 4.2.2). Lastly, the modern computer graphics imagery also tuned for aerial vehicles confirmed the decision to use *VBS3*.

Table 4-2: Comparison between VBS3, CryEngine 2 and Presagis' Modelling and Simulation Toolkit. A higher value means better performance (5 = highest value, 1 = lowest value, 0 = not available).

Criteria	CryEngine 2	VBS3	Presagis M&S Toolkit
Graphical Quality	5	3	1
Model DB Size	2	5	0
Terrain Size	2	4	5
Visual Quality of Terrain	4	3	2
Geo-referenced Terrain creation	0	4	5
Geo-referenced Scenario creation	0	5	5
Price	1	5	3
Resulting Scores:	2,00	4,14	3,00

4.2.1 Software implementation

The synthetic environment was used to generate synthetic image datasets in sync to previously acquired natural datasets. The flowchart in Figure 4-10 details the functional steps necessary to achieve this goal. Processing blocks with by parallel lines mark features that are more deeply explained later on.

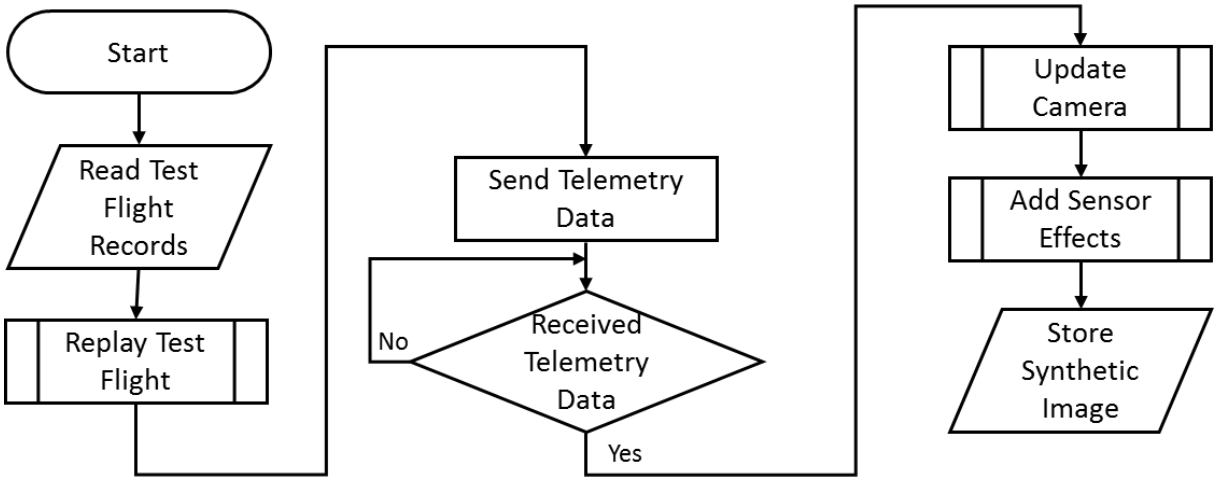


Figure 4-10: Flowchart diagram of the synthetic dataset generation implementation.

To acquire synchronous synthetic and natural datasets, the identical position and pose of the camera needed to be extracted from the actual test flight. A replay mechanism pushed telemetry data frame-by-frame to the synthetic environment. Newly received data was then used to update the position and orientation of the virtual camera. The rendered image was then copied and forwarded to the mission sensor simulation to add sensor effects (e.g. noise or distortion). Finally the image was stored on the hard drive and labelled with the same frame number as the corresponding natural image. The following paragraphs now detail the implementation of *Replay Test Flight*, *Update Camera* and *Add Sensor Effects* (Mission Sensor Simulation).

The replay was conducted by using the network based group communication middleware *AnyCom* Toolkit developed at the Institute of Flight Systems for inter-process communication. The toolkit concept is based on the data distribution service (DDS) standard defined by the *Object Management Group* (OMG) and is detailed in (F Boehm & Schulte, 2012), (Böhm & Schulte, 2012) and (Florian Boehm & Schulte, 2013). The Toolkit provides tools for real-time recording, monitoring and replay of sensor data. Thus, telemetry and sensor data were transmitted via *AnyCom* to the virtual environment. The synthetic environment *VBS3.4* was connected to *AnyCom* via an *Telemetry Plugin* as depicted in Figure 4-11.

The *Telemetry Plugin* (using the *Application Scripting Interface* (ASI) C++ API) checked for new telemetry data prior to every rendering (40-50 fps or Hz), which was sent from the replay application at 10Hz. When new data was received, the camera was oriented and positioned.

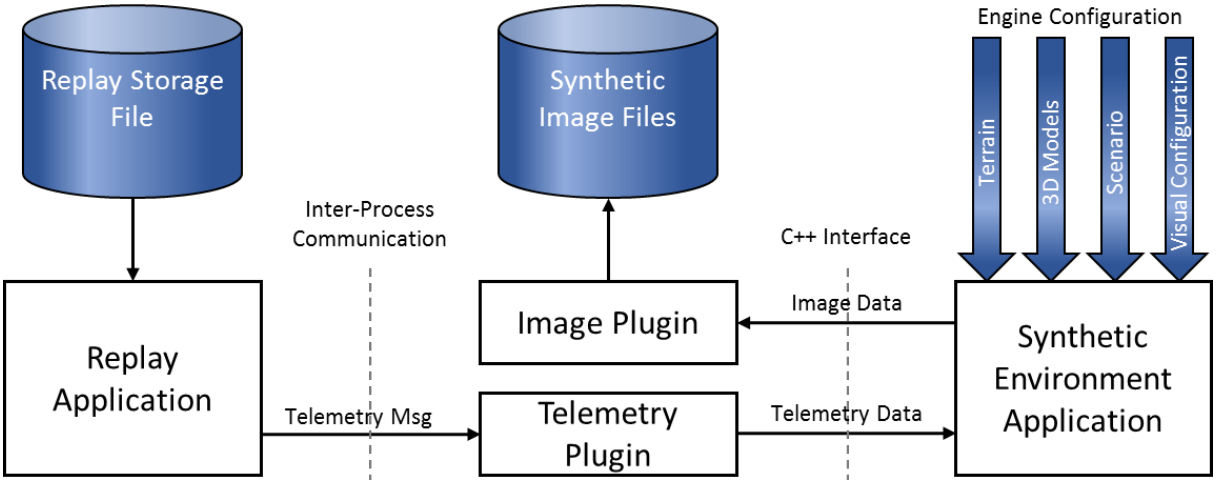


Figure 4-11: SW-setup used to generate synthetic datasets. The plugins are used to position the camera and store rendered image data. All other necessary configurations are performed from within the engine.

When a telemetry message has been received and after the camera is positioned the *Image Plugin* is executed. Here, a modification of the *Integrated Test bed for Experimentation on Mission Sensors* (ITEM) architecture (Hummel & Stütz, 2011) using *VBSFusion C++ API*[18] from SimCentric Technologies was used to save rendered imagery as depicted in Figure 4-12. In ITEM the *Virtual Sensor Layer* accesses the rendered image right before the buffer swap and forwards this image data to a MSID (multiple instruction, single data) architecture (Downton & Crookes, 1998) based *Filter Module*. This module handles post-processing filters and data distribution. In this work, a *stochastic noise filter* and a *lens distortion filter* based on the distortion model of (Z. Zhang, 1999) are deployed. The *noise filter* allows simulating temporal and spatial noise on all three colour channels. The *distortion filter* applies the same radial lens distortion to the rendered images as measured during the calibration process of the deployed UAV camera. Afterwards, the result is stored using the *Communication Module*.

The visual configuration was accessed within VBS3 via GUI, since the *one-factor-at-a-time* experimental setup only needed one parameter changed between the creating of different datasets. The scenario was created with the VBS *Offline Mission Editor* (OME), which provides a top-down view on the loaded terrain and allows adding of objects and definition of their interactions. Complex scenario elements were scripted via the provided scripting language. The most complex components in test flight recreation were the design of the terrain database, the design and placement of 3D-buildings as well as their surrounding vegetation in the terrain map. This step is detailed in the next section.

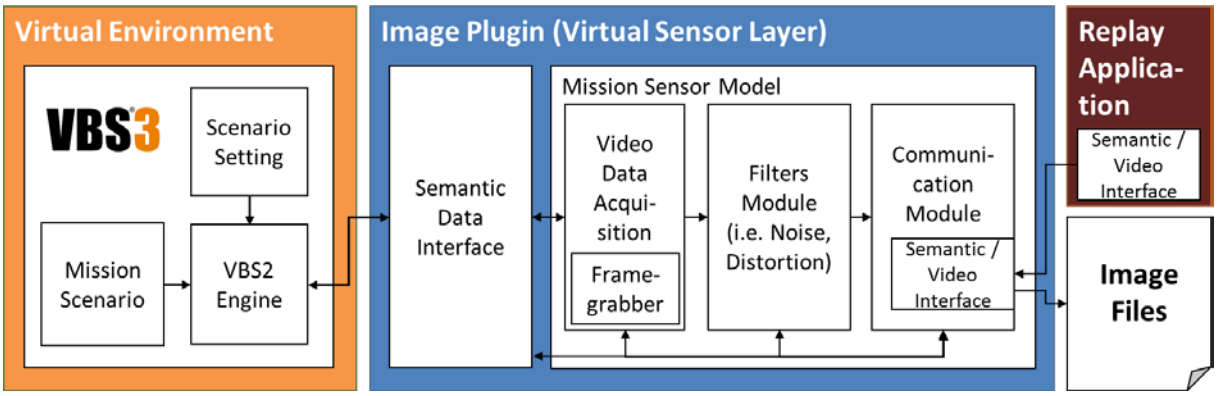


Figure 4-12: Modified ITEM Architecture adapted and streamlined to the needs of the synthetic image generation.

[18] Fusion is a C++ API for VBS developed by SimCentric Technologies https://www.simct.com/?page_id=712 [Last Accessed: 05.01.2016]

4.2.2 Generation of terrain database

To simulate the scene captured in the natural dataset the actual terrain environment of the natural test flight (depicted in Figure 4-2) had to be reproduced for the synthetic environment in a geo-referenced way. Since the terrain database had been mainly but not exclusively generated for this investigation all major buildings of the university had been modelled as well. While for roads and trees existing models provided by the model database of the synthetic environment were used, each building of the university was manually designed. When all desired objects have been placed in the *terrain map*, the result was compiled to create the *terrain database* (see Figure 4-13). This database then directly was loaded in the synthetic environment for rendering. Simplified spoken, the *terrain map* is a coloured ground texture mapped onto a 3D elevation grid; a more detailed explanation follows in the next chapter.

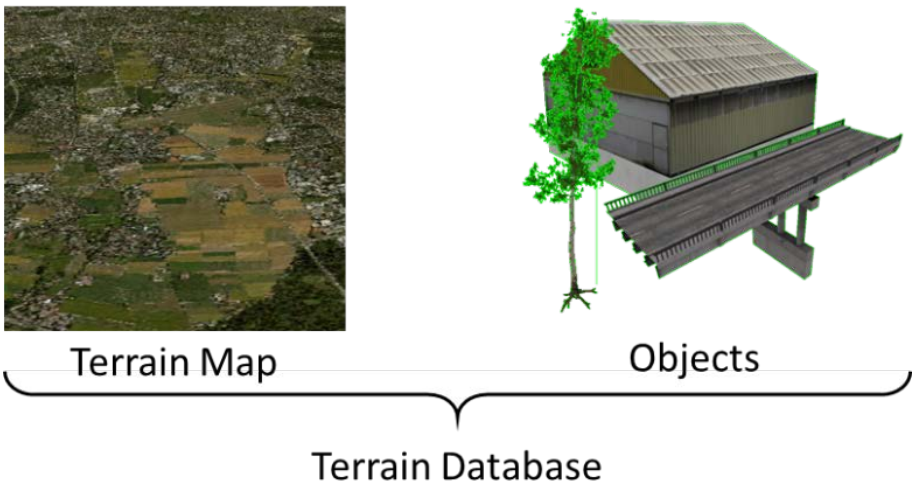


Figure 4-13: Terrain database consisting of terrain map and positional references to 3D-objects.

Terrain map

In general, a terrain map for 3D rendering applications comprises a height mesh with mapped textures. For this work, three types of geo-referenced terrain data had been accessed to generate the terrain maps for VBS3 (see Figure 4-14):

- **Rasterized aerial images** are oblique, orthographic aerial images made by aircrafts or satellites. This investigation uses aerial imagery of the *Bavarian office of Land-Surveying and Geo-Information* provided by the *Bundeswehr Geoinformation Service.* Since, the aerial imagery was taken in summer 2014 with a resolution of 0.2 meters

per pixel (mpp) some infrastructural modifications in the test flight area had been conducted until the actual date of the test flights.

- **Elevation data** describes the altitude of geolocations at a specific area and is captured via aircrafts or satellites with stereo-cameras, light detection and ranging (LIDAR) or synthetic aperture radar (SAR) sensors. The resulting point cloud typically is geo-referenced and converted to a rasterized *digital elevation model* (DEM) image, where values represent the altitude above mean sea level (MSL). Digital elevation models are categorized into *digitized surface model* (DSM) or *digitized terrain model* (DTM). While the DSM provides the measured altitudes of the surface including man-made structures, the DTM has been filtered for these structures to provide only the bare ground altitudes. Since buildings are modelled separately for this investigation a DTM was used. Again, the data used has been provided by the *Bundeswehr Geoinformation Service*. The resolution of the rasterized DTM is 15mpp, which is sufficient due to the limitations of the virtual engine and the general flatness of the modelled region.

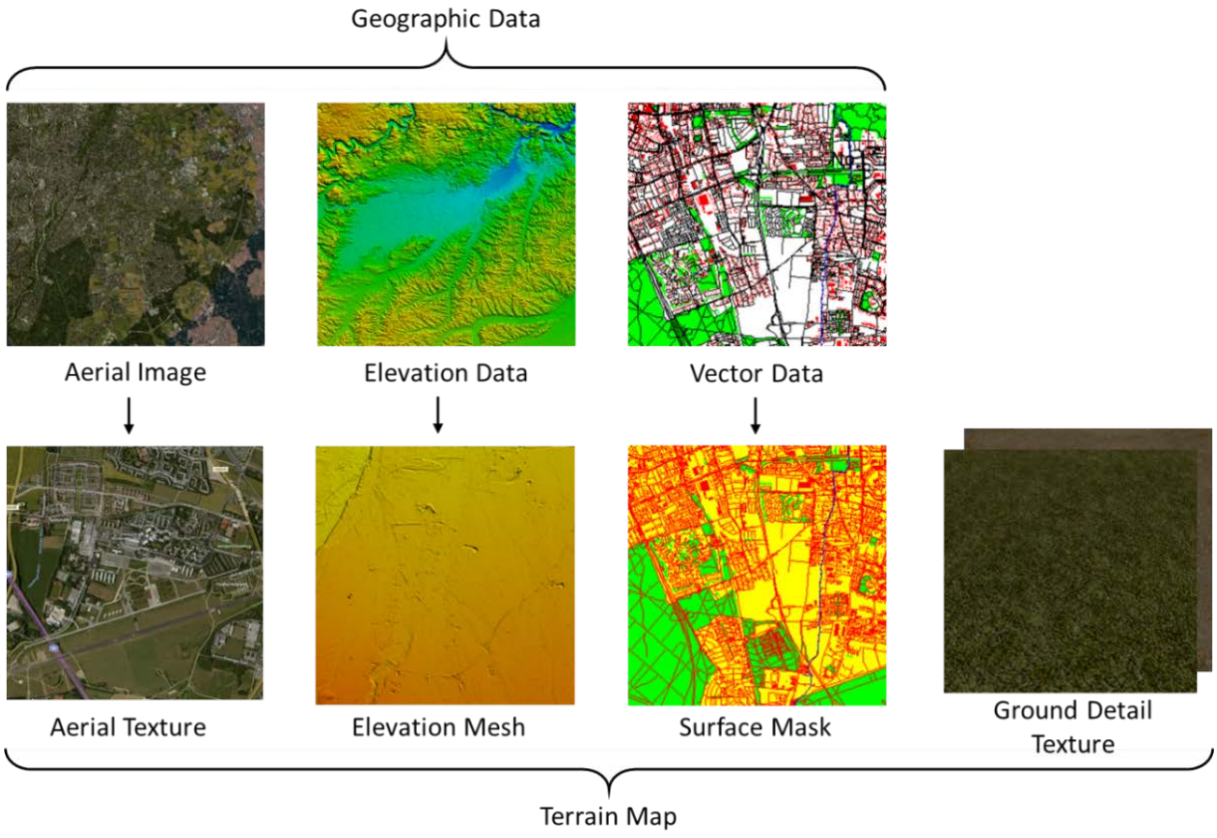


Figure 4-14: Source data used to generate the terrain database for VBS3 and their engine compatible conversions.

- **Vector Data** contain geo-spatial descriptions of geographical features (e.g. rivers, roads, lakes, etc.) in vectorised form (e.g. points, lines, shapes). Common formats are *ESRI Shapefile*[19] or *Simple Features* standardized by the *Open Geospatial Consortium* (OGC) in ISO 19125[20]. The data used for this work was acquired from OpenStreetMap, a project providing open-source geo-data[21]. Here, geographical features are categorized into nat*ural, land use, waterways, buildings, railways, roads* and are provided with a single *Shapefile* for each class.

The process and related toolchain to create VBS3 terrain maps is pictured in Figure 4-15. *Global mapper* is used to convert the provided geographic data into formats supported by landscape development tool *Visitor4*. In this step, the actual geographical location and size of the map are defined. The aerial image and elevation data are transformed to a fitting resolution. A surface mask is generated using vector data, which enables the blending of satellite image texture used as ground texture with detail textures of the grounds surface (e.g. asphalt for roads). This technique called *Layered Terrain Surface Representation*[22] (introduced by Bohemia Interactive 2006) raises the visual detail of the landscape in close ground camera views, independently (Roupé & Johansson, 2009) present a very similar method. The actual texture seen in the engine is the ground detail texture blended with the satellite texture. With the advent of VBS3, the blending method has been reworked. Now, at a distance of 50 meters the defined detail texture is (as the distance increases) slow replaced by a generic detail texture, which completely replaces the user specified detail textures at 100 meters distance from the ground. This boosts the image quality of ground views between 100 and 300 meters since the generic detail texture is of higher scale and reduces the tile-like repetitive pattern (which appears when small texture patches are repeated).

[19] The data format ESRI Shapefile is described in the Whitepaper of 1998: http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf [Last Accessed: 03.01.2016]

[20] Simple Featues is a data format after the standard ISO 19125 of 2004: http://www.iso.org/iso/catalogue_detail.htm?csnumber=40114 [Last Accessed: 03.01.2016]

[21] The download area of OpenStreetMap to acquire the shapefiles fo Upper Bavaria: http://download.geofabrik.de/europe/germany/bayern/oberbayern.html [Last Accessed: 06.01.2016]

[22] The Description of the Layered Terrain Surface Representation: *https://community.bistudio.com/wiki/Layered_Terrain_Surface_Representation* [Last Accessed 07.01.2015]

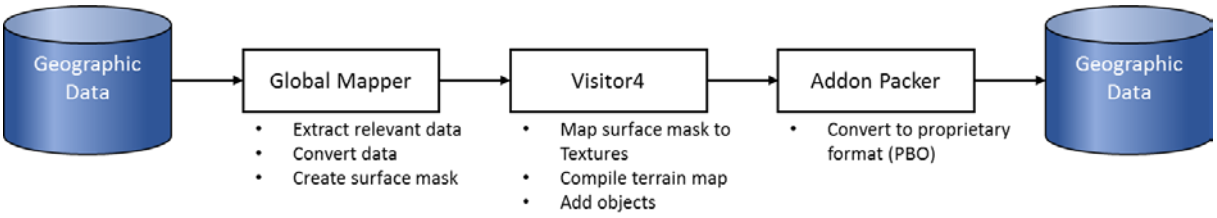


Figure 4-15: Toolchain to create VBS terrain databases.

In the surface mask, areas with man-made structures (e.g. roads, buildings) are coloured in red, rivers and lakes in blue, forests in green while default is yellow. The result is then converted to a rasterized image with the same size and resolution as the aerial texture. In *Visitor4*, the colours are then mapped to the respective textures (Figure 4-16).

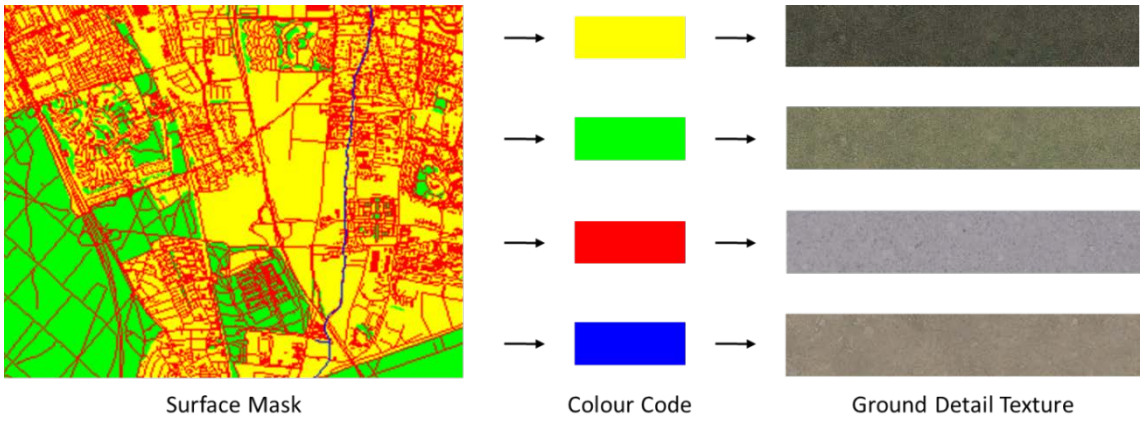


Figure 4-16: The Surface mask is generated based on GIS vector data. Additionally, colour mapping and detail textures are presented.

After all data was inputted into *Visitor 4*, the geographical details as well as terrain and satellite data resolution was set and the surface texture to colour mapping has been performed; the terrain was built and converted into a *Packed Bohemia Object* (PBO). This file was then added to VBS for rendering.

Figure 4-17: A synthetic view of a scene displayed for each different terrain database.

This step in the modelling process determines the quality of the satellite and ground detail texture. To analyse design variants of synthetic image quality in respect to performance of computer vision algorithms, three different terrain maps with different qualities were modelled. Figure 4-17 highlights differences of these terrain databases. The terrain database *LQ (Low Quality)* and *HQ (High Quality)* had been modelled with different resolution for satellite image texture and respective surface mask (see Table 4-3). In database *HQ_NB*, 3D-objects have been omitted to additionally identify their influence. The optional placement of these objects is addressed in section 0.

Table 4-3: Configuration of the three terrain databases produced.

Terrain Database	Resolution Satellite Images	Resolution Digital Terrain Model	Resolution Surface Mask	Objects
Low Quality (LQ)	5 mpp	15 mpp	5 mpp	Yes
High Quality (HQ)	0.2 mpp	15 mpp	0.2 mpp	Yes
High Quality no Buildings (HQ_NB)	0.2 mpp	15 mpp	0.2 mpp	No

3D-objects

Up to now, the terrain database comprises just the terrain contour colorized by several layers of textures. However, scenes in the natural world rarely can be reduced to such simplified view (e.g. deserts). Therefore, terrain databases typically are “populated” with various types of static objects. Concerning the test area to be modelled, such objects are mainly roads, fences, vegetation and buildings.

Not only to reduce the effort necessary to create the terrain model but also to emulate the classical industry approach for database modelling all objects except for buildings were taken from the VBS model library. All these object types there are specified as *geo-typical* that means being typical for a certain geographical region to be modelled. For example, types of trees that appear in the middle European region are used even though they are only similar but not identical to the specimens found in the test area. Due to their size and recognize-ability, buildings are modelled in *geo-specific* quality. This indicates that the models of buildings have same dimensions and locations (with an accuracy of 0.5 meters) as their real counterparts and photographs had been used to texture their surfaces when possible. The information necessary to create a *geo-specific* model of a building is depicted in Figure 4-18. The building’s dimensions were read from its blueprints. Since photographing rooftops was

difficult, the type of roof and its form have been visually identified and then modelled using freely available textures from texture databases[23].

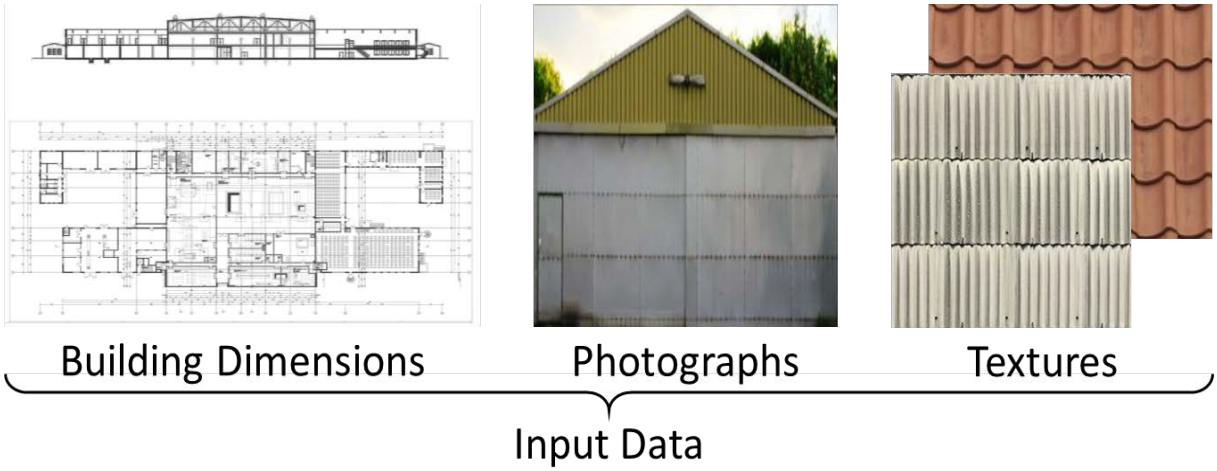


Figure 4-18: Information necessary to create the exterior of a geo-referenced Building.

In VBS3, the relatively simple *Phong local illumination model* (Phong, 1975) is used (see appendix A). To enhance the visual quality of buildings, VBS3 allows the deployment of multi-layered shaders. In this work, the *NormalMapSpecularMap*[24] *shader* employing *normal mapping* and *specular mapping* was used. *Normal mapping* preserves the appearance of an object even though the polygon mesh is of low detail (Jonathan Cohen, Olano, & Manocha, 1998). Similarly, the specular map as implemented in VBS3[25] preserves the local reflection information of an object and its reflective power (an example is shown in Figure 4-19).

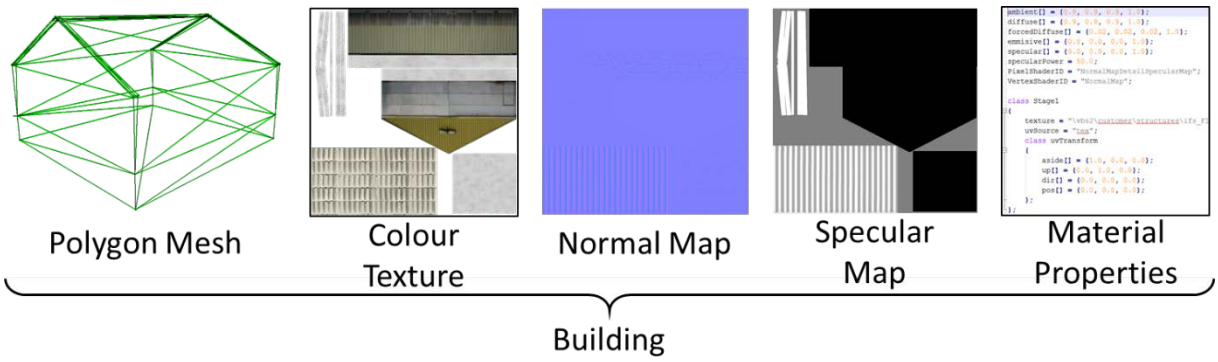


Figure 4-19 The components of a 3D model as used in this work to model geo-specific buildings.

[23] Two Example free texture databases are http://texturelib.com/ [Last Accessed 07.01.2016] or http://www.textures.com/ [Last Accessed 07.01.2016].

[24] More detail on VBS shaders can be found at https://resources.bisimulations.com/w/index.php?title=RVMAT [Last Accessed: 08.01.2016]

[25] Documentation of Normal Maps and Specular Maps in VBS: https://resources.bisimulations.com/wiki/HQ_Normal_Maps [Last Accessed: 08.01.2016]

Object placement

Eventually the terrain map had to be populated with 3D-objects to generate a 3D representation of the test area in the synthetic environment. Figure 4-20 presents the objects placed into the terrain database.

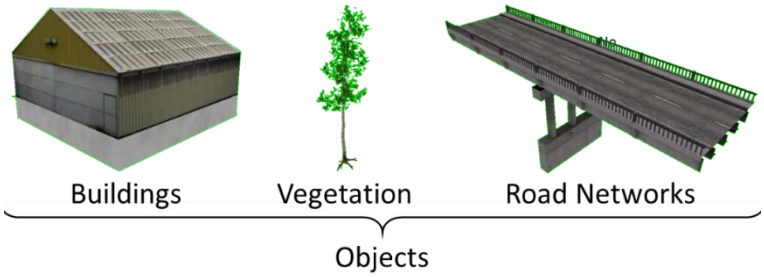


Figure 4-20: Objects placed in the terrain map.

The placement of objects on the terrain map was performed with the tool VB-Edit from Eurosimtec[26]. The resulting terrain database *HQ* is depicted in Figure 4-21. In the lower left corner of the image the test flight aerial can be seen. A direct comparison between an aerial image of the University of the Bundeswehr and the same scene depicted in the synthetic environment using database *HQ* is presented in Figure 4-22.

Figure 4-21: Terrain database HQ viewed from within the synthetic engine.

[26] Product Webiste of VB-Edit: http://www.eurosimtec.de/products/vb-edit/ [Last Accessed: 08.01.2016]

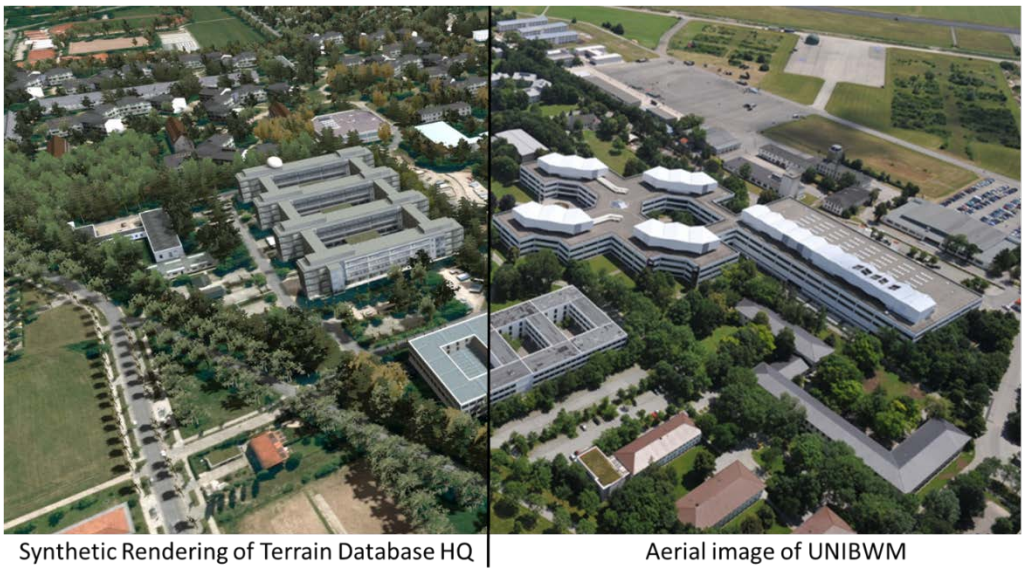


Figure 4-22: University of the Bundeswehr Munich (UNIBWM) depicted as aerial photograph (right) and as synthetic rendering using terrain database *HQ* from same viewing position and angle (left).

4.3 Concept demonstrator

In this chapter, the software implementation of the concept demonstrator discussed in chapter 3.2 is presented. The logical software architecture of the concept demonstrator (see Figure 3-3) is again described as flowchart in Figure 4-23. The three presented threads were executed independently. Executed initially, the *Feature Detector Performance Evaluation* read an image pair from a dataset to detect SIFT, SURF and MSER features. These resulting features were compared to the ground truth to measure the detectors performance. Computed performance results were then saved to the hard drive. The process iterated until the last image pair was encountered. The next thread, the *Image Content Distance Computation* also read images pairwise from the dataset and then computed image descriptors for each image. These descriptors were then used to compute the image pair distances and results were saved. Lastly, the *Influence Factor Analysis* loaded the results gained in performance evaluation and the distance results to compute regression models identifying image distances the feature detectors are sensitive to.

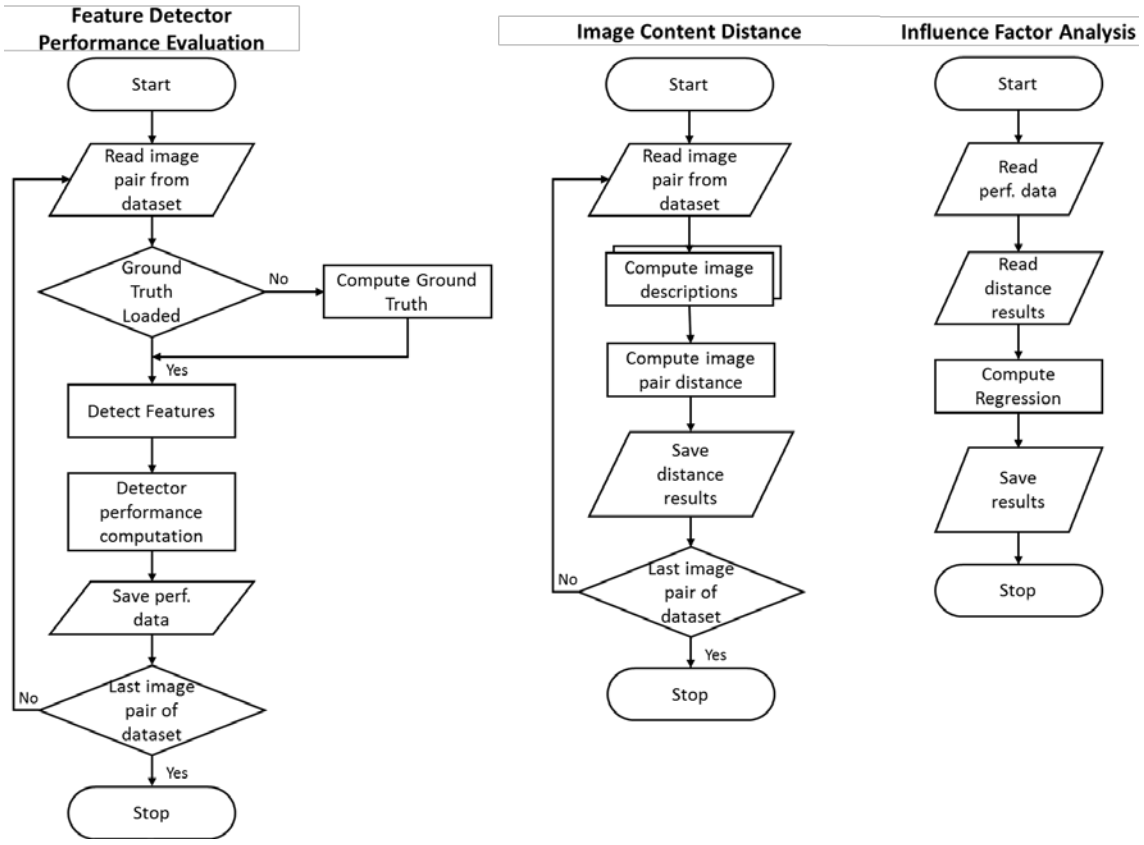


Figure 4-23: Flowchart diagrams of the concept demonstrators three components.

The ground truth generation of the *feature detector performance evaluation* (presented in chapter 3.4) was implemented in C++ and used the SURF, brute force matching and RANSAC implementation of OpenCV[27]. The *feature detection performance evaluation* itself was implemented in MATLAB and based on the framework *VLBenchmarks* (Lenc, Gulshan, & Vedaldi, 2012), which implemented the evaluation concept of (Mikolajczyk & Schmid, 2005; Mikolajczyk et al., 2005). This framework computes the *absolute* and *relative repeatability* of feature detectors on datasets used in (Mikolajczyk & Schmid, 2005). The framework was modified to fit the concept of this investigation. *VLBenchmarks* provides several feature detectors out-of-the-box. Thus, the implementation of SIFT and MSER as provided by the framework were deployed. As third feature detector, SURF had been added to the framework using the implementation provided by MATLAB.

The *image content distance computation* has been separated into image descriptors (see chapter 3.5) and distance measures (see chapter 3.6). Both utilized the C++ *mpeg7FexLib* library of (Bastan, Cam, Gudukbay, & Ulusoy, 2010) providing visual descriptors of the

[27] An open-source computer vision library. http://opencv.org/ [Last Accessed: 10.05.2016]

MPEG7 experimental model (XM) (Motion Picture Expert Group, 2003; Yamada et al., 2001) in a simple object oriented architecture. In a separate step, the descriptions of the added image quality descriptors were computed using MATLAB. For SSIM and MSE / PSNR the implementations of (Z. Wang et al., 2004) have been used[28]. In case of the NIQE descriptor (Mittal et al., 2013), the original MATLAB implementation of the author has been used[29]. The *image descriptor comparator* in Figure 4-24 read the image description XML files using again the *mpeg7FexLib* library, which provided the distance measures for all deployed MPEG7 descriptors except for DCD. For DCD the distance measure presented in (Ma et al., 1997) has been used. The comparison results were then saved to a CSV-file for further processing. The distance measures of non-MPEG7 descriptors were computed in MATLAB and appended to the existing CSV file.

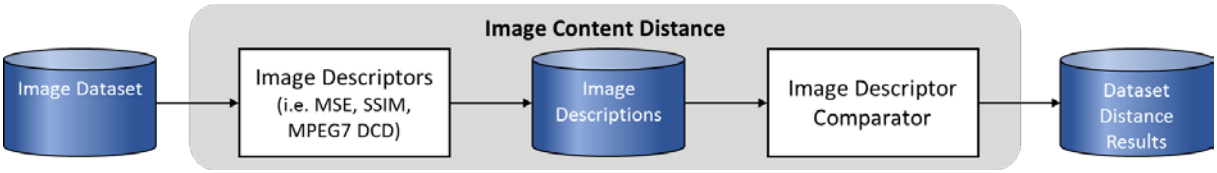


Figure 4-24: Image Content Distance Implementation

The result files acquired in the previous two steps were then loaded by the *influence factor analysis* implemented in MATLAB, which performed a *backward stepwise regression analysis* (see chapter 3.7).

[28] https://ece.uwaterloo.ca/~z70wang/research/ssim/ssim_index.m and https://ece.uwaterloo.ca/~z70wang/research/iwssim/psnr_mse.m [Last Accessed 10.05.2016]

[29] http://live.ece.utexas.edu/research/quality/niqe_release.zip [Last Accessed 10.05.2016]

5 Preliminary experiments and experimental datasets

This chapter first presents the preliminary experiments conducted to verify the necessary accuracy of used geographical data (synthetic environment) and employed methods (automatic ground truth generation) needed for further investigations. Thereafter, the datasets designed for the principle experiments are presented in detail.

5.1 Preliminary experiments

This chapter provides the necessary foundation for the principle experiments. First, the geographical accuracy of the modelled terrain database was investigated. Due to the considerable amount of test data, an automatic ground truth annotation has been implemented, which was based on the concept of homographic correlation (plane surface relation) between two views as explained in chapter 3.4. This approach is assumed valid due to the top-down perspective with negligible height differences concerning the altitude of the aircraft. This assumption was experimentally evaluated using three different samples from the dataset. In the last subchapter, the results of the ground truth acquisition used here are compared to published results of another implementation using well-known datasets.

5.1.1 Validation of database accuracy

Here, the positional accuracy of terrain maps was checked considering potential errors already present in source data (terrain textures) or introduced during the database modelling procedure. Validation was conducted by measuring the geo-position at six geographical locations distributed among the test area as depicted in Figure 5-1. These points were selected, because they were simple to identify and provided sharp edges for accurate selection in the terrain databases. The natural geo-position was acquired by measuring the location with a NAVILOCK 602U GPS mouse connected to a notebook. The software *Visual GPSXP* was used to measure the location of each geo-feature 100 times. The mean average of these samples was computed and defined as the *reference geo-coordinate* of that location (M).

Figure 5-1: Locations of measured geographical features.

Respective geo-coordinates of the corresponding features were manually extracted based on their visual appearance from the two databases of the virtual environment (HQ and LQ, see chapter 4.2.2). As comparison, the same features were also measured in *Google Earth* (GE) and the source (Src) texture data used to create the databases (chapter 0). Before computing the difference in distance from the measured reference (M), all coordinates were transformed from latitude / longitude using the WGS84 ellipsoid reference into *Universal Mercator Transformation* (UTM). The distance between the measurement (used as reference) and the specific dataset was computed using the Euler distance L_2. For all terrain datasets, the median, the 25th percentile, and the 75th percentile of the distances to the measured reference positions were computed and provided in Figure 5-2.

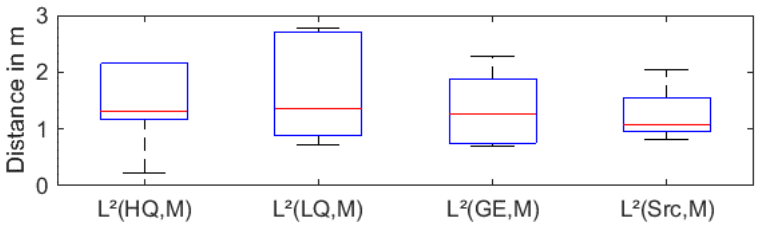


Figure 5-2: Euler Distance measures of the terrain databases, google earth and the source GIS geo-coordinates against the measured coordinates M.

The results show that the accuracy of all terrain databases are close to the accuracies of GIS data, demonstrating good positional accuracy of both databases. For the following investigations, this accuracy is sufficient. The higher precision of GIS data (depicted by a smaller box size) results from inaccuracies in visually identifying the exact geographical

features, which becomes more difficult with lower visual quality of the satellite texture. Overall, the results show an average error of 1.5m ± 1.2m. This amount of error is close to the error of the source data and thus demonstrates the high accuracy of the generated terrain databases.

5.1.2 Validation of auto-generated ground truth concept

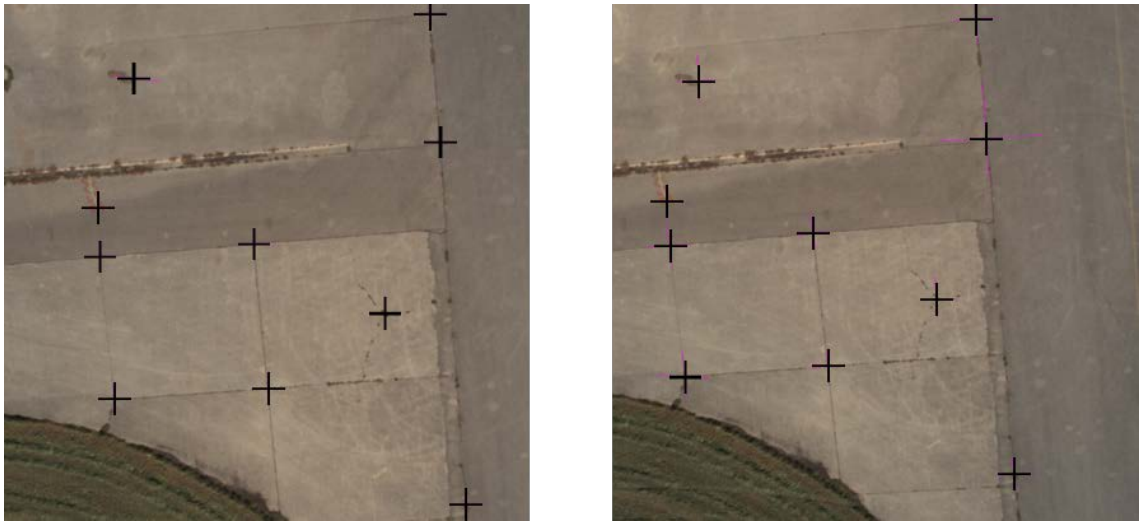


Figure 5-3: Image pair example (Frame 7869, left; Frame 7884, right) with annotated ground truth (black crosshairs).

Ground truth necessary for the main experiments (chapter 6) was auto-generated using the approach proposed in (Mikolajczyk et al., 2005) (chapter 3.4). It is based on a homographic correlation between these two views, which can be achieved by either having no translational movement between the images (which is not valid in this case) or by displaying a homogenous scene (flat-earth model). All following experiments were based on image data recorded from a camera, mounted perpendicular to the airframe of the Multicopter. The camera was equipped with a perspective lens (FOV = 25° in horizontal and vertical angle), flying at a constant altitude of 70m above ground level. The experiment was conducted on the 01.10.2015 at 11:15 a.m. with sunny weather conditions. Due to the field of view, the flight altitude and the levelled terrain, a flat-earth (scene) is assumed. In this chapter, manually annotated ground truth is used to evaluate the accuracy of automatic ground-truth generation. In Figure 5-3, an example is presented demonstrating the result from the manual annotation process. All pixel pairs between the two views have been numbered and their coordinates (with full-pixel accuracy) logged into text files. In each view, ten easily detectable features distributed on the region covered by both views have been selected.

Since manual ground truth annotation is expensive, three typical example image pairs have been selected representing the variants of scenes during the test flights. The image pair in Figure 5-4a *street* depicts a flat structured scene containing no objects of height > 0.1m. This image pair shows a completely homogenous flat surface that is expected to work well with homography-based ground truth methods. The image pair in Figure 5-4b *forest* contains a natural scenery of trees with an altitude between five and ten meters. Additionally, the ground surface itself is only partially visible. This pair shall help to evaluate the accuracy of ground truth in sceneries with large natural objects. In Figure 5-4c, *hangar* a complex scene depicting numerous man-made objects including a large hangar and its shadow challenge the automatic ground truth methods due to the difference in altitude (the building is 6m high at the highest point) and the repetitive texture of the roof.

(a) *street* **(b) *forest*** **(c) *hangar***

Figure 5-4: The three sample pairs to measure the accuracy of automatically generated ground truth. Each sample consists of an image pair with synchronized telemetry data.

The resulting manually acquired image pair coordinates have been used as reference for the automatic ground-truth generation methods: *Image based homography estimation* (further called IHE) and *Telemetry-based homography estimation* (THE). Additionally, a third homography matrix was computed using the manually annotated points as inputs to evaluate the maximum possible accuracy possible with the homography approach for the specific

samples. The IHE extracted feature points using the SURF detector and matched them with brute-force matching based on their SURF descriptions. These points were used as inputs for an initial homography estimation. A RANSAC algorithm then used all found feature pairs to fit the homography plane optimally (see chapter 3.4 for more details).

In contrast, THE used position and attitude information of the aircraft acquired through the aircraft sensors presented in chapter 4.1.1 to calculate the geographic outline the camera depicted on the ground (*sensor footprint*). The method to compute the homography matrix for THE is detailed in appendix B.

After having defined the image-based ${}^{IHE}H_{10}$ and telemetry-based homography ${}^{THE}H_{10}$, for comparison a third homographic relation ${}^{MGT}H_{10}$ using the manually annotated ground truth points (p_0, p_1) was generated. For each of the three image pairs presented in Figure 5-4 all three homography matrices were computed and the ground truth was used to evaluate their accuracy by measuring the deviance of point $p_1' = H_{10} * p_0$ and p_1 .The absolute deviance d was acquired by computing the Euclidean distance of the x and y components of p:

$$d = \|p_1 - (H_{10} * p_0)\|_2 \tag{31}$$

Table 5-1: Deviations of estimated and actual points using homography on image pair *street*.

street	$d({}^{MGT}H_{10})$	$d({}^{IHE}H_{10})$	$d({}^{THE}H_{10})$
Mean [px]	0.53	0.90	205.32
Std. Dev. [px]	0.34	0.23	1.59
Mean [m]	0.02	0.03	7.45
Std. Dev. [m]	0.01	0.01	0.06
RMS [px]	0.62	0.93	205.33

The image pair evaluated first was *street*. The results are depicted in Table 5-1 as mean and standard deviation over all ground truth features, both in pixel and meter. The *root-mean-square error* (RMS) serves to simplify the comparison to ground truth accuracies by (Mikolajczyk et al., 2005), where the resulting RMS deviance was less than 1 pixel. Using the points acquired by annotation the RMS value of 0.62 pixel deviance between projected and actual position, correlates with (Mikolajczyk et al., 2005) even though the scene *street* and the experiment in general were much more complex than Mikolajczyk's datasets. Compared to manual annotation (MGT) the automatically acquired homography IHE cannot achieve the same accuracy, however because the accuracy is still below one pixel it is acceptable. *Telemetry-based homography estimation* (THE) on the other hand shows the problem of accumulated error resulting from the need to combine sequential AHRS measurements

(forward and backward computation). Interestingly the standard deviation of THE results is only about one pixel higher than that of other homography matrices suggesting that the orientation measurements have been precise. The lateral positioning has been identified as the main source of error of which hints to insufficient accuracy of the position measurement (GPS (4Hz) enhanced with Kalman filtering). Standard deviations of MGT and IHE are higher than the actual measurement values indicating different accuracies of the selected points showing the limit of manual image annotation (selecting the exact pixel in both images). When considering the actual deviance in meters, a mean accuracy of 5cm using image based homography presents a good result taking into regard the complexity of the experiment. Even the mean deviation of 7.45 meters of the telemetry approach is not too bad, considering that the positional error of the GPS is rated at 2m CEP. However, the telemetry-based approach does not provide the accuracy necessary to conduct the planned experiment. Thus, *image based homography estimation* is used in the main experiments of this thesis as automatic ground truth generation method.

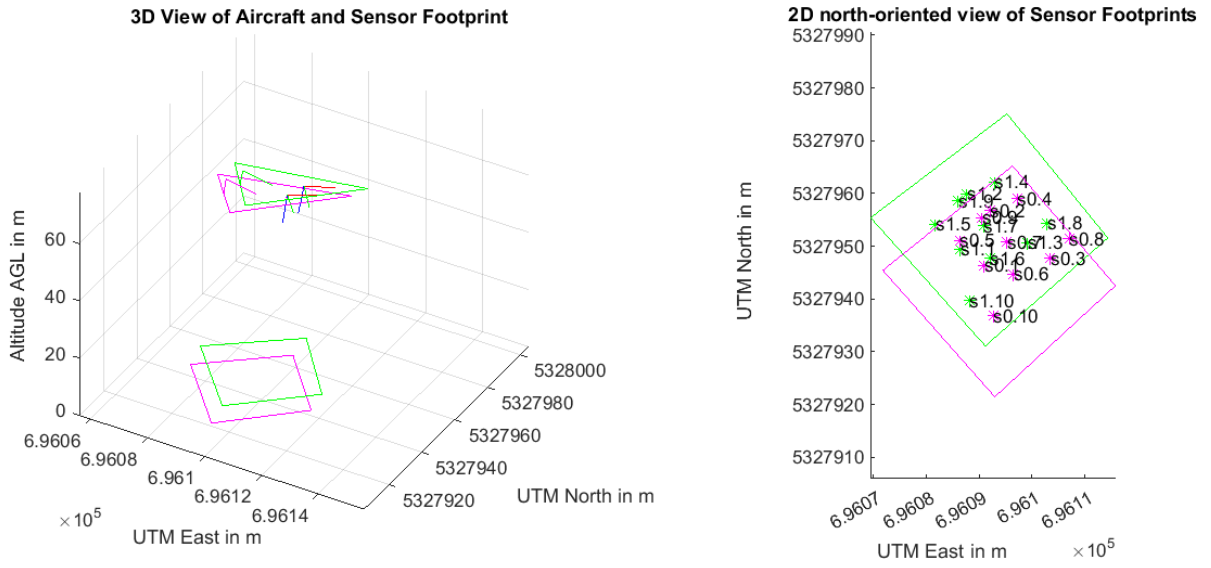


Figure 5-5: 3D and 2D visualization of image pair *street* (0 = magenta, 1= green) depicting aircraft and sensor footprint on the left and sensor footprint with ground truth points on the right.

In Figure 5-5, the left diagram shows the 3D representation of the s*treet* image pair with the sensor footprint of image 0 coloured in magenta and image 1 in green. Position and orientation of the aircraft during the image acquisitions is indicated by simplified representation in the same colours in metric Cartesian coordinates. The right diagram shows the sensor footprints in a 2D top-down view together with ten ground truth pixel coordinates converted into world space for both images (s0.6 = intersection point of ground truth point 6

with the world surface in image 0). Thus, the resulting deviation (distance between s0.x and s1.x points) between the two views can be derived.

Scene *hangar* introduces an occluded scene with several objects of different height (hangar = 7m height). Table 5-2 displays the rise of deviance in MGT data, which can be directly correlated to the flat earth condition of the homography (annotation accuracy is assumed to be similar to image pair *street*). For MGT and IHE the deviation mean and the standard deviation rise. The standard deviation of IHE is lower than MGT since the optimization algorithm reduced the deviation by determining a plane that fits most points. In this case, the pixels on the right are on higher ground than pixels on the left of the image leading to a shifted homography. THE's standard deviation is about two times higher than of IHE or MGT and the deviation mean is even 36 times higher. This underlines the previously made assumption that the GPS localization is the main source of error for THE. When displaying the results in meters the differences between the actual points as identified by manual annotation and the projected point using the estimated or calculated homography is 0.09m for IHE and 3.33m for THE in average. While this example demonstrated the introduction of noticeable errors for cluttered scenes, the concept of using homography as automatic ground truth is not compromised. However, the standard deviation of the experiments needs to be monitored in order to react in case the deviations reach critical levels.

Table 5-2: Deviations of estimated and actual points using homography on image pair *hangar*.

hangar	$d(^{MGT}H_{10})$	$d(^{IHE}H_{10})$	$d(^{THE}H_{10})$
Mean [px]	2.23	2.39	90.42
Std. Dev. [px]	2.65	2.28	4.74
Mean [m]	0.08	0.09	3.33
Std. Dev. [m]	0.10	0.08	0.17
RMS [px]	3.36	3.23	90.54

Image pair *forest* has been selected for a worst-case evaluation. It presents a scene with many occluded trees of varying height and many natural features, which do not provide clear edges or corners. Surprisingly the evaluation results in Table 5-3 can be ranked between *street* and *hangar* for MGT, meaning the ground truth points can be represented quite well with a homography. IHE is still equal to values from *hangar*, even providing lower standard deviation indicating good representability using homography. In addition, the telemetry homography estimation presented its lowest results, due to high overlap between the images (see Figure 5-6) and the low velocity of the aircraft compared to other scenes. Still, the measured errors are far above the limit disabling this method as a valid option.

Table 5-3: Deviations of estimated and actual points using homography on image pair *forest*.

forest	$d(^{MGT}H_{10})$	$d(^{IHE}H_{10})$	$d(^{THE}H_{10})$
Mean [px]	0.90	2.52	31.13
Std. Dev. [px]	0.84	1.30	6.41
Mean [m]	0.03	0.09	1.15
Std. Dev. [m]	0.03	0.05	0.24
RMS [px]	1.20	2.81	31.72

The results present IHE as a valid option, while THE has been identified as too inaccurate. The resulting standard deviation errors now need to be set in to perspective to identify their impact on used performance criteria (see chapter 3.4). Currently the size of an (non-projected) region is normalized to 30 pixels. Assuming no deviation in scale, the distance at which the overlap error ϵ_O (see equation (6)) is higher than 40% is six pixel. Thus, if mean deviation of the ground truth is larger than six pixel *repeatability* cannot be measured. This limit defines the necessary accuracy of the ground truth homography.

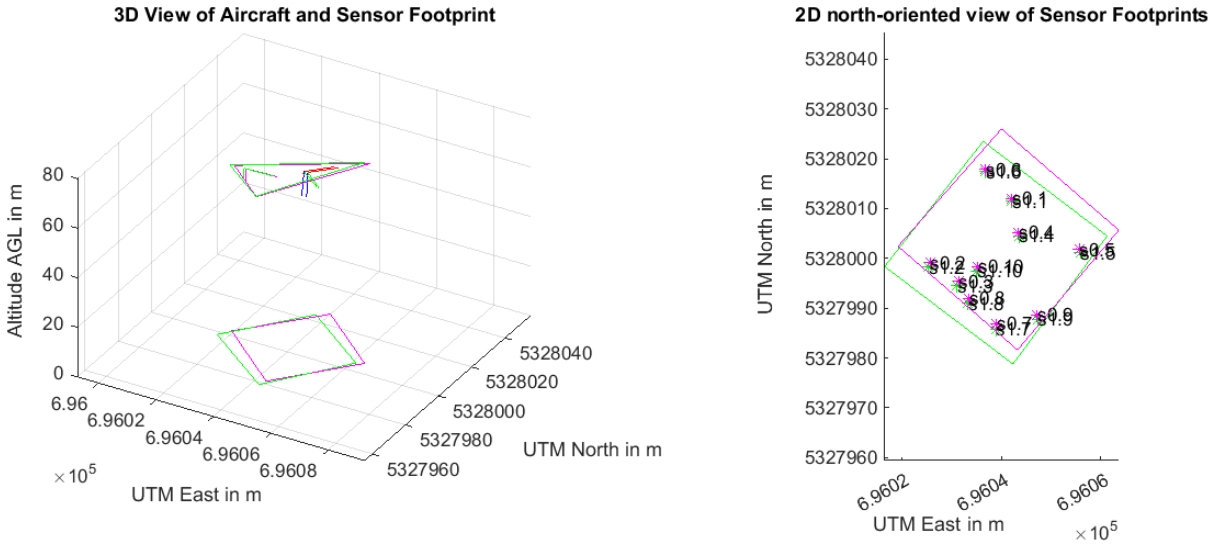


Figure 5-6: 3D and 2D visualization of image pair *forest* (0 = magenta, 1= green) depicting aircraft and sensor footprint on the left and sensor footprint with ground truth points on the right.

5.1.3 Validation of automatic ground truth computation implementation

The computation method of ground truth homography matrices has been implemented in C++ similar to (Mikolajczyk et al., 2005) using OpenCV. A detailed description of this approach can be found in chapter 3.4. This approach differs from (Mikolajczyk et al., 2005) by removing the manual feature selection step to identify the approximate homography matrix and the warping of the test image using this matrix to then compute the residual homography using RANSAC. Here, the SURF-algorithm was used to automatically detect and describe features, which were then matched using a brute force method. The resulting approximate

homography was used as initial value for the RANSAC-based optimization, which optimized the results by reducing the error and identifying outliers. In this chapter, this implemented method is compared to the resulting homography matrices of Mikolaczyk, which are provided together with the image datasets on the website of the visual geometry group of the University of Oxford[30]. The dataset *graffiti* used in this evaluation consisted of six images showing a graffiti on a wall at increasing viewing angles (examples presented in Figure 5-7).

(a) **(b)** **(c)**

Figure 5-7: Example images of dataset *graffiti* used in (Mikolajczyk et al., 2005) at increasing viewing angles from 0° (a) over 20° (b) to 30° (c).

Seven points in the reference image of "graffiti" have been selected as reference points and were manually annotated for every image. Using these points, the *manually annotated ground truth homography* ${}^{MGT}H$ (in chapter 5.1.2) was computed. *Mikolajczyk's homography estimation* matrices ${}^{MHE}H$ were provided in the dataset. These were used as references for the image-based homography matrices ${}^{IHE}H$ computed using the aforementioned implementation. The comparison was performed by projecting manually annotated points p_x of an image to reference image 0 using the homography matrix and the projection error was computed using the Euclidean distance L_2 of p (Equation (31)). The distance measurement statistics of all points are then provided using *RMS* and *standard deviation* in Table 5-4. These show in general that projections using ${}^{MGT}H$ perform best, which is not surprising because the homography has been computed with the reference points used in this evaluation. This indicates whether the datasets can be represented with a homography and demonstrates the lowest possible error. The homography matrices of Mikolaczyk are very close to MGT at all viewing angles only rising above one pixel RMS at 40°. The implementation IHE provides also results below or equal one pixel RMS for viewing angles up to 30° but then leads to high

[30] http://www.robots.ox.ac.uk/~vgg/research/affine/index.html [Last Accessed: 19.01.2016]

error rates. Similarly, the standard deviation for all the types of homography acquisition are similar up to a viewing angle of 40°.

Table 5-4: Evaluation of the ground truth implementation IHE used in this thesis against manually annotated ground truth MGT and ground truth used in (Mikolajczyk et al., 2005) MHE.

Viewing Angle	Measures	$d(^{MGT}H)$	$d(^{IHE}H)$	$d(^{MHE}H)$
20°	RMS	0.68	0.92	0.85
	Std. Dev.	0.38	0.49	0.39
30°	RMS	0.62	1.00	0.93
	Std. Dev.	0.26	0.52	0.50
40°	RMS	0.94	7.34	1.22
	Std. Dev.	0.41	4.09	0.51
50°	RMS	1.21	979.61	1.71
	Std. Dev.	0.95	731.68	1.09
60°	RMS	1.78	637.03	1.98
	Std. Dev.	1.63	250.01	1.44

The high values at wider viewing angles result from the strong distortion of features, which reduces their recognisability for the automatic acquisition method used in the IHE approach. However, as depicted in Figure 5-8 the errors of IHE are very close to MHE and MGT for viewing angles smaller than 30° and due to the fact that the experiments use a fixed camera a viewing angle difference larger than 30° will not appear in the experiments. Thus, the implementation is providing good results as long as the viewing angle difference is not above 30° and will be used further in this work.

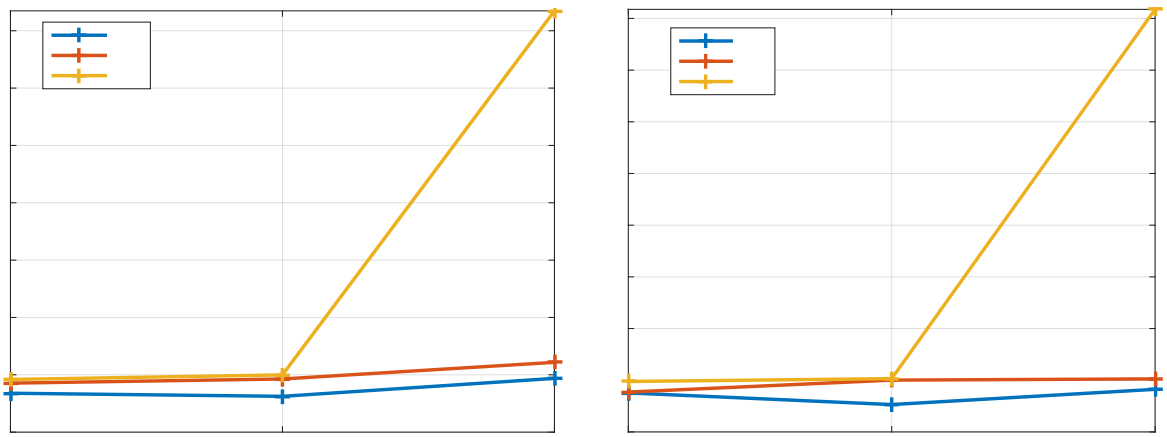


Figure 5-8: Reprojection Error in RMS (left) and Standard Deviation (right) for all tested homography estimation implementations on the dataset "graffiti".

5.1.4 Validation of image content distance measures

The image descriptors have been designed for use cases such as image quality assessment presented in chapter 2.5.1 (PSNR, MSE, MSSIM and NIQE) or image retrieval (CSD, CLD, SCD, DCD, HTD, EHD) presented in chapter 2.5.3. Their functionality as image content distance measure had to be validated in order to use them in further investigations.

A capable measure needs to able to identify whether two images are from different datasets or from the same dataset (all depicting the same scene). Thus, the distances were measured **within** a dataset (by comparing two subsequent images and iterating through the dataset) and had been compared to the distances that arised **between** the two dataset types.

The resulting distances within each dataset are $Within_b$ for *baseline* and $Within_p$ for *photo*. The image content distance results between the two datasets are denoted $Between_{bp}$. The deployed distance measures for each descriptor were detailed in chapter 3.6.

The boxplots in Figure 5-9 depicts the within and between results of each image content descriptor on scene *concrete*. Each box statistically summarizes the results of all images in a scene. The red line is the median, the upper and lower outline of the blue box the 25th and 75th percentile of all measurements. Whiskers define the maximum and minimum value within a standard deviation of 3σ. Outliers are depicted with a red plus.

As depicted in Figure 5-9 the descriptors NIQE, MSE, CSD, CLD, SCD, DCD, HTD and EHD can clearly separate the different image types. The measures PSNR, MSSIM and DCD are unable to do this in this scene. To evaluate their capabilities objectively their discriminative power was statistically analysed for all scenes.

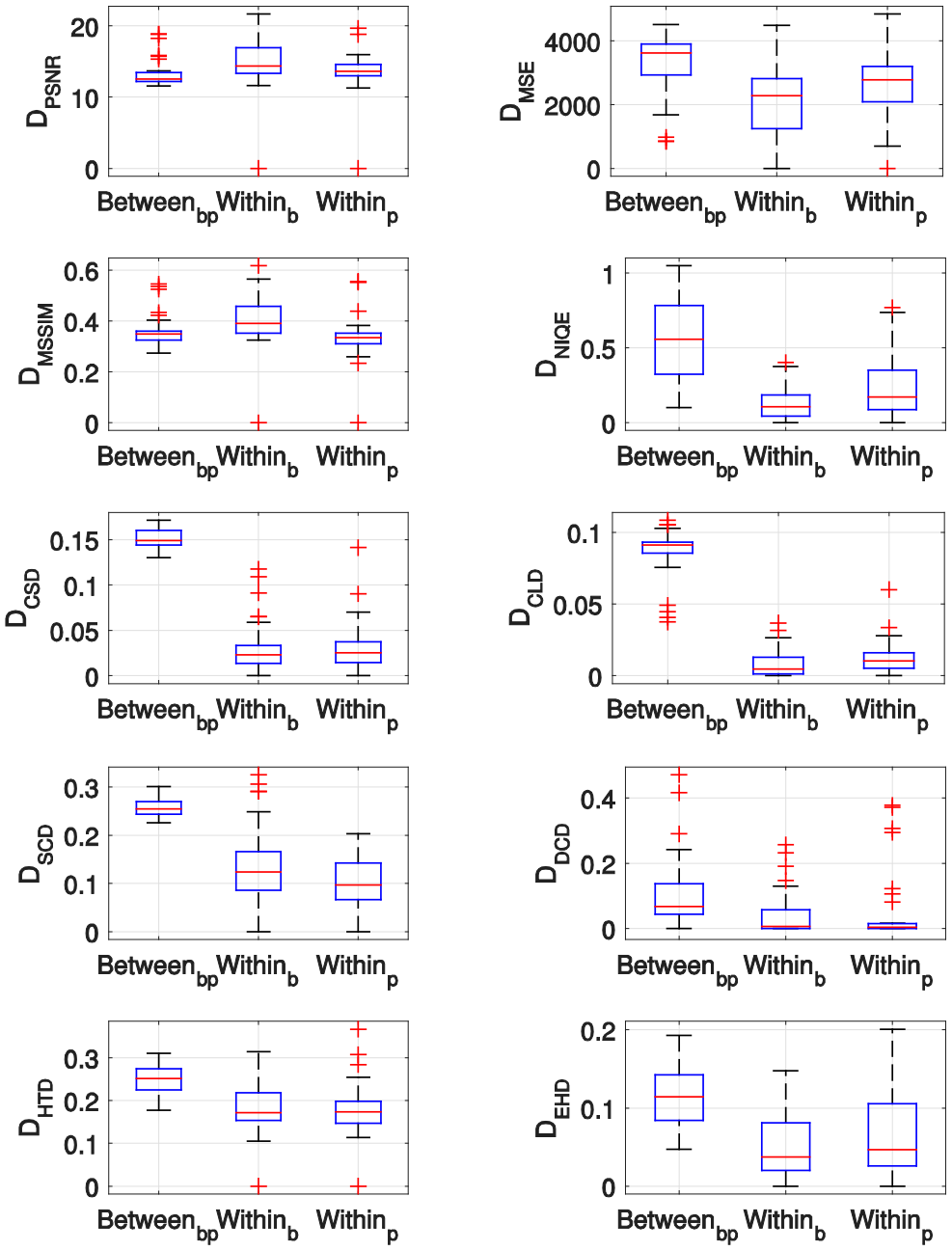


Figure 5-9: Descriptor distances for scene *concrete*. *Between*$_{bp}$ presents the distance between *photo* and *baseline*, *Within*$_b$ the distances within *baseline* and *Within*$_p$ the distances within *photo*.

Statistical analysis

To identify the sensitivity and thus the usability of a measure the results have been statistically analysed. Therefore, the mean distance $Between_{bp}$ was compared against the mean distance between two subsequent images of the reference (*photo*) dataset. The probability whether a measure can differentiate between the two was evaluated using the independent *t-test* measure, where $\overline{D}_{between}$ is the mean distance of all images between the two configurations,

$\overline{D}_{ref}$ the mean distance within the reference, $n_{between}$ the number of images used in the between evaluation and n_{ref} the number of *'inner'* images:

$$t = \frac{\overline{D}_{between} - \overline{D}_{ref}}{\sqrt{\frac{s_p^2}{n_{between}} + \frac{s_p^2}{n_{ref}}}} \tag{32}$$

The number of samples differs since the sequential evaluation always needs two images for each measurement. Thus its number of evaluations is $n_{ref} = n_{between} - 1$. Due to the varying sample size, the standard deviation s_p had to be calculated accordingly:

$$s_p^2 = \frac{(n_{between} - 1)s_{between}^2 + (n_{ref} - 1)s_{ref}^2}{n_{between} + n_{ref} - 2} \tag{33}$$

With $s_{between}^2$ being the standard deviation of the between dataset measures and s_{ref}^2 the standard deviation of the within reference dataset measures. The significance is given by the probability p (read from the cumulative distribution function of *Students t-distribution* using the acquired t-value (Student & Gosset, 1908)). The effect size *r* is given using the *Pearson linear correlation coefficient* (PLCC), which can be computed using the t-value:

$$r = \sqrt{\frac{t^2}{t^2 + df}} \tag{34}$$

Where df is the degree of freedom that is:

$$df = n_{between} + n_{ref} - 2 \tag{35}$$

This evaluation will identify measures more sensitive to changes between natural and synthetic images and insensitive measures. In case $Between_{bp}$ values differ significantly from the $Within_p$ results, the measure can identify the differences between two different image content types. If the deviation is non-significantly different, the measure shows no ability to distinguish both image types.

In Table 5-5 the resulting effect sizes *r* for all measures and scenes are presented together with their probability-value *p*. If $p < .05$ the resulting effect size is statistically significant. PSNR and MSSIM cannot provide significant results, which means these measures cannot distinguish between natural and synthetic imagery. All other measures show large effect sizes

in high significance indicating a strong capability to distinguish synthetic images from photographs.

Table 5-5 Effect sizes *r* and their probability value *p* encoded using asterisks presenting, which measures are able to distinguish images of different content. Large results are bold.

Effect Size *r*	PSNR	MSE	MSSIM	NIQE	CSD	CLD	SCD	DCD	HTD	EHD
concrete	.043 Ns	.241*	.162 Ns	**.538*** **	**.949*** **	**.929*** **	**.899*** **	.272*	**.566*** **	**.471*** **
forest	.595 Ns	**.847*** **	.008 Ns	**.717*** **	**.977*** **	**.977*** **	**.948*** **	**.701*** **	**.547*** **	**.513*** **
hangar	.311 Ns	**.506*** **	.216 Ns	**.764*** **	**.964*** **	**.992*** **	**.928*** **	**.730*** **	**.896*** **	**.638*** **
heath	.602 Ns	**.761*** **	.077 Ns	**.919*** **	**.983*** **	**.889*** **	**.957*** **	**.500*** **	**.904*** **	**.650*** **
house	.525 Ns	**.725*** **	.022 Ns	**.863*** **	**.976*** **	**.946*** **	**.906*** **	**.599*** **	**.734*** **	**.476*** **
junkyard	.479 Ns	**.753*** **	.107 Ns	**.903*** **	**.920*** **	**.969*** **	**.971*** **	**.704*** **	**.743*** **	**.841*** **
sport	.733 Ns	**.931*** **	.418 Ns	**.768*** **	**.992*** **	**.913*** **	**.980*** **	**.747*** **	**.904*** **	**.878*** **
street	.569 Ns	**.829*** **	.274 Ns	**.604*** **	**.982*** **	**.760*** **	**.978*** **	**.662*** **	**.896*** **	**.468*** **

Ns = not significant (p > .05), *p < .05, ** p < .01, *** p < .001

Thus, Table 5-5 shows that except for PSNR and MSSIM all investigated descriptors are able to distinguish image of two different types. Consequently, the two unable descriptors won't be used in further investigations. The image descriptors MSE and NIQE can differentiate very well, however due to their original purpose and concept to detect any kind of image error, they cannot be used to pinpoint on a specific causing property. Thus, these measures allow no conclusion on which image property needs to be adapted in order to reduce the visual difference of natural and synthetic data. It can be summarized that some image quality assessment measures have the ability to discern natural and synthetic data, however cannot present the reason of difference. Thus, in further investigations the measures CSD, CLD, SCD, DCD, HTD and EHD are deployed. In future investigations, current IQA measure such as MSSIM could be separated in measures determining only structure, brightness or contrast between two images.

5.2 Experimental datasets

The core interest of this evaluation lies in the performance difference of computer vision algorithms on natural images (photographs) and synthetic images (computer-generated imagery). In chapter 4.1 and 4.2, the implementation to generate images of both kinds had been presented. In the context of this investigation, a *dataset* is defined as a group of sequential images of one scene and one configuration. A *scene* presents a specific geographic location, at a specific time in a defined camera position and angle. The next subchapter details this explanation. The *configuration* specifies the changed parameter in the rendering of the

synthetic environment in reference to the standard configuration *baseline*. All configurations are presented in chapter 5.2.2. Each dataset is represented by 35 images, since according to Table 3-4, at least 35 samples are necessary to enable statistical measurements of medium correlation effects (r = 0.3). For correct comparison between datasets, images need to be of equal resolution. The camera used during the flight experiment provides a native resolution of 2048x2048 pixel with an aspect ratio of 1:1. The resolution for dataset images has been fixed to 1024x768, since the synthetic environment supported it, and it allowed resizing of natural images without interpolation while additionally most of the image content could be retained (aspect ratio). The natural images were converted into this resolution by resizing the image to 1024x1024 followed by cropping the lower and upper 128 rows. This allowed recording in the natural resolution of the camera and keeping the horizontal field of view due to vertical cropping. The resulting images therefore were depicting roughly the same scene (limited by accuracy of the UAV's AHRS system) with the same field of view in the same resolution. Thus, the difference of image content is limited to the different representation of a scene in natural and rendered imagery. In summary, each dataset depicts a specific scene with 35 sequential pictures with a resolution of 1024x768. The interval between dataset images was constant within a scene, but differed for each scene due to varying lengths of shots. The following chapter describes the captured scenes in detail.

5.2.1 Description of scenes

The test flight (chapter 4.1) had been categorized in eight scene types (Figure 5-10) to provide different challenges for the tested feature detectors, and thus vary in number and type of natural and man-made objects, terrain surface and presence of shadows. Telemetry (necessary to create synthetic datasets) and imagery (for natural datasets) data were recorded during a flight experiment on the 01.10.2015 at 11:15am with sunny cloudless weather. The UAS (chapter 4.1) was preprogramed with the flight path as depicted in Figure 5-10 to ensure exact route following. The altitude had been fixed to 70m above ground level. Take-off and landing was performed by a safety-pilot.

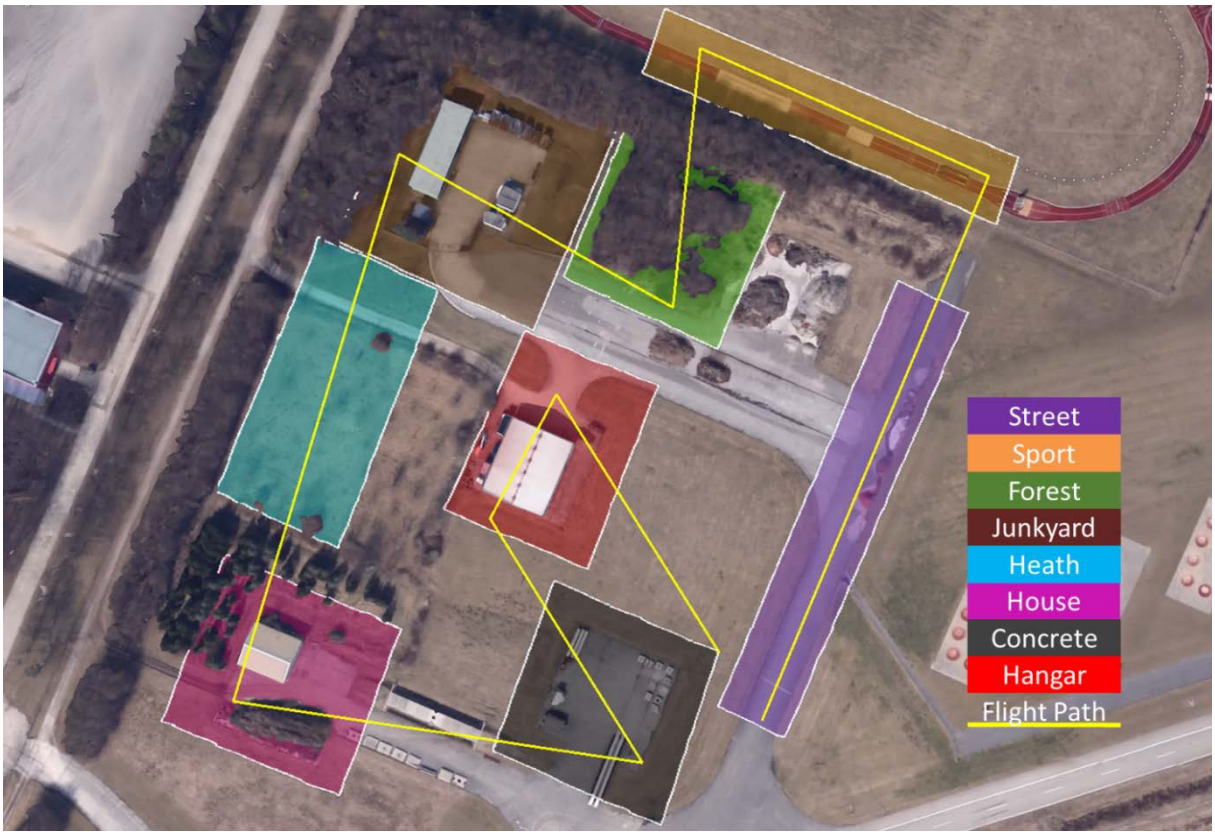


Figure 5-10: Part of the test flight area used to generate synthetic and natural datasets, together with aircraft route of the test flight and defined scene types.

For each scene of the reference dataset *photo*, several corresponding datasets were generated in the virtual environment (one for each configuration). In Figure 5-11 and Figure 5-12, samples present each scene for *photo* and *baseline*. Please note that **synthetic scenes** always differ to some extend to natural scenes since they are **limited by the appearance of the aerial image used as ground texture** (e.g. daytime, season, weather or current situation).

Figure 5-11: The eight scenes in reference dataset *photo*. From upper left to lower right: *Street, sport, forest, junkyard, heath, house, concrete* and *hangar*.

Figure 5-12: The eight scenes in synthetic dataset *baseline*. From upper left to lower right: *Street, sport, forest, junkyard, heath, house, concrete* and *hangar*.

The scene *street* is of simple complexity, depicting a paved *street*, a concrete field and meadow. The richly textured surface is free from any objects or bushes. The terrain itself is extremely flat, making this scene the most planar out of all tested.

Scene *sport* contains a tartan track, meadow and shadows of trees. The scene itself is also flat except for the obstacles on the track. Further, trees outside of the images cast shadows on the track making this dataset preferable for testing the effect of shadows on (almost) planar surfaces.

Scene *forest* puts great demands on the homography condition since it consist almost exclusively of various densely put trees in different sizes. This dataset is useful to analyse the test objects performance on scenes with occluded natural objects.

Junkyard is the first scene containing a larger building. Furthermore, the scene depicts meadow, trees, concrete surface, several small man-made structures, trash container and small objects. Due to the low height of most objects, shadows are rare. This scene is preferable when effects on a high number of small man-made objects and homogenous industrial textures shall be tested.

In scene *heath*, a meadow with several sparsely placed trees of medium height (2-4m) and bushes is depicted. A gravel road separates the meadow. The scene is preferable when in the influence of natural textures; shadows and vegetation shall be tested.

The second scene containing a building is *house*. This house is roofed with corrugated metal. In the scene there are also trees, meadow, a concrete surface, two vehicles, a pile of earth and

two people depicted making it the most versatile dataset. These persons are moving and have been modelled in the synthetic scenes.

In scene *concrete*, several man-made structures and objects can be seen on a concrete surface. The structures are heterogeneous in colour, form and height making this dataset interesting for analysing textures and lighting.

The last scene *hangar* depicts the largest and tallest building in the dataset together with a shipping container, a garage, two cars and two trailers. Additionally, trees occlude the scene. The surface is switching between freshly mowed grass and concrete. Both, trees and the hangar cast large shadows on the surface. The high amount of man-made objects, the large building with the homogenous texture, large shadows and how occlusion make this scene probably the most demanding for database modelling. Additionally, the behaviour of the aircraft differs between the different scenes as presented in Table 5-6.

Table 5-6: Duration and aircraft movement description of scenes.

Name	Duration in s	Time Frame δt in s	Movement description	Total Distance covered in m	Distance covered per Frame in m
street	18.95	0.54	Fast transition	109	3.1
sport	20.13	0.58	Fast transition	106	3.0
forest	20.75	0.59	Hover and transition	43	1.2
junkyard	18.16	0.52	Slow transition	49	1.4
heath	17.12	0.49	Fast transition	96	2.7
house	22.83	0.65	Hover and transition	65	1.9
concrete	24.64	0.70	Hover and transition	42	1.2
hangar	12.14	0.35	Slow transition	29	0.8

5.2.2 Synthetic environment configurations of test datasets

After having acquired the reference natural dataset through UAV test flights, the synthetic datasets had to be prepared to allow comparison of the test algorithm performances on both types of datasets. To investigate the effect of technical parameters (of database generation and rendering) of the synthetic environment a considerable number of synthetic test datasets was generated. To support a systematic approach, related parameters of potential influence were grouped in five sets (*configuration sets*): *illumination*, *texture*, *edge*, *3D-objects* and *camera model*. Starting from a *baseline* configuration (default parameter set), variations of these parameters resulted in a specific *configuration* for each investigated parameter. In the following chapters configuration sets and contained configurations are explained. For each *scene* (eight in total) and *configuration* (21 in total) a synthetic test *dataset* was generated

leading to 168 investigated *datasets*. In every *configuration set*, the synthetic configurations have been compared to the reference natural dataset *photo* and the default synthetic configuration *baseline*.

5.2.2.1 Configuration set “Illumination”

In *configuration set “Illumination”*, all parameters modifying illumination properties are grouped. It needs to be mentioned that VBS3 (the used rendering engine) provides a *hemispherical* lighting model that computes brightness, direction and colour tone of sunlight considering the date, time of day, weather and the geolocation of the depicted scene. These have been set to coincide with the actual recorded imagery of the flight experiment for all evaluations.

The investigated parameters are *Shadow Detail*, *Shadow Filtering* and *Screen Space Ambient Occlusion* (SSAO). In configuration *baseline* configuration all mentioned parameters are disabled, thus resulting in images without shadows and SSAO. The *configurations* and their parameter settings are presented in Table 5-7. Example images can be found in Figure 5-13.

Table 5-7: Configuration set *“Illumination”* with parameter settings.

Configuration	Shadow Detail	Shadow Filtering	Screen Space Ambient Occlusion (SSAO)
baseline	*Disabled*	*Disabled*	*Disabled*
shadow	**High**	Disabled	Disabled
shadow filter	**High**	**Enabled**	Disabled
SSAO	Disabled	Disabled	**High**

In configuration *shadow* the shadow detail is set to high, which activates the shadow buffer techniques (cascaded shadow maps (CSM) (Dimitrov, 2007) and variance shadow maps (VSM) (Donnelly & Lauritzen, 2006); see appendix A). The shadow is generated by creating a depth-view from the light sources viewpoint and declaring non-visible pixels as shadow. Since depth computation is applied after rasterization, shadows may appear aliased. This *configuration* shall investigate the influence of shadows on the test algorithms performance.

Configuration *shadow filter* additionally filters the shadows of the previous configuration using percentage closer filtering (PCF) (Bunnell & Pelacini, 2004). This filter reduces the aliasing of shadows.

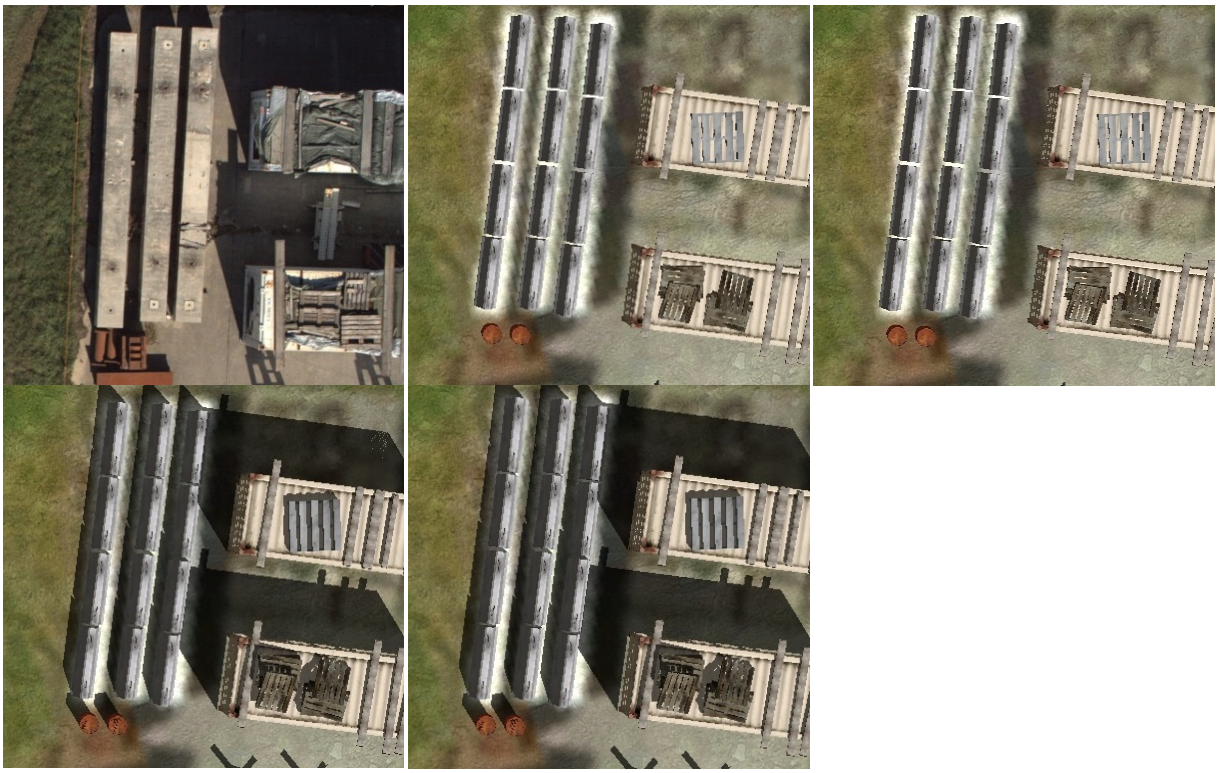


Figure 5-13: Examples of set *Illumination* on *concrete* (image 35). Upper left to lower right: *photo*, *baseline*, *SSAO*, *shadow* and *shadow filter.*

The last configuration of this set *SSAO*, activates the computation of *screen space ambient occlusion* (SSAO) (Bavoil & Sainz, 2008) in its highest fidelity as implemented in VBS3. This technique darkens image pixels that acquire less ambient light due to occlusion. Such occlusion can originate from other objects or complex geometries. This effect exclusively concurs on objects. Since ambient occlusion most prominently appears in dim lit complex indoors environments, the influence this technique has on outdoor top-down imagery is investigated.

5.2.2.2 Configuration set “Texture”

Texturing is an essential part of database generation since it improves the visual appearance of 3D-objects, terrain surface and vegetation (3D-object and 2D-sprite representation) while keeping the object complexity (wire mesh) simple (for real-time applications. The ground surface is modelled in VBS3 as a *layered terrain surface representation* (see chapter 0) where the satellite image of the location is combined with a procedural detail map to improve the visual quality near ground. The parameters grouped in *texture* are *Texture Detail*, *Satellite image texture resolution* and *Anisotropic Filtering* (AF). Configuration *baseline* sets *texture detail* to very high, *image resolution* to 0.2mpp and disables *AF*. The parameter *Texture Detail*

(see Table 5-8) reduces the resolution and therefore the detail of all textures used in the engine.

Table 5-8: Configuration set *"Texture"* with parameter settings.

Configuration	Texture Detail	Satellite image texture resolution	Anisotropic Filtering (AF)
baseline	*Very High*	*0.2mpp*	*Disabled*
texture low	**Low**	0.2mpp	Disabled
surface low	Very High	**5mpp**	Disabled
AF	Very High	0.2mpp	**Very High**

As depicted in Figure 5-14 this results in a reduced prominence of the detail map and creates aliasing artefacts in the satellite imagery. The respective configuration is named *texture low*. In configuration *surface low* the satellite image texture resolution is lowered to 5mpp, which greatly hides ambient geo-specific details. In *AF,* the anisotropic filter is set to maximum. This improves the quality of textures at oblique viewing angles in reference to the camera. In a top-down view, this mainly affects the sides of 3D-objects.

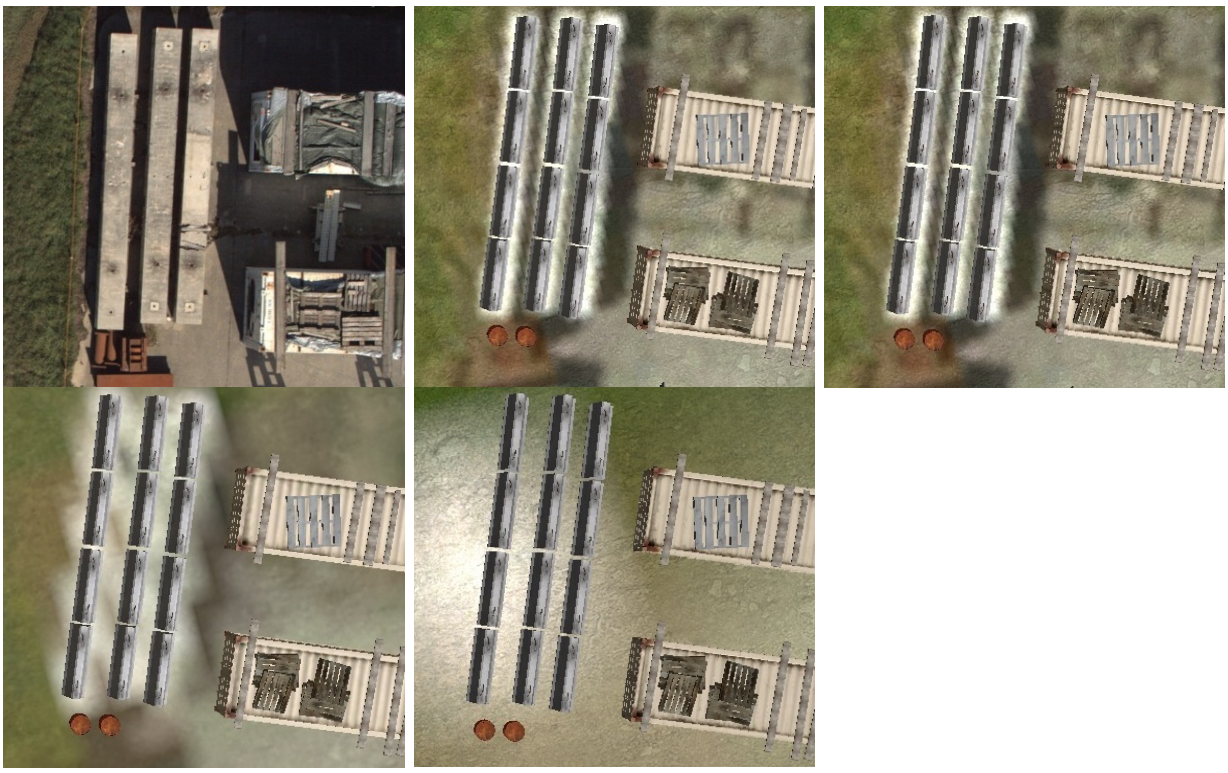


Figure 5-14: Examples of set *Texture* on scene *concrete* (image 35). Upper left to lower right: *photo, baseline, anisotropic filtering (AF), Texture low* and *Surface low*.

5.2.2.3 Configuration set "Edge"

Configurations in this set consider the activation of four different anti-aliasing techniques, namely *Multi Sampling* (*MSAA*), *Fast Approximate* (*FXAA*), *Subpixel Morphological (SMAA)* and *Super Sampling* (*SSAA*). These techniques all aim to reduce the jagged nature of sharp edges or lines, which are introduced during rasterization. They mainly differ in image quality and required computation effort. As a side note, aliasing could also appear in natural images since the imaging sensor also performs rasterization, however in general the deployed optics produce enough natural blur to remove any antialiasing artefact. The methods can be roughly categorized in methods incrementing the sampling rate (*SSAA* & *MSAA*) or post-processing methods blurring the rendered image (*FXAA* & *SMAA*). As depicted in Table 5-9 in *baseline* no anti-aliasing method is enabled.

Table 5-9: Configuration set "*Edge*" and their parameter settings.

Configuration	Render Resolution	Antialiasing	Post-Process Antialiasing	Alpha-To-Coverage
baseline	*100%*	*0*	*Disabled*	*Disabled*
SSAA	**200%**	0	Disabled	Disabled
MSAA	100%	**8**	Disabled	Disabled
FXAA	100%	0	**FXAA Ultra**	Disabled
SMAA	100%	0	**SMAA Ultra**	Disabled
AToC	100%	**8**	Disabled	**Grass & Trees**

The configuration *SSAA* (super sampling anti-aliasing) simply renders the view in the doubled output resolution and later resizes it to its original resolution. This method usually produces the best antialiasing results but is also the most demanding. The more efficient *MSAA* selectively samples based on polygon-pixel coverage, thus simple sprites (e.g. tree leaves) are omitted. Sprites are 2D textures that represent complex objects to reduce the otherwise necessary computational effort (a nice introduction is given in (Szijártó & Koloszár, 2003)). To allow *MSAA* to operate on sprites (trees and grass) the alpha channel of the texture is used as aliasing mask telling MSAA to operate on these pixels. This technique is called *Alpha-To-Coverage (AToC)* and is the last parameter evaluated in this group. In *MSAA*, antialiasing is set to eight. *FXAA is* a post-processing antialiasing method using a high pass filter to detect edges followed by a blur. *SMAA* (Jimenez, Echevarria, Sousa, & Gutierrez, 2012) is another morphological post-processing antialiasing method enhancing the edge detection by using multi-sampling instead of blurring. Concerning computational efficiency *SMAA* ranks between *MSAA* and *FXAA*. Both post-processing methods are set to the highest possible

setting for their respective *configurations*. Since *AToC* needs *MSAA* activated, it is set to eight, while alpha-to-coverage is set to Grass & Trees, enabling the multi-sampling of these sprite based objects. Examples can be seen in Figure 5-15. Especially the horizontal beams on the container show enhanced image quality when using antialiasing. Since post-process methods cannot extract the underlying edge information, the actual edge information cannot be restored. Additionally, the wooden board on top of the container clearly demonstrated the effects of the different employed methods.

Figure 5-15: Configuration examples of set *Edge* on image 35 of scene *concrete*. Upper left to lower right: *photo, baseline, SSAA, MSAA, SMAA* and *FXAA*.

As no foliage is visible in dataset *concrete*, the effect of *AToC* is depicted on a detail view of scene *forest* in Figure 5-16. Here it can be seen that the influence of *AToC* is subtle and mainly blurs the edges of the sprite.

Figure 5-16: AToC example on a detail view of scene *forest* image 20. Left to Right: *baseline* and *AToC*.

5.2.2.4 Configuration set “3D-Objects”

This set embraces configurations *objects high*, *no objects* and *modelling errors*. These increase the quality of the objects wire mesh or modify their appearance. In *baseline* the object detail is set to very low, objects are visible and correctly textured (see Table 5-10).

Table 5-10: Configuration set *“3D-Objects”* with parameter settings.

Configuration	Objects detail	Display objects	Objects textures
baseline	*Very Low*	*Yes*	*Normal*
objects high	**Very High**	Yes	Normal
no objects	100%	**No**	Normal
modelling errors	100%	Yes	**Modified**

Figure 5-17 presents the impact of mentioned parameters on the image appearance for scene *house*. The parameter *objects detail* sets the distance thresholds where the engine switches to lower levels of detail (LOD). In VBS3 the same object can be represented by several meshes of different detail, each called a LOD. This shall boost the real-time performance by reducing the detail of distant objects. In configuration *objects high* the objects detail is set to the highest setting, thus displaying the highest quality LODs. The geo-specific buildings modelled for this thesis have a very simple wire mesh and thus only one LOD.

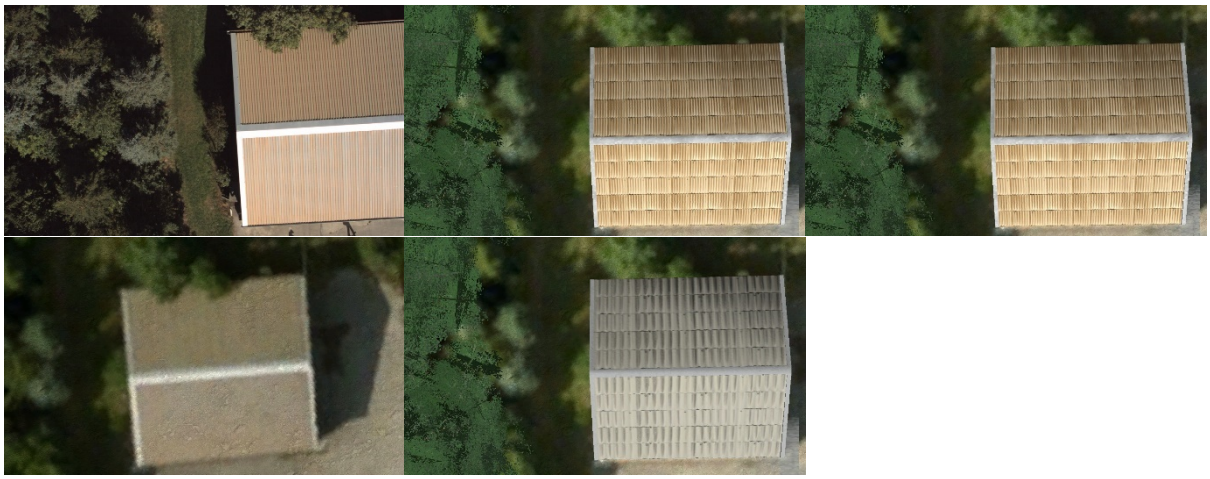


Figure 5-17: Configuration examples of set *3D-Objects* on scene *house* (image 4). Upper left to lower right: *objects high, no objects* and *modelling errors*.

Therefore, effects of this parameter can only be identified on trees and stock 3D models of VBS3. For configuration *no objects*, all 3D entities have been removed from the map leaving only the terrain texture mapped on the terrain mesh to evaluate the effect of 3D-objects. To

investigate the importance of correctly applied geo-specific textures three buildings in scenes *house*, *hangar* and *junkyard* are provided with modified roof textures for the configuration *modelling errors*.

5.2.2.5 Configuration set "Camera Model"

The most prominent specifications (e.g. field of view, resolution) of the camera are fixed to the specifications of the camera used during the test flight (see chapter 4.1) to recreate the same scene accurately. However, the camera model can still be modified by the parameters *Noise Filter, Distortion Filter, Camera aperture, HDR Quality, Bloom* and *Blur*. In *baseline*, all parameters are disabled or set to their default values as listed in Table 5-11.

Table 5-11: configuration set *"Camera model"* with parameter settings.

Configuration	Noise Filter σ	Distortion Filter	Camera aperture	HDR Quality	Bloom	Blur
baseline	*0*	*Disabled*	*39*	*Normal*	*Disabled*	*Disabled*
noise	**3**	Disabled	39	Normal	Disabled	Disabled
distortion	0	**Enabled**	39	Normal	Disabled	Disabled
aperture	0	Disabled	**55**	Normal	Disabled	Disabled
HDR	0	Disabled	39	**Very High**	Disabled	Disabled
bloom	0	Disabled	39	Normal	**Enabled**	Disabled
blur	0	Disabled	39	Normal	Disabled	**1.5**

The noise filter introduces Gaussian colour noise to the image. The employed normal distribution is centred on the original colour value of the pixel and the spread is defined by standard deviation σ, which is set to 3.0 for configuration *noise*. The distortion filter has been implemented by warping the image using the first radial distortion coefficient of the Zhang camera calibration model (Z. Zhang, 1999). The natural images of configuration *photo* are not calibrated. The distortion coefficients have been computed by using a set of natural images of the calibration pattern acquired directly before the flight experiment. In configuration *distortion*, the synthetic image is now warped using the negated first distortion coefficient. Configuration *aperture* closes the simulated aperture reducing the amount of light reaching the sensor. This shall help to identify the influence brightness and contrast changes have on the test object performance. VBS3 is based on DirectX9, which has high dynamic range lighting implemented to enhance the contrast in rendered scenes. The parameter HDR quality is set to very high in configuration *HDR*, which raising the light computation precision to 24bit (16.7 million levels of brightness) instead to the common 8bit (256 levels). This rendering method mainly affects the contrast. Configuration *bloom* simulates oversaturation of an imaging sensor, where an overexposed region overflows in the neighbouring pixels

creating a glowing effect around the region. In *blur* the whole image is low pass filtered using a 3x3 kernel. Examples demonstrating the usage of this and the other discussed parameters can be seen in Figure 5-18.

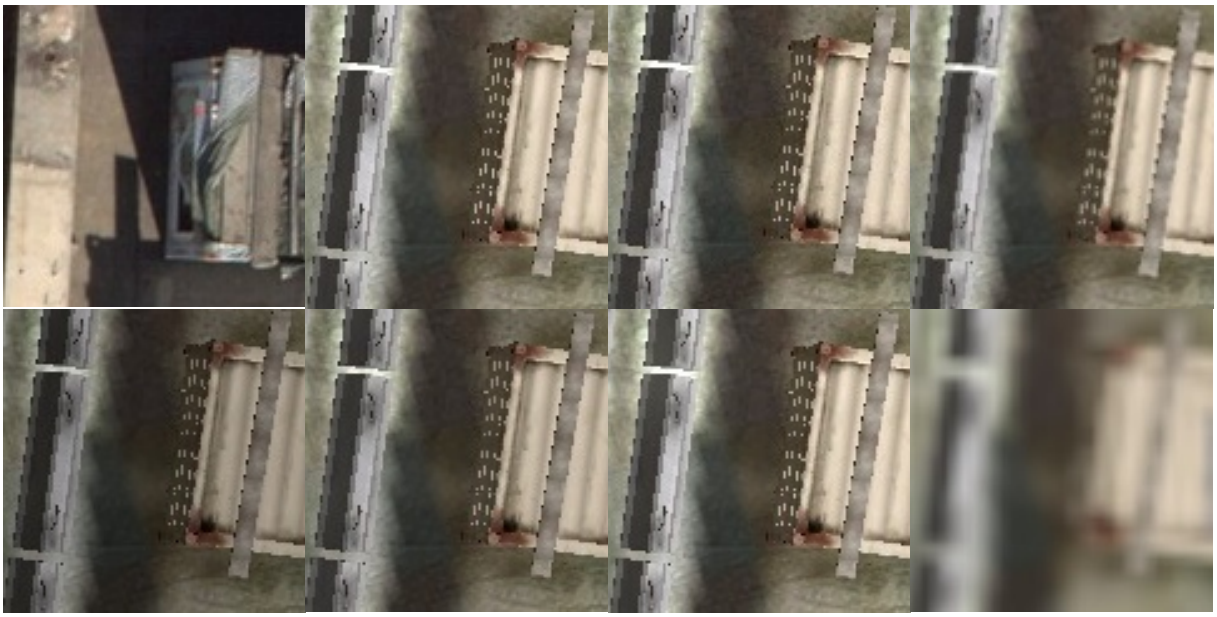


Figure 5-18: Configuration examples of set *Camera Model* on a snippet of image 35 of scene *concrete*. Upper left to lower right: *photo*, *baseline*, *noise*, *distortion*, *aperture*, *HDR*, *bloom* and *blur*.

6 Principle experiments and results

This chapter deals with the experiments conducted to prove the concept presented in chapter 3 and provides respective results. During the principle experiments, natural and synthetic datasets were subjected to the three-step investigation:

- First, the test algorithms **performance differences** on photographs and rendered images were determined.
- In a next step, the **image content differences** between the two datasets were measured.
- Eventually the both results were combined to identify the image content actually **causing** the algorithm performance differences.

Each investigation is followed by a **summary chapter** clearly presenting the acquired results. **A reader mainly interested in results should read can skip the in depth discussion of the three steps and only read the summaries.** The following experiments have been separated into two parts:

- **Baseline experiments:** In the first part (chapter 6.1) the evaluation scheme is only applied on natural data and **synthetic** in their ***baseline*** configuration (see also chapter 5.2.2), which results from selecting the baseline setting for each parameter of the synthetic environment.
- **Configuration set experiments:** In the second part (chapter 6.2), the evaluation scheme is applied using **synthetic** data generated with **varying configurations** of the synthetic environment to identify rendering methods or modelling issues that influence the performance of the CV-algorithm.

6.1 Baseline experiments

The baseline experiments compared the performance of feature detector SIFT, MSER and SURF on the configuration *baseline* and natural reference *photo.* In *baseline,* all parameters of the synthetic environment except high-resolution textures had been disabled as depicted in Table 6-1 (more details can be found in chapter 5.2.2.).

Table 6-1: Baseline Configuration of the synthetic environment

ID	Parameter	Baseline Value	ID	Parameter	Baseline Value
2	Noise filter	0σ (Disabled)	12	Alpha-to-coverage (AToC)	Disabled
3	Distortion filter	Disabled	13	Object detail	Very Low
4	Camera aperture	39 (default)	14	Display objects	No
5	HDR quality	Normal (lowest value)	15	Object Textures (Errors)	Normal
6	Bloom	Disabled	16	Texture Detail	Very High
7	Blur	Disabled	17	Satellite image texture resolution	0.2mpp
8	Render Resolution (SSAA)	100%	18	Anisotropic Filtering (AF)	Disabled
9	Antialiasing(MSAA)	0 (Disabled)	19	Shadow detail	Disabled
10	Post-Process AA (FXAA)	Disabled	20	Shadow filtering	Disabled
11	Post-Process AA (SMAA)	Disabled	21	SSAO	Disabled

The baseline experiments were performed for several reasons:

1. to present the results for the *baseline* configuration, which is used in all configuration set experiments for comparison, thus removing the need to present it repeatedly
2. to demonstrate the actual investigation approach in full detail (statistical evaluation) for one specific example
3. to question and validate the presented approach on this example
4. to select most suitable formats and representation of results
5. to select feature detectors for further experimentation

6.1.1 Object performance

Object performance was evaluated by measuring the *relative* and *absolute repeatability* of three feature detectors on specific scenes for different image types (*photo* and *baseline*). The result graphs are explained together with the presentation of results.

Result representations and illustrations

The timeline plot in Figure 6-1 allows detailed evaluation; here of scene *concrete*. Each diagram column presents the performance results of a dedicated feature detector. The upper row shows the *relative repeatability* between 0% and 100% while the lower row provides the *absolute repeatability* (see chapter 3.4). The performance is acquired by finding features detected in image *n* again in image *n+1* for the complete dataset (35 images).

The percentage of recovered features is called *relative repeatability* while the total number of recovered feature pairs is called *absolute repeatability*. It needs to be highlighted that *absolute repeatability* is not normed and thus results between feature detectors can differ greatly. Therefore, the vertical axis of the *absolute repeatability* graphs is adapted for each graph. The

horizontal axis gives the sequence number of dataset images. The *relative repeatability* graphs show fairly similar performance of all feature detectors on *photo* and *baseline* with MSER differing most. This already shows a surprisingly high compatibility of feature detectors with synthetic data. The SIFT feature detector performs better on synthetic data than on natural images, which means more features pairs have been recovered. In contrast, SURF and MSER perform better on natural images.

A performance oscillation of SURF and MSER appears prominently between image 20 and 24 on the synthetic dataset. For MSER also *photo*'s performance is affected. This variation is also visible, though in minor magnitude in the SIFT diagram. During this period, the aircraft changed directions resulting change of pitch and roll angle by 10° and 5° that induced a strong camera movement reducing the overlap of the images. Regarding *absolute repeatability*, the drop of detected feature pairs is clearly visible. While SIFT features drop to 1000 detected feature pairs during this period, SURF and MSER practically detect no feature pairs. This suggests that *relative repeatability* always needs to be considered together with the absolute measure. For example, a *relative repeatability* of 100% measured for SURF at image 20 and 21 of *baseline* is relativized as in total only one feature pair per image had been found.

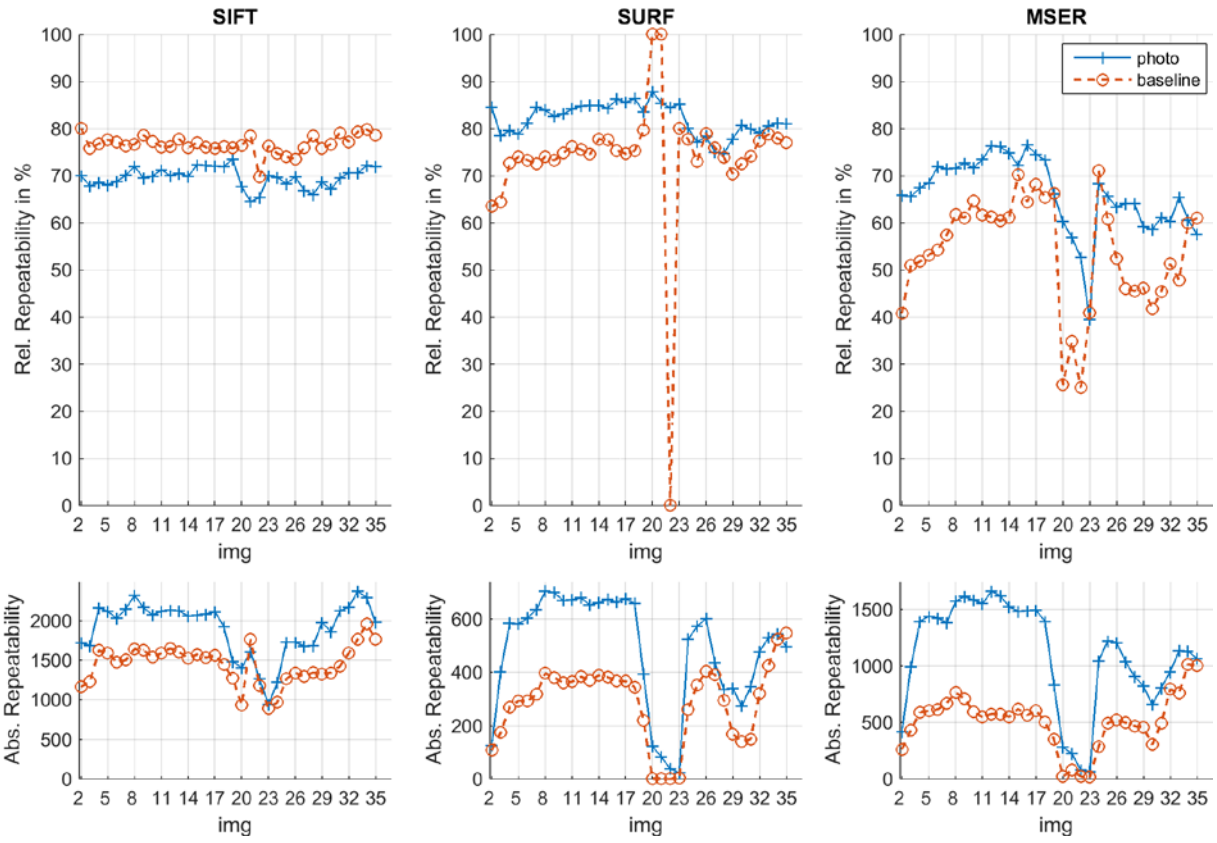


Figure 6-1: Timeline plot: *Relative* and *absolute repeatability* of SIFT, SURF and MSER on scene *concrete*.

While *relative repeatability* between *photo* and synthetic *baseline* datasets shows small differences, the *absolute repeatability* differs considerably. The graph trends seem to correlate for both datasets. The above presented timeline plot is useful to detect anomalies as discussed above to identify potential measurement errors. However, such plots do not allow easy comparison between datasets or detectors. Thus, a more condensed comparable form of presentation is needed for such tasks.

Therefore, the matrix diagrams in Figure 6-2 have been generated. Here, the x-axis presents the results for each scene, while the y-axis lists the configuration. All measures are relative to the reference *photo*, meaning a result of zero demonstrates the most optimal result, which is equal algorithm performance on synthetic and natural datasets. The *relative repeatability* is evaluated by computing the median deviance over all images of one scene to compress results and to reduce the effect of outliers:

$$\Delta Rel.\,Repeatability_{parameter} = median(Rel.\,Rep._{\cdot synth,parameter} - Rel.\,Rep._{\cdot photo}) \quad (36)$$

A **positive** $\Delta Relative\ Repeatability$ indicates the feature detector performing **better on synthetic data** than on natural images. Since *relative repeatability* is already given in percentage, the results are simply subtracted. *Absolute repeatability* is additionally normalised by $Absolute\ Repeatability_{photo}$ to provide these results in percentage:

$$\Delta Abs.\,Rep._{\cdot parameter} = median\left(\frac{(Abs.\,Rep._{\cdot synth,parameter} - Abs.\,Rep._{\cdot photo})}{Abs.\,Rep._{\cdot photo}}\right) \quad (37)$$

In this evaluation, *photo* and its results are always the reference. The measures $\Delta Rel.\,Repeatability_{baseline}$ and $\Delta Abs.\,Repeatability_{baseline}$ provide the information on how a feature detector behaves on *baseline* compared to *photo*.

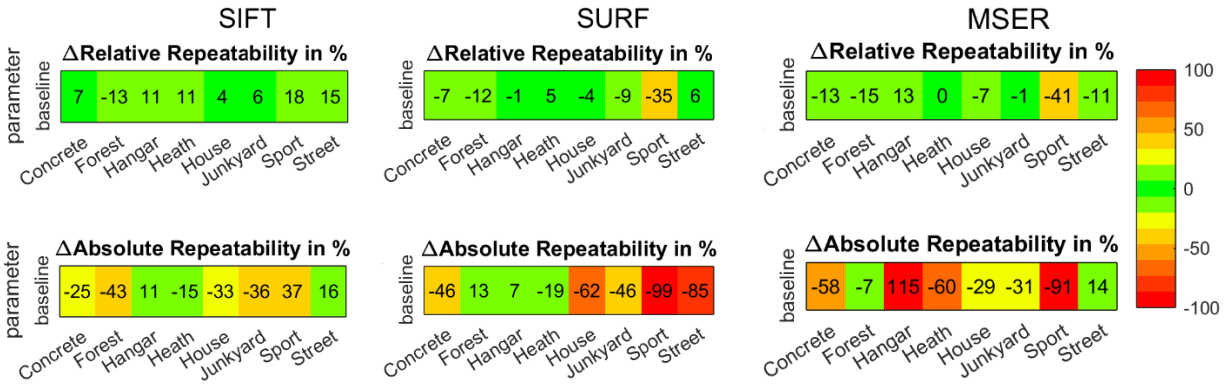


Figure 6-2: Colour coded matrices presenting the performance deviances of the specified feature detector for each scene and parameter in regard to natural data.

Results are given in percent and provided using colour coded lookup tables (see Figure 6-2), which present the results for each scene, parameter and metric in a condensed form. In this diagram, each column presents the results of a dedicated feature detector. The upper row presents $\Delta Relative\ Repeatability$ and the lower $\Delta Absolute\ Repeatability$ for each scene and configuration (only *baseline* in this case). A value zero indicates equal performance on the synthetic dataset and *photo*. If given values are **negative** the algorithm performs **better on natural data** and vice versa.

Starting with $\Delta Relative\ Repeatability$ *baseline* deviates between ±4% to ±18% from the performance of *photo* using detector SIFT. This deviance indicates that **SIFT** can work with synthetic data similar to real images within an interval of **±18%**. The highest difference is measured for scene *sport*. A possible reason is the presence of low contrast textures in this scene due to texture blending. SURF behaves more sensitive to SIFT, deviating more in scene *sport* but less on *hangar, heath* and *house*. Feature detector MSER also performs closely to *photo* for all scenes, except *sport*. *Relative repeatability* of scene *heath* and *junkyard* are almost equal to *photo*. This shows that **all used detectors** can identify previously detected features in synthetic images, in qualities **close to natural data**. Only scene *sport*, deviates assumingly due to the contained low contrast textures (will be determined later on).

The *absolute repeatability*, the actual number of found corresponding feature pairs in subsequent images of a dataset, deviate much more between the two dataset types. All feature detectors detected more features on natural images, indicating more local gradients are present. However, highly cluttered scenes such as *hangar* and *forest* show that this is content related and more accurate modelling can compensate these effects. For instance in scene *hangar* the contrast rich synthetic images lead to the doubled number of detected feature pairs by MSER. Here, reflections in *photo* reduce the visibility and contrast of the hangar roof, while synthetic images provide it in full quality. **SIFT is the least affected detector** with deviation ranging from **±11% to ±43%**. While using *baseline* on SURF performs similar to *photo* for the scenes *forest, hangar* and *heath* (<19%) all other scenes show larger differences.

To investigate the difference between scenes further, the results of each separate scene are depicted in the bar plots of Figure 6-3. The coloured bars show the *relative repeatability* in percentage while the grey bars show the *absolute repeatability* (Hint: not to be confused with *Δabs. repeatability*). Here, *absolute repeatability* is not normed and the vertical axis intervals differ between the different plots and measures.

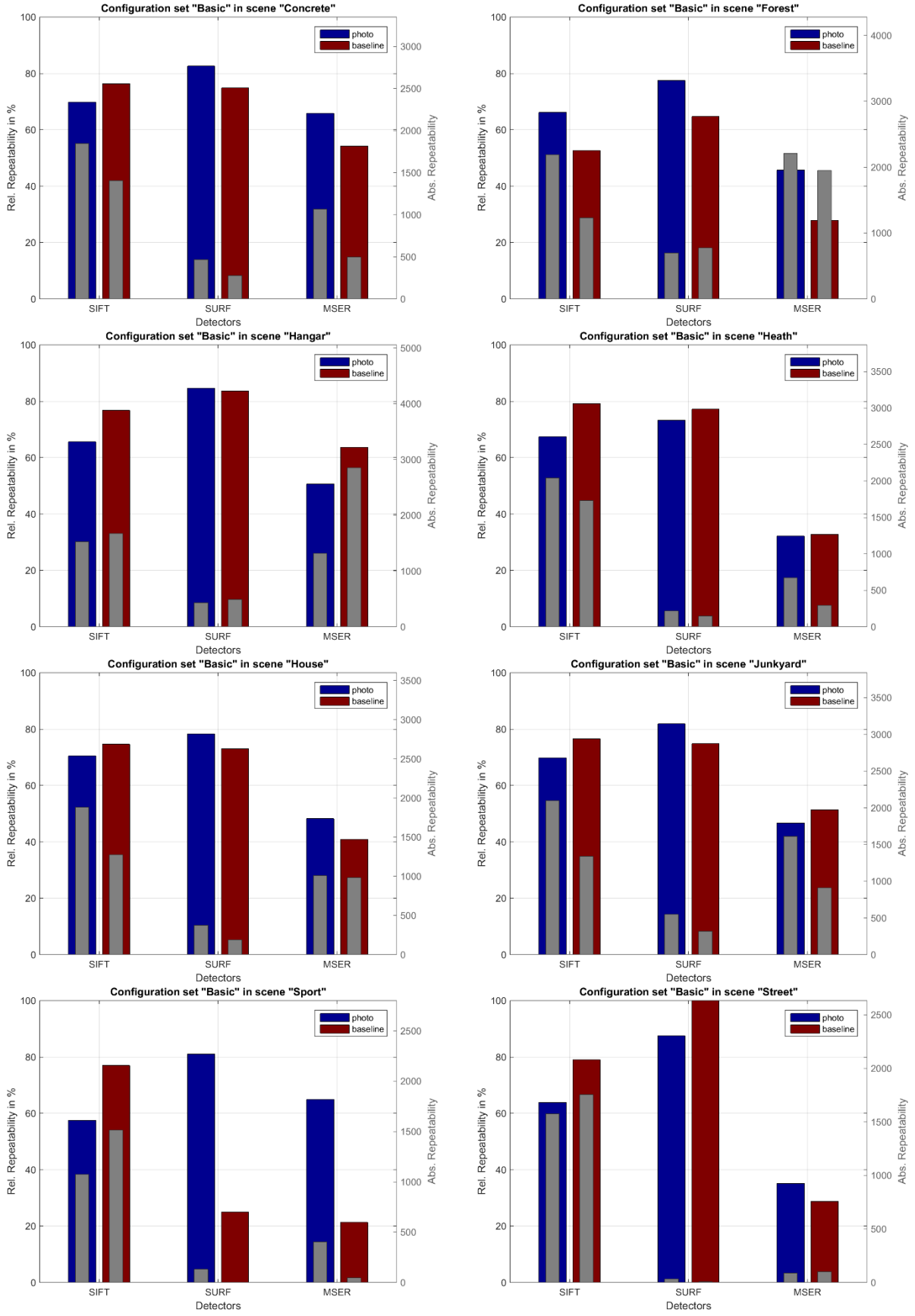


Figure 6-3: *Relative* and *Absolute Repeatability* for all scenes and all evaluated Detectors on *photo* and *baseline*.

SURF is detecting the least amount of feature pairs (*absolute repeatability*) regardless of the scene. For texture heavy scenes such as *sport* and *street* the amount of feature pairs is extremely low resulting in a drastic collapse off the overall feature detection rate as can be seen by the (almost) non-existent grey bars in these two diagrams. This is strange, considering the performance of the methodically quite similar feature detector SIFT. After investigation, it was identified that the MATLAB implementation uses an unusual default configuration of SURF (1000 as determinant threshold of the Hessian matrix, instead of 600 as used in the original paper (Bay et al., 2006)) that early omits features. Since the idea was to compare detectors in their default configuration and the used SURF detector deviates from this configuration, its results should be considered with care. **SURF will be omitted** in the following experiments in chapter 6.2, due to timely limitations and its close relation to SIFT. Testing the performance of other feature detectors including SURF can be of interest in future investigations.

When comparing the detectors performances, SIFT is the most robust followed by MSER. SURF works considerably well whenever it detects enough feature pairs, which is unfortunately often not the case.

Sometimes MSER also detects only a small number of feature pairs, due to its design. In general, the number of found feature pairs is sufficient. MSER has trouble with scenes mainly depicting textures such as *sport, street* and *heath*.

Overall, the best correlation on *relative repeatability* for all detectors is achieved on scene *hangar* and *concrete*. *Absolute repeatability* on the other hand is very scene specific, performing close to *photo* for scene *hangar*, *forest* or *street* depending on the used detector. Especially interesting findings in scenes *forest, hangar* and *sport* are now discussed in detail using timeline plots.

Detailed Analysis of scene *forest*

In Figure 6-4 the line plot of scene *forest* is given. All detectors perform lower on synthetic data for the first 17 to 20 images and then close in on equal performance.

The first half of the dataset shows images only depicting treetops, while the later also depicts some ground texture. Since this effect can be seen with all detectors, it can be concluded that the synthetic dataset violates the plane surface condition of the homography-based ground

truth generation. The detectors are also reacting more drastically to changes in synthetic images. Interestingly, when reviewing the absolute repeatability of SURF and MSER a high correlation between *photo* and *baseline* can be seen (trend and absolute numbers).

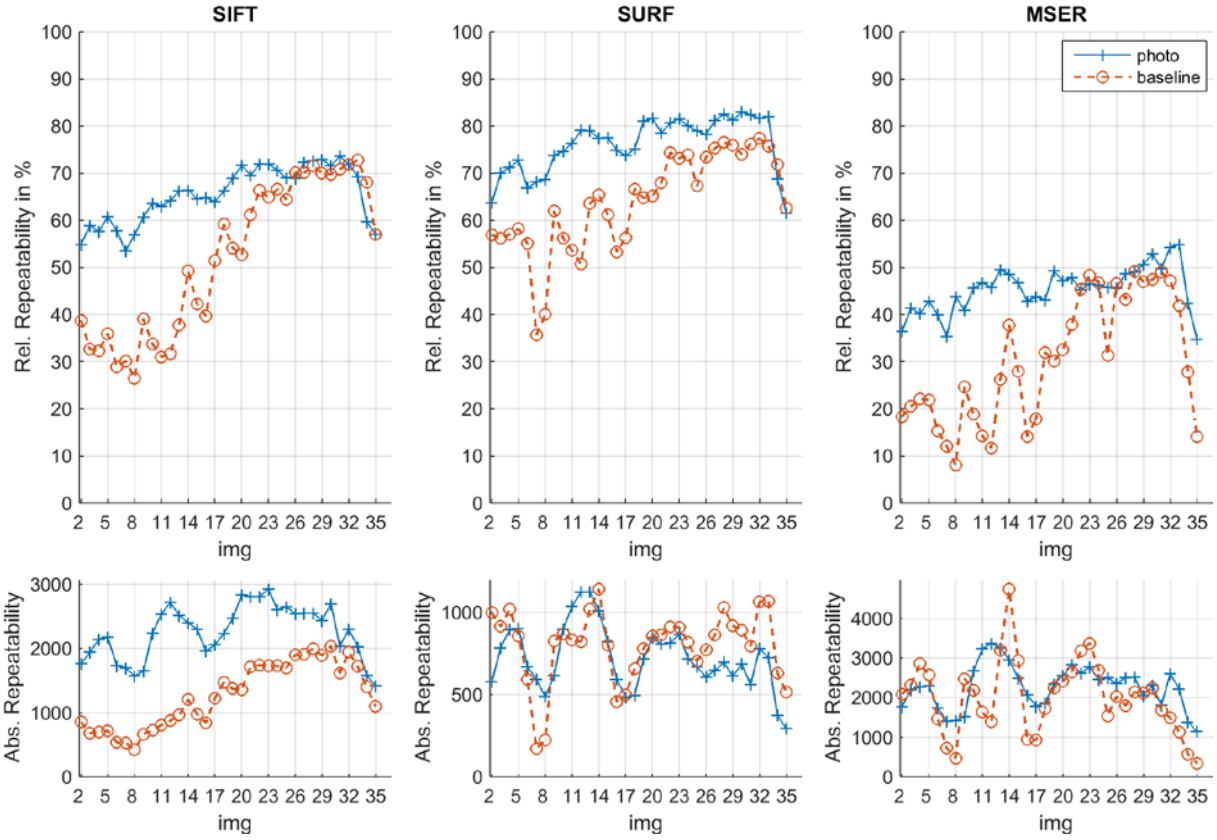


Figure 6-4: *Relative* and *absolute repeatability* of SIFT, SURF and MSER on scene *forest*.

Detailed analysis of scene *hangar*

In scene *hangar* SURF performs almost equally on *baseline* and *photo* according to *relative repeatability* (see Figure 6-5). In addition, absolute repeatability runs close though more reactive to changes in the images. When looking at the *relative repeatability* of SIFT, it starts equally at 70% but then reacts inversely on images of *photo* and *baseline*. The *absolute repeatability* is similar in numbers but the trends found in *photo* are not replicated in *baseline*. MSER achieves the lowest *relative repeatability* for natural and synthetic imagery. However, it also detects the most features of all detectors as can be seen in the diagram.

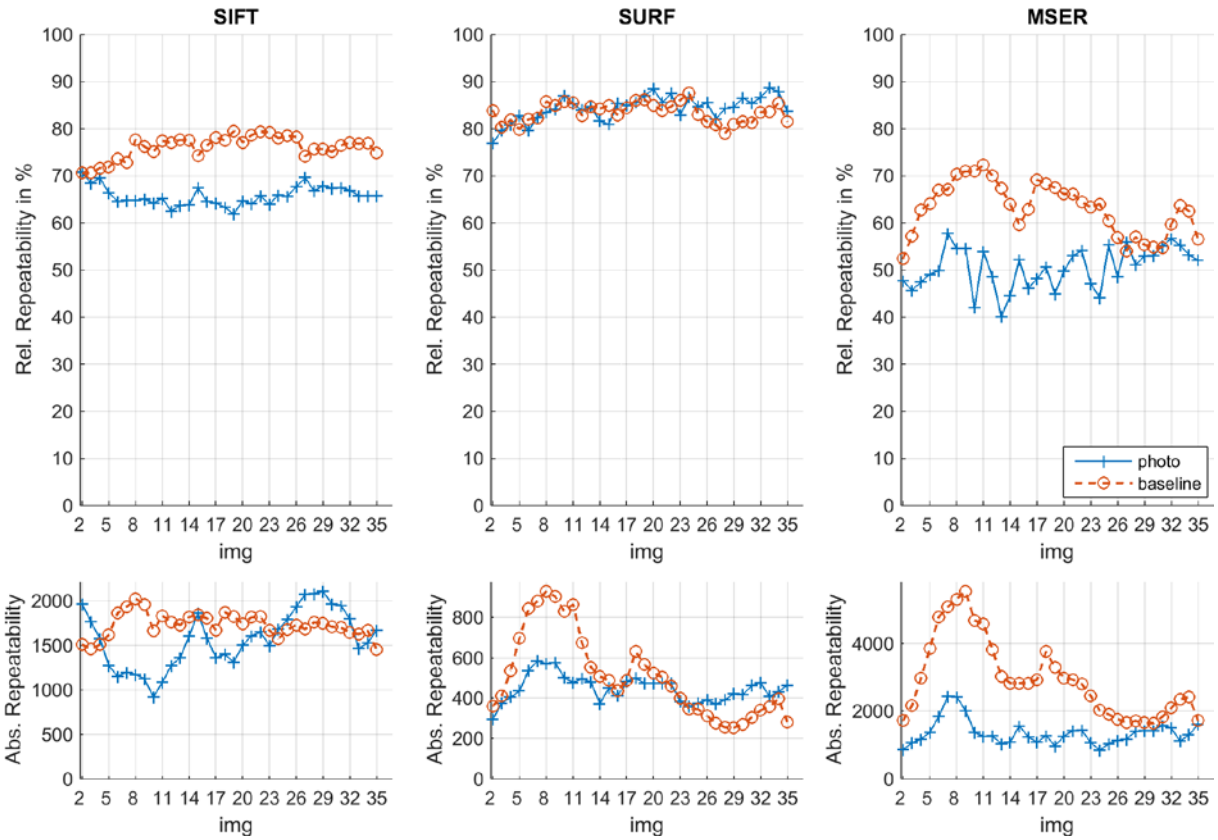


Figure 6-5: *Relative* and *absolute repeatability* of SIFT, SURF and MSER on scene *hangar*.

Detailed analysis of scene *sport*

In scene *sport* results for SIFT are most robust, while MSER and SURF are unable to process the synthetic imagery as depicted in Figure 6-6. This scene is mostly based on textures (no 3D-objects), which are of low contrast due to the texture blending mechanism of VBS3. This seems to be the reason for MSER to perform worse on the synthetic data. SURF and MSER have trouble detecting features at all. Thus, *relative performance* do not present the detection robustness in this case (*relative repeatability* jumps between 100% and no detection 0%). While SURF and MSER hardly detected features on synthetic images, SIFT performs about 20% better. This shows that detection algorithms can react very differently to the same data solely based on the inherent principle on how to detect features. Since the contrast of textures in the synthetic dataset is low, the feature detection approach of MSER cannot detect the necessary robust features. The low detection rate of SURF on low contrast synthetic data can be read clearly from its absolute repeatability diagram.

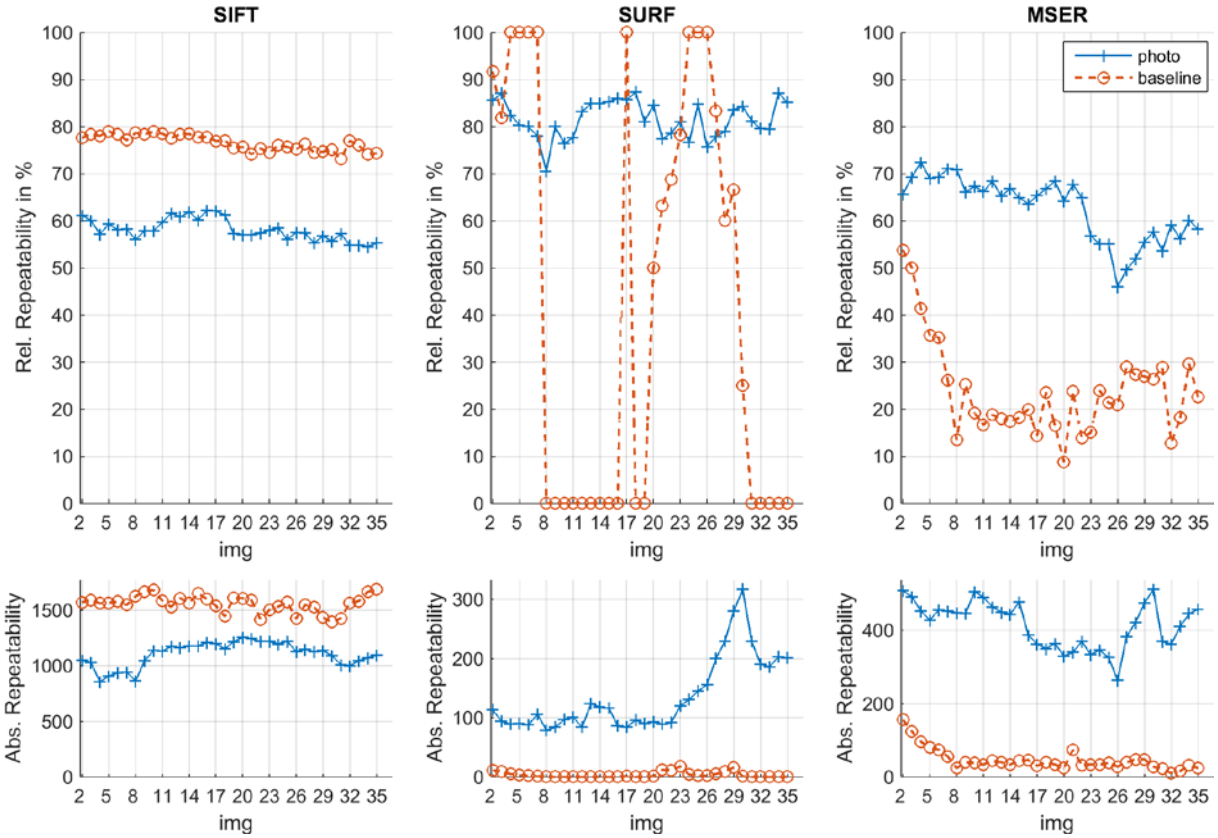


Figure 6-6: *Relative* and *absolute repeatability* of SIFT, SURF and MSER on scene *sport*.

Summary

This chapter presented three illustrations to depict object performance results, where timeline plots are used for detailed analysis and matrix diagrams for result overviews and comparison purposes. Further only matrix diagrams are used to present these results, while timeline plots were used during data acquisition validate the data.

This step of the concept presented that it is a valid method to evaluate CV-algorithm performance. SIFT, SURF and MSER show similar *relative repeatability* trends for all detectors on natural and synthetic images for most Scenes. Scenes depicting mainly low contrast textures in the synthetic dataset, such as *sport*, lead to poor detector performances for SURF and MSER. *Absolute repeatability* can differ quite strongly due to more details in natural images.

SURF is not used for further investigation due to its parametrization and close relation to SIFT.

6.1.2 Image content distances

This evaluation measures the visual differences in image content based on the image descriptors presented in chapter 2.5. After their validation in chapter 5.1.4 the image content measures CLD, CSD, DCD, SCD, HTD and EHD presented themselves as capable measures.

The previous chapter identified the varying performance of feature detectors on the datasets *photo* and *baseline*. Afterwards, to quantify the differing image properties between the two data types, the image content difference was objectively measured. In Figure 6-7, the statistically summarized distance measures of all image descriptors for all scenes are provided. The results vary strongly for each descriptor, indicating their sensitivity to different content. In addition, the measures vary for each scene indicating that content differences are mostly scene dependent. Scene *street* has the highest similarity between natural and rendered images according to the descriptors CSD, CLD, SCD and EHD. The most differing scene is *hangar* as depicted by CLD, SCD and HTD.

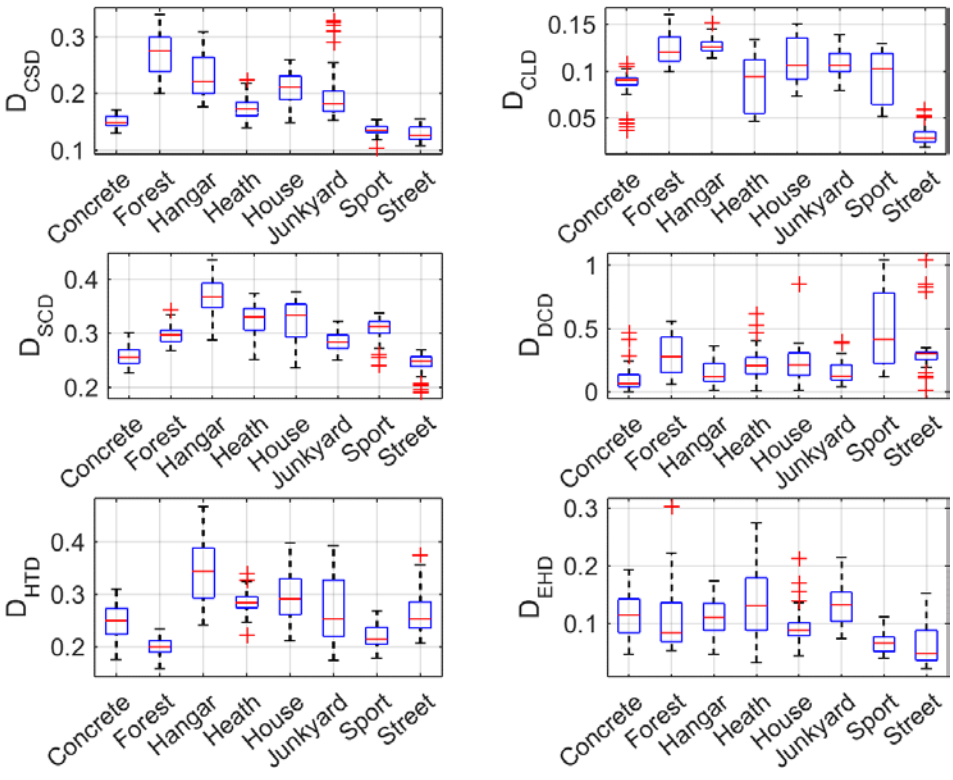


Figure 6-7: Boxplots for each image content measure presenting the varying distances between photographs and rendered imagery for each scene.

Colour structure (CSD) is highly similar for *concrete, sport* and *street* and most different for *forest, hangar* and *house*. This implicates modelling errors in regard to dominant colour tones or placement errors in these last named scenes.

The *colour layout* (CLD; spread of colour among the image) is changing marginal between most scenes when also including the variance. Only *forest, hangar* and *street* show large deviations from the other scenes. In *forest*, the colour distribution in the synthetic images is larger due to the placement of generic trees (some had bright green and yellow leaves) while the trees in natural images had a quite homogenous lush green. Similar, the roof in *hangar* is a dense colour patch that is not as present in the natural image due to sunlight reflections. Since *street* is dominated by the blend of satellite image and detail texture, the differences here are especially low for the descriptor CLD.

When reviewing the results of SCD for *hangar, heath* and *house* the colour histogram of synthetic images differs strongly from natural images. Indicating modelling errors in selection of the correct colour tones.

The *dominant colour descriptor* (DCD) provides the distance between the five most dominant colours and their percentages. Here, *sport* varies strongly, most probably due to the texture of the tartan track (wrong colour tone). In addition, the use of multi-coloured trees in scene *forest* increased the DCD distance.

The HTD descriptor segments the spatial frequency image into 30 segments and describes them by the amount of signal presence. Thus, measured frequencies appear repeatedly in the image. *Forest* achieved the lowest HTD distance showing that the geo-typical trees replicate the “busy” appearance of dense leave trees well. In *hangar, heath, house* and *junkyard* the homogenous frequencies differ more strongly. A possible reason may be the prominently visible roofs in scenes *hangar, house* and *junkyard*. These may have not the same spatial texture (e.g. tiling density) according to these measurements. A basic distance in HTD is always given due to the reoccurring detail texture not present in natural images.

The EHD distances are similar for almost all scenes; only *sport* and *street* are closer to their natural counterpart. This can be explained by the existence of edges in the satellite image that are very similar to the natural images despite blending and the missing objects which would naturally lead to sharp edges in synthetic data.

Thus, feature detectors might perform different on specific scenes due to their differences in specific image content. The following analysis will now try to combine these findings to identify possible relations between performance changes and image appearances.

6.1.3 Influence factor analysis

The *influence factor analysis* (chapter 3.7) aims to estimate the effect of image content distances on the algorithm performance differences by computing and interpreting a *stepwise backward multiple regression analysis*. Thus, this analysis shall identify image content causing performance shifts of the tested algorithm. Thus as outcome variable is the Δ*relative* or Δ*absolute repeatability* of the tested feature detector (chapter 6.1.1) is used and MPEG7 *image content descriptor* distances (chapter 6.1.2) as predictors to fit the regression model. The experiment is based on the datasets *photo* (natural reference dataset) and *baseline* (synthetic default dataset).

The iterative approach reveals image content differences affecting the performance of the tested feature detector as these explain remaining performance differences using *photo* and *baseline* (and remain within the model).

For each repeatability measure, feature detector and scene, a regression model is generated (16 models). Additionally, all investigated configurations of the synthetic environment are considered during the fitting of each model. However, to ease the explanation of regression analysis **only** the results of ***photo* and *baseline* are presented in this chapter**. The discussion of rendering parameters can be found in chapter 6.2. Now, all steps performed for every regression analysis in this thesis are detailed on the example of scene *concrete*.

Regression analysis in detail on scene *concrete*

Because high collinearity among predictors may to unreliable determination of predictor coefficients, this needs to be excluded first.

Handling multicollinearity

The analysis has been performed in MATLAB. If collinearity is detected, one (causing) measure was removed. This could be tolerated since a similar behaving measure remains.

Multicollinearity between image content distance measures was analysed by combining the $x_1, \ldots, x_p$ measures of all predictors on all images of a scene in a Matrix X of dimension $(N-1) \times P$ with N being the number of images and P the number of image descriptors. The *condition index* (CI; detailed in chapter 3.7) expresses the amount of collinearity. It was obtained by performing a *singular value decomposition* (SVD) of matrix X (for more detail

see (Belsley, 1991)). A condition index of 100 has been selected as threshold to represent problematic high collinearity. Additionally, the *variance decomposition proportions* (VDP) are provided. These indicate the proportion of variance shared between descriptors and the decomposition component, which is one matrix dimension (described by the CI).

Figure 6-8 presents the results of the multicollinearity evaluation using a *tableplot* (Friendly & Kwan, 2009). In this diagram, the vertical axis presents the decomposition components and sorts them according to the condition index (most critical component is in the upmost row).

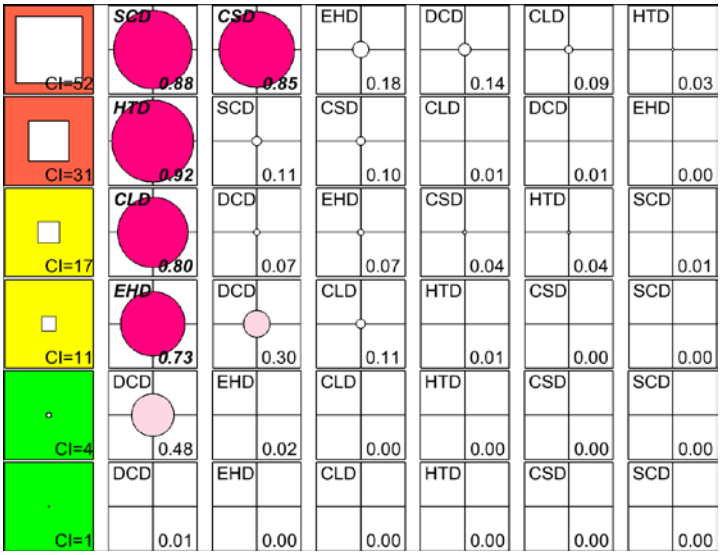


Figure 6-8: *Tableplot* presenting collinearity of image content descriptor distances between *photo* and *baseline* acquired for scene *concrete* combined.

The severity levels of the condition index are colour coded. Strong collinearity is indicated in red (CI>100), followed by moderate in orange (CI>30), light in yellow (CI>10) and very light to none in green. Additionally, the CI is also provided by the size of the white rectangle in the colour coded boxes. For each decomposition component, the VDP for every descriptor are provided in white boxes named after the represented image descriptor. The closer the number is to one the more collinearity is shared between the descriptor and the decomposition component. Thus, a conflict only occurs when at least two descriptors correlate with more than 50% of their variance (0.50) with the component. The descriptors are sorted from the highest shared variance on the left to the lowest on the right. The VDP of each descriptor is provided as numerical value and the radius of the circle. The criticality threshold are provided by the intensity of the circles colour fill (VDP>0.5 = full pink, VDP>0.3 = light pink and white for values below). Thus, a *tableplot* is best read row by row from top to bottom starting on the upper left corner. The example in Figure 6-8 shows two moderate condition indexes, one between SCD and CSD and one on HTD only. The second decomposition component

(row) simply shows strong relation between HTD and the component and no collinearity. The first row shows a relation (collinearity) between SCD and CSD of moderate level. Since the *Condition Indexes* are below 100 there is no need to act, otherwise it would have been necessary to remove one of the (redundant) measures.

Analysing the fitting quality

After assuring that collinearity lies within acceptable limits, the regression analysis was performed. In a next step, the fitting quality was investigated by analysing the data for existing outliers using the *Cook's distance*, which is the originally measured value minus the model predicted result (Cook & Weisberg, 1982). Figure 6-9 shows the *Cook's distance* of SIFT's *Δrelative repeatability* on scene *concrete* for each image (displayed as row number) as a red x. The dotted line presents the recommended threshold value (three times the mean distance). However, according to (Field, 2009) a data point should only be considered an outlier if the *Cook's distance* is equal or larger than one. In this case, all measures were within this boundary, otherwise data points exceeding this value would have to be removed from the data and the model fit recomputed.

Figure 6-9: *Cook's distance* of the Regression model of *SIFT*'s *Δrelative repeatability* on scene *concrete*. Recommend threshold is given as dotted line. However, only data points exceeding value 1 are outliers.

Thereafter, the residuals (differences between measured and via model predicted *y*-values) were investigated in to discover whether the model can be generalized or not. If not the regression model only would be valid for the used scene and results cannot be transferred to similar scenes.

The *histogram of residuals* in Figure 6-10 shows their quantified distributions as bars and a normal distribution function with the same variance in black as comparison. The figure shows that the data points follow a normal distribution, which is a necessary condition to allow generalization of the regression model.

The next plot is the *normal probability plot of residuals* or *P-P plot*, which presents deviations from normality (Field, 2009). The diagonal line represents the normal distribution; each x presents one measurement (image). The plot shows a strong correlation to normality with slight deviations on both ends. These minor alignments however do not violate the condition for residuals to be normally distributed.

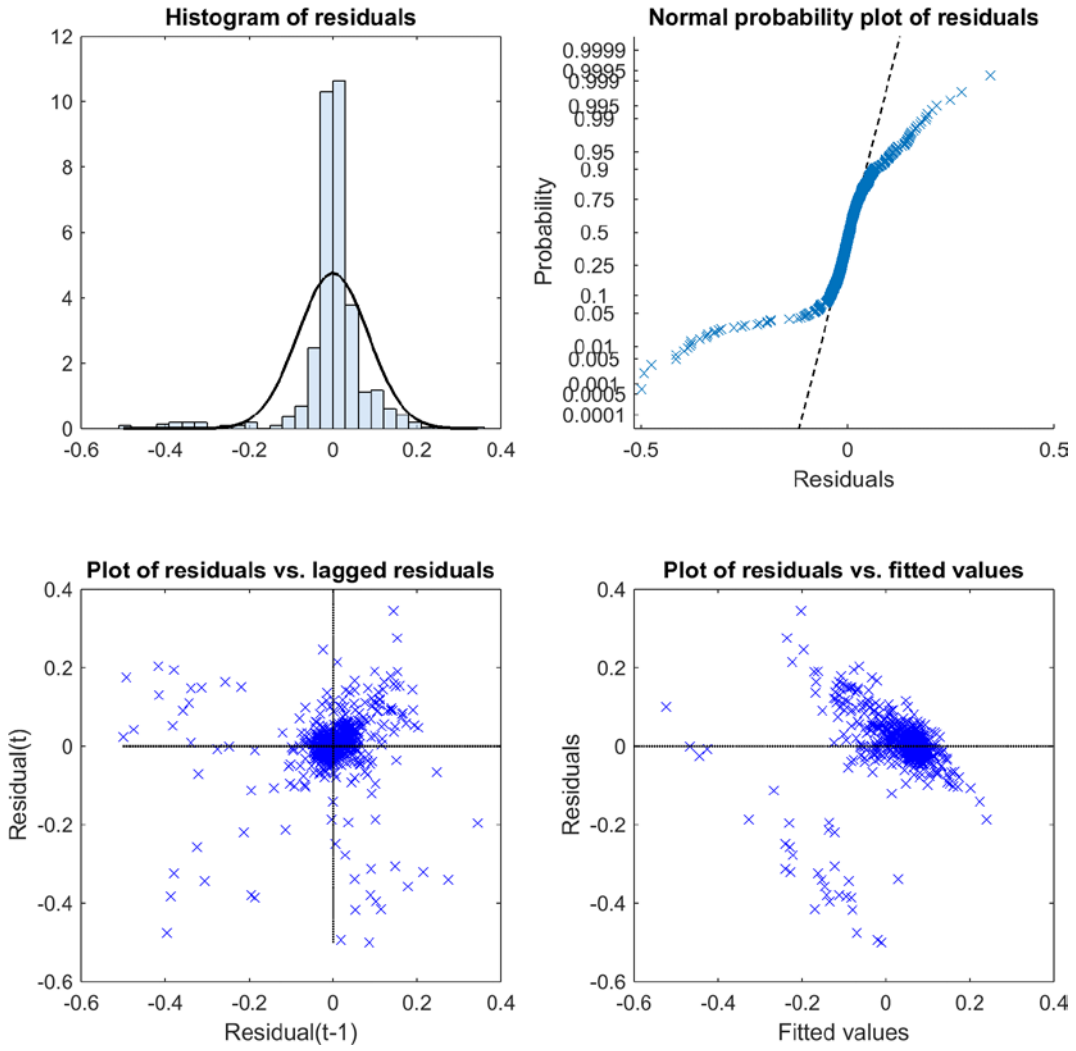


Figure 6-10: Residual plots to identify errors, outliers and correlations in model or data on scene *concrete* for *Δrelative repeatability* of SIFT.

The *plot of residuals vs. lagged residuals* is a scatter plot that investigates whether errors are independent or correlated. In this plot, the residual of the current data point is the vertical

coordinate while the horizontal axis describes the residual of the previous data point. If there is no correlation, the crosses are randomly distributed. In case correlation exists, a dominant direction could be seen. The plot shows that residuals are dominantly positive, but no correlation between subsequent data points exists.

The *plot of residuals vs. fitted values* displays the fitted values (the *y*-values; the possible range of *y*) on the horizontal-axis and appearing residuals on the vertical-axis. Now, when residuals are scattered non-uniformly along *y* the assumption of constant standard deviation of random errors is not given. The plot presents that most residuals are fitted to values of y around 0.05 with the distribution showing no correlation to the fitted values. The *relative repeatability* results of *photo* have been subtracted from *baseline* results to compute *Δrelative repeatability*. Therefore, a negative fitted value indicate the relative repeatability on synthetic imagery was higher than on natural imagery. Thus, constant standard deviation is given and the conditions for generalization of the model are met. The mentioned evaluations are performed for all measurements but will further not be presented. Scenes violating the named conditions will not be considered in the following chapters. In summary, the generalization of the model is possible while considering limitations for large residuals (results that do not fit well).

Δrelative repeatability regression models

Finally, after excluding outliers and validating that generalization of the resulting models is possible, the regression models can be discussed. In Table 6-2 the two regression models of *Δrelative repeatability for feature detector SIFT and MSER on scene concrete* are presented (SURF has been dropped due to reasons explained in chapter 6.1.1). To be remembered, these regression models describes the relationship between image differences identified using image content descriptors and the difference of repeatability (repeated detection of same features) of a specific feature detector compared between photographs and computer generated imagery of the same scene. First, the model quality is presented by the measures R^2, $Adj\text{-}R^2$ and *F-ratio* (explained in chapter 3.7).

Now, the model for SIFT can be discussed. The R^2 value explains that the model represents 48% of the total variance of *Δrelative repeatability*. Thus, the remaining 52% variance are dependent to causes not covered by the predictors used in this evaluation. The F-ratio expresses how much variance is explained be the model divided by how much remains in the

residuals. A value of four shows a good fit and that it is significantly different compared to mean. The regression model of MSER fits the data better with an F-ratio of six and a R^2 of 56%. In this case, the model explains almost 60% of all existing variance.

Table 6-2: Regression models of *Δrelative repeatability* for SIFT and MSER on scene *concrete*.

Model	Model Fit		Coefficients	Value	SE	β
Δrel. Repeatability	R^2	48%	Intercept	0.241 Ns	0.308	**0.00**
SIFT	Adj-R^2	37%	HTD	-0.034 Ns	0.092	**-0.08**
concrete	F-Ratio	4***	EHD	-0.065 Ns	0.182	**-0.07**
			SCD	-0.050 Ns	0.191	**-0.07**
			DCD	0.002 Ns	0.053	**0.01**
			CLD	-0.004 Ns	0.489	**-0.00**
Model	**Model Fit**		**Coefficients**	**Value**	**SE**	**β**
Δrel. Repeatability	R^2	56%	Intercept	-1.162 Ns	0.640	**0.00**
MSER	Adj-R^2	47%	SCD	0.257 Ns	0.242	**0.26**
concrete	F-Ratio	6***	HTD	0.138 Ns	0.140	**0.22**
			CSD	0.486 Ns	0.834	**0.15**
			CLD	0.244 Ns	0.619	**0.06**
			EHD	0.031 Ns	0.232	**0.02**

Ns = not significant (p > .05), *p < .05, ** p < .01, *** p < .001

The right half of Table 6-2 lists the statistics of the coefficients populating the model. Most image content differences influence the outcome, however not to a significant measureable degree. Putting these into the linear regression equation together with the coefficient values results in following equation:

$$\Delta rel.\ rep. = 0.241 - 0.034\ HTD - 0.065\ EHD - 0.05\ SCD + 0.002\ DCD - 0.004\ CLD \tag{38}$$

The *Intercept* of both models is considerably large pointing towards additional non-identified influences. The *standard error* (SE) indicates how much the coefficient can change with different samples of the population. The most interesting measure β is computed by multiplying the normed coefficient value with the standard deviation of the predictor measurements $\sigma(x)$ and divided by the standard deviation of the outcome measurements $\sigma(y)$:

$$\beta_x = \frac{\|coefficient_x\| * \sigma(x)}{\sigma(y)} \tag{39}$$

The **β-values** are given in standard deviation units making them directly comparable. It provides the amount the outcome will change (in standard deviations of the outcome) when the currently discussed predictor changes by one standard deviation. Thus, it indicates the

predictors influence on the outcome**.** The standardization makes these values comparable with coefficients of other models.

In Table 6-2, the coefficients are sorted by their β-values from top to bottom (except for the intercept, which is always the first). In model *SIFT ⊿relative repeatability,* the *homogenous texture* distances (HTD) influence the outcome by -0.08σ when changed by 1σ. In *MSER's ⊿relative repeatability,* the colour distribution (SCD) is most influential with -0.26σ followed by homogeneous textures (HTD) with -0.22σ. The **coefficients sign** indicates on which **dataset type** the feature detector **is performing better**. A **positive** coefficient shows the descriptor to performs **better** on the **synthetic dataset**.

The regression analysis ranks predictors by their effect on the outcome. It should be noted that a **predictor is significant** when the **behaviour** of its values is **well represented** by the model (small residuals). Thus, **a non-significant predictor can be influential** but its behaviour could not be fitted to the linear model.

Further, it can be extracted from the results on which datatype the feature detector performs better and which image property (measured by an image content descriptor) is causing the remaining performance difference. For scene *concrete* mainly the differences in dominant colours are responsible for the remaining difference in algorithm performance (only positive coefficient).

⊿absolute repeatability regression models

The models fitted for *⊿absolute repeatability* of SIFT and MSER on scene *concrete* are presented in Table 6-3. To be remembered, the relative measure shows the performance of the feature detector to detect the same features repeatedly, while the absolute measure presents the ability of the feature detector to find a certain amount of valid feature pairs. Therefore, the *⊿absolute repeatability* model should explains why the detector detects more (or less) features on each of the two dataset types.

The SIFT model covers 75% of all variance in the images used and 70% of all variance for the population. The F-ratio of 14 shows that the model improves the prediction in comparison to the remaining residuals and that the chance of the null hypothesis being true is slim ($p < .001$). The regression model uses all descriptors with CLD, SCD and DCD being the most influential followed by EHD, CSD and HTD. Coefficients of $\beta < .1$ are considered as

not influential. Interestingly only CLD, DCD and CSD are significant and thus well represented by the linear model. In case the distance of all predictors would be zero (natural and synthetic images are identical in the eyes of the descriptors) the intercept of -204 feature pairs points towards a static difference not covered by any image content descriptor. With increasing distance of predictor CLD, more feature pairs are found in natural images compared to synthetic ones. On the other hand, expanding SCD distances lead to more detected features in synthetic images compared to their natural counterparts. CLD is the most influential image descriptor with 0.39σ followed by SCD and DCD. As can be seen in chapter 6.1.1 SIFT finds less features on synthetic data than on natural data. According to the model, the best way to increase the feature detection is to lower the colour layout descriptor distance, which means to review the dataset for local areas of differing colour and to remodelling them (e.g. by adding an additional 3D-object of specific colour to the scene).

Table 6-3: Regression Model of *Δabsolute repeatability* for SIFT and MSER on scene *concrete*.

Model	Model Fit		Coefficients	Value	SE	β
Δabs. Repeatability	R^2	75%	Intercept	-204 Ns	776	**-0.00**
SIFT	Adj-R^2	70%	CLD	-4315***	1183	**-0.39**
concrete	F-Ratio	14***	SCD	714 Ns	461	**0.28**
			DCD	-257*	129	**-0.28**
			EHD	-417 Ns	448	**-0.13**
			CSD	-820*	343	**-0.10**
			HTD	94 Ns	224	**0.06**
Model	**Model Fit**		**Coefficients**	**Value**	**SE**	**B**
Δabs. Repeatability	R^2	75%	Intercept	-534 Ns	1084	**0.00**
MSER	Adj-R^2	70%	CLD	-6032***	1099	**-0.56**
concrete	F-Ratio	17***	EHD	-1141**	406	**-0.36**
			SCD	573 Ns	414	**0.23**
			DCD	180***	25	**0.20**
			CSD	880 Ns	1285	**0.11**

Ns = not significant (p > .05), *p < .05, ** p < .01, *** p < .001

The R^2 and F-ratios of the regression model for MSER shows a good and highly significant fit with 75% covered outcome variance using the five predictors CLD, EHD, SCD, DCD and CSD (in their order of influence). Only CLD, EHD and DCD distances are of significant nature. Since MSER also detects less features on synthetic data, it is necessary to increase this amount to minify the performance difference. This can be achieved by reducing local colour differences and improving the edge representation to align more closely with the real scene. While SCD and DCD are influential, a reduction in distance would lead to less detected features on synthetic data and a non-desired increase in performance differences.

This concludes the detailed example discussion of the regression analysis results for one scene. The remaining scenes will now be presented with less detail.

Presentation of all other regression models

After presenting the detailed result discussion and the steps necessary to acquire valid results, now the other scenes are discussed in a more compacts form, which will be used for the remaining experiments.

Measuring multicollinearity for all scenes

Again, collinearity was measured using the *condition index*. Detail views of all *tableplots* are depicted in Figure 6-11 to display possible collinearity in each individual scene.

Figure 6-11: Detail view of *tableplots* measuring collinearity for each scene.

All scenes show collinearity of medium severity, which is considered acceptable. The CI of 98 for scene *heath* is close to the threshold of 100 and may lead to untrustworthy predictor coefficients when fitting the regression model. Therefore, results of *heath* should be observed carefully.

For each combination of scene, repeatability measure and feature detector a regression model has been fitted, leading to 16 regression models. Again, data was tested for outliers using *Cook's distance*. Further, the normal distribution of residuals was checked via the earlier presented residual histograms and P-P plots.

Δrelative repeatability regression models for SIFT

Each model fit including coefficients and probabilities is summarized in Table 6-4 for *Δrelative repeatability* and in Table 6-5 for *Δabsolute repeatability* of SIFT. To conserve space the full model descriptions are presented in appendix C.3.

Each row presents a fitted model for the specific scene. The first six columns give the standardized coefficients β of each descriptor in the model, followed by the general model statistics R^2 and F-ratio. As the absolute goal is to measure equal performance on both dataset types, the measured performance of the feature detector is of importance and given in the last row (for details see chapter 6.1.1). The colour code indicates on which dataset the detector performs higher (orange = synthetic data; blue = natural data). As explained, a **positive *Δrelative repeatability*** (or *Δabsolute repeatability*) exhibits a feature detector to perform **better on synthetic data** compared to *photo*. To reduce this difference the *image content distance* between the two dataset types for a specific *should only be decreased*. Otherwise, increasing a content distance to gain a smaller performance difference would suggest that the current level of similarity is too good for the feature detector. Thus,

- when lowering a distance measure attributed with a **positive** *standardized β-coefficient* the performance of SIFT on synthetic data is **lowered** (compared to photo).
- when lowering a distance measure attributed with a **negative** *standardized β-coefficient* the feature detector performance on synthetic data is increased.

In conclusion, the influence of a predictor β given by the model allows the determination of

- predictors affecting the algorithms performance ($\beta > .1$).

- image content distances that should be lowered to gain equal performance (sign of β).
- their order of effect (size of β).
- whether the model represents the behaviour of the predictor ($p < .05$)).

Now each model except *concrete* (already presented above) can be discussed. All scenes could be fitted while covering medium *(~40-50%)* to high variance *(>70%)* of the outcome.

Scene *forest* is affected by CSD, EHD, CLD and SCD distances. EHD and SCD measures are contained in the model even though they are non-significant because the presented models are the default for the following configuration set experiments. Whenever any configuration significantly affects the influence of an image property on algorithms performance, it is listed in the *baseline* model as well (because it is the *default* model). Here, the coefficients of the baseline models are discussed whenever they are significant or their effect is notable. According to *Δrel. Repeatability* (-13%) SIFT performs worse on synthetic data in this scene, which could be corrected by reducing the CSD, CLD, EHD and SCD distances (coefficients with negative sign) and be achieved by adaptation of image colours, contrast and remodelling of 3D-models for trees.

Table 6-4: Regression models of *Δrelative repeatability* SIFT. Each row presents a model. Last column depicts *Δrel. repeatability* (orange = higher on synth. data; blue = higher on *photo*). Grey values are non-significant. Green background presents coefficients affecting performance difference positively, if reduced.

Terrain	CSD	CLD	SCD	DCD	HTD	EHD	R^2	F-Ratio	Δrel. Rep.
concrete			-0.07 Ns	0.01 Ns	-0.08 Ns	-0.07 Ns	48%	4***	7%
forest	-0.52*	-0.19***	-0.05 Ns			-0.23 Ns	68%	16***	-13%
hangar		-0.15**	0.47***	-0.20***		0.20 Ns	36%	9***	11%
heath	0.21 Ns	-0.09 Ns	-0.29 Ns	-0.11 Ns		-0.11*	49%	6***	11%
house	-0.37 Ns		0.21 Ns	-0.12 Ns		0.28 Ns	55%	6***	4%
junkyard		-0.38*		-0.11 Ns	-0.35***	0.15 Ns	49%	7***	6%
sport	0.10 Ns	0.01 Ns	-0.01 Ns	0.03 Ns	0.05 Ns	0.03 Ns	87%	31***	18%
street	-0.08 Ns	0.08 Ns*	-0.02 Ns				66%	15***	15%

Ns = not significant (p > .05), *p < .05, ** p < .01, *** p < .001

Hangar is affected by differences in SCD, DCD, EHD and CLD. To decrease *Δrel. Repeatability* the edges appearing in the image should match the natural dataset more closely.

For scene *heath*, SCD and CSD measures reveal to be most influential but non-significant image content differences, which indicates poor representation of their behaviour by the linear model. EHD is the only significant variable. Only the colour structure distance is eligible to reduce performance differences. Here, a remodelling of the detail texture would be beneficial.

Δrel. Repeatability for SIFT in scene *house* is only 4%. The remaining difference is mainly caused by colour distribution (SCD) and edge representation (EHD).

In scene *junkyard*, the model determined CLD and HTD as the most influential coefficients. However, the model indicates that lowering the edge distance could mainly reduce the remaining performance difference of 6%. Possibly the aliased edges are the causing factor.

The large performance difference in scene *sport* can be shortened by adapting the textures to decrease the colour structure distance (all others are smaller than .1).

In scene *street*, no notable influence of measured image content differences on the performance were determined.

Δabsolute repeatability regression models for SIFT

The *Δabs. Repeatability* model for SIFT (see Table 6-5) generally fits the outcome variance better (75% - 88%) compared to their *Δrel. Repeatability* counterparts. CSD, CLD, SCD, DCD and EHD distances measurably affect the performance in scene *forest*. Trimming colour structure, dominant colour and edge differences should lead to a lower performance difference.

Table 6-5: Regression models of *Δabsolute repeatability* SIFT. Each row presents a model. Last column depicts *Δabs. repeatability* (orange = higher on synth. data; blue = higher on *photo*). Grey values are non-significant. Green background presents coefficients affecting performance difference positively, if reduced.

Terrain	CSD	CLD	SCD	DCD	HTD	EHD	R^2	F-Ratio	Δabs. Rep.
concrete	-0.10*	-0.39***	0.28 Ns	-0.28*	0.06 Ns	-0.13*	75%	14***	-25%
forest	-0.86***	0.25***	0.41***	-0.26*	0.02 Ns	-0.23 Ns	66%	14***	-43%
hangar	0.49**		0.51***	-0.21 Ns	0.17**	-0.08*	88%	53***	11%
heath	0.41***	-0.26*	-0.30***			-0.12**	62%	24***	-15%
house	-0.43 Ns	0.05 Ns	-0.31***	-0.21 Ns	0.22***	0.81***	66%	11***	-33%
junkyard	-0.30 Ns	0.04 Ns	0.13**	-0.14 Ns	-0.10**	0.04 Ns	79%	17***	-36%
sport		-0.05 Ns	0.05 Ns	0.06 Ns	0.06 Ns	0.18 Ns	85%	26***	37%
street	-0.23 Ns	0.14***	0.27*			-0.09*	76%	31***	16%

Ns = not significant (p > .05), *p < .05, ** p < .01, *** p < .001

SIFT performs best on *hangar*. The remaining difference (11%) can be minified by lowering the content distance in colour (SCD and CSD, e.g. hangar roof) and texture (HTD).

The performance difference of SIFT in scene *heath* can be lessened by a better match of colour layout, colour distribution and edge appearance.

The model for scene *house* is described by EHD, CSD, SCD, HTD and DCD distances (in order of effect). A better harmonisation of colour structure, distribution and dominance would be beneficial for this scene. This can be performed by heightening the resolution of tree textures and by remodelling their colour spectrum.

Reduced content differences CSD, DCD and HTD in scene *junkyard* would decrease the current performance difference.

SIFT performs better on synthetic data in s*port*. The model exhibits EHD as the most influential image property. All other influencing predictors have been identified to be non-significant and of small influence. Thus, edges in the synthetic data should be revisited.

Similarly, SIFT performs better on synthetic data for scene *street*. Here, lowering the difference in CLD and SCD will lead to a better representation of the natural performance.

Δrelative repeatability regression for MSER

The regression models of MSER *Δrelative repeatability* are presented in Table 6-6. R^2 ranges from 39% to 66% showing that not all effects on the performance have been revealed. On almost all scenes, MSER performs better on natural data, except for *hangar*. MSER even performs equal in *heath*. Results on scene *concrete* are displayed in the previous presentation of regression analysis.

Table 6-6: Regression models of *Δrelative repeatability* MSER. Each row presents a model. Last column depicts *Δrel. repeatability* (orange = higher on synth. data; blue = higher on *photo*). Grey values are non-significant. Green background presents coefficients affecting performance difference positively, if reduced.

Terrain	CSD	CLD	SCD	DCD	HTD	EHD	R^2	F-Ratio	Δrel. Rep.
concrete	0.15 Ns	-0.06 Ns	0.26 Ns		0.22 Ns	-0.02 Ns	56%	6***	-13%
forest	-0.60***	-0.12 Ns	-0.12 Ns	0.09**	-0.31 Ns		48%	7***	-15%
hangar	0.25**	-0.42*	0.27 Ns	-0.11 Ns	0.35***	-0.06 Ns	61%	16***	13%
heath	-0.80***	0.13 Ns	0.38 Ns		-0.22*	0.20***	39%	6***	0%
house	0.23**	0.40*		-0.22 Ns	0.39*	0.36 Ns	52%	6***	-7%
junkyard	-0.45 Ns	-0.13 Ns	0.56*	-0.16 Ns	-0.14**		55%	7***	-1%
sport	-0.75*	0.04 Ns	0.12 Ns		0.17 Ns	0.03 Ns	66%	9***	-41%
street	0.27 Ns	-0.26***	0.51 Ns	0.22***	-0.11 Ns	-0.21***	55%	9***	-11%

Ns = not significant (p > .05), *p < .05, ** p < .01, *** p < .001

The model of scene *forest* explains 48% variance of *Δrel. Repeatability*. A reduction of CSD, CLD, SCD and HTD differences is advised to close the performance gap between the dataset types.

Scene *hangar* performs better on synthetic data. Here, the performance difference can be lowered by shorten the CSD, SCD and HTD distance (by tuning colour; scattering of colour or texture).

Scene *heath* is performing equally well on both dataset types. The algorithms performance is affected by colour structure and homogeneous textures. The dataset does not need to be adjusted, since the goal of equal performance is adequately achieved.

The image content difference dominantly affecting the MSER's performance on scene *house* is CLD ($\beta=0.40$). Making the dominant colours more similar to *photo* will equalize the algorithms performance on both dataset types.

Junkyard is performing almost equal on both dataset types, according to the regression model adjusting CSD, DCD, HTD and CLD distances can help to achieve equal performance.

In scene *sport,* MSER is performing much better on natural data due to significant differences in colour structure.

Scene *street* is influenced by all described differences. Adjusting edge differences (HTD and EHD) as well as the colour layout of the scene should help to increase its performance on synthetic data.

Δabsolute repeatability regression for MSER

The *Δabsolute repeatability* measures of MSER are highly scene dependant (-91% to 115%). The models cover the outcome variance between 22% to 93%. MSER's remaining performance difference in scene *forest* can be reduced by adjusting the colour layout (CSD) and the edge appearance (HTD and EHD) of the synthetic dataset (see Table 6-7).

Scene *hangar* performs immensely better on synthetic data due to CSD, SCD and HTD differences according to the fitted model.

In *heath,* MSER performs much better on natural data, which can be adjusted by looking into appearance difference of colour layout, homogenous edges and dominant colours. Here, probably the surface detail texture is causing these appearance disparity (different seasons).

According to the model, minifying the distance of CSD and DCD could lower *Δabsolute repeatability* of MSER in scene *house*.

Table 6-7: Regression models of *Δabsolute repeatability* MSER. Each row presents a model. Last column depicts *Δabs. repeatability* (orange = higher on synth. data; blue = higher on *photo*). Grey values are non-significant. Green background presents coefficients affecting performance difference positively, if reduced.

Terrain	CSD	CLD	SCD	DCD	HTD	EHD	R^2	F-Ratio	Δabs. Rep.
concrete	0.11 Ns	-0.56***	0.23 Ns	0.20***		-0.36**	74%	17***	-58%
forest	0.55***	-0.15**		0.1***	-0.08 Ns	-0.24 Ns	59%	14***	-7%
hangar	0.45***	-0.18 Ns	0.26*	-0.08**	0.34***	-0.02 Ns	93%	81***	115%
heath	0.33 Ns	-0.81***		-0.12***	-0.50 Ns	0.36*	78%	17***	-80%
house	-0.23***	0.91***	0.19 Ns	-0.21*	0.70***	0.48*	88%	55***	-29%
junkyard	-0.26 Ns	-0.16**	-0.14***		-0.49***	-0.20 Ns	76%	24***	-31%
sport	0.01 Ns	0.03 Ns	-0.69**		0.25 Ns	-0.40***	80%	24***	-91%
street		-0.21***	0.05 Ns	0.07 Ns		0.20***	22%	4***	14%

Ns = not significant (p > .05), *p < .05, ** p < .01, *** p < .001

Junkyard on the other hand is fitted to differences of HTD, CSD, EHD, CLD and SCD. All of them should be minified to achieve equal performance.

In scene *sport*, shortened colour and non-homogenous edge differences would improve the performance of synthetic data, according to the model.

In *street*, MSER performs better on synthetic data. The edge differences could be adjusted for mitigation.

6.1.4 Summary of baseline experiment results

The presented investigation compared measured image content differences against performance differences of feature detectors applied to photographs and rendered imagery (in baseline configuration) to identify the reasons for existing performance differences. The results can be summed and discussed as follows:

Object (feature detector) performance

Performance of feature detectors was measured in *relative* and *absolute repeatability*. The performance of the feature detectors SIFT, SURF and MSER was determined on all scenes.

- The *absolute repeatability* measure is more scene specific.
- **SIFT** is the **best performing** algorithm, achieving the highest *relative* and *absolute repeatability* measures. It yields constantly good performances over the course of a scene with slightly better results on synthetic data for *Δrelative repeatability.*

- Additionally, the **performance differences** of **SIFT** for *Δrelative* (-13% to 18%) and *Δabsolute repeatability* (-43% to 37%) are generally **smaller than** those of **MSER** (*Δrelative*: -41% to 13%; *Δabsolute*: -91% to 115%).
- **MSER** achieves the **lowest performance**. It is more sensitive to changes during the scenes. In general, it performs better on photographs with few exceptions.
- **SURF** generally performs similar to **SIFT** on real and synthetic data, but in **some cases fails completely** due to a deviating default configuration (hessian determinant set to 1000 instead of 600; leading to lower absolute detection rate and unreliable behaviour). Thus, the acquired results do not reflect SURF's default performance. This circumstance and its close relation to SIFT lead to an **exclusion of SURF** in the following *configuration set* evaluations (chapter 6.2).

Image content distance

In the next evaluation step (chapter 6.1.2), the datasets *photo* and *baseline* had been compared using six different *image descriptors* (which showed the necessary capabilities in chapter 5.1.4). The evaluation showed distinctively that *image content distances* varied for each scene and *content descriptor*.

Influence factor analysis

Eventually the results have been fitted to models designed to explain the influence of specific image content descriptors on the performance of a feature detector.

The presence of collinearity between descriptors was analysed. The models and associated input data were also analysed for several **assumptions** necessary to **allow generalization** of the model. The investigation **revealed** the **compliance** of these assumptions.

For each combination of scene, repeatability measure and feature detector one regression model was fitted. The resulting standardized regression coefficients β where analysed for their size (amount of effect) and sign (direction of effect). In most cases, individual scenes were described best by a combination of descriptors. Further, **only coefficients promising a more balanced feature detector performance** on synthetic and natural data when reducing the content distance have been discussed, since increasing the difference between the dataset types to achieve identical performance is not desired.

Δrelative repeatability of SIFT can be lowered by minifying the edge differences (for all scenes except *concrete* and *heath*). Generally, each scene was affected by varyingly combinations of image attributes. For all scenes, at least one image attribute was identified to reduce the performance gap notably. Colour structure (CSD) and colour layout (CLD) have been identified affecting SIFT's performance on several scenes.

For *Δabsolute repeatability,* downsizing colour structure, colour distribution and dominant colour distance positively affect the performance gap on five scenes. In scene *concrete, heath* and *street* a lowered colour layout distance (CLD) was identified in having a positive effect. The appearance of edges (EHD) in *concrete*, *forest*, *heath* and *sport* has a detrimental effect. Repeating homogeneous edges have been identified in *hangar*, *junkyard* and *sport* to influence SIFT.

The *Δrelative repeatability* of MSER is positively affected by a trimmed colour structure (CSD) for scene *forest, hangar, junkyard and sport* and colour layout (CLD) for scene *forest, junkyard* and *street.* Differences in repeating edges (HTD) affect MSER also in four scenes (*forest*, *hangar*, *junkyard* and *street*). For *house,* the dominant colour distance (DCD) should be lowered. The model coefficients for scene *concrete* indicate two distances (CLD and EHD) of low impact ($\beta < .1$) reducing the performance. Colour distribution (SCD) distances explain existing performance differences in *forest* and *hangar*. EHD differs significantly in scene *street* (and should be lowered, more in chapter 6.2).

The *Δabsolute repeatability* of MSER can be lessened by lowering colour layout (for scene *concrete*, *forest, heath* and *junkyard*), colour structure (*hangar*, *house* and *junkyard*), colour distribution (*hangar, junkyard* and *sport*), homogeneous edge (*hangar*, *heath* and *junkyard*) and non-homogeneous edge (for all scenes except *hangar*, *house* and *heath*) distances. Further, *heath* and *house* are positively affected whenever dominant colour differences are lessened. As can be seen, the type of image content is highly dependent on the scene (and how it is modelled in the synthetic environment) and the feature detector. Still, edge differences (EHD) appear more often in models than other distances, demonstrating their effect on the performance on feature detectors.

Conclusions for next test steps

The regression analysis has been identified as a valid tool to identify the relations between image content distances and performance differences. The following configuration set

experiments (evaluating several *rendering engine parameters)* were conducted using only the feature detectors SIFT and MSER. Due to its non-standard parametrization, its similarity to SIFT and the necessity to delimit the amounts of test data SURF has been discarded.

6.2 Configuration set experiments

In the following experiments, the configuration of the *rendering engine* was changed one-parameter at a time to investigate whether it replicates the behaviour of CV-algorithms on natural data better compared to *baseline*. The same three-step analysis concept with slight modifications was applied here:

- The Object performance evaluation is identical to the previous chapter (6.1.1).
- Image content distances in these experiments are presented only in a bar plot depicting the median of each dataset to compress the results. The regression models (32 in total; one for each scene, metric and feature detector) are the same as used in chapter 6.1. This time the various parameters of the synthetic environment and their influence on image content appearance and test object performance are evaluated.
  The various parameters have been incorporated into the regression model using a categorical predictor. A detailed description on how to read the models can be found in appendix C.1. For more theoretical background on this topic see (Field, 2009).

- The previous chapter presented the baseline results using standardized β-coefficients. Here, the **non-standardized *b*-coefficients** are used as these values can be directly substituted into the model equation and the capability to compare between various models is not necessary here.
- These ***b*-coefficients** are **multiplied by the image content distance** to acquire the influence of each image content descriptor in object performance units and to ease comprehension of the results. (*Δrepeatability*).

Before, the specific configuration sets are discussed, a short introduction on how to read the influence factor analysis results.

How to read the influence analysis results

The approach to **identify interesting results** in these experiments is following (per feature detector):

- Is a rendering engine **parameter affecting $\Delta repeatability$?**
- If yes, is it also **lowering** the **content distances**?
- **Which predictors** (large b*distance values) are dominantly influencing the performance difference positively (per configuration)?
    - If $\Delta repeatability > 0$ then **positive** terms (b-value*distance) indicate distances allowing **reduction of** the remaining **difference** (positive influence)
    - If $\Delta repeatability < 0$ then **negative** terms (b-value*distance) indicate distances allowing **reduction of** the remaining **difference** (positive influence)
- Is it **scene dependent** or a **general effect**?

This approach is now conducted on the *Δrelative repeatability* SIFT *forest* regression model using configuration set "Illumination" as an example. In Figure 6-12, the object performance results are given on the left and the $b * distance$ terms are given on the right.

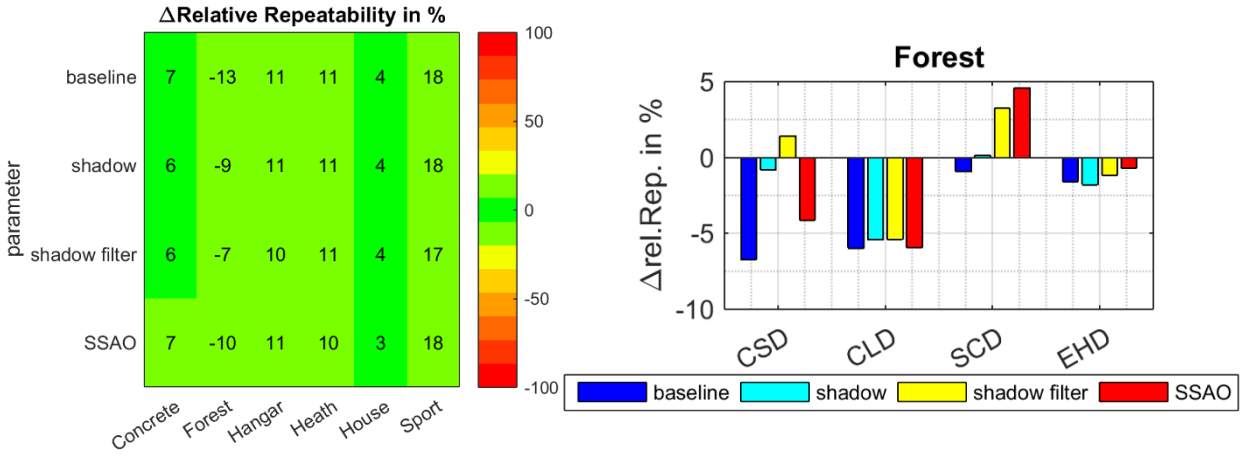


Figure 6-12: Object performance results (left) and regression coefficients b*distance results (right) of the model *Δrelative repeatability* SIFT on scene *forest* for configuration set "Illumination".

First, the object performance results are reviewed whether any parameter has an effect on the feature detectors performance. Small deviations of ~1% are considered measurement errors, when they do not appear on more than two scenes. The results depict that in scene *forest* all three tested parameters reduce the performance deviance.

The second step is to check, which content distances are actually diminished by these parameters, since the target is to identify beneficial parameters that close the performance and appearance gap to natural imagery. Here the information that all content distances except EHD (which increases for *shadow* and *shadow filtering*) are lowered by the parameters (a more detailed discussion can be found in 6.2.1.2 and Figure 6-15).

In the next step, first the $b * distance$ terms of *baseline* results are reviewed (blue bars). Here, it can be seen that colour structure differences are responsible for -7% of the total performance deviation, colour layout for -6%, EHD for-2% and SCD for -1%. Resulting in -16% when summed up. This misalignment to the actual performance of -13% is compensated by the intercept. This component is simply the offset of the regression model. An offset of zero would indicate that the behaviour of SIFT is fully described by the model. The intercept is not displayed as it has no interpretable meaning (Freund & Littell, 2000). Interested readers can find the full models in appendix C.3.

Generally, a bar indicates the impact a specific image content difference has on the performance (bigger bar, bigger influence). The sign indicates, which dataset performs higher (negative = natural data; positive = synthetic data). In this example $\Delta repeatability < 0$, which means the performance deviance is reduced when negative effects (bars) are getting smaller and positive impacts (bars) are rising (the detector performs better on natural data).

Now, the impact of a parameter setting can be read as the induced differences in bar sizes. Thus, adding *shadows* lowers the influence of colour structure and colour distribution (SCD) to having no further impact on the performance anymore. The colour layout on the other hand is only slightly minified. Thus, the content differences between the two datasets still affect SIFT's performance. Now the performance explained by the content descriptors adds up to -8%. This shows that the behaviour of SIFT is almost fully explained by the used image content descriptors. In total, the reduction of performance difference when *shadows* are enabled is caused by the closer relation of the two image types in colour structure, colour distribution and a little bit in colour layout.

This similarly can be said for shadow filtering. The only difference here is that the remaining CLD effect is compensated by the impact of CSD and SCD distances, which now positively influence SIFT's performance on synthetic data. SSAO affects the actual image content distances only slightly, but has a strong impact on the SCD coefficient. The constant effect of CLD distances on the performance indicate that local colour deviations (e.g. specific trees)

need to be focused to make the performance of SIFT more similar to natural images. Thus, careful remodelling of trees and their colours should be conducted.

As the performance results depict, the previous effect are only surveyed on scene *forest*. Thus, the given recommendations are scene dependent.

In the following chapters, only scenes with noticeable differences (in performance difference, image content distance AND compared to baseline) in repeatability performance are discussed (for each feature detector and performance metric).

The $b * distance$ terms have been normed (after equation (37)) for *Δabsolute repeatability* to ease comparison with the object performance results. However, since *b*-values are constant over a scene and norming of the object performance is conducted frame by frame the results can slightly differ. Still, portraying the results in *Δabsolute repeatability* should allow the reader to better identify the causing image content.

6.2.1 Configuration set “Illumination”

In this chapter, the configuration set “Illumination” consisting of parameters *shadow*, *shadow filter* and *SSAO* is examined (for more detail see chapter 5.2.2.1). In configuration *baseline*, all parameters have been disabled. Only datasets with prominent shadows are evaluated. Each of the following chapters shows the results in a condensed format, for details on the evaluation process the reader is referred to chapter 6.1.

6.2.1.1 Object performance

SIFT’s performance results

The test objects performance was measured identically to chapter 6.1.1. In Figure 6-13, the performance of feature detector SIFT on the synthetic datasets *baseline*, *shadow*, *shadow filter* and *SSAO* is depicted. It should be recalled that the goal is to aim for equal performance for synthetic datasets and the natural dataset *photo*, which would be achieved when *Δrelative* and *Δabsolute repeatability* become zero. This is highlighted by the appended colour code with green indicating good results and red undesired results. Details on plots and metrics can be found in chapter 6.1.1.

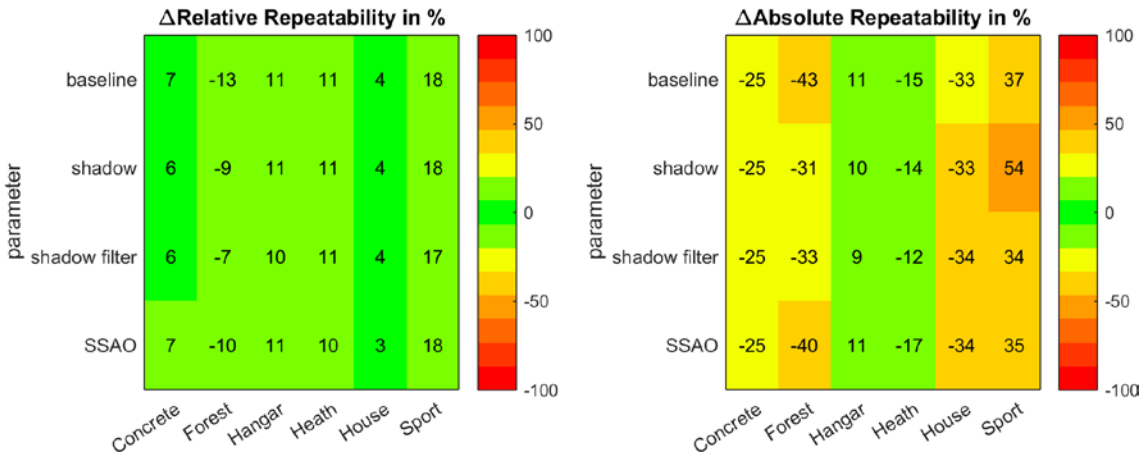


Figure 6-13: Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of SIFT on selected scenes and illumination parameters.

Marginal positive differences concerning *baseline* are measured for all parameters. For instance activating *shadows* in scene *forest* lowered the *Δrelative repeatability* by 4%. Only *forest* and *concrete* benefited from adding shadows to the scene. In summary, SIFT's *relative repeatability* is mostly independent to the parameters *shadow*, *shadow filtering* and *SSAO*.

Scene *hangar* yields the closest results to the *photo* reference on *Δabsolute repeatability* using SIFT with +9 to +11% performance difference. All parameter positively influence the performance of SIFT concerning *baseline* albeit marginally. *Forest*, *sport* and *house* exhibit the largest differences (31% to 54%). As pointed out, *absolute repeatability* measures the absolute amount of correlating features pairs detected between two images. Again, *forest* benefits the most by adding *shadows* to the scenes, due to densely placed trees casting shadows on each other. In scene *sport*, adding *shadows* increases the amount of features and thus the difference to natural images due to artificial edges originating from aliased shadows. Thus, when activating *shadow filtering*, the scene profits from adding *shadows*.

In general, only *forest* benefits noticeable from activating *shadow* and *shadow filter*. The almost equal results of *SSAO* compared to *baseline* indicate it has no notable impact on SIFT.

MSER's performance results

The *Δrelative repeatability* of MSER performs closely to *photo* for all scenes, except for *sport* as shown in Figure 6-14. *Relative repeatability* of scene *heath* is even equal to *photo*. The performance on MSER on *sport* can be attributed to the high amount of blended textures in this scene. Improvements when applying *shadow* or *shadow filter* are existent but low. In scene *heath*, the deviance to natural images even extends when *shadows* were activated. This

effect is nullified by additionally applying *shadow filtering*. Only scene *sport* benefits strongly from *shadows*. *SSAO* shows no effect on *relative repeatability* for feature detector MSER. In case of *Δrelative repeatability*, the parameters *shadow* and *shadow filter* strongly improve the number of detected features for *heath*, *house* and *sport* only. The edgy shadows of parameter *shadow* heighten the detection of robust features even more in *sport* and *heath*. *SSAO* has no impact.

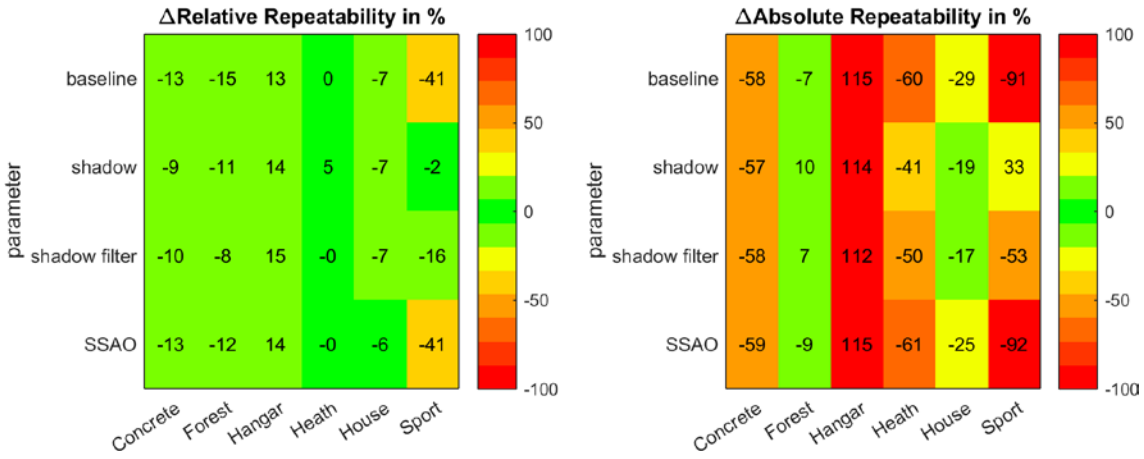


Figure 6-14: Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of MSER for selected scenes and parameters.

In summary *SSAO* has no visible influence on the performance of any tested feature detector. For SIFT the same applies to *shadow*. MSER profits from shadows (although from edgy and therefore more artificial shadows).

6.2.1.2 Image content distances

This chapter provides results on the degree and type of changes that are introduced by the parameters on the image content (see Figure 6-15). The distance D between two images (for the used descriptors) is always positive and is zero in case the images are identical for the specific image property. These distances are measured between images of the current parameter and the actual photographs. Compared to the *baseline* experiment the boxplot (Figure 6-7) has been compressed to a bar plot presenting only the median of each dataset. This way the influence of a parameter can be read in absolute figures and can be compared to the baseline value for each content descriptor. The **lower the value** the **closer** is the **similarity** to the *photo* reference.

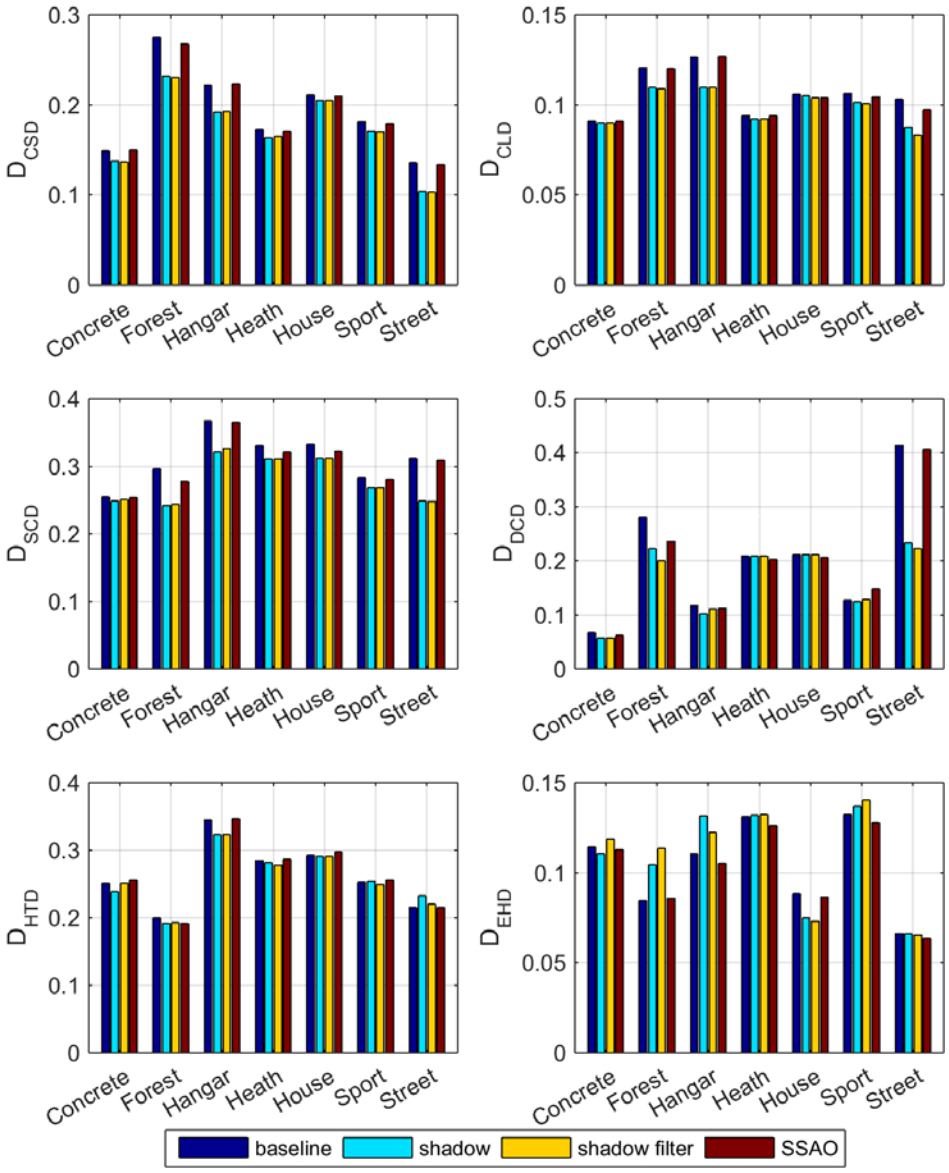


Figure 6-15: Median image content distances between synthetic imagery and *photo* for all used content descriptors and examined configuration set parameters.

The distances of CSD and SCD are lessened compared to *baseline* when *shadow* or *shadow filtering* is activated. HTD distances remain mostly unchanged. Only scene *hangar* is positively affected by *shadows*, due to the large shadow areas visible in the dataset. *Shadow* also positively affects the distance of CLD for all scenes except *house* and *concrete*. *SSAO* has no notable influence on CSD, SCD, HTD, CLD and EHD.

DCD distances are very scene dependent. The activation of *shadows* in the synthetic environment leads to a reduction of dominant colour distances in *forest*, *concrete* and *hangar* due to the presence of shadows in the natural imagery. *SSAO* only affects scene *forest*. EHD describes the spatial distribution of edges. *Shadow* enlarges the distance to *photo* for all scenes except *house*. Depending on the scene, *shadow filtering* can increase or decrease this

effect slightly. Only *house* benefits by introducing shadows making it the best modelled scene for EHD after *street*. Even though shadows occupy a considerable part of each image in dataset *sport*, EHD is not affected by their presence.

6.2.1.3 Influence factor analysis

Now, the influence factor analysis presents the influence of image content differences on the performance of the feature detectors. How these results are interpret is presented in chapter 6.2. More information on the used regression models can be found in appendix C.

SIFT model coefficients

The example in chapter 6.2 explains the procedure on how to read these results on scene *forest* using SIFT's *Δrelative repeatability*. Since this was the only scene not failing the pre-selection steps presented in chapter 6.2, no further scenes are discussed here.

SIFT's *Δabsolute repeatability* on *forest* (Figure 6-16) was strongly impacted by enabled *shadows* and *shadow filtering*. The performance mainly influenced by differences in edge appearance (HTD and EHD). It is also benefits from diminished distances in CSD and DCD but is negatively affected by shortened distances in CLD and SCD (though minor). SSAO obviously changes edge appearance (HTD and EHD).

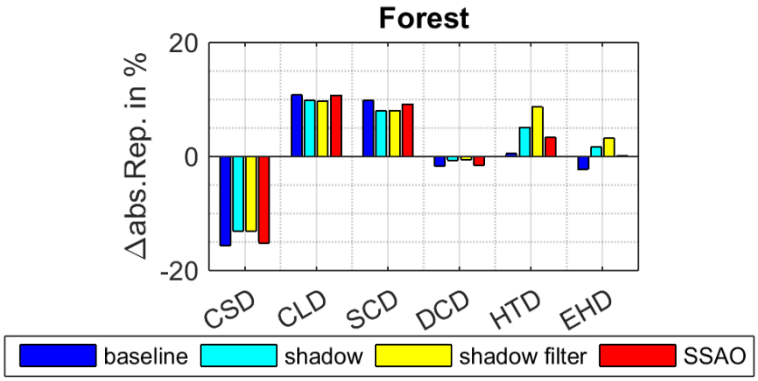


Figure 6-16: Influence of image content on SIFT *Δabsolute repeatability* and their behaviour when "illumination" parameters are applied.

The large effect results show the remarkable effect of image content differences on the absolute repeatability performance of a feature detector. However, the resulting performance cannot be computed without the intercept given in appendix C.3. Still due to the normalisation

differences will remain. However, the actual goal of this investigation is to identify image content influencing an algorithms performance not the deployment of a predicting model.

MSER model coefficients

Δrelative repeatability of MSER is robust to "illumination" parameter variation for most scenes. Only *concrete, forest and sport* presented in Figure 6-17 show responses.

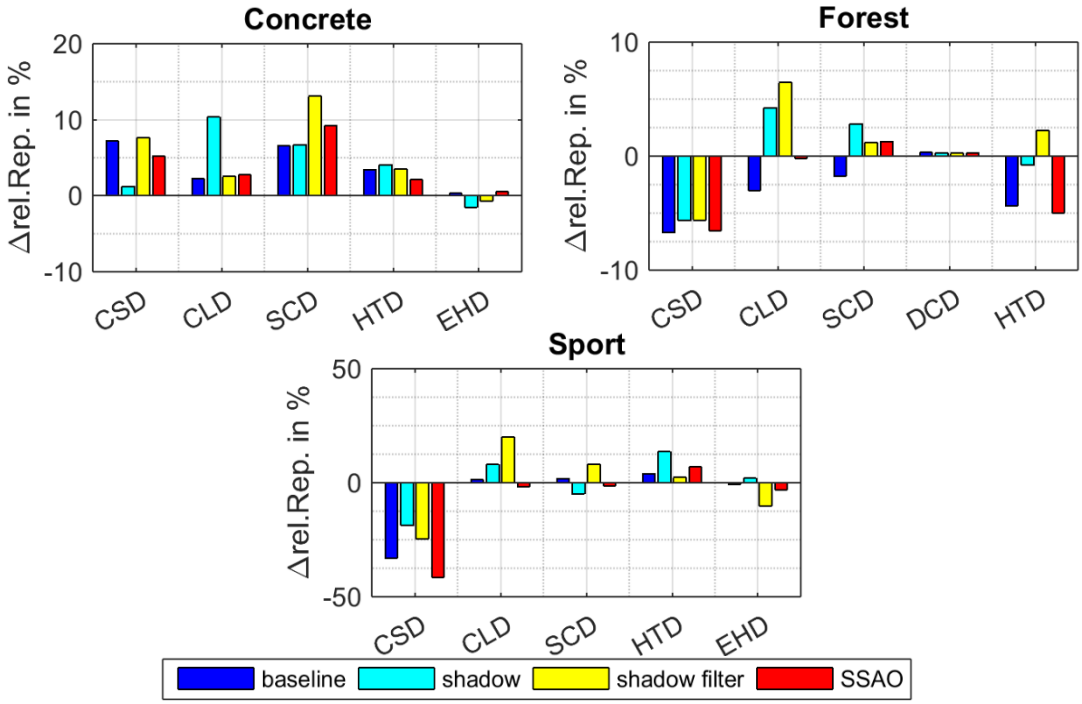


Figure 6-17: Influence of image content on MSER *Δrelative repeatability* and its variation over parameter change.

Their performance difference is reduced when *shadows* or *shadow filtering* is activated (chapter 6.2.1.1). In *concrete* this is caused by the gained influence of CLD (*shadow*) and SCD (*shadow filtering*), while in scene *forest* CSD, CLD, SCD and HTD distances are beneficial to the performance. The strong effects on scene *sport* can be attributed to a change in CSD and CLD distance. The distributed colour (SCD) affects MSER positively, when *shadow filtering* is activated. SSAO only causes performance differences in *forest* with MSER being affect by distributed colour and colour layout differences.

Beware, since *absolute repeatability* is normed by the number of image pairs detected in the natural reference, large deviations in performance leads to large coefficient values. As identified in the object performance evaluation in chapter 6.2.1.1 the *absolute repeatability* of MSER is lowered on *heath*, *house* and *sport* when *shadow* and *shadow filtering* is activated.

As depicted in Figure 6-18, the performance changes of MSER in scene *heath* can be attributed to differences in homogeneous texture (with activated *shadow*). MSER's performance with *shadow filtering* is however not explained by the model.

Colour layout and distributed colours are responsible for the performance gain on synthetic data in scene *house* for both shadow configurations. *SSAO* marginally increases the weight of HTD and EHD compared to *baseline*.

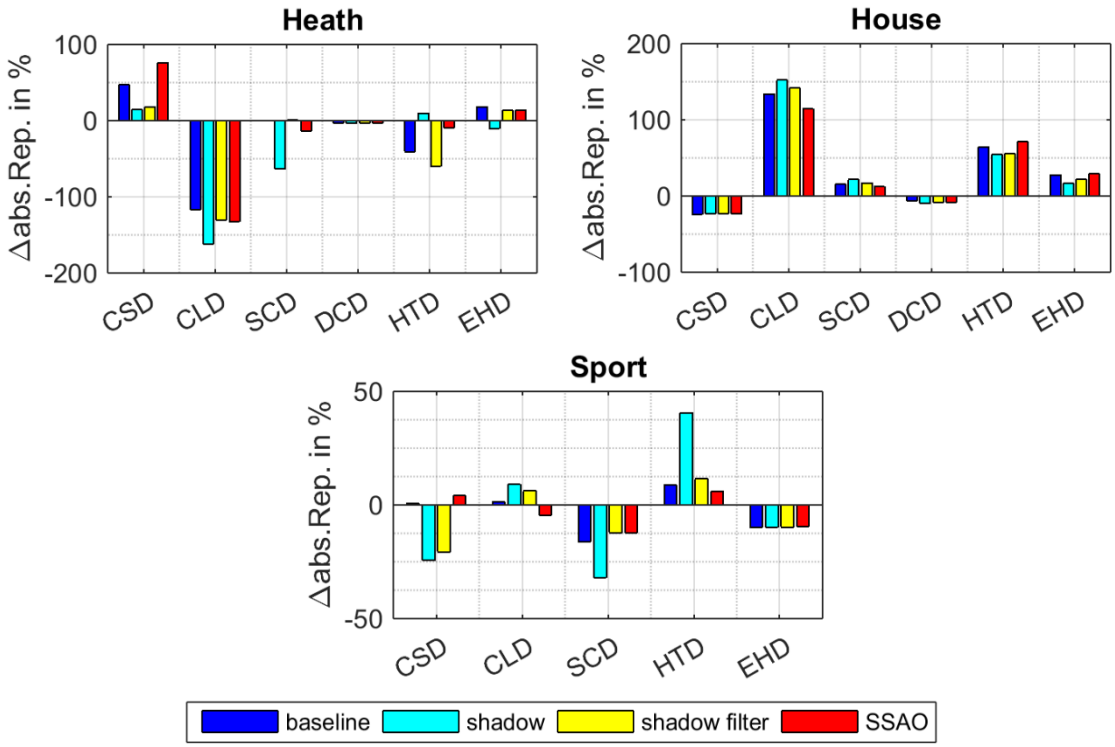


Figure 6-18: Influence of image content on MSER *Δabs. repeatability* using parameter set "Illumination".

Adding *shadows* in scene *sport* lowers MSER's *Δabsolute repeatability* strongly, due to the impact gain of texture differences on the outcome. The change in colour layout is also beneficial. *Shadow filtering* also minifies the performance difference though less drastically thanks to the lowered image content differences in SCD, CLD and HTD. Thus, unfiltered shadows decrease the performance difference to natural photos the most out of these three tested parameters by increasing the effect of repeating frequencies and colour layout. *SSAO* decreases the *Δabs. repeatability* of *sport* by only 1% reducing the influence of SCD, CSD and EHD differences compared to *baseline*.

6.2.1.4 Summary of configuration set results

In this configuration set the three parameters *SSAO*, *shadow* and *shadow filtering* were tested. Generally, the introduction of *shadows* and *shadow filtering* reduces the *relative repeatability* difference to natural images mainly for scenes with many objects. The performance of other scenes is not as affected because the satellite image used as ground texture contains the shadows present during image capturing.

In summary, activating shadows has a positive effect on *Δabsolute* and *Δrelative repeatability* for both feature detectors, with the absolute measure being more affected. The benefit of *shadows* and *shadow filtering* depends on the scene, in some the performance difference is not affected, in others it is positively affected. All in all the shadow parameters affect mainly the colour histogram (SCD), colour layout (CLD), colour structure (CSD) and gradients (HTD and EHD). Thus, the usage of *shadows* and *shadow filtering* is recommended for both feature detectors. Further, both feature detectors benefit more from filtered shadows, since it filters the aliased shadow edges (which would introduce unnaturally large gradients and high frequencies in the synthetic images). The evaluation identified that *SSAO* can marginally decrease performance difference of tested feature detectors in scenes containing densely placed trees (e.g. *forest* or *house*) due to the slight reduction of gradient distances (HTD and EHD).

6.2.2 Configuration set “Texture”

In this experiment the parameters *texture low*, *surface low* and *anisotropic filtering (AF)* were tested. See chapter 5.2.2.2 for more details on these parameters. The baseline setting is presented in Table 6-1 on page 140.

6.2.2.1 Object performance

SIFT’s performance results

The performance of SIFT with all “Texture” parameters is depicted in Figure 6-19 and can be compared to *baseline*. The *Δrelative repeatability* of SIFT is only marginally affected by *AF* and *low textures*. In case of *anisotropic filtering* only in scene *forest* the performance difference to *photo* is increased by 5%. In *Texture low* the resolution of all textures in the scene (object textures, sprites, and detail texture on ground) are quartered. This affects mainly

scene *heath* by lowering the performance difference by 4%. In *surface low* only the resolution of the satellite image (unchanged in *texture low*) is downscaled, which decreases the amount of geo-referenced features. *Δrelative repeatability* is reduced in almost all scenes for *surface low*. Scene *heath* benefits the most performing now equal to dataset *photo*, while for *forest* the performance detoriates by 5%.

When evaluating *Δabsolute repeatability* of SIFT, *texture low* boosts the number of detected features by about 1000 for all scenes. Depending on the number of features found in *baseline* and the differences to *photo* this change is reflected either positively or negatively. It should be noted that with downsized texture the number of outliers increases. Downscaling the satellite image (*surface low*) lowers the number of features in all scenes by 300 to 500 feature pairs per image. This change is mostly not beneficial. Only *sport* profits from this development since even less feature pairs are found in natural images. *Anisotropic filtering* also slightly reduces the amount of images pairs for all scenes, therefore the same scenes are benefitting as for *surface low*. For *hangar,* this leads to equal performance with the reference.

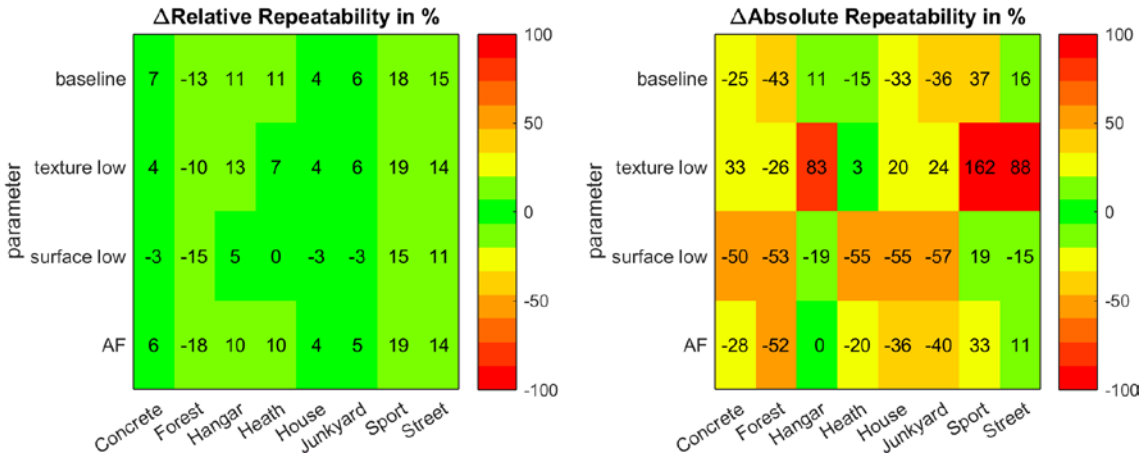


Figure 6-19 Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of SIFT on selected scenes and texture parameters.

MSER's performance results

Analysing the results of MSER's *Δrelative repeatability* (see Figure 6-20) the *baseline* dataset performs higher than *photo* on all scenes except for *heath* and *junkyard* (where it is almost equal). Scaling down textures (*texture low*), increases its performance but also lessens the robustness of measurements strongly (fast switches between high and low measurements within a dataset). Parameter *surface low* diminishes *Δrelative repeatability* compared to

baseline and is only beneficial in scene *forest*. The appliance of *anisotropic filtering (AF)* enlarges the performance difference for all scenes by 1 to 3 percentage points.

Δabsolute repeatability of MSER is very content specific (different for each scene) with *forest* comparing best to its natural counterpart. *Texture low* also cuts down on the number of detected feature pairs, which enlarges *Δabsolute repeatability* for all scenes except *hangar*. Here, MSER detects less features on synthetic data, gradients in texture are heavily blurred. Downscaling the satellite texture resolution (*surface low*) also diminishes the number of found feature pairs but less drastic than *texture low*. Thus, the detector reacts similar on all scenes, detecting less features in synthetic data when the resolution of textures is reduced. Parameter *AF* has a positive influence, enabling MSER to detect more features.

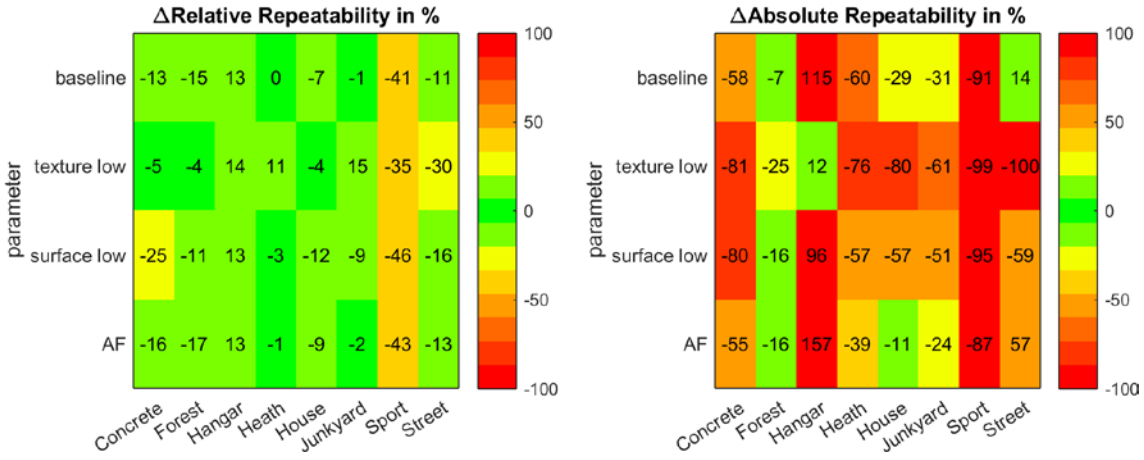


Figure 6-20: Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of MSER on selected scenes and texture parameters.

6.2.2.2 Image content distances

In this configuration set the impact of different texture resolutions and effects are tested. When textures are manipulated, so are the image frequencies. Figure 6-21 shows the mean frequencies of scene *sport* on configuration set “Texture”. Since frequencies can only be half the size of the image (Nyquist-Theorem), the maximum frequency is 512Hz (@ 1024x768). The vertical axis indicates the logarithmic amount of frequencies present in each configuration type. Compared to *photo*, all synthetic image sets lack lower frequencies, while having a surplus in higher frequencies. The break-even point is at 200-255 Hz depending on the parameter. *AF* increases the presence of frequencies above 200 Hz. Downscaling the satellite texture mainly affects the low frequency components in the band of 10-100 Hz,

which indicates the loss of low resolution texture information. Changing the texture resolution of all textures (except satellite) reduces the presence of frequencies above 70Hz massively.

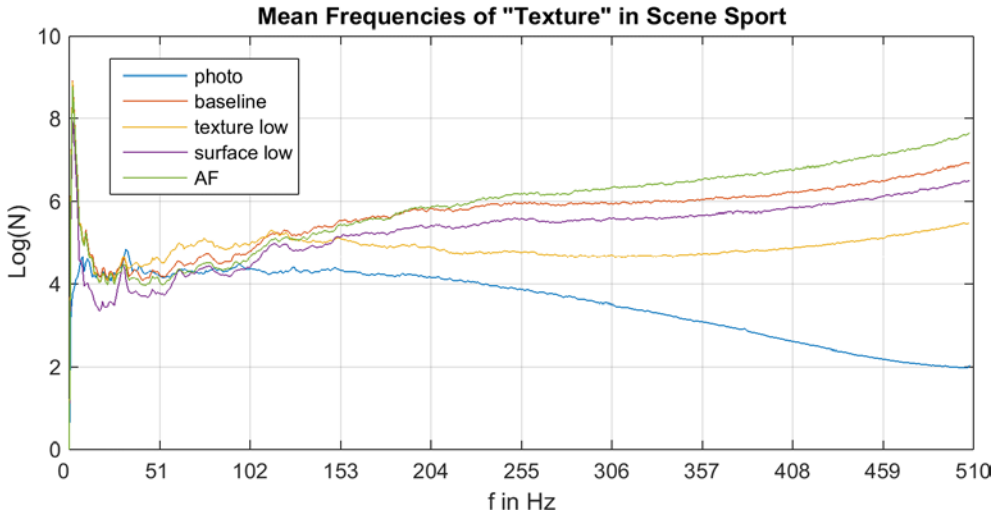


Figure 6-21: Mean frequency distribution of scene *sport* for all parameters tested in set "Texture".

Thus, scaling down the resolution of textures leads to a larger distance to *photo* in HTD and EHD (see Figure 6-22). Only completely texture based scenes such as *sport* or *street* benefit from *texture low* because the impact of the detail texture is diminished while the satellite image remains untouched. Therefore, the difference to the actual photograph is lessened. Further distances of SCD and CSD are lowered due to less cluttered colour distributions. The effect of down scaling textures affects CLD only slightly, whether the distance is increased or decreased depends on the specific scene setup, since it depends on the loss of spatial colour information.

Surface low lowers the resolution of the satellite image while retaining the resolution of the detail textures. The configuration leads to strong reductions in distance for descriptors CSD, CLD and SCD on all scenes. This trend indicates that the image composition of *baseline* is too cluttered (in texture and colour compared to *photo) and* scaling down the satellite image resolution cuts this effect. Normally the difference to *photo* should rise, due to the blurring and thus disguising of geo-features. This contradiction shows the current surface texture generation (blending with detail texture) creates results worse than the pure usage of either texture. For HTD, only the two texture heavy scenes (*sport* and *street*) benefit from this configuration.

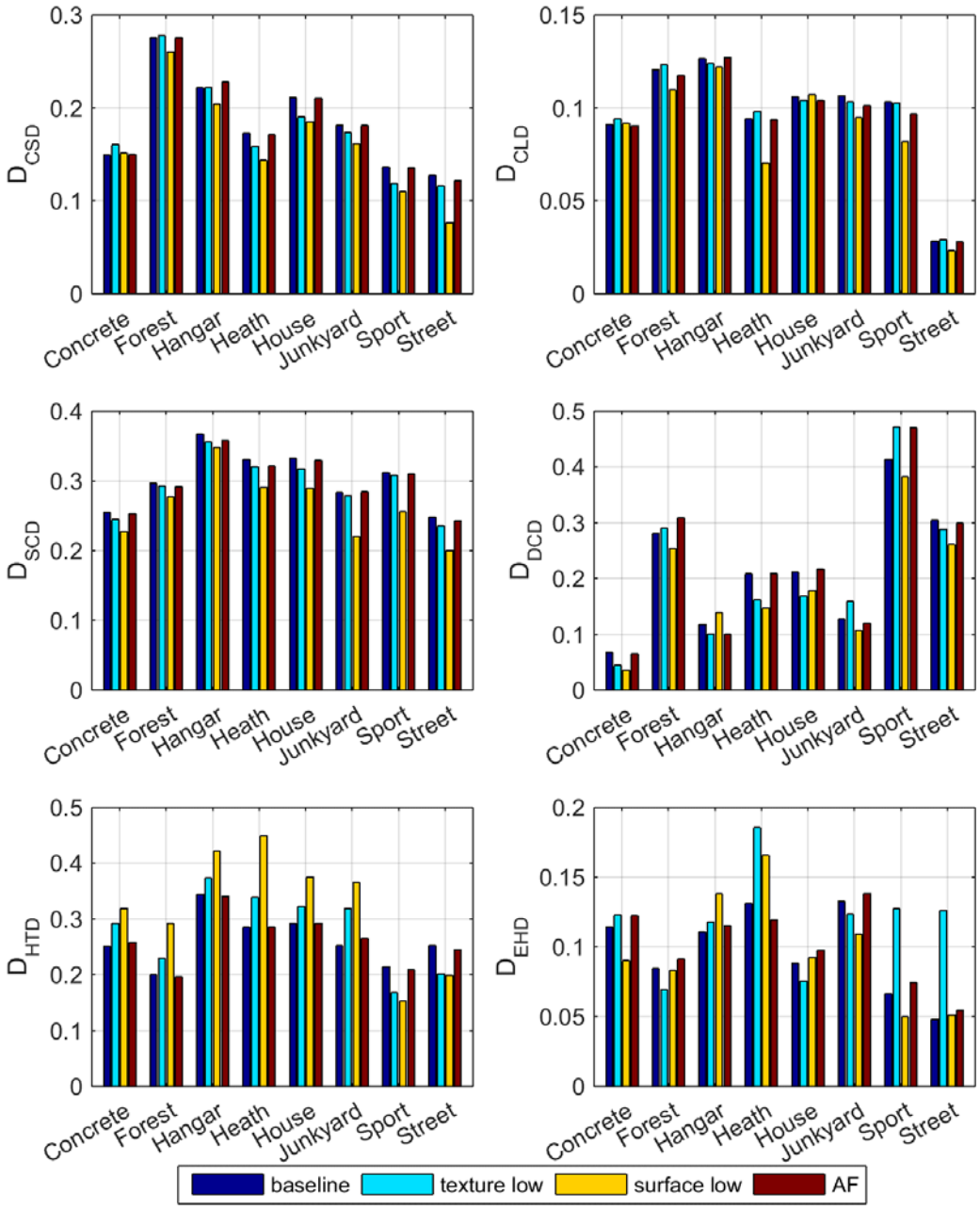


Figure 6-22: Median image content distances between synthetic imagery and *photo* for all used content descriptors and examined parameters.

Anisotropic Filtering affects the distances of all descriptors only slightly. SCD, CSD and CLD measurements are marginally reduced on almost all scenes. DCD also changes only to a small degree for *AF*, except for the scenes *forest* and *sport*. Further *AF* leads to a rise in distance of content descriptor EHD.

6.2.2.3 Influence factor analysis

Again, image content distance measurements and performance differences have been used to compute the regression models. The variance covered by these models can be found in appendix C.2 (Figure C-3). In this chapter, the **model coefficients *b* multiplied by the measured distance** are used to present the causing image content differences.

SIFT model coefficients

Figure 6-23 presents the results of each "Texture" parameter on the scenes *forest*, *hangar* and *heath*. The resulting terms ($b * distance$) are given in the same unit as object performance results (see chapter 6.2.2.1). In scene *forest*, the improved performance on *texture low* can be explained by the reduction of colour structure (less scattering) and high frequency edge gradients (EHD). *Surface low* and *AF* slightly enlarge the performance difference of SIFT.

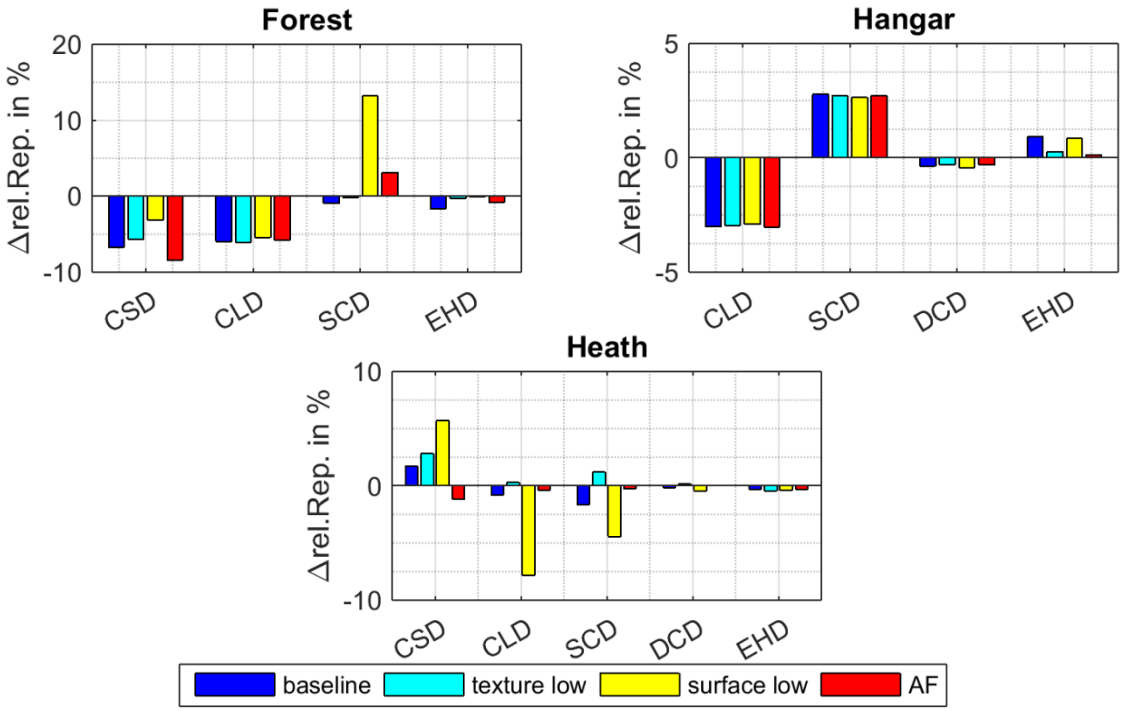


Figure 6-23: Influence of image content on SIFT *Δrelative repeatability* and their behaviour when "Texture" parameters are applied.

Configuration *texture low* raises the *performance difference* in *hangar* by 2%, which can be correlated to an increased distance in colour layout. Changes in Colour layout and edge appearance explain the lowered performance difference when downscaling the *surface texture*. In the case of *AF*, the lower influence of edge appearance on the outcome leads to a smaller performance difference.

In Scene *heath*, the model cannot explain the smaller performance gap for configuration *texture low*. Equal performance is measured with configuration *surface low* due to colour layout and colour distribution. This suggests that satellite images from a different season used as ground texture caused the colour differences.

Texture low strongly boosts the *absolute repeatability* of SIFT on synthetic imagery on all scenes. *Surface low* on the other hand lessens SIFT's *absolute repeatability* on synthetic data in all scenes. Scenes *hangar*, *heath* and *sport* are presented in Figure 6-24 representing all scenes.

In scene *hangar*, the CLD, SCD, DCD and HTD terms explain the further upturned performance difference with *texture low*. The slightly negative performance of *surface low* is caused by the distances of CSD, SCD and DCD. *AF* actually leads to equal performance due to slight changes in the SCD, CSD and CLD terms.

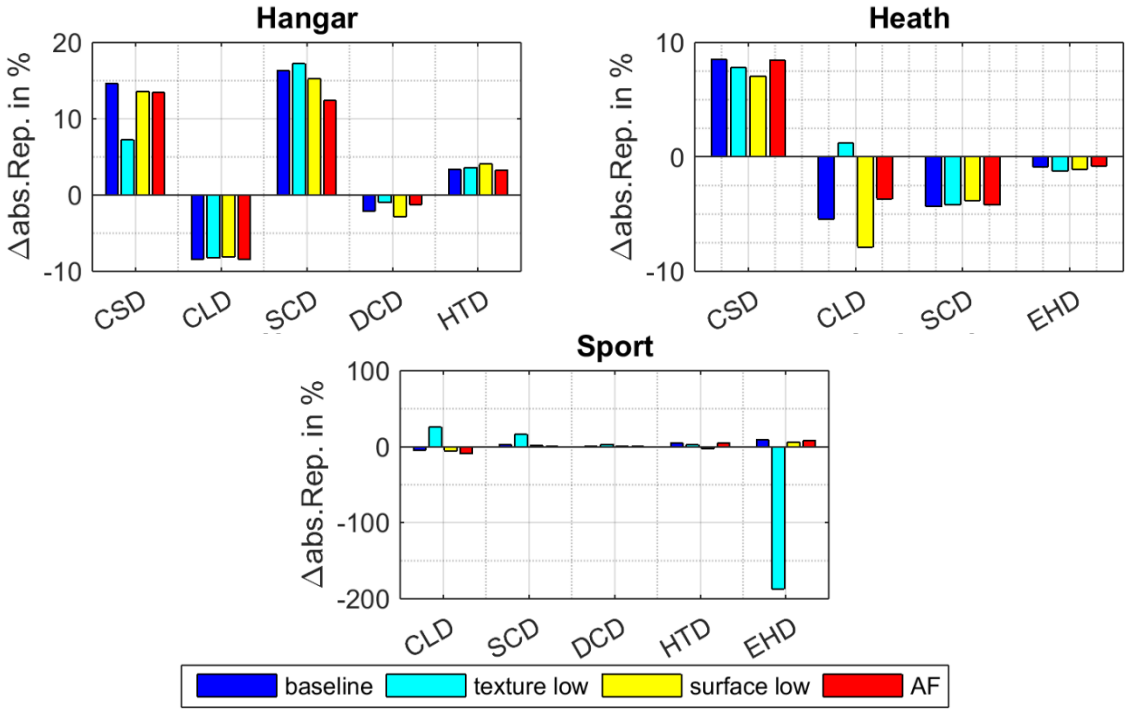


Figure 6-24: Influence of image content on SIFT *Δabsolute *repeatability* and their behaviour when "Texture" parameters are applied.

In case of *heath*, t*exture low* lowers the performance difference to 3%. According to the model, the change in colour layout is responsible. *Surface low* leads to a strongly reduced number of image features in synthetic images of scene *heath*. Colour structure and layout are identified as the main cause.

In scene *sport* the already strong performance of SIFT on synthetic data is further enlarged in configuration *texture low* by the drastic effect of the heightened EHD distance. This change in texture quality leads to much more feature pairs detected by SIFT due to the blurring of edges. *Surface low* is beneficial to the scene by lowering colour layout, homogeneous texture (HTD) and thus the amount of detected features in synthetic images.

MSER model coefficients

The *Δrelative repeatability* of MSER is affected by *texture low* and *surface low*, while remaining relatively robust to AF (see Figure 6-25). Therefore, AF results will not be discussed in detail. *Texture low* diminished the performance difference of MSER in scene *concrete* to only 5% mainly due to the change in CLD followed by SCD and EHD, which all raised their content distance compared to *photo*. The rise in repeatability when *surface low* is active can be traced to the changes in CSD and SCD distances.

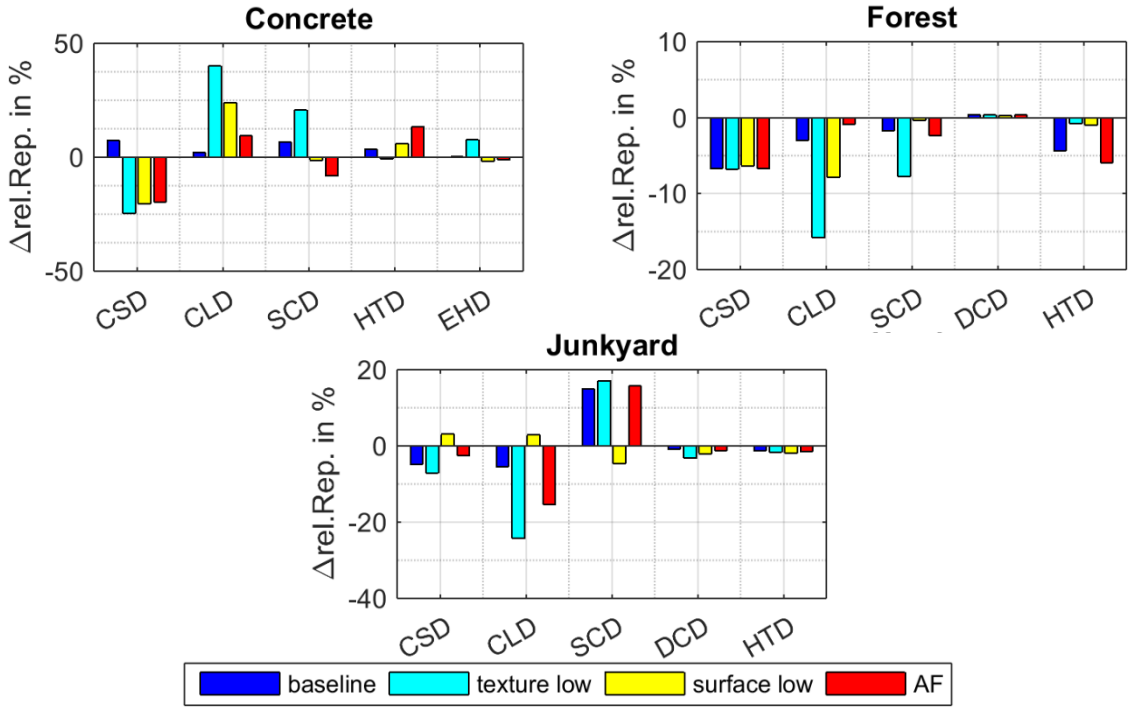


Figure 6-25: Influence of image content on MSER *Δrelative repeatability* and their behaviour when "Texture" parameters are applied.

Configuration *texture low* minifies the performance difference in scene *forest* to -4%. This is caused by the lowered impact of HTD distances. In case of *surface low*, the performance reduction results from a smaller influence of the SCD and EHD terms.

MSER's performance on *junkyard* is explained by five descriptors. Both measures *texture low* and *surface low* heighten the *performance difference* to *photo*. A loss in colour distribution (SCD) causes a performance shift when the textures are scaled down. *Surface low* diminishes the performance on synthetic data leading to -9% performance difference caused by changes in colour structure and colour layout.

Evaluation of MSER's *Δabsolute repeatability* revealed a general increase of performance difference to *photo* for all configuration on all scenes with the exception of *texture low* on *hangar* and *AF* on *heath*. Thus, only these two are presented in Figure 6-26 and discussed in the following.

The performance lowering effect of *texture low* on synthetic data results in a remaining performance difference of 12% for scene *hangar*. This is caused by higher distance in CSD and EHD and a higher similarity in CLD and SCD.

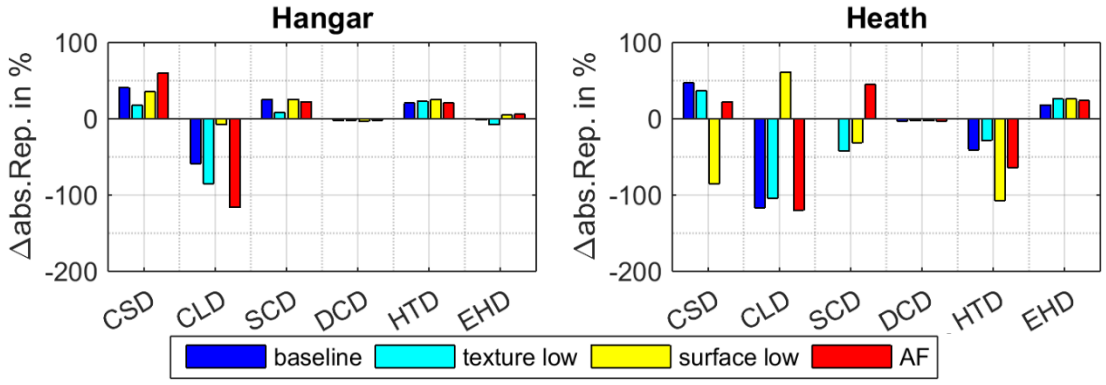


Figure 6-26: Influence of image content on MSER *Δabsolute repeatability* and their behaviour when "Texture" parameters are applied.

Heath mainly benefited only from configuration *AF*. Anisotropic filtering blurs step surfaces to pare down aliasing effects in texture. The closer performance to photo is induced by the change in colour distribution (SCD). *Surface low* only slightly improves the performance of MSER on *heath*. Here, the downscale of the satellite texture shortens the distance in CLD, thus allowing the dataset to perform more similar to *photo*.

6.2.2.4 Summary of configuration set results

The *Δrelative repeatability* of SIFT and MSER is affected by the parameters *texture low* and *surface low*. *AF* has either no or negative influence on the performance difference. Changing the texture parameters mainly affects the *Δabsolute repeatability* of both feature detectors

negatively (with some exceptions). Thus, the reduction of texture resolution (*texture low*) or satellite image resolution (*surface low*) is not recommended. Further, *AF* does not lead to a closer performance of synthetic data in regard to *photo*, except for *Δabsolute repeatability* on *heath* and *house* using SIFT and *hangar* and *street* using MSER. Compared to baseline *AF* affects the performance of both feature detectors slightly negatively and thus should not be used in this case.

When a lower performance difference was measured it was generally caused by shortened distance in colour layout and colour distribution (for both feature detectors) indicating that a high-resolution surface may lead to an enlarged difference when the actual colours do not align (possibly due to construction of buildings or seasonal colour changes).

Texture low strongly increases the *absolute repeatability* of SIFT on synthetic imagery on all scenes. *Surface low* on the other hand reduces SIFT's *absolute repeatability* on synthetic data in all scenes.

In some cases, *Δrelative repeatability* of MSER can benefit from downscaled textures (texture low). *Surface low* and *AF* are never beneficial. According to prior investigation, all configurations generally expand the *Δabsolute repeatability* of MSER.

In general, down scaling textures affects the number of detected feature pairs (*Δabsolute repeatability*) strongly. Whether it also effects the *Δrelative repeatability*, depends on the detector and type of change. The detectors are mainly affected by spatial and global colour distribution (CLD, SCD). *Anisotropic filtering* has marginal impact on feature detector performance.

6.2.3 Configuration set "Edge"

In this configuration set the different antialiasing techniques *SSAA*, *MSAA*, *FXAA*, *SMAA* and *AToC* are evaluated. *AToC* is an aliasing technique for sprites only, which can be found on grass and tree models of VBS3 (for more detail see chapter 5.2.2.3).

6.2.3.1 Object performance

SIFT's performance results

Δrelative repeatability of SIFT is affected only to a minor degree by methods AA method tested as can be seen in Figure 6-27. *Forest* benefits the most getting 4% closer to the performance of *photo*. In general, the methods *FXAA* and *SMAA* robustly decrease the distance to *photo*. Since the effect of *AToC* can only be seen on trees and bushes its relevant scenes are limited to *forest*, *heath* and *junkyard*.

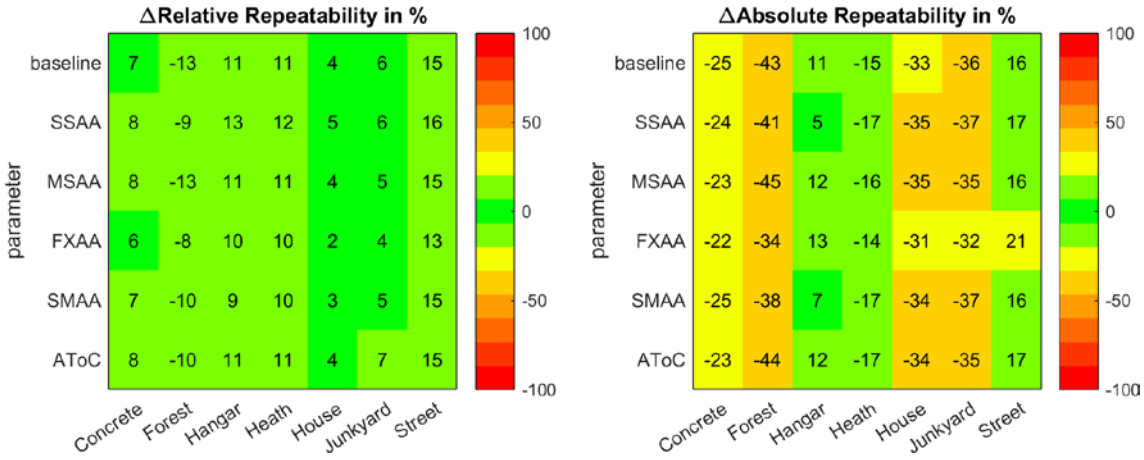


Figure 6-27: Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of SIFT on selected scenes and edge parameters.

Antialiasing methods affect SIFT's *Δabsolute repeatability* also only slightly. Here, scene *forest* and *hangar* profit most while *street*, *house* and *heath* generally are not affected. The technique improving most of the scenes is *FXAA* followed by *SMAA* and *SSAA*. It can be assumed that *AToC* and *MSAA* are not affecting the number of corresponding feature pairs.

MSER's performance results

Aliasing occurs on edges of objects, thus scenes devoid us such as *forest*, *heath* and *street* benefit from least antialiasing techniques (except for *AToC*) when evaluating the *Δrelative repeatability* of MSER. When excluding these scenes MSER benefits to a certain degree from *SSAA*, *MSAA*, *FXAA* and *SMAA* depending on the scene (see Figure 6-28). The equal performance of SMAA to *photo* on *hangar* is due to strong outliers and therefore is not considered in the discussion. *FXAA* leads to increased *relative repeatability* on synthetic data but also instability leading to a wide spread of results. MSER benefits only in scenes with many objects (which induce strong edge features) from antialiasing such as *concrete*. *AToC*

works on trees, bushes and grass present in scenes *forest* and *heath*. However, it marginally reduces *Δrelative repeatability* for all scenes except these two. Thus, in this case *AToC* can be neglected. In general, MSER benefits only slightly from antialiasing methods when analysing *Δrelative repeatability* with *SSAA* having the overall best results.

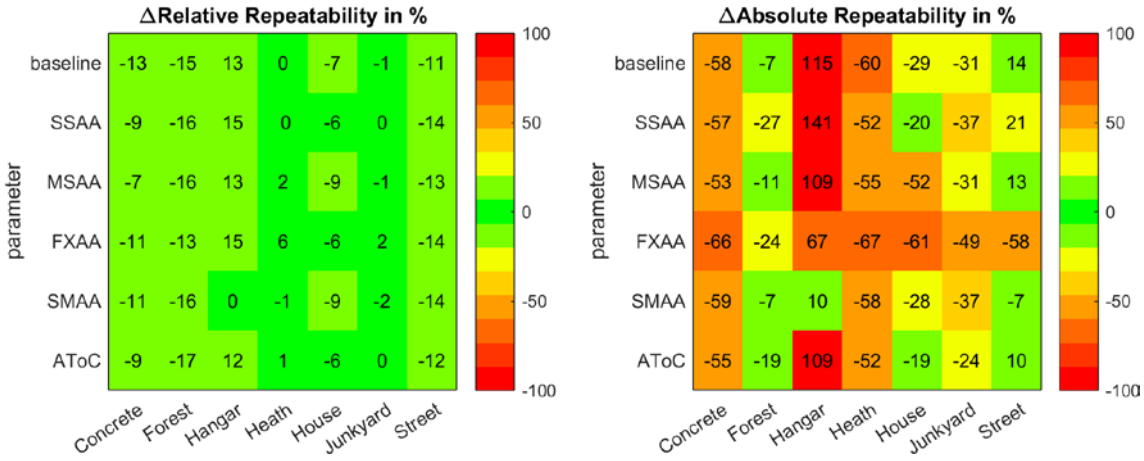


Figure 6-28: Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of MSER for selected scenes on edge parameters.

When considering *absolute repeatability* of MSER the relation to *photo* is very scene dependent. *SSAA* and *MSAA* boost the number of detected feature pairs on synthetic data. Again, the result of *hangar* with *SMAA* is due to outliers and cannot be considered. In *hangar* and *street*, the number of features on synthetic images is already larger than that on natural images (leading to negative trends). *FXAA* raises the performance gap to *photo* for every scene revealing an incompatibility with MSER. Depending on the Scene *SSAA*, *SMAA* and *MSAA* can slightly cut down the performance difference. *AToC* fits the number of detected features in each frame best to the reference making it the best AA method to use. Remember, *AToC* only works in combination with *MSAA*.

6.2.3.2 Image content distances

All antialiasing methods are used to smooth jagged edges resulting from the rendering process. These edges induce a spectrum of unnatural high frequencies in the image as depicted in Figure 6-29. Here, *FXAA* filters the number of high frequencies most successfully compared to all other methods. *MSAA* and the related *AToC* affect the image frequencies only slightly as presented here on the example of scene *forest*.

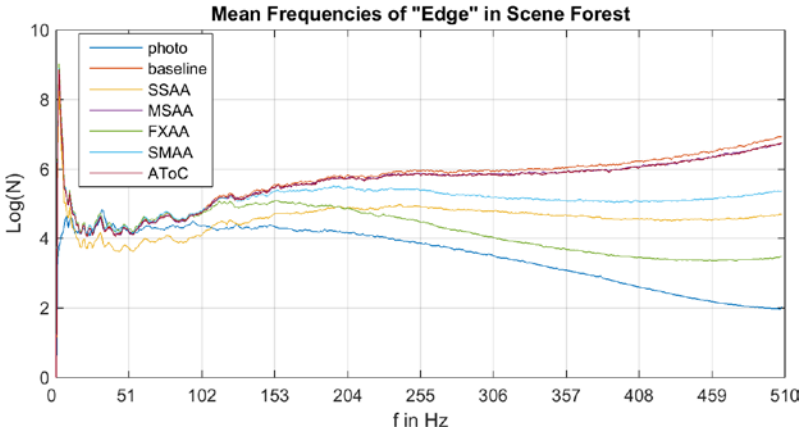


Figure 6-29: Mean frequency distribution of scene *forest* for all parameters tested in set "Edge".

SSAA cuts down the distance to natural images for CSD, CLD, SCD, DCD and EHD as depicted in Figure 6-30. Since AA methods blur strong gradients the colours fade as well leading to less dominance of the dominant colours especially in scene *concrete*.

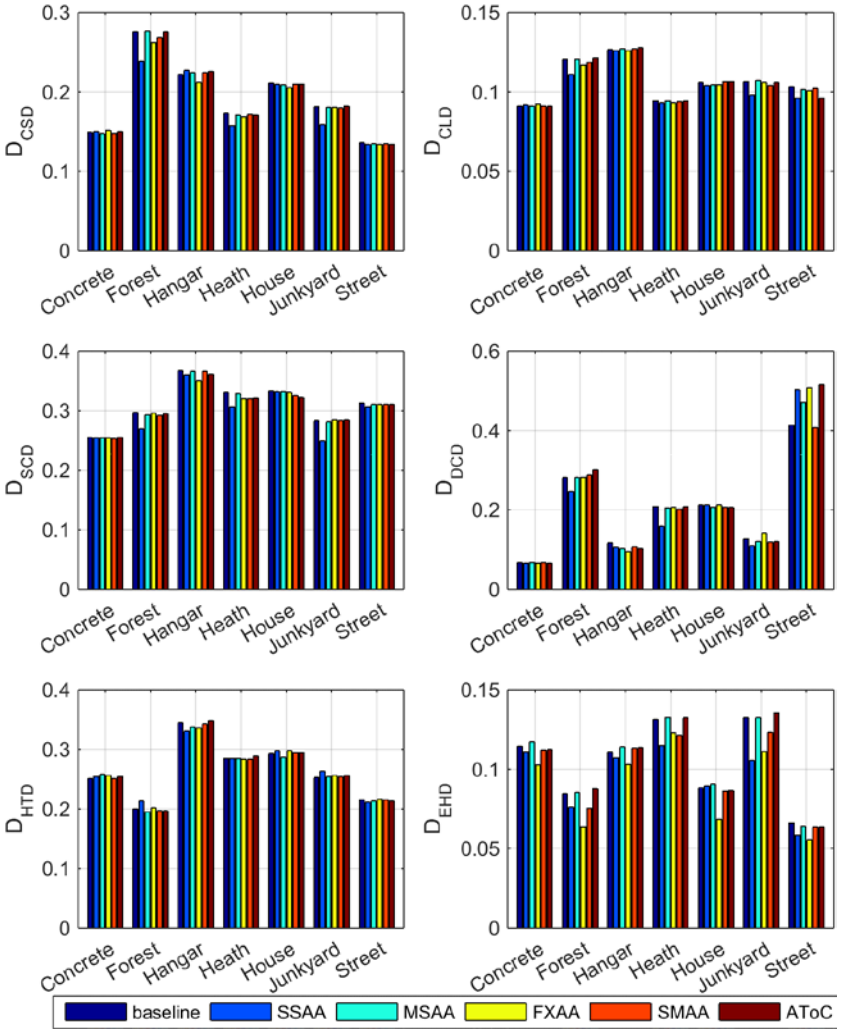


Figure 6-30: Median image content distances between synthetic imagery and *photo* for all used content descriptors and examined parameters.

FXAA shortens the distances of CSD, EHD and slightly SCD. As expected the edge histogram (EHD) is the most affected descriptor with *FXAA* lowering the distance to photo the most. *FXAA* detects edges in an image after rendering and blurs these. A drop in distance indicates that aliased edges in rendered images have unnatural large gradients. The application of *MSAA* and *AToC* does not affect the image descriptors used in this evaluation. *SMAA* slightly reduces the distance of CSD, SCD and EHD. The effect on other descriptors is depending on the scene. *SMAA* is considered to outperform *FXAA* (Jimenez et al., 2012), but its effect on image content is lower. The graphs indicate that *street* and *concrete* are especially influenced by all AA-methods.

6.2.3.3 Influence factor analysis

Each combination of metric, feature detector, scene is fit to a *regression model* using the *results* from the *image content analysis* and the different configuration set parameters as predictors. The quality of fit together with the significance have been presented in appendix C.2.

SIFT model coefficients

The regression model explaining *Δrelative repeatability* of SIFT on scene *forest* is sensitive to image properties measured by CSD, CLD, EHD and SCD (see Figure 6-31). Adding *SSAA* raises the similarity to *photo* by 4% and enlarges the weight of edge appearance (EHD) strongly, while its actual distance is shortened. It was expected that edge AA methods mostly affect edge based measures (EHD or HTD), but it seems SCD is far more impacted. In case of *SSAA* changes in colour distribution and colour layout lead to the observed performance shift.

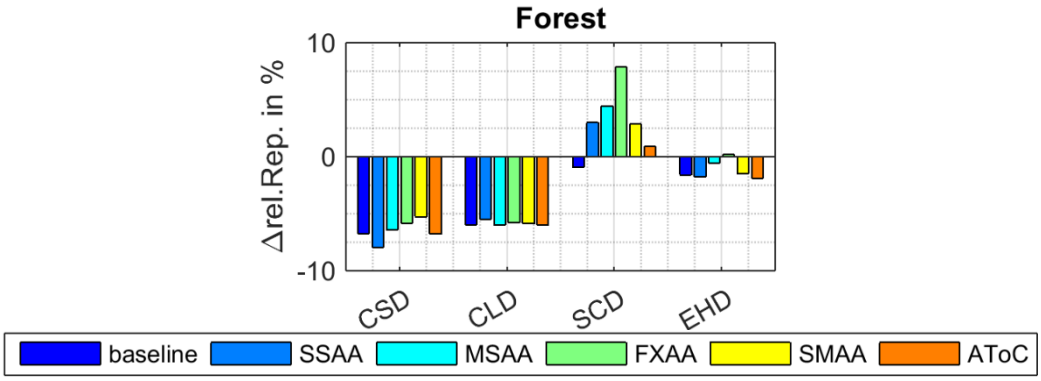


Figure 6-31: Influence of image content on SIFT *Δrelative repeatability* and their behaviour when "Edge" parameters are applied.

The high effect of other descriptors indicate that SIFT's performance is mainly not controlled by the appearance of edges. When adding *FXAA* the *Δrelative repeatability* is lowered by 5% while the impact of edges is pared down to being negligible, suggesting that the remaining edge difference is irrelevant to the performance of SIFT in *forest* . The usage of *SMAA* lessens *Δrepeatability* by 3% caused by its influences on colour distribution. Adding *AToC* (together with *MSAA*) lessens the distance to *photo* by 3% and slightly heightens the effect of edge appearance. All other scenes where only influenced by a maximum of 2% when using AA-methods, showing the benefit of using them is marginal when the scene is not heavily populated by environmental objects.

In Figure 6-32 it is shown that the regression model of SIFT *Δabsolute repeatability* is slightly more sensitive to antialiasing. In scene *forest, FXAA* lowers the performance by 9% and *SMAA* by 5% compared to *baseline*. The coefficients of the models remained mostly static for most tested techniques. In case of *forest* SIFT is by all measured image contents. The model shows that the effect of colour differences on the detection of SIFT features outweighs the influence of edge differences. This is also applies for *hangar* and *street*.

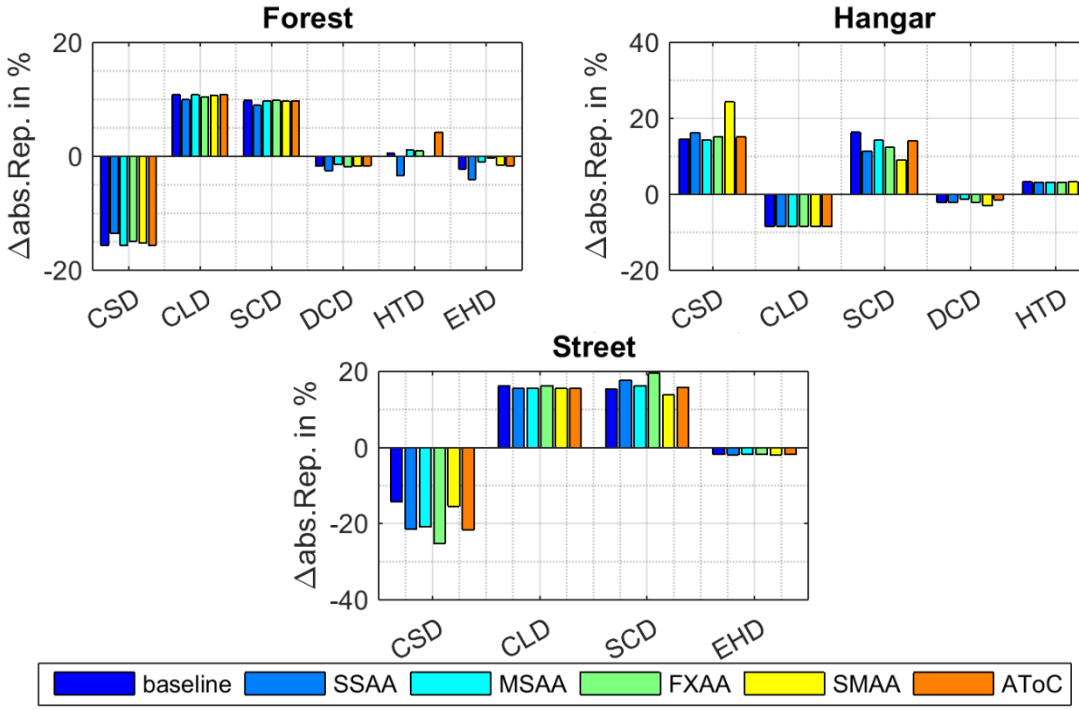


Figure 6-32: Influence of image content on SIFT *Δabsolute repeatability* and their behaviour when "Edge" parameters are applied.

Hangar benefits in case of *SSAA* and *SMAA* from antialiasing. Both lower the colour distribution when activated. *Street* is presented as a negative case where the scenes absolute

distance to the reference actually rises by 5% when method *FXAA* is applied. The model indicates this is also caused by changes in SCD.

MSER model coefficients

The effect of antialiasing methods on *Δrelative repeatability* of MSER is small. In fact, an effect on MSER's performance can only be observed in scene *concrete* (changes in *heath* and *hangar* were identified as outliers). Especially *MSAA* changes *Δrel. repeatability* caused by an increased impact of edge appearance (EHD) as shown in Figure 6-33.

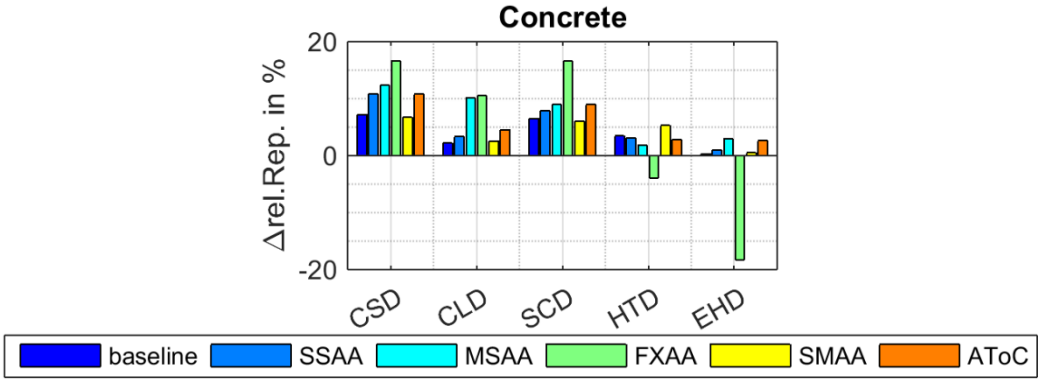


Figure 6-33: Influence of image content on MSER *Δrelative repeatability* and their behaviour when "Edge" parameters are applied.

The reduction of performance difference by 2% using *FXAA* is correlated with changes in CSD, CLD and SCD according to model. The shortened difference in *EHD* would actually enlarge the deviation. Enabling *AToC* positively influences colour structure, distribution and layout as well as edge appearance.

Analysing the regression models for *Δabsolute repeatability* of MSER (Figure 6-34) the effect of edge-based image properties (EHD; HTD) is apparent, except for scene *street*. Still, these image properties are less influential as colour based properties most notably colour layout.

The large values in given in the graphs for MSER *Δabsolute repeatability* exhibit the large impact a small deviation in distance can have on the result.

Scene *hangar* benefits from *FXAA* (SMAA is not considered due to outliers), which means it decreases the number of detected features in synthetic imagery. MSER's performance also benefits from *MSAA* and *AToC*, while *SSAA* increases the difference to *photo*. The interaction terms of the model present CLD as the most influential descriptor followed by HTD, SCD,

EHD and CSD. FXAA has mainly an impact on colour layout, while MSAA and AToC mainly influence edge appearance (though to a small degree).

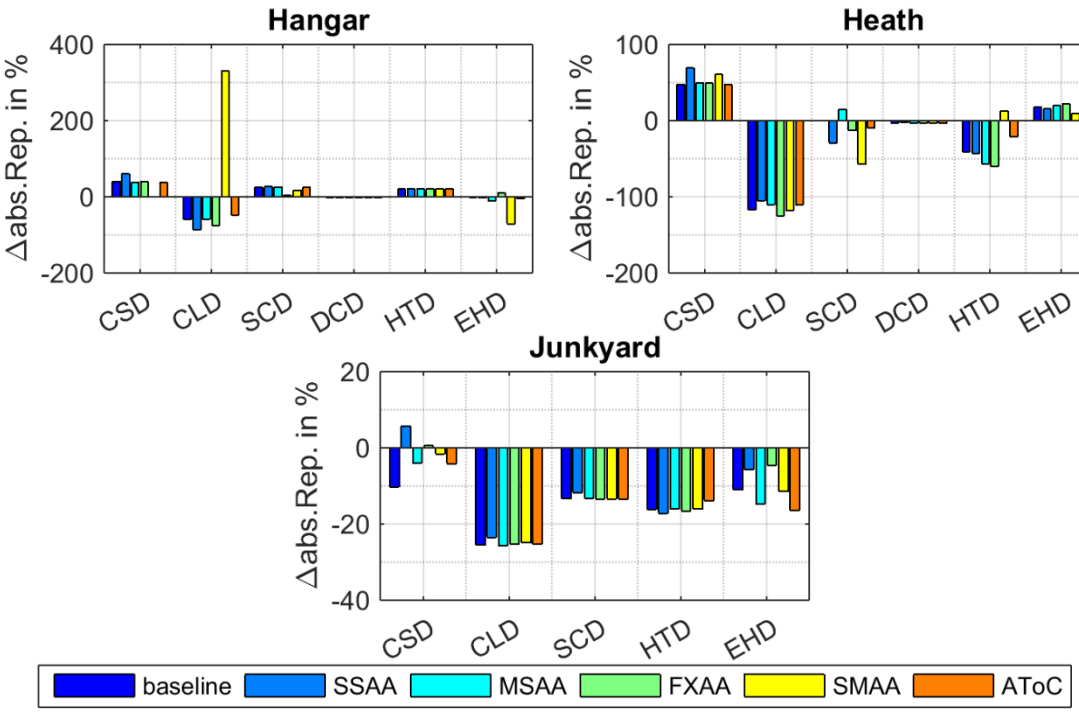


Figure 6-34: Influence of image content on MSER *Δabsolute repeatability* and their behaviour when "Edge" parameters are applied.

SSAA leads to a higher weight of CSD and CLD terms, while their distances remain constant. All other methods only slightly change the coefficient values.

In scene *heath*, all AA-methods except *FXAA* lower *Δabsolute repeatability* due to their impact on colour layout and homogeneous textures. *SSAA* mainly affects colour structure and layout. In addition, MSAA benefits mostly from non-edge-based distances demonstrating the large effect colour-based measures have on the performance. AToC lowers the influence of homogeneous textures similar to SMAA, which closes the remaining performance gap.

In scene *junkyard*, all model coefficients have a negative effect on the performance difference meaning a larger distance lowers the performance on synthetic data. The scene only benefits from enabled *AToC*, with a reduction of 7% in performance difference. This is caused changes in colour structure and repeating frequencies (HTD).

6.2.3.4 Summary of configuration set results

In the previous chapters, the benefit of antialiasing methods on synthetic data towards the performance of feature detectors compared to natural images was investigated. The edge difference is just one small component out of all deviations between synthetic and natural imagery. This is also true for the performance of feature detectors as the influence factor analysis revealed. Spatial colour layout or global colour histogram deviates impact the performance of both tested feature detectors more than edge differences (homogeneous and non-homogeneous). At first glance, this seems unintuitive, since feature detectors grey scale images before actual processing them. However, even after fusion of colour channels the distribution, layout and structure of their values still exist in the values of the grey scale images.

Feature detector SIFT profits from applying *FXAA* or *SMAA* using both performance metrics. In general, *SSAA* can slightly lower the *Δrelative repeatability* of MSER. However, *MSAA* should be preferred for urban scenes (many man-made objects) and *AToC* for rural scenes (mostly depicting trees and vegetation). *Δabsolute repeatability* on the other hand generally benefits slightly from *MSAA* and *AToC*.

When analysing image frequencies, *FXAA* lowers high image frequencies the most minimizing the frequency distribution differences of image types. *SMAA* slightly decreases the higher frequencies while *MSAA* and *AToC* show almost no effect. Out of all tested AA-methods, *SSAA* lowers the content distances to natural images the most (on CSD, CLD, SCD, DCD and EHD). *FXAA* robustly shortens the distances of CSD, CLD and EHD. *AToC* robustly lowers the colour distribution distance, in all other measures its effect is mostly small and scene dependent.

The last evaluation step correlates the performance changes between images to the image content changes to identify the causing image properties. The reduction of *Δrelative* and *Δabsolute repeatability* for SIFT when applying *FXAA* is mainly impacted by image content distances of SCD and EHD. When activating *SMAA* the performance of SIFT is affected by colour distribution and structure. *SSAA* additionally influences the colour layout of synthetic data. MSER only reacts to *MSAA* and *AToC* in some scenes when observing *Δrelative repeatability*. Here, the AA-methods force distance changes mainly in colour layout, structure and distribution. MSER's *Δabsolute repeatability* is reactive to changes in colour layout, homogeneous textures, edge appearance and colour structure.

6.2.4 Configuration set “3D-Objects”

Now the influence of 3D models on the terrain map is investigated. The three parameters are *objects high*, *no objects* and *modelling errors*. *Objects high* increases the polygon count and detail of geo-typical 3D-models. The geo-referenced models are modelled at only one level of detail. *No objects* removes all 3D-objects, leaving only the terrain map in the camera view. *Modelling errors*, changes the texture colour, scale or material of three geo-referenced buildings in scenes *hangar*, *house* and *junkyard*.

6.2.4.1 Object performance

SIFT’s performance results

Increasing the object polygon count obviously has no effect on SIFT for both performance measures as depicted in Figure 6-35. Eliminating objects from the scene has only a slight negative impact on the *relative* measure. Even though removing 3D-objects affects the *rel. repeatability* of scene *forest* strongly (a change of +28%) the deviance to *photo* remains similar at 15%. *Modelling errors* only affect scene *hangar* by raising *Δrelative repeatability*.

No objects lowers *Δabsolute repeatability* in scene *forest*, due to the larger count of detected feature pairs. After removal of 3D-objects in scene *concrete* the number of detected features drops. In this case, SIFT still detects more feature pairs in natural images. For scene *hangar*, *heath* and *house* the removal of objects is also non-beneficial.

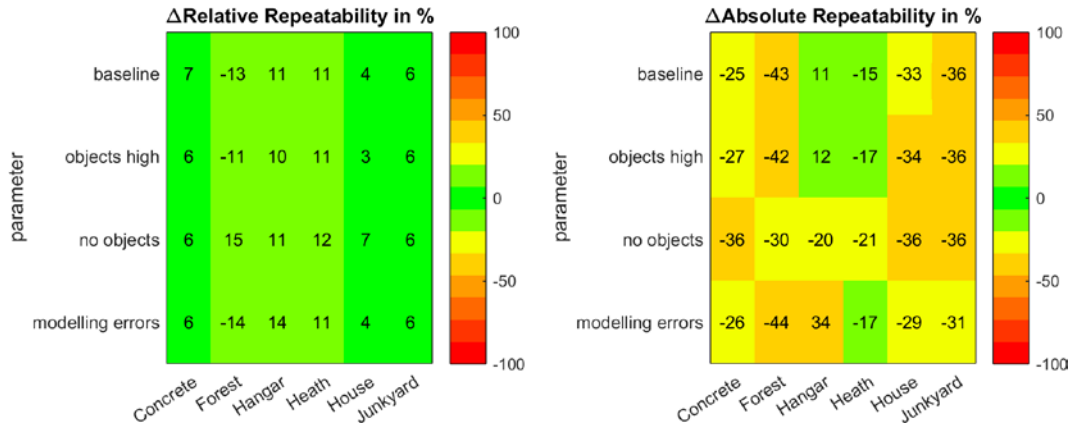


Figure 6-35: Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of SIFT on selected scenes and 3d object parameters.

The variation of textures to simulate *modelling errors* all increase the number of features detected in synthetic imagery. For example, the applied texture in *hangar* only was scaled to ensure a visual difference between synthetic and natural images. However, this already strongly affects the *absolute repeatability* of SIFT on synthetic data. In other scenes (*house* and *junkyard*), modification of roof textures consistently improves *Δabsolute repeatability*.

MSER's performance results

Evaluating *Δrelative repeatability* of MSER shows *object high* to have no notable effect as can be seen in Figure 6-36. It lowers the relative performance of MSER only slightly through all scenes. The *modelling errors* in *hangar* and *house* slightly affect the measure while remaining the same in *junkyard*.

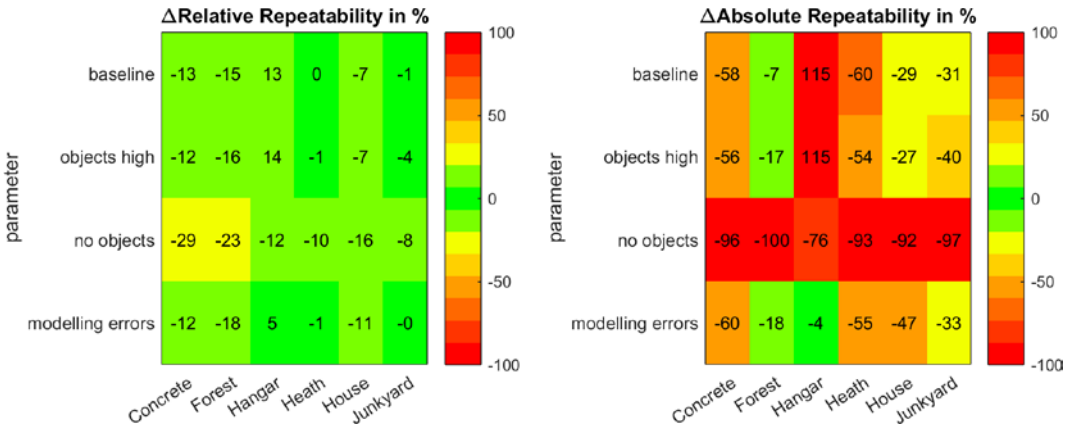


Figure 6-36: Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of MSER on selected scenes and 3d object parameters.

Increasing the *objects detail* of natural objects (found in *forest*, *heath* and *junkyard*) raises the *absolute repeatability* of MSER. Adding *modelling errors* leads to an almost identical performance of scene *hangar* compared to the reference. In *junkyard*, there is no difference and in *house* the number of features dropped during the period where the house is visible.

In general, increasing the detail of objects has no beneficial effect on the performance difference to the reference for any detector. SIFT's *Δrelative repeatability* is not affected by removal of 3D-objects. The *absolute repeatability* of SIFT and MSER's performance in general strongly deviate from their results on natural data. The parameter *modelling error* shows the influence textures have on the performance of feature detectors. Depending on the type of modification, the effect can be massively positive or negative. Thus, texturing needs to be conducted carefully.

6.2.4.2 Image content distances

Image content distances for the different parameters are displayed in Figure 6-37.

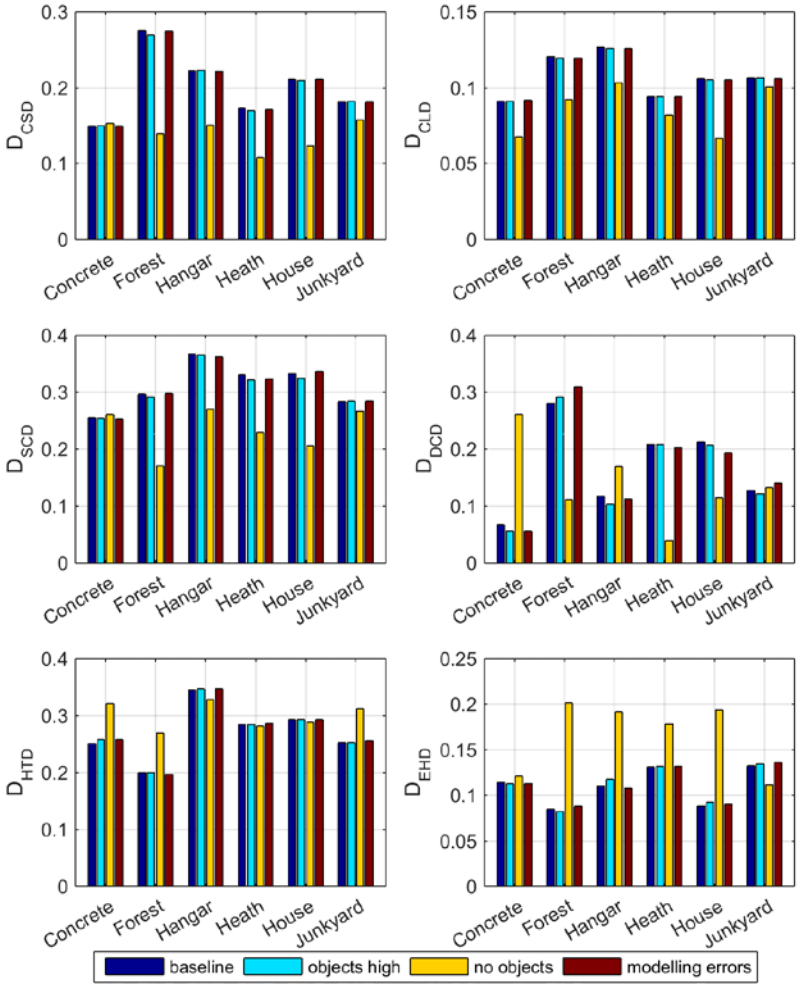


Figure 6-37: Median image content distances between synthetic imagery and *photo* for all used content descriptors and examined parameters.

Raising the polygon count of objects (*object high*) has a marginal effect on DCD and EHD distances depending on the scene. For example, in *forest* the dominant colour distance enlarge, while in scene *hangar* HTD the distance increases. Other content descriptors show no reaction. Now, *removing all objects* strongly reduces the distances of CSD, CLD and SCD. This shows that the colour palette of the satellite image fits the natural images better than the colours of used objects. On the other hand, the distance of DCD, HTD and EHD indicates that the most prominent colours and edge information relies on the presence of 3D-objects. *Modelling errors* lead to slight changes in HTD, DCD, SCD and CSD. This drastically demonstrates the influence a slightly modified roof texture can have on the image content of a scene.

6.2.4.3 Influence factor analysis

SIFT model coefficients

The model of SIFT *Δrel. repeatability* on *forest* is sensitive to CSD, CLD, EHD and SCD distances (Figure 6-38).

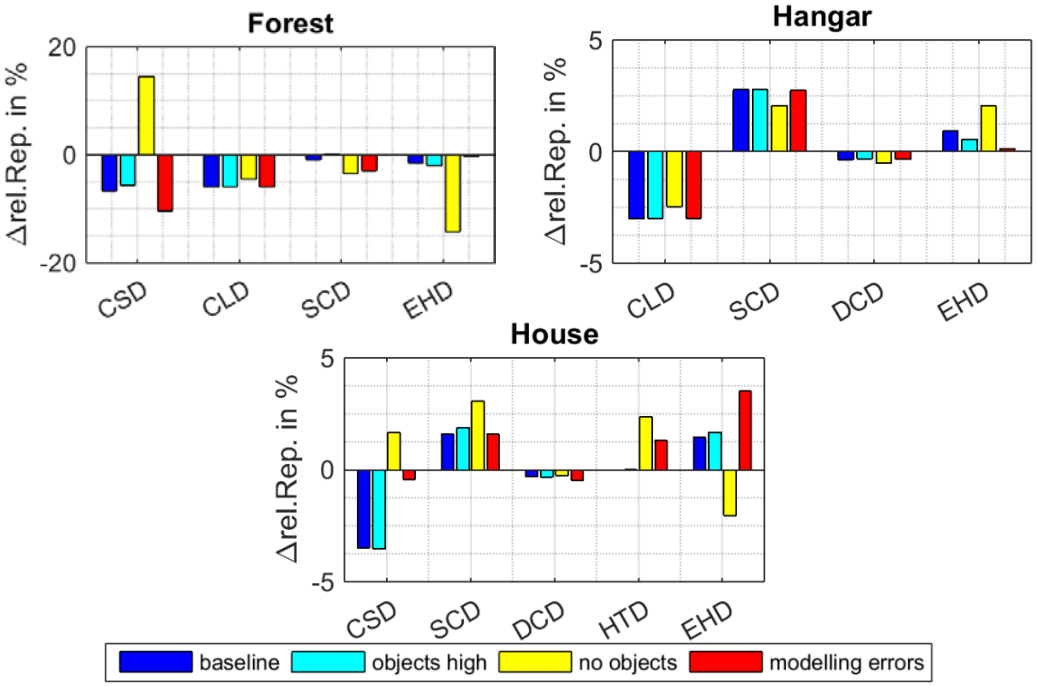


Figure 6-38: Influence of image content on SIFT *Δrelative repeatability* for scenes *forest, hangar* and *house* and their behaviour when "3D-Objects" parameters are applied.

Objects high marginally affects the terms of CSD, CLD and SCD leading to a performance difference reduction of 2%. *Removing 3D-objects* strongly enlarges the weight of CSD and EHD. The effects of *modelling errors* are marginal. The variation of the roof texture in scene *hangar* widens the existing performance gap to *photo* by 3% caused by slight changes in colour layout and dominant colours. In scene *house*, the *Δrel. repeatability* of SIFT is affected by *no objects* because of distance changes in colour structure (CSD), colour distribution (SCD) and homogeneous textures (HTD).

When measuring SIFT with *Δabsolute repeatability, objects high* notably affects no scene. This can be seen in Figure 6-35 presenting its same performance. In contrast, all scenes are affected by parameter *no objects (*Figure 6-39*)*. Here, e.g. *Δabsolute repeatability* of *forest* is decreased by 13% due to the influence of CSD and DCD distances. Scene *hangar* on the other hand negatively affected (9% higher performance difference) by the large change in colour information (SCD). The absolute performance difference in scene *house* is only slightly

changed by parameter *no objects* (3% higher) as well as *modelling errors* (4% lower). Removing the objects diminishes the colour layout of the scene, which SIFT is very sensitive to as can be seen in Figure 6-39. The enlarged distance in edge appearance, affect SIFT marginally in scene *house*, thus reducing the effect of EHD. Adding *modelling errors* to *house* affects the distances of CSD, CLD, EHD, DCD and lowers the performance difference.

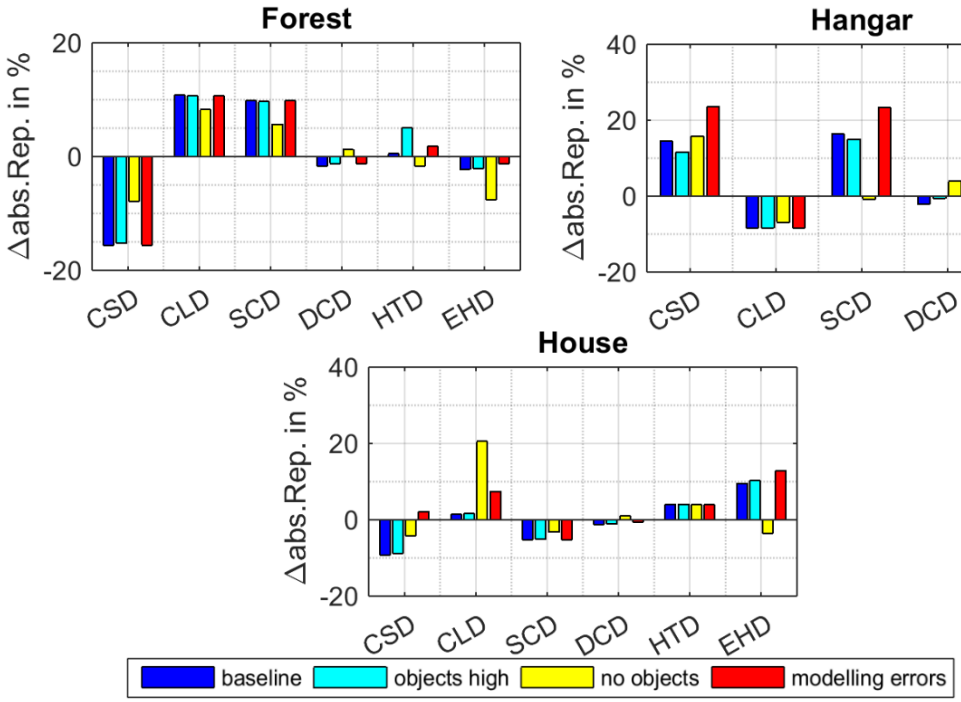


Figure 6-39: Influence of image content on SIFT *Δabsolute repeatability* for scenes *forest, hangar* and *house* and their behaviour when "3D-Objects" parameters are applied.

MSER model coefficients

Δrelative repeatability of MSER is insensitive to distance changes in EHD in scene *forest* since it was not fitted to the model as can be seen in Figure 6-40. Removing the trees in *forest* influences mainly colour layout and repetitive textures (HTD) and enlarges *Δrelative repeatability* of MSER by 16%. Here, the model actually indicates all changes would lead to a decrease in difference; however, MSER cannot perform robustly on the ground surface texture of *forest*. Thus, even though both measures show more similarity to natural images the new impact of the detail texture leads to low performance values.

On *hangar*, *modelling errors* lower *Δrelative performance* to only 5% induced by a smaller colour layout distance. Interestingly removing the objects affects the performance and the model coefficients only marginally. MSER's *Δrelative performance* on scene *house* is

enlarged by *no objects* (9% higher) and *modelling errors* (4% higher). Both are caused by distance changes in edge appearance (EHD and HTD), colour structure and colour layout.

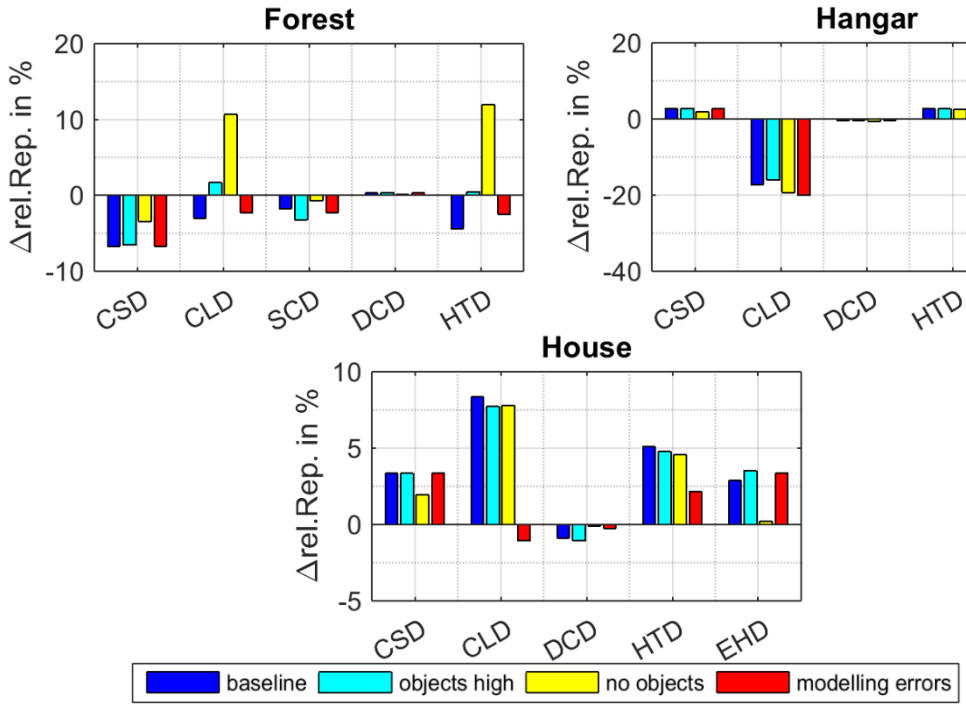


Figure 6-40: Influence of image content on MSER *Δrelative repeatability* for scenes *forest, hangar* and *house* and their behaviour when "3D-Objects" parameters are applied.

MSER's *Δabsolute repeatability* on *hangar* is explained by all descriptors (see Figure 6-41). Parameter *no objects* decreases the performance distance to *photo* by 39%, which results from changes in colour structure and distribution. The performance shift of parameter *modelling error* is additionally affected by edge appearance.

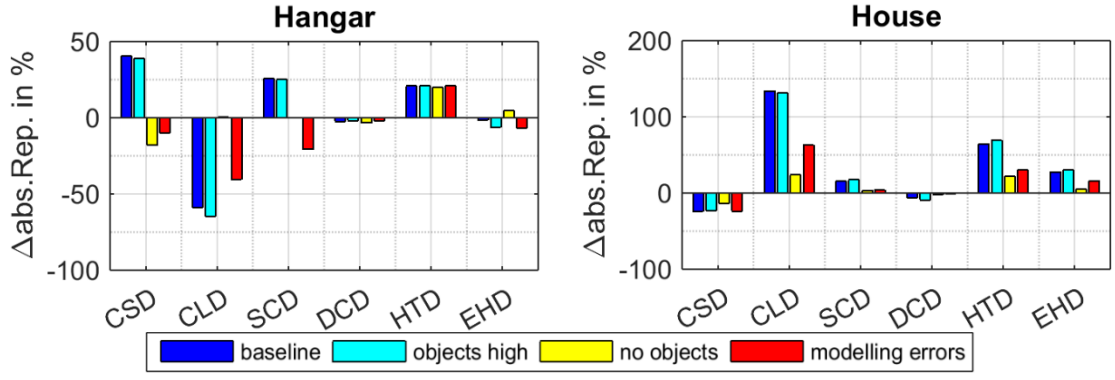


Figure 6-41: Influence of image content on MSER *Δabsolute repeatability* for scenes *forest, hangar, house* and *junkyard* and their behaviour when "3D-Objects" parameters are applied.

Removing objects in *house* enlarges *Δabsolute repeatability* to -93% caused by colour layout, colour distribution and edge appearance (HTD and EHD). A different roof texture raises the

performance difference to -47% due to changes in colour layout, repetitive texture and edge appearance.

6.2.4.4 Summary of configuration set results

In the previous chapters, the influence of "3D model" parameters on the performance of feature detectors and image content is analysed. The *Δrelative repeatability* of SIFT is marginally affected by removal of all objects, thus accepting a small reduction of similarity to *photo* allows very simple scene replications. However. The *Δabsolute repeatability* SIFT and MSER in general are measurably affected negatively by removing objects.

SIFT is generally negatively affected also by the introduced *modelling errors*. This highlights the impact textures have on the performance of feature detectors in synthetic data. Thus, reproductions of real world scenes need to focus on careful and correct texturing. Raising the polygon count of objects has generally no effect on SIFT's performance.

Additionally, the *Δabsolute repeatability* of MSER can also be impacted by the representation quality of objects (*objects high)*, though of negligible magnitude. The effect of *modelling errors* is scene dependant, but can be of large magnitude.

When analysing the image content, the changes introduced by *object high* and *modelling errors* are small. Increasing the polygon count of objects marginally affects the image distance of SCD, HTD and DCD. *Modelling errors* affect CSD, SCD, HTD and DCD measures depending on the type of error applied (rescaling, material change, colour change). Removing all objects reduces the distances of colour-based descriptors such as CSD, CLD and SCD while it boosts the distance of edge based descriptors such as HTD and EHD (additionally the DCD distance is raised in some cases).

The regression models identify majorly content distances in CSD, SCD, CLD and HTD as the cause explaining the drop in performance when objects are removed (*no objects*) for all detectors and metrics. However, edge appearance and dominant colours are also influencing the performance of feature detectors. Adding texture errors affects the absolute performance of SIFT due to distance changes in CSD, CLD, EHD and DCD. This differs for MSER only in the last distance descriptor, which is exchanged by HTD. The CSD and HTD differences explain the influence of *objects high* on MSER.

6.2.5 Configuration set “Camera model”

In this experiment the “Camera model” parameters influence is investigated. In detail the manipulation or addition of *noise*, *lens distortion*, *aperture*, *HDR*, *bloom* and *blur* are analysed. For more detail on these parameters, please refer to chapter 5.2.2.5.

Noise adds probabilistic colour noise using a Gaussian function with a std. deviation of 3σ to manipulate the original pixel value. *Lens distortion* adds the radial distortion of camera used in flight experiments to the image (only the first coefficient). *Aperture* simply closes the synthetic aperture, which results in darker images. *HDR* rearranges the existing colour depth to simulate HDR content, resulting in images that are more greyish. *Bloom* adds light bleeding effects existing on sensors to the synthetic image. *Blur* simply blurs the image (see chapter 5.2.2.5 for more details).

The properties are evaluated only on the four scenes *concrete*, *heath*, *house* and *street* representing for types of scenes to pare down the amount of results and redundancy. *Concrete* represents scenes populated with dense man-made objects; *Heath* natural objects (covering for *forest*) and *house* a mixture of both (covering for *hangar* and *junkyard*). Scene *street* represents a mostly texture based scene (almost no 3D-objects present; covering for *sport*).

6.2.5.1 Object performance

SIFT’s performance results

The *Δrelative repeatability* of SIFT is not affected by *HDR* and *bloom* as depicted in Figure 6-42. *Aperture*, *distortion* and *noise* slightly lessen the performance difference to *photo*. Adding *blur* expands the *Δrelative performance* on all scenes, which leads to lower similarity compared to the *photo* reference.

The *Δabsolute repeatability* of SIFT is not influenced by *HDR*, *bloom* and *lens distortion*. Changing the *aperture* leads to scene dependent results. While *street* profits, all other scenes are either not or negatively affected. *Blur* heightens the number of detected features on synthetic data for all scenes. While SIFT performs in *baseline* lower than the reference for the scenes *concrete*, *heath* and *house*, it already detects more features in synthetic data for scene *street*. Thus, except for *street*, *which* massively boosts the performance distance, all other scenes benefit from adding *blur*.

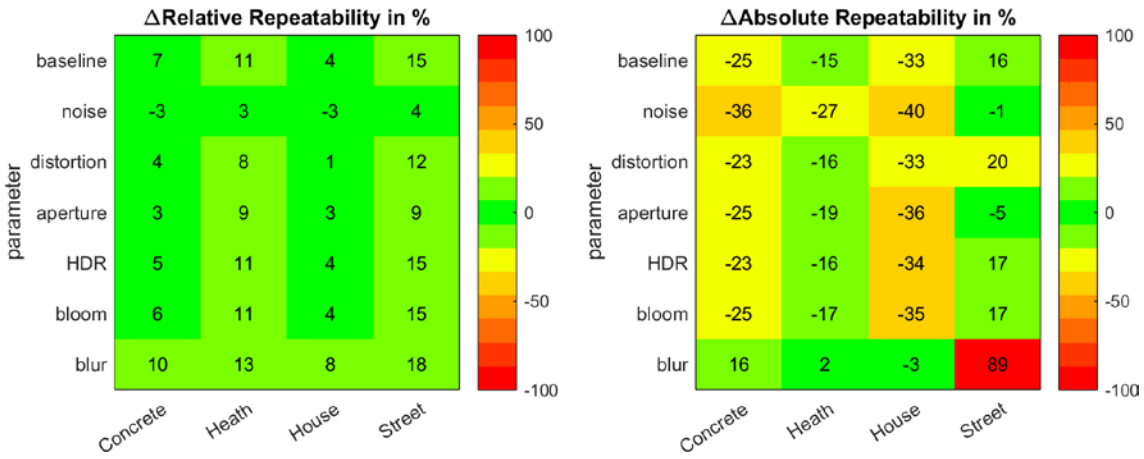


Figure 6-42: Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of SIFT on selected scenes and camera model parameters.

MSER's performance results

In Figure 6-43, the effect of camera model parameters on the performance of MSER is presented. *Δrelative performance* is negatively affected by *noise* (although only slightly). *HDR* does not seem to have an effect. All other parameters produce strongly scene-dependent results. For instance, *street* is negatively affected by *aperture*, *bloom* and *blur*, which all lower the contrast and thus the gradients in the image. The performance similarity of *heath* is dwindled when *blur*, *aperture* and *distortion* are applied, while being unaffected by *HDR* and *bloom*. Scene *house* is actually unaffected by all parameters. The very edge heavy scene *concrete* benefits from all parameters except *noise*.

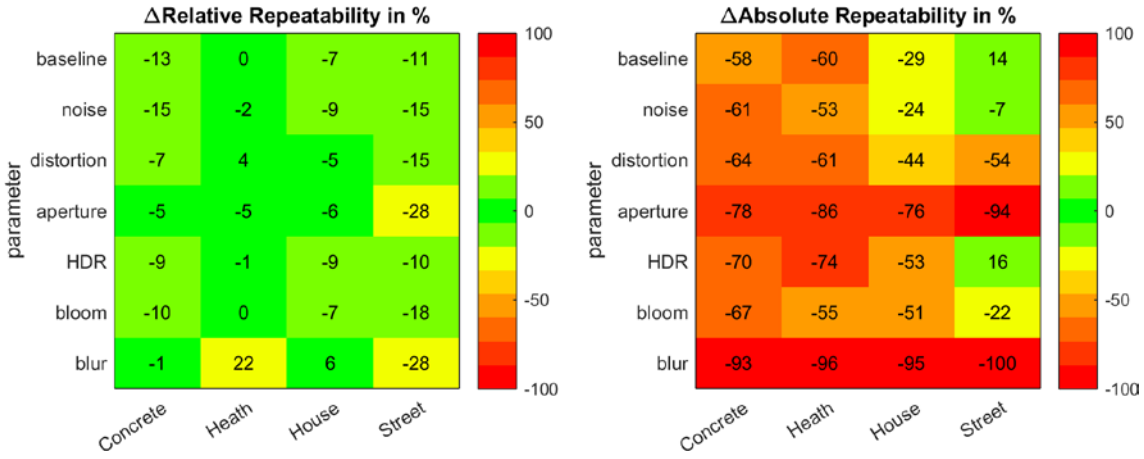


Figure 6-43: Colour coded lookup tables presenting *Δrelative* and *Δabsolute repeatability* of MSER on selected scenes and camera model parameters.

The *Δabsolute repeatability* of MSER in most cases is enlarged by all parameters with *blur* having the largest effect. Only scenes *heath*, *house* and *street* benefit from *noise*. Thus, *noise*

can boost the number of detected features for MSER at the cost of *Δrelative performance*. Showing the non-reliability of additionally detected features due to *noise*.

6.2.5.2 Image content distances

As shown in Figure 6-44 no image descriptor reacts to the introduction of *noise*. This is not surprising, because MPEG7 descriptors have been developed to find images of same content regardless of image quality differences. This could be considered in future investigations. *Distortion* slightly shortens the distances in colour (SCD, CSD and DCD) while raising location based measures such as EHD and CLD. A smaller *aperture* (darker image) lowers the distances of CSD, DCD and SCD indicating that the *baseline* images are too bright.

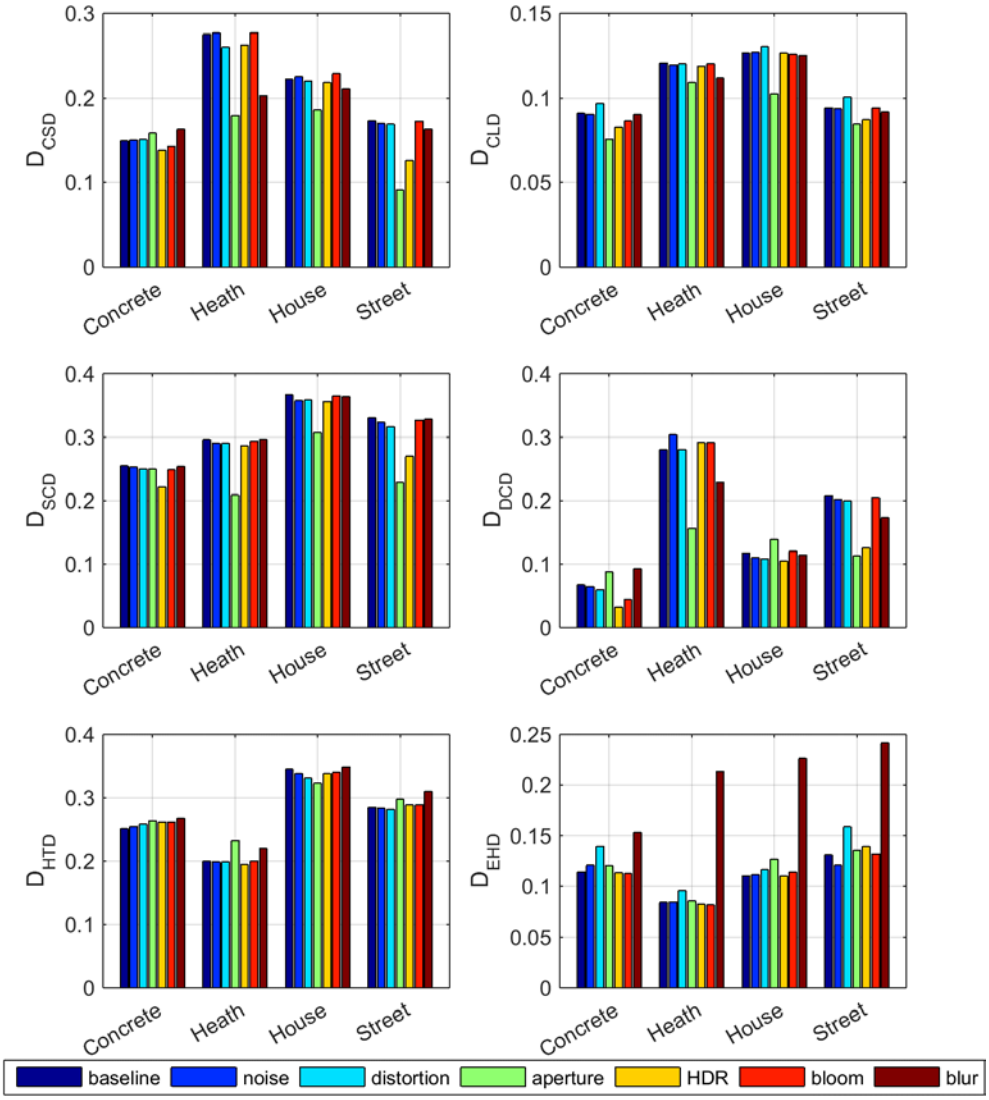


Figure 6-44: Median image content distances between synthetic imagery and *photo* for all used content descriptors and examined parameters.

In general, EHD distances rise, because darkening the image lessens the contrast of edges. Activating *HDR* increases the similarity of colour presentation (CSD, SCD and DCD), while edges remain mostly constant (HTD and EHD). *Bloom* also raises the colour similarity to photo but with less magnitude (CSD, SCD and DCD). *Blurring* the image weakens the gradients, which enlarges the distance of edge-based descriptors EHD and HTD but also influences the colour structure (CSD).

6.2.5.3 Influence factor analysis

SIFT model coefficients

The *Δrelative repeatability* of SIFT in scene *concrete* is explained by all image content measures except CSD (see Figure 6-45). The effect of the EHD distance is strongly raised by adding *noise*, *aperture* or *HDR* and beneficial to the performance similarity to *photo*. Adding *bloom* and *blur* lowers the impact of edges. The change in SCD distance is also causing reduction of performance difference for *noise* and *aperture*. HTD positively affects the performance of SIFT using configuration *aperture* and *bloom*.

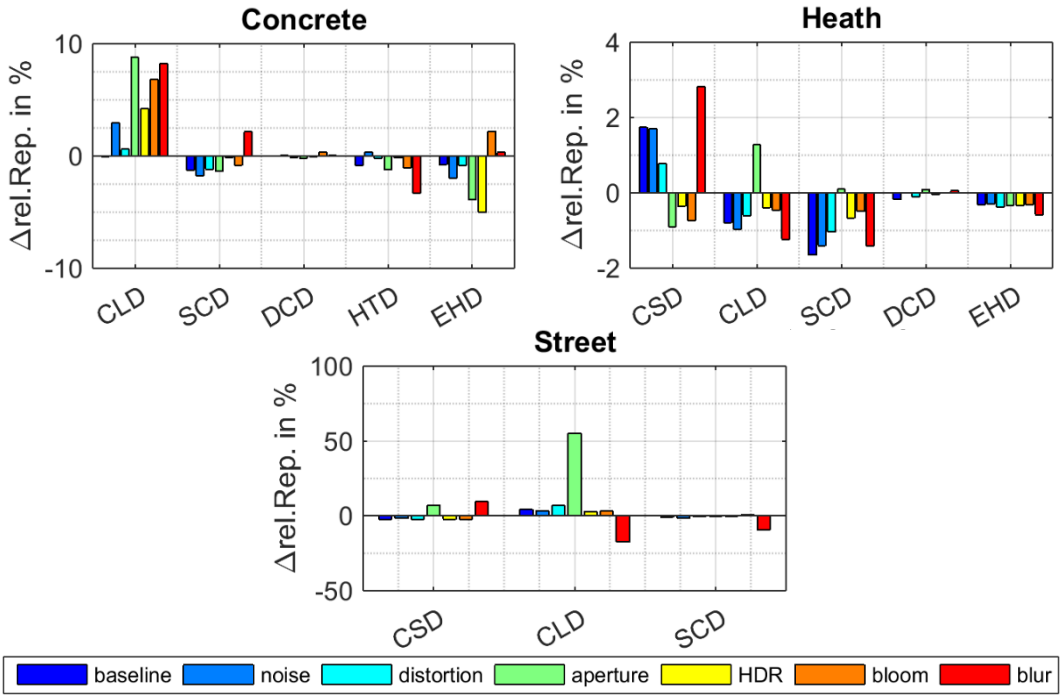


Figure 6-45: Influence of image content on SIFT *Δrelative repeatability* and their behaviour when "camera model" parameters are applied.

In scene *heath*, the performance change with added *noise* is induced by differences in CSD and CLD. Performance variations using *distortion* on the other hand are caused by colour

layout and edge appearance. The performance boost using *noise* on *street* stems from colour layout differences.

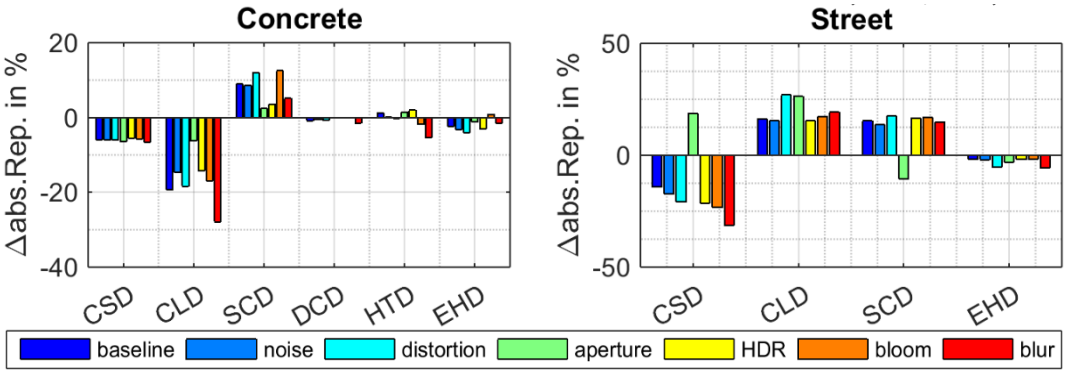


Figure 6-46: Influence of image content on SIFT *Δabsolute repeatability* and their behaviour when "camera model" parameters are applied.

The *Δabsolute repeatability* of SIFT (presented in Figure 6-46) in scene *concrete* improves for *blur* due to changes in edge presentation and *distortion* due to changes colour layout and distribution. *HDR* lowers the effect of all colour distance measures and trims the performance difference by 2%. The impact of *blurring* on edges lowers the difference to *photo* to 16%.

Noise reduces *Δabsolute repeatability* to -1% in scene *street*, which can be accredited to changes in colour structure, layout and distribution. In case of *aperture*, the influence of colour distribution on the performance indicate that images in *baseline* are brighter than the reference *photo*. Adding *blur* heightens SIFT's *Δabs. repeatability* due to its negative effect on colour layout (CLD).

MSER model coefficients

The *Δrelative repeatability* of MSER is presented in Figure 6-47. The performance of scene *concrete* is reduced by the parameters *distortion*, *aperture*, *HDR* and *blur* due to distance changes in CSD, CLD, HTD, and EHD. For instance, *distortion* profits from its effect on colour layout and repetitive textures. In scene *heath*, the performance is explained by the descriptors except for DCD. The *baseline* dataset of *heath* is actually performing equal to *photo*, which obviously cannot be improved by any parameter. Scene *street* is described by all six image content descriptors. Only *HDR* lowers the performance difference to *photo*, which is caused by raised influence on repetitive textures (HTD).

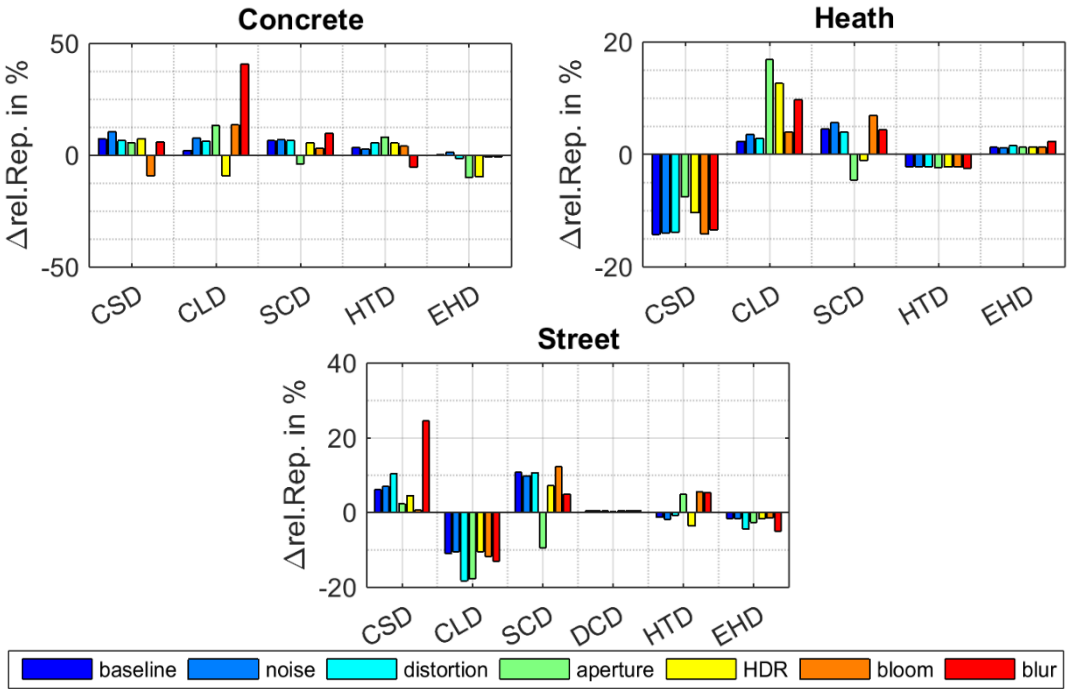


Figure 6-47: Influence of image content on MSER *Δrelative repeatability* and their behaviour when "camera model" parameters are applied.

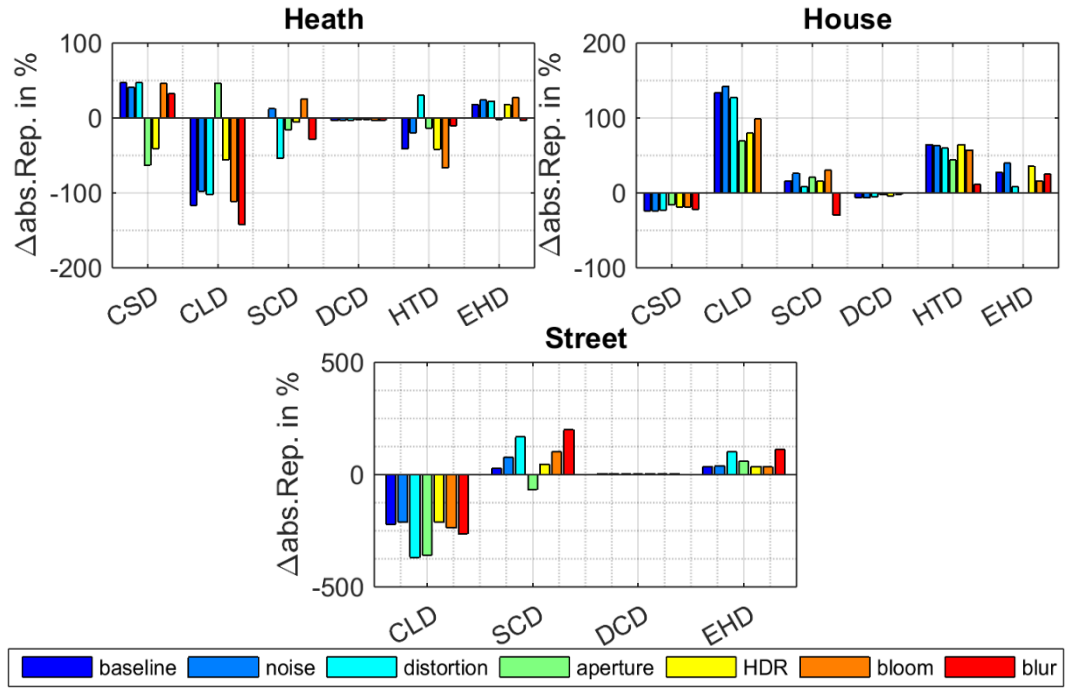


Figure 6-48: Influence of image content on MSER *Δabsolute repeatability* and their behaviour when "camera model" parameters are applied.

Δabsolute repeatability of MSER is described by all image content predictors (see Figure 6-48). *Blur* drastically expands *Δabsolute repeatability* and affects edge appearance, colour distribution and colour layout. Also *aperture* affects the performance of MSER negatively and causes changes in colour structure and layout as well as edge presentation (EHD; HTD).

Adding *distortion* affects the distances of SCD and EHD when analysed over all scenes. *HDR* has an impact on the SCD and CLD terms. *Bloom* mainly affects EHD, SCD and HTD distances. Overall, only *noise* reduces *Δabsolute repeatability* in *house*, *heath* and *street* stemming from its influence on colour layout, colour distribution and edge appearance.

6.2.5.4 Summary of configuration set results

In the previous chapters, the effect of *camera model* parameters on the performance of feature detectors was investigated.

The relative performance of SIFT is benefiting from enabling *noise*, *lens distortion* and *aperture* (smaller aperture). Adding *blur* negatively affects SIFT's performance. When analysing *Δabsolute repeatability* of SIFT, scenes with many environmental objects benefit from image *blurring*, while in texture-based scenes the performance heavily deteriorates. On the other hand, *noise* and *aperture* lower the performance of SIFT in object-heavy scenes. All others profit from these two configurations.

Δrelative repeatability MSER is independent to *noise* but reactive to *HDR* and *lens distortion*. Only scene *concrete* profits additionally *aperture* and *blur*. In case of *Δabsolute repeatability*, MSER benefits only from *noise*.

Afterwards, the image content changes introduced by these parameters were measured using image content descriptors. These reveal that *lens distortion* affects edges (HTD, EHD), colour layout (CLD) and colour composition (SCD). Darkening the image by reducing the size of the *aperture* affects the same image components as well as colour structure (CSD) and dominantly present colours (DCD). *HDR* lowers the distances to natural images for all colour-based content measures, while the distance of non-homogenous edges is increased. *Bloom* shortens the distances in colour structure and colour composition. Depending on the scene, it can also affect dominant colours. *Blur* drastically enlarges the edge and location based distances (EHD, HTD and CLD), while lowering the colour structure. This indicates that the colour structure of synthetic images is more spread than in natural images, most probably due to the influence of the detail texture. The Effect *noise* has on the image content is not sufficiently covered by the deployed descriptors, which indicates that additional descriptors would be beneficial.

The influence factor analysis combines the above investigation to identify, which of the observed image content differences is actually causing the performance difference of a feature detector. In general, the colour layout (CLD) is the most influential descriptor independent to parameter, metric or feature detector. *Noise* affects the performance of SIFT and MSER, however it is not sufficiently measured. Possible causes are its influence on CLD, SCD and EHD distances. SIFT and MSER are sensitive to changes in colour layout and edges (HTD and EHD). Thus, *blur* negatively affects the performance difference to natural images for all feature detectors. Introducing *lens distortion, noise and aperture* impacted SIFT positively in both performance measures, due to their effects on colour layout, colour structure and edge appearance. The camera model parameters mainly affected MSER by their induced changes in colour layout and homogeneous edges. Thus depending on the scene, *HDR* improve the *Δrelative repeatability* of MSER. The *Δabsolute repeatability* of MSER can be boosted by *noise*. The benefit of tested parameters is largely depending on the tested scenes and feature detector no parameter in this configuration set showed universal abilities to close the performance gap between synthetic and natural images.

7 Discussion and design recommendations

In chapter 1.2, the scientific question was decomposed into several objectives. This chapter mainly discusses whether this thesis was able to meet the three formulated objectives. To investigate them, the general concept was proposed as a framework in chapter 3.1. To prove the concept, it was applied to a specific use case (chapter 3.2-3.7) and implemented (chapter 4). This applied concept introduced some constraints. Their consequences will be discussed in chapter 7.1 together with the experiment results from chapter 6. Consequently, design recommendations based on these results are presented in chapter 7.2.

7.1 Discussion

The scientific question (chapter 1.2), whether “testing CV-algorithms with synthetic data generates results transferable to real world conditions” shall be answered using the concept proposed in chapter 3.1. It was designed to measure the performance on both image types, whether equal results have been achieved (Objective 1) and to identify their cause of difference (Objective 2). Before discussing the experiment results, the applied concept and its constraints are examined.

Applied concept

The applied concept (chapter 3.2) is used to validate the general approach (chapter 3.1). The constraints of the applied concept and its ability to answer the scientific question are discussed below (more detail can be found in chapter 3.2-3.7 and chapter 4).

Scenario: The concept demands synthetic and natural data depicting the same scene. Natural data was recorded using an available unmanned aerial aircraft, which set constraints on the scenario (environmental, e.g. daytime, sunny weather, acceptable wind speeds or location-based). The scenario was further simplified by deploying a perpendicular mounted camera (VIS-spectrum). The scenario had no moving entities. In this regard more complex scenes and differing environmental conditions are to be considered in future work.

Consequently, the acquired results are currently limited to the given constraints, but can be generalised for similar appearing scenes. The quality of the synthetic terrain database

modelling of the flight test area has been evaluated in chapter 5.1.1, which showed sufficient accuracy (close to the satellite imagery source).

Test object: The selected feature detectors (e.g. SIFT, MSER) were a valid choice for first experiments. Further evaluations need to investigate subsequent algorithms (descriptors and matching algorithms) to evaluate a full feature matching chain.

Performance measure: Similar to the test object, the well-known measures *relative* and *absolute repeatability* have been selected (Mikolajczyk et al., 2005). These measures demand the availability of ground truth, which is usually produced semi-automatically by assuming a homographic relationship between two measured subsequent images. This assumption limits the direction of the deployed camera to a top-down view to lower the influence of height. This is a clear limitation in the applied concept.

Datasets had been tested for violations of the homography assumption in chapter 5.1.2. The result was below 1px RMS in deviation, which according to (Mikolajczyk & Schmid, 2005) indicates no violation. Further, their original ground truth acquisition approach required an initial manual step. To allow larger datasets, a full automatic approach was proposed and implemented. Its evaluation in chapter 5.1.2 revealed an error of <1px RMS, which is definitely acceptable. The effect of the automatic approach was additionally evaluated against a dataset of (Mikolajczyk et al., 2005) in chapter 5.1.3. Here, a reduction of accuracy compared to the original semi-automatic approach was identified; however, the error remained below the threshold. Thus, the fully-automatically computed ground truth provided acceptable accuracy and was further used in the concept.

Image comparison: In chapter 3.5 ten methods have been selected to measure the difference between synthetic and natural data. A preliminary experiment evaluated their ability as image content measures (chapter 5.1.4). This experiment showed that MSE and PSNR prioritise spatial over content-related differences and thus have been excluded from the selection. NIQE and MSSIM on the other hand exhibited promising capabilities. However, these measures represent the full image and thus cannot fulfil the goal to identify specific image properties as the cause of performance differences. Therefore, these two were also excluded. The remaining capable measures were SCD, DCD, CLD, CSD, HTD and EHD.

Influence factor analysis: The algorithm performance results have been related to the image content differences using the statistical method *backward stepwise multiple linear regression*

analysis (see chapter 3.7). In this first evaluation, the nature of relation between predictors and outcome were unknown, therefore all relations have been assumed linear. This can be criticised as it might lead to poorly fitted models. Future experiments should expand the model or use a different method such as fractional factorial design of experiments (Box et al., 2005) to identify and measure interdependencies between predictors.

In summary, the concept in general proved to be valid and applicable. However, certain shortcomings exist (as described above) and need to be eliminated in future work.

Baseline experiment results

The baseline experiment (chapter 6.1) was used to validate the general concept. The synthetic imagery was captured with the virtual environment in its default configuration. The derived results are limited to the tested or similar operating feature detectors. Similarly, the gained content differences refer to synthetic images acquired using VBS3 (or similar local illumination engines). Still, the fundamental goal to validate the concept was achieved.

Test object performance results

This evaluation demonstrated that the concept successfully allowed quantification of performance differences between synthetic and natural data (chapter 6.1.1). It even has been shown that transferability (equal performance on both image types) is given in some explicit cases. This step was able to sufficiently determine the performance difference of a CV-algorithm between both image types, hence fulfilling the **Objective 1** (chapter 1.2).

Generally, feature detector SIFT achieved higher *relative repeatability* on synthetic images, while MSER's is higher on natural data. Both mostly deviate within a range of ±15% between both data types. The much more scene dependent *absolute repeatability* differs more strongly for both feature detectors (SIFT ±43%; MSER ±115%). Note that absolute repeatability is normed by number of detected feature pairs in natural images and thus can exceed 100%.

Image content difference results

The results in 6.1.2 show the successful quantification of appearance differences between synthetic and natural imagery. Results showed that in most scenes general colour distribution, dominant colours and homogeneous textures had the largest deviances.

Influence factor analysis results

Afterwards, the influence of these deviances on the performance of feature detectors has been computed. The results in 6.1.3 revealed the image content influencing the performance of tested feature detectors. In general, the performance is affected by image content differences in colour structure, colour layout and edge appearance (homogenous and arbitrary). The actual rank and amount of influence depends on the tested feature detector, the employed performance measure and the depicted scene. For instance differences in the edge histogram (EHD) highly affect the performance of SIFT, while MSER is only affected mildly.

Thus, the presented concept **allowed the determination of image attributes influencing the performance of tested CV-algorithms (Objective 2).** Adapting the respective image content enables the developer or programmer to improve the transferability of results efficiently by adjusting the image appearance of synthetic data.

Configuration set experiment results

The experiments described in chapter 6.2 were conducted to reason for rendering techniques capable to lower the remaining performance difference. The results of each chapter in 6.2 revealed the effect of techniques (beneficial, neutral or detrimental). These findings have been formulated into recommendations given in chapter 7.2.

In chapter 3 a set of image content measures were preselected for testing (Table 3-3). After the actual experiment in chapter 6.2 some assumptions on the sensitivity of image descriptors towards image characteristics could be confirmed, others were refuted (Table 7-1).

Table 7-1: Sensitivity of image descriptors towards image characteristics. (X = previous assumption)

Image Characteristics \ Image Descriptors	DCD	SCD	CSD	CLD	EHD	HTD
Blur (1/Clarity)					X	X
Noise		X	X	X	X	X
Geom. Lens Distortion					X	X
Modelling Detail	X		X	X	X	
Modelling Errors	X		X	X	X	
Aliasing					X	X
Aperture	X	X				
Texture Quality			X		X	X
Shadow	X		X	X	X	

Legend: High effect | Medium effect | Low effect | No effect

7.2 Design recommendations

Even though the results in chapter 6 are limited to the VBS3 rendering engine, they allowed the extraction of more general design recommendations (independent to a specific synthetic environment). This fulfils the last objective **(Objective 3)** given by chapter 1.2, showing that the presented approach is fully capable to answer, *how synthetic datasets need to be designed and generated for development of computer vision algorithms to achieve performance results transferable to real world conditions?*, within the discussed limitations.

The recommendations are sorted into fields of interest each directed towards a specific user. The given suggestions mainly apply to scenarios of airborne birds-eye view CV-applications using feature detectors, but engineers can also use them to extract requirements for their own virtual environment (having their use case in mind).

General recommendations for terrain generation and synthetic engine parametrisation

These suggestions are directed towards the modelling artist, database modeller or engine programmer designing the terrain database or configuring the rendering engine.

a) Carefully design textures

   Rational: Slightly wrong coloured or scaled textures and false materials applied to 3D-models in scenes can influence large performance changes as the results show in chapter 6.2.4. Therefore, prominently visible textures (e.g. roofs) should be carefully designed to replicate colour tone, frequency and material of the target scene.

b) Focus on texturing over (exact) 3D-modelling

   Rational: The results in chapter 6.2.2 and 6.2.4 reveal the heavy influence of textures deviating in colour or frequency. The effects of 3D-mesh (chapter 6.2.4) and edge appearance (chapter 6.2.3) are much lower.

c) Use multiplication based texture blending

   Rational: Image frequencies describe the steepness of gradients in the image (sharp edges = high image frequencies). To achieve similarity, the ground sample distance (GSD) of synthetic images needs to coincide with the natural images as has been shown in chapter

6.2.2. Since available satellite images are commonly of lower resolution as needed (here: 0.2mpp instead of 0.03mpp) high image frequencies are missing. Thus, virtual environments use procedural detail textures (e.g. concrete texture) and blend those with the ground texture to provide the needed detail (and image frequencies). However, texture-blending techniques needs to conserve the contrast of the satellite image. Thus, blending textures by multiplication is suggested.

d) Use antialiasing methods

Rational: Natural images are slightly blurred making high image frequencies less common. In synthetic images, the rendering process produces by default high gradients on edges as has been measured in chapter 6.2.3. These can be decreased using antialiasing methods. Out of all methods tested in the configuration set “edge”, fast approximate antialiasing (FXAA) reduces this effect most robust and effectively. However, depending on the feature detector the most preferable antialiasing technique can vary (see list item j)).

e) Use shadow drawing and filtering

Rational: Activating shadow drawing in the virtual environment trims the local colour and edge differences and the performance differences of feature detectors between both image types as the results show in chapter 6.2.1. Shadow filtering (PCF) further cuts the performance differences, due to antialiasing of edges induced by shadow drawing.

f) Negligible rendering techniques

Rational: Chapter 6.2 revealed that several rendering methods did generally not affect the performance of feature detectors:

- *Screen Space Ambient Occlusion* (SSAO) as implemented in VBS3.
- *Anisotropic filtering* (AF) as implemented in VBS3.
- Enabling the *HDR effect* (rise of colour depth during rendering followed by rescaling to screen capabilities) reduced the colour content distances, but did not affect the performance differences (chapter 6.2.5).
- Adding a *bloom* effect (oversaturation of bright image areas) does not affect SIFT.

g) Avoid detrimental rendering techniques

Rational: The experiments showed that some techniques these enlarge the performance difference in most cases (obviously, some exceptions exist):

- *Blurring* of the synthetic image (see chapter 6.2.5).
- *Lowering the texture resolution* (procedural or detailed) in synthetic images (starting with Equal GSD on natural and synthetic data; see chapter 6.2.2).
- *Removal of 3D-objects* (see chapter 6.2.4).

h) Adapt the terrain database to the capabilities of the sensor

Rational: Reducing the texture quality or the aperture (darkening of the image) are steps depending on the current modelling state of the synthetic scene and setting of the sensor (chapter 6.2.2 or 6.2.5). In these cases, the developer is advised to adjust the resolution of textures or the brightness of the synthetic images in compliance with natural data.

Detector-based recommendations

These recommendations shall help algorithm developers in selecting the most suitable feature detector whenever synthetic datasets shall be used. This chapter also provides hints towards optimizing the rendering engine, which depend on the deployed CV-algorithm.

i) Preferably use SIFT (over MSER) when possible

Rational: The results in chapter 6.1 show that SIFT is in general the better performing feature detector for both measured metrics and on both image data types. In general, MSER usually performs low on synthetic data and is very scene dependent. Influencing the colour structure or colour layout might improves these facts. Thus, MSER is biased towards natural data, while SIFT slightly overperforms on synthetic data. Consequently, general performance and lower performance difference of SIFT make it the recommended feature detector (of the two tested) for airborne CV-applications and deployment on synthetic data.

j) Use feature detector dependent rendering techniques

Rational: As can be seen in chapter 6.2 the two tested feature detectors can react very differently on different rendering techniques.

When **SIFT** is given as detector the application of antialiasing (FXAA or SMAA), lens distortion, slight noise, slight blur, smaller aperture, changing the texture blending and adding filtered shadows lowers the difference to natural images in the given experiments. These indicate that SIFT benefits from deploying a camera model.

When **MSER** is given as detector the application of antialiasing (MSAA or AToC) and filtered shadows are beneficial to lower its performance difference. Further, the influence of the detail texture should be cut down by changing the texture blending method.

k) Removal of 3D-objects does not (majorly) influence the relative repeatability of SIFT

Rational: In specific circumstances (see chapter 6.2.4) the performance of SIFT is almost robust (~3% deviance) to the removal of all 3D-objects in the scene. If this reduction of comparability to natural data is acceptable, the effort to model the environment decreases massively. This result is limited to similar airborne scenarios using relative repeatability as performance metric and SIFT as feature detector. All other tested combinations are negatively affected by the removal of 3D-objects.

l) Use high contrast textures

Rational: The synthetic images in this evaluation were low in contrast due to the texture blending method employed in the synthetic environment. Generally, all feature detectors are depending on strong gradients inside an image. Therefore, low contrast images naturally lead to lower performances. This affects MSER more heavily than SIFT as can be seen in scene *sport* (chapter 6.1.1 and 6.2.2). On the other hand, natural and synthetic images of cluttered scenes such as *heath* or *junkyard* MSER performs more equally on compared to SIFT (albeit lower).

Scenario-based recommendations

All scenes in this work are limited to the use case of birds-eye view aerial photography. In this work, the following scene types have been considered:

- *Urban*: 3D-objects and edge heavy scenes (*concrete* and *hangar*)
- *Infrastructure*: Mainly background with one dominating entity such as *street* or *sport*
- *Forest*: Dense placement of trees or vegetation

- *Rural*: Mainly background with occasionally placed trees, houses and other man-made objects (*house*, *heath* and junkyard)

Whenever high performance of feature detectors is important (and the performance deviance is of lower importance), SIFT is the better choice for any type of scene.

m) For *urban* scenes use SIFT

Rational: In urban scenes SIFT has already low performance differences between both image types. Here, additionally to the above mentioned parameters SSAA and HDR should be enabled.

n) For *infrastructure* scenes use MSER

Rational: When scenes mainly comprise infrastructure then MSER achieves smaller performance differences, closely followed by SIFT. When using MSER shadow drawing should be enabled. On the other hand, SIFT benefits from activating anisotropic filtering and lowering the satellite image resolution. In general, the overall poor performance of scene *sport* indicates modelling problems (low contrast and resolution of running track).

o) In *forest*-like scenes it depends

Rational: The investigation showed that on scenes of type *forest* the smallest Δ*relative* repeatability is achieved by SIFT. Activating *shadows* and *shadow filtering* closes the relative performance difference for both feature detectors. Using Δ*absolute repeatability* MSER is the closest performing detector. The similarity of the datasets can be raised for SIFT by activating *SSAA*, *HDR*, *Objects high*, and *texture low*. MSER benefits from activating *texture low* and *noise*. Due to the high density of natural objects the actual ground texture is not visible, thus setting texture quality to low affects only tree textures. Reducing these makes them more similar to their natural counterparts.

p) For rural scenes it does not matter

Rational: In rural scenes, the performance differences between the image types are equal for both feature detectors. When deploying SIFT, the similarity between datatypes is increased by downscaling the texture quality. When MSER is used *anisotropic filtering* and *SSAA* can help to lower existing performance differences.

8 Summary

Prototyping of airborne CV-algorithms poses problems due to limited availability of test data. Using synthetic environments to generate the necessary data may mitigate this issue. However, the question arises how results derived on such datasets can be transferred to the natural environment.

The current state-of-the-art of using computer generated imagery with computer vision and related transferability considerations have been presented. In this regard, a procedural concept has been developed which allows comparing rendered images and photographs based on algorithm performance and image content. It further identifies the sensitivity of the algorithm towards specific image content. The concept characterizes the CV-algorithm of interest and identifies deficiencies and abundances existing in synthetic images compared to their natural equal. Its applicability has been investigated in an airborne remote sensing example testing three different feature detectors. This investigation validated the capabilities of the concept and allowed the derivation of design recommendations.

SIFT, SURF and MSER have been selected as example test algorithm as they are well understood, used in many applications and regularly constitute the first step in a CV-processing-chain interfacing signal and feature domain. The necessary natural image content was then sampled during test flights. After modelling the equivalent geographical setting in a synthetic environment, the flight was replicated and the corresponding synthetic image data were acquired. After that, the performance differences between synthetic and natural images have been measured using the well-known performance measures for feature detectors *relative* and *absolute repeatability*. The *image content distances* between rendered images and photographs were evaluated afterwards using MPEG7 image descriptors. Since *image content distances* may lead to performance differences, both have been used to fit *regression models*. The resulting *standardized coefficients* then provided information about the influence of specific image content on the performance of the tested algorithm. The results derived from the test datasets show in general that the synthetic images yield a performance within a range of 15% equal to natural scenes when evaluating the *relative performance* of feature detectors. The absolute number of features detected thereby differs up to 115%.

After having identified the basic differences, the rendering pipeline and the content of the synthetic environment has been varied to isolate the impact of specific rendering methods.

The results show that synthetic data can profit greatly from adding *filtered shadows* and *specific antialiasing methods*. The application of techniques such as *SSAO*, *AF*, *HDR* or *bloom* did not affect the results and thus can be neglected. Further textures applied to the virtual scene contribute clearly the performance as a result it is deemed necessary to render at the same ground sample distance. In addition *brightness*, *contrast*, *colour tone*, *frequency* and *material* of prominent textures need to be modelled closely to their natural equivalent. To obtain the necessary ground sample distance at low altitude high-resolution satellite textures (0.2mpp) had to be blended with (procedural) detail textures. The blending of such textures should be conducted by *multiplication*, thereby preserving the contrast of the satellite texture. Respective findings providing hints for database modellers or rendering engine programmers have been formulated and derived. For instance, it has been derived that changes in textures present in the synthetic environment influence the performance of the algorithms much higher than changes in the 3D-mesh of entities and thus these textures need to be carefully designed. Further, the use of antialiasing (smoothing of edges) is encouraged as it allows edges to appear more natural. However, the specific method to be deployed depends on the used test algorithm.

9 Prospects

This thesis provides a novel concept to measure the differences between natural and synthetic images and their impact on the performance of tested CV-algorithms objectively. A long-term goal is identifying specifications for synthetic environments based on underlying principles defining image qualities that produce functional realism for computer vision methods at low implementation and modelling effort. In this respect, several extensions to the demonstrated paradigm can be proposed.

As has been explained, the current ground truth method cannot be used for perspective scene due to the homography-constraint. Creating performance measures and ground truth based on multi-view geometry may allow evaluations of perspective scenarios to enhance the range of the concept.

This work focused on evaluating the performance differences of feature d on natural and synthetic images. However, the concept also allows the analysis of more complex algorithms such as object detectors, trackers or image registration whether they are provided as closed- or open-source. However, it is important to apply performance criteria appropriate for the tested algorithm.

The current image content evaluation should be extended by additional very specific image quality measures for noise, contrast or brightness to enhance the characterization of content and CV-algorithms.

The influence factor analysis applied relies on linear regression models to relate content differences to CV-algorithm performance. Using non-linear regression models may characterizes such better. Also currently, inter-dependencies between parameters (e.g. texture resolution and antialiasing) are not considered. Now after having identified basic relations, switching to a fractional factorial design of experiments approach could help to unravel inter-dependencies.

References

AFRL. (2007). CLIF 2007 dataset over Ohio State University. Retrieved March 3, 2015, from https://www.sdms.afrl.af.mil/index.php?collection=clif2007

Agrawal, M., Konolige, K., & Blas, M. R. (2008). CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. In Computer Vision–ECCV 2008. *Computer Vision -- ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV, 5305 LNCS*, 102–115.

Alexiadis, V., Colyar, J., Halkias, J., Hranac, R., & McHale, G. (2004). The next generation simulation program. *ITE Journal (Institute of Transportation Engineers)*, *74*(8), 22–26.

Avcibaş, I., Sankur, B., & Sayood, K. (2002). Statistical evaluation of image quality measures. *Journal of Electronic Imaging*, *11*(2), 206–223.

Backhaus, K., Erichson, B., Plinke, W., & Weiber, R. (2006). *Multivariate Analysemethoden. Springer-Lehrbuch* (12th ed.). Berlin/Heidelberg: Springer-Verlag.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., & Szeliski, R. (2010). A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, *92*(1), 1–31.

Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of Optical Flow Techniques. *International Journal of Computer Vision*, *12*(1), 43–77.

Bastan, M., Cam, H., Gudukbay, U., & Ulusoy, O. (2010). Bilvideo-7: an MPEG-7- compatible video indexing and retrieval system. *IEEE MultiMedia*, *17*(3), 62–73.

Bavoil, L., & Sainz, M. (2008). *Screen space ambient occlusion*. *NVIDIA White Paper*.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. *Computer Vision -- ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I, 3951 LNCS*, 404–417.

Belsley, D. a. (1991). A guide to using the collinearity diagnostics. *Computer Science in Economics and Management*, *4*(1), 33–50.

Bender, M., & Brill, M. (2003). *Computergrafik* (1st ed.). Hanser.

Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., & Silva, C. T. (2013). A Benchmark for Surface Reconstruction. *ACM Transactions on Graphics*, *32*(2), 20:1–20:17.

Blender Institute. (2010). Premiere animated film “Sintel” at Netherlands Film Festival. Retrieved June 16, 2017, from http://download.blender.org/durian/Sintel-premiere-pressrelease.pdf

Blinn, J., Greenberg, D. P., Hagen, M. A., Feiner, S., & Mackinlay, J. (1988). Designing Effective Pictures : Is Photographic Realism the Only Answer? (Panel Session). In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques* (Vol. 22, pp. 351–399). New York, NY, USA: ACM.

Bober, M. (2001). MPEG-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, *11*(6), 716–719.

Boehm, F., & Schulte, A. (2012). Scalable COTS Based Data Processing and Distribution Architecture for UAV Technology Demonstrators. In *European Telemetry and Test Conference (ETC) 2012*.

Boehm, F., & Schulte, A. (2013). Air to ground sensor data distribution using IEEE802.11N Wi-Fi network. In *2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*. IEEE.

Böhm, F., & Schulte, A. (2012). UAV Autonomy Research – Challenges and Advantages of a Fully Distributed System Architecture. *International Telemetering Conference Proceedings*, 1–10.

Boulenguez, P., Airieau, B., Larabi, M., & Meneveaux, D. (2012). Towards a perceptual quality metric for computer-generated images. In *Image Quality and System Performance IX* (Vol. SPIE 8293, p. 82930K–82930K–10).

Bourdis, N., Marraud, D., & Sahbi, H. (2011). Constrained optical flow for aerial image change detection. In *2011 IEEE International Geoscience and Remote Sensing Symposium* (pp. 4176–4179). IEEE.

Bowyer, K. W., & Phillips, P. J. (1998). Overview of work in empirical evaluation of computer vision algorithms. In *Empirical Evaluation Techniques in Computer Vision* (pp. 1–11). IEEE Computer Society.

Box, G. E. P., Hunter, J. S., & Hunter, W. G. (2005). *Statistics for Experimenters* (2nd ed.). Hoboken, New Jersey: John Wiley and Sons, Inc.

Breckon, T. P., Barnes, S. E., Eichner, M. L., & Wahren, K. (2009). Autonomous Real-time Vehicle Detection from a Medium-Level UAV. In *Proc. 24th International Conference on Unmanned Air Vehicle Systems* (pp. 29.1–29.9).

Brook, A., & Ben-Dor, E. (2011). Automatic Registration of Airborne and Spaceborne Images by Topology Map Matching with SURF Processor Algorithm. *Remote Sensing*, *3*(1), 65–82.

Brosius, F. (2013). Distanz- und Ähnlichkeitsmaße. In *Spss 21* (1st ed., pp. 693–709). MITP VERLAG.

Bundesministerium für Verkehr Bau und Stadtentwicklung. Gemeinsame Grundsätze des Bundes und der Länder für die Erteilung der Erlaubnis zum Aufstieg von unbemannten Luftfahrtsystemen gemäß § 16 Absatz 1 Nummer 7 Luftverkehrs-Ordnung (LuftVO), 161 Nachrichten für Luftfahrer § (2012). Bonn.

Bunnell, M., & Pelacini, F. (2004). Shadow Map Antialiasing. In *GPU Gems*. Addison-Wesley Professional.

Butler, D. J., Wulff, J., Stanley, G. B., & Black, M. J. (2012). A Naturalistic Open Source Movie for Optical Flow Evaluation. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer Vision -- ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI* (pp. 611–625). Berlin, Heidelberg: Springer Berlin Heidelberg.

Butler, D. J., Wulff, J., Stanley, G., & Black, M. (2012). *Mpi-Sintel Optical Flow Benchmark: Supplemental Material*.

Buturovic, A. (2005). MPEG 7 Color Structure Descriptor for visual information retrieval project VizIR 1. Citeseer.

Cha, S. (2007). Comprehensive Survey on Distance / Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, *1*(4), 300–307.

Choi, Y., Won, C. S., Ro, Y. M., & Manjunath, B. S. (2002). Texture Descriptors. In *Introduction to MPEG-7* (pp. 213–229). West Sussex: John Wiley & Sons, Ltd.

Cieplinski, L., Kim, M., Ohm, J.-R., Pickering, M., & Yamada, A. (2000). *Information Technology – Multimedia Content Description Interface – Part 3: Visual*.

Clark, P. J. (1952). An extension of the coefficient of divergence for use with multiple characters. *Copeia*, *1952*(2), 61–64.

Clauss, S., & Schulte, A. (2014). Implications for operator interactions in an agent supervisory control relationship. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 703–714). Orlando, FL, USA: IEEE.

Cohen, B., & Byrne, J. (2009). Inertial aided SIFT for time to collision estimation. In *2009 IEEE International Conference on Robotics and Automation* (pp. 1613–1614). Kobe, Japan: IEEE.

Cohen, J. (1992). A power primer. *Quantitative Methods in Psychology*, *112*(1), 155–159.

Cohen, J., Olano, M., & Manocha, D. (1998). Appearance-preserving Simplification. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques SIGGRAPH 98* (Vol. 32, pp. 115–122). New York, NY, USA: ACM.

Collins, R., Zhou, X., & Teh, S. K. (2005). An open source tracking testbed and evaluation web site. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 17–24.

Cook, R. D., & Weisberg, S. (1982). *Residuals and Influence in Regression*. New York: Chapman and Hall.

Daly, S. J. (1992). Visible differences predictor: an algorithm for the assessment of image fidelity. *Human Vision, Visual Processing, and Digital Display III*, *1666 SPIE*(3), 2–15.

De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L. (2000). The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, *50*(1), 1–18.

Delepoulle, S., Bigand, A., & Renaud, C. (2012). A no-reference computer-generated images quality metric and its application to denoising. In *2012 6th IEEE International Conference Intelligent Systems* (pp. 67–73). Sofia,Bulgaria: IEEE.

Demonceaux, C., Vasseur, P., & Pègard, C. (2007). UAV attitude computation by omnidirectional vision in urban environment. In *Proceedings 2007 IEEE International Conference on Robotics and Automation* (pp. 2017–2022). Rome, Italy: IEEE.

Deng, Y., Manjunath, B. S., Kenney, C., Moore, M. S., & Shin, H. (2001). An efficient color representation for image retrieval. *IEEE Transactions on Image Processing*, *10*(1), 140–147.

Dimitrov, R. (2007). *Cascaded shadow maps*. *Developer Documentation, NVIDIA Corporation*. Santa Clara, CA, USA.

Donnelly, W., & Lauritzen, A. (2006). Variance Shadow Maps. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games* (pp. 161–165). New York, NY, USA: ACM.

Downton, A., & Crookes, D. (1998). Parallel architectures for image processing. *Electronics & Communications Engineering Journal*, *10*(3), 139–151.

Dreschler, L., & Nagel, H.-H. (1982). Volumetric model and 3D trajectory of a moving car derived from monocular TV frame sequences of a street scence. *Computer Graphics and Image Processing*, *20*(3), 199–228.

Eidenberger, H. (2003a). A new perspective on visual information retrieval. In M. M. Yeung, R. W. Lienhart, & C.-S. Li (Eds.), *Storage and Retrieval Methods and Applications for Multimedia 2004* (Vol. 5307 SPIE, pp. 133–144). San Jose, CA, USA: International Society for Optics and Photonics.

Eidenberger, H. (2003b). Distance Measures for MPEG-7-based Retrieval. In *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval* (p. 130). New York, NY, USA: ACM Press.

Ellis, T. (2002). Performance metrics and methods for tracking in surveillance. In J. M. Ferryman (Ed.), *3rd IEEE Workshop on Performance Evaluation of Tracking and Surveillance* (pp. 26–31). Copenhagen, Denmark: IEEE Computer Society.

Farooque, M. A., & Rohankar, J. S. (2013). Survey on Various Noises and Techniques for Denoising the Color Image. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, *2*(11), 217–221.

Ferwerda, J. A. (2003). Three varieties of realism in computer graphics. In *Electronic Imaging 2003* (Vol. SPIE 5007, pp. 290–297). International Society for Optics and Photonics.

Ferwerda, J. A., & Pellacini, F. (2003). Functional difference predictors (FDPs): measuring meaningful image differences. *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, *2*, 1388–1392.

Ferwerda, J. A., Ramanarayanan, G., Walter, B., & Bala, K. (2008). Visual equivalence: an object-based approach to image quality. *Color and Imaging Conference*, *2008*(1), 347–354.

Field, A. (2009). *Discovering Statistics using SPSS* (3rd ed.). London, UK: SAGE Publications Ltd.

Fischler, M. A., & Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, *24*, 381–395.

Freund, R. J., & Littell, R. C. (2000). *SAS system for regression* (3rd ed.). Sas Institute.

Frew, E., McGee, T., Kim, Z., Xiao, X., Jackson, S., Morimoto, M., … Sengupta, R. (2004). Vision-based road-following using a small autonomous aircraft. In *2004 IEEE Aerospace Conference Proceedings* (Vol. 5, pp. 3006–3015). Big Sky, MT, USA: IEEE.

Friendly, M., & Kwan, E. (2009). Where's Waldo? Visualizing Collinearity Diagnostics. *The American Statistician*, *63*(1), 56–65.

Garratt, M. A., & Chahl, J. S. (2008). Vision Based Terrain Following for an Unmanned Rotorcraft. *Journal of Field Robotics*, *25*(4-5), 284–301.

Gini, R., Pagliari, D., Passoni, D., Pinto, L., Sona, G., & Dosso, P. (2013). Uav Photogrammetry : Block Triangulation Comparisons. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XL-1/W2*, 155–162.

Goshtasby, A. A. (2012). Similarity and Dissimilarity Measures. In *Image Registration* (1st ed., pp. 7–66). London: Springer-Verlag London.

Grana, C., & Cucchiara, R. (2006). Performance of the MPEG-7 Shape Spectrum Descriptor for 3D Objects Retrieval. *Proceedings of the 2nd Italian Research Conference on Digital Library Management Systems*, 3–6.

Granlund, G., Nordberg, K., Wiklund, J., Doherty, P., Skarman, E., & Sandewall, E. (2000). Witas: An intelligent autonomous aircraft using active vision. *Proceedings of the UAV 2000 International Technical Conference and Exhibition (UAV)*.

Grapinet, M., De Souza, P., Smal, J.-C., & Blosseville, J.-M. (2012). Characterization and Simulation of Optical Sensors. *Procedia - Social and Behavioral Sciences*, *48*, 962–971.

Gross, D. C. (1999). Report from the Fidelity Implementation Study Group. In *Fall Simulation Interoperability Workshop Papers* (p. 88). Orlando.

Gschwandtner, M., Kwitt, R., Uhl, A., & Pree, W. (2011). BlenSor: Blender sensor simulation toolbox. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kyungnam, … J. Ming (Eds.), *Advances in Visual*

Computing: 7th International Symposium, ISVC 2011, Las Vegas, NV, USA, September 26-28, 2011. Proceedings, Part II (Vol. 6939 LNCS, pp. 199–208). Berlin, Heidelberg: Springer Berlin Heidelberg.

Haeusler, R., & Klette, R. (2010). Benchmarking stereo data (not the matching algorithms). In *Pattern Recognition: 32nd DAGM Symposium, Darmstadt, Germany, September 22-24, 2010. Proceedings* (Vol. 6376 LNCS, pp. 383–392). Berlin, Heidelberg: Springer Berlin Heidelberg.

Haralick, R., Shanmugan, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-3*(6), 610–621.

Harris, C., & Stephens, M. (1988). A Combined Corner and Edge Detector. *Proceedings of the 4th Alvey Vision Conference*, 147–151.

Hendriks, F., Tideman, M., Pelders, R., Bours, R., & Liu, X. (2010). Development tools for active safety systems: Prescan and VeHIL. *Proceedings of 2010 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2010*, 54–58.

Herzog, R., Čadík, M., Aydčin, T. O., Kim, K. I., Myszkowski, K., & Seidel, H.-P. (2012). NoRM: No-Reference Image Quality Metric for Realistic Image Synthesis. In *Computer Graphics Forum* (Vol. 31, pp. 545–554). Chichester, UK: John Wiley & Sons, Ltd.

Hoffman, N., & Preetham, A. J. (2002). Rendering Outdoor Light Scattering in Real Time. In *Game Developer Conference*. San Jose.

Hoogendoorn, S. P., & Schreuder, M. (2005). Tracing Congestion Dynamics with Remote Sensing. *Transportation Research Board Annual Meeting 2005*.

Hoogs, A., & Hackett, D. (1995). Model-Supported Exploitation as a Framework for Image Understanding, 1–4.

Horé, A., & Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. *2010 20th International Conference on Pattern Recognition (ICPR)*, 2366–2369.

Hranac, R. (2004). *NGSIM Task E . 3 : High-Level Data Plan*. *Federal Highway Administration*. Cambridge, MA, USA.

Hummel, G., Kovács, L., Stütz, P., & Szirányi, T. (2012). Data Simulation and Testing of Visual Algorithms in Synthetic Environments for Security Sensor Networks. In N. Aschenbruck, P. Martini, M. Meier, & J. Tölle (Eds.), *Future Security: 7th Security Research Conference, Future Security 2012, Bonn, Germany, September 4-6, 2012. Proceedings* (Vol. 318, pp. 212–215). Bonn: Springer Berlin Heidelberg.

Hummel, G., & Stütz, P. (2011). Conceptual design of a simulation test bed for ad-hoc sensor networks based on a serious gaming environment. In *International Training and Education Conference (ITEC 2011)*. Cologne.

Hummel, G., & Stütz, P. (2014). Using Virtual Simulation Environments for Development and Qualification of UAV Perceptive Capabilities: Comparison of Real and Rendered Imagery with MPEG7 Image Descriptors. In J. Hodicky (Ed.), *Modelling and Simulation for Autonomous Systems: First International Workshop, MESAS 2014, Rome, Italy* (Vol. 8906, pp. 27–43). Rome: Springer International Publishing.

Hummel, G., & Stütz, P. (2015). Evaluation of Synthetically Generated Airborne Image Datasets using Feature Detectors as Performance Metric. In H. R. Arabnia, L. Deligiannidis, & F. G. Tinetti (Eds.), *IPCV 2015* (pp. 231–237). Las Vegas: CSREA Press.

Institut für Normung. (1990). DIN 9300-2:1990-10 - Aerospace; concepts, quantities and symbols for flight dynamics; motions of the aircraft and the atmosphere relative to the earth; ISO 1151-2:1985 (status as of 1987) modified. Beuth Verlag GmbH.

Irgenfried, S., Dittrich, F., & Wörn, H. (2014). Realization and evaluation of image processing tasks based on synthetic sensor data: 2 use cases. In *Forum Bildverarbeitung 2014* (pp. 35–46). Karlsruhe.

Itoh, M., & Shishido, Y. (2008). Fisher information metric and Poisson kernels. *Differential Geometry and Its Application*, *26*(4), 347–356.

Jimenez, J., Echevarria, J. I., Sousa, T., & Gutierrez, D. (2012). SMAA: Enhanced subpixel morphological antialiasing. *Computer Graphics Forum*, *31*(2), 355–364.

Ke, Y., Tang, X., & Jing, F. (2006). The Design of High-Level Features for Photo Quality Assessment. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*, *1*, 419–426.

Kondermann, D. (2013). Ground Truth Design Principles: An Overview. In *Proceedings of the International Workshop on Video and Image Ground Truth in Computer Vision Applications* (pp. 5:1–5:4). New York, NY, USA: ACM.

Kudelka, M. J. (2012). Image Quality Assessment. In J. Pavlu & J. Safrankova (Eds.), *WDS'12 Proceedings of Contributed Papers: Part I – Mathematics and Computer Sciences* (Vol. 1, pp. 94–99). Prague: Matfyzpress.

Kundu, D., & Evans, B. L. (2014). Spatial Domain Synthetic Scene Statistics. In *2014 48th Asilomar Conference on Signals, Systems and Computers* (pp. 948–954). IEEE.

Kundu, D., & Evans, B. L. (2015a). Full-Reference Visual Quality Assessment for Synthetic Images: A Subjective Study. In *2015 IEEE International Conference on Image Processing (ICIP)* (pp. 2374–2378). Quebec City, Canada: IEEE.

Kundu, D., & Evans, B. L. (2015b). No-reference Synthetic Image Quality Assessment using Scene Statistics. In *2015 49th Asilomar Conference on Signals, Systems and Computers* (pp. 1579–1583). Pacific Grove, CA, USA: IEEE.

Larson, E. C., & Chandler, D. M. (2010). Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, *19*(1), 011006.

Lenc, K., Gulshan, V., & Vedaldi, A. (2012). VLBenchmarks. Retrieved March 25, 2015, from http://www.vlfeat.org/benchmarks/

Lillesand, T. M., Kiefer, R. W., & Chipman, J. W. (2015a). Applications of Remote Sensing. In *Remote Sensing and Image Interpretation* (7th ed., pp. 609–698). New York, NY, USA: John Wiley and Sons, Inc.

Lillesand, T. M., Kiefer, R. W., & Chipman, J. W. (2015b). Concepts and Foundations of Remote Sensing. In *Remote Sensing and Image Interpretation* (7th ed., pp. 1–85). New York, NY, USA: John Wiley and Sons, Inc.

Loncaric, S. (1998). A survey of shape analysis techniques. *Pattern Recognition*, *31*(8), 983–1001.

Longhurst, P., Ledda, P., & Chalmers, A. (2003). Psychophysically based artistic techniques for increased perceived realism of virtual environments. In *Proceedings of the 2nd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa* (pp. 123–132). New York, NY, USA: ACM.

Lowe, D. G. (2004). Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Lubin, J., & Fibush, D. (1997). Sarnoff JND vision model. IEEE.

Luo, W., Wang, X., & Tang, X. (2011). Content-based photo quality assessment. In *2011 IEEE International Conference on Computer Vision (ICCV)* (pp. 2206–2213). IEEE.

Ma, W. Y., Deng, Y., & Manjunath, B. S. (1997). Tools for texture- and color-based search of images. In B. E. Rogowitz & T. N. Pappas (Eds.), *Human Vision and Electronic Imaging II* (Vol. SPIE 3016, pp. 496–507). San Jose, CA, USA: International Society for Optics and Photonics.

Mahalanobis, P. C. (1936). On the generalised distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*.

Manjunath, B. S., Ohm, J.-R., Vasudevan, V. V., & Yamada, A. (2001). Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, *11*(6), 703–715.

Manjunath, B. S., Wu, P., Newsam, S., & Shin, H. D. (2000). A texture descriptor for browsing and similarity retrieval. *Signal Processing: Image Communication*, *16*(1), 33–43.

Martull, S., Peris, M., & Fukui, K. (2012). Realistic CG Stereo Image Dataset with Ground Truth Disparity Maps. In *Technical Committee on Pattern Recognition and Media Understanding* (Vol. 111, pp. 117–118). The Institute of Electronics, Information and Communication Engineers (IEICE).

Marziliano, P., Dufaux, F., Winkler, S., & Ebrahimi, T. (2002). A no-reference perceptual blur metric. In *Proceedings. International Conference on Image Processing* (Vol. 3, pp. III–57–III–60). Rochester, NY, USA: IEEE.

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2002). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In D. Marshall & P. L. Rosin (Eds.), *Proceedings of the British Machine Vision Conference* (pp. 384–393). BMVA Press.

McCane, B., Novins, K., Crannitch, D., & Galvin, B. (2001). On Benchmarking Optical Flow. *Computer Vision and Image Understanding*, *84*(1), 126–143.

McGraw-Hill, S. P. P. (2002). *McGraw-Hill Dictionary of Scientific & Technical Terms* (6th ed.). McGraw-Hill Education.

Meehl, P. E. (1997). The problem is epistemology, not statistics: Replace significance tests by confidence intervals and quantify accuracy of risky numerical predictions. In L. L. Harlow, S. A. Mulaik, & J. H. Steiger (Eds.), *What if there were no significance tests?* (pp. 393–425). Mahwah, NJ, USA: Erlbaum.

Meister, S., & Kondermann, D. (2011). Real versus realistically rendered scenes for optical flow evaluation. In *2011 14th ITG Conference on Electronic Media Technology* (pp. 1–6). Dortmund, Germany: IEEE.

Meister, S. N. R. (2014). *On Creating Reference Data for Performance Analysis in Image Processing*. Ruperto-Carola University of Heidelberg.

Michel, C. (2000). Cardinal, nominal or ordinal similarity measures in comparative evaluation of information retrieval process. *Second International Conference on Language Resources and Evaluation*, 1509– 1513.

Mikolajczyk, K., & Schmid, C. (2002). An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I* (p. 15). London, UK: Springer-Verlag.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), 1615–1630.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., … Van Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, *65*(1), 43–72.

Mittal, A., Soundararajan, R., & Bovik, A. C. (2013). Making a “completely blind” image quality analyzer. *IEEE Signal Processing Letters*, *20*(3), 209–212.

Mori, M., MacDorman, K. F., & Kageki, N. (2012). The uncanny valley. *IEEE Robotics and Automation Magazine*, *19*(2), 98–100.

Motion Picture Expert Group. (2003). Information Technology - Multimedia Content Description Interface - Part 6: Reference Software. In *ISO/IEC JTC 1 SC29* (Vol. 15938–6:20).

Murphy, D., & Cycon, J. (1999). Applications for mini VTOL UAV for law enforcement. In E. M. Carapezza & D. B. Law (Eds.), *Sensors, C3I, Information, and Training Technologies for Law Enforcement* (Vol. SPIE 3577, pp. 35–43). International Society for Optical Engineering.

Nabi, S., Balike, M., Allen, J., & Rzemien, K. (2004). An Overview of Hardware-In-the-Loop Testing Systems at Visteon. *SAE Technical Paper*.

Nakamura, Y., Matsuura, T., Satoh, K., & Ohta, Y. (1996). Occlusion detectable stereo - occlusion patterns in camera matrix. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 371–378). San Francisco, CA, USA: IEEE.

Nejadasl, F. K., Gorte, B. G. H., & Hoogendoorn, S. P. (2006). Robust vehicle tracking in video images being taken from a helicopter. *Procedings, International Society for Photogrammetry and Remote Sensing Commission VII Mid-Term Symposium*, 8–11.

Nentwig, M., Miegler, M., & Stamminger, M. (2012). Concerning the applicability of computer graphics for the evaluation of image processing algorithms. In *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)* (pp. 205–210). IEEE.

Nentwig, M., & Stamminger, M. (2010). A method for the reproduction of vehicle test drives for the simulation based evaluation of image processing algorithms. In *13th International IEEE Conference on Intelligent Transportation Systems* (pp. 1307–1312). IEEE.

Nentwig, M., & Stamminger, M. (2011). Hardware-in-the-loop testing of computer vision based driver assistance systems. In *2011 IEEE Intelligent Vehicles Symposium (IV)* (pp. 339–344). IEEE.

Nex, F., Gerke, M., Remondino, F., Przybilla, H.-J., Bäumker, M., & Zurhorst, A. (2015). Isprs Benchmark for Multi-Platform Photogrammetry. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *II-3/W4*, 135–142.

Nicodemus, F. E. (1965). Directional Reflectance and Emissivity of an Opaque Surface. *Applied Optics*, *4*(7), 767–775.

Niethammer, U., Rothmund, S., Schwaderer, U., Zeman, J., & Joswig, M. (2011). Open source image-processing tools for low-cost UAV-based landslide investigations. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *38*(1).

North Atlantic Treaty Organization. (2005). *NATO Intelligence, Surveillance, and Reconnaissance (ISR) Interoperability Architecture (NIIA) VOLUME 1: Architecture Description* (1st ed., Vol. 1).

Oelbaum, T. (2008). *Design and Verification of Video Quality Metrics*. Technical University Munich.

Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C. C., Lee, J. T., … Desai, M. (2011). AVSS 2011 demo session: A large-scale benchmark dataset for event recognition in surveillance video. In *2011 8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)* (pp. 527–528). IEEE.

Ohm, J.-R., Cieplinski, L., Kim, Heon J. Krishnamachari, S., Manjunath, B. S., Messing, D. S., & Yamada, A. (2002). Color Descriptors. In *Introduction to MPEG-7* (pp. 187–212). West Sussex: John Wiley & Sons, Ltd.

Ollero, A., Lacroix, S., Merino, L., Gancet, J., Wiklund, J., Remuss, V., … Caballero, F. (2005). Multiple eyes in the skies - Architecture and perception issues in the comets unmanned air vehicles project. *IEEE Robotics & Automation Magazine*, *12*(2), 46–57.

Phillips, P. J., Hyeonjoon Moon, Rizvi, S. a., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(10), 1090–1104.

Phillips, P. J., Martin, A., Wilson, C. L., & Przybocki, M. (2000). Introduction to evaluating biometric systems. *Computer*, *33*(2), 56–63.

Phong, B. T. (1975). Illumination for computer generated pictures. *Communications of the ACM*, *18*(6), 311–317.

Pitteway, M. L. V. (1967). Algorithm for drawing ellipses or hyperbolae with a digital plotter. *The Computer Journal*, *10*(3), 282–289.

Plöger, F. W. (JAPCC). (2010). *Strategic Concept of Employment for Unmanned Aircraft Systems in NATO*. Kalkar, Germany.

Quaritsch, M., Kruggl, K., Wischounig-Strucl, D., Bhattacharya, S., Shah, M., & Rinner, B. (2010). Networked UAVs as aerial sensor network for disaster management applications. *E & I Elektrotechnik Und Informationstechnik*, *127*(3), 56–63.

Rahman, M., Bhattacharya, P., & Desai, B. C. (2005). Probabilistic Similarity Measures in Image Databases with SVM Based Categorization and Relevance Feedback. In *Image Analysis and Recognition: Second International Conference, ICIAR 2005, Toronto, Canada, September 28-30, 2005. Proceedings* (Vol. LNCS 3656, pp. 601–608). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ramesh, V. (1995). *Performance Characterization of Image Understanding Algorithms*. University of Washington.

Ro, Y. M., & Yoo, K. (1999). Texture featuring and indexing using matching pursuit in Radon space. In *Proceedings 1999 International Conference on Image Processing* (Vol. 2, pp. 580–584). Kobe, Japan: IEEE.

Rönnfeldt, M. (2013). *Modulare Sensorplattform für den Einsatz auf Mini-UAVs : Konstruktion , Umsetzung und Erprobung*. University of the Bundeswehr Munich.

Rosten, E., & Drummond, T. (2005). Fusing points and lines for high performance tracking. In *10th IEEE International Conference on Computer Vision* (Vol. 2, pp. 1508–1515).

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *Computer Vision -- ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I* (Vol. 3951 LNCS, pp. 430–443). Berlin, Heidelberg: Springer Berlin Heidelberg.

Roupé, M., & Johansson, M. (2009). Visual quality of the ground in 3D models: using color-coded images to blend aerial photos with tiled detail-textures. In *Proceedings of the 6th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa* (pp. 73–79). New York, NY, USA: ACM.

Rudol, P., & Doherty, P. (2008). Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery. In *2008 IEEE Aerospace Conference* (pp. 1–8). Big Sky, MT, USA: IEEE.

Russ, M., Schmitt, M., Hellert, C., & Stuetz, P. (2013). Airborne sensor and perception management: Experiments and Results for surveillance UAS. In *AIAA Infotech@Aerospace (I@A) Conference,*

Guidance, Navigation, and Control and Co-located Conferences (pp. 1–16). Boston, MA, USA: American Institute of Aeronautics and Astronautics.

Russ, M., & Stütz, P. (2012). Airborne sensor and perception management: A conceptual approach for surveillance UAS. In *2012 15th International Conference on Information Fusion* (pp. 2444–2451). IEEE.

Safar, M., Shahabi, C., & Sun, X. (2000). Image Retrieval By Shape: A Comparative Study. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia* (Vol. 1, pp. 141–144). New York, NY, USA: IEEE.

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). 300 faces in-the-wild challenge: The first facial landmark Localization Challenge. In *2013 IEEE International Conference on Computer Vision Workshops* (pp. 397–403). Sydney, NSW, Australia: IEEE.

Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nesic, N., Wang, X., & Westling, P. (2014). High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. In X. Jiang, J. Hornegger, & R. Koch (Eds.), *Pattern Recognition: 36th German Conference, GCPR 2014* (pp. 31–42). Münster, Germany: Springer International Publishing.

Scharstein, D., & Szeliski, R. (2001). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*(1), 7–42.

Scharstein, D., & Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *1*, 195–202.

Schmitt, M., Rudnick, G., Stütz, P., & Schulte, A. (2015). A Tool Set for UAS Flight-Testing on Mission Level. In *NATO SCI-269 Symposium on Flight Testing of Unmanned Aerial Systems (UAS)* (pp. 1–12). Ottawa, Canada: STO/NATO.

Schröder, G., & Treiber, H. (2002). *Technische Optik* (9. ed.). Würzburg: Industrie Medien GmbH & Co. KG.

Seung-Seok, C., Sung-Hyuk, C., & Tappert, C. C. (2010). A Survey of Binary Similarity and Distance Measures. *Journal of Systemics, Cybernetics & Informatics*, *8*(1), 43–48.

Sheikh, H. R., Bovik, A. C., & Cormack, L. (2005). No-reference quality assessment using natural scene statistics: JPEG2000. *IEEE Transactions on Image Processing : A Publication of the IEEE Signal Processing Society*, *14*(11), 1918–1927.

Sheikh, H. R., Bovik, A. C., & de Veciana, G. (2005). An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Transactions on Image Processing : A Publication of the IEEE Signal Processing Society*, *14*(12), 2117–28.

Siam, M., & ElHelw, M. (2012). Robust autonomous visual detection and tracking of moving targets in UAV imagery. In *2012 IEEE 11th International Conference on Signal Processing* (Vol. 2, pp. 1060–1066). Beijing, China: IEEE.

Sikora, T. (2001). The MPEG-7 visual standard for content description-an overview. *IEEE Transactions on Circuits and Systems for Video Technology*, *11*(6), 696–702.

Sint, P. P. (1975). *Ähnlichkeitsstrukturen und Ähnlichkeitsmasse*. Wien, Österreich: Institut für Sozio-Ökonomische Entwicklungsforschung, Akademie der Wissenschaft.

Sirmacek, B., & Reinartz, P. (2011). Automatic crowd density and motion analysis in airborne image sequences based on a probabilistic framework. In *2011 IEEE International Conference on Computer Vision Workshops* (pp. 898–905). Barcelona, Spain: IEEE.

Spyrou, E., Tolias, G., Mylonas, P., & Avrithis, Y. (2009). Concept detection and keyframe extraction using a visual thesaurus. *Multimedia Tools and Applications*, *41*(3), 337–373.

STEMMER IMAGING GmbH. (2013). *Handbuch der Bildverarbeitung* (2013/2014 ed.). Puchheim, Germany: STEMMER IMAGING GmbH.

Stricker, M., Stricker, M., Orengo, M., & Orengo, M. (1995). Similarity of color images. *Storage and Retrieval for Image and Video Databases*, *SPIE 2420*(3), 381–392.

Student, & Gosset, W. S. (1908). The probable error of a mean. *Biometrika*, *6*(1), 1–25.

Sulc, Z. (2014). Similarity measures for nominal variable clustering. In *The 8th International Days of Statistics and Economics* (pp. 1536–1545).

Swain, M. J., & Ballard, D. H. (1991). Color Indexing. *International Journal of Computer Vision*, *7*(1), 11–32.

Szeliski, R. (2011). *Computer vision: algorithms and applications*. Springer-Verlag London.

Szijártó, G., & Koloszár, J. (2003). Hardware Accelerated Rendering of Foliage for Real-time Applications. In *Proceedings of the 19th Spring Conference on Computer Graphics* (pp. 141–148). ACM.

Tang, X., Luo, W., & Wang, X. (2013). Content-based photo quality assessment. *IEEE Transactions on Multimedia*, *15*(8), 1930–1943.

Taylor, G. R., Chosak, A. J., & Brewer, P. C. (2007). OVVV: Using virtual worlds to design and evaluate surveillance systems. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–8). Minneapolis, MN, USA: IEEE.

Thacker, N. a., Clark, A. F., Barron, J. L., Ross Beveridge, J., Courtney, P., Crum, W. R., … Clark, C. (2008). Performance characterization in computer vision: A guide to best practices. *Computer Vision and Image Understanding*, *109*(3), 305–334.

Thornton, K., Nadadur, D. C., Ramesh, V., Liu, X., Zhang, X., Bedekar, A., & Haralick, R. (1994). Groundtruthing the RADIUS model-board imagery. In *Proceedings of the ARPA Image Understanding Workshop* (pp. 319–329). Monterey, USA.

Tong, H., Li, M., Zhang, H., & Zhang, C. (2004). Blur detection for digital images using wavelet transform. In *2004 IEEE International Conference on Multimedia and Expo* (Vol. 1, pp. 17–20). Taipei, Taiwan: IEEE.

Toyama, K., Krumm, J., Brumitt, B., & Meyers, B. (1999). Wallflower: principles and practice of background maintenance. In *Proceedings of the 7th IEEE International Conference on Computer Vision* (Vol. 1, pp. 255–261). Kerkyra, Greece: IEEE.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*, 327–352.

Van Aken, J., & Novak, M. (1985). Curve-Drawing Algorithms for Raster Displays. *ACM Transactions on Graphics*, *4*(2), 147–169.

Vasilakis, A. a., & Fudos, I. (2013). Depth-fighting aware methods for multifragment rendering. *IEEE Transactions on Visualization and Computer Graphics*, *19*(6), 967–977.

Vaudrey, T., Rabe, C., Klette, R., & Milburn, J. (2008). Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In *23rd International Conference Image and Vision Computing New Zealand*. Christchurch, New Zealand: IEEE.

Vedaldi, A., Ling, H., & Soatto, S. (2010). Knowing a good feature when you see it: Ground truth and methodology to evaluate local features for recognition. In R. Cipolla, S. Battiato, & G. M. Farinella (Eds.), *Computer Vision: Detection, Recognition and Reconstruction* (pp. 27–49). Berlin, Heidelberg: Springer Berlin Heidelberg.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vol. 1, pp. I–511–I–518). Kauai, HI, USA: IEEE Computer Society.

Vondrick, C., Ramanan, D., & Patterson, D. (2013). Efficiently Scaling up Crowdsourced Video Annotation. *International Journal of Computer Vision*, *101*(1), 184–204.

Wang, J. Z., Li, J. L. J., & Wiederhold, G. (2001). SIMPLIcity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(9), 947–963.

Wang, L., Zhang, Y., & Feng, J. (2005). On the Euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1334–1339.

Wang, Z., Bovik, A. C., & Evan, B. L. (2000). Blind measurement of blocking artifacts in images. In *Proceedings 2000 International Conference on Image Processing* (Vol. 3, pp. 981–984). Vancouver, BC, Canada: IEEE.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612.

Wang, Z., & Li, Q. (2011). Information content weighting for perceptual image quality assessment. *IEEE Transactions on Image Processing*, *20*(5), 1185–1198.

Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multi-scale structural similarity for image quality assessment. In *The 37th Asilomar Conference on Signals, Systems and Computers* (Vol. 2, pp. 9–13). Pacific Grove, CA, USA: IEEE.

Watt, A. (2000). *3D Computer Graphics* (3rd ed.). Pearson Education Limited.

Wilkinson, G. N., & Rogers, C. E. (1973). Symbolic description of factorial models for analysis of variance. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *22*(3), 392–399.

Witkin, A. P. (1983). Scale-space Filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence* (Vol. 2, pp. 1019–1022). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Wood, E., Baltrusaitis, T., Zhang, X., Sugano, Y., Robinson, P., & Bulling, A. (2015). Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. *Computer Vision and Pattern Recognition*, *abs/1505.0*, 1–9.

Xsens Technologies B.V. (2014). *MTi User Manual*. Enschede, Netherlands.

Yamada, A., Pickering, M., Jeannin, S., Cieplinski, L., Ohm, J.-R., & Kim, M. (2001). *MPEG-7 Visual part of eXperimentation Model Version 8.0*.

Yang, N.-C., Chang, W.-H., Kuo, C.-M., & Li, T.-H. (2008). A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval. *Journal of Visual Communication and Image Representation*, *19*(2), 92–105.

Zach, C., Pock, T., & Bischof, H. (2007). A duality based approach for realtime TV-L 1 optical flow. In F. A. Hamprecht, C. Schnörr, & B. and Jähne (Eds.), *Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany* (Vol. LNCS 4713, pp. 214–223). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zhang, L., da Fonseca, M. J., & Ferreira, A. (2007). *Survey on 3D shape descriptors*.

Zhang, L., & Li, H. (2012). SR-SIM: A fast and high performance IQA index based on spectral residual. In *2012 19th IEEE International Conference on Image Processing* (pp. 1473–1476). Orlando, FL, USA: IEEE.

Zhang, L., Zhang, L., Mou, X., & Zhang, D. (2012). A comprehensive evaluation of full reference image quality assessment algorithms. In *2012 19th IEEE International Conference on Image Processing* (pp. 1477–1480). Orlando, FL, USA: IEEE.

Zhang, L., Zhang, L., Mou, Y., & Zhang, D. (2011). FSIM: A Feature Similarity Index for Image. *IEEE Transactions on Image Processing*, *20*(8), 2378–2386.

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the 7th IEEE International Conference on Computer Vision* (Vol. 1, pp. 666–673). Kerkyra, Greece: IEEE.

Appendix

A First questionnaire on computer graphic engines

In order to identify the rendering techniques used in VBS3 a questionnaire was sent to Bohemia Simulations. The lead programmer has filled the following questionnaire:

Integrated Test bed for Experimentation on Mission sensors

1st QUESTIONNAIRE

ON

COMPUTER GRAPHIC ENGINES

VBS2 1.6, VBS2 2.12

AND

VBS3

Research Abstract

In the recent development of assisting technologies in automobile, aeronautical and aerospace applications the ability of machines to perceive the environment and/or objects through computer vision algorithms has become more and more common. The development of such algorithms is either based on machine learning through extensive datasets or on perception models developed by researchers using live video footage. For use in simple applications these approaches are feasible. However for complex and/or highly dynamic applications as obstacle/pedestrian detection, vehicle classification or reasoning algorithms this approach is insufficient due to missing or expensively to acquire test data. Therefore such applications are usually tested on lacking dataset (which leads to algorithms performing only during well-defined boundary conditions of small range) or are tested on purely synthetical datasets, which do not allow extrapolation of acquired results to real world situations. Therefore the need for extensive but reliable video footage for complex applications in high variation is still an issue in the computer vision community. To fully utilize the capabilities and opportunities of synthetical environments, the missing link between camera captures and computer generated video streams needs to be researched. Therefore ITEM, the integrated test bed for experimentation on mission sensors, has been designed. This test bed allows the generation of synthetic video streams of areal mounted sensors (including environmental and camera introduced effects as noise, distortion, etc.) and the comparison with prior recorded real flight tests. Obviously, the generation of synthetic video streams is based computer generated graphics. In the case of ITEM the video output is created using serious game engines because these allow the creation of vivid and dynamic scenarios with higher optical quality as current commercial simulation engines. However all engines use simplifications to allow themselves to be real time simulations. These simplifications may create artefacts or behaviours which are not suitable for test and development of specific computer vision algorithms intended for real world applications.

Our main research goal is to identify behaviours of computer graphic engines that influence the results of computer vision algorithms by using specific example algorithms. Then we try to minimize the influence by configuring the engines allowing us to identifying suited render technology candidates. These results will then be merged into development recommendations for computer graphic engines to enable full synthetical computer vision algorithm testing with correlation to real world behaviours.

Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2012) under the agreement No 285320

About this questionnaire

Currently ITEM can use the game engines of VBS2 1.6 and VBS2 2.12, which use different computer graphic engines. Therefore it is essential to identify which technologies are used to enable the research to map experimentation results to their tested technologies.

This questionnaire is addressed to the employees of Bohemia Simulations involved with the computer graphic engines of VBS2 1.6 and/or VBS2 2.12. Please fill in the game engine you answer to in the field on this page. In case you cannot answer a specific question, leave it blank.

The questionnaire is divided into four main Sections. Section A provides a number of questions that are used to identify the illumination model used in the game engine. Section B is related to texturing technology utilized. Section C lists a number of questions regarding the camera model and the camera frustum used and Section D provides questions about modelling and object placement.

At the end of the questionnaire we ask you to provide some information concerning the person who completed the questionnaire, which is optional. You can provide answers to only the questions that you feel more comfortable.

Computer Graphics Engine

All following answers to this questionnaire are about the following Computer Graphics Engine (please also list the Program\Game):

_____

Or

If you want to fill the questionnaires for multiple Computer Graphic Engines simultaneously then mark the questions with the following symbols:

X **for Game/Engine: Real Virtuality 2, VBS2 1.X, ArmA**

O **for Game/Engine: Real Virtuality 3, VBS2 2.X, ArmA2**

✓ **for Game/Engine: _____**

In case answers in text form are required, start the question with the symbol the game engine has been associated with.

A. Illumination Model

A. 1 What kind of Illumination model type do you use (mark with X)?

Local Illumination Model	X, O
Global Illumination Model	X, O

A. 2 In case of a global illumination model, which technology do you use (mark with X)?

Radiosity (Radiance)	X, O - only precomputed
Ray-tracing	
Beam-tracing	
Cone-tracing	
Path-tracing	
Metropolis light transport	
Ambient occlusion	O
Photon mapping	
Other: Hemispherical ambient lightin	O

A. 3 In case of a local illumination model, which technology do you use (mark with X)?

Blinn-Phong	
Cook-Torrance	
Minnaert	
Oren-Nayar	
Phong	X, O
Ward anisotropic	
Other: _____	

A. 4 In case local illumination model is used, which shading model has been implemented (mark with X)?

Gouraud Shading	
Phong Shading	X, O
Other: _____	

A. 5 And can the shading model be changed per video settings (mark with X)?

Yes	
No	X, O

If the answer is yes, please give a detailed explanation:

_____

_____

_____

_____

A. 6 In case local illumination models are used, which shadow model (SM) has been implemented (mark with X)?

Simple Shadow Mapping (SSM)	X
Parallel Split Shadow Mapping (PSSM)	
Cascaded Shadow Mapping (CSM)	O
Light Space Perspective Shadow Mappin (LiSPSM)	
Trapeoid Shadow Mapping (TSM)	
Perspective Shadow Mapping (PSM)	
Percentage Closer Filtering (PCF)	O
Exponential Shadow Mapping (ESM)	
Convolution Shadow Mapping (CSM)	
Variance Shadow Mapping (VSM)	O
Summed Area Variance Shadow Mappin (SAVSM)	
Percentage Closer Soft Shadows (PCSS)	
Screen space soft shadows (SSSS)	
Adaptive Shadow Mapping (ASM)	
Adaptive Volumetric Shadow Mappin (AVSM)	
Camera Space Shadow Mapping (CSSM)	
Deep Adaptive Shadow Mapping (DASM)	
Dual Paraboloid Shadow Mapping (DPSM)	
Deep Shadow Mapping (DSM)	
Forward Shadow Mapping (FSM)	
Logarithmic Shadow Mapping (LogSM)	
Multiple Depth Shadow Mapping (MDSM)	
Sample Distribution Shadow Mapping (SDSM	
Separating Plane Perspective Shadow Mappin	

(SPPSM)	
Shadow silhouette Shadow Mapping (SSSM)	
Other: Shadow volume	X, O

A. 7 And can the shadow mapping be changed per video settings (mark with X)?

Yes	X, O
No	

If the answer is yes, please give a detailed explanation:

In video options we can choose distance to which shadows are being drawn and whether percentage closer filtering is used or not. Both have significant and predictable impact on performance. We have also video option to specify whether shadow volume or shadow buffer technique should be used for shadows. In certain situations shadow volumes provide better performance. However there are also situations where shadow volumes are more expensive than shadow buffers (which is more frequently valid on newer HW that is becomming more optimized for shadow buffer approach). Because of this confusion and because of smaller demands on art side with shadow buffers (we don't have to create shadow volumes for models for shadow buffer technique), we are moving away from shadow volumes, towards shadow buffer technique.

B. Texturing

B. 1 What kind of texture mapping technologies can be applied (mark with X)?

Cube Mapping	
Mip mapping	X, O
Displacement Mapping	
Reflection Mapping	X, O
Normal Mapping (Dot3 bumb mapping)	X, O
Parallax mapping	O
Relief mapping	
UV mapping	X, O
UVW mapping	
Other: Shadow maps	X, O
Other: Specular maps	X, O
Other: _____	

B. 2 What is the maximum texture size in pixels recommended?

X - 2048, O - 8192 are the limits. Most practical for regular models (to save a disk space) seems to be 2048. The bigger sizes don't automatically trigger bigger VRAM consumption in the engine, it streams into memory only those levels that are really required.

B. 3 What is the maximum number of textures per model recommended (please add the texture size as reference)?

The smaller amount, the better. Problem is that every texture switch during rendering is expensive for GPU, as it requires a new draw command. Some

models have one texture, we consider the model to be within limits when it has around 5 textures.

C. Camera Model / Frustum

C. 1 What kind of camera model is deployed in your engine (mark with X)?

Pinhole camera model	
Synthetic camera model	
Other:_____	

If other has been selected, please provide the name or a link to the publication of the mentioned camera model.

C. 2 Which Parameters define the Camera Frustum (mark with X)?

Field of View	X, O
Aspect Ratio	X, O
Terrain Distance	X, O
Object Draw Distance	
Other: Camera position and orientation, position of nearest visib object (to shift front plane as far a possible, to fight with z-fighting)	X, O

C. 3 The Setting Objects Detail lowers the distance where the LODs are changed to higher details when set lower settings. This Effect ("Popping") is very visible for machine vision applications due to immediate changes in the image structure. Is there a possibility to reduce / eliminate this effect (mark with X)?

Yes	O
No	
Other:_____	

In either case, please give a short explanation:

The transition could be greatly reduced for trees and vegetation, mainly by updating our content to new type of shaders - using the alpha-to-coverage technique the trees then blend smoothly between LODs.

The situation is more difficult with objects. Native way would be to do the alpha blending, however simple switch to use alpha blending would not work as we would get alpha sorting artefacts. To solve that, we would need to implement technique like depth peeling (order independent transparency) and therefore move from DX9 to DX11.

D. Modelling / Object Placement

For ITEM we are in need to generate high quality geo-referenced terrain databases for comparison to aerial photographs. These photographs mostly show urban terrain with several thousands of objects (see Image 1), please take this into considerations when answering these questions.

Image 1: Modelled terrain database of UniBw Munich on the left and aerial photography of same terrain on the right.

D. 1 What is the positioning accuracy of objects in VBS2 (mark with X)?

1 m	
10 cm	
1 cm	
1 mm	
Dependant to the size of the map. Add rule of thumb: X uses flo precision, O uses double precision. I case of X the precision whe coordinates are around 100km fron origin is 7.8mm and error grows wi size of the map. In case of O th precision when coordinates ar around 100km from origin i 0.000000000015m.	X, O
Other: _____	

D. 2 What is the angular accuracy of objects in VBS2 (mark with X)?

1 °	
1.X °	
1.XX °	
1.XXX °	
Other: To represent orientation w use 3x3 matrices with float number Since the numbers there are in interv <-1,1>, we never met any problem with precision there. (the precision considerably higher than 1.XXX °)	

D. 3 What is the maximum number of vertices per model recommended on the highest level of detail?

We are aiming at models to have under 10000 vertices. More important models can have more when needed and what it costs us it mainly disk space.

The limit of X is roughly 30000 triangles. The limit of O is 1000000 triangles

D. 4 What is the maximum number of vertices per model recommended on the lowest level of detail?

Theoretically the smaller number the better, if nothing else than to save space in vertex buffers. We have models that have units of triangles (like one polyplane to represent a tree in a distance - that means 12 triangles). In usual cases the number in last LOD could be higher, like dozens or units of hundreds of vertices and we would still not see impact on performance - we are usually not vertex limited in our scenes, rather CPU or fillrate limited instead.

D. 5 Is it possible to create a Terrain on the edge of two UTM zones and still get correct coordinates using the PosToCoord Scripting command (mark with X)?

Yes	X, O
No	
Other:_____	

In either case, please give a short explanation:

For conversion from projected surface (flat surface) to UTM coordinates including change in UTM zones we rely on Geotrans library (http://earth-info.nga.mil/GandG/geotrans/). The support for crossing UTM zones was significantly improved in O (f.i. in X the grids in 2D map don't respect UTM zones).

E. RECOMMENDATIONS

Please provide any issue related to this questionnaire based on your experience.

F. PROFESSIONAL INFORMATION

Please describe briefly your professional role and experience in Computer Graphics (Programmer, Studies, Scientist, etc.):

Professional role Programmer	Experience in Computer Graphics development/usage: 13/18 years	Employed as a: Lead programmer
How your work concerns Computer Graphics? We develope games / visualization software.	Describe briefly: ArmA, ArmA2, VBS2, VBS3.	

B Telemetry-based homography estimation

Here, the computational method to acquire the homographic relation between two aerial images is detailed. World coordinates $p_{w0} = (X, Y, Z, 1)$ can be derived from screen coordinates $p_0 = (x_p, y_P, 1, 1)$ using the *intrinsic camera matrix K* and the *3D Euclidean rigid-body transformation E* (Szeliski, 2011):

$$p_{w0} = E_0^{-1} K_0^{-1} p_0 \tag{40}$$

The *intrinsic camera matrix K* describes the conversion from metric sensor coordinates to the raster image coordinates and considers the focal length in horizontal and vertical directions f_x and f_y as well as the optical centre $C = (x_c, y_c)$. The Euclidian transformation matrix E consists of the Euclidian rotation matrix R and translation vector t, describing the location of the aircraft in UTM coordinates and altitude above ground level. Choosing the *Universal Transverse Mercator* (UTM) coordinate system allows description of world positions in a Cartesian metric coordinate system.

$$K = \begin{bmatrix} f_x & 0 & x_c & 0 \\ 0 & f_y & x_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad E = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \qquad t = \begin{pmatrix} UTM_{east} \\ UTM_{north} \\ Altitude_{AGL} \end{pmatrix} \tag{41}$$

The rotation matrix $R \triangleq R_{EC}$ describes the rotation from the camera coordinate system to the earth coordinate system. It consists of four separate rotation matrices:

$$R \triangleq R_{EC} = R_{CS_{EA}} R_{EA} R_{CS_{AC}} R_{AC} \tag{42}$$

The rotation matrix R_{AC} describes the rotation of camera coordinate system to the aircraft coordinate system defined as depicted in Figure B-1 models a camera that can be rotated around x-axis (elevation, in short $elev$) and y-axis (Azimuth in short azi). Both angles are defined relative to the aircraft body. The notation has been shortened for the following equation, *s* stands for *sine* and *c* for *cosine*:

$$R_{AC} = \begin{bmatrix} c(azi) & s(azi) & 0 \\ -s(azi) & c(azi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(elev) & -s(elev) \\ 0 & s(elev) & c(elev) \end{bmatrix} \tag{43}$$

Rotation matrix R_{EA} describes the rotation of aircraft coordinate system to the earth coordinate system as depicted in Figure B-1. The coordinate system has been modelled as specified by DIN 9300 (Institut für Normung, 1990). The following equation describes the three rotations in Euler angles (roll Φ around x-axis; pitch Θ around y-axis; yaw Ψ around z-axis):

$$R_{EA} = \begin{bmatrix} c(\Psi) & -s(\Psi) & 0 \\ s(\Psi) & c(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(\Theta) & 0 & s(\Theta) \\ 0 & 1 & 0 \\ -s(\Theta) & 0 & c(\Theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\Phi) & -s(\Phi) \\ 0 & s(\Phi) & c(\Phi) \end{bmatrix} \tag{44}$$

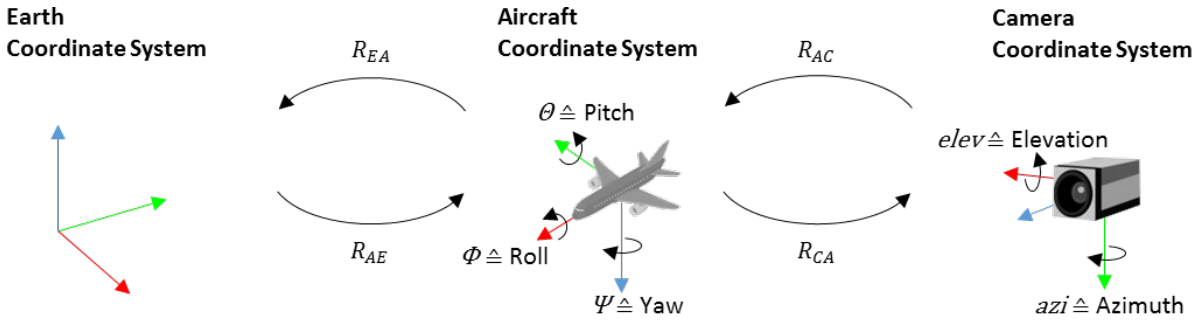


Figure B-1: Rotation matrices and their variables to rotate points defined in one coordinate system into the other.

Since the initial definitions (when all variables are zero) of used rotation matrices differs as depicted in Figure B-2, two auxiliary rotation matrices $R_{CS_{EA}}$ and $R_{CS_{AC}}$ (CS stands for coordinate system) are introduced to apply the necessary conversion to R:

$$R_{CS_{EA}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad R_{CS_{AC}} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{45}$$

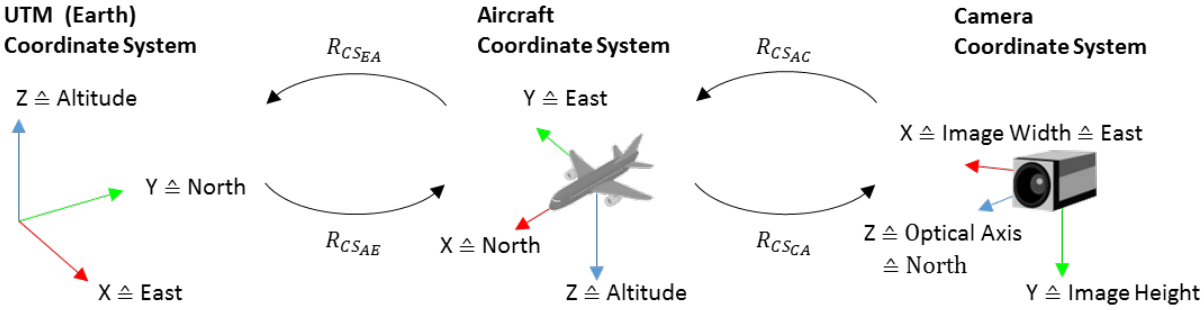


Figure B-2: Initial starting positions of all coordinate systems and their correlation to geographical north and east.

THE now uses telemetry information only to acquire the correlation between the two images. The homographic correlation of two images can be computed using following equation (Szeliski, 2011):

$$p_1 = K_1 E_1 E_0^{-1} K_0^{-1} p_0 = M_{10} p_0 \tag{46}$$

The matrices are given in homogenous coordinates as 4x4 matrices, where the last row saves the depth information. Without the last column and row M_{10} equals the homography matrix H_{10}.Thus, homography can be computed using the previously defined mathematic transformations K and E, since the necessary information about pose and position of the aircraft and camera are known.

C Regression models

C.1 Regression models with categorical predictor

For the configuration set experiments the categorical predictor *Config* was introduced. It was not discussed in the baseline experiments due to its irrelevance in that specific experiment. The regression model had been fitted using six image content distance measures and *Config* as predictors and the performance metric as outcome variable. The predictor *Config* contains the **rendering engine parameters** labels (see Table 6-1 for the baseline setting and IDs) that cannot be ranked, e.g. gender or religion. See (Field, 2009) for theoretical background on categorical variables in multiple regression.

In our application, *Config* represents all rendering parameter sets (21 in total with *baseline* being the *default*). Thus, in case of *baseline* predictor *Config* can be removed from the equation (as done in chapter 6.2). This approach reveals the **influence of parameters** and whether it is of significant magnitude. Additionally, it increased the number of samples by the number of parameters (sample size 35 times 21 groups = 735), since the data of all parameters is used to fit the model.

As regression can only handle *interval* or *nominal* variables, the *categorical* variable is split into several 'dummy' *nominal* variables (0/1 being their only states). Thus, for instance *Config_20* represents the differences in the model when *shadow filtering* is enabled. If all dummy variables are zero the overall model defaults to the *baseline* model equation.

For illustration, a reduced model for *forest, ⊿relative repeatability, SIFT* is given in Table 0-1 with parameter 20 (*shadow filtering)* being set. In the coefficients column first the image content distances used as predictors and their intercept are presented (cf. chapter 6.1.3). The specific parameter is represented by the name of the predictor (*Config)* and the index number of the parameter, which is consequently the name of the dummy variable. Thus, *Config_20* presents the change of the intercept when *shadow filtering* is enabled. The following so called **interaction terms** (e.g. *CLD:Config_20)* represent the introduced change of effect on colour layout when *shadow filtering (Config_20)* is enabled and is expressed as $b * CLD * Config\_20$ in the linear equation. The asterisks in the table present the significance. Thus a regression model can consist at max of 147 terms of sum (6 image descriptors + intercepts times 21 configurations = 147).

Table 0-1: Example regression model with *Config* variable to investigate the differences between parameters (using the Wilkinson Notation (Wilkinson & Rogers, 1973)**).**

Model	Model Fit		Coefficients	*b*-Value	SE	β
Δrel. Repeatability	R^2	68%	Intercept	0.658 Ns	0.602	**0.00**
SIFT	Adj-R^2	64%	CSD	-1.668*	0.843	**-0.52**
forest	F-Ratio	16***	EHD	-0.648 Ns	0.493	**-0.23**
			CLD	-1.611***	0.413	**-0.19**
			SCD	-0.237 Ns	1.519	**-0.05**
			Config_20	-0.844 Ns	0.776	**0.00**
			EHD:Config_20	0.293 Ns	1.351	**0.11**
			CSD:Config_20	2.090 Ns	0.288	**0.65**
			SCD:Config_20	1.233 Ns	1.912	**0.29**

Ns = not significant (p > .05). *p < .05. ** p < .01. *** p < .001

The model equation for *baseline* (already shown for this example in chapter 6.1.3) can be derived by ignoring all terms containing the *Config* predictor (for the sake of the example non-significant coefficients are included):

$$\Delta rel.\ Rep. = 0.66 - 1.67CSD - 0.65EHD - 1.61CLD - 0.24SCD \tag{47}$$

As mentioned before, whenever the feature detector performs better on synthetic data *Δrelative repeatability* becomes positive. Zero indicates an equal detection rate, which would be the optimal result. The equation of *shadow filtering* can be extracted by considering all coefficients containing "Config_20" and adding them to the baseline measures. The value of *Config_20* is added to the intercept:

$$\begin{aligned}\Delta rel.\ Rep. = {} & (0.66 - 0.84) + (-1.67 + 2.09)CSD + (-0.65 + 0.29)EHD \\ & - 1.61CLD + (-0.24 + 1.23)SCD\end{aligned} \tag{48}$$

Neither *Config_20* nor any interaction term shows significant difference (due to the high standard error). The equation now explains the performance difference between synthetic images with *shadow filtering* and natural images. The complete model covers 68% variance of the outcome variable as presented by R^2. This example shows that *shadow filtering* lessens the impact of colour structure (CSD) and edge appearance (EHD) on the outcome *Δrelative repeatability*, while the influence of the colour distribution is enlarged (SCD). Interestingly, the coefficients of CSD and SCD changed the sign; this means a lowered distance compared to *baseline* would now raise the performance on synthetic data. Since the actual change in distance is known from the previous step (chapter 6.2.1.2), the terms $b * distance$ are presented to simplify the discussion (chapter 6.2.1.3).

C.2 General model fitting quality R²

In Figure C-3, an overview over the degree of fitting for each model is given accompanied by its significance value *p*. All models are significant as depicted by the asterisks. R^2 describes the amount of outcome variance explained by the model. Low values indicate that the selected predictors do not cover all factors affecting the feature detector performance. The colour scheme was selected purposely, since low fits do not necessary express bad results. Generally, *Δabsolute repeatability* models for fit better than their *relative* comparisons. The fitting proves to be scene dependent, e.g. *forest* or *sport*.

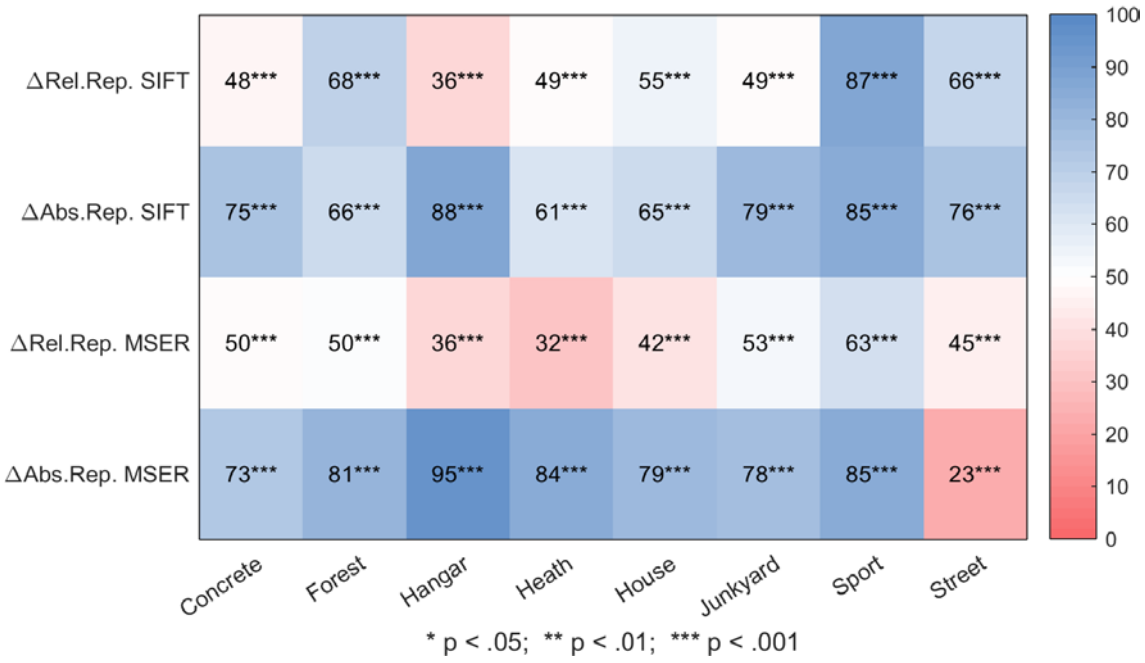


Figure C-3: R^2 for all models together with their significance encoded by asterisk.

C.3 Regression model coefficients

SIFT Δrelative Repeatability

Δrelative repeatability, SIFT, concrete														
Linear regression model:	RelRepeatability ~ 1 + CLD*Config + SCD*Config + DCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.241	0.308			-0.009	1.129	-0.332	1.272	0.009	0.223	-0.240	0.649	-0.219	0.612
noise	-0.181	0.442			0.760	1.579	-0.127	1.810	0.038	0.311	0.334	0.954	-0.337	0.828
distortion	-0.085	0.435			0.153	1.652	0.015	1.977	-0.089	0.277	0.178	1.006	0.017	0.787
aperture	-0.099	0.364			2.693	2.566	-0.031	1.337	-0.106	0.262	-0.084	0.781	-0.867	0.868
HDR	-0.092	0.350			1.188	2.056	0.291	1.500	-0.112	0.277	0.200	0.823	-1.269	0.924
bloom	-0.328	0.371			1.834	1.699	0.112	1.600	0.328	0.339	-0.039	0.912	0.879	0.919
blur	-0.267	0.431			2.115	1.862	0.898	1.811	0.015	0.298	-0.638	0.910	0.291	0.736
SSAA	0.011	0.435			-0.653	1.547	0.226	1.836	-0.006	0.317	-0.005	0.927	-0.027	0.882
MSAA	-0.026	0.435			-0.034	1.575	0.001	1.900	-0.021	0.318	0.135	0.957	0.073	0.841
FXAA	-2.490	0.438			3.453	1.624	6.789	1.817	0.613	0.283	1.533	0.886	-0.973	0.953
SMAA	-0.052	0.446			0.126	1.557	-0.216	1.868	0.015	0.316	0.316	0.941	0.118	0.863
AToC	-0.138	0.424			0.783	1.564	0.088	1.767	0.054	0.318	0.084	0.943	0.252	0.873
objects high	-0.026	0.435			-0.241	1.584	-0.018	1.829	0.031	0.308	0.146	0.950	0.077	0.835
no objects	0.081	0.377			3.573	1.952	1.138	1.335	-0.675	0.299	-1.501	0.756	0.128	0.836
modelling errors	-0.205	0.433			1.011	1.616	0.251	1.847	0.087	0.311	0.071	0.984	0.151	0.846
texture low	-0.968	0.460			6.816	1.794	1.001	1.769	-0.159	0.276	-1.230	0.877	3.044	0.830
surface low	-0.925	0.396			8.004	1.675	-1.329	1.624	0.787	0.309	-0.015	0.798	3.027	1.366
AF	-0.041	0.432			3.072	1.569	-4.983	1.825	1.144	0.322	2.347	0.946	2.233	0.828
shadow	-0.177	0.422			2.014	1.626	-0.114	1.826	-0.041	0.288	0.247	0.898	-0.402	0.760
shadow filter	0.029	0.440			-0.833	1.577	-0.079	1.808	-0.047	0.280	0.207	0.870	0.062	0.757
SSAO	-0.046	0.426			-0.051	1.585	0.043	1.772	0.017	0.313	0.138	0.922	0.018	0.849
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-4: *Δrelative repeatability* regression model for SIFT on scene *concrete*.

Δrelative repeatability, SIFT, forest														
Linear regression model:	RelRepeatability ~ 1 + CLD + CSD*Config + SCD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.658	0.602	-1.668	0.843	-1.611	0.413	-0.237	1.519					-0.648	0.493
noise	-1.007	0.861	1.469	1.150			2.079	2.212					-0.102	0.683
distortion	0.455	0.943	-1.103	1.183			-0.787	2.462					0.190	0.688
aperture	-0.092	0.781	-1.233	1.333			0.635	2.133					-0.487	0.924
HDR	-0.279	0.751	-0.087	0.995			0.996	2.111					-0.066	0.679
bloom	1.148	0.907	-0.128	1.167			-3.931	2.326					0.151	0.682
blur	-3.175	0.812	0.055	1.233			7.622	2.340					3.407	1.028
SSAA	-0.215	0.802	-0.598	1.095			1.069	2.174					-0.143	0.737
MSAA	-0.467	0.845	0.094	1.127			1.369	2.189					0.420	0.692
FXAA	-0.791	0.857	0.148	1.166			2.232	2.179					0.757	0.696
SMAA	-0.383	0.818	0.333	1.120			0.971	2.098					-0.009	0.694
AToC	-0.130	0.863	-0.011	1.149			0.473	2.201					-0.096	0.679
objects high	-0.131	0.873	0.260	1.140			0.264	2.241					-0.198	0.674
no objects	-0.628	0.633	8.743	1.408			-1.291	1.617					-1.778	0.642
modelling errors	0.334	0.828	-0.910	1.135			-0.494	2.104					0.517	0.697
texture low	-0.135	0.931	0.268	1.280			0.193	2.295					0.507	0.717
surface low	-1.428	0.895	0.848	1.140			3.810	2.375					0.622	0.701
AF	-0.245	0.874	-0.421	1.159			1.034	2.230					0.361	0.684
shadow	-0.449	0.790	1.425	1.360			0.281	1.967					0.059	0.706
shadow filter	-0.844	0.776	2.090	1.351			1.233	1.912					0.293	0.697
SSAO	-0.609	0.842	0.614	1.199			1.480	2.113					0.365	0.688
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-5: *Δrelative repeatability* regression model for SIFT on scene *forest*.

Δrelative repeatability, SIFT, hangar														
Linear regression model:	RelRepeatability ~ 1 + CLD + SCD + DCD + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-0.079	0.063			-0.786	0.291	0.725	0.069	-0.140	0.030			0.300	0.419
noise	-0.028	0.072											-0.313	0.621
distortion	-0.008	0.064											-0.064	0.533
aperture	0.005	0.059											-0.019	0.477
HDR	0.023	0.068											-0.152	0.583
bloom	-0.175	0.069											1.403	0.593
blur	0.110	0.064											-0.825	0.459
SSAA	0.021	0.065											0.012	0.559
MSAA	0.037	0.069											-0.255	0.585
FXAA	0.004	0.059											-0.137	0.508
SMAA	0.173	0.066											-1.792	0.566
AToC	0.026	0.069											-0.166	0.591
objects high	0.021	0.068											-0.133	0.577
no objects	0.017	0.055											0.081	0.439
modelling errors	0.055	0.069											-0.256	0.592
texture low	0.042	0.054											-0.219	0.444
surface low	-0.040	0.060											-0.080	0.482
AF	0.030	0.073											-0.258	0.621
shadow	0.038	0.067											-0.288	0.555
shadow filter	0.047	0.069											-0.348	0.577
SSAO	0.013	0.067											-0.069	0.571
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-6: *Δrelative repeatability* regression model for SIFT on scene *hangar*.

Δrelative repeatability, SIFT, heath														
Linear regression model:	RelRepeatability ~ 1 + EHD + CSD*Config + CLD*Config + SCD*Config + DCD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.212	0.152	0.432	1.126	-0.200	0.522	-0.410	0.738	-0.053	0.128			-0.101	0.048
noise	-0.107	0.197	-0.002	1.383	-0.045	0.654	0.054	0.919	0.051	0.144				
distortion	-0.044	0.208	-0.234	1.468	0.057	0.705	0.140	0.981	0.021	0.166				
aperture	-0.104	0.177	-0.861	1.456	0.561	1.269	0.447	0.840	0.102	0.187				
HDR	-0.011	0.198	-0.551	1.407	0.092	0.881	0.207	0.863	0.032	0.209				
bloom	-0.006	0.190	-0.616	1.385	0.083	0.639	0.286	0.902	0.054	0.142				
blur	-0.040	0.214	0.312	1.919	-0.122	0.932	0.056	1.160	0.074	0.161				
SSAA	-0.022	0.203	-0.813	1.357	0.113	0.651	0.441	0.919	0.037	0.201				
MSAA	-0.025	0.193	-0.933	1.386	0.108	0.648	0.489	0.904	0.081	0.144				
FXAA	0.086	0.210	0.108	1.518	-0.182	0.711	-0.314	1.014	-0.009	0.164				
SMAA	0.003	0.220	-0.380	1.437	-0.030	0.705	0.159	1.005	0.025	0.176				
AToC	-0.056	0.194	-0.614	1.387	0.163	0.648	0.413	0.911	0.061	0.143				
objects high	-0.031	0.191	-0.993	1.377	0.163	0.639	0.527	0.893	0.074	0.143				
no objects	0.188	0.172	-0.854	1.493	1.497	1.153	-0.923	0.831	-0.025	0.164				
modelling errors	0.027	0.219	-0.092	1.595	0.008	0.710	-0.068	1.087	0.020	0.180				
texture low	-0.385	0.190	0.332	1.384	0.275	0.708	0.716	0.841	0.116	0.142				
surface low	0.114	0.198	1.270	1.819	-2.449	1.599	-0.844	0.972	-0.140	0.145				
AF	-0.019	0.195	-0.726	1.407	0.104	0.645	0.354	0.930	0.062	0.144				
shadow	0.028	0.181	-5.707	1.536	2.753	0.690	1.729	0.869	0.521	0.186				
shadow filter	-0.033	0.178	-0.686	1.445	0.157	0.627	0.374	0.827	0.065	0.144				
SSAO	0.382	0.211	4.582	1.449	-2.959	0.708	-2.524	0.977	-0.641	0.178				
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-7: *Δrelative repeatability* regression model for SIFT on scene *heath*.

Δrelative repeatability, SIFT, house														
Linear regression model:	RelRepeatability ~ 1 + CSD*Config + SCD*Config + DCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.066	0.409	-0.876	0.670			0.390	0.475	-0.095	0.139	-0.002	0.562	0.520	0.653
noise	-0.068	0.598	0.239	0.901			-0.095	0.734	-0.013	0.195	-0.099	0.784	0.131	1.135
distortion	-0.106	0.534	0.494	0.879			-0.197	0.656	0.153	0.216	0.015	0.780	-0.034	0.884
aperture	0.124	0.434	0.688	0.754			-0.775	0.794	0.054	0.267	-0.345	0.682	-0.159	0.783
HDR	0.193	0.503	-0.022	0.828			-0.490	0.944	-0.182	0.286	-0.232	0.796	0.022	0.970
bloom	0.125	0.469	0.280	0.775			-0.516	0.881	-0.112	0.225	-0.150	0.716	-0.086	0.831
blur	-0.012	0.505	0.729	0.936			-0.062	0.626	0.062	0.183	-0.062	0.717	-0.634	0.709
SSAA	-0.238	0.554	0.331	0.878			0.250	0.689	-0.026	0.195	0.267	0.787	0.202	0.890
MSAA	1.216	0.577	-1.844	0.907			0.016	0.689	0.885	0.193	-3.382	0.794	-1.352	0.958
FXAA	0.915	0.557	-1.831	0.946			-0.031	0.683	0.183	0.198	-1.767	0.755	-1.182	0.814
SMAA	-0.086	0.539	0.143	0.881			-0.045	0.651	-0.006	0.192	0.169	0.794	0.232	0.907
AToC	-0.029	0.601	-0.005	0.934			0.130	0.713	0.015	0.199	-0.049	0.811	-0.009	1.012
objects high	-0.033	0.575	-0.012	0.904			0.080	0.690	-0.007	0.201	0.005	0.810	0.055	0.943
no objects	-0.436	0.468	1.593	1.071			0.822	1.303	-0.048	0.200	0.666	0.840	-0.854	0.757
modelling errors	-0.301	0.529	0.769	0.827			-0.009	0.671	-0.064	0.227	0.362	0.761	0.706	1.150
texture low	0.978	0.559	-0.287	0.834			-0.871	0.623	-0.566	0.198	-1.366	0.914	-1.746	0.733
surface low	-2.014	0.534	3.476	0.820			2.738	0.644	-0.665	0.278	1.067	0.824	1.945	0.917
AF	0.013	0.569	0.054	0.891			-0.030	0.680	0.019	0.194	-0.131	0.767	0.134	1.198
shadow	0.039	0.567	0.170	0.986			-0.074	0.693	0.068	0.231	-0.134	0.719	-0.330	0.847
shadow filter	-0.041	0.548	0.309	0.932			-0.002	0.669	0.074	0.218	-0.068	0.720	-0.225	0.883
SSAO	0.174	0.554	-0.103	0.928			-0.186	0.665	0.016	0.198	-0.293	0.792	-0.131	0.950
Legend:	Not-significant		p < .05				p < .01		p < .001					

Figure C-8: *Δrelative repeatability* regression model for SIFT on scene *house*.

Δrelative repeatability, SIFT, junkyard														
Linear regression model:	RelRepeatability ~ 1 + HTD + CLD*Config + DCD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.602	0.200			-3.681	1.759			-0.183	0.283	-0.845	0.116	0.543	0.698
noise	-0.322	0.294			2.699	2.451			0.133	0.434			-0.535	0.973
distortion	-0.032	0.426			0.014	2.642			-0.195	0.497			0.186	1.234
aperture	-0.215	0.248			2.405	2.145			0.274	0.416			-0.596	0.879
HDR	-0.354	0.254			3.100	2.194			1.137	0.511			-1.120	0.908
bloom	0.017	0.301			0.161	2.643			-0.017	0.430			-0.175	0.947
blur	-0.607	0.473			6.496	3.855			0.712	0.635			-1.031	0.815
SSAA	-0.181	0.279			2.200	2.434			0.037	0.483			-0.325	0.947
MSAA	-0.313	0.289			8.285	2.458			-1.693	0.404			-3.244	0.954
FXAA	-0.243	0.308			2.554	2.525			0.031	0.452			-0.199	0.965
SMAA	-0.127	0.289			1.801	2.420			0.022	0.402			-0.411	0.984
AToC	-0.183	0.289			3.095	2.448			0.104	0.409			-1.206	0.948
objects high	0.934	0.298			-6.930	2.594			-1.501	0.415			-0.276	0.965
no objects	-0.571	0.278			6.554	2.763			0.061	0.315			-0.177	0.769
modelling errors	-0.287	0.299			2.993	2.513			0.183	0.433			-0.322	0.937
texture low	-0.632	0.445			6.125	3.590			0.848	0.747			-0.593	0.872
surface low	-0.229	0.252			3.998	2.537			0.441	0.587			-2.252	0.944
AF	0.013	0.281			0.210	2.473			-0.122	0.412			-0.184	0.975
shadow	0.263	0.325			-3.234	3.096			-1.048	0.511			1.050	0.969
shadow filter	-0.543	0.333			6.381	3.243			0.663	0.456			-1.692	0.983
SSAO	-1.383	0.283			10.344	2.358			1.986	0.412			-1.683	0.979
Legend:	Not-significant		p < .05				p < .01		p < .001					

Figure C-9: *Δrelative repeatability* regression model for SIFT on scene *junkyard*.

Δrelative repeatability, SIFT, sport														
Linear regression model:	RelRepeatability ~ 1 + CSD + CLD*Config + SCD*Config + DCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-0.119	0.409	1.450	0.772	0.054	1.305	-0.075	1.117	0.032	0.119	0.417	1.503	0.249	1.378
noise	-0.953	0.563			9.072	1.851	-0.521	1.701	0.750	0.207	-0.532	2.080	-3.489	2.229
distortion	-1.301	0.658			3.943	1.883	-0.829	1.500	0.338	0.166	1.072	2.098	4.083	2.263
aperture	-0.252	0.429			-1.334	1.924	-5.961	1.732	-0.173	0.151	6.681	1.922	-8.829	2.618
HDR	0.060	0.560			0.015	2.022	-0.179	1.622	0.014	0.192	0.045	2.216	-0.304	2.022
bloom	0.081	0.509			-0.078	1.820	-0.053	1.458	0.043	0.169	-0.161	1.997	-0.502	1.895
blur	1.020	0.507			-2.623	2.049	-4.852	1.861	0.245	0.175	4.268	1.903	-5.939	2.799
SSAA	1.802	0.566			-12.810	1.756	-2.166	1.555	-0.415	0.168	-2.351	2.047	6.621	2.146
MSAA	0.024	0.564			0.235	1.811	-0.732	1.600	0.049	0.173	0.488	2.046	0.685	1.923
FXAA	2.637	0.518			-10.359	1.849	-3.496	1.639	-0.426	0.174	-4.105	2.112	3.538	2.573
SMAA	2.505	0.556			-10.566	1.914	-3.026	1.597	-0.643	0.170	-4.003	2.263	4.185	1.952
AToC	0.114	0.556			0.123	1.868	-0.091	1.720	0.008	0.189	-0.384	2.063	-0.210	2.020
objects high	0.062	0.564			0.063	1.819	-0.219	1.603	0.009	0.190	0.051	2.001	-0.170	2.034
no objects	0.498	0.461			-1.637	1.976	0.990	1.551	0.123	0.247	-1.992	2.183	-3.686	2.684
modelling errors	0.010	0.562			-0.272	1.852	0.004	1.610	-0.006	0.172	-0.019	2.145	0.248	2.039
texture low	0.562	0.534			1.857	2.426	0.395	2.079	0.211	0.203	0.120	1.757	-7.891	3.645
surface low	0.156	0.476			-0.105	2.034	-0.046	1.389	0.062	0.224	-0.729	1.690	-0.112	2.844
AF	0.016	0.560			-0.089	1.884	0.078	1.512	0.016	0.182	-0.129	2.083	-0.146	1.963
shadow	-0.043	0.481			0.638	2.093	-0.080	1.635	0.027	0.194	0.109	1.847	0.267	1.970
shadow filter	-0.387	0.496			2.288	2.211	1.497	1.809	-0.226	0.196	0.081	1.921	-1.282	1.993
SSAO	0.642	0.561			3.841	1.933	-0.693	1.586	0.227	0.173	-2.898	2.109	-4.468	2.042
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-10: *Δrelative repeatability* regression model for SIFT on scene *sport*.

Δrelative repeatability, SIFT, street														
Linear regression model:	RelRepeatability ~ 1 + CSD*Config + CLD*Config + SCD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.223	0.332	-0.647	2.074	1.421	2.724	-0.156	1.001						
noise	-0.070	0.460	0.274	2.952	-0.305	3.898	-0.246	1.396						
distortion	-0.066	0.450	-0.009	2.929	-0.017	4.285	0.063	1.412						
aperture	-1.004	0.365	3.465	3.128	9.893	5.092	0.051	1.478						
HDR	-0.009	0.451	0.076	2.941	-0.493	3.923	0.060	1.399						
bloom	-0.083	0.412	-0.017	3.253	-0.403	4.332	0.386	1.456						
blur	0.307	0.422	3.548	3.286	-6.298	3.997	-2.168	1.447						
SSAA	-0.012	0.452	0.182	2.908	-0.541	3.983	0.062	1.387						
MSAA	0.019	0.450	-0.293	2.928	0.007	3.936	0.082	1.395						
FXAA	-0.039	0.448	-0.128	3.031	-0.304	3.900	0.187	1.413						
SMAA	-0.084	0.458	1.825	3.023	-0.987	3.968	-0.633	1.413						
AToC	0.009	0.451	-0.153	2.922	-0.330	3.898	0.088	1.386						
objects high	0.009	0.453	0.065	2.943	-0.585	3.907	0.012	1.390						
no objects	0.385	0.447	11.512	3.746	-4.565	5.161	-6.413	1.787						
modelling errors	1.087	0.455	-7.581	2.956	3.284	3.917	-1.395	1.400						
texture low	0.437	0.432	1.288	3.169	1.369	3.740	-3.183	1.499						
surface low	0.315	0.400	-3.773	3.107	2.507	3.530	-0.923	1.455						
AF	0.030	0.457	-0.054	2.838	-0.360	3.937	-0.054	1.392						
shadow	1.516	0.469	-1.515	2.964	-8.913	3.948	-5.114	1.409						
shadow filter	0.003	0.451	-0.139	2.950	-0.249	3.944	0.102	1.395						
SSAO	-0.783	0.453	14.266	2.954	-7.163	3.930	-3.998	1.395						
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-11: *Δrelative repeatability* regression model for SIFT on scene *street*.

SIFT Δabsolute Repeatability

Δabsolute repeatability, SIFT, concrete														
Linear regression model:	AbsRepeatability ~ 1 + CSD + CLD*Config + SCD*Config + DCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-204	776	-3268	1368	-9967	2733	4765	3076	-1076	542	663	1585	-1403	1508
noise	-207	1069			2362	3817	-163	4376	295	751	-589	2306	-415	2001
distortion	-89	1054			1059	4006	1742	4779	185	672	-791	2432	-582	1910
aperture	80	897			6082	6271	-3441	3255	1049	639	125	1889	717	2109
HDR	290	860			1949	5008	-2552	3631	1106	679	401	1990	-422	2273
bloom	-516	909			826	4108	2048	3901	1103	821	-1679	2205	1931	2222
blur	2646	1043			-4476	4568	-2018	4378	-255	723	-3505	2204	748	1795
SSAA	259	1052			-205	3743	516	4439	98	768	-1335	2242	-258	2132
MSAA	217	1052			1187	3809	-1006	4594	174	769	-166	2313	-41	2033
FXAA	-6899	1060			4985	3930	19244	4393	2179	684	4526	2146	988	2305
SMAA	-17	1078			1869	3766	-963	4517	343	765	147	2274	302	2086
AToC	3	1026			2627	3783	-763	4272	479	769	-350	2279	289	2112
objects high	182	1051			610	3830	-1066	4422	539	745	-194	2296	232	2018
no objects	-163	946			13617	5082	-3387	3239	992	733	-2439	1828	2388	2031
modelling errors	-90	1046			1939	3906	-211	4464	349	751	-420	2378	231	2045
texture low	-1787	1121			31163	4365	3886	4295	726	668	-8828	2128	11090	2049
surface low	-43	963			16353	4082	-8613	3934	871	747	-1652	1931	5118	3302
AF	-26	1044			5471	3796	-7319	4414	1896	780	2736	2289	3309	2003
shadow	-212	1021			1896	3937	1138	4422	456	698	-1690	2181	690	1849
shadow filter	-169	1065			-393	3819	1226	4377	578	678	-1533	2112	1435	1844
SSAO	124	1031			300	3832	154	4287	115	757	-680	2230	-229	2053
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-12: *Δabsolute repeatability* regression model for SIFT on scene *concrete*.

Δabsolute repeatability, SIFT, forest														
Linear regression model:	AbsRepeatability ~ 1 + CSD + CLD + SCD + DCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-631	1068	-8892	875	6716	1351	5717	1013	-1061	466	334	4384	-2075	1361
noise	-1224	1358							194	745	4979	5798	1740	1793
distortion	-1866	1366							476	725	7541	5818	1291	1876
aperture	-232	1152							2802	1684	-2100	5011	-2533	2649
HDR	-1124	1378							164	693	5440	6227	-236	1717
bloom	914	1401							1044	658	-7544	6226	2019	1790
blur	-6771	1236							4099	803	13872	5599	11755	2664
SSAA	631	1281							-767	807	-2430	5546	-2148	1948
MSAA	-236	1341							154	650	414	5981	1178	1787
FXAA	-206	1287							-91	709	328	5612	1784	1843
SMAA	25	1388							-9	702	-363	6064	444	1791
AToC	-589	1385							80	641	2462	6189	520	1780
objects high	-693	1382							295	683	2982	6024	79	1782
no objects	-20	1229							3004	690	-1151	5014	-897	1961
modelling errors	-392	1351							336	654	915	6034	928	1739
texture low	170	1170							183	652	680	4866	1271	2052
surface low	-494	1123							623	601	-257	4548	1874	2218
AF	-795	1480							14	648	3117	6567	728	1797
shadow	-892	1308							532	690	3178	5667	3367	1900
shadow filter	-1509	1320							560	699	5539	5725	4302	1950
SSAO	-568	1404							-109	710	2001	6205	2263	1910
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-13: *Δabsolute repeatability* regression model for SIFT on scene *forest*.

Δabsolute repeatability, SIFT, hangar														
Linear regression model:	AbsRepeatability ~ 1 + CLD + HTD + CSD*Config + SCD*Config + DCD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-3626	481	6254	1965	-3453	1748	6694	1289	-1300	785	1448	464		
noise	-232	629	-266	2764			275	1829	630	1093				
distortion	163	670	-516	2994			156	1822	220	1253				
aperture	1211	541	-1353	2737			-1050	1714	-843	1056				
HDR	125	703	-1012	2981			275	1759	660	1181				
bloom	269	642	-1332	2817			-415	1791	1112	1145				
blur	-1196	777	24179	3415			-7185	1784	-8210	1346				
SSAA	526	616	554	2871			-1944	1817	-102	1151				
MSAA	325	619	-222	2730			-817	1828	406	1073				
FXAA	546	644	560	3110			-1354	1793	-260	1345				
SMAA	212	624	4076	2761			-2974	1807	-725	1109				
AToC	253	623	96	2853			-806	1839	335	1100				
objects high	402	626	-1303	2955			-526	1818	913	1145				
no objects	1420	561	3723	3148			-7135	1857	2984	1021				
modelling errors	-1681	659	3895	3022			2985	1848	1089	1138				
texture low	1551	689	-3150	3458			594	1870	623	1518				
surface low	-280	598	80	2645			-90	1591	-189	1383				
AF	471	621	-653	2857			-1480	1822	426	1165				
shadow	1726	578	-4747	2792			-1674	1722	1301	1422				
shadow filter	1283	573	-4155	2807			-607	1806	643	1332				
SSAO	421	692	-1940	2962			-296	1737	1163	1199				
Legend:	Not-significant		p < .05				p < .01			p < .001				

Figure C-14: *Δabsolute repeatability* regression model for SIFT on scene *hangar*.

Δabsolute repeatability, SIFT, heath														
Linear regression model:	AbsRepeatability ~ 1 + CSD + SCD + EHD + CLD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-35	177	4386	668	-2849	1221	-2231	442					-573	178
noise	-219	152			-64	1678								
distortion	30	154			205	1670								
aperture	465	180			-4794	1917								
HDR	335	165			-3384	1813								
bloom	-1	151			-125	1663								
blur	393	155			1133	1684								
SSAA	-47	154			346	1706								
MSAA	-5	152			-45	1672								
FXAA	129	152			-621	1674								
SMAA	-18	151			-14	1667								
AToC	-31	151			342	1663								
objects high	-13	152			151	1670								
no objects	43	171			-1096	1879								
modelling errors	-27	151			128	1659								
texture low	85	157			3468	1690								
surface low	-662	168			-2679	2124								
AF	-192	154			914	1691								
shadow	-64	150			1333	1671								
shadow filter	101	150			-524	1671								
SSAO	238	151			-4141	1660								
Legend:	Not-significant		p < .05				p < .01			p < .001				

Figure C-15: *Δabsolute repeatability* regression model for SIFT on scene *heath*.

Δabsolute repeatability, SIFT, house														
Linear regression model:	AbsRepeatability ~ 1 + SCD + HTD + CSD*Config + CLD*Config + DCD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-71	721	-4450	3979	788	3659	-2439	437	-749	502	2163	469	6521	1856
noise	-363	897	1356	5282	-1017	4981			-62	690			760	3211
distortion	-263	840	-1639	5267	4580	5611			357	779			-717	2668
aperture	-1571	723	7459	4282	-4284	4813			1960	920			-1185	2357
HDR	-1211	774	5030	5182	-3115	5438			658	986			813	2941
bloom	-1378	753	4934	4757	-451	5279			952	831			-427	2559
blur	-337	948	3648	5859	6299	5927			789	674			-6715	2248
SSAA	-383	878	1183	5322	-23	5116			166	680			691	2524
MSAA	-676	884	6322	5331	-10731	4956			1218	688			1355	2587
FXAA	-234	913	3126	5359	-2634	4895			562	720			-1723	2371
SMAA	-130	861	250	5261	-10	5034			167	676			448	2560
AToC	-130	908	348	5478	-473	5069			144	714			716	2714
objects high	-139	891	225	5291	81	4962			122	688			209	2599
no objects	-1519	710	955	4585	16652	5584			1892	752			-7658	2087
modelling errors	-1620	864	5473	5118	3192	5478			342	732			2103	3101
texture low	2065	758	2921	4719	-5413	5263			-2218	689			-10822	2129
surface low	-2626	786	10701	4459	-1175	5072			-719	912			71	2737
AF	-404	904	-352	5429	154	5102			26	686			3392	3569
shadow	-468	880	2535	5429	869	5342			115	823			-1733	2506
shadow filter	-362	871	1458	5268	1180	5228			189	779			-1208	2636
SSAO	303	904	-2382	5608	1191	5165			-136	697			848	2688
Legend:	Not-significant		p < .05		p < .01		p < .001							

Figure C-16: *Δabsolute repeatability* regression model for SIFT on scene *house*.

Δabsolute repeatability, SIFT, junkyard														
Linear regression model:	AbsRepeatability ~ 1 + SCD + CSD*Config + CLD*Config + DCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-845	664	-2841	2612	1446	8275	2792	921	-841	884	-875	1824	463	2157
noise	-146	923	3899	3650	-5396	11538			441	1285	-1441	2597	587	2912
distortion	-953	1300	-1116	3460	3395	12074			-92	1422	448	2352	3986	3235
aperture	1106	834	8490	4051	-12266	11753			730	1363	-4267	2732	-1114	2531
HDR	-1013	1003	-1641	4305	9682	12429			1057	1632	2108	3066	-3956	2709
bloom	-74	945	-40	3563	1081	12448			-11	1261	291	2356	-653	2830
blur	1862	1390	14815	3555	-21497	13954			5584	1640	-6669	2319	-837	2567
SSAA	564	1068	2995	3979	-8115	13792			-1110	1435	-307	2527	-145	2798
MSAA	286	952	3188	3735	1681	12365			-3557	1281	216	2544	-5984	2924
FXAA	253	1030	3248	3991	-5379	13104			-401	1386	-640	2594	144	3089
SMAA	70	910	1788	3605	-2586	11804			-222	1211	148	2472	-939	2998
AToC	98	918	1845	3648	-1151	11880			-193	1278	-316	2576	-1520	2949
objects high	2371	943	4965	3552	-23375	12156			-2038	1312	-3726	2528	2513	2975
no objects	-1163	791	6420	3682	-5980	12509			1069	953	1842	2362	-348	2435
modelling errors	-525	900	2252	4021	-1696	12079			-208	1307	1420	2605	116	3044
texture low	293	1261	13610	3063	-21876	12491			5699	1843	-1410	2465	2087	2504
surface low	-430	800	3505	3707	1663	12101			1256	1610	-1505	2328	-1705	2702
AF	199	914	871	3671	-3396	12437			-535	1291	306	2504	-591	3004
shadow	2126	1065	7740	3663	-34646	13536			-4393	1672	187	2483	3782	2720
shadow filter	906	1094	10003	3599	-18665	14025			-214	1495	-2202	2631	-1563	2770
SSAO	-3009	937	-2245	4035	23893	12647			4595	1304	277	2553	-2072	2998
Legend:	Not-significant		p < .05		p < .01		p < .001							

Figure C-17: *Δabsolute repeatability* regression model for SIFT on scene *junkyard*.

Δabsolute repeatability, SIFT, sport														
Linear regression model:	AbsRepeatability ~ 1 + CLD*Config + SCD*Config + DCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-230	1097			-1294	3543	649	3010	169	319	1332	4082	3495	3707
noise	-1505	1529			17958	5026	-1987	4619	1536	563	-1178	5650	-9161	6040
distortion	-2733	1786			9365	5107	-2503	4074	791	450	1839	5699	8504	6144
aperture	-44	1157			-4480	5119	-11248	4701	-657	407	13347	5207	-22869	7089
HDR	188	1523			-755	5494	-1118	4406	46	521	1320	6020	-1293	5490
bloom	283	1380			-1019	4943	-1011	3959	108	459	861	5424	-1512	5143
blur	4960	1375			-8691	5483	-14321	5052	727	474	14871	5168	-29451	7534
SSAA	4015	1537			-27933	4771	-4248	4223	-1006	456	-5359	5559	13219	5819
MSAA	10	1533			283	4919	-2452	4346	84	469	2448	5554	2076	5223
FXAA	6732	1407			-24422	5019	-9526	4448	-1027	473	-9130	5735	7648	6986
SMAA	5442	1510			-23477	5198	-7003	4339	-1465	461	-7161	6146	7802	5303
AToC	446	1509			719	5075	-1349	4670	95	512	-327	5603	-1065	5483
objects high	251	1532			-105	4942	-1439	4355	37	515	1031	5435	-558	5522
no objects	1330	1236			-916	5243	-482	4194	398	671	-5391	5858	-8173	7198
modelling errors	175	1525			-750	5029	-508	4374	-27	467	173	5828	149	5537
texture low	3848	1445			8312	6515	3855	5642	797	552	-303	4766	-40000	9796
surface low	324	1280			-582	5432	-116	3772	277	609	-2345	4584	-262	7725
AF	208	1520			-1387	5117	-490	4107	15	495	184	5659	-708	5330
shadow	-915	1307			6009	5676	1772	4441	146	526	1235	5018	-955	5350
shadow filter	-1587	1345			7827	5982	5559	4913	-265	532	-121	5216	-3759	5412
SSAO	1139	1524			6821	5251	-1446	4309	574	469	-4961	5730	-8578	5541
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-18: *Δrelative repeatability* regression model for SIFT on scene *sport*.

Δabsolute repeatability, SIFT, street														
Linear regression model:	RelRepeatability ~ 1 + CSD*Config + CLD*Config + SCD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-769	882	-5666	4949	8688	1844	6201	2795					-1430	664
noise	68	1217	-1310	6575			-549	3725						
distortion	155	1196	-3074	6782			1246	3776						
aperture	372	950	17572	6848			-12306	4019						
HDR	277	1194	-3229	6532			586	3733						
bloom	252	1066	-4801	7106			1479	3391						
blur	2071	1125	-9574	7310			-101	4243						
SSAA	164	1196	-3293	6361			1066	3705						
MSAA	279	1193	-3048	6496			496	3719						
FXAA	237	1187	-4868	6764			1808	3779						
SMAA	149	1208	-696	6664			-500	3776						
AToC	345	1196	-3276	6544			312	3698						
objects high	446	1201	-3808	6572			158	3707						
no objects	1841	1099	19812	7089			-21385	4701						
modelling errors	3656	1205	-23698	6579			-4008	3732						
texture low	3849	1143	4301	7397			-14546	4322						
surface low	866	1067	-6257	7519			-4010	3888						
AF	350	1207	-3547	6256			114	3724						
shadow	3248	1243	-9922	6723			-10184	3754						
shadow filter	359	1195	-3123	6545			210	3716						
SSAO	-1235	1198	26720	6557			-10875	3725						
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-19: *Δrelative repeatability* regression model for SIFT on scene *street*.

MSER Δrelative Repeatability

Δrelative repeatability, MSER, concrete														
Linear regression model:	RelRepeatability ~ 1 + CSD*Config + CLD*Config + SCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-1.162	0.640	1.939	3.326	0.564	1.430	1.716	1.614			0.976	0.987	0.103	0.780
noise	-0.299	0.896	0.866	4.449	1.392	2.038	0.106	2.313			-0.187	1.404	0.317	1.030
distortion	-0.130	0.808	-0.211	4.197	0.970	2.112	0.054	2.486			0.599	1.419	-0.468	1.019
aperture	0.538	0.688	-0.558	3.993	3.499	3.380	-2.728	1.744			1.232	1.150	-2.819	1.097
HDR	0.543	0.681	0.210	3.973	-3.152	2.707	-0.013	1.859			0.523	1.190	-2.938	1.181
bloom	0.630	0.699	-4.488	4.413	3.064	2.177	-0.871	2.230			0.124	1.370	-0.279	1.106
blur	-0.321	0.841	-0.444	4.957	9.893	2.897	0.916	2.234			-2.405	1.273	-0.260	0.899
SSAA	-0.215	0.876	0.939	4.503	0.280	1.954	0.358	2.322			-0.112	1.348	0.209	1.117
MSAA	-0.490	0.922	1.406	4.703	2.004	1.991	0.651	2.441			-0.467	1.467	0.757	1.091
FXAA	-0.170	0.917	2.427	4.565	2.074	2.049	2.653	2.318			-2.051	1.277	-6.099	1.268
SMAA	-0.083	0.890	-0.126	4.510	0.091	1.969	-0.116	2.369			0.517	1.394	0.051	1.108
AToC	-0.356	0.849	0.944	4.403	0.595	1.976	0.640	2.230			-0.191	1.364	0.684	1.076
objects high	-0.193	0.848	-0.136	4.535	1.531	2.004	0.644	2.311			-0.368	1.369	0.089	1.058
no objects	0.820	0.698	-9.038	4.133	12.623	4.198	-1.528	1.733			-0.594	1.148	-0.487	1.027
modelling errors	0.222	0.874	-2.179	4.694	0.966	2.068	-0.363	2.332			0.543	1.486	-0.280	1.065
texture low	-0.505	0.776	-8.061	4.088	9.258	2.177	3.908	2.254			-1.149	1.219	2.012	0.997
surface low	0.865	0.749	-7.321	4.738	5.481	2.493	-2.088	2.093			0.351	1.139	-0.753	1.632
AF	1.214	0.868	-7.182	4.598	1.841	1.980	-3.892	2.285			2.700	1.368	-0.411	1.030
shadow	0.078	0.826	-1.580	4.612	2.097	2.213	0.092	2.350			0.237	1.260	-0.564	0.909
shadow filter	-0.412	0.834	0.289	4.613	0.092	2.146	1.775	2.310			0.024	1.232	-0.307	0.910
SSAO	-0.022	0.840	-0.569	4.488	0.137	2.013	0.712	2.232			-0.387	1.349	0.053	1.057
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-20: *Δrelative repeatability* regression model for MSER on scene *concrete*.

Δrelative repeatability, MSER, forest														
Linear regression model:	RelRepeatability ~ 1 + CSD + DCD + CLD*Config + SCD*Config + HTD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.752	0.602	-1.661	0.263	-0.817	1.411	-0.441	1.502	0.097	0.037	-1.243	1.448		
noise	-0.447	0.803			1.268	1.809	0.316	2.160			1.105	1.810		
distortion	-0.410	0.876			-0.668	1.933	0.527	2.337			1.536	1.924		
aperture	-0.991	0.753			2.576	2.141	2.539	2.010			-0.042	1.653		
HDR	-0.135	0.835			-0.027	1.678	1.945	2.181			-2.195	2.077		
bloom	0.403	0.918			-0.260	1.995	-1.306	2.221			0.100	2.198		
blur	-3.837	0.785			-2.316	2.025	12.677	2.191			1.397	1.869		
SSAA	-0.712	0.757			0.580	1.764	2.303	2.069			-0.142	1.833		
MSAA	-0.818	0.802			1.561	1.826	1.716	2.266			0.600	2.047		
FXAA	-0.872	0.813			0.295	1.857	3.124	2.177			-0.429	1.875		
SMAA	0.058	0.807			-0.057	1.806	0.686	2.059			-1.311	1.903		
AToC	-1.265	0.837			2.414	1.905	0.869	2.196			3.438	2.127		
objects high	-0.338	0.829			1.279	1.849	-0.380	2.117			1.387	1.915		
no objects	-1.731	0.666			4.580	1.708	0.118	1.665			3.782	1.664		
modelling errors	-0.104	0.833			0.189	1.900	-0.126	2.014			0.517	1.981		
texture low	0.758	0.867			-3.365	2.342	-1.537	2.118			1.050	1.671		
surface low	-0.128	0.822			-1.524	2.014	0.337	2.253			1.049	1.540		
AF	0.037	0.885			0.586	1.880	-0.172	2.125			-0.476	2.073		
shadow	-0.819	0.803			2.081	2.188	1.314	1.970			1.011	1.899		
shadow filter	-0.932	0.770			2.750	2.236	0.812	2.017			1.912	1.975		
SSAO	-0.251	0.781			0.762	1.954	0.793	1.950			-0.241	2.056		
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-21: *Δrelative repeatability* regression model for MSER on scene *forest*.

Δrelative repeatability, MSER, hangar														
Linear regression model:	RelRepeatability ~ 1 + CSD + DCD + HTD + CLD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.315	0.261	0.757	0.253	-4.518	1.843			-0.156	0.089	0.740	0.143	-0.192	0.709
noise	-0.046	0.349			0.560	2.433							-0.427	1.044
distortion	0.278	0.413			-1.858	2.867							0.038	0.916
aperture	0.047	0.409			-2.093	3.164							0.254	0.879
HDR	0.076	0.389			-0.393	2.690							-0.253	1.005
bloom	-0.190	0.360			1.266	2.453							0.074	1.016
blur	-0.396	0.486			3.042	3.468							0.356	0.790
SSAA	0.034	0.378			0.082	2.620							-0.329	0.966
MSAA	0.039	0.374			-0.106	2.563							-0.262	1.005
FXAA	-0.163	0.443			0.291	3.093							1.034	0.931
SMAA	-1.487	0.362			14.376	2.515							-5.172	0.963
AToC	-0.060	0.361			0.550	2.480							-0.084	1.007
objects high	-0.028	0.364			0.318	2.503							-0.118	0.987
no objects	-0.175	0.362			-1.649	3.194							0.428	0.747
modelling errors	0.028	0.367			-0.747	2.532							-0.062	1.013
texture low	0.407	0.397			-3.018	2.808							-0.310	0.786
surface low	-0.731	0.358			4.330	2.383							0.910	0.874
AF	0.089	0.359			-0.474	2.552							-0.437	1.041
shadow	0.110	0.320			-0.994	2.372							-0.403	0.931
shadow filter	0.127	0.324			-1.140	2.373							-0.361	0.970
SSAO	-0.011	0.367			0.285	2.530							-0.183	0.976
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-22: *Δrelative repeatability* regression model for MSER on scene *hangar*.

Δrelative repeatability, MSER, heath														
Linear regression model:	RelRepeatability ~ 1 + CSD + HTD + EHD + CLD*Config + SCD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.310	0.269	-3.525	0.368	0.600	0.763	1.136	0.730			-0.619	0.274	0.413	0.103
noise	-0.136	0.376			0.309	1.058	0.300	0.985						
distortion	0.038	0.362			0.077	1.040	-0.095	0.953						
aperture	0.152	0.310			4.162	1.102	-2.775	0.994						
HDR	0.036	0.336			2.861	1.033	-1.458	1.009						
bloom	-0.222	0.363			0.412	1.030	0.599	0.954						
blur	-0.029	0.367			1.941	1.049	-0.040	0.963						
SSAA	-0.191	0.353			1.100	1.038	0.277	0.954						
MSAA	-0.128	0.367			0.190	1.048	0.366	0.961						
FXAA	0.092	0.365			0.234	1.033	-0.244	0.960						
SMAA	-0.039	0.384			0.338	1.048	0.015	1.019						
AToC	0.010	0.367			-0.173	1.039	0.037	0.964						
objects high	-0.068	0.362			0.076	1.034	0.184	0.950						
no objects	0.214	0.321			5.759	1.081	-3.738	1.048						
modelling errors	-0.069	0.376			0.177	1.043	0.184	0.992						
texture low	0.380	0.373			1.344	1.046	-1.403	0.995						
surface low	0.410	0.419			3.320	1.392	-2.187	1.139						
AF	-0.128	0.370			-0.054	1.054	0.392	0.972						
shadow	0.116	0.324			-1.794	0.985	0.219	0.879						
shadow filter	0.146	0.327			-0.633	0.984	-0.330	0.888						
SSAO	-0.343	0.359			-1.078	1.038	1.257	0.942						
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-23: *Δrelative repeatability* regression model for MSER on scene *heath*.

Δrelative repeatability, MSER, house														
Linear regression model:	RelRepeatability ~ 1 + CSD + CLD*Config + DCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-0.938	0.250	0.844	0.324	2.329	0.995			-0.270	0.207	1.411	0.705	1.038	0.893
noise	-0.088	0.328			0.496	1.347			-0.039	0.288	-0.313	0.941	1.230	1.460
distortion	-0.258	0.434			0.373	1.399			-0.127	0.326	0.601	1.028	0.626	1.314
aperture	0.415	0.307			0.149	1.305			0.322	0.396	-1.049	0.976	-1.031	1.078
HDR	0.093	0.369			0.024	1.452			0.371	0.390	-0.715	1.225	0.689	1.454
bloom	0.306	0.333			0.340	1.315			0.494	0.326	-1.199	1.029	-0.527	1.206
blur	1.515	0.301			-4.342	1.422			-0.471	0.265	-2.511	0.952	-1.000	1.012
SSAA	-0.187	0.328			0.850	1.382			0.054	0.287	0.091	0.977	0.703	1.219
MSAA	1.038	0.338			-3.305	1.342			0.895	0.290	-2.980	0.995	-0.914	1.269
FXAA	0.822	0.315			-1.314	1.381			0.524	0.291	-2.384	0.955	-1.077	1.131
SMAA	-0.064	0.346			0.030	1.385			-0.064	0.289	0.107	1.051	0.408	1.273
AToC	-0.091	0.343			-0.268	1.367			-0.053	0.295	0.232	0.985	0.827	1.343
objects high	0.040	0.338			-0.166	1.361			-0.057	0.302	-0.087	1.003	0.157	1.254
no objects	0.044	0.344			1.127	1.445			0.221	0.309	-0.120	1.057	-1.003	0.975
modelling errors	0.421	0.337			-2.631	1.490			0.183	0.344	-0.819	1.035	0.141	1.541
texture low	1.239	0.423			-4.606	1.394			-0.345	0.300	-1.813	1.196	-1.660	1.024
surface low	-0.503	0.644			1.297	1.804			-0.341	0.440	0.213	1.359	1.709	1.336
AF	-0.167	0.363			0.086	1.345			-0.033	0.287	-0.090	0.978	1.726	1.665
shadow	0.273	0.332			-0.222	1.369			-0.445	0.348	-0.414	0.885	-0.072	1.153
shadow filter	0.146	0.331			-0.623	1.371			-0.447	0.334	0.027	0.896	0.559	1.208
SSAO	-0.032	0.345			-0.399	1.411			-0.069	0.295	0.194	1.051	0.438	1.336
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-24: *Δrelative repeatability* regression model for MSER on scene *house*.

Δrelative repeatability, MSER, junkyard														
Linear regression model:	RelRepeatability ~ 1 + HTD + CSD*Config + CLD*Config + SCD*Config + DCD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-0.510	0.569	-1.416	0.879	-1.435	3.522	3.858	1.877	-0.332	0.398	-0.408	0.156		
noise	-0.108	0.791	0.417	1.160	-1.577	4.939	0.655	2.566	0.006	0.607				
distortion	0.531	0.848	-0.363	1.174	-0.781	5.113	-0.980	2.767	-0.531	0.666				
aperture	1.889	0.655	5.089	1.553	-9.714	5.281	-7.082	2.343	0.572	0.688				
HDR	1.244	0.751	1.559	1.351	0.111	5.053	-5.672	2.550	-0.676	0.658				
bloom	0.202	0.815	-0.011	1.220	0.085	5.086	-0.732	2.513	-0.070	0.596				
blur	0.541	0.998	1.938	1.249	-1.546	5.678	-1.877	2.933	0.210	0.766				
SSAA	0.422	0.763	1.038	1.262	-4.525	5.447	0.158	2.514	-0.807	0.650				
MSAA	0.182	0.813	1.245	1.301	-4.690	5.413	0.452	2.566	-0.192	0.595				
FXAA	0.165	0.816	0.803	1.290	-3.579	5.544	0.593	2.565	-0.400	0.640				
SMAA	-0.176	0.795	0.573	1.220	-2.058	5.038	1.020	2.563	0.000	0.579				
AToC	-0.171	0.783	0.784	1.256	-2.333	5.195	1.007	2.503	-0.003	0.585				
objects high	1.457	0.834	1.516	1.229	-10.564	5.207	-1.459	2.597	-1.742	0.608				
no objects	-0.365	0.685	3.860	1.263	4.066	4.433	-2.748	2.215	0.133	0.425				
modelling errors	0.477	0.853	1.212	1.370	-3.895	5.209	-0.758	2.679	-0.445	0.611				
texture low	0.738	1.047	-0.762	1.209	-5.213	5.742	0.651	2.951	-0.598	0.921				
surface low	0.759	0.656	2.453	1.318	2.295	4.757	-5.401	2.630	-0.547	0.850				
AF	0.118	0.838	0.648	1.277	-2.867	5.127	0.211	2.623	-0.153	0.622				
shadow	-1.021	1.098	2.956	1.251	-5.923	5.367	3.907	3.138	-0.125	0.919				
shadow filter	1.170	1.158	4.940	1.245	-15.180	5.703	-1.442	3.200	-1.014	0.833				
SSAO	-1.407	0.838	1.339	1.343	9.300	5.382	-1.159	2.637	2.106	0.624				
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-25: *Δrelative repeatability* regression model for MSER on scene *junkyard*.

Δrelative repeatability, MSER, sport														
Linear regression model:	RelRepeatability ~ 1 + CSD*Config + CLD*Config + SCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	0.338	0.510	-8.176	3.218	0.319	1.204	0.471	1.438			1.052	1.742	-0.190	1.680
noise	0.281	0.709	0.697	4.615	-1.121	1.650	-0.893	2.101			0.259	2.415	-1.231	2.415
distortion	-0.358	0.773	-10.931	5.344	-0.866	1.925	-1.433	1.982			4.474	2.446	7.417	3.127
aperture	-1.069	0.534	8.350	4.696	-0.797	2.446	-7.127	2.087			8.233	2.266	-7.472	3.083
HDR	0.234	0.719	-0.192	4.574	0.046	1.827	-0.722	2.085			0.054	2.571	-0.274	2.369
bloom	0.169	0.609	-1.627	4.354	0.204	1.817	0.576	1.968			-0.639	2.318	-0.396	2.402
blur	1.029	0.637	14.343	5.596	-9.015	2.819	-8.082	2.464			6.404	2.176	-5.695	3.403
SSAA	0.523	0.710	4.649	4.461	-2.499	1.729	-3.250	1.990			-0.304	2.381	0.805	2.516
MSAA	0.066	0.703	-0.285	4.733	-0.730	1.745	-0.115	2.139			0.164	2.425	0.270	2.387
FXAA	1.472	0.670	-7.491	4.824	-1.981	1.769	-2.104	2.048			-0.581	2.456	7.629	3.189
SMAA	0.297	0.698	3.608	4.658	-3.545	1.864	-3.197	2.090			1.866	2.648	0.469	2.434
AToC	0.321	0.698	-0.361	4.645	-0.066	1.688	-0.597	2.243			-0.229	2.421	-0.795	2.388
objects high	0.145	0.702	-0.214	4.572	-0.585	1.668	0.071	2.064			-0.298	2.323	-0.466	2.417
no objects	-0.152	0.556	6.568	4.565	1.953	2.595	-4.577	1.900			-0.451	2.638	2.823	3.195
modelling errors	0.126	0.707	-2.311	4.512	-0.354	1.807	0.839	2.069			-0.241	2.491	0.236	2.393
texture low	-0.608	0.632	-7.788	5.909	7.582	3.298	5.345	2.920			1.829	2.073	-9.567	4.761
surface low	-1.393	0.583	9.576	6.042	1.079	3.116	0.205	1.992			0.405	2.013	-0.980	2.868
AF	-0.117	0.719	1.698	4.500	-0.118	1.817	-0.213	1.934			0.078	2.422	-0.976	2.248
shadow	-0.330	0.679	2.067	5.477	1.943	2.203	-1.932	2.090			2.234	2.168	0.846	2.474
shadow filter	-0.627	0.656	0.081	5.406	5.706	2.372	2.037	2.364			-0.430	2.282	-3.263	2.491
SSAO	0.494	0.706	-2.274	4.583	-0.725	1.931	-0.810	2.040			0.762	2.452	-0.862	2.412
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-26: *Δrelative repeatability* regression model for MSER on scene *sport*.

Δrelative repeatability, MSER, street														
Linear regression model:	RelRepeatability ~ 1 + CLD + DCD + EHD + CSD*Config + SCD*Config + HTD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-0.785	0.609	1.536	2.035	-3.638	0.656	2.713	1.928	0.175	0.028	-0.341	1.098	-0.757	0.216
noise	0.034	0.804	0.203	2.726			-0.200	2.536			-0.151	1.498		
distortion	-0.056	0.791	1.199	2.766			0.096	2.505			0.143	1.444		
aperture	0.936	0.618	-0.616	3.076			-6.070	2.141			2.254	1.513		
HDR	0.442	0.777	-0.355	2.694			-0.886	2.465			-0.613	1.432		
bloom	-0.437	0.750	-1.332	2.744			0.726	2.241			1.945	1.607		
blur	-0.714	0.704	5.871	3.077			-1.464	2.454			2.090	1.343		
SSAA	0.349	0.816	-0.266	2.720			-0.981	2.580			-0.214	1.537		
MSAA	0.026	0.790	-1.308	2.737			0.277	2.522			0.348	1.464		
FXAA	-0.007	0.780	-0.366	2.833			-0.068	2.551			0.349	1.431		
SMAA	-0.100	0.758	-0.107	2.757			0.139	2.455			0.292	1.378		
AToC	0.594	0.820	0.612	2.734			-1.972	2.597			-0.759	1.519		
objects high	0.344	0.744	-0.697	2.691			-0.751	2.356			-0.255	1.379		
no objects	1.666	0.676	3.420	2.930			-7.898	2.281			-0.516	1.488		
modelling errors	1.444	0.808	0.178	2.720			-3.860	2.556			-2.208	1.492		
texture low	-1.000	0.716	0.038	3.359			2.112	2.740			2.058	1.313		
surface low	-0.592	0.738	-2.360	2.714			2.388	2.366			2.307	1.336		
AF	0.210	0.793	0.411	2.735			-0.679	2.529			-0.405	1.503		
shadow	0.341	0.819	-7.905	2.679			1.093	2.552			1.513	1.460		
shadow filter	0.538	0.753	-0.694	2.692			-1.319	2.397			-0.482	1.391		
SSAO	-0.239	0.786	2.780	2.776			-0.670	2.520			0.059	1.466		
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-27: *Δrelative repeatability* regression model for MSER on scene *street*.

MSER Δabsolute Repeatability

Δabsolute repeatability, MSER, concrete														
Linear regression model:	AbsRepeatability ~ 1 + DCD + CSD*Config + CLD*Config + SCD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-534	1084	3509	5123	-13934	2539	3824	2760	753	107			-3841	1365
noise	592	1537	-3399	7003	-587	3574	90	3902					-464	1811
distortion	-1338	1415	2753	6778	-2695	3837	5393	4291					298	1806
aperture	2162	1180	-12353	6334	1452	6238	-4995	3009					3842	1930
HDR	1843	1164	-1558	6382	812	4951	-8907	3225					2006	2129
bloom	-145	1198	-1317	6839	-3231	3895	2817	3968					-1124	1880
blur	-607	1474	-10516	8307	2863	5309	2273	3781					7502	1616
SSAA	-378	1506	619	7216	211	3533	1396	3898					-471	1959
MSAA	16	1524	-878	7160	1099	3598	494	3943					-300	1873
FXAA	3065	1607	-3042	7449	-4645	3687	-6066	4031					-7730	2251
SMAA	169	1511	-2259	7086	-996	3541	1299	3967					-579	1885
AToC	276	1472	-1308	7044	1052	3566	-207	3800					-552	1878
objects high	748	1471	-2620	7307	549	3617	-1161	3963					-744	1855
no objects	260	1152	-21274	6398	34060	6646	-4763	2981					7245	1853
modelling errors	660	1480	-2923	7237	384	3634	-676	3969					-704	1846
texture low	-2744	1352	-1765	6691	-2612	3981	8538	4002					7750	1815
surface low	2419	1298	-18196	7893	3503	4553	-991	3693					-608	2973
AF	1412	1506	-4221	7408	1007	3562	-2850	3921					-951	1802
shadow	267	1449	-4984	7808	-408	4029	1212	4074					1891	1599
shadow filter	115	1465	-5123	7770	-1664	3884	2334	4093					1658	1599
SSAO	698	1456	-3337	7157	-26	3598	-430	3885					-692	1845
Legend:	Not-significant		p < .05				p < .01		p < .001					

Figure C-28: *Δabsolute repeatability* regression model for MSER on scene *concrete*.

Δabsolute repeatability, MSER, forest														
Linear regression model:	AbsRepeatability ~ 1 + CSD + CLD + DCD + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-1510	1801	10195	1620	-7127	2569			719	217	-2192	8294	-3891	2444
noise	-622	2221									1647	10460	4130	3174
distortion	-1888	2309									9991	10652	-1853	3424
aperture	303	2086									-210	9460	-2141	4996
HDR	1733	2461									-9452	11807	422	3138
bloom	-2761	2530									10645	11892	6264	3227
blur	-4581	2186									5490	10543	11836	5070
SSAA	385	2296									-1006	10586	-1335	3584
MSAA	-2779	2370									10631	11297	7333	3223
FXAA	856	2256									-5648	10564	1111	3384
SMAA	2073	2312									-9682	10966	-83	3250
AToC	-5558	2538									23986	11884	5580	3281
objects high	-3743	2329									16931	11024	2501	3182
no objects	-63	2121									-2683	9327	3163	3662
modelling errors	-1981	2361									8027	11299	2475	3167
texture low	-566	2113									457	9272	3056	3796
surface low	-1760	2019									4441	8625	5493	4121
AF	659	2593									-5459	12252	3757	3266
shadow	2462	2288									-13091	10699	7443	3237
shadow filter	-392	2330									-1003	10853	12294	3367
SSAO	584	2495									-4504	11721	5194	3470
Legend:	Not-significant		p < .05				p < .01		p < .001					

Figure 0-29: *Δabsolute repeatability* regression model for MSER on scene *forest*.

Δabsolute repeatability, MSER, hangar														
Linear regression model:	AbsRepeatability ~ 1 + DCD + HTD + CSD*Config + CLD*Config + SCD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-4589	1432	13985	2741	-19426	10431	8466	2686	-1161	448	7303	883	-588	3356
noise	-533	2051	-3018	3625	10279	14132	-441	3843					-583	4931
distortion	2830	2410	3059	3696	-21497	17144	-2719	4112					-50	4608
aperture	4213	2126	-12502	4449	1629	16521	-10030	3832					1184	4131
HDR	369	2115	-3041	3628	849	15398	798	3689					-917	4755
bloom	651	2027	2213	3514	-4564	14363	-2107	3670					394	4756
blur	8088	2388	-24470	3716	-4535	20351	-13808	3743					835	3770
SSAA	-89	2059	6483	3700	-9175	14898	521	4015					-682	4930
MSAA	425	2037	-836	3507	186	14767	188	3659					-3289	4757
FXAA	2127	2209	186	3739	-5428	16193	-7289	4207					4751	4816
SMAA	-10185	2023	-14271	3612	127449	15059	-2725	3703					-27343	4628
AToC	-203	2001	-894	3671	3544	14834	155	3749					-1143	4851
objects high	645	1980	-607	3524	-2014	14384	-155	3646					-1721	4687
no objects	2606	1899	-23173	5446	19636	15933	-8579	3532					1680	3560
modelling errors	7452	1979	-17432	3632	6065	13972	-15374	3836					-2291	4871
texture low	3711	2082	-7839	3385	-9348	16541	-5578	4736					-2380	4177
surface low	-2965	2217	-705	6683	16762	21934	404	3770					2370	4623
AF	1424	2044	6137	3694	-18623	14857	-884	3746					2986	4982
shadow	2501	1767	-5391	3527	-5040	13905	-2440	3420					1848	4271
shadow filter	2345	1764	-7103	3579	-4815	13474	-653	3576					887	4439
SSAO	677	2052	934	3592	-4265	14908	-831	3701					-518	4679
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-30: *Δabsolute repeatability* regression model for MSER on scene *hangar*.

Δabsolute repeatability, MSER, heath														
Linear regression model:	AbsRepeatability ~ 1 + DCD + CSD*Config + CLD*Config + SCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	983	981	4992	3617	-12499	2695	42	3210	-425	99	-4858	3136	2479	1195
noise	-1308	1410	-583	4841	1973	3698	1274	4667			2472	4105	1011	1708
distortion	-872	1462	94	5660	2262	4020	-5913	5407			8554	4831	22	2507
aperture	321	1273	-17618	6703	17987	6752	-2461	3802			3327	4458	-2738	1691
HDR	1127	1323	-10933	5405	6082	4405	-782	3686			-35	4480	-254	1625
bloom	-203	1356	-115	4877	573	3609	2659	4674			-2960	4663	1209	1685
blur	573	1410	-1368	8137	-3177	4561	-3048	5257			3734	4383	-2678	1547
SSAA	563	1322	3022	4912	1089	3416	-3351	4010			-273	4738	-44	1631
MSAA	-62	1404	306	4846	702	3764	1500	4507			-1907	4278	205	1640
FXAA	1052	1290	350	5574	-1065	3655	-1444	5171			-2381	4952	693	1829
SMAA	52	1342	1480	5299	-158	3716	-6235	5317			6389	4686	-1082	1790
AToC	-444	1432	77	4912	696	3692	-1084	4615			2391	4632	414	1690
objects high	-815	1360	-1535	4841	2024	3598	2343	4489			166	4221	936	1654
no objects	-1930	1236	-21232	6196	15932	6508	6911	3724			6213	4212	-3074	1423
modelling errors	-290	1368	265	5141	583	3729	-1298	5713			2124	5304	-38	1841
texture low	765	1708	-727	5923	1742	4281	-4646	3751			2031	4295	1	1411
surface low	3768	1693	-15769	7938	21325	7985	-3809	4391			-3271	3415	322	1508
AF	-324	1408	-2616	5041	-427	3620	4865	5076			-2782	4745	1069	1725
shadow	2118	1318	-3386	5301	-5235	3519	-7133	4574			6059	5462	-3844	1655
shadow filter	1398	1360	-2973	5180	-1764	3388	55	4161			-2414	4763	-623	1600
SSAO	-949	1309	3119	4901	-1653	3629	-1527	4701			3692	4368	-503	1731
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-31: *Δabsolute repeatability* regression model for MSER on scene heath.

Δabsolute repeatability, MSER, house														
Linear regression model:	AbsRepeatability ~ 1 + CSD + CLD*Config + SCD*Config + DCD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-10259	2495	-6123	1501	38401	5054	4031	3531	-1967	930	18430	3819	10199	4562
noise	-1235	3843			2288	6994	2550	5575	-158	1309	-427	5334	4085	8172
distortion	1572	3345			-3050	6878	-1830	4882	66	1489	-1668	5210	-6865	6264
aperture	3135	2706			-15329	6725	4819	5378	160	1918	-5635	4965	-10194	5410
HDR	723	3229			-13016	6902	1691	6742	-284	1749	333	5938	1109	6849
bloom	153	2988			-8352	6460	6475	6098	862	1462	-1840	5148	-4734	5819
blur	11149	3042			-38378	7305	-11470	4677	1836	1188	-15319	4903	-7101	5050
SSAA	-4550	3484			7991	6849	5875	5108	-746	1312	6139	5427	2614	6227
MSAA	11417	3581			-22133	6861	-8274	5145	5190	1303	-22574	5407	-14126	6709
FXAA	-3490	3300			6150	6900	10434	4991	2288	1312	-2836	5131	-4728	5644
SMAA	524	3390			-3149	6911	-2528	4856	-390	1291	1898	5526	1380	6366
AToC	-3266	3710			3190	7016	4350	5314	-390	1328	4035	5489	5050	7075
objects high	-263	3629			-464	6956	432	5198	-980	1369	1255	5534	479	6645
no objects	7072	2988			-27458	8355	-2750	7774	884	1430	-12126	5456	-9341	5136
modelling errors	5720	3382			-20330	7131	-3087	4909	1786	1553	-9672	5408	-4404	8101
texture low	8347	3537			-24477	7257	-11365	4679	154	1352	-8797	6077	-6965	5206
surface low	9080	4495			-11221	9569	-11359	5058	-334	2037	-16007	6837	-5431	7136
AF	-2106	3634			2174	6855	1954	5131	-691	1294	2736	5271	7082	8511
shadow	271	3359			5420	7500	1780	5116	-989	1608	-2737	4712	-2907	5881
shadow filter	558	3294			3037	7239	555	4936	-595	1529	-2381	4773	-241	6140
SSAO	448	3452			-4986	7142	-882	4971	-634	1321	1642	5522	841	6671
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-32: *Δabsolute repeatability* regression model for MSER on scene *house*.

Δabsolute repeatability, MSER, junkyard														
Linear regression model:	AbsRepeatability ~ 1 + CLD + SCD + CSD*Config + HTD*Config + EHD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	4666	513	-3981	2051	-8909	2810	-4543	1257			-6596	1921	-4033	2672
noise	33	557	901	2656							-101	2810	-1466	3712
distortion	-1365	871	-1091	2673							-851	2720	10153	4361
aperture	-1836	634	-1185	3184							3397	3193	3313	3395
HDR	-1164	575	-767	2801							1457	2997	2069	3573
bloom	-93	553	-1294	2595							950	2670	385	3553
blur	-444	571	-2392	2805							770	2670	203	3274
SSAA	-1562	572	6492	2663							-113	2832	1410	3557
MSAA	-268	551	2460	2659							148	2743	-1408	3695
FXAA	-1232	565	4214	2840							-83	2887	2017	4035
SMAA	-634	572	3363	2577							92	2700	-467	3790
AToC	-446	552	2366	2649							1034	2795	-1862	3703
objects high	803	548	-588	2641							-4112	2712	1827	3677
no objects	195	804	3838	2868							-6916	2710	4743	3034
modelling errors	-89	569	104	2907							1697	2921	-2612	3979
texture low	353	874	-2046	2483							798	3284	-2710	3137
surface low	-4755	873	7141	2683							10709	2565	-5563	3713
AF	210	559	-137	2656							346	2779	-1203	3587
shadow	-777	555	2149	3029							-163	2792	1406	3349
shadow filter	-955	565	8754	3059							-2120	3025	-1105	3423
SSAO	-1100	564	967	2771							2656	2802	-1029	3774
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-33: *Δabsolute repeatability* regression model for MSER on scene *junkyard*.

Δabsolute repeatability, MSER, sport														
Linear regression model:	AbsRepeatability ~ 1 + EHD + CSD*Config + CLD*Config + SCD*Config + HTD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	6	217	83	1362	136	530	-1698	602			971	790	-1399	192
noise	119	296	-687	1917	-34	729	273	887			-490	1076		
distortion	-21	306	-1098	2056	163	816	623	833			186	1110		
aperture	-362	228	-6223	2069	3984	1104	1380	907			961	968		
HDR	199	307	-1060	1915	720	801	646	869			-1480	1168		
bloom	-83	257	-285	1784	530	800	905	817			-971	1053		
blur	0	275	-6373	2380	2505	1111	2008	909			91	987		
SSAA	-121	303	426	1878	-409	761	310	850			-73	1075		
MSAA	46	298	-430	1983	85	754	401	882			-577	1102		
FXAA	-47	293	-2282	2022	531	779	1080	897			-310	1115		
SMAA	118	297	-478	1940	-54	812	262	853			-662	1203		
AToC	102	295	-976	1935	397	726	771	918			-1138	1099		
objects high	83	297	-843	1907	160	726	507	855			-650	1055		
no objects	-379	236	-5426	1948	3102	1172	1676	784			645	1100		
modelling errors	30	299	-735	1895	-241	795	186	861			155	1131		
texture low	-396	275	-8091	2385	4819	1085	3383	887			-380	940		
surface low	-393	230	-8455	2623	6050	1356	3069	815			-145	914		
AF	14	303	-212	1913	-37	806	328	806			-294	1094		
shadow	423	295	-3478	2257	970	980	-2505	908			3227	963		
shadow filter	185	285	-2978	2248	668	1065	95	1039			290	1023		
SSAO	-40	300	372	1902	-630	857	401	844			-296	1112		
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-34: *Δabsolute repeatability* regression model for MSER on scene *sport*.

Δabsolute repeatability, MSER, street														
Linear regression model:	AbsRepeatability ~ 1 + CLD + DCD + EHD + SCD*Config													
	Intercept		CSD		CLD		SCD		DCD		HTD		EHD	
Configuration	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE	b	SE
baseline	-27	299			-3239	646	305	1213	62	35			768	209
noise	-144	397					551	1648						
distortion	-434	394					1627	1663						
aperture	224	335					-1328	1478						
HDR	-50	397					196	1650						
bloom	-237	345					936	1468						
blur	-593	409					1933	1715						
SSAA	189	395					-760	1639						
MSAA	-79	396					315	1644						
FXAA	-472	400					1762	1662						
SMAA	-456	402					2211	1669						
AToC	-23	394					77	1635						
objects high	-57	395					215	1640						
no objects	322	343					-1824	1456						
modelling errors	-129	397					571	1651						
texture low	-733	412					2591	1744						
surface low	-392	369					1831	1679						
AF	409	396					-1571	1645						
shadow	65	402					-146	1660						
shadow filter	-30	395					99	1643						
SSAO	-968	397					4572	1647						
Legend:	Not-significant			p < .05			p < .01			p < .001				

Figure C-35: *Δabsolute repeatability* regression model for MSER on scene *street*.