Hai Huang

Extraction of Unfoliaged Trees from Terrestrial Image Sequences

der Bundeswehr
Universität München

Neubiberg 2010

Extraction of Unfoliaged Trees from Terrestrial Image Sequences

Vollständiger Abdruck der von der
Fakultät für Bauingenieur- und Vermessungswesen der
Universität der Bundeswehr München
zur Erlangung des akademischen Grades
eines
Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation

vorgelegt von

Dipl.-Ing. Hai Huang

Neubiberg 2010

Prüfungskommission:

Vorsitzender: Univ.-Prof. Dr.-Ing. Friedrich Kröll

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Helmut Mayer
2. Univ.-Prof. Dr.-Ing. Stefan Hinz (KIT – Karlsruher Institut für Technologie)

Die Dissertation wurde am 10.10.2010 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 10.11.2010 angenommen.

Tag der Promotion: 17.12.2010

Acknowledgments

"$PS_n \rightarrow TT_n \rightarrow EE_n \rightarrow PS_{n+1}$"
– K. R. Popper

I would like to thank my advisor, Prof. Helmut Mayer, for showing me a whole new world of photogrammetry and computer vision. I am grateful for his guidance, advice, and patience. He generously gave his time and was very supportive concerning all issues of research and also daily life.

I thank Prof. Stefan Hinz and Prof. Friedrich Kröll for their instantaneous acceptance to become second supervisor and chairman of the PhD committee, respectively, as well as their support and advice.

Dr. Sergej Reznik, has always been available for research discussions, technical problems, and the cigaret break. Jan Bartelsen, never hesitated to share his ideas and interesting work with me. I will never forget the flight experiments with the Unmanned Aircraft System (UAS). Martin Drauschke, supported me till the last minute of my PhD adventure by organizing the whole promotion celebration.

I say thank you to Prof. Egon Dorrer, Angelika Arnorld, Gisela Pietzner, Jürgen Peipe, Ton Hoang Anh Duc, and all other people at the Institute of Photogrammetry and Cartography for making it such an interesting and pleasant place to work.

Participating in the Sino-German bundle project "Interoperation of 3D Urban Geoinformation" was an exciting experience. Many people have offered me advice and guidance, for which I am especially grateful to: Prof. Claus Brenner, Prof. Jun Chen, Prof. Wolfgang Förstner, Prof. Christian Heipke, Prof. Olaf Hellwich, Prof. Liqiu Meng, Prof. Lutz Plümer, Prof. Monika Sester, Prof. Wenzhong (John) Shi, Prof. Uwe Stilla, and Prof. Zuxun Zhang. With the other PhD students Yi Ding, Anne Grote, Matthias Heinrichs, Ludwig Hoegner, Tao Ke, Birgit Kieler, Holger Kumke, Judith Milde, Jens Saatkamp, Jörg Schmittwilken, Mingwei Sun, and Ying Yang I had a nice time leaving footprints in Beijing, Wuhan, Urumqi, Hong Kong, Berlin, Hannover, Bonn, and Munich.

My final gratitude goes to my wife Na Liu for going with me through a lot more than she anticipated.

This thesis is based upon work supported by German Science Foundation (Deutsche Forschungsgemeinschaft – DFG) by Grants No. MA 1651/11-1 and MA 1651/12-1.

Abstract

This thesis presents a generative statistical approach for the fully automatic three-dimensional (3D) extraction and reconstruction of unfoliaged deciduous trees from wide-baseline image sequences. Tree models improve the realism of 3D Geoinformation systems (GIS) by adding a natural touch. Unfoliaged trees are, however, difficult to reconstruct from images due to partially weak contrast, background clutter, occlusions, and particularly the possibly varying order of branches in images from different viewpoints. The proposed approach combines generative modeling by L-systems and statistical maximum a posteriori (MAP) estimation for the extraction of the 3D branching structure of trees. Background estimation is conducted by means of mathematical (gray scale) morphology as basis for generative modeling. A Gaussian likelihood function based on intensity differences is employed to evaluate the hypotheses. A mechanism has been devised to control the sampling sequence of multiple parameters in the Markov Chain considering their characteristics and the performance in the previous step. A tree is classified into three typical branching types after the extraction of the first level of branches and more specific Production Rules of L-systems are used accordingly. Generic prior distributions for parameters are refined based on already extracted branches in a Bayesian framework and integrated into the MAP estimation. By these means most of the branching structure besides tiny twigs can be reconstructed. Results are presented in the form of VRML (Virtual Reality Modeling Language) models demonstrating the potential of the approach as well as its current shortcomings.

Zusammenfassung

Diese Dissertationsschrift stellt einen generativen statistischen Ansatz für die vollautomatische drei-dimensionale (3D) Extraktion und Rekonstruktion unbelaubter Laubbäume aus Bildsequenzen mit großer Basis vor. Modelle für Bäume verbessern den Realismus von 3D Geoinformationssystemen (GIS), indem sie Letzteren eine natürliche Note geben. Wegen z.T. schwachem Kontrast, Störobjekten im Hintergrund, Verdeckungen und insbesondere der möglicherweise unterschiedlichen Ordnung der Äste in Bildern von verschiedenen Blickpunkten sind unbelaubte Bäume aber schwierig zu rekonstruieren. Der vorliegende Ansatz kombiniert generative Modellierung mittels L-Systemen und statistische Maximum A Posteriori (MAP) Schätzung für die Extraktion der 3D Verzweigungsstruktur von Bäumen. Hintergrund-Schätzung wird auf Grundlage von mathematischer (Grauwert) Morphologie als Basis für die generative Modellierung durchgeführt. Für die Bewertung der Hypothesen wird eine Gaußsche Likelihood-Funktion basierend auf Intensitätsunterschieden benutzt. Es wurde ein Mechanismus entworfen, der die Reihenfolge der Verwendung mehrerer Parameter für die Markoff-Kette basierend auf deren Charakteristik und Performance im letzten Schritt kontrolliert. Ein Baum wird nach der Extraktion der ersten Stufe von Ästen in drei typische Verzweigungstypen klassifiziert und es werden entsprechend Produktionsregeln von spezifischen L-Systemen verwendet. Basierend auf bereits extrahierten Ästen werden generische Prior-Verteilungen für die Parameter in einem Bayes'schen Rahmen verfeinert und in die MAP Schätzung integriert. Damit kann ein großer Teil der Verzweigungsstruktur außer kleinen Ästen extrahiert werden. Die Ergebnisse werden als VRML (Virtual Reality Modeling Language) Modelle dargestellt. Sie zeigen das Potenzial aber auch die noch vorhandenen Defizite des Ansatzes.

Contents

Chapter 1

Introduction

The basic motivation of the work described in this thesis is to add a natural touch to three-dimensional (3D) city models by the extraction and reconstruction of trees and thus to improve the realism of 3D Geoinformation systems (GIS).

In our environment trees play an essential role. Particularly in urban areas, they are too often the only representatives of nature. Because of their complex structure and costly acquisition, trees are often neglected, or at least only acquired in a very simplified form for GIS, especially for 3D city models. The distinctive shape and texture of individual trees that can strongly influence the appearance of their whole environment is only represented for very locally limited architectural models. Detailed 3D representations of individual trees can substantially enhance the realism of city models. Additionally, they can be used for applications such as architectural planning or the production of movies.

In this thesis we aim at the extraction of the 3D branching structure of individual unfoliaged deciduous trees from terrestrial wide-baseline image sequences. Deciduous trees are popular in most cities worldwide (in the Temperate Zone) as they provide shadow in summer while letting the sunlight through in winter. Thus, they form the majority of trees in urban areas. From a practical point of view, terrestrial images for the data acquisition for city models are often acquired when trees are unfoliaged, as facades, etc. are more readily visible then. From a scientific, but also from a practical point of view, unfoliaged trees have the advantage, that they explicitly show the branches. Extracting the 3D branching structure from images automatically is a challenging task, which to our knowledge nobody has tried to do without (substantial) manual intervention.

Images are attractive as source data for object extraction because of the low cost for data acquisition and the comprehensive information on color and texture they contain. However, to construct 3D models of trees bottom-up/data-driven from wide-baseline image sequences, the branches have to be matched. This is difficult due to the geometric complexity and mutual occlusions of branches along with weak contrast and background clutter in the images. Additionally, as shown in Figure 1.1, the order of branches can vary from different points of view due to the pronounced 3D structure of trees. This means that the ordering constraint, i.e., a point left of another point on an epipolar line in one image is also left of the corre-

sponding point on the epipolar line in the other image, often employed to guide matching, is often not valid for branches even for images taken close to each other.

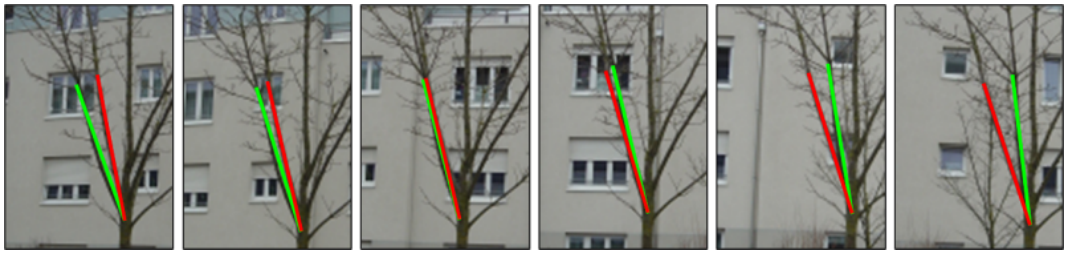

Figure 1.1: Varying order of branches from different points of view. Two different branches are marked in green and red.

In this thesis we show how generative statistical modeling makes it feasible to extract and reconstruct 3D branching structures from wide-baseline image sequences in spite of the problems stated above. Figure 1.2 shows the basic concept of the approach: trees are generatively modeled by means of L-systems from computer graphics and the parameters of tree models are determined statistically by maximum a posteriori (MAP) estimation. In this framework for tree extraction, Lindenmayer-, or in short L-systems (Prusinkiewicz and Lindenmayer 1990) bring biological information into the modeling while the statistical sampling builds the link to the images of the real scene.

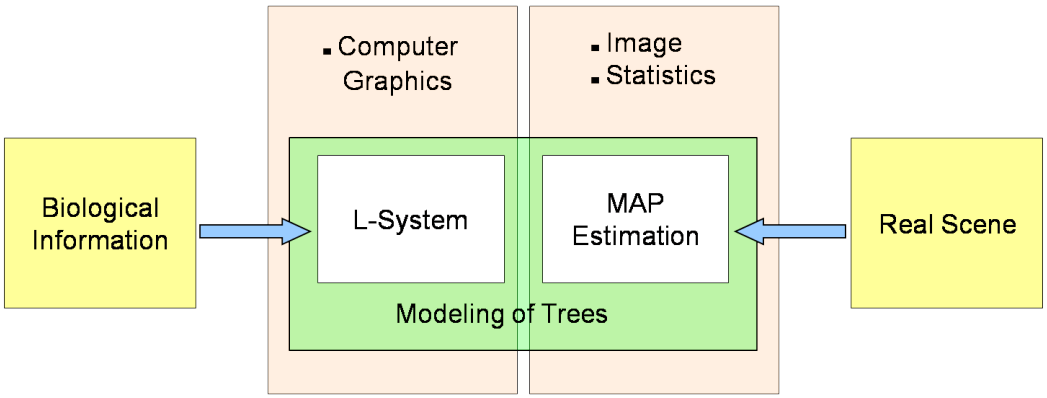

Figure 1.2: Generative statistical modeling: L-systems are used to model trees in a general way based on botanical knowledge – MAP estimation is employed to find their individual characteristics in real scenes.

Our main contributions can be summarized as follows:

1. L-systems are devised for modeling trees in images conditioned on different branching types.

2. Background estimation by means of gray scale morphology and detail recovery is conducted as basis for generative modeling.

3. MAP estimation is employed for the optimization of parameters using a Gaussian likelihood function based on intensity differences.

4. A Bayesian framework has been devised to refine the generic prior distributions for the parameters based on the extracted branches. The improved priors are integrated into the MAP estimation.

5. A proposal mechanism has been designed to guide the sampling sequence. Considering the characteristics of individual parameters and their mutual influence, the sampling becomes more reasonable and efficient.

6. Trees are classified into different branching types and the corresponding L-system is applied for a more plausible description.

By these means, main branches of trees can be extracted and reconstructed in 3D from image sequences in the form of Virtual Reality Modeling Language (VRML 2.0, ISO/IEC 14772-1:1997) models, as shown in Figure 1.3.

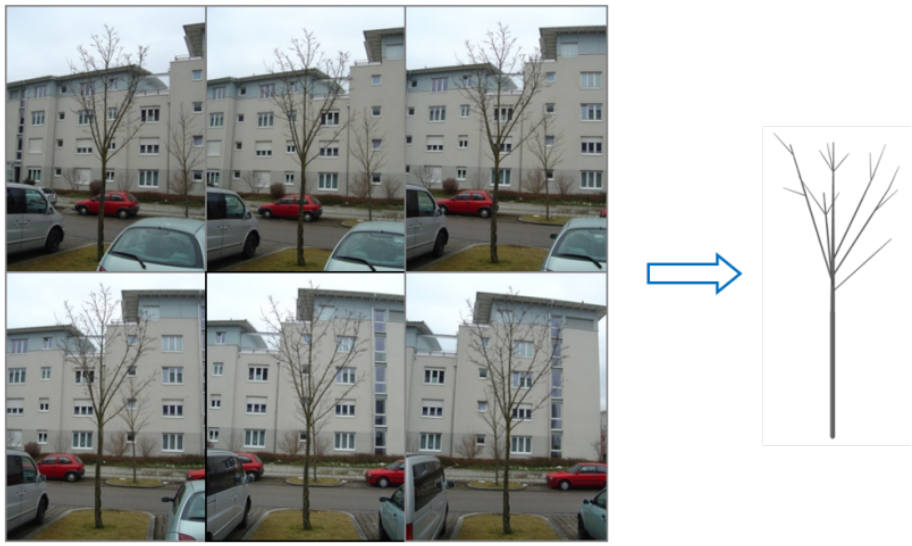

Figure 1.3: 3D tree extraction and reconstruction in the form of a VRML model (right) from an image sequence (left).

The thesis is organized as follows: Chapter 2 sets the methodological background. Selected previous approaches on tree extraction from various sources are summarized in Chapter 3. Chapter 4 describes the generative modeling of trees by means of L-systems along with the definition of different branching types. The extraction of the branching structure is presented in Chapter 5 comprising the generation and evaluation of 3D hypotheses, MAP estimation, the classification of branching types, and the Bayesian refinement of priors. Results of experiments on a simulated tree model and trees in real scenes are presented and discussed in Chapter 6. Finally, Chapter 7 summarizes the thesis and gives an outlook.

Chapter 2

Methodological Background

This chapter reviews some of the concepts and techniques employed in the presented approach, including modeling methods, statistical algorithms, and the object extraction strategy.

Markov Chain Monte Carlo (MCMC) is used as basic statistical sampling method. We show properties of Markov Chains and advantages of MCMC sampling for scientific computation in Section 2.1 introducing and discussing some MCMC algorithms of interest. In Section 2.2 we present the statistical framework for learning and optimization with Bayesian inference at the core. Maximum a posteriori (MAP) estimation, which also works based on Bayesian inference, is also included in this Section.

We introduce generative modeling and the concept of generative statistical object extraction from images in Section 2.3 followed by gray-scale morphology in Section 2.4. Section 2.5 focuses on L-systems, which can be used to model the structure of trees by simulating their growth. In the same section state-of-the-art approaches of tree modeling including one commercial software we have used to produce a sample tree are presented.

Finally, the geometric basis of the approach – a highly precise structure from motion procedure – is briefly described in Section 2.6.

2.1 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a class of statistical sampling algorithms widely applied as a general purpose computing technique for probability model simulation, integration, and optimization. It has been introduced in physics in the late 1940's and nowadays plays an important role in statistics, computer science, and econometrics. The main advantage of MCMC techniques is that they can sample efficiently in large spaces with high dimensionality.

2.1.1 Monte Carlo Sampling

Monte Carlo sampling is a basic numerical computing technique. In many practical (statistical) applications it is hard to make exact inferences for the proposed probabilistic models. Approximation by means of numerical sampling, e.g., Monte Carlo sampling, thus becomes very important.

Monte Carlo sampling is named after the town Monte Carlo, which is world famous because of the luxurious casino. The essential idea of Monte Carlo sampling is therefore random sampling, i.e., drawing dices. It can be traced to a rudimentary version invented by Fermi in the 1930s, even earlier than the first computer appeared (Robert and Casella 2008). Its first well-known practical application was during building the first nuclear reactor in 1942.

Monte Carlo simulation draws a set of samples $x_i, i = 1, ..., n$, from a target distribution $p(x)$ in a space Ω. They are independent and identically-distributed (iid) random variables.

The n samples can be used to approximate the target density of a function $f(x)$

$$E\{f(x)\} = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \xrightarrow[n\to\infty]{} \int_{\Omega} f(x)p(x)dx\,, \tag{2.1}$$

where $E\{f(x)\}$ is an unbiased estimate of $f(x)$. According to the law of large numbers, E will surely converge to the integral of $f(x)$.

Theoretically, if the sampling is dense enough, the Monte Carlo samples can also be directly used for optimization:

$$\widehat{x} = \arg_{\Omega'} \max p(x_i) \xrightarrow[n\to\infty]{} \arg_{\Omega} \max p(x)\,, \tag{2.2}$$

where Ω' is the sample space, $\Omega' \in \Omega$.

In practice, however, one often encounters complicated and/or combined distributions with high-dimensional sampling spaces, and then the inefficiency of Monte Carlo can be fatal. More sophisticated techniques which can guide the sampling routine, e.g., MCMC, are needed.

2.1.2 Markov Chains

A Markov Chain, as shown in Figure 2.1, is a mathematical model for stochastic systems whose states, discrete or continuous, are governed by transition probabilities. It consists of a series of random variables, named states $X_0, X_1, ..., X_n$ and can be denoted by

$$(\Omega, P(X_0), K)\,,$$

with

Ω: the state space;

$P(X_0)$: the initial probability – the marginal distribution for X_0, which specifies the starting state;

K: the kernel – the transition probability from the previous state(s) to the current state.

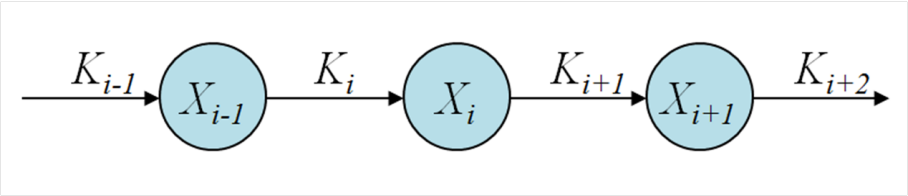

Figure 2.1: A first order Markov Chain.

A Markov process starts from the initial state X_0 and moves successively to the next state controlled by the transition probability. Each move is usually called a "step", as the Markov process is considered to walk in the state space Ω. Giving equal probability to every potential step, the Markov process represents a random walk.

The main property of the Markov process is that the current state only depends on the most recent previous states. The typical Markov Chain, namely the first order Markov Chain, is shown in Figure 2.2 and can be formulated as:

$$X_t|X_{i-1}, ..., X_0 \sim P(X_i|X_{i-1}, ..., X_0) = P(X_i|X_{i-1}) \,, \tag{2.3}$$

in which only the last state influences the current one.

The kernel K is defined as the conditional probability for subsequent variables, i.e., the probability of X_i given X_{i-1}:

$$K_i(X_i, X_{i-1}) \equiv P(X_i|X_{i-1}) \ .$$

The simplest form of a Markov Chain is the homogeneous (or stationary) one, in which the transition probability is constant for all states (independent on time or position $i \in \{0, n\}$):

$$P(X_i|X_{i-1}) = const. \tag{2.4}$$

For Markov Chains in discrete state spaces, we denote the transition probability by $T = T(X_i, X_{i-1})$. The probability of state X_i as derived from that of the previous state is given as:

$$P(X_i) = \sum_{X_{i-1}} P(X_{i-1})T(X_i, X_{i-1}) \,. \tag{2.5}$$

For continuous state spaces, the representation of a Markov Chain can be expressed using the integral kernel K:

$$P(X_i) = \int P(X_{i-1})K_i(X_i, X_{i-1})dX_{i-1} \ . \tag{2.6}$$

Stability of Markov Chains

Stability means that starting from any initial state, a Markov Chain will convergence to an invariant distribution, i.e., a stationary probability. This is a very important and attractive feature of Markov Chains.

In practical applications, the target distribution $p(x)$ often cannot be directly observed. The goal of designing Markov Chains is then to devise a stationary probability that can represent $p(x)$. I.e., the samples x_i from Markov Chain states X_i simulate samples drawn from the target distribution.

The invariant distribution should be reached by

$$P(X_0)K^n \xrightarrow[n\to\infty]{} P(X) \ , \tag{2.7}$$

where n indicates the number of times that K has been multiplied. Stability means that after a limited number of transitions (steps), any initial distribution will converge to a stationary distribution. This implies that the stability of Markov Chains depends on the transition probability K. To ensure stability, K must be constructed guaranteeing the following two properties:

1. Ergodicity: A Markov Chain is called ergodic if it is possible for the chain to explore the whole state space. I.e., the probability of visiting all other states is always positive. Ergodic Markov Chains are often also called "irreducible".

2. Aperiodicity: An ergodic chain is called aperiodic (or acyclic) if there does not exist a periodic structure in the chain.

The purpose of employing Markov Chains is to guide the "routine" of sampling more efficiently, i.e., conducting a walk mostly in interesting regions, and to converge as soon as possible.

2.1.3 MCMC Techniques

Markov Chain Monte Carlo is a class of techniques for generating fair samples x_i for the states X_i in the state space Ω using a Markov Chain guiding the sampling. The stability of a Markov Chain, as mentioned above, is advantageous to ensure convergence. A typical MCMC is illustrated in Figure 2.2.

The current state X_i is determined by the transition kernel K and in first order Markov Chains additionally only by the last state X_{i-1} ($X \sim P(x)$). The variable x_i is drawn stochastically ($x \sim U[a, b]$, iid) for each state X_i.

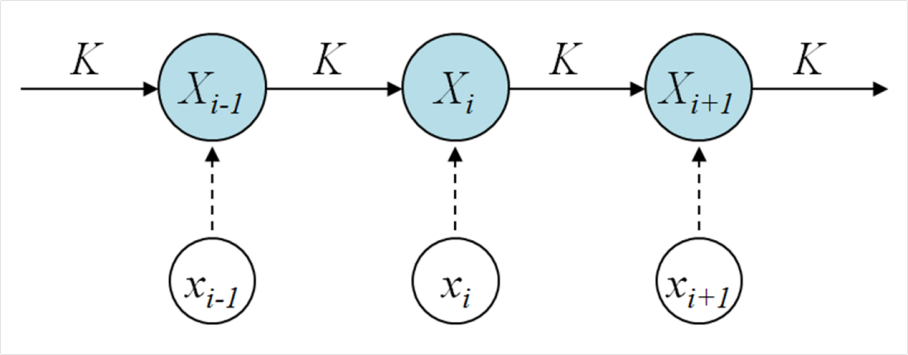

Figure 2.2: MCMC with first order Markov Chain.

Detailed Balance

MCMC samplers are Monte Carlo samplers which ideally employ ergodic and aperiodic Markov Chains. The invariant distribution derived from the stationary chains can be used to approximate the target distribution $p(x)$. One way to construct an appropriate sampler is to satisfy the reversibility, i.e., detailed balance, condition:

$$p(X_i)T(X_{i-1}, X_i) = p(X_{i-1})T(X_i, X_{i-1}) \,, \tag{2.8}$$

which is a sufficient, but not a necessary condition to ensure that the invariant distribution of the Markov Chain is identical to the target distribution $p(x)$.

Metropolis-Hastings Algorithm

The Metropolis-Hastings (MH) algorithm was introduced in (Metropolis et al. 1953) and has been further developed by Hastings (1970). MH is one of the most popular MCMC techniques. It is a generalization of the basic Metropolis algorithm and many practical MCMC algorithms can be seen as special cases or extensions of the MH algorithm.

In basic MCMC, a new step X_{i+1} proposed by the Markov Chain will only be accepted when its probability is higher than that for its predecessor. The MH algorithm relaxes this criterion by accepting "worse" candidates, conditioned on an acceptance probability $A(X_i, X^*)$:

$$A(X_i, X^*) = \min\left\{1, \frac{p(X_i)q(X^*|X_i)}{p(X^*)q(X_i|X^*)}\right\} . \tag{2.9}$$

Figure 2.3 shows the basic MH algorithm. The Markov Chain moves forward to X^* as the next state X_{i+1} if $A(X_i, X^*) >$ random number u, otherwise it remains at X_i.

The acceptance probability guarantees the following two properties:

1. In case the new candidate is better than the current, i.e., the second term is larger than 1, $A(X, X^*) = 1$. Since u is sampled in $[0, 1]$, a better candidate will always be accepted.

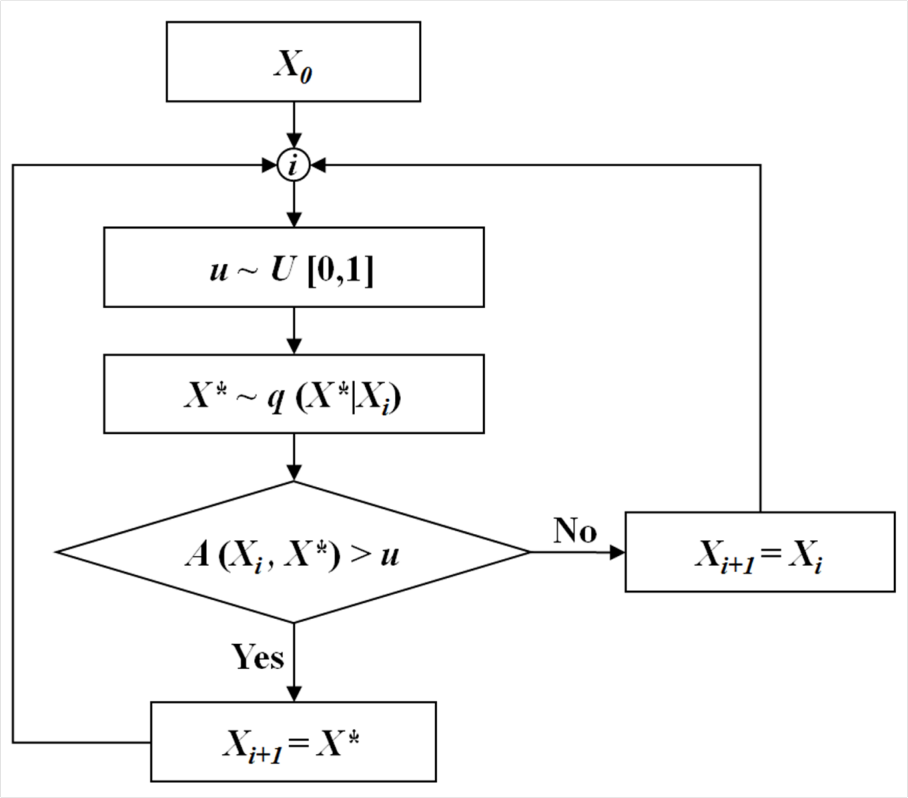

Figure 2.3: Metropolis-Hastings algorithm.

2. In case the new candidate is evaluated worse, instead of being rejected immediately, it still has a chance to be accepted proportional to its evaluation based on the acceptance probability.

The latter property gives the MH algorithm the potential to overcome local minima, which often occur in complex distributions. Figure 2.4 shows a function with multiple local minima. Instead of converging to the next local minimum, the sampler has the possibility to go "up-hill" by conditionally accepting worse candidates and thus to reach the global minimum.

The transition kernel in MH is based on the ratio of the proposed and the current step, implying the gradient of the function. Small hills can thus be easily overcome because a low gradient means a large acceptance probability.

An important feature of the MH algorithm is, that the target density $p(X)$ needs only to be known up to a multiplicative factor, since only the ratio $p(X_i)/p(X^*)$ has to be computed. This makes this sampling algorithm very attractive for Bayesian computation, where the posterior distribution is often only known up to a normalization factor.

Applications of MCMC Sampling

MCMC is used as a general purpose (computing) technique, e.g., in the following areas:

1. Simulation: MCMC produces samples of the probability model underlying the target system and represents typical states of the latter.

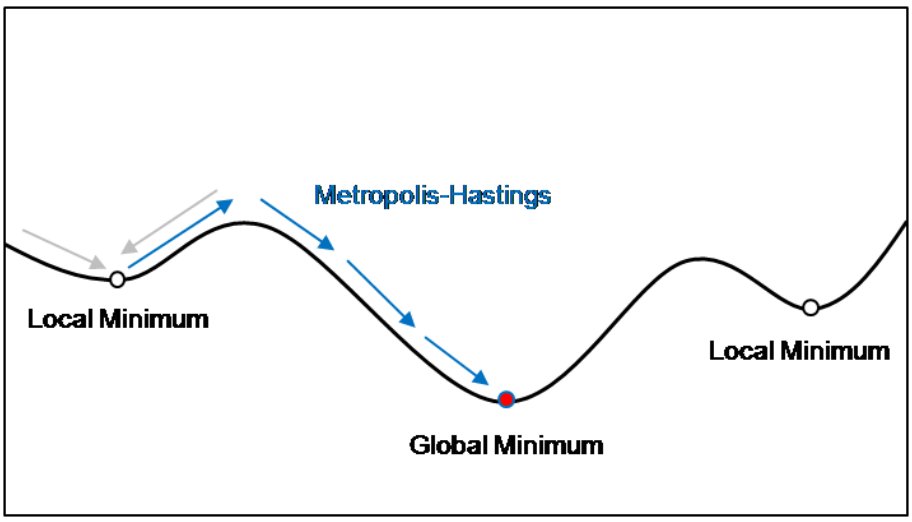

Figure 2.4: Metropolis-Hastings algorithm renders it feasible to overcome local minima by allowing "worse" hypotheses.

Many natural processes are inherently stochastic, but still follow essential rules, which can be expressed in the form of probability models. For example, the random growth of tree models can be governed by L-systems (cf. Section 2.5.1). The fair samples generated by MCMC will vary in appearance, but still present typical tree structure as described by L-systems. Another example is the generation of Gaussian noise with given mean and variance in image processing.

2. Integration and Computing: The computation of integrals is a typical task in scientific computing. Monte Carlo integration is used to deal with distributions in very high dimensional spaces. The expectation c is estimated as follows:

$$c = \int_{\Omega} p(x) f(x) dx \;\; ; \qquad \widehat{c} = \frac{1}{n} \sum_{i=1}^{n} f(x_i) \, , \tag{2.10}$$

where Ω is the integral space, $\widehat{c}$ the approximation with Monte Carlo, and n the number of samples drawn from $p(x)$.

3. Bayesian Inference and Learning: MCMC is an important computing tool for maximum likelihood estimation (MLE) learning of parameters $p(x;\theta)$ as well as maximum a posteriori (MAP) estimation (cf. Section 2.2.2). Unsupervised learning with hidden variables (simulated from the posterior) needs simulation to find suitable models. As Bayesian inference has become an important framework in many areas, e.g., image understanding and computer vision, MCMC techniques are of increasing interest.

4. Optimization: MCMC is the most often used technique for searching the global optima of complex distributions, e.g., the Bayesian posterior probability (MAP).

MCMC is a general technique for many problems with a high-dimensional solution space. In some cases it is even the only known approach to provide solutions within an acceptable computation time.

Reversible Jump Markov Chain Monte Carlo

Reversible Jump Markov Chain Monte Carlo (RJMCMC) is an extension of the MCMC algorithm to handle state spaces of variable dimension. It is a variant of the Metropolis-Hastings algorithm and can deal with dimension jumps, i.e., allows modeling with varying numbers of parameters or/and objects. As proposed in (Green 1995), the Markov Chain sampler can extend its search by switching between spaces with variable dimensions. These movements are called "reversible jumps" as any "jump" must be reversible, i.e., it must be possible to return to the previous state.

RJMCMC has a mixed transition kernel, as demonstrated in Figure 2.5. A set of moves – birth, death, split, and merge of components – is defined and chosen according to mixture probabilities.

1. Initialisation: set $(k^{(0)}, \mu^{(0)})$.
2. For i = 0 to N-1
   -- Sample $u \sim U[0,1]$
   -- If $(u \leq b_k)$
      - then "birth" move.
      - else if $(u \leq b_k + d_k)$ then "death" move.
      - else if $(u \leq b_k + d_k + s_k)$ then "split" move.
      - else if $(u \leq b_k + d_k + s_k + m_k)$ then "merge" move.
      - else update

      End if.
   -- Sample other parameters.

Figure 2.5: Reversible jump consists of a mixture of MCMC kernels (Andrieu et al. 2003). Assume that we are concerned with sampling the locations μ and the number k of components of a mixture model. The various moves are carried out according to the mixture probabilities (b_k, d_k, s_k, m_k, u). The superscript "(0)" indicates the initial state.

RJMCMC allows for a large, even infinite, number of candidate models, as long as the cross-model kernels keep the balance condition. In this framework a Bayesian process can be integrated to generate a wide variety of new models, which are inferred based on old ones and can be more promising. Thus, RJMCMC is also a powerful model selection scheme.

Unfortunately, there is also a number of problems with RJMCMC: 1) The design of the transition kernels is more complicated. 2) The extension of the search space makes RJMCMC usually time-consuming. The latter is the reason why RJ is not integrated into our search scheme, although it is attractive for possible future work (cf. Chapter 7).

2.2 Bayesian Inference and MAP Estimation

Classical probability theory is based on the study of independent trials. This means that knowledge of previous trials in one area has no influence on the prediction for a current trial in another, but similar area. Contrary to this, modern probability theory studying chance processes assumes that investigating previous observations of a similar model will very likely help us to derive better predictions for further experiments.

A basic assumption is since Helmholtz (1860) that biologic (and also machine) vision computes the most probable interpretation(s) from input images. Let I be an image and X be a semantic representation of the world. Then, with the conditional probability $\pi(X|I)$ holds:

$$X^* = argmax\Big\{\pi(X|I)\Big\} \tag{2.11}$$

In numerical statistics, usually the posterior is sampled and multiple solutions are kept:

$$(X_1, X_2, ..., X_k) \sim \pi(X|I) \tag{2.12}$$

When studying a physical process, one often has only the observation data rather than the underlying distributions. Statistical inference is used to estimate the adequate model along with the associated parameters based on the observed data. Bayesian inference is a means for statistical inference using Bayes' theorem. Observations, also called evidence, are used to infer the probability that a hypothesis may be true.

As our knowledge about the underlying model and its parameters is "updated" along with the accumulation of more and more evidence, this can be also seen as a learning process about the statistical characteristics of the parameters from the observations.

2.2.1 Bayesian Inference and Learning

Assume that we want to estimate the parameter θ based on observations $x = \{x_1, x_2, ..., x_n\}$. Bayes' theorem shows how to update the probability, or in other words to infer the posterior, with given evidence as follows:

$$P(\theta|x) = \frac{f(x|\theta)p(\theta)}{P(x)}\,, \tag{2.13}$$

where

– $p(\theta)$ is the prior of the parameter θ;

– $f(x|\theta)$ is the likelihood function (the conditional probability of observing evidence x given the parameter θ);

– $P(\theta|x)$ is the posterior (the conditional probability of a hypothesis given the observed evidence).

The denominator

$$P(x) = \int_{\Omega} f(x|\theta')p(\theta')d\theta' \tag{2.14}$$

is the marginal probability of x, i.e., the probability of observing x under all possible θ. This integral can also be seen as a normalization term to make sure that the posterior integrates to unity. Since $P(x) \geqq P(x \cap \theta) = f(x|\theta)p(\theta)$, the updated posterior will never become greater than 1. And more important, as it does not depend on θ, this term can be treated as constant when optimizing the latter.

The term

$$\frac{f(x|\theta)}{P(x)}$$

summarizes the influence of the evidence on the belief in the hypothesis. It shows that better evidence, i.e., evidence that supports the proposed parameter, leads to a higher posterior probability for the hypothesis.

While evidence accumulates, there are two kinds of learning processes based on Bayesian inference:

- Discriminative learning models the posterior $P(\theta|x)$: The degree of belief in a hypothesis tends to become either very high or very low making a discriminative learning for a decision (accept/reject) or a classification possible.
- Generative learning models the joint probability $P(x, \theta)$: Knowledge, i.e., priors, about the proposed parameter(s) θ of the model is improved to the conditional probability $P(\theta|x)$ by integrating verified evidence x (cf. Section 5.7).

Taking the classification task as example, generative learning creates explicit models, which represent the training data of the object category. A generative classifier learns the prior and the likelihood of the classes γ $P(\gamma)$ and $P(x|\gamma)$ and classifies x by maximizing $P(\gamma|x) \propto P(x|\gamma)P(\gamma)$. A discriminative approach, in contrast, learns the posterior $P(\gamma|x)$ directly. It finds the best classifier for the given data set focusing on discriminative characteristics between individual categories. Current approaches often combine these two concepts to deal with the classification problems with relatively similar categories. I.e., they learn object categories generatively and train models for each category discriminatively.

The predominant feature of Bayesian inference is that the predictions for the parameters are in the form of probability distributions (posteriors) instead of point predictions for conventional approaches. Usually, the posterior has a low entropy and the effective volume of the search space is relatively small.

Another advantage is the potential for automatic model selection (Bishop 2008): The additional uncertainty of a set of candidate models with different complexity can be addressed in a Bayesian framework. A prior is given for the different models and the posterior for the models based on the observation X can be expressed as:

$$p(M|X) \propto p(M)p(X|M) \,. \tag{2.15}$$

The model which best balances complexity and goodness of fit is determined based on optimizing the "marginal likelihood", also called "model evidence", presenting the preference of the evidence for the candidate model (Bishop 2008).

2.2.2 Maximum A Posteriori Estimation

Maximum a posteriori (MAP) estimation optimizes the posterior based on Bayesian inference. In comparison with the often used maximum likelihood (ML) estimation, MAP estimation does not only employ the likelihood as the optimization objective, but also integrates the prior of the estimated parameters.

Let X be the observations and Θ the space of parameters. Then, the likelihood function can be expressed as

$$\Theta \mapsto L(\mathcal{D}) = L(X|\Theta) \,. \tag{2.16}$$

The maximum likelihood estimate of Θ is

$$\widehat{\Theta}_{ML} = arg_{\Theta}max\Big\{L(X|\Theta)\Big\} \,. \tag{2.17}$$

Based on Bayesian inference, the posterior distribution of Θ can be written as:

$$\Theta \mapsto P(\Theta|X) = \frac{L(X|\Theta)p(\Theta)}{\int_{\Omega} L(X|\Theta')p(\Theta')d\Theta'} \tag{2.18}$$

with p the prior and Ω the domain of Θ.

The denominator of the posterior does not depend on Θ. Therefore, it can be seen as a constant in the optimization. The maximum estimate for the posterior is thus equivalent to that of the numerator:

$$\widehat{\Theta}_{MAP} = arg_{\Theta}max\left\{\frac{L(X|\Theta)p(\Theta)}{\int_{\Omega} L(X|\Theta')p(\Theta')d\Theta'}\right\} = arg_{\Theta}max\Big\{L(X|\Theta)p(\Theta)\Big\} \,. \tag{2.19}$$

MAP estimation can be seen as an extension or a regularization of ML estimation: the empirical knowledge about the parameters is considered as well by integrating the prior information of them. Using the posterior also lowers the entropy of the target function to be optimized and thus the search space is narrowed down to a smaller, often more reasonable region.

Because of the augmented optimization objective, computing of the MAP estimate has often to deal with more complicated (very likely non-analytic) distributions in a much larger search space. A general numerical solution for the computation of MAP estimates is to use MCMC techniques (cf Section 2.1.3).

2.3 Generative Statistical Object Extraction

Conventional object extraction focuses on bottom-up, feature-based methods. These data-driven methods, however, can encounter big difficulties when the extraction is strongly uncertain. This is particularly true if images are used as, possibly the only, data source, and problems with weak contrast, background clutter, occlusions, etc., have to be tackled. Additionally, for complex 3D structures like branching systems, invalid conventional constraints for matching make feature-based methods even less suitable.

As a top-down model-driven method, generative modeling has become a tendency of recent research to deal with the uncertainty of data and complex target structures. The basic idea of generative statistical modeling is to generate multiple hypothetic models by statistically varying the parameters of the model and evaluating the proposed hypotheses by comparing them with the observed (image) data, possibly after projecting them into the image space. The goal is to find the best combination of parameters, i.e., the best estimate of the underlying model, to describe the target physical process or object.

2.3.1 Generative Statistical Modeling and Extraction from Images

Generative models in a statistical framework are a class of models employing full probability information for all parameters. Suppose we want to study the unobservable parameter θ based on the observation X. Generative modeling means to generate samples from the joint probability distribution $P(\theta, X)$ over both observation data and target (unobserved) parameters.

For a better understanding of generative modeling, we compare it with discriminative modeling. In the latter, only models of the target parameter(s) conditioned on the observed parameters are available. I.e., a discriminative model only allows sampling from $p(\theta|X)$ the probability of one or a very limited number of target parameters conditioned on the observed quantities X with known values of other parameters. Opposed to this, in generative modeling the probability for all parameters $p(\theta)$ in the model is obtained. Generative models produce the observable population, also called simulated data, randomly/statistically by sampling all parameters θ. While discriminative modeling shows better performance in classification and

regression, for which the joint distribution is not required, generative modeling provides more flexibility for complex learning tasks, for which the goal is often to tune θ and maximize the data likelihood (for ML) or the posterior for the parameters conditioned on the data (for MAP).

In many cases, the space of parameters is vast and meaningful solutions may be distributed very sparsely. Usually, the more flexible a model is used, the higher dimensional a solution space must be searched. Furthermore, besides the large sampling space, the distributions are often complex with possibly many local extrema. MCMC techniques, as introduced in Section 2.1, are therefore in many cases used to find solutions.

For generative statistical object extraction in images, the target object is modeled with geometrical and physical features as parameters. Probability distributions are derived for these parameters based on their characteristics known from empirical knowledge. By sampling the parameter distributions, a large number of hypothetic models is generated. The best evaluated hypothesis, usually in the form of an ML or a MAP estimate, describing the target object along with its parameter values, is the result of the extraction.

When using images, the evaluation of 3D hypotheses must be conducted in 2D space, as the images are the observed data. The models are projected into the geometry of the given images resulting in simulated images which are compared with the given images resulting in the likelihood for the evaluation. For a correct projection and thus a reliable evaluation, highly precise 3D orientation should be used (cf Section 2.6).

The procedure of image-based generative statistical 3D object extraction can be summarized as follows (Mayer et al. 2008):

1. A model is devised with a possibly large number of parameters, which may include the number of components.
2. By sampling the parameters statistically, a 3D hypothesis is generated.
3. The 3D hypothesis is projected into image space using as good as possible known orientation parameters, resulting in 2D simulated images.
4. The hypothesis is evaluated by comparing its simulated images to the corresponding given images.
5. The solution space will be sampled, to find the best set of parameters describing the underlying model.

2.3.2 Generative Statistical Approaches

In this section, several approaches for generative statistical object extraction are presented. Although these approaches are not concerned with tree extraction, approaches for which are given in Chapter 3, our work is inspired by them.

(Dick et al. 2004) describes the automatic 3D reconstruction of architectural models from image sequences (cf. Figure 2.6). Buildings, especially their facades, are interpreted by stochastically varying the given models and projecting them into images. The results shows the potential of this approach although it is restricted to a small number of objects (facades, windows, columns, etc.) and it has been tested with a very limited number of examples.

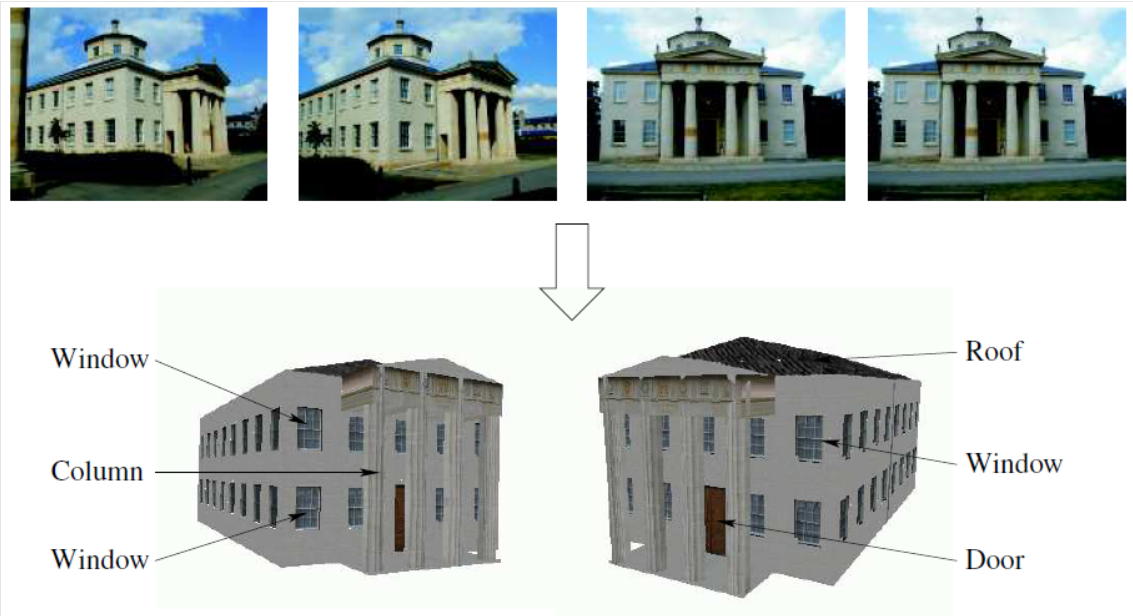

Figure 2.6: Reconstruction of architectural models from image sequences (Dick et al. 2004).

Stoica et al. (2004) employ a framework based on a Gibbs point process. A data term describing roads in images in the form of statistical hypothesis tests is linked to a prior stating that roads form a connected network of lines, favoring aligned segments and penalizing superposition. An estimate for the road network is found by minimizing an energy function employing RJMCMC (cf. Section 2.1.3). As for Dick et al. (2004) above, an interesting feature of the approach is, that one can sample from the prior distribution. If this results in road-like structures, one knows that the structures described by the model are actually road-like. Results for SPOT and aerial optical images as well as ERS radar data show the feasibility but also the shortcomings of the approach.

(Alegre and Dallaert 2004) presents a probabilistic approach for the semantic interpretation of building facades. It combines low-level segmentation and high-level hierarchical classification of structural elements generated by vertical splits and horizontal divisions (cf. Figure 2.7).

(Brenner and Ripperda 2006) and (Ripperda and Brenner 2007) focus on stochastical grammars for the organization of facade elements. They recursively split facades in the form of a derivation tree (cf. Figure 2.8). They employ RJMCMC (cf. Section 2.1.3) for the control of the stochastical search process. Besides the images, Ripperda and Brenner (2007) use also depth (terrestrial laser) data for facade reconstruction.

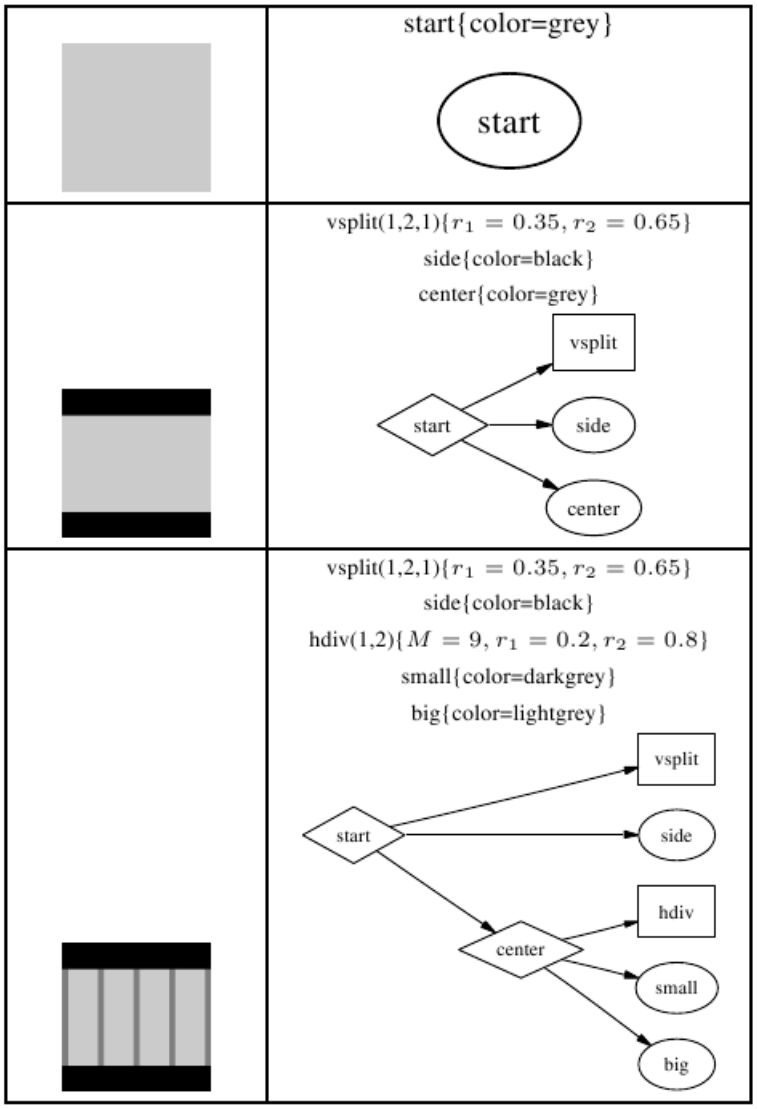

Figure 2.7: Hierarchical partitioning of a facade (ALEGRE and DALLAERT 2004).

MÜLLER et al. (2006) introduce a shape grammar, which is named CGA shape, for the procedural modeling of buildings with high visual quality and much geometric detail (cf. Figure 2.9). CGA shape improves the split grammar with the repeat split rule, the scaling of rules, and the component split. Although it is inspired by L-systems (cf. Section 2.5.1), CGA shape works as a "sequential grammar", i.e., it simulates the spatial variation of features and components instead of the growth over time. Furthermore, this approach has been proven to be efficient by successfully generating massive urban models. More recent work of VAN GOOL et al. (2007) also uses an architecture-oriented shape grammar focusing on 3D facade reconstruction from single images based on prior knowledge of architectural structures.

MAYER and REZNIK (2007) employ terrestrial image sequences to automatically generate facade planes oriented vertically by determining the vertical vanishing point. They use information in patches around corners in the rectified facade images to delineate windows (cf.

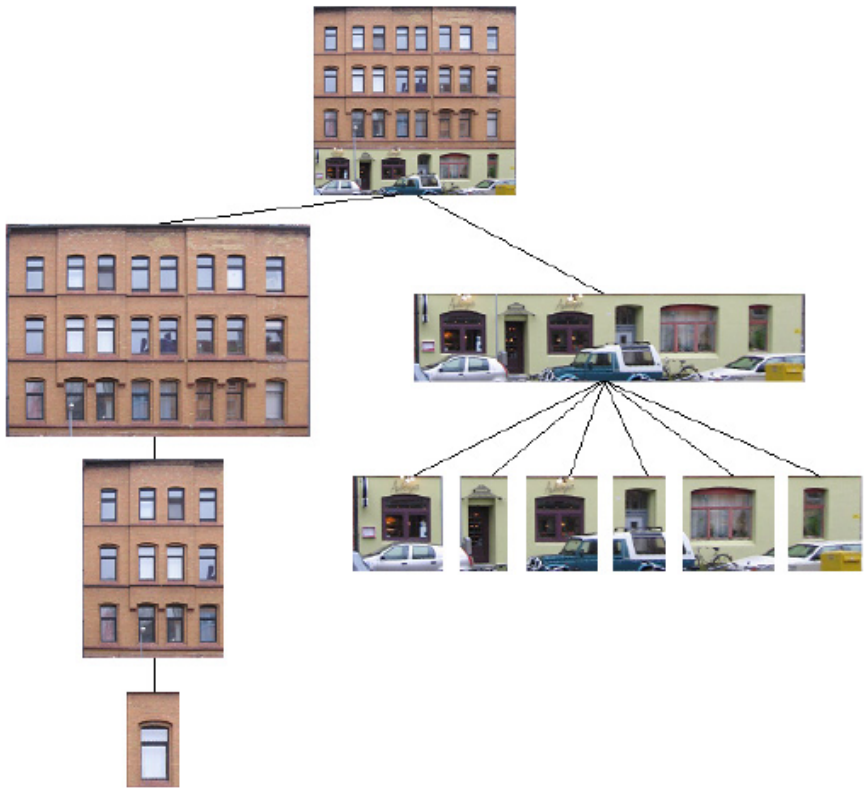

Figure 2.8: The facade is split in the form of a derivation tree (Brenner and Ripperda 2006).

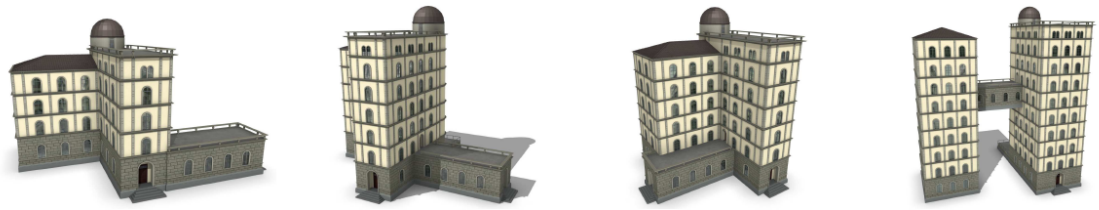

Figure 2.9: Different models are generated by statistical sampling based on a shape grammar (Müller et al. 2006).

Figure 2.10) in an appearance-based way using an Implicit Shape Model – ISM (Leibe and Schiele 2004). A generative approach simulating windows by black rectangles is employed for bright facades.

2.4 Gray-Scale Morphology

Mathematical Morphology was developed in 1964 for analyzing mineral characteristics from thin cross sections and was later generalized to gray-scale functions and images (Serra 1983).

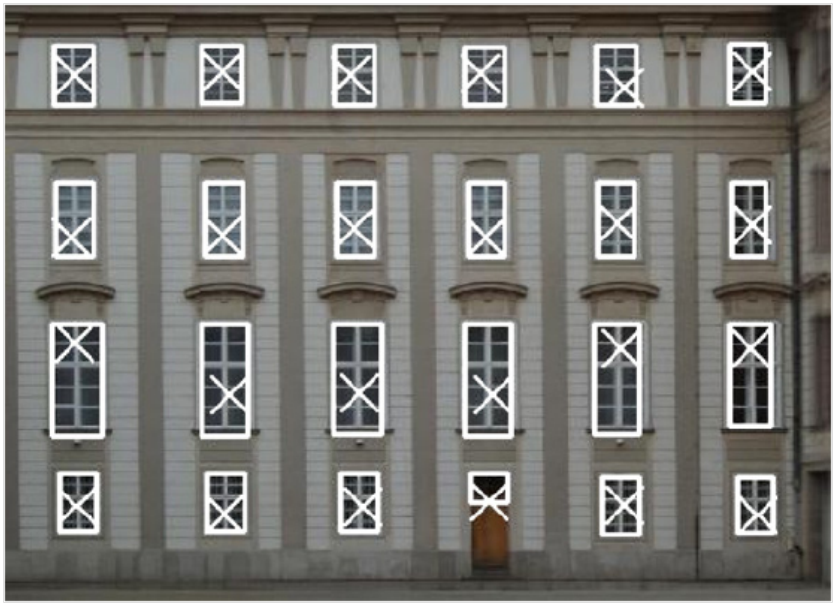

Figure 2.10: Facade interpretation (Mayer and Reznik 2007) – hypotheses of window centers (crosses) and extracted windows (boxes) after MCMC search.

The basic morphological operations are Dilation and Erosion. They are used for removing noise, isolating individual elements, and joining disparate elements.

Dilation expands the shapes contained in the input image using the so called "structuring function". It is mathematically defined as:

$$(f \oplus g)(x) = inf_{x' \in G}(f(x - x') + g(x')) \ ,$$

where f indicates the input image and g the structuring function.

The structuring function can be of any shape or size, but is often a relatively small square or disk with the reference point at the center. The structuring function is shifted over the image. If the structuring function consists of a region with value 0 and everything else is $-\infty$, Dilation is similar to a local maximum operator – the maximal pixel value in the region is determined and replaces the current image pixel (at the reference point).

Generally, Dilation causes an expansion (dilation) of the object in size, while Erosion is the converse process and causes a contraction of objects:

$$(f \ominus g)(x) = inf_{x' \in G}(f(x + x') - g(x')) \ .$$

The structuring function plays a leading role, but both Dilation and Erosion have a smoothing effect with Dilation smoothing concavities and Erosion eliminating protrusions.

The higher order operations, Opening and Closing, are defined by combining Dilation and Erosion:

$$(f \circ g)(x) = ((f \ominus g) \oplus g)(x) \ ;$$

$$(f \bullet g)(x) = ((f \oplus g) \ominus g)(x) \ .$$

(Köthe 1996) proposes to design the structuring function $g(x)$ as an "isotropic" square or disk with s the radius of the disk or the square's inscribed circle:

$$g_s(x) := d_s(x) = \begin{cases} 0, & if \ |x| \leq |s| \\ -\infty, & otherwise \end{cases} \tag{2.20}$$

For our problem domain there are two important characteristics of Opening and Closing when dealing with so-called "blobs":

- Light/dark blobs that smaller than $|s|$ can be eliminated by Opening/Closing with d_s.
- Light or dark blobs that larger than $|s|$ are not changed by Opening or Closing with d_s.

Here, blobs indicate smooth regions, which are lighter or darker than the background, and different from their environment. They have a close link to extrema of the image function (Mayer 1998).

Figure 2.11 presents results of morphological operators with square-shaped structuring functions. Please note that although in many cases Dilation has a similar effect to Closing (and Erosion to Opening), Opening and Closing do less damage to the original image information as the second step undoes parts of the first step.

2.5 Modeling and Visualization of Trees

The modeling of trees is an interesting but rather challenging topic in both botanical science and computer graphics although they have different interests. The latter is focusing on the realistic visualization of existing trees while the botanical scientists are more interested in the underlying rules of trees' growth. Current approaches in computer graphics try to make use of grammar descriptions, e.g., L-systems (Section 2.5.1), to simulate potential structures instead of directly drawing them. The combination of botanical- and geometry-based modeling helps to understand the organization of branches better and leads to more reasonable results.

2.5.1 L-systems

Lindenmayer-, or in short L-systems are a mathematical tool for modeling plant development based on formal grammars generating strings. They have been introduced by the Hungarian biologist Astrid Lindenmayer in 1968 and initially been used to simulate morphogenesis, e.g., the development of multicellular organisms. In the book "The Algorithmic Beauty of

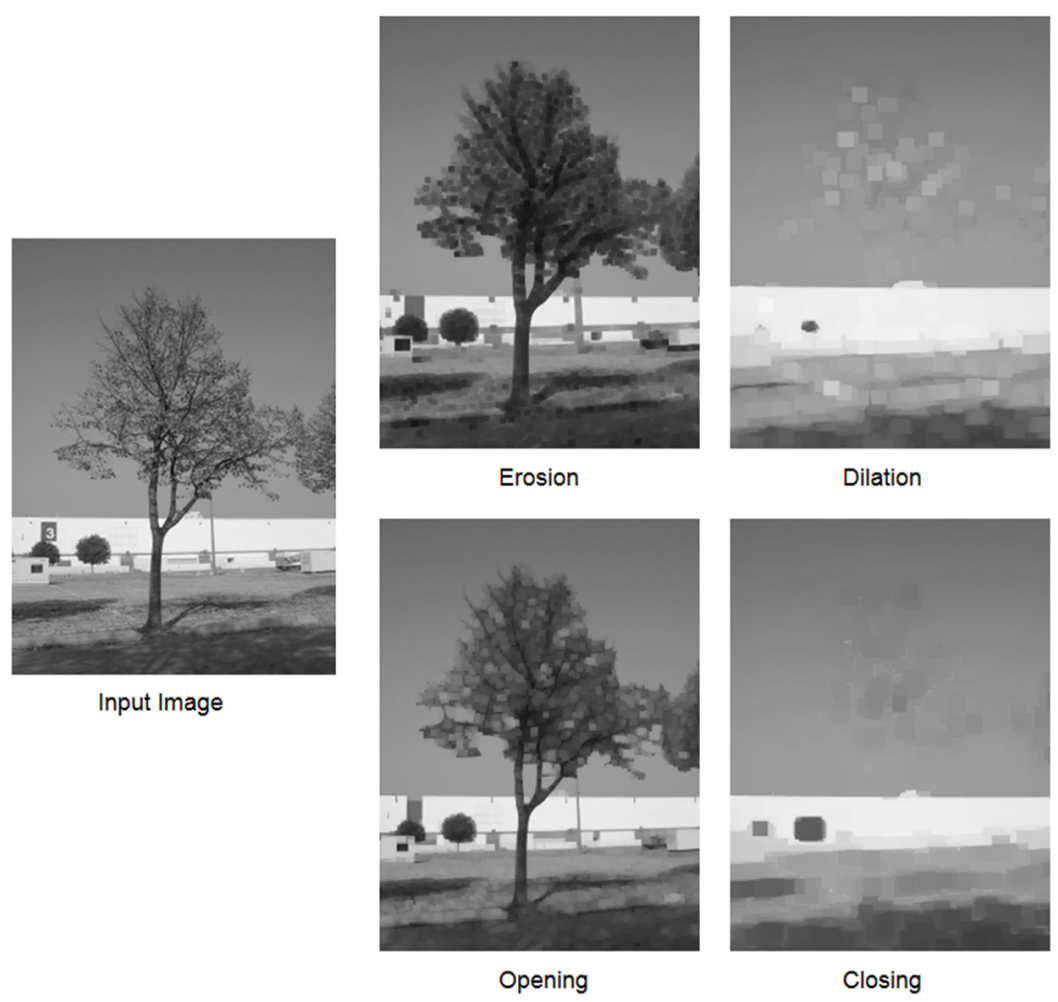

Figure 2.11: Results of morphological operators with square-shaped structuring functions with $s = 2pixels$ (for Erosion and Opening) and $s = 6pixels$ (for Dilation and Closing), respectively.

Plants", Prusinkiewicz and Lindenmayer (1990) describe algorithmic plant generation over time using L-systems.

L-systems are based on fractal mathematics (Koch snowflake or Koch island, introduced by Koch in 1904) and the concept of string rewriting, i.e., successive replacement of the input string, resulting into new strings based on predefined rules (called Productions). The simulation of development, i.e., branching of trees, starts with an initial string, called axiom. By means of string rewriting using Production Rules, subsequent strings grow in length seen to reflect development. An important property of L-systems is that they preserve the characteristics of the original string while generating new patterns.

Figure 2.12 shows an example of the growth process based on an L-system. It is generated by the simulation tool described in Section 4.2.4. A deterministic L-system, as the one shown in Figure 2.12, is the simplest instance of an L-system, where the structure is predefined with constant values, e.g., the branching angle of a new branch is fixed to 28°. In practice,

the most common extension for L-systems is parametrization, i.e., to set parameters such as length, diameter, and branching angle, to control the structure. This extends the flexibility and allows to design more sophisticated tree structures.

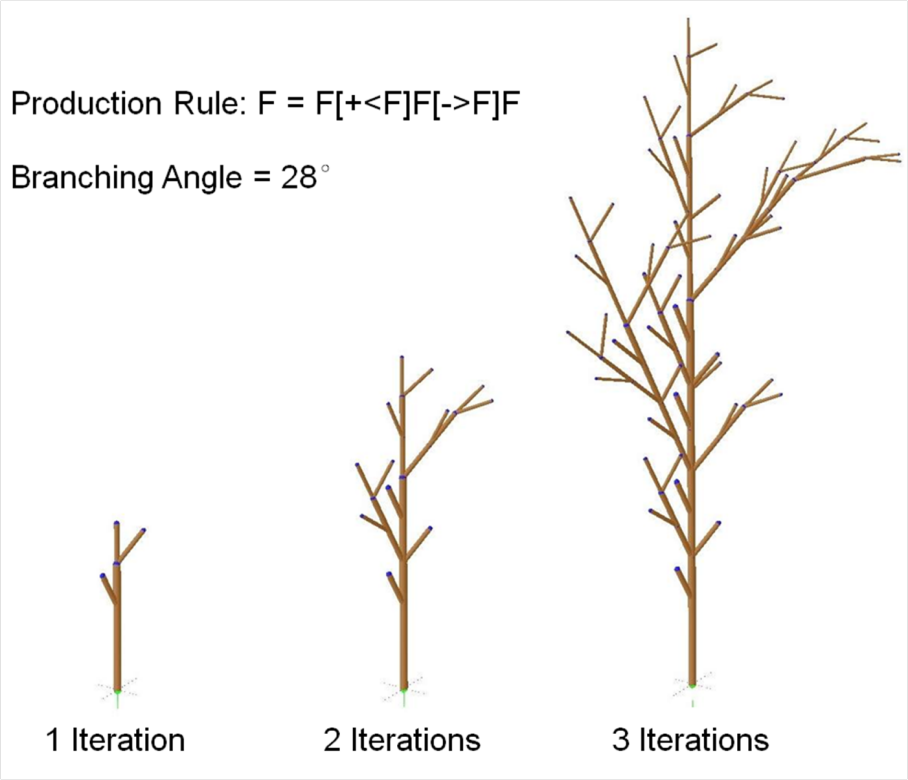

Figure 2.12: Example of branching structures generated by an L-system after 1 (left), 2 (middle), and 3 (right) iterations with the given Production Rule and branching angle (cf. also Section 4.2).

A further extension, namely stochastical L-systems, can simulate randomness during growth by a stochastic rewriting mechanism. Trees of the same type show a diverse appearance, which can be simulated by setting the values of the parameters and/or choosing the Production Rules randomly. The basic botanical characteristics are preserved by using a certain set of Production Rules. Random simulation results in unique tree models.

Because they recursively use the same Production Rule(s), models produced by L-systems show basic biological features such as a hierarchical structure and self-similarity. Modeling with L-systems thus enforces tree-like branching structures. Yet, L-systems alone only give means to (possibly) "randomly" generate and visualize trees, but not to analyze image data.

2.5.2 Approaches and Tools for Tree Modeling

Because of their elegant mathematical formulation, L-systems have become widely used in Computer Graphics. The modeling, visualization, and animation of trees are interesting for applications such as the production of movies and video games.

Related Approaches

Měch and Prusinkiewicz (1996) extend L-systems into a framework, termed "open L-systems", to model and visualize plants by their growth and their interaction with the environment. Plant and environment are modeled separately and information is exchanged. The development of plants is modeled considering branch collisions, the competition for growth space and light (cf. Figure 2.13), and the interaction between roots competing for water and soil.

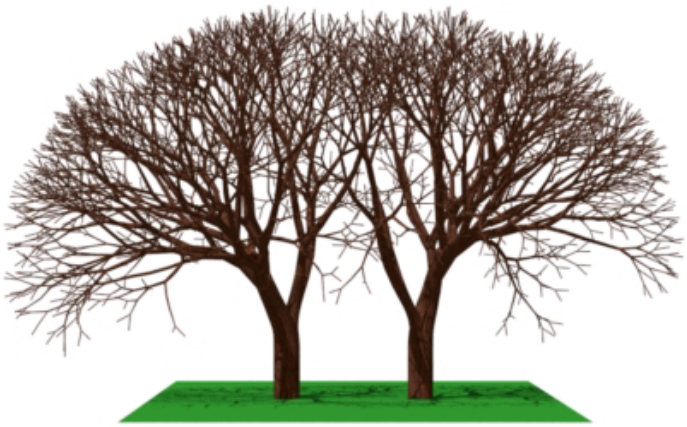

Figure 2.13: A model of deciduous trees competing for light (Měch and Prusinkiewicz 1996).

(Lintermann and Deussen 1996) presents a method for interactive modeling and animation of natural branching structures. Objects are created according to a rule based description. Geometric information is encapsulated in the objects and can be edited by methods such as free form deformation. Global and partial constraints allow the modeling of specific plants. The rule system is represented by a structure tree with components that can be edited graphically. Overall, complex branching structures can be developed fast and flexibly.

(Lintermann and Deussen 1999) gives an extension of the above approach by letting the user design a wide variety of plants by mostly intuitive mechanisms. The introduced modelor provides a highly interactive user-interface for plant generation. Figure 2.14 shows an example of a tree modeled with this approach.

Deussen and Lintermann (2004) present a broad overview over different ways to model vegetation, showing impressive results for the visualization of trees. Their focus is on a naturally looking visualization. They do not deal with the problems arising when one tries to invert visualization by extracting objects and estimating their parameters from given images.

Modeling with $Xfrog^{TM}$

The efforts of Bernd Lintermann and Oliver Deussen resulted in the software $Xfrog^{TM}$, which has been introduced in (Lintermann and Deussen 1996) and (Lintermann and Deussen 1999). $Xfrog^{TM}$ developed by *Greenworks* is one of the most popular tools for

Figure 2.14: Tree modeling by sequences of components: First, two components are combined and the parameters are adjusted (left). Two more branching levels are then constructed (middle). Adding the leaves yields the final tree (right) (LINTERMANN and DEUSSEN 1999).

plant modeling and animation. It provides a user friendly interface for interactive tree modeling (cf. Figure 2.15). Focusing on realistic visualization, the plants in $Xfrog^{TM}$ are divided into components such as tree, leaf, stem, wreath, and ball, not considering their botanical description. Users have comprehensive local control over each branch and leaf and can globally control environmental influences like phototropism and gravitropism. $Xfrog^{TM}$ works more like a simulator than an illustrator, i.e., the user cannot expect to reconstruct or design a specific tree. The result is a textured wireframe model, which can be exported in popular 3D formats, such as VRML, 3DMax, and Maya. We use $Xfrog^{TM}$ to produce sample trees for our experiments (cf. Figure 2.15 and Section 6.1).

2.6 3D Reconstruction

The geometrical basis of this work is highly precise 3D orientation and reconstruction for wide-baseline image sequences. A Structure from Motion (SfM) procedure is used to locate both the target object in the form of 3D points and the camera positions. (MAYER 2005) and (MAYER 2007) present a robust procedure to automatically relatively orient images highly precisely making use of calibration information.

We assume that the images are taken unconstrained with a hand-held camera and, therefore, (slight) rotations of the camera cannot be avoided. The vertical direction is important for tree extraction, because the trunk is supposed to be a (mostly) vertical line in the scene. An approach to determine the vertical direction from the vertical vanishing points in the images, which was devised in (MAYER and REZNIK 2007) for facade interpretation (cf. Section 2.6.2), is employed.

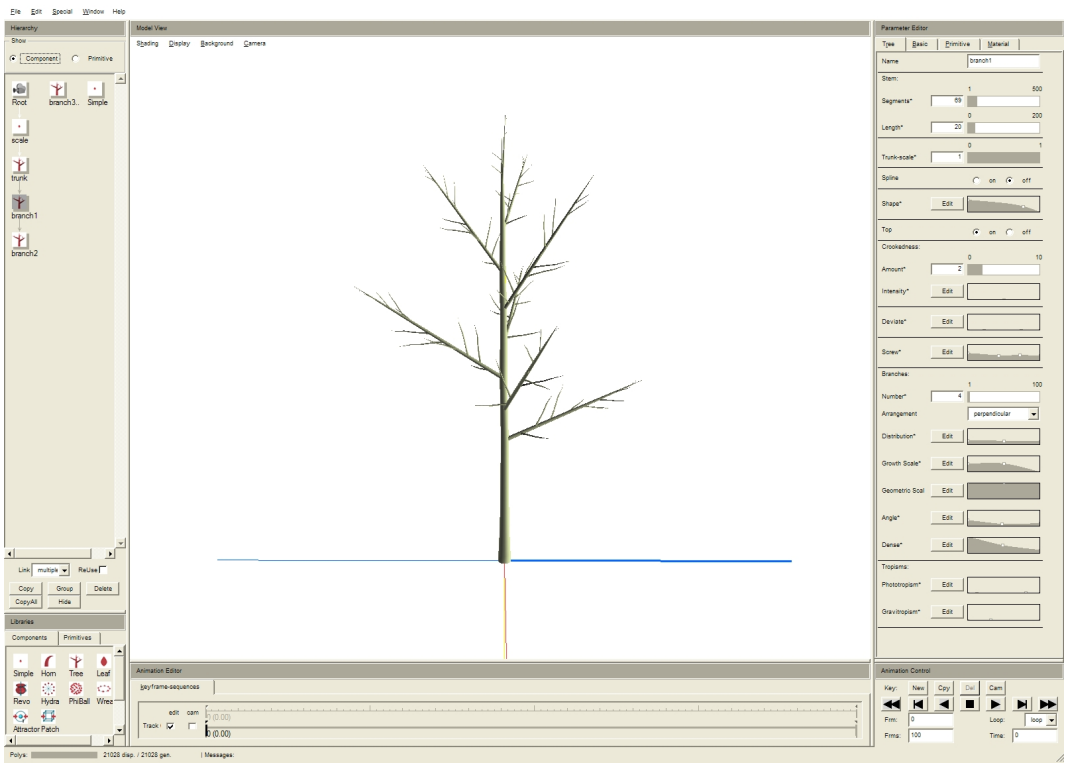

Figure 2.15: A simulated model for experiments in this thesis is generated by $Xfrog^{TM}$, "an interactive tool for tree modeling" (Lintermann and Deussen 1996).

2.6.1 Structure from Motion

SfM procedures are the basis for 3D object extraction from image sequences. SfM aims at producing 3D geometrical information from a set of (2D) images. A standard SfM setup consists in imaging a static 3D scene with a moving camera providing an image sequence.

A highly precise SfM procedure making use of camera calibration information, (Mayer 2005) and (Mayer 2007), is employed. At the core of the approach are least-squares matching, direct relative orientation by means of the five-point-algorithm (Nistér 2004), robust estimation based on random sample consensus – RANSAC (Fischler and Bolles 1981), and robust bundle adjustment.

First, Förstner (Förstner and Gülch 1987) points are extracted. They are matched via cross-correlation. Likely conjugate pairs are refined via least-squares matching with an affine geometrical model. These highly precise points are used for direct relative orientation employing the five-point-algorithm and (calibrated) trifocal tensors T (Hartley and Zisserman 2004) robustly estimated by means of RANSAC. The use of calibration information via the five-point-algorithm allows to much better deal with scenes close to or even plainly planar. Maximum sets of inliers to RANSAC are robustly bundle adjusted, allowing for a much better differentiation between correct and false hypotheses.

The trifocal tensors link two images to a third resulting in image triplets. Image triplets are connected based on the projection matrices for images common between triplets. E.g., the

triplets [1, 2, 3] and [2, 3, 4] have the image 2 and 3 in common. Already known 3D points are further projected into newly linked images to generate i+1-fold points, with i being the current number of images the point is visible in. After the connection, the sequence is bundle adjusted.

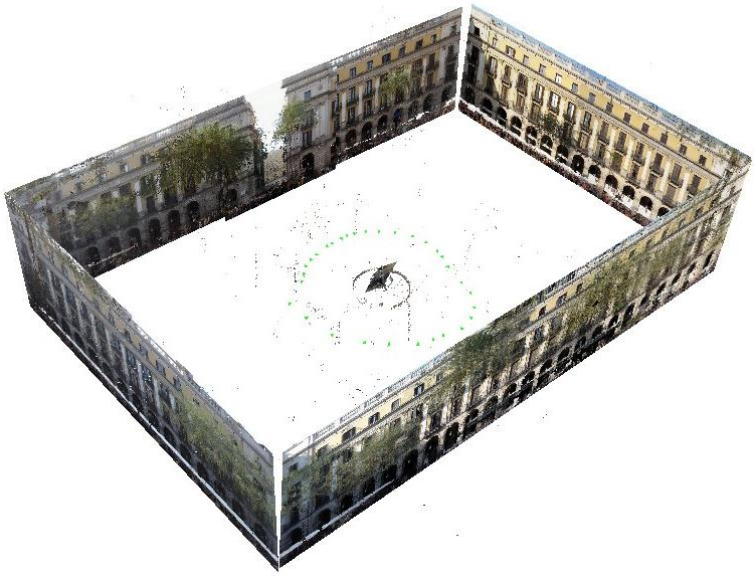

Figure 2.16: Result of structure from motion reconstruction and detection of vertical planes (Mayer 2007).

2.6.2 Vertical Orientation Using Vanishing Points

In (Mayer 2007) an approach to determine the vertical direction in 3D from the vertical vanishing points in the images of a sequence is devised for facade interpretation. Facades usually comprise a considerable number of vertical lines. The 3D Euclidean model can be oriented vertically based on the vertical vanishing point derived from the vertical lines robustly extracted using RANSAC (Fischler and Bolles 1981) along with the given calibration information. Figure 2.16 shows planes determined based on the vertically oriented model via RANSAC.

The vertical direction, i.e., the direction of gravity, is also essential for modeling the growth of trees and their extraction. Particularly, most trunks grow nearly vertically. We assume that a number of vertical lines, e.g., of building facades, can be found in the background for images taken in urban scenes enabling the determination of the vertical direction.

Figure 2.17 gives an example (one scene in Chapter 6; cf. Figure 6.4) for 3D orientation: green pyramids show the camera positions and spheres indicate the 3D feature points.

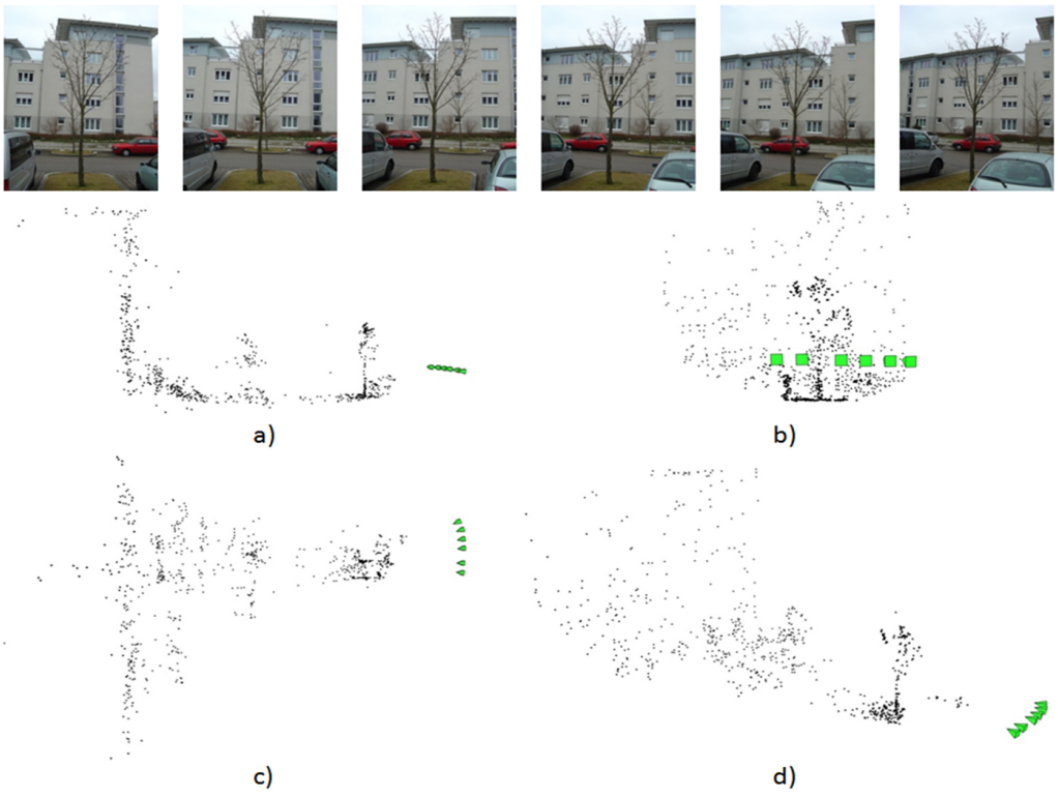

Figure 2.17: Six images (top) and 3D orientation for tree in the foreground (cf. Section 6.2) seen from a) left, b) front, c) right, and d) "bird view". 3D points are presented as gray spheres, with the color taken from the images. The green pyramids present the cameras with the tip at the location of the projection center and the base symbolizing the orientation of the image plane.

Chapter 3

Related Research

Early work on tree extraction was based on aerial imagery, e.g., (Pinz et al. 1993), and only later Light Detection and Ranging (LIDAR) data (usually recording first and last pulse) was used. It focused on forest inventories trying to determine parameters such as tree height, density of trees, and stem diameter. Recently, novel small footprint and full waveform LIDAR systems have been successfully used for more advanced tasks such as single tree segmentation and tree species classification. Most related work has focused on aerial images and LIDAR data or a combination of them.

Because of the weaker relationship to our approach, in this chapter we only summarize the work on tree extraction based on 1) aerial images and LIDAR data (restricted to individual trees) and 2) terrestrial LIDAR data. We only present details for approaches which employ terrestrial images as we do.

3.1 Extraction of Individual Trees from Aerial Images and LIDAR Data

As we deal in our work with individual trees, we restrict the discussion of approaches for aerial data to those focusing on single trees.

Of first interest is (Cheng et al. 2006), which like our work proposes a statistical (Bayesian) framework also consisting of a generative component for extracting individual trees from aerial images. It comprises segmentation, stereo, and 3D fitting. Both data-driven (inverse modeling, 2D data to 3D geometries) and generative – model-driven (3D models to 2D images) components are integrated. 3D fitting based on super-ellipsoids resolves inconsistencies of segmentation and stereo. Experiments with small synthetic and real scenes show the basic feasibility of the approach.

Most research on aerial data focuses on LIDAR data, possibly in conjunction with image data, the latter particularly in the near infrared, where vegetation has particularly distinctive reflection properties.

Hyyppä et al. (2005) aim in their empirical study on the accuracy of the estimation of tree volumes from aerial data. They estimate a segment for each tree from image data, but found, that the height is much more reliably determined from LIDAR data than from data generated by image matching.

Persson et al. (2004) use LIDAR data for the determination of the outline as well as the height of individual trees. Image data, again in the infrared, is used to mostly reliably differentiate pine, spruce, and deciduous trees.

Andersen et al. (2002) present a Bayesian object recognition framework for the analysis of forest scenes. The scenes are modeled by the locations, heights, and crown configurations of the trees in the area (cf. Figure 3.1). Posteriors for the parameters are inferred via RJMCMC (cf. Section 2.1.3), which is then used in the MAP (cf. Section 2.2.2) estimation.

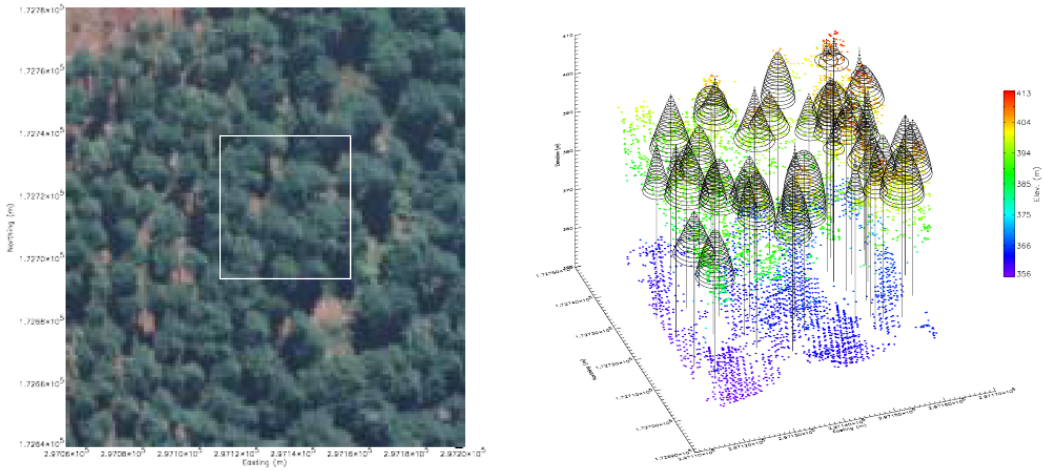

Figure 3.1: A forest scene (left) and the MAP estimate of tree locations and crown dimensions superimposed on LIDAR data (Andersen et al. 2002).

Reitberger et al. (2006) employ full waveform LIDAR data focusing on the classification into deciduous and coniferous trees. They show, that by defining features making use of the high point densities possible by the full waveform information, it is possible to reliably distinguish the two classes by means of unsupervised classification.

Finally, Horváth et al. (2008) construct a "gas of circles" model – regions in the image domain composed of an unknown number of circles of approximately the same radius – using higher-order active contours for tree crown extraction from aerial images.

3.2 Tree Extraction from Terrestrial LIDAR Data

With the more widespread availability of terrestrial laser-scanners, there is a growing interest to use them in forest applications. The large number of points from terrestrial scans renders it possible to generate very detailed models. Yet, opposed to our work, the focus of the research is mostly on the determination of parameters for forest inventory.

Aschoff and Spiecker (2004) compute horizontal layers from oriented scans, detect cross-sections of the trunks as (semi-) circles by Hough transformation in them, and then fit circles or ellipses. Cross sections at a height of 1.3 m are used to compute the diameter at breast height (DBH), a very important parameter for forest inventory.

Bienert et al. (2006) have conducted a number of experiments showing that it is possible to mostly robustly automatically detect trees and determine forest inventory parameters such as DBH and tree height. They determine the ground height in a histogram of point heights and based on it they compute the density of points in a horizontal layer at a height of 1.3 m. If the density is high, they try to fit a circle. If a circle with the typical diameter of trees is found, it is checked by using the same procedure a couple of centimeters higher, thus avoiding random results for bushes.

Closer to our work, (Pfeifer et al. 2004), (Gorte and Pfeifer 2004), as well as (Gorte and Winterhalder 2004) focus on detailed models for trees (cf. Figure 3.2). Besides computing the visual hull on horizontal layers for dense coniferous trees, which the laser cannot penetrate, Pfeifer et al. (2004) aim at fitting cylinders to the trunk and the thicker branches. The fitting is based on non-linear least-squares optimization and thus needs initial values. A partial solution for the latter is to track detected branches by extending a found cylinder and searching for points close to it. A possibly better solution is provided by Gorte and Pfeifer (2004): The laser points are rasterized in 3D voxel space and operations such as closing from mathematical morphology and thinning are used to obtain a connected 3D skeleton (Figure 3.2, b). By means of shortest path computation wrong connections are eliminated. A labeled skeleton is obtained, which could be used to determine centers for the cylinders for (Pfeifer et al. 2004). (Gorte and Winterhalder 2004) further segments the remaining voxels into different branches and finally, branch labels are assigned to the original laser points (Figure 3.2, c). We note that for all these approaches a relatively high point density is needed to avoid gaps in the skeleton, or even more critical, to fit cylinders at all.

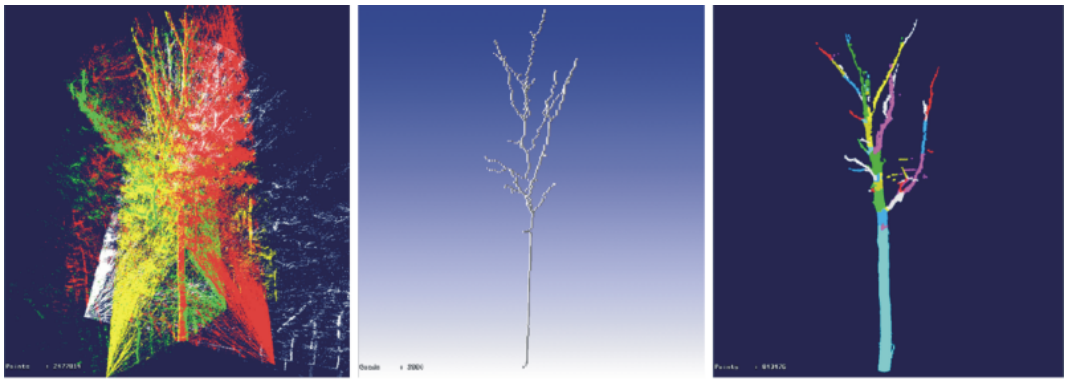

Figure 3.2: Detailed modeling of individual trees based on laser point cloud segmentation. a) Combined laser point cloud from four scan positions. b) Skeletonization. c) Labeled laser points indicating different branches (Gorte and Winterhalder 2004).

3.3 Tree Extraction from Terrestrial Images

The approaches introduced in this Section are more similar to our work, employing terrestrial images as the only data source and focusing on the representation of individual trees with their specific characteristics.

Haering et al. (1997) segment groups of foliaged deciduous trees in color images based on texture without any 3D interpretation. Also (Forsyth et al. 1996) focus only on a 2D interpretation, yet for individual trees. They particularly model the symmetries of coniferous trees.

In (Sakaguchi and Ohya 1999), images from multiple viewpoints are employed. First, the images are converted to gray-scale images and the tree silhouette in each image is extracted by manually removing the background. A volume is then carved out by intersecting the view cones generated from the tree silhouettes in multiple images. The voxels of the volume are colored with the average brightness of the rays from the different images. A branching process is started on the ground extending into dark areas assumed to correspond to the trunk or branches. The given results are plausible, but there is much human intervention involved.

Shlyakhter et al. (2001) generate volumes as in (Sakaguchi and Ohya 1999), but from them 3D medial axes are constructed preserving the "botanical fidelity of the branching pattern and the leaf distribution" (Shlyakhter et al. 2001) via an open L-system (cf Section 2.5.1). Figure 3.3 shows the proposed framework, in which the manual intervention includes the segmentation of trees from background and the control of the L-system, e.g., the determination of Production Rules and the number of iterations.

In (Reche et al. 2004), as shown in Figure 3.4, volumetric opacity estimation is used for geometrical reconstruction and a realistic visualization is achieved by means of view-dependent interactive texturing. Neubert et al. (2007) employ 3D particle flow for the estimation of the tree volume in voxel space (cf. Figure 3.5). Trees are captured and rendered by estimating opacity in a volume, then generating and displaying view-dependent textures attached to cells of the volume. Textures generated based on the estimated opacity are attached to billboards in cells of the volume.

(Quan et al. 2006) presents a semi-automatic approach for modeling and reconstructing plants from close-range images (cf. Figure 3.6). It focuses on realistic visualization with interactive aid for the segmentation by the user and uses high quality SfM (cf. Section 2.6.1) and dense depth estimation as basis. The geometry of each leaf is automatically determined from the multiple views by fitting a deformable generic leaf model. The texture of each leaf comes from image segmentation or the generic model if the leaf is occluded in the images. The high quality result is ensured by much manual effort and the proposed model only works well for relatively large leaves and thick branches compared to the image resolution.

Some of the shortcomings in (Quan et al. 2006) are overcome by (Tan et al. 2007), which is again based on high quality SfM and dense depth estimation (cf. Figure 3.7). In contrast to (Quan et al. 2006), not each leaf is extracted individually, but the leave population is partly supplemented with leaf replicas. First, the source image is segmented into leaf-regions based

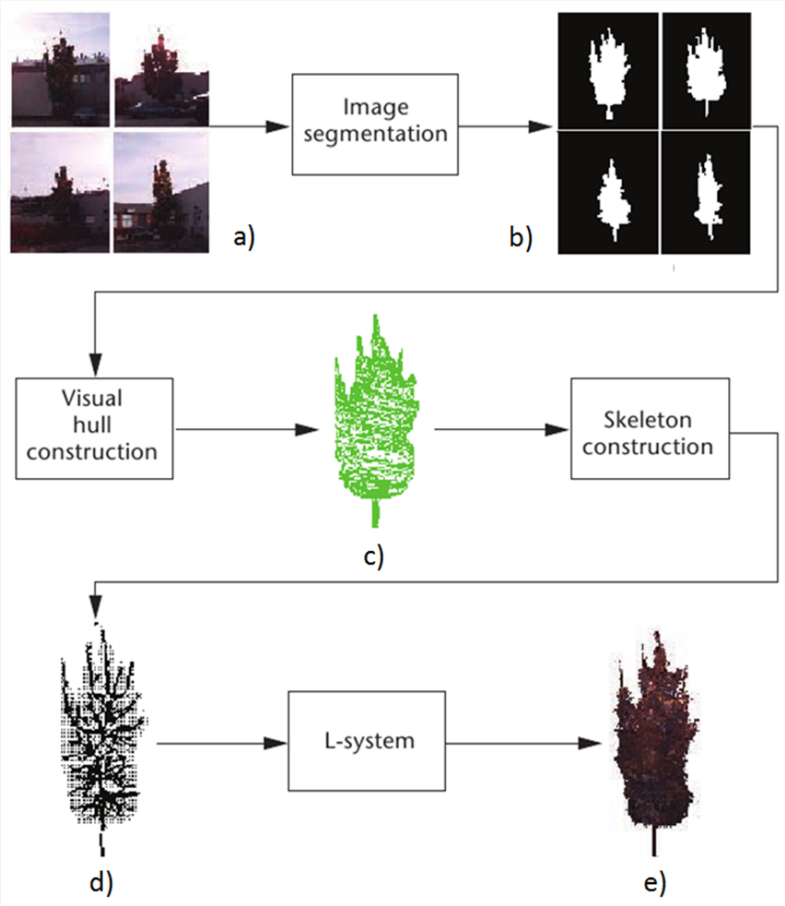

Figure 3.3: Tree extraction with visual hull and skeleton construction: Each of the input images a) is segmented into the tree and the background b), and the silhouettes are used to construct the visual hull c). The system constructs a tree skeleton d) as an approximation to the medial axis of the visual hull, and an L-system is applied to grow the small branches and the foliage e) (Shlyakhter et al. 2001).

on color employing the user-defined image footprint of leaves. The 3D position of each leaf is determined by the closest 3D points or estimated branch segment. Its orientation is approximated by fitting the shape of the region to the leaf model or leaf points in its vicinity. In the reconstruction it is tried to utilize the structural regularity, i.e., self-similarity and structural patterns of branches and arrangement of leaves, of trees. Yet, practically just shape patterns of visible branches are employed to reconstruct missing parts of branches. The modeling is more automatic, but the user needs to set several parameters for the recovery of the occluded branches and leaves and is given the option to refine the branch shapes.

The most current approaches are (Chen et al. 2008) and (Tan et al. 2008). Both model trees based on user sketches achieving very realistic-looking results.

In (Chen et al. 2008) the input is a freehand sketch from the user including main branches and optionally the crown. The sketch is matched to a database of predefined tree models, which contains typical tree exemplars and their associated global parameters, to find a suit-

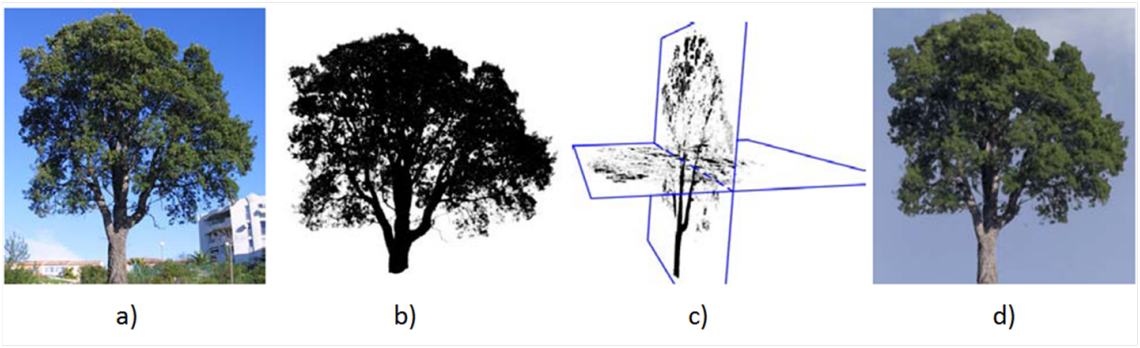

Figure 3.4: Capturing and rendering of trees from photographs: a) An image of an oak. b) Mask used for opacity estimation. c) Two cross slices of the resulting opacity. d) Synthetic image from the direction of the original view, using view-dependent rendering (Reche et al. 2004).

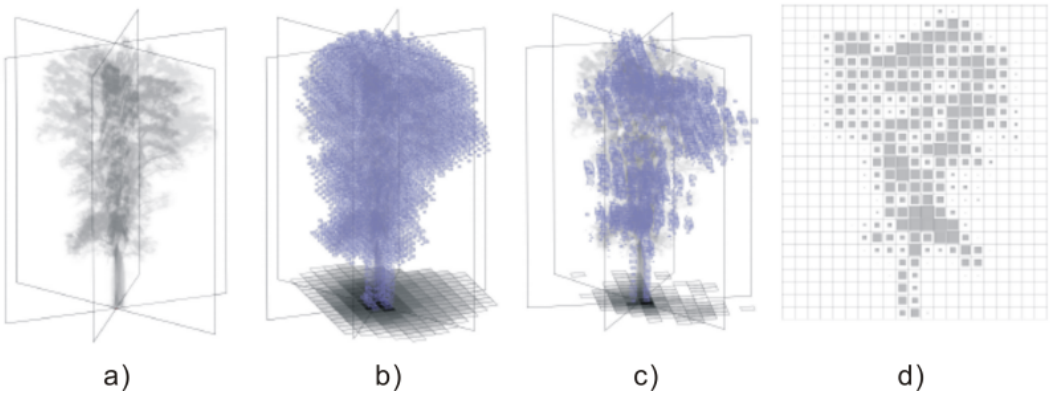

Figure 3.5: Estimation of the tree density: a) Initial density values from three input images. b) Voxel grid by back-projection. c) Refined voxel grid. d) Density values for one image plane. High density values are marked by large squares (Neubert et al. 2007).

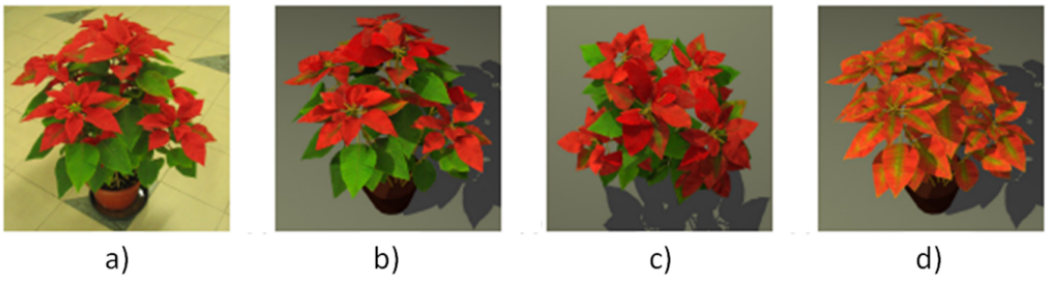

Figure 3.6: Image-based modeling of poinsettia plant: a) One image out of 35 images. b) Recovered model rendered from the same viewpoint as a). c) Recovered model rendered from a different viewpoint. d) Recovered model with modified leaf textures (Quan et al. 2006).

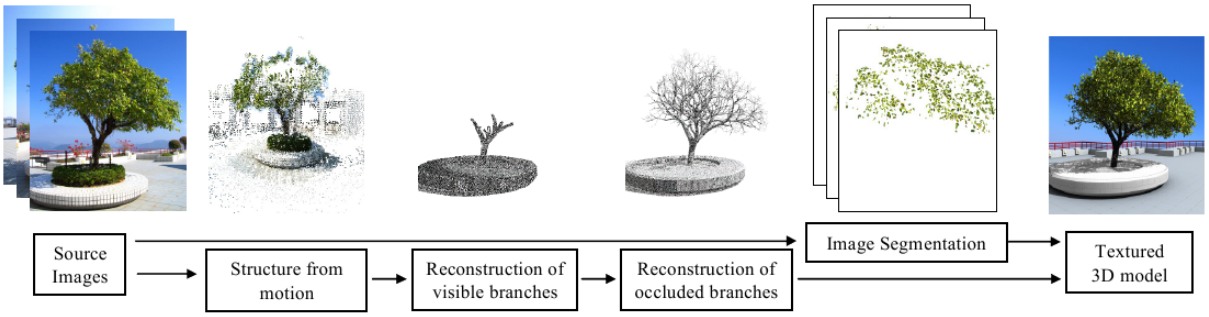

Figure 3.7: Image-based tree modeling based on high quality SfM (Tan et al. 2007).

able "template" for modeling. The matching is conducted by comparing the sketch and the orthographic projection of the tree exemplar by means of a Markov Random Field (MRF) with each branch segment modeled as a node and its depth as a variable. Similar to (Tan et al. 2007), the principle of self-similarity is used to complete the rest of the branches and the leaves (cf. Figure 3.8). The template for leaves is determined by the tree template or selected by the user from the database.

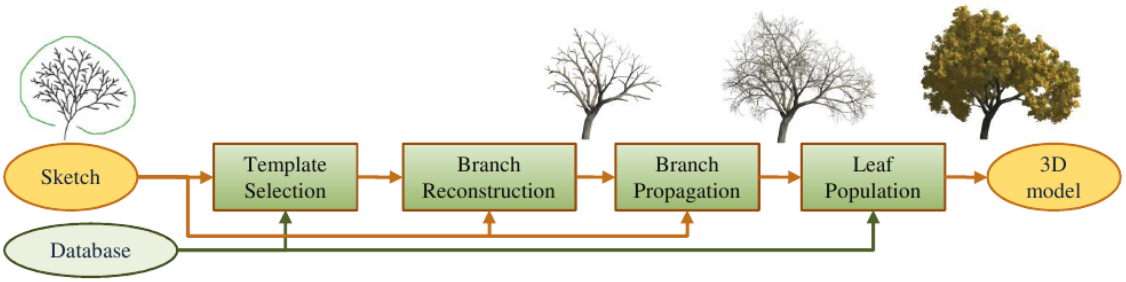

Figure 3.8: Overview of the sketch-based tree modeling (Chen et al. 2008).

Tan et al. (2008) (cf. Figure 3.9) use a sketch and a (single) image for the modeling of trees. Users are required to indicate the crown in the form of the leaf region as well as the branch structure including the trunk from its root upward and possibly a part of the visible branches. The visible branches are completed by image matching. Their encoded structure patterns together with some predefined patterns are used to build up a small library of elementary sub-trees for the derivation of the entire tree. Tan et al. (2008) have improved their rule based basic method by simulating the growth of trees by the replacement with sub-trees. Leaves are then added according to the branching structure and textured based on the input image.

All the given approaches provide plausible results in terms of visualization, but substantial human intervention is needed. Approaches conducting a fully automatic extraction are still limited.

Figure 3.9: Tree modeling from a single image: a) Input image. b) Strokes drawn by the user: only two strokes for this example. c) Automatic synthesis of the tree branches. d) Complete tree model rendered from the same viewpoint as the input image (Tan et al. 2008).

Chapter 4

Modeling of Trees

The target objects of this thesis are unfoliaged trees, which implies visible branching structures. The 3D tree structure has a high complexity because of the numerous components (branches) and their weak mutual constraints. When it is projected into terrestrial images from different points of view, the order of branches on corresponding epipolar lines in different images might change. Additionally, the branches can have a weak contrast with the background, can be occluded by each other, and disturbed by clutter in the background. Because of all this, it is one of the harder objects to be extracted in urban scenes.

In our generative statistical approach, for which preliminary versions have been published in (Huang and Mayer 2007a, Huang and Mayer 2007b, Huang 2008, Huang and Mayer 2009), we employ L-systems (cf. Section 2.5.1) for the modeling of trees. Parametrized and context-sensitive L-systems are devised for a profound semantic description of branching structures.

For the actual 3D reconstruction, a branch is modeled as a cylinder or a cylinder sequence, using VRML for graphical interpretation. VRML (Virtual Reality Modeling Language, also known as Virtual Reality Markup Language formerly) is a standard file format for 3D vector graphics. We chose VRML because it is a cross-platform format and can be rendered by many popular (free or commercial) 3D modeling tools and web browsers. The version we use is VRML 2.0 (ISO/IEC 14772-1:1997).

In this chapter, we firstly introduce three basic branching types that we employ for trees in Section 4.1. In Section 4.2, L-systems are devised for the pre-defined branching types. The geometrical representation of tree models in 3D is described in Section 4.3. Finally, based on botanical and empirical knowledge, generic priors for branching parameters are given in Section 4.4 for the initial sampling.

4.1 Branching Types

We integrate basic botanical knowledge for the branching structure of trees into our modeling. According to (Deussen and Lintermann 2004), branching structures of trees can be basically divided into two main groups: “monopodial” and “sympodial”.

The monopodial – m branching system (cf. Figure 4.1, m) has a prominent main axis, which is stronger and longer than the side branches. The side branches are again stronger and longer than their side branches of the second order, etc. Because of the dominant axis, monopodial branching structures have a radially symmetric crown.

Figure 4.1 (sd) and (sm) show the two main types of sympodial branching. "Sympodial-dichasium" – sd branching means, that two buds sprout and grow synchronously. Thus, these trees have bilaterally symmetric crowns. The most common branching structure is "sympodial-monochasium" – sm, where one of the secondary branches has approximately the same direction as the original branch. Sympodial-monochasium branching results into only partially symmetric structures, which still often appear very similar to monopodial branching.

Figure 4.1: Branching types: (m) monopodial; (sd) sympodial-dichasium; (sm) sympodial-monochasium.

Knowing the branching type helps to describe the structure more specifically. After the classification of branching types (cf. Section 5.6), specific Production Rules (cf. below) can be used for the further extraction.

4.2 L-systems for Tree Modeling

We generatively model trees by means of L-systems (cf. Section 2.5.1), which provide descriptive power for the structure of trees in terms of their growth, or more particularly branching. As shown in Fig 4.2, the modeling starts based on the predefined Variable(s) V, Constants S, and the Initial State ω. The core process of L-systems is the generation of

symbol-chains (sometimes also called “strings”) with the given Production Rules P to describe the growth of trees. In generative statistical modeling, prior distributions are added for the sampling of parameters.

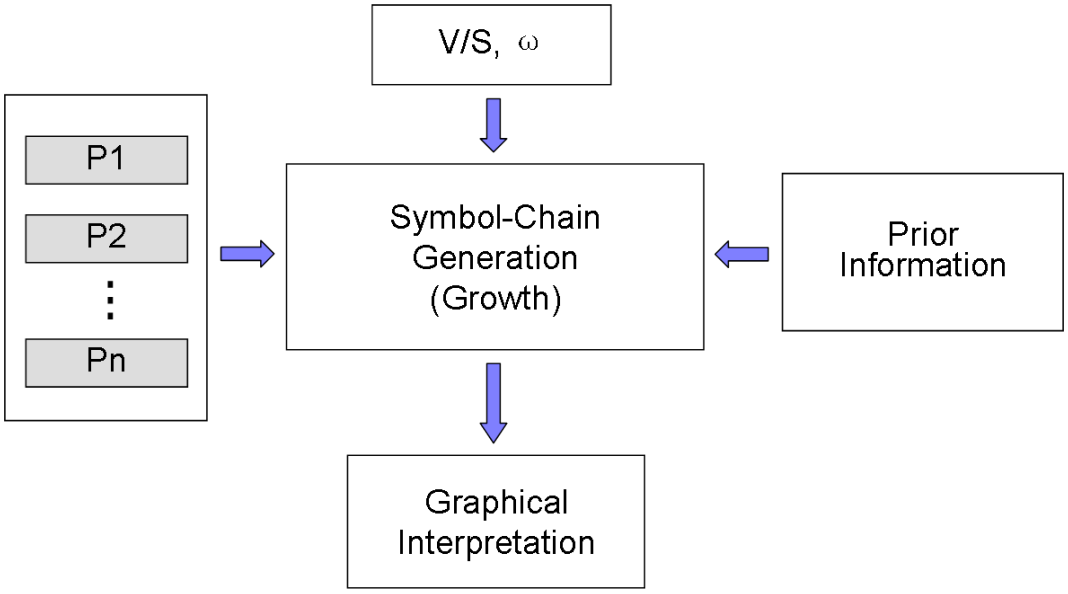

Figure 4.2: Overview of L-system.

4.2.1 Parametrized L-systems

An L-system is formally defined as follows:

$$G = (V, S, \omega, P)$$

with

V (Variable): F

S (Constants): +, –, <, >, [,]

ω (Initial State): F

where the Variable “F” corresponds to growth, i.e., a new branch, while the Constants describe rotation around certain axes (“+” and “–” indicate turn left and right – inclinations; “<” and “>” indicate roll left and right – azimuths) and the creation of sub-branches (enclosed by brackets “[” and “]”). The Production Rules instruct to replace “F” with the given string in the next iteration.

We have devised Production Rules for monopodial – m, sympodial-dichasium – sd and sympodial-monochasium – sm branching as follows:

$$\begin{aligned} P_{(m)} &: \text{F} = \text{F[+>F][–<F]F} \\ P_{(sd)} &: \text{F} = \text{F[+<F][–<F]} \\ P_{(sm)} &: \text{F} = \text{F[+>F]F} . \end{aligned}$$

The Variable and the Constants are parametrized, i.e., the values of the size of the cylinder and the angles of rotation are not fixed. A Production Rule can be expressed in parametrized form, e.g.,

$$P_{(m)} \quad : \quad \mathrm{F} = \mathrm{F}(l_1,d_1)[+(\beta_1) >(\alpha_1)\ \mathrm{F}(l_2,d_2)][-(\beta_2) <(\alpha_2)\ \mathrm{F}(l_2,d_2)]\mathrm{F}(l_3,d_3)\ ,$$

which indicates the basic parameters:

l_i : Length;
d_i : Diameter;
α_i : Angle of azimuth – angle with x-axis of branch projected into horizontal plane;
β_i : Angle of inclination – angle between branch and horizontal plane.

The setup of the three coordinate-axes as well as the definition of the parameters are shown in Figure 4.3.

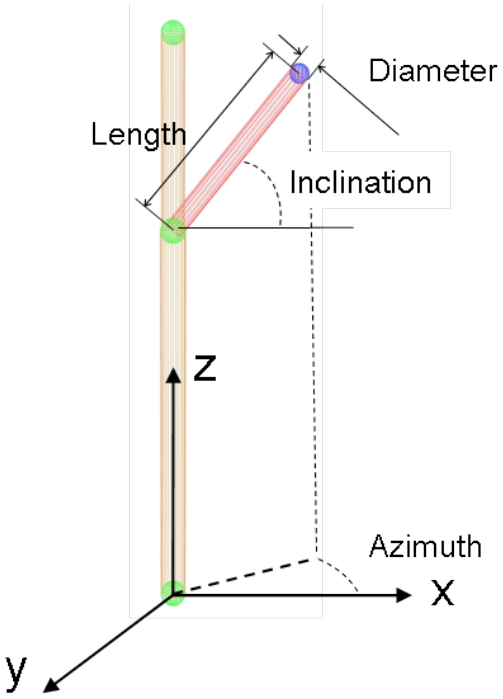

Figure 4.3: Definition of parameters.

Parametrization is an important extension for L-systems largely enhancing the flexibility of modeling. For instance, a so-called “stochastic L-system” can generate a large number of different examples from a simple set of Production Rules by sampling the values for parameters randomly. In the generative statistical framework we propose, the parameters are estimated by a sophisticated statistical search, as described in Chapter 5, adapting the models to trees in real scenes projected into images from different points of view.

4.2.2 Dual-variable L-systems

When extending L-systems to dual-variable systems, additionally to the basic Variable "F", which indicates growth, an additional Variable "I" is employed to control the branching. "I" is only used to derive the next level string and does not generate any branches.

For example, for monopodial – m trees, the L-system G_m is defined as follows:

$$G_{(m)} = (V, S, \omega, P)$$

with

$V_{(m)}$ (Variables): F, I

$S_{(m)}$ (Constants): +, –, <, >, [,]

$\omega_{(m)}$ (Initial State): I

$P1_{(m)}$ (Production Rule): F = FF

$P2_{(m)}$ (Production Rule): I = F[+>I][–<I] I ,

where the Production Rules instruct to replace "F" or "I" with the given strings in the next iteration.

Starting from $\omega_{(m)}$=I, after two iterations of P2, applying it to all possible instances of "I", and one iteration of P1, the result is:

$$\text{FF[+>F[+>I][–<I]I][–<F[+>I][–<I]I]F[+>I][–<I]I} \ .$$

Using the Production Rule P1 (possibly several times) results into branches with a number of connected "F"s accumulated. Hence, there are two description levels of branches: The first level is "branch", i.e., a sequence of "F"s (cf. Figure 4.4, right), while the second level is "branch components", i.e., individual "F"s.

The L-systems for sympodial-dichasium – sd and sympodial-monochasium – sm trees are defined in the same way, but with different Production Rules for "I":

$$\begin{aligned} P2_{(sd)} &: \text{I = F[+<I][–<I]} \\ P2_{(sm)} &: \text{I = F[+>I]F[–<I]I} \ . \end{aligned}$$

A Production Rule can again be expressed in parametrized form, e.g.,

$$P2_{(m)} : \text{I = F}(l_1,d_1)[+(\beta_1)>(\alpha_1)\text{I}][–(\beta_2)<(\alpha_2)\text{ I}]\text{I} \ .$$

As the additional variable can be seen as a "structuring variable", dual-variable L-systems have an extended ability to control the branching structure. Figure 4.4 shows a comparison of results from single- and dual-variable L-systems.

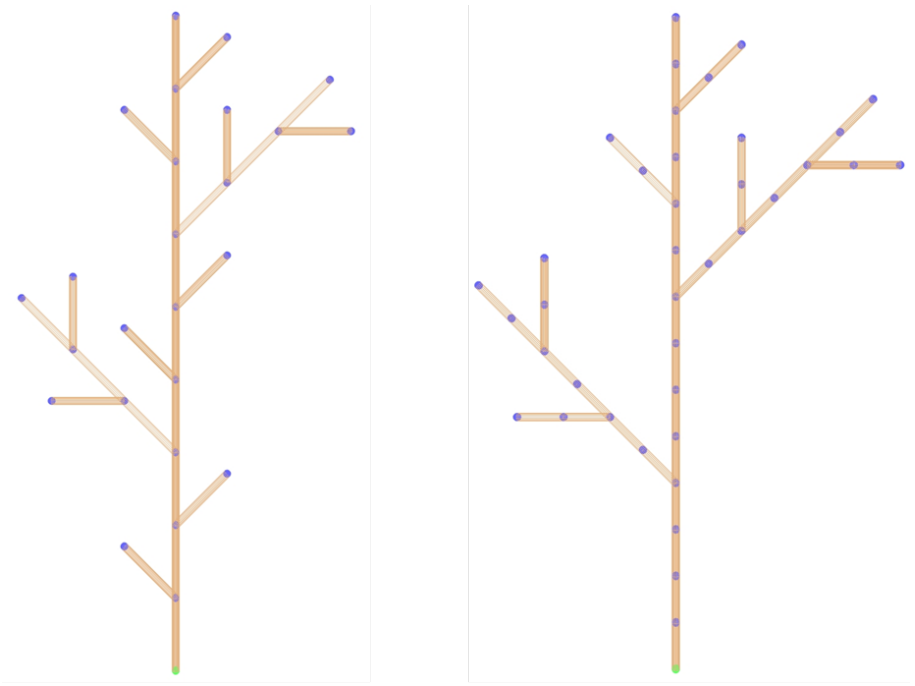

Figure 4.4: Resulting branches for dual-variable L-system (right) compared to the result from single-variable system (left) with similar Productions (bottom). It is assumed that all branches/branch components have the same length.

The most important advantage of dual-variable systems is their hierarchical representation of branches on different levels. As branches are described in the form of sequences of "F"s, the predecessors always have a larger or at least an equal number of "components" than their successor. This implies that the parent branches have a higher probability to be longer, stronger, and most important, more complex in geometry. Dual-variable systems thus allow for a more flexible structure control and hence have the capability to represent botanical features more precisely.

4.2.3 Context-sensitive L-systems

The L-systems that we use are "context-sensitive" (also known as "context-dependent"). They have the advantage that influences between members, i.e., branches can be represented. Particularly, context influences the values of parameters, or more precisely, their distributions (cf. also Section 5.7). Table 4.1 shows the relationships between parameters. The different types of influence can be summarized as follows:

1. Iid (independent and identically-distributed) parameters: There is no influence on each other. Each parameter follows a distribution with a range of values.

Relation	Azimuth (α)	Inclination (β)	Size (l, d)
First Level Only	full circle	iid	iid
Inside Level	neighbor-based	neighbor-based	iid
Between Levels	predecessor-based	predecessor-based	contraction coefficient

Table 4.1: Context relationship of branch parameters.

2. Neighbor-based parameters: The interaction of adjacent branches (neighbors) represents the competition for growth space. This narrows the distribution range for the parameters of each branch based on the density of its neighbors.

   The Azimuth on the first level is a special case. There, the Azimuths should be distributed over the full circle [0°,360°), but at the same time leave space between neighbors. However, no prior information limits the direction of each branch. Thus, each Azimuth still follows iid on the full range, but values close to each other will be accumulated into one hypothesis. This results in stronger hypotheses because more evidence is concentrated.

3. Predecessor-based parameters: This models the influence of parent branches on their children. The values of the predecessor are employed when deriving the distributions of its successors.

4. Contraction coefficient: This relationship is similar to the predecessor-based one. It is listed separately, because only a fixed ratio is used rather than deriving distributions.

Detailed information on the actual implementation is given in Section 5.7.

4.2.4 L-system Simulator

We have developed a simulation tool for L-systems, first to understand their basic behavior, but also as the L-system generator when extracting trees from images. An example generated by it simulating a plant's growth process is shown in Figure 2.12 (cf. Section 2.5.1). This example is quite simple as the parameters are fixed and only one Production Rules has been used. More simulation results can be found in Figures 4.4 and 4.5 (2D and 3D). The simulator consists of two main parts:

1. String generation: Starting from the Initial State ω, new strings, possibly with dual-variables, are generated according to the pre-defined Production Rule(s).

   - The parameters are stochastically sampled according to the given distributions, e.g., Gaussians.
   - The Production Rules are (possibly stochastically) selected.
   - The growth level is controlled by the manually given number of iterations.

2. Representation in 3D: The final string is interpreted and visualized using VRML.

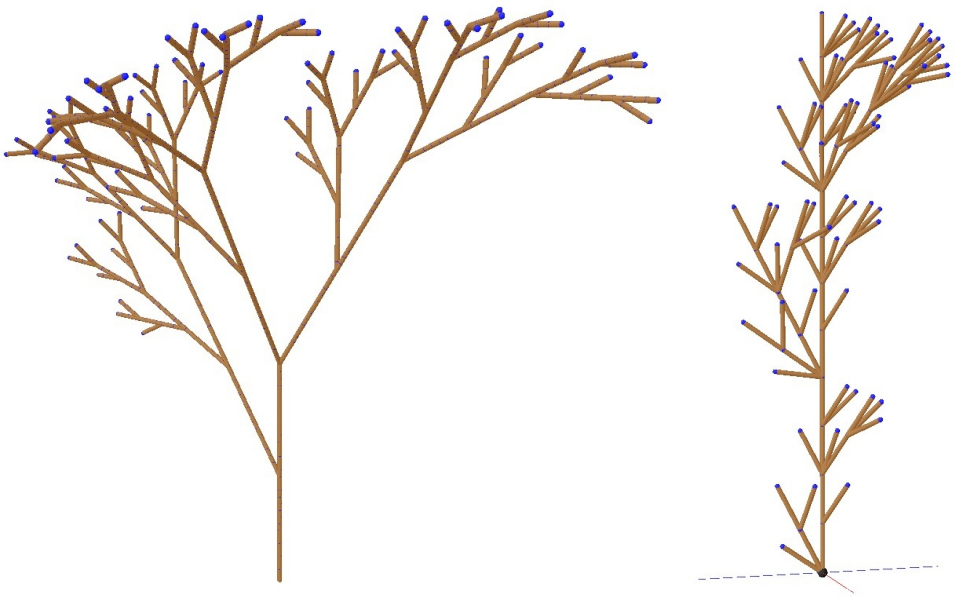

Figure 4.5: Examples generated by L-system simulator.

4.3 Geometrical Modeling of Branches

We employ cylinders as the geometrical primitive for the 3D modeling of trees. Branches are represented as cylinders or cylinder sequences.

As we focus on automatic and precise reconstruction rather than realistic representation, relatively simple models are used instead of triangle or polygon meshes used for representing 3D geometry in visualization. The most important reason is that simple 3D models can be more efficiently projected into 2D image space. As we model trees generatively, for evaluation (cf. Section 5.4.1) every hypothesis needs to be projected into the 2D image for each camera position. Because a large number of projections needs to be computed, a simple geometric model is advantageous. Another advantage of a simple model is that it can be more reliably derived, as fewer parameters can be determined more stable.

Varying Width

For the dominant structure of trees, i.e., trunks and main branches, the variation of width is sometimes hard to be ignored. A simple cylinder is no more appropriate for the representation. A cylinder sequence, which consists of multiple cylinders providing much more flexibility for modeling, is therefore employed. The parameters for each component are then varied, to match the local details.

Figure 4.6 shows an example. The cylinder pieces are connected by spheres to avoid abrupt changes of the profile. Spheres are used as they are the simplest geometric element for projection – they are invariant concerning a projection in any direction. This simple transition is not perfectly smooth, but considering the ratio of sphere size to diameter change, the effect is usually still acceptable.

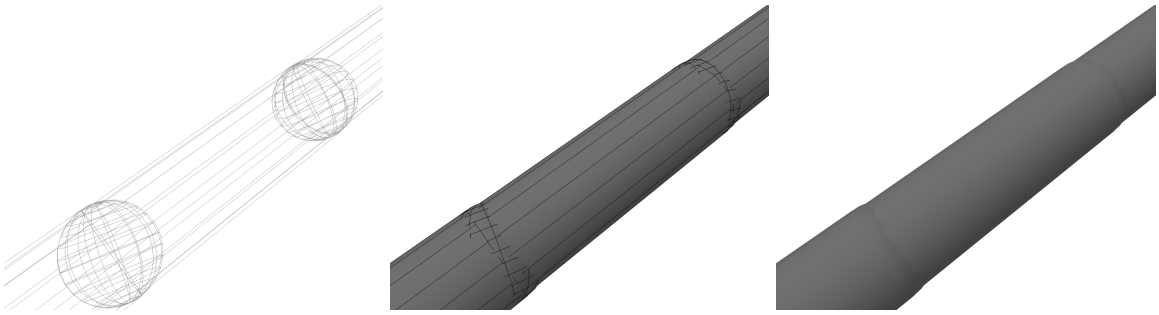

Figure 4.6: Modeling of branch with varying width.

Most branches, as shown in Figure 4.7, are modeled as sequences of cylinders in this work. This “refinement” renders a more realistic visualization of trees possible.

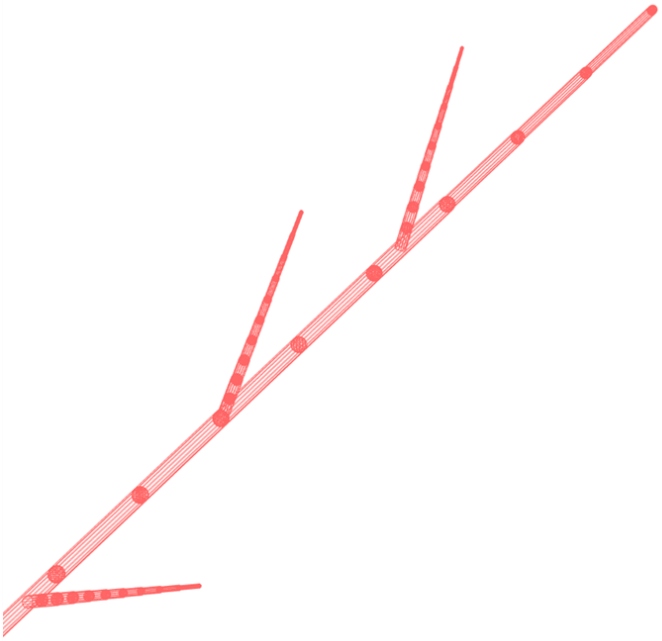

Figure 4.7: Branches modeled as sequences of cylinders in 3D.

Hierarchical Structure Improvement

A hierarchical process is employed to balance the reconstructed detail and the computational effort for modeling. First, a single cylinder is used for modeling the whole branch. The main structure is acquired, resulting in the first-level model (cf. red cylinder in Figure 4.8). Yet, some branches might not be correctly presented. In the second phase, the individual branches are modeled in more detail by sequences of cylinders. Each cylinder is adjusted to fit the real scene more precisely. As shown in Figure 4.8 (green), the Diameter of each of the component cylinders (green) in the refined model is adjusted individually, to match the local branch width.

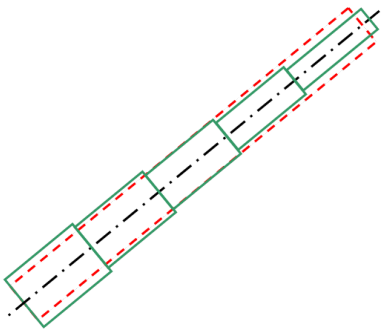

Figure 4.8: Hierarchical improvement for varying width: Initially reconstructed branch (red) and refined structure (green).

Figure 4.9 shows an implementation for a trunk. Additionally to the adaption of the width, the lower end position is adjusted to match the given model. Although the lower end of the trunk is often better determined than the upper one because of less clutter, refinement is still needed: The blue sphere indicates the initial width of the single-cylinder model and the initial position of the lower end position preliminarily determined by means of line extraction and matching (cf. Section 5.1).

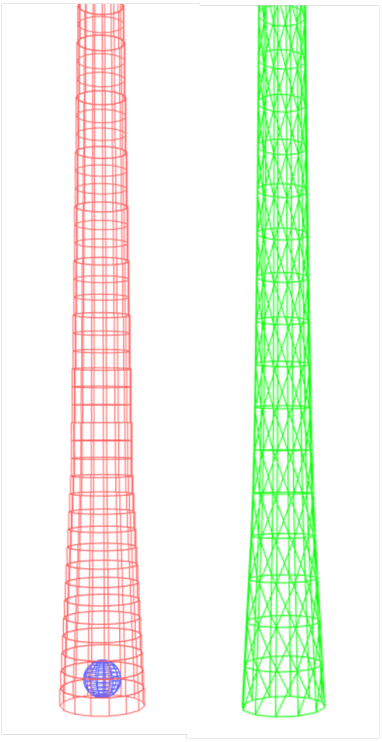

Figure 4.9: Detailed modeling of a trunk. The model is refined by a sequence of individual cylinders (left), whose widths are adjusted matching the given model (right). The blue sphere indicates the end position and width before the refinement.

4.4 Priors for Parameters

The Production Rules for the different branching types only give a general description of the natural branching process. For generative modeling (cf. Section 2.3), we thus also statistically sample the parameters which are defined for the general model, i.e., the parametrized Production Rules. Prior distributions play an important role in the sampling by encoding empirical knowledge. They guide the search and therefore reasonable samples can be produced more efficiently.

We have derived a set of generic prior distributions from basic botanical knowledge and empirical observations. The priors of this Section are used for the the first level of branches, i.e., when the extraction starts, and no further information about the target tree is available yet. For the second and further levels, the distributions for the branching angles are updated considering the predominant influence from their parents (cf. Section 4.2.3).

4.4.1 Prior for Azimuth

As shown in Figure 4.3, the Azimuth is defined as the angle around the z-axis with 0° at the x-direction. Empirically we know, that most trees have (partially-) radially symmetric crowns. Therefore, we assume that on the first level the branches are symmetrically distributed on the full circle and, as shown in Figure 4.10, a uniform distribution in the domain [0°,360°) is used for the Azimuth α as basic prior:

$$\alpha \sim U[0°, 360°) \ .$$

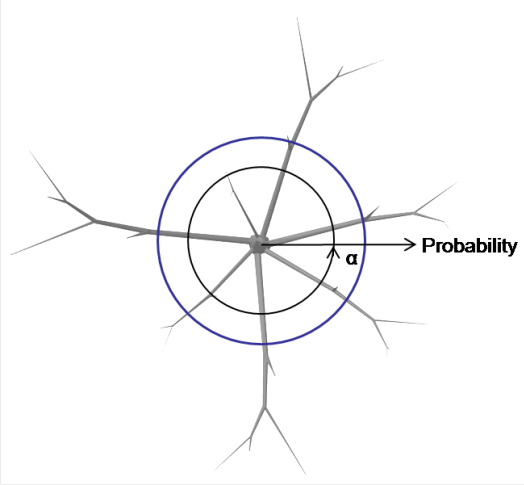

Figure 4.10: Prior distribution of Azimuth.

4.4.2 Prior for Inclination

Inclination is defined as the angle to the horizontal plane (cf. Figure 4.3). Because we use the full-circle Azimuth, the Inclination β is restricted to the half-circle, from -90° (downwards) to 90° (upwards). The 0° Inclination is defined for the horizontal plane.

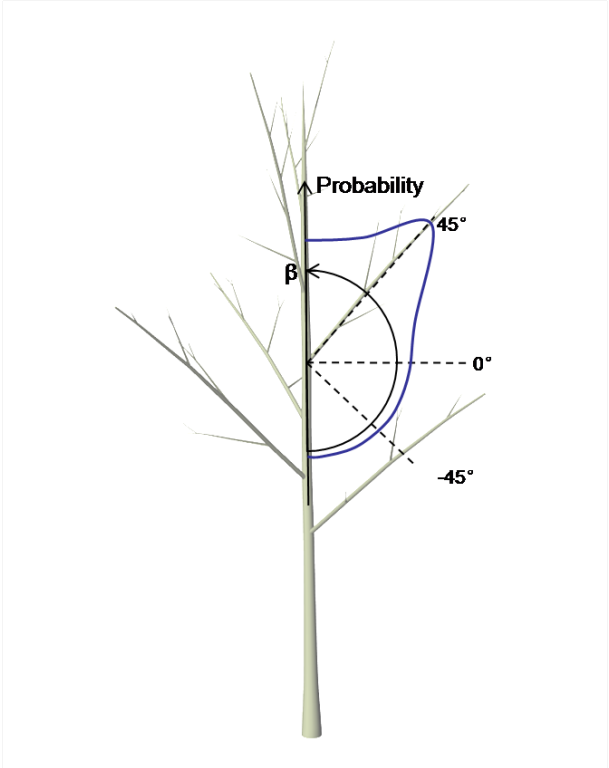

Figure 4.11: Prior distribution of Inclination.

If nothing else is known, we use a normal distribution

$$\beta \sim N(\mu_\beta, {\sigma_\beta}^2) \ ,$$

as shown in Figure 4.11, with mean $\mu_\beta = 45°$ and standard deviation $\sigma_\beta = 20°$, because for most types of trees a majority of branches points upwards due to phototropism.

4.4.3 Priors for Branch Size

We assume that the lengths of branches on the same level are independent of each other and follow a similar distribution, i.e., are iid. Without any further information, a Gaussian distribution seems to be most suitable. Mean and standard deviation are manually given for the first level of branches:

$$l \sim N(\mu_l, {\sigma_l}^2)$$

Although the diameter of a branch should be constrained by its length, i.e., a long branch should not be too thin, we assume that it follows an iid Gaussian as well, because the underlying ratio varies too much for different kinds of trees:

$$d \sim N(\mu_d, {\sigma_d}^2) \ .$$

For most examples in this work, we use the following empirically defined values $\mu_l = 1$ meter, $\sigma_l = 0.3$ meter and $\mu_d = 0.1\mu_l$, $\sigma_d = 0.2\mu_d$.

Chapter 5

Framework for Generative Statistical Tree Extraction

The branching structure of trees is hard to extract from images from different points of view because of the following difficulties:

1. Background clutter and weak contrast: The extraction can be disturbed by possibly weak contrast in the images and the background clutter from other objects, e.g., facades or other trees, as we assume an urban environment.

2. Occlusions: Because of the geometric complexity of trees, branches, which have a relatively long and thin structure, are often partially or entirely occluded by the trunk or other branches from certain points of view.

3. Invalid ordering constraint: To construct 3D models of trees, one needs to match the branches. The ordering constraint, i.e., a point left of another point on an epipolar line in one image is also left of the corresponding point on the epipolar line in the other image, employed in many cases to guide matching, is often not valid for branches even for images taken from points of view close to each other.

All this means that the bottom-up extraction and matching of branches does not seem promising and suitable constraints describing the structure of trees are essential for their 3D reconstruction. We thus decided to model the tree structure generatively and to extract the tree top-down/model-driven. The modeling is based on L-systems, which are widely used in computer graphics to simulate the structure and growth of vegetation (cf. Section 4.2). During generative modeling (cf. Section 2.3), the parameters of the model are determined by statistical sampling to fit the image sequence, which is our (only) link to the real scene. The branches are presented in 3D object space as (sequences of) cylinders (cf. Section 4.3).

L-systems provide a semantic description for trees, which ensures reasonable structures. Yet, L-systems alone only give means to generate random trees, because of the missing link to the information from real scenes. Therefore, we employ statistical optimization by maximum a

posteriori estimation – MAP (cf. Section 2.2.2), to link the 3D tree model to the real scene, i.e., to find a plausible structure for the tree visible in the images.

After the trunk has been located by line extraction and image matching (cf. Section 5.1), the generative statistical approach based on the estimated background images (cf. Section 5.2) is employed for branch extraction. Section 5.3 shows how likely candidates for branches are generated by statistically sampling parameters. The candidates are verified by comparison with real images (cf. Section 5.4) leading to a MAP estimate as presented in Section 5.5. In Section 5.6 it is shown how the trees are classified into different branching types (cf. Section 4.1) after the first level of branches have been extracted, so that specific Production Rules can be applied for a more efficient modeling. The prior distributions for the parameters are refined in a Bayesian framework based on the extracted branches (Section 5.7) and integrated into the MAP estimation.

5.1 Extraction of the Trunk

While this thesis focuses on the branching structure, a basic part of many trees we are interested in is the trunk. Here, we define the trunk as the part under the crown, or more precisely, the part from the ground to the first branch(es). For it, we extract straight lines, assuming that the trunk corresponds to a thick, mostly vertical line (cf. Section 2.6.2). It thus defines the lower part of the main axis outside the crown.

Vertical lines, i.e., hypotheses for trunks, are extracted based on the line extractor introduced in (Steger 1998), which extracts curvilinear structures from images with sub-pixel precision. Lines are extracted using an explicit model for the lines as well as their surroundings and the bias induced by asymmetrical (contrast along) lines can be removed. By knowing the vertical vanishing point and that trunks are mostly straight, we can restrict the hypotheses for trunks to approximately vertical straight lines.

These hypotheses are verified by matching them in several images. We use trifocal tensors (TFT) (Hartley and Zisserman 2004) derived from the known orientation parameters to predict from lines in two images hypotheses for lines representing the trunk in further images.

In an urban environment, there may be several other trees, tree-like objects, or even vertical structures of buildings near the target tree. I.e., multiple vertical straight lines can be extracted by image matching and all these lines are hypotheses for the target trunk. We sort out wrong hypotheses based on the assumption that the images have been taken specifically to extract the target tree and that thus all the photos have been taken convergently focusing on the target tree. This implies that the target tree appears in all images and is located not too far from the center. Lines, indicating other objects, can thus be filtered because of the following two reasons:

1. A line cannot be found in at least one image of the sequence: A line found by matching in two images is projected to all other images based on TFTs (cf. Figure 5.1) and if it is outside the image space, or cannot be found there, it is rejected.

2. A line moves largely in the images: Figure 5.2 shows the projections of hypotheses into several images. As all images were assumed to be taken focusing on the target tree, all other lines than line 0 show larger displacements from different points of view.

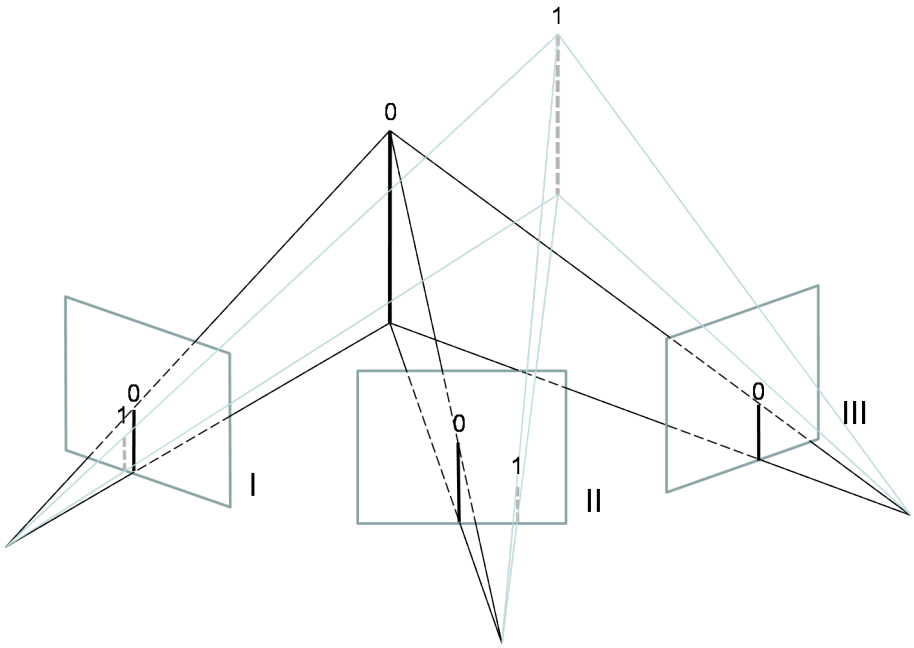

Figure 5.1: Trifocal tensor used for verification of trunk hypotheses (images I to III and lines 0 and 1).

Thus, the trunk is basically determined and then modeled as a single cylinder to provide a basis for further extraction. Yet, the extraction of line segments cannot guarantee a precise trunk. Especially, the upper end of the trunk will very likely not be at the correct position. Thus, the "growth" of branches begins by searching in the vicinity of the upper end of the trunk, to compensate for the uncertainty of the line extraction, rather than starting exactly there. After that, the trunk model is improved in the form of a cylinder sequence (cf. Section 4.3) and the lower end position of the trunk is refined at the same time.

5.2 Background Estimation

In the proposed generative statistical extraction, branch hypotheses generated by the L-system and statistical parameter sampling are projected into 2D images from all viewpoints resulting in simulated images for evaluation (cf. Section 5.4). A simulated image is composed of the projection of hypotheses and an estimated background, which reconstructs the scene except for the target tree. We use gray-scale morphology in the background estimation to remove the fine structure of trees from the foreground, and an iterative process reconstructing details from the original images which are then integrated into the background.

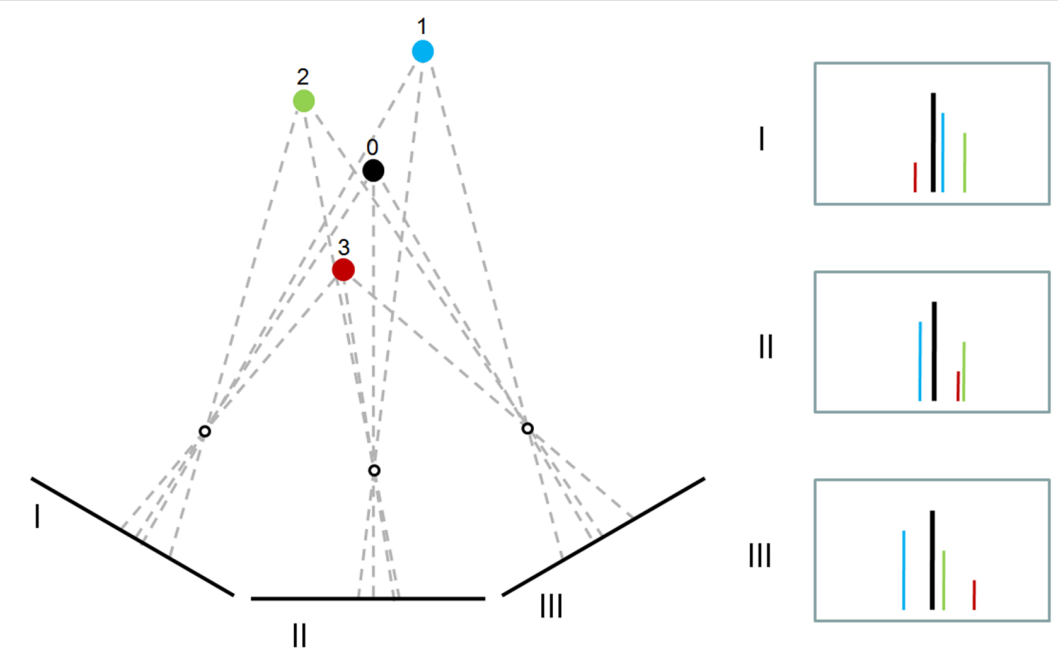

Figure 5.2: The target tree (0-black) shows a high stability concerning its location in different views, because the photos were taken convergently focusing on it.

5.2.1 Morphological Transformation

In images unfoliaged trees mostly consist of relatively thin linear structures and it can be assumed that in large parts they have a color different from the background, e.g., sky or building facades. This fine structure has to be removed to obtain an image of the background into which we then can project our hypotheses for branches. Here this is done by means of mathematical gray-scale morphology (cf. Section 2.4).

According to the characteristics of the scene, particularly the lighting conditions, both Closing (for dark trees on light background) and Opening (for light trees on dark background, e.g., night scenes) are employed.

A circular structuring, i.e., rotation invariant, element is used according to (Köthe 1996), because the branching structures consist of lines in substantially different directions. The size of the structuring element is derived from the diameter of the trunk in an image, as the trunk is always at least as thick as all branches. Figure 5.3 shows an example for background estimation using gray-scale Closing.

Unfortunately, Closing or Opening with such a large structuring element strongly blur, i.e., eliminate details of, the background. As described below, we therefore have devised an iterative recovery process to reconstruct background details.

Figure 5.3: Background estimation: original image (left) and after gray-scale Closing (right).

5.2.2 Detail Reconstruction

The basic idea is to iteratively reconstruct details from the original image to enhance the background. The basis for this is a distinction between fore- and background.

Segmentation into Foreground and Background

As shown in Figure 5.4, the estimated background is compared with the original image in a very small sub-domain D_s with a radius of 3 pixels, i.e., smaller than the width of most branches. The criterion for the segmentation is if the average absolute gray value difference inside the domain is lower than 3 gray values, used as an estimate for the image noise: if it is true, D_s belongs to the Background (Ω_B), otherwise to the Foreground (Ω_F).

Reconstructing Details

The background image is reconstructed by adding details from the original image as follows (cf. Figure 5.5):

$$g'_e(x,y) = \begin{cases} g_r(x,y), & \forall D_s \subset \Omega_B; \\ \\ g_e(x,y), & \forall D_s \subset \Omega_F \end{cases} \tag{5.1}$$

with g_e and g_r the gray value at the current center pixel of D_s in the estimated background image and the original image, respectively. The g'_e is the gray value at the corresponding pixel in the reconstructed background image. By inserting gray values from the original image for the Background Ω_B into the estimated background, fine detail is reconstructed. After Gaussian smoothing, the improved background image is used for the next iteration.

Figure 5.6 shows the original image (left), the result after gray-scale Closing (center), and after three reconstruction iterations (right). For the latter two, the histograms below show the

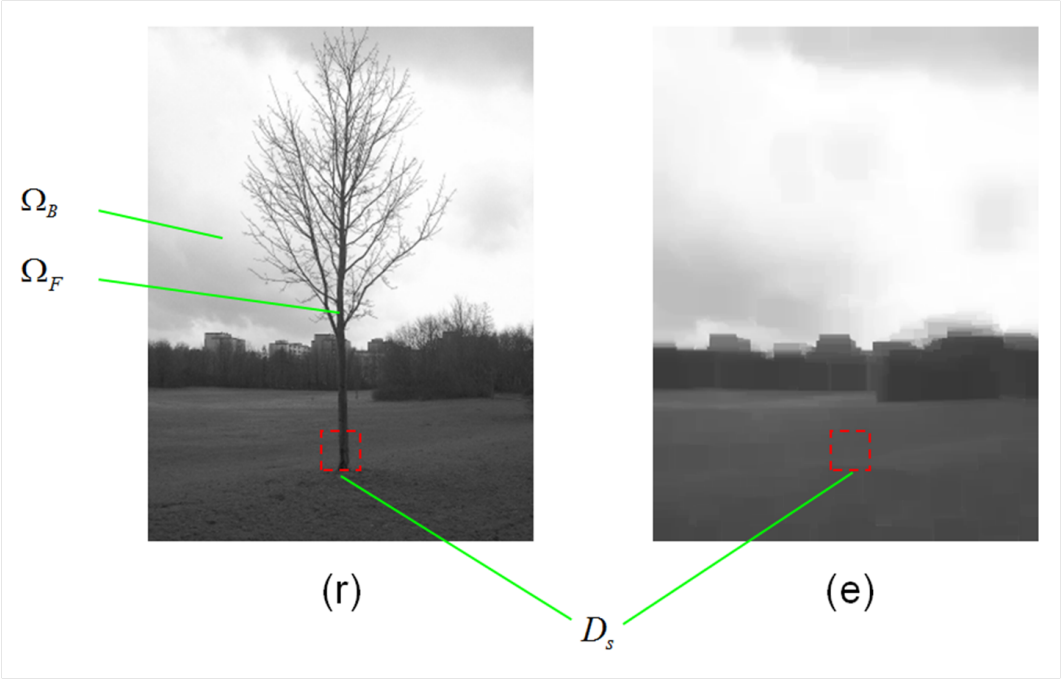

Figure 5.4: Determination of Background Ω_B and Foreground (Ω_F) by comparing the original image (r) and the estimated background (e) in a sub-domain D_s (enlarged to make it clearly visible).

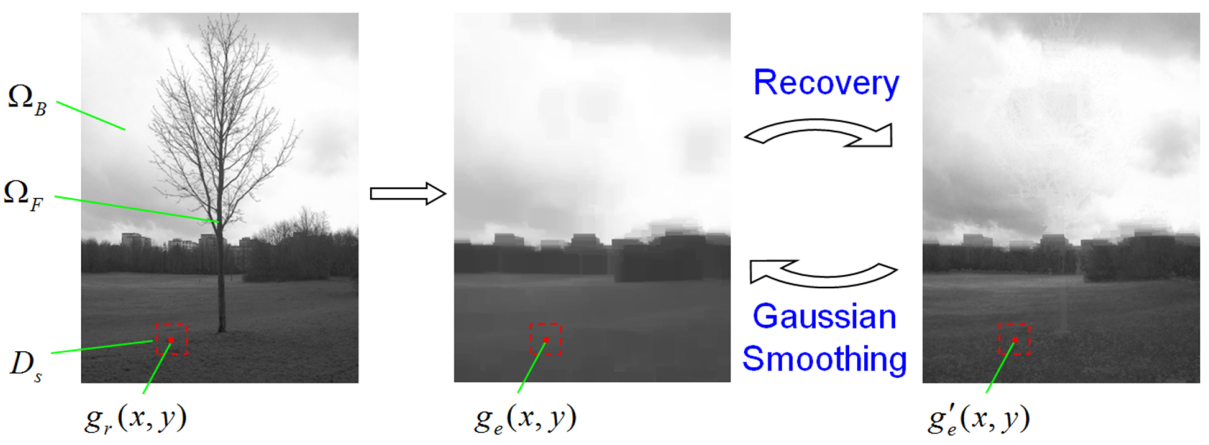

Figure 5.5: Iterative detail reconstruction by inserting gray values of pixels in the Background Ω_B from the original image $g_r(x,y)$ into the estimated image $g'_e(x,y)$.

gray value differences to the original image. In the histograms the effect of detail recovery, i.e., most small deviations are reconstructed, can be seen very clearly.

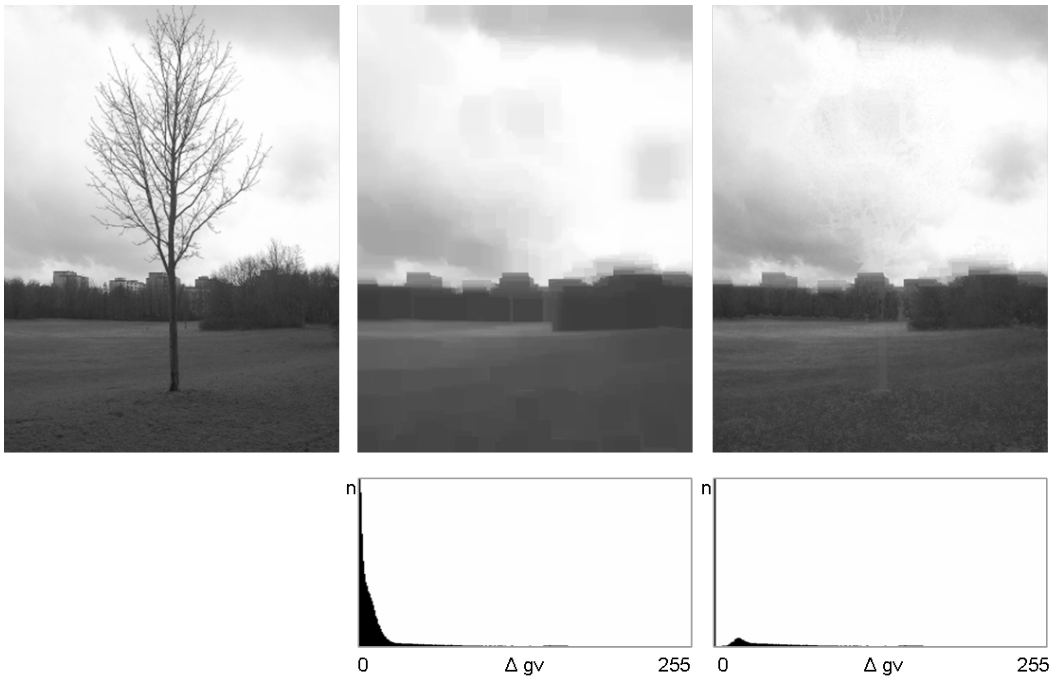

Figure 5.6: Original image (left), after gray-scale Closing (center), and after three reconstruction iterations (right). The bottom row shows the histograms of the gray value differences to the original image with the same scale for the vertical axis.

5.3 Sampling of Branch Parameters

Hypotheses for branches are generated by sampling their parameters. We have devised a combined scheme with initial plain Monte Carlo and sequential Markov Chain Monte Carlo (cf. Section 2.1), in short MC+MCMC, sampling. Additionally, a flexible way to determine the sampling sequence of the parameters is introduced. The parameters to be sampled are chosen by a proposal mechanism, which is driven by conditional probabilities considering the nature of individual parameters and their mutual influence.

Figure 5.11 illustrates the generation of branches. Based on the trunk, branches are grown by sampling their parameters Azimuth, Inclination, Length, and Diameter statistically, guided by appropriate prior distributions (cf. Section 4.4). A newly generated hypothesis is projected into the background images (cf. Section 5.2) via the given highly precisely known orientation parameters (cf. Section 2.6). The generated simulated images are matched with the given images (cf. Section 5.4).

5.3.1 Combined MC and MCMC Sampling

A combined sampling method – Monte Carlo (MC) plus sequential Markov Chain Monte Carlo (cf. Section 2.1) – is used for sampling. Since the parameters, especially the branching angles, are distributed sparsely in a large space, plain MC, i.e., random numbers are drawn

from the given prior distributions, is used in the first coarse sampling phase. The parameters are refined in the second phase by means of MCMC, i.e., the space of parameters is locally sampled.

The following procedure, as shown in Figure 5.7, was empirically found to work reasonably well: The best samples from MC, i.e., ten from one hundred, are taken as candidates for a refined search using ten MCMC iterations for each. The number of candidates is reduced to three after the first round of refinement and the best solution is finally found after another twenty MCMC iterations for each of them.

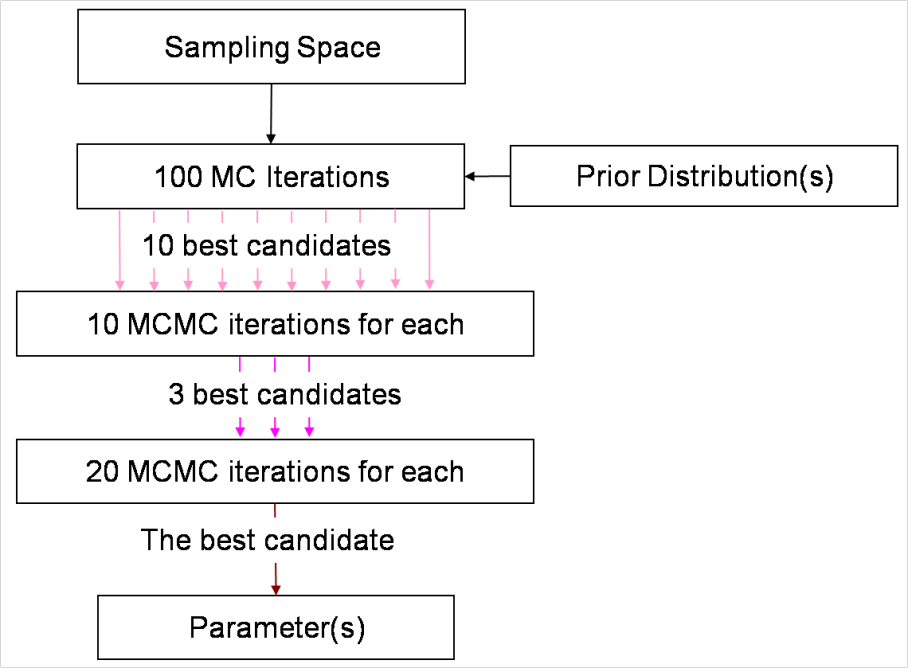

Figure 5.7: Combined MC and MCMC sampling: flow chart. The sampling is conducted for either a single parameter or a parameter combination (cf. Section 5.3.2).

Basic MC search as well as the Metropolis-Hastings algorithm (cf. Section 2.1.3) integrated into our MCMC algorithm, both help to avoid local minima while solutions are efficiently found. Figure 5.8 illustrates how local minima can be avoided by these means.

Initial prior distributions for the parameters have been devised (cf. Section 4.4). They are updated during the extraction according to the already extracted branches by means of a Bayesian refinement process (cf. Section 5.7), resulting in an efficient and plausible search.

5.3.2 Flexible Sampling Sequence

When generating a new proposal in a Markov Chain, often a combination of multiple parameters is to be sampled and these parameters are in most cases at least very slightly correlated. We empirically found that it is not efficient to change the values of all parameters at the same time or even in a fixed order. This is exemplified in Figure 5.9 for a simple two-parameter

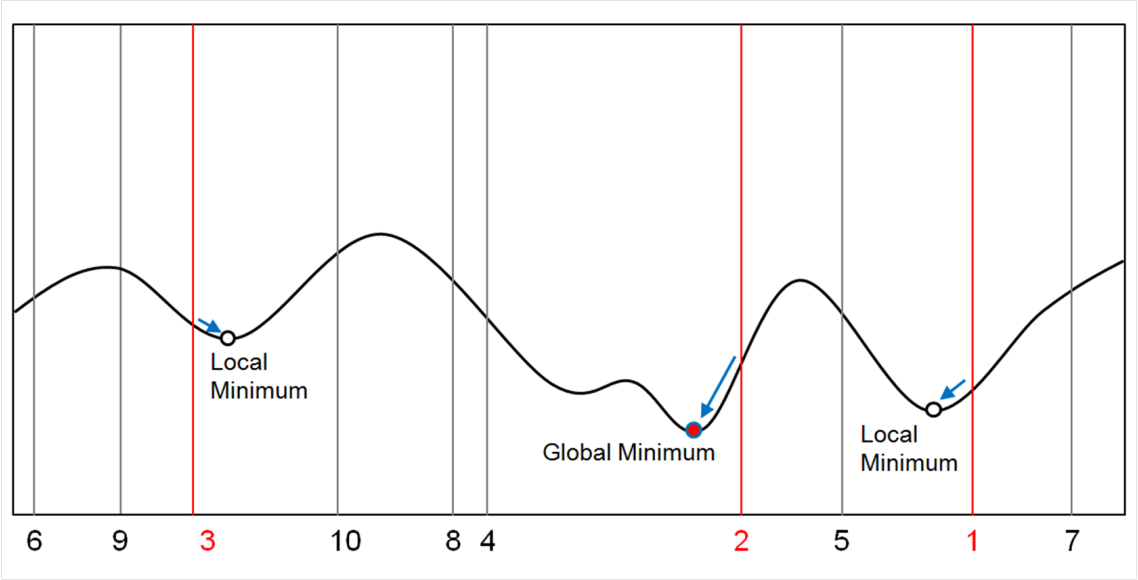

Figure 5.8: Combined MC+MCMC sampling helps to avoid local minima: Simple MCMC can very likely be trapped in local minima (empty circles). The MC step generates multiple start points for the further MCMC sampling to enlarge the probability for finding the global minimum. In this case, the best three (red) of ten MC iterations are chosen to be optimized via MCMC (blue arrows) and the position 2 (second best result after MC) finally leads to the global minimum (red filled circle).

search: If both parameters θ_1 and θ_2 are sampled independently at the same time, the quality of the proposed candidates becomes random – while one change leads to a better result, the other might lead to a worse one.

Sampling the parameters in a fixed order (branching angles jointly, length and then diameter) as proposed in Section 5.3.1 has been proven meaningful. Yet, also it can be inefficient: The sampling can be pointless even when only a single parameter is tuned while keeping the others constant. E.g., as shown in Figure 5.10 (left), given an improper angle from the previous step, varying the length will lead to almost the same evaluation in terms of overlap in the image. Or if a wrong start position for a branch is used for the determination of an angle, as shown in Figure 5.10 (right), the tuning of the angle can be meaningless.

Thus, a proposal mechanism has been devised to choose potentially meaningful candidates rather than random ones. We dynamically consider the characteristics of individual parameters and their performance in the sampling. As only the performance of the last candidate(s) is used (to either reject or accept the proposals, i.e., to stay or move forward in the search), it works in a Markov Chain style. The proposal mechanism also takes into account that for long linear structures in 3D space such as branches, the angles play an important role particularly as achieving solutions is computationally expensive.

This sampling mechanism is driven by a probability function:

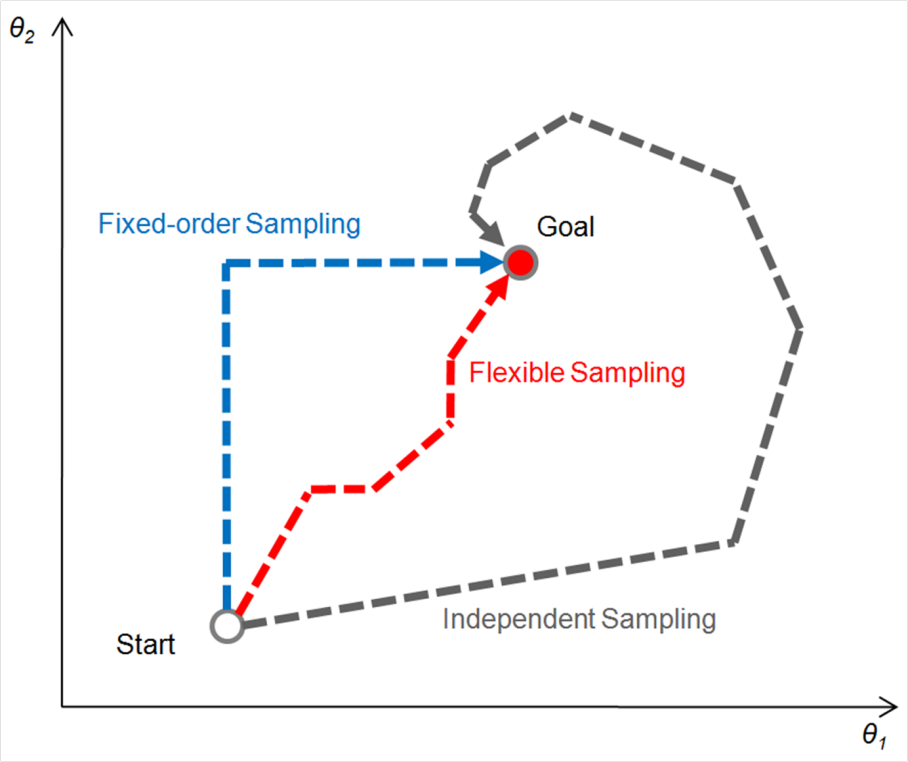

Figure 5.9: Generation of proposals in a Markov Chain. The devised flexible sampling mechanism works more efficient by considering the characteristics of individual parameters as well as their mutual influence.

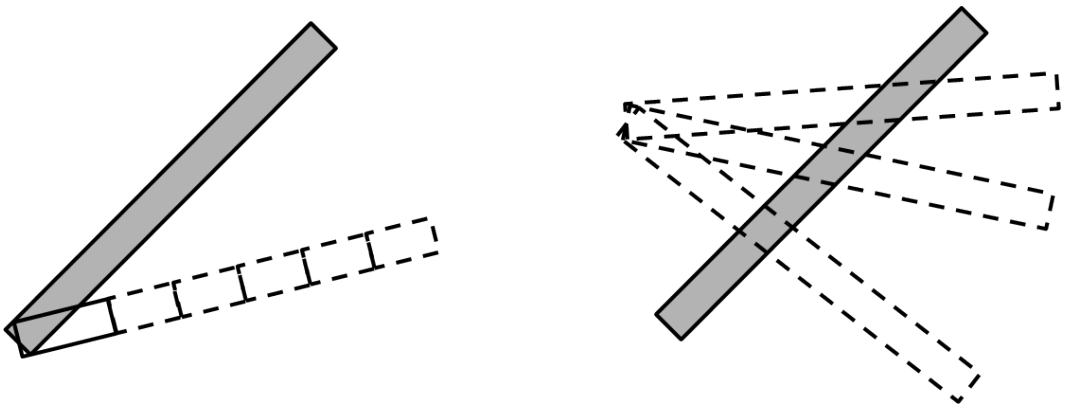

Figure 5.10: Tuning a single parameter in a fixed sampling order can be inefficient. Hypotheses: contours with dashed lines; branch in image: filled contours with solid lines.

$$g(\overline{\theta}) = g_0(\overline{\theta}) \sum_{\theta_i \in \Theta'} g(\overline{\theta} \mid \theta_i)\, g(\theta_i) \tag{5.2}$$

with $\overline{\theta}$ the current candidate, $\overline{\theta} \in \Theta$ (the global space of parameters), and Θ' indicating the space of parameters sampled in the last step, $\Theta' \subseteq \Theta$.

This probability function consists of two parts: 1) The initial probability g_0, which is derived from the distributions of individual parameters. 2) A probability conditioned on the sampling result in the last step, representing the influence of the sampling sequence.

Initial probability

The initial probability is defined in a way giving parameters with a larger search space a higher priority. Table 5.1 gives the g_0 for the parameters of branches on the first level. For parameters following Gaussian distributions, $g_0 = 2\sigma/\Omega$, where Ω indicates the solution space. For Azimuth and Inclination, the solution spaces are 360° and 180°, respectively. The Ω for Length and Diameter are approximately limited by 0 and a derived upper limit of $\mu + 3\sigma$. For parameters with uniform distribution, $g_0 = 1$.

Parameter	Distribution	Solution Space (Ω)	g_0
Azimuth	$U[0°, 360°)$	[0°, 360°)	1
Inclination	$N(45°, (20°)^2)$	(-90°, 90°]	0.22
Length	$N(1, 0.3^2)$	≈(0, 1+0.9]	0.32
Diameter	$N(0.1, 0.02^2)$	≈(0, 0.1+0.06]	0.25

Table 5.1: Initial probabilities g_0 for parameters on the first level based on their distributions.

Conditional probabilities

The parameters sampled in the last step ($\theta_i \in \Theta'$) and their performance influence the current step via a conditional probability:

$$g(\overline{\theta} \mid \theta_i) = \begin{cases} |\Theta'|^{-1} P_{erf}, & \forall \theta_i = \overline{\theta}; \\ |\Theta'|^{-1}, & \forall \theta_i \neq \overline{\theta}, \end{cases} \tag{5.3}$$

with $|\Theta'|$ the cardinality of Θ', i.e., the number of parameters that have been sampled in the last step, and P_{erf} indicating the performance of the sampling: if the hypothesis had been accepted, then $P_{erf} = 0$, otherwise $P_{erf} = 1$.

To illustrate how this works, a simplified example for the Azimuth is given in Table 5.2, with $g(\theta_i) = 1\ \forall \theta_i \in \Theta'$: For sampling four parameters, the number of all possible combinations is 15. However, assuming that all previously sampled parameters have an identical influence on the sampling for the Azimuth in the current step, this number reduces to five: Azimuth alone, Azimuth with one to three other parameter(s) and any other parameter(s) without Azimuth. Taking the performance of the sampling into account (cf. Equation 5.3), the probabilities for Azimuth for all possible cases are listed in Table 5.2.

In summary, the main effects of the proposed mechanism on the sampling sequence are as follows:

1. If a parameter has not been selected in the last step, its probability to be selected increases.

Θ'	α	α+1	α+2	α+3	no α
$g(\alpha)_{P_{erf}=0}$	0	1/2	2/3	3/4	1
$g(\alpha)_{P_{erf}=1}$	1	1	1	1	1

Table 5.2: Probabilities for Azimuth α conditioned on the selection and the performance in the last step for all possible cases. ($\alpha + 1$ means Azimuth plus one additional parameter, etc.; $P_{erf} = 0$ means that the last hypothesis had been accepted.)

2. Parameters which led to better results have the tendency to be kept fixed while other parameters are varied.

5.4 Evaluation of Branch Hypotheses

For every image the generated 3D hypotheses (cf. Section 5.3) are projected into the estimated background (cf. Section 5.2) via the given highly precisely known orientation parameters (cf. Section 2.6). This results into the simulated images, as shown in Figure 5.11 (top). Hypotheses are evaluated by comparing simulated and given images (Figure 5.11, bottom) by means of a Gaussian likelihood function (see below).

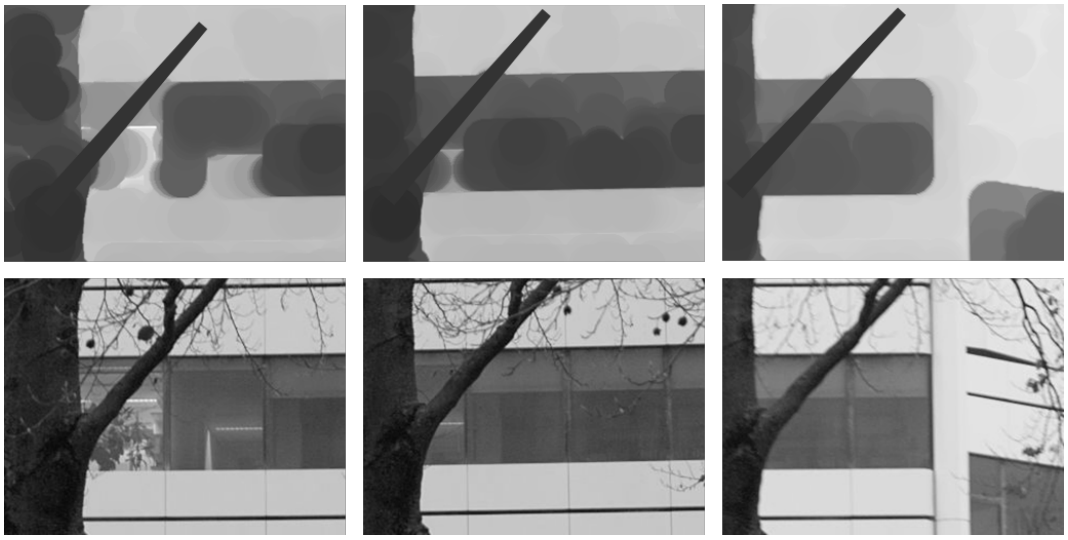

Figure 5.11: A hypothesis is projected into the estimated background for images from different points of view (top), to be compared with the original images (bottom).

5.4.1 2D Projection

We use a simple and efficient 2D representation derived from the 3D representation instead of the actual projection of the 3D cylinders. The latter would entail a larger computational effort and for statistical sampling many projections are needed. Another reason is that the

projection of the branches results into patches of nearly constant brightness anyhow: Empirical investigations have shown that the texture of the bark is hard to estimate and shading is not significant for most branches for the images we have used in our experiments.

The chosen 2D representation consists of trapezoids. A trapezoid is described by its

– direction (angle with x-axis)

– length

– width of begin

– width of end.

We determine the parameters of a trapezoid as follows: The 2D coordinates of the centers of the begin and the end are obtained by projecting the centers of the 3D circles, i.e., the end points of the axis delimiting the cylinder on both sides, into the image via

$$x' = PX \tag{5.4}$$

with (homogeneous) 3D points X, image points x', and the projection matrix P (Hartley and Zisserman 2004).

To compute reasonable approximations for the widths, we connect each end point of the axis of the cylinder with the projection center and determine the normal plane to this vector. The distance between the projections of the end point of the axis and of a point on the normal plane with distance radius of the cylinder from the axis, i.e., a circle on the normal plane, equals half the width in the image, as shown in Figure 5.12. The color is taken as average of the trunk color.

5.4.2 Likelihood Function

The likelihood function is defined as follows:

$$\theta \mapsto f(x|\theta) \tag{5.5}$$

with x the observation (image), based on which the underlying population parameter θ is to be estimated.

Often the likelihood function is written as $L(\theta|x) = f(x|\theta)$, because the likelihood is seen to estimate unknown parameters based on known observations, which is opposite to a conditional probability. To avoid confusion, we present the likelihood function in this work, always in probability form, i.e., $L(x|\theta)$ or $L(X|\Theta)$.

If a set of mutually independent random samples $X_1, X_2, ..., X_n$ has been drawn, the observed random variables $x = x_1, x_2, ..., x_n$ are also mutually independent and the joint probability

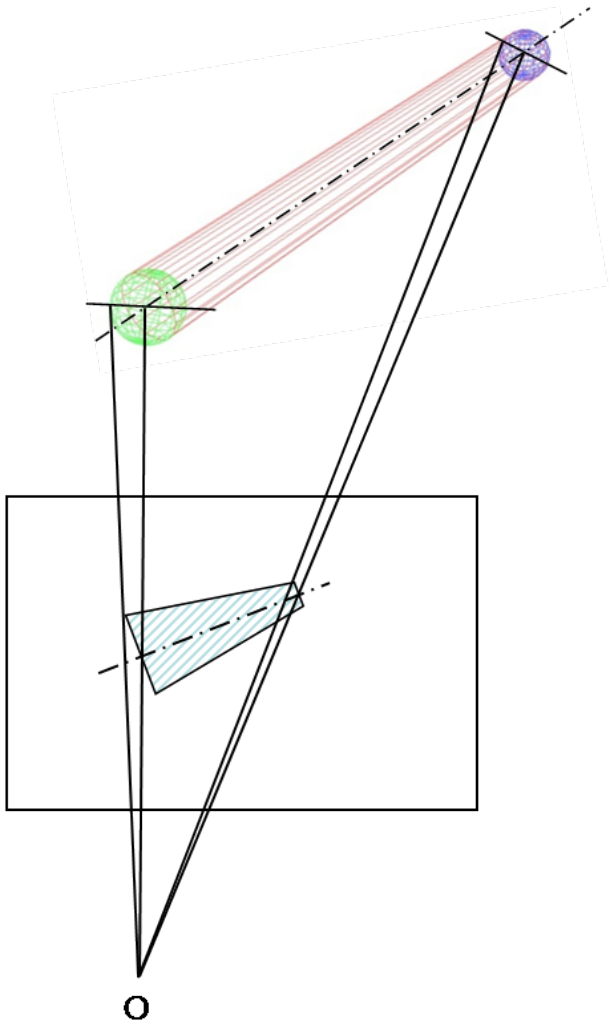

Figure 5.12: Determination of the widths of the projected ends of a branch.

density function (PDF) is therefore equivalent to the product of the marginal PDFs. The likelihood function can then be expressed as

$$L(x|\theta) = P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n|\theta) = \prod_{i=1}^{n} P_{X_i}(x_i|\theta)\,. \tag{5.6}$$

For the evaluation of the hypotheses we follow (Tu and Zhu 2006). We let $\mathcal{D}$ indicate the domain of comparison and $I_s(x, y)$ and $I_o(x, y)$ the intensities of a pixel (x, y) in the simulated and the original image, respectively. Assuming that the difference for each pixel (reconstruction residual) follows an iid Gaussian $N(0, \sigma^2)$, with σ^2 the variance of the intensity noise, the likelihood function L can be expressed as:

$$L(\mathcal{D}) = \prod_{(x,y)\in\mathcal{D}} N(I_s(x, y) - I_o(x, y), \sigma^2) \tag{5.7}$$

5.5 MAP Estimation

One of the difficulties for tree extraction in urban scenes is the high uncertainty of the appearance caused by the complex structure of urban spaces leading to background clutter. The reliability of the likelihood function defined above, which only takes into account the 2D representation of hypotheses, is therefore limited. We thus employ maximum a posteriori (MAP) estimation (cf. Section 2.2.2) for the parameters' optimization with the priors substantially stabilizing in the decision process:

$$\widehat{\theta}_{MAP}(x) = arg_{\theta}max\Big\{f(x|\theta)p(\theta)\Big\}\,. \tag{5.8}$$

$p(\theta)$ in the objective function is the summarized prior information for all parameters:

$$p(\theta) = \prod_{\theta_i \in \Theta} p(\theta_i)\,. \tag{5.9}$$

The priors are combined by multiplication as we assume that they are basically independent of each other. Please note that the priors introduced in Section 4.4 are updated during the extraction process by Bayesian refinement, introduced later in Section 5.7, based on already accepted branches.

The MAP estimation can therefore be formulated as:

$$\widehat{\Theta}_{MAP} = arg_{\Theta}max\Big\{L(\mathcal{D})\prod_{\theta_i \in \Theta} p(\theta_i)\Big\}\,. \tag{5.10}$$

In MAP estimation, thus not only the appearance of the hypotheses, but also the probabilities of their parameters are taken into account. By integrating prior information, hypotheses with reasonable parameters have a higher probability to be accepted.

Please note, that in the optimization, the multipliers, $L(\mathcal{D})$ and $p(\theta_i)$ in the objective function do not have to be normalized, as the criterion of acceptance is:

$$\frac{L(\mathcal{D})\prod\limits_{\theta_i \in \Theta} p(\theta_i)}{L'(\mathcal{D})\prod\limits_{\theta'_i \in \Theta} p(\theta'_i)} = \frac{L(\mathcal{D})}{L'(\mathcal{D})} \cdot \frac{p(\theta_1)}{p(\theta'_1)} \cdot \quad \cdots \quad \cdot \frac{p(\theta_n)}{p(\theta'_n)}\,. \tag{5.11}$$

I.e., the normalization term for the individual multiplier does not influence the decision. Another advantage of using multiple products in the objective function is that unreasonable hypotheses can be filtered out very early, as their influence on the result is strong.

The MAP estimates can be computed by 1) combined MC+MCMC sampling (cf. Section 5.3.1) and 2) a selection mechanism controlling the sampling sequence of the parameters (cf. Section 5.3.2).

5.6 Classification of Branching Types

We have divided branching structures into three types and devised particular L-systems (Production Rules) for every type (cf. Sections 4.1 and 4.2). Before the extraction starts, however, it is usually unknown which type the target tree belongs to.

If no prior knowledge is available, we classify the target tree into the most similar branching type by analyzing the layout of the first level branches after extracting them. Then, specific Production Rules are used for a more plausible and efficient extraction.

5.6.1 Relaxed Search and Vertical Shift

The extraction of the first level branches defined as those that grow directly from the trunk begins without knowing the branching type. Thus, there is no constraint for organizing the search for the branches, such as, that for m-trees, multiple branches should grow from the same position on the trunk. We, therefore, employ a relaxed search for this level.

Particularly, the begin point of the first level branches, particularly the vertical shift of their joint position along the trunk, is added as an additional parameter (cf. Figure 5.13). This parameter is sampled together with the other parameters. This way, branches do not have to start exactly at the determined end point of the trunk, but are allowed to grow in its vicinity.

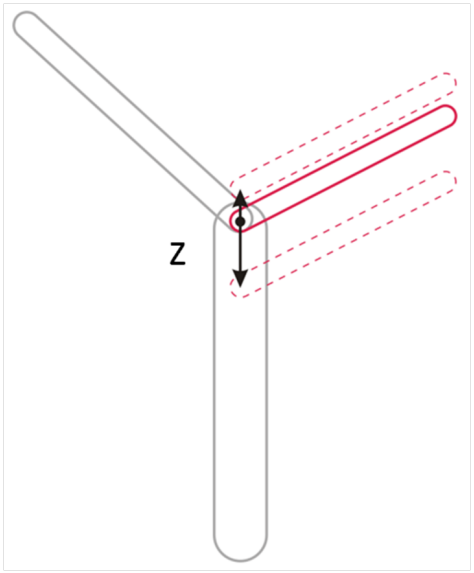

Figure 5.13: Additional parameter Z describing the vertical shift for a flexible begin point of first level branches.

The main advantage of the relaxed search is that it makes it possible to compensate for errors in the determination of the length of the trunk. As the trunk is extracted by line extraction and image matching (cf. Section 5.1), its length can only be roughly determined. By adding the additional parameter, the trunk does not need to end exactly where the first branch appears.

The proper branching type is derived based on the layout of the branches extracted with the relaxed search.

5.6.2 Classification Criteria

As shown in Table 5.3, we use two Criteria, which are independent, i.e., can be checked in parallel, to distinguish the branching types:

Branching Type	Criterion 1	Criterion 2
monopodial (m)	$\Delta z_{max} \leq \Delta z_{sm}$	$(\alpha_{trunk} - \alpha) \leq \Delta\alpha_m$
sympodial-dichasium (sd)		$(\alpha_{trunk} - \alpha) > \Delta\alpha_m$
sympodial-monochasium (sm)	$\Delta z_{max} > \Delta z_{sm}$	–

Table 5.3: The two Criteria used for the classification of branching types.

1. According to Criterion 1, trees are supposed to be of the sympodial-monochasium – sm type if for the maximal difference of the Z-coordinates holds $\Delta Z_{max} > \Delta Z_{sm}$, an empirically chosen threshold. This means that not all joint positions of the extracted branches are concentrated in a relatively small area (cf. also Figure 5.14).

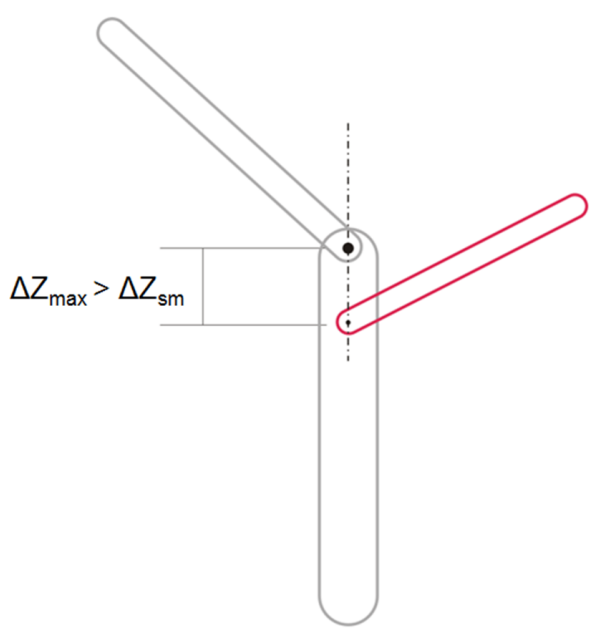

Figure 5.14: Classification of branching types: sm type according to Criterion 1.

2. Criterion 2 pertains to the deviation angles of the first level branches from the trunk direction. If for one of them holds $(\alpha_{trunk} - \alpha) \leq \Delta\alpha_m$, with α_m again an empirically chosen threshold, trees are classified as monopodial – m type (cf. Figure 5.15). It means that there is a branch which follows the direction of the trunk. Otherwise, the

branches are assumed to be more suitably described by the sympodial-dichasium – sd type.

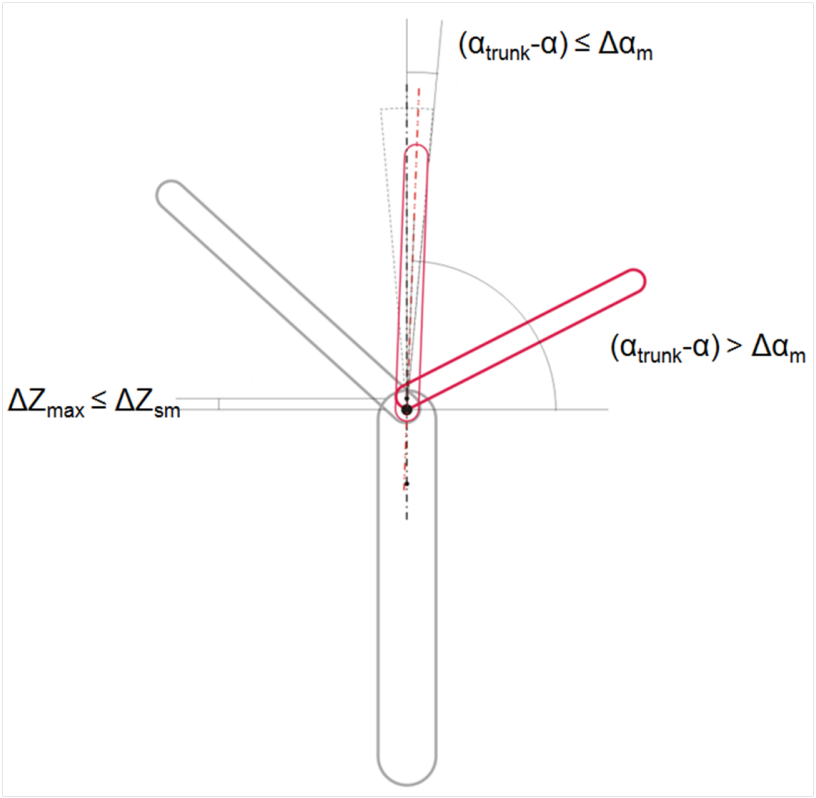

Figure 5.15: Classification of branching types: m type versus sd type according to Criterion 2 while Criterion 1 does not hold.

For the further levels the appropriate Production Rules and priors of the corresponding L-system (cf. Sections 4.1, 4.2, and 4.4) are employed.

5.6.3 Botanical versus Geometrical Classification

The classification used up to this point is based on the botanical features of trees. The main difference between the branching types is their development, i.e., the way of generating branches. This is encoded into the different Production Rules of the L-system. Yet, for tree extraction for real scenes we actually focus on the existing geometry of trees rather than the rules behind their growth as the latter cannot be observed in the images.

Particularly, the classification Criterion 1 described above implies the possibility to generalize the branch organization according to what can actually be observed in images: The difference between the m- and sm-type in the given case is just if a single or multiple branches are allowed for one growth node. E.g., if the growth nodes of the branches of an sm-tree are found very near to each other, it looks like an m-type (Figure 5.16, left). Inversely, if

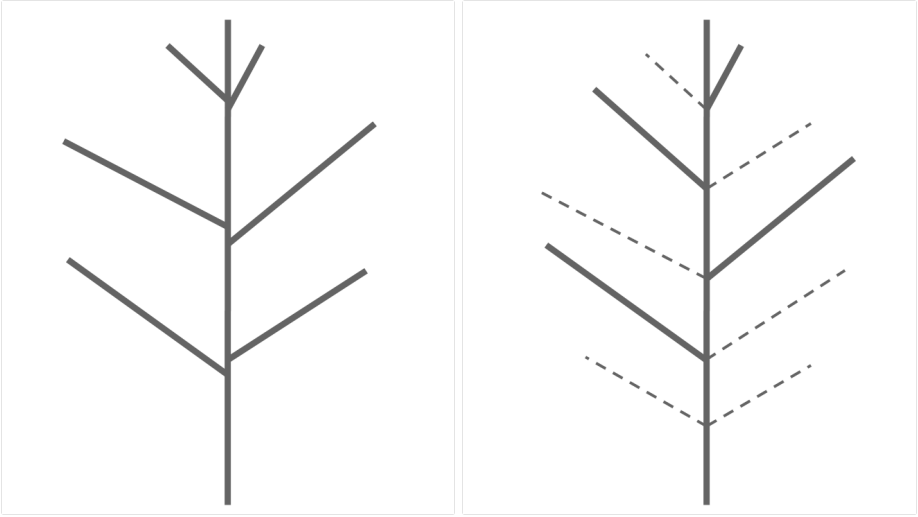

Figure 5.16: Example of an sm-tree with growth nodes very near to each other which looks like an m-tree (left) and an m-tree with missing branch (dashed lines) which resembles an sm-tree (right).

the branches of an m-tree are extracted only sparsely (cf. Figure 5.16, right), its structure resembles an sm-type tree.

In practice, the appearance of trees shows a high diversity and mixed, unclear structures can often be found. Considering the errors possible in extraction, trees in cluttered city scene can thus be falsely classified. That the sm-type can roughly be seen as a special case (sparse version) of the m-type has the following consequences: If an m-tree is recognized as an sm-type, some branches could very likely be missed. In the reverse situation, meaningless search will be conducted. Considering the diversity of tree structures and the possible errors in the search, a more generalized type of classification can be helpful.

We thus propose a new classification as shown in Figure 5.17: Both m- and sm-type are summarized into a generalized type named “mono-axial”. We use the synonymous word “axial” instead of “podial” here, because the former is a geometrical term. From the botanical point of view, the central axis of an sm-tree consists of several individual components that are defined on different levels and, therefore, it is different from that of the m-type in spite of that they are geometrically the same. Yet, as we can only observe and reconstruct geometrical characteristics, this difference is of no interest here.

The sd-type is extended from “dichasium” to “pleiochasium”, which means that pleio- (i.e., multiple-) branching is allowed. This provides more flexibility for many other tree structures as shown in Figure 5.17 (top right).

5.7 Bayesian Refinement of Priors

Prior information for the model can be derived from scientific knowledge, but also from previous empirical evidence. In this work, we make use of both: 1) Basic botanical knowledge

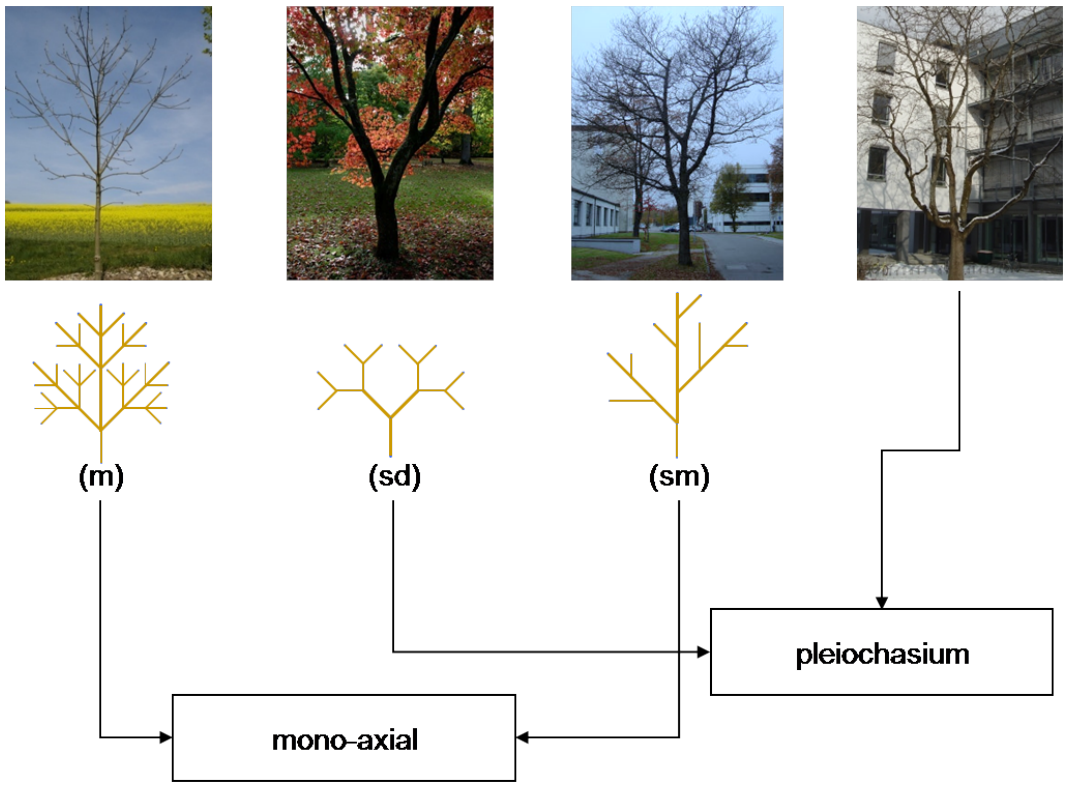

Figure 5.17: Geometrical classification of branching types.

is encoded for parameter estimation; 2) Already verified hypotheses for branches provide reliable evidence.

The basic prior distributions introduced in Section 4.4 are employed at the beginning of the sampling. Additional evidence can make the sampling more reasonable and efficient. Particularly, we make use of the evidence gained during extraction and improve the priors based on extracted (intermediate) results.

Ideally, the priors for the parameters could be learned based on the extracted trees, possibly conditioned on specific species. In practice, however, despite of the same genes, i.e., underlying rules for growth, even trees of the same species can show a very different appearance because of, e.g., different climate, lighting conditions, and growth space competition. Priors can only be meaningfully learned if the trees live in similar environment and a large number of example trees has been extracted. The former cannot be ensured in city scenes due to the strong influence of dense built-up areas on temperature and illumination. The latter is promising, but entails a lot of effort.

We, therefore, propose a "local" learning/adaptation of priors, which only considers the target tree and refines its priors based on already found branches. The refined priors are then used for the extraction of further branches of the tree. This leads to more plausible results and restricts the further search, thus making the sampling more efficient. Please note that the refined priors are not suitable for other trees, because they are inferred under possibly very special conditions.

The prior refinement is conducted in a Bayesian framework. In Bayesian learning, the degree of belief in a hypothesis tends to become more specific (rather higher or lower) by frequently using Bayes' theorem while accumulating evidence.

According to the statistical inference terminology, "population" refers to the branches about which we want to infer and "evidence" (or "observation" or "sample") to the subset of the population that has been observed. The goal is to learn about the statistical characteristics of the population, i.e., the branching parameters, from the evidence. Based on the given priors (cf. Section 4.4), we use Bayesian inference to update the probability density function (PDF) for parameters based on new evidence.

Denoting the evidence by E, the general inference function is

$$p'(\theta|E) = \frac{p(E|\theta)p(\theta)}{p(E)} \,. \tag{5.12}$$

$p(E)$ is thus the marginal probability of E

$$p(E) = \sum p(E|\theta_i)p(\theta_i) \,. \tag{5.13}$$

We denote the result of the inference by $p'(\theta|E)$ instead of $P(\theta|E)$. This is because in this refinement framework we prefer to call it updated prior rather than posterior.

Prior refinement must consider different relationships between parameters, as described in Section 4.2.3. An overview of the relationships can be found in Table 4.1. For context-independent parameters, e.g., Inclinations of branches inside one level, Bayesian refinement is used to integrate new evidence. For context-sensitive parameters, i.e., on different branching levels, inheritance and a decay model for the growing space, simulating the influence of the parent and the competition in the neighborhood, respectively, are employed.

5.7.1 Context-independent Parameters

We assume that context-independent parameters follow iid Gaussians. The initial prior $p(\theta_0)$ is refined by new evidence as follows:

$$p(\theta) = p(\theta \mid \theta_0)p(\theta_0) = N(\theta_i, \sigma_i^2)N(\theta_0, \sigma_0^2) \,. \tag{5.14}$$

After multiple evidence (employing weight w_i for each of them) has been integrated, the refined prior can be expressed as

$$p(\theta) = \prod_{i=0}^{n} w_i \, N(\theta_i, \sigma_i^2) \,. \tag{5.15}$$

We give the first evidence the weight $w_1 = 1$. The weight of other evidence is scaled according to the likelihoods $w_i = L_i/L_1 \cdot w_1$, i.e., $w_i = L_i/L_1$. $w_0 = 2$ is used to give the prior value a slightly stronger weight. After normalization, this means

$$\mu_\theta = \frac{\sigma_\theta^2 \sum_{i=0}^{n} \frac{w_i \mu_i}{\sigma_i^2}}{\sum_{i=0}^{n} w_i} \; ; \quad \sigma_\theta^2 = \frac{\left(\sum_{i=0}^{n} \frac{w_i}{\sigma_i^2} \right)^{-1}}{\sum_{i=0}^{n} \frac{1}{w_i}} \; . \tag{5.16}$$

An example for the refinement of the Inclination is given in Figure 5.18.

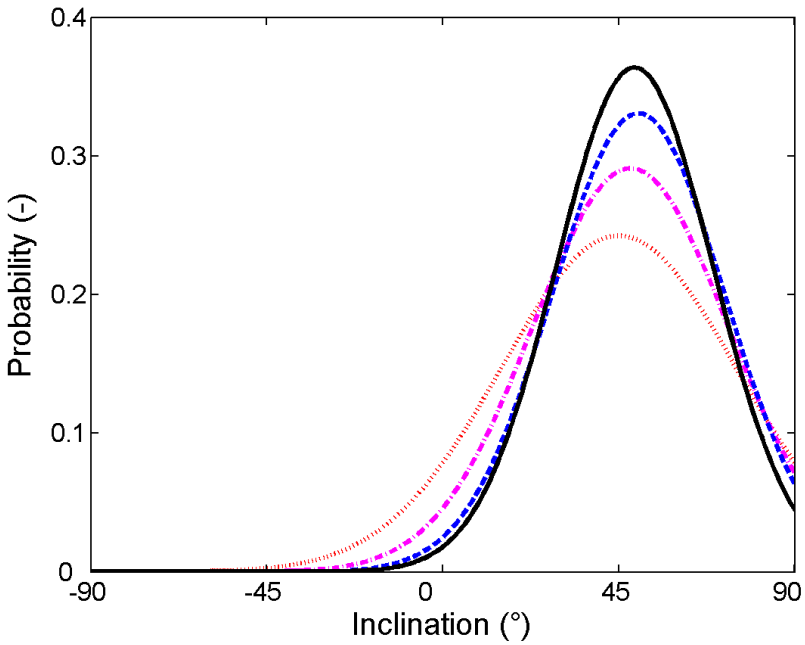

Figure 5.18: Prior distribution for Inclination and its refinement: dotted line (red) – initial prior; dash-dot line (magenta) – after first branch; dashed line (blue) – after second branch; solid line (black) – final prior after three iterations.

5.7.2 Context-sensitive Parameters

For the k^{th} level of branches, with $k > 1$, the context-sensitive Azimuths and Inclinations are assumed to obey Gaussians around the values of their predecessor branches j

$$\begin{aligned} \alpha_{j,k} &\sim N(\alpha_{j,k-1}, \sigma_{\alpha,k}^2) \; ; \\ \beta_{j,k} &\sim N(\beta_{j,k-1}, \sigma_{\beta,k}^2) \; . \end{aligned}$$

Initial values are inherited from the parent branches without explicitly considering phototropism or gravity. In the upper levels, the density of branches increases, restricting the space for each branch. Assuming that branches grow exponentially

$$N_{k+1} = N_k e^{\lambda} \tag{5.17}$$

with N_k the number of branches on the k^{th} level and λ the growth constant, the sequential reduction of the space Φ for an individual branch on level k can be formulated by an exponential decay

$$\Phi_k = \Phi_0 e^{-\lambda k} \tag{5.18}$$

with Φ_0 indicating the initial space and $-\lambda$ the decay constant. The latter can be derived from the L-system. For a typical sd-type tree, $N_1 = N_0 e^{\lambda} = 2N_0$, and thus λ is $\ln 2$. For sm- and m-trees, λ is $\ln 3$ according to their Production Rules.

As the standard deviations of the angle distributions imply the possible growth space, they follow the derived exponential decay:

$$\sigma_{\alpha,k} = \sigma_{\alpha,0} e^{-\lambda k} \; ; \quad \sigma_{\beta,k} = \sigma_{\beta,0} e^{-\lambda k} \; . \tag{5.19}$$

Chapter 6

Experiments

With the experiments we demonstrate the potential, but also the shortcomings of the proposed generative statistical approach for tree extraction. We test our approach on both simulated model and real scenes. The input data are wide baseline image sequences, which consist for the simulated 3D models of snapshots made while rotating them and for real scenes of photos taken unconstrained with hand-held consumer cameras. The output is the reconstructed branching structure represented in the form of VRML models (cf. Section 4.3).

Current approaches such as (Chen et al. 2008) and (Tan et al. 2008) aim at the efficient generation of realistic looking trees based on significant user input, particularly in terms of the basic tree structure. Opposed to this, we want to demonstrate the strengths and weaknesses of the devised modeling for trees and their extraction in a fully automatic way with minimal human intervention (cf. Section 6.3).

First, an experiment on a simulated model is described in Section 6.1. This makes it possible to evaluate the accuracy of the reconstructed results, as ground-truth data is available. Section 6.2 shows several reconstructions based on real scenes, demonstrating that problems, e.g., with background clutter and weak contrast, can also be handled. Assessment and discussion in Section 6.3 conclude the chapter.

6.1 Tests with Simulated Tree Models

The first experiment uses a simulated model of a sympodial-monochasium (sm-) tree generated manually with the software $Xfrog^{TM}$. As input images, six snapshots were taken while rotating the 3D model unconstrained. Figure 6.1 shows an overview over the tree extraction process.

Figure 6.2 shows the original (top) and the reconstructed (bottom) model from different viewpoints. Most branches, even those completely occluded in some views, have been reconstructed. Figure 6.3 presents a comparison with the ground-truth model from the front and the top, and more important, from an additional direction of view (bottom left). The latter is an extrapolation far from the given snapshot positions (cf. Figure 6.8) and shows

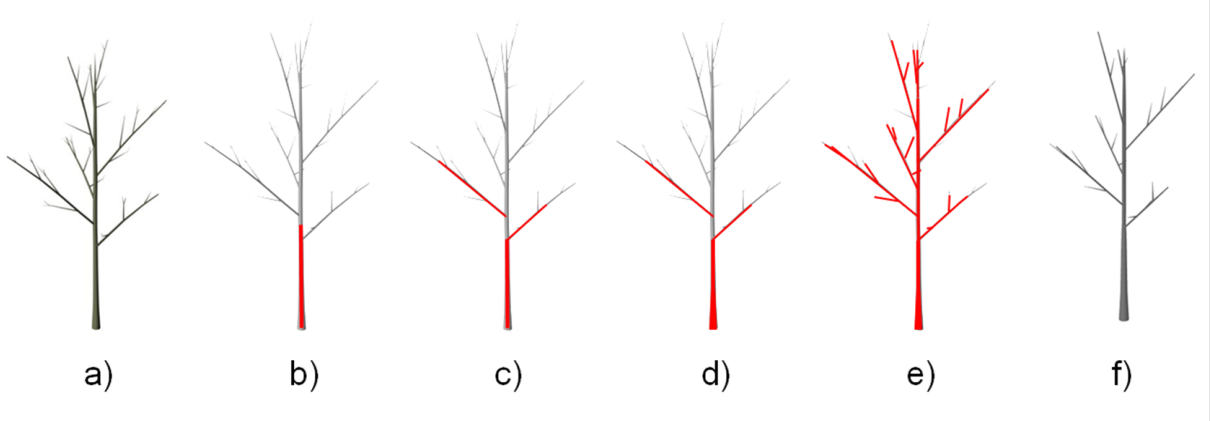

Figure 6.1: Tree extraction process overview: a) One of the input images. b) Locating the trunk by line extraction. c) Extraction of first level branches and determination of the exact upper end of the trunk. d) Detailed modeling of trunk. e) Further extraction of branches. f) Reconstructed model from the same view point as a).

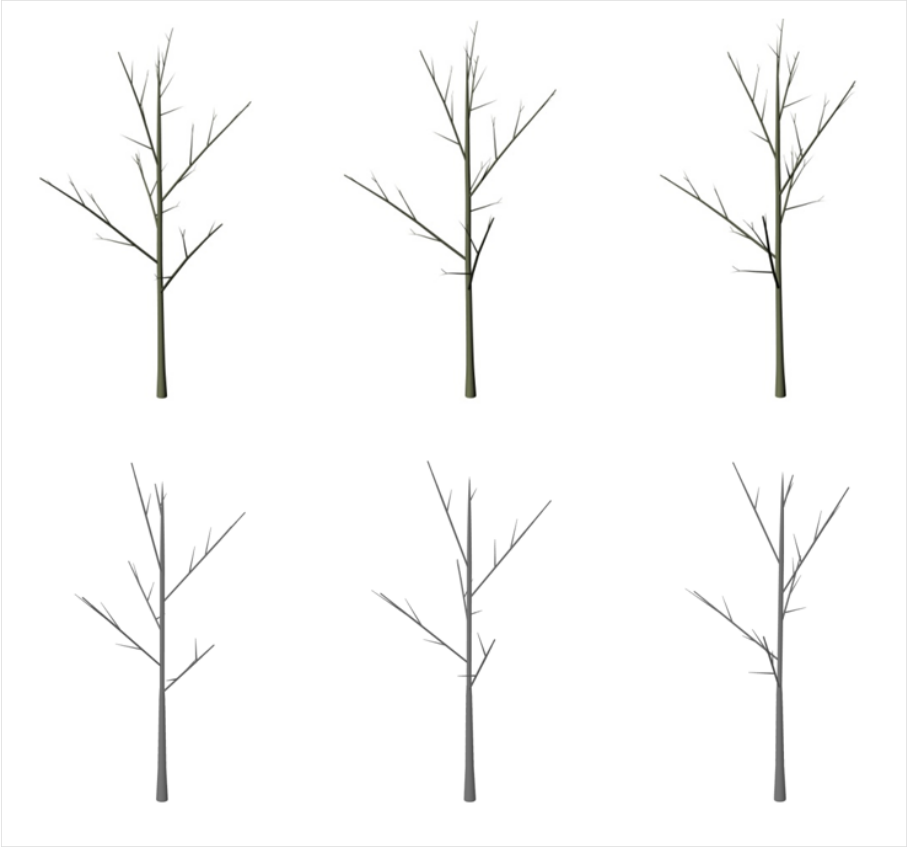

Figure 6.2: Images of model of a sympodial-monochasium (sm-) tree generated by $Xfrog^{TM}$ from three of six different viewpoints (top) and result from the same viewpoints (bottom).

that although the input images cover a relatively narrow angle of view (about 45°), the main branches were determined with correct branching angles.

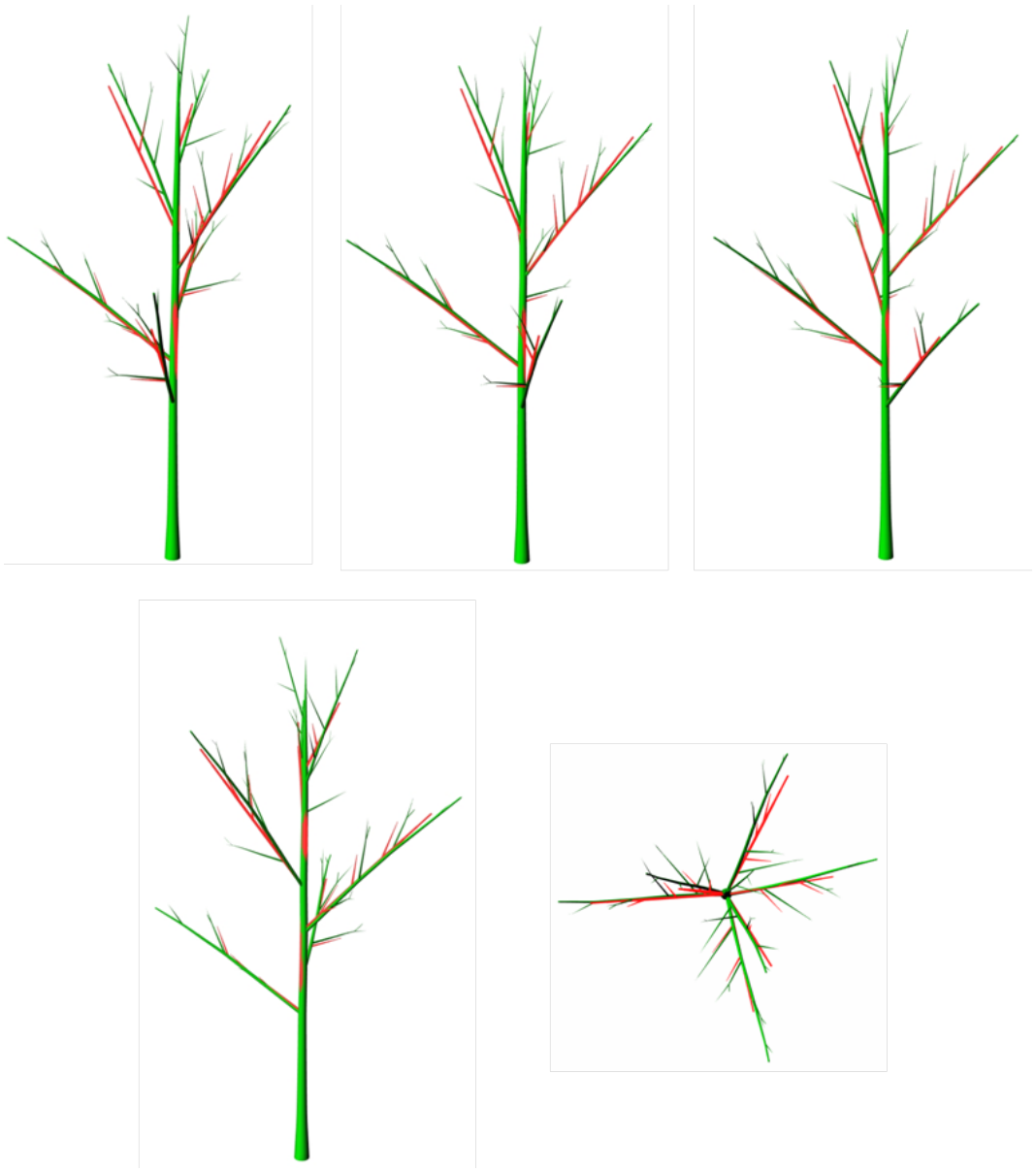

Figure 6.3: Comparison of simulated (green) and reconstructed (red) model from three of six different viewpoints (top), one other viewpoint (bottom left), and the top (bottom right).

In our generative model, L-systems are employed because of their mathematically elegant form and their suitability for this application. For a computer graphics tool such as $Xfrog^{TM}$ with emphasis on realistic visualization and ease-of-use, however, directly working with L-systems is complicated and inefficient. Although L-systems are also the basis of $Xfrog^{TM}$, the latter works on an abstraction level distinctly above the L-systems and its models do not directly follow any L-system, which can be compared with those of our work. We, therefore, treat $Xfrog^{TM}$ models with respect to the L-systems in the same way as real trees without using any extra prior information, e.g., the tree type or particular Production Rules.

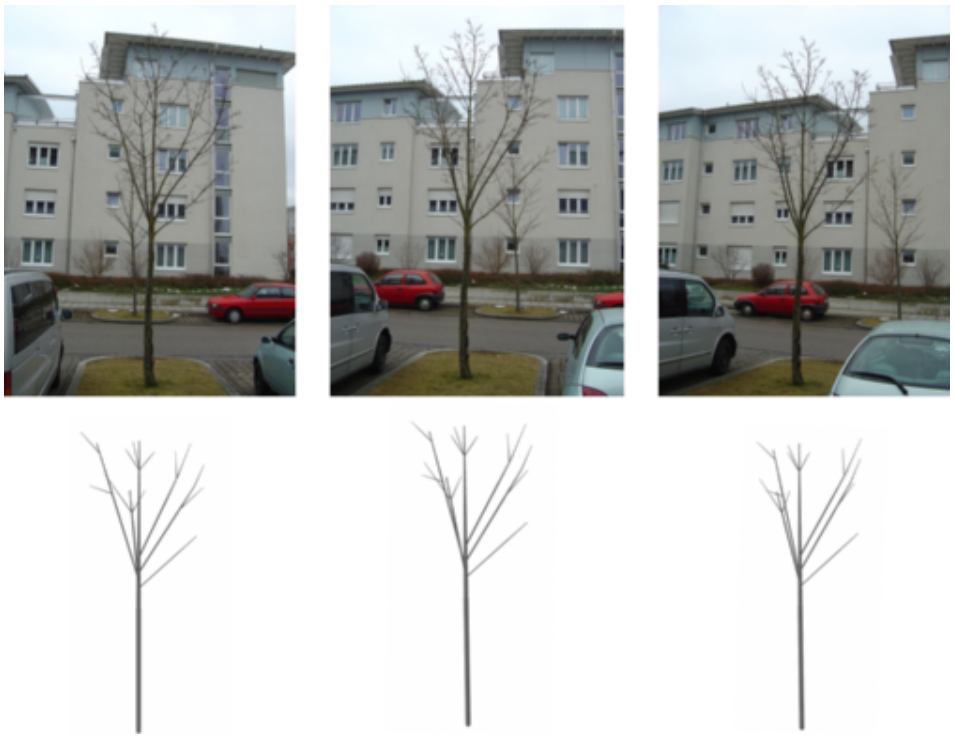

Figure 6.4: Result for a young monopodial (m-) tree – three of six images (top) as well as the reconstructed model seen from the same points of view (bottom).

6.2 Experiments for Real Scenes

For real scenes, we employ wide-baseline image sequences as source data. The photos are taken unconstrained with hand-held consumer cameras with about 5 Mega pixels.

Figure 6.4 presents in the top row three of six images showing a young monopodial (m-) tree. Below, the reconstruction result is presented from the same viewpoints. Although tiny twigs could not be extracted, the 3D model consists of most of the main branches and basically represents the characteristics of the target tree. One can notice that in spite of the occlusion (Figure 6.4, middle) and varying order (see also Figure 1.1) of the left two branches, they are correctly found and reconstructed.

Please note that the new "mono-axial" type is applied here instead of typical m-type (cf. Section 5.6.3) to produce a better result, as trees in real scene often show a mixed structure.

In Figure 6.5, for the scene introduced in Fig 6.4 a rough model of the surrounding objects including background buildings is manually given as a reference. The cameras are shown as green pyramids with the tips of the pyramids symbolizing the projection centers and the bases the directions.

Figure 6.6 presents an aged tree (sm-type). Two difficulties of extracting aged trees are the complex geometry of the (main) branches and a large number of small twigs. Again, the input sequence consists of six images and the majority of the branching structure has been reconstructed. Here, one can see the limitations by using only cylinders for the branches:

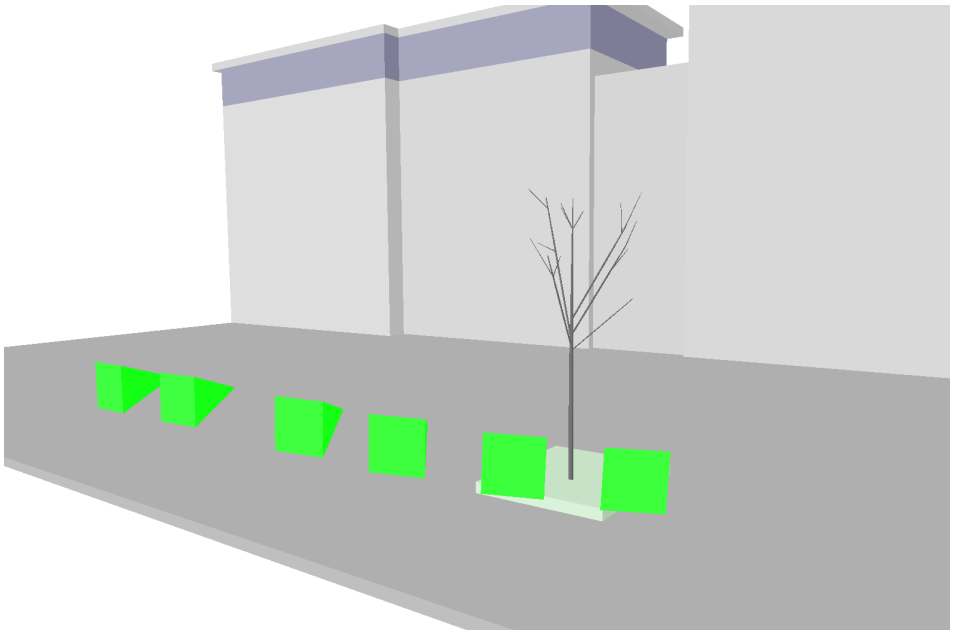

Figure 6.5: Reconstruction of a young m-tree in a city scene (cf. also Figure 6.4). Green pyramids symbolize the camera positions (tip) and poses (base).

the two curved long branches on the right side can only be roughly represented as connected straight parts. The attempt to extract the small twigs encountered strong clutter and failed to give a meaningful result.

Figure 6.7 shows that our approach works also under very different lighting conditions. The background estimation can detected the night scene and adapt itself to the switched fore- and background intensities by employing Opening (cf. Section 2.4) instead of Closing. Again, although several branches are completely occluded in some of the images, they still could be reconstructed.

6.3 Assessment and Discussion

A quantitative assessment of the results is carried out by comparing the reconstructed model with the simulated (ground truth) model based on their 2D projections, even though an evaluation in 3D would have been preferable. Yet, the computation of intersections of cylinders in 3D is far from trivial. Particularly, the output of $Xfrog^{TM}$ is a triangulated surface model without explicit axes for the branches. This makes it impossible to directly compare it to our cylinders for the branches. One could avoid the intersection of the cylinders by rasterizing them in 3D voxel space, but this is again not directly possible for the surface representation of the ground truth data.

For the quantitative evaluation in 2D space the 3D models are marked with a particular color, green for the ground truth model and red for the reconstruction, and projected into images taken from all six given viewpoints as well as an additional view point (cf. Figure 6.8). The

Figure 6.6: Result for an aged sympodial-monochasium (sm-) tree – three of six images (top) as well as the reconstructed model seen from the same points of view (bottom).

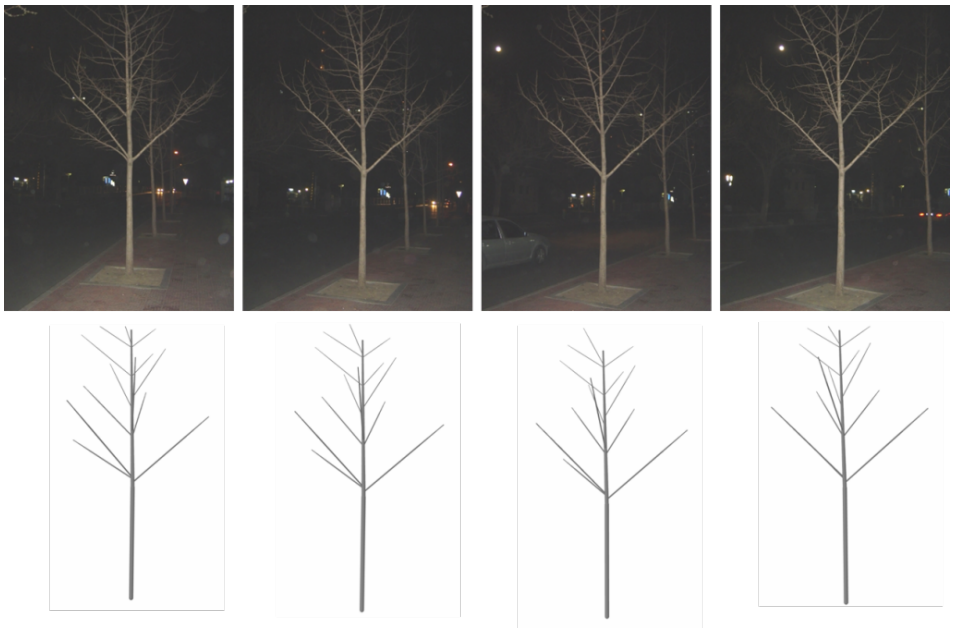

Figure 6.7: Result for a monopodial (m-) tree in night scene– image quadruple (top) as well as the reconstructed model seen from the same points of view (bottom).

latter is added to demonstrate the quality of the 3D result for an extrapolation outside the given range of views.

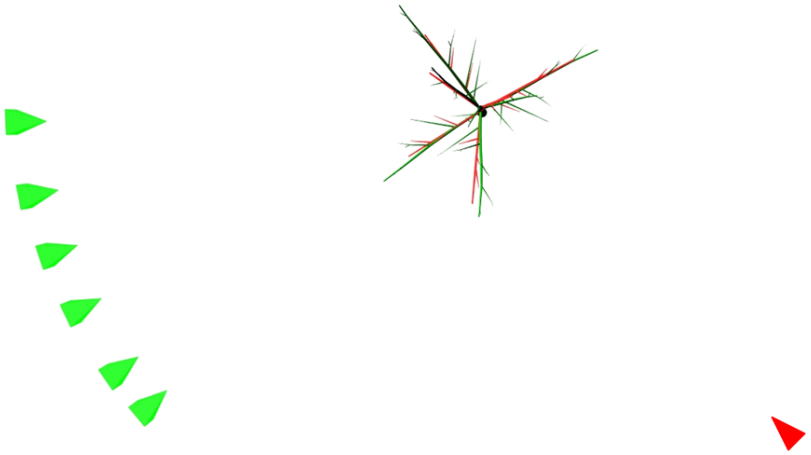

Figure 6.8: An additional extrapolated view (red), which is approximately perpendicular to the last of the snapshot positions (green), is also used for the evaluation.

The corresponding projections from each view are composed into a sequence of evaluation images. One of them is shown in Figure 6.9. We use blue color to indicate the overlap of both projections and thus define different types of regions as follows:

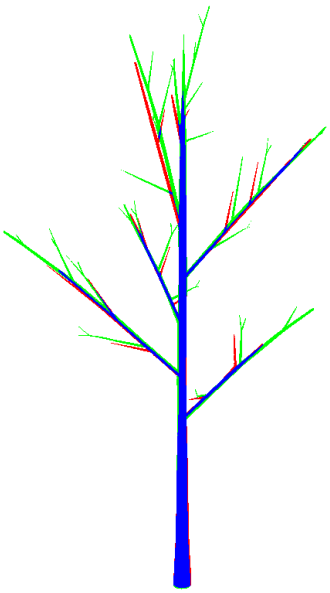

Figure 6.9: Regions in evaluation image: True Positive (blue), False Negative (green), and False Positive (red).

– True Positive (TP, blue): Reconstructed regions comply with the ground truth data.

– False Negative (FN, green): Ground truth data which could not be reconstructed.

– False Positive (FP, red): Incorrect reconstruction – does not comply with the ground truth data.

Following (Heipke et al. 1998), we use the "Completeness" and "Correctness" of the result for evaluation.

Completeness

The Completeness is the percentage of the ground truth data which complies with the reconstructed data:

$$Completeness = \frac{TP}{TP + FN}, \tag{6.1}$$

with $Completeness \in [0, 1]$.

Figure 6.10 shows the Completeness values for the seven views given in Figure 6.8. It can be seen that the Completeness of reconstruction is stable concerning different view angles.

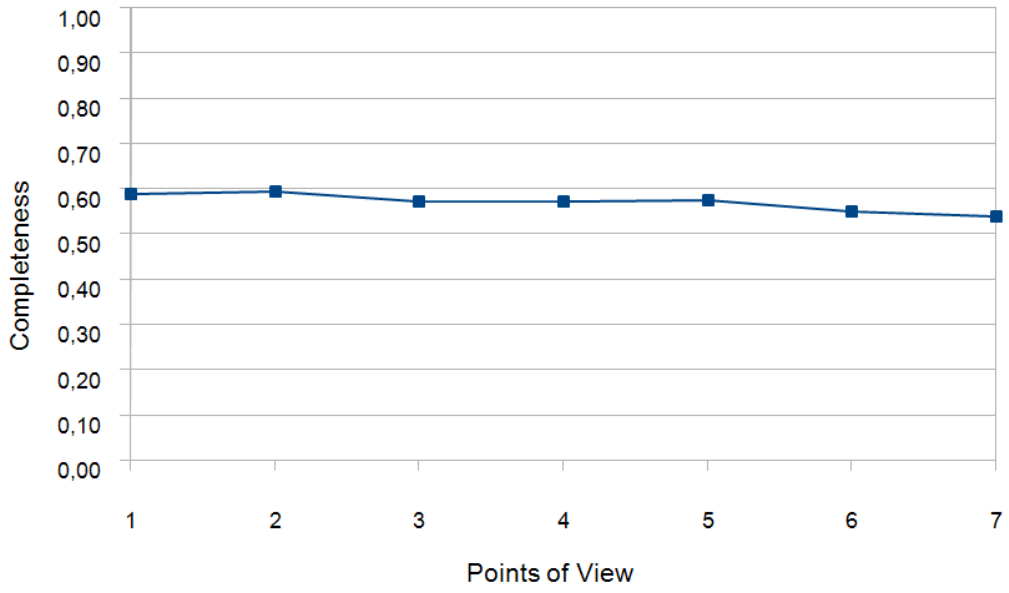

Figure 6.10: Completeness of the reconstructed result for seven points of view.

Correctness

The Correctness is the percentage of the reconstructed data which represents ground truth data, i.e., the percentage of correctly reconstructed data:

$$Correctness = \frac{TP}{TP + FP}, \tag{6.2}$$

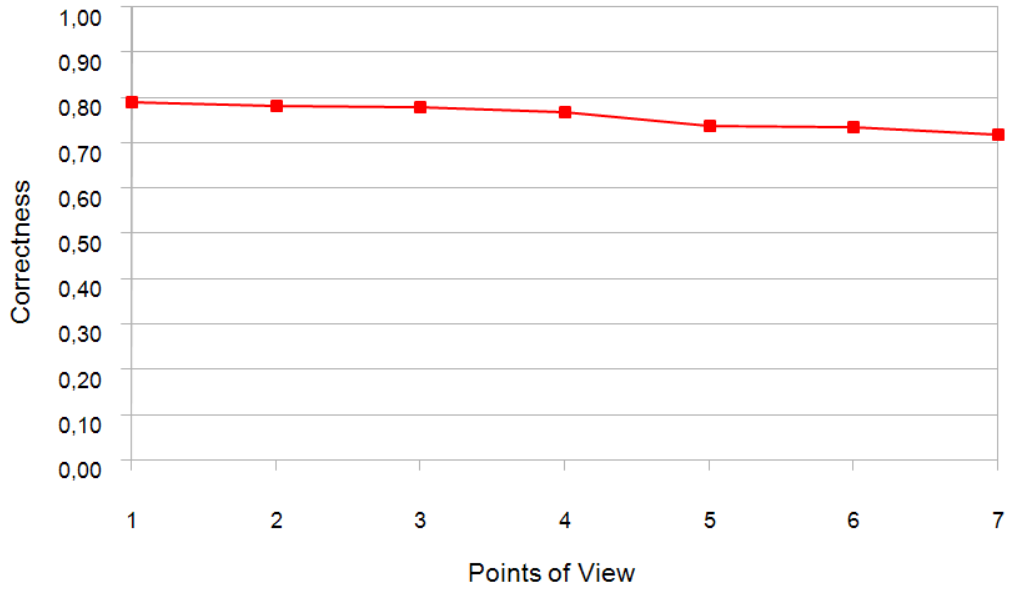

Figure 6.11: Correctness of the reconstructed result for seven points of view.

with $Correctness \in [0, 1]$.

Figure 6.11 plots the Correctness. Also the Correctness is rather stable for all points of view.

Particularly, Completeness and Correctness on view 7, i.e., the additional extrapolated view, show only a slight deviation from that of the given views, although the result has not been calculated based on this view.

Discussion – Automation

The main strength of the work presented in this thesis is its high degree of automation. In comparison with the approaches given in Section 3, there is no need for user input, e.g., sketching or indicating the trunk and the crown area, or user intervention, e.g., manual correction. The whole process from reading the image sequence to generating the VRML model is fully automatic.

Particular highlights are the automation of the following two steps:

– Determination of the trunk and its joint point with the first level branches, i.e., the crown (cf. Section 5.1)

– Selection of Production Rules of L-systems (cf. Section 5.6)

They are the key links connecting all the other steps in the extraction and reconstruction of trees. Being manually conducted in related approaches implies their importance as well as complexity.

As our output is a “relative model”, manual input needed for our work consists of absolute coordinates and scale when we want to integrate our model with building models.

Discussion – Visualization

Focusing on the branching structure, we direct less effort on texture and realistic geometry. We use a uniform color for our tree model, particularly the average gray value of the target tree in the images, assuming that in comparison with the background, the color difference of the branches can be ignored. Although some geometrical refinement is done as shown in Section 4.3, the geometric modeling of complex structures is still limited. E.g., the curved shapes in Figure 6.6 is only roughly represented.

Chapter 7

Summary and Outlook

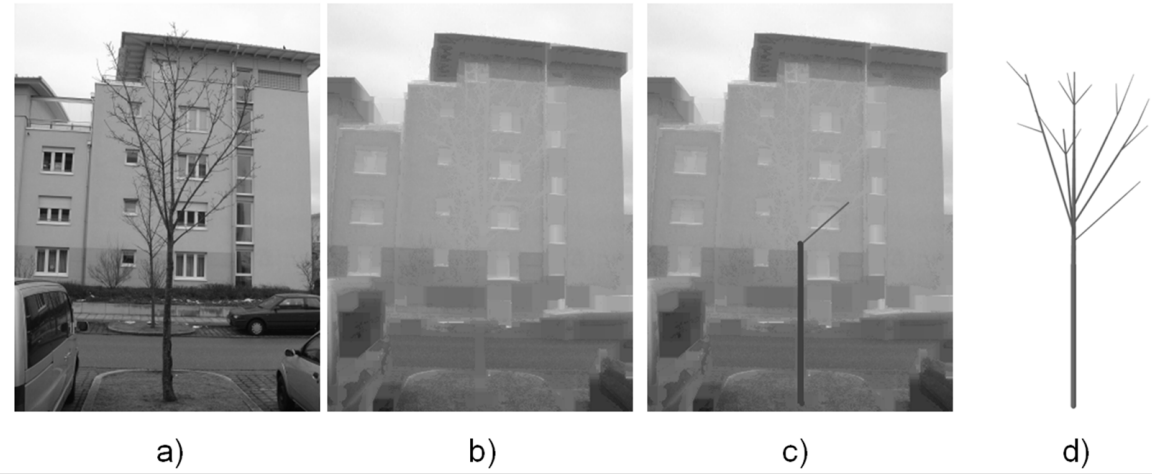

Figure 7.1: Generative statistical extraction of trees. a) Target tree in one of the input images. b) Background estimation. c) A hypothesis for a branch is projected into the estimated background for evaluation. d) 3D reconstruction of the target tree.

In this thesis we present an approach for the generative statistical 3D extraction of unfoliaged trees with their particular characteristics from terrestrial wide-baseline image sequences of urban scenes. The descriptive power for trees of L-systems, statistical sampling with Markov Chain Monte Carlo (MCMC), Bayesian inference, and maximum a posteriori (MAP) estimation are combined into a generative statistical approach. Figure 7.1 gives an overview of important steps and the result.

The highlights of our approach can be summarized as follows:

1. Background estimation: For the generative modeling projecting the branches into the individual images, the background is estimated by means of gray-scale morphology and an iterative detail reconstruction process.

2. MC+MCMC sampling: For statistically sampling the sparsely distributed branching parameters we have proposed an efficient scheme combining plain Monte Carlo (MC)

and sequential MCMC. By using the Metropolis-Hastings algorithm, the search becomes robust.

3. Probability controlled sampling sequence: A proposal mechanism has been devised to control the sampling sequence in the Markov Chain taking into account the characteristics of the individual parameters and their recent performance. By the more effective sampling, much computational effort is saved.

4. Bayesian refinement of priors: The generic prior distributions for parameters are refined in a Bayesian framework taking into account already accepted hypotheses. The updated priors are used in the MAP estimation.

5. Classification of branching types: After extracting the basic branches on the first level, the type of the L-System is derived by means of classification and Production Rules of specific L-systems are applied for a refined modeling.

By these means we are able to extract and reconstruct major branches representing the basic structure of individual trees even when they are partly occluded and the images are taken under difficult lighting conditions.

Concerning future work, a more detailed geometrical modeling by means of generalized cylinders (Deussen and Lintermann 2004) seems to be a promising way for a better representation of complex branches. Only approximately circular cross-sections on not necessarily parallel planes that are connected by means of triangles lead to more or less short cylindrical pieces. The trunk or a branch will be modeled by attaching the generalized cylinders and the triangles can be colored or textured to represent the bark.

The disadvantage of conventional MCMC is that the number of objects that can be represented as well as their parameters is fixed. One thus cannot change the components of the model or switch between different models during search. Reversible Jumps (Green 1995), which allow proposals that change the dimensionality of the parameter space in MCMC, could be integrated into our approach for a more flexible search possibly in conjunction with model selection, e.g., using Akaike's Information Criterion – AIC (Akaike 1973). The latter allows to better decide between thin twigs and the background. Search is thus optimized by an informed selection of competing hypotheses while at the same time avoiding overfitting to the data.

Generative statistical modeling is not confined to L-systems. Basically, just a means to construct realistically looking trees that can be efficiently controlled is needed. For this, e.g., the tree modeling of (Deussen and Lintermann 2004) could be a basis.

We finally assume that the upper stages of branches with very thin twigs which are hard to estimate from the noisy data might be grown stochastically with the derived Production Rules and the learned priors to match the image density.

Bibliography

Akaike, H. (1973): Information Theory and an Extension of the Maximum Likelihood Principle, *Second International Symposium on Information Theory*, Akademiai Kiado, Budapest, Hungary, 267–281.

Alegre, F. and Dallaert, F. (2004): A Probabilistic Approach to the Semantic Interpretation of Building Facades, *International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres*, 1–12.

Andersen, H.-E., Reutebuch, S. and Schreuder, G. (2002): Bayesian Object Recognition for the Analysis of Complex Forest Scenes in Airborne Laser Scanner Data, *International Archives of Photogrammetry and Remote Sensing*, Volume 34(3A), 35–41.

Andrieu, C., Freitas, N., Doucet, A. and Jordan, M. I. (2003): An Introduction to MCMC for Machine Learning, *Machine Learning* **50**(1-2): 5–43.

Aschoff, T. and Spiecker, H. (2004): Algorithms for the Automatic Detection of Trees in Laser Scanner Data, Volume 36(8/W2), 66–70.

Bienert, A., Scheller, S., Keane, E., Mullooly, G. and Mohan, F. (2006): Application of Terrestrial Laser Scanner for the Determination of Forest Inventory Parameters, Volume 36(5), 5 pages.

Bishop, C. M. (2008): A New Framework for Machine Learning, *Computational Intelligence: Research Frontiers* **5050/2008**: 1–24.

Brenner, C. and Ripperda, N. (2006): Extraction of Façades Using RJMCMC and Constraint Equations, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 36(3), 155–160.

Chen, X., Neubert, B., Xu, Y. Q., Deussen, O. and Kang, S. B. (2008): Sketch-Based Tree Modeling Using Markov Random Field, *SIGGRAPH Asia 2008, ACM Transaction on Graphics*, Volume 27(5), ACM, New York, NY, USA, 1–9.

Cheng, L., Caelli, T. and Sanchez-Azofeifa, A. (2006): Component Optimization for Image Understanding: A Bayesian Approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28(5)**: 684–693.

Deussen, O. and Lintermann, B. (2004): *Digital Design of Nature: Computer Generated Plants and Organics*, Springer, Berlin, Germany.

Dick, A., Torr, P. and Cipolla, R. (2004): Modelling and Interpretation of Architecture from Several Images, *International Journal of Computer Vision* **60**(2): 111–134.

Fischler, M. and Bolles, R. (1981): Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Communications of the ACM* **24**(6): 381–395.

Förstner, W. and Gülch, E. (1987): A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features, *ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, Interlaken, Switzerland, 281–305.

Forsyth, D., Malik, J., Fleck, M., Greenspan, H., Leung, T., Belongie, S., Carson, C. and Bregler, C. (1996): Finding Pictures of Objects in Large Collections of Images, *Object Representation in Computer Vision II*, Springer, Berlin, Germany, 335–360.

Gorte, B. and Pfeifer, N. (2004): Structuring Laser-scanned Trees Using 3D Mathematical Morphology, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 35(B5), 929–933.

Gorte, B. and Winterhalder, D. (2004): Reconstruction of Laser-Scanned Trees Using Filter Operations in the 3D Raster Domain, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 36(8/W2), 39–44.

Green, P. (1995): Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination, *Biometrika* **82**: 711–732.

Haering, N., Myles, Z. and Vitoria, N. (1997): Locating Deciduous Trees, *IEEE Workshop on Content Based Access of Image and Video Libraries*, 18–25.

Hartley, R. I. and Zisserman, A. (2004): *Multiple View Geometry in Computer Vision, Second Edition*, Cambridge University Press, Cambridge, UK.

Hastings, W. K. (1970): Monte Carlo Sampling Methods Using Markov Chains and Their Applications, *Biometrika* **57**(1): 97–109.

Heipke, C., Mayer, H. and Wiedemann, C. (1998): External Evaluation of Automatically Extracted Road Axes, *Photogrammetrie – Fernerkundung – Geoinformation* **(2/1998)**: 81–94.

Horváth, P., Jermyn, I. H., Kato, Z. and Zerubia, J. (2008): A Higher-order Active Contour Model of A "Gas of Circles" and Its Application to Tree Crown Extraction, *Pattern Recognition* **42/5**: 699–709.

HUANG, H. (2008): Terrestrial Image Based 3D Extraction of Urban Unfoliaged Trees of Different Branching Types, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 37(B3a), 253–258.

HUANG, H. and MAYER, H. (2007a): Extraction of 3D Unfoliaged Trees from Image Sequences via a Generative Statistical Approach, *The Annual Symposium of the German Association for Pattern Recognition (DAGM), Springer Lecture Notes in Computer Science*, Volume 4713, Springer, Heidelberg, Germany, 385–394.

HUANG, H. and MAYER, H. (2007b): Extraction of the 3D Branching Structure of Unfoliaged Deciduous Trees from Image Sequences, *Photogrammetrie – Fernerkundung – Geoinformation* **(6/2007)**: 429–436.

HUANG, H. and MAYER, H. (2009): Generative Statistical 3D Reconstruction of Unfoliaged Trees from Terrestrial Images, *Annals of GIS* **15**(2): 97–105.

HYYPPÄ, J., MIELONEN, T., HYYPPÄ, H., MALTAMO, M., YU, X., HONKAVAARA, E. and KAARTINEN, H. (2005): Using Individual Tree Crown Approach for Forest Volume Extraction with Aerial Images and Laser Point Clouds, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 36(3/W19), 144–149.

KÖTHE, U. (1996): Local Appropriate Scale in Morphological Scale-space, *Fourth European Conference on Computer Vision (ECCV)*, Volume I, Springer, Heidelberg, Germany, 219–228.

LEIBE, B. and SCHIELE, B. (2004): Scale Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search, *The Annual Symposium of the German Association for Pattern Recognition (DAGM), Springer Lecture Notes in Computer Science*, Volume 3175, Springer, Heidelberg, Germany, 145–153.

LINTERMANN, B. and DEUSSEN, O. (1996): Interactive Modelling and Animation of Branching Botanical Structures, *Seventh International Workshop on Computer Animation and Simulation (EGCAS '96)*, 139–151.

LINTERMANN, B. and DEUSSEN, O. (1999): Interactive Modeling of Plants, *IEEE Computer Graphics and Applications* **19**(1): 2–11.

MAYER, H. (1998): *Automatische Objektextraktion aus digitalen Luftbildern*, Volume C Number 494, Deutsche Geodätische Kommission, Munich, Germany.

MAYER, H. (2005): Robust Least-Squares Adjustment Based Orientation and Auto-Calibration of Wide-Baseline Image Sequences, *ISPRS Workshop in conjunction with ICCV 2005 "Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images" (BenCos), Beijing, China*, 1–6.

MAYER, H. (2007): 3D Reconstruction and Visualization of Urban Scenes from Uncalibrated Wide-Baseline Image Sequences, *Photogrammetrie – Fernerkundung – Geoinformation* **3/07**: 167–176.

MAYER, H. and REZNIK, S. (2007): Building Façade Interpretation from Uncalibrated Wide-baseline Image Sequences, *ISPRS Journal of Photogrammetry and Remote Sensing* **61(2007)**: 371–380.

MAYER, H., HINZ, S. and STILLA, U. (2008): Automated Extraction of Roads, Buildings and Vegetation from Multi-Source Data, *Advances in Photogrammetry, Romote Sensing and Spatial Information Sciences: 2008 ISPRS Congress Book*, Taylor & Francis, London, UK, 213–226.

MĚCH and PRUSINKIEWICZ, P. (1996): Visual Models of Plants Interacting with Their Environment, *SIGGRAPH 1996, ACM Transactions on Graphics*, ACM, New York, NY, USA, 397–410.

METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H. and TELLER, E. (1953): Equation of State Calculation by Fast Computing Machines, *Journal of Chemical Physics* **21(6)**: 1087–1092.

MÜLLER, P., WONKA, P., HAEGLER, S., ULMER, A. and VAN GOOL, L. (2006): Procedural Modeling of Buildings, *SIGGRAPH 2006, ACM Transactions on Graphics*, Volume 25(3), ACM, New York, NY, USA, 614–623.

NEUBERT, B., FRANKEN, T. and DEUSSEN, O. (2007): Approximate Image-Based Tree-Modeling Using Particle Flows, *SIGGRAPH 2007, ACM Transactions on Graphics*, ACM, New York, NY, USA, 88.

NISTÉR, D. (2004): An Efficient Solution to the Five-Point Relative Pose Problem, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(6): 756–770.

PERSSON, Å., HOLMGREN, J., SÖDERMAN, U. and OLSSON, H. (2004): Tree Species Classification of Individual Trees in Sweden by Combining High Resolution Laser Data with High Resolution Near-infrared Digital Images, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 36(8/W2), 204–207.

PFEIFER, N., GORTE, B. and WINTERHALDER, D. (2004): Automatic Reconstruction of Single Trees from Terrestrial Laser Scanner Data, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 35(B5), 114–119.

PINZ, A., ZARMBA, M., BISCHOF, H., GOUGEON, F. and LOCAS, M. (1993): Neuromorphic Methods for Recognition of Compact Image Objects, *Machine Graphics and Vision* **2**: 209–229.

PRUSINKIEWICZ, P. and LINDENMAYER, A. (1990): *The Algorithmic Beauty of Plants*, Springer, New York, NY, USA.

QUAN, L., TAN, P., ZENG, G. YUAN, L., WANG, J. and KANG, S. (2006): Image-based Plant Modeling, *SIGGRAPH 2006, ACM Transactions on Graphics*, Volume 25(3), ACM, New York, NY, USA, 772–778.

RECHE, A., MARTIN, I. and DRETTAKIS, G. (2004): Volumetric Reconstruction and Interactive Rendering of Trees from Photographs, *SIGGRAPH 2004, ACM Transactions on Graphics*, Volume 23(3), 720–727.

REITBERGER, J., KRZYSTEK, P. and STILLA, U. (2006): Analysis of Full Waveform LIDAR Data for Tree Species Classification, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 36(3), 228–233.

RIPPERDA, N. and BRENNER, C. (2007): Data Driven Rule Proposal for Grammar Based Facade Reconstruction, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 36(3/W49A), 1–6.

ROBERT, C. and CASELLA, G. (2008): A History of Markov Chain Monte Carlo – Subjective Recollections from Incomplete Data, *ArXiv e-prints* **stat.CO**: 0808.2902.

SAKAGUCHI, T. and OHYA, J. (1999): Modeling and Animation of Botanical Trees for Interactive Virtual Environments, *VRST '99: ACM symposium on Virtual reality software and technology*, ACM, New York, NY, USA, 139–146.

SERRA, J. (1983): *Image Analysis and Mathematical Morphology*, Academic Press, New York, NY, USA.

SHLYAKHTER, I., ROZENOER, M., DORSEY, J. and TELLER, S. (2001): Reconstructing 3D Tree Models from Instrumented Photographs, *IEEE Computer Graphics and Applications* **21**(3): 53–61.

STEGER, C. (1998): An Unbiased Detector of Curvilinear Structures, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**: 113–125.

STOICA, R., DESCOMBES, X. and ZERUBIA, J. (2004): A Gibbs Point Process for Road Extraction from Remotely Sensed Images, *International Journal of Computer Vision* **57**(2): 121–136.

TAN, P., FANG, T., XIAO, J., ZHAO, P. and QUAN, L. (2008): Single Image Tree Modeling, *SIGGRAPH Asia 2008, ACM Transactions on Graphics*, ACM, New York, NY, USA, 1–7.

TAN, P., ZENG, G., WANG, J., KANG, S. and QUAN, L. (2007): Image-based Tree Modeling, *SIGGRAPH 2007, ACM Transactions on Graphics*, Volume 26(3), Article 87, ACM, New York, NY, USA, 87.

TU, Z. and ZHU, S. (2006): Parsing Images into Regions, Curves, and Curve Groups, *International Journal of Computer Vision* **69(2)**: 223–249.

VAN GOOL, L., ZENG, G., VAN DEN BORRE, F. and MÜLLER, P. (2007): Towards Mass-produced Building Models, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume 36(3/W49), 209–220.