

GENERIERUNG VON OBJEKTEN FÜR 3D-STADTMODELLE

Zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
bei der Fakultät für Bauingenieur- und Vermessungswesen
der Universität der Bundeswehr München
genehmigte Dissertation

von

Lothar Koppers

Angaben in den Pflichtexemplaren

Universität der Bundeswehr München
Fakultät für Bauingenieur- und Vermessungswesen

Thema der Dissertation Generierung von Objekten für 3D-Stadtmodelle

Verfasser Dipl.-Ing. Lothar Koppers

Promotionsausschuss

Vorsitzender Univ.-Prof. Dr.-Ing. Helmut Mayer

1. Berichterstatter Univ.-Prof. Dr.-Ing. i. R. Dr.-Ing. W. Caspary

2. Berichterstatter Univ.-Prof. Dr.-Ing. Wolfgang Reinhardt

Tag der Prüfung 18.10.2002

Mit der Promotion erlangter akademischer Grad:

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

Neubiberg, den 18. Oktober 2002

Inhaltsverzeichnis

1.	Einleitung und Zielsetzung	3
1.1	Gliederung	3
1.2	Grundkonzept und Einsatzbereiche für 3D-Stadtmodelle	4
1.2.1	Visualisierung von 3D-Stadtmodellen	5
1.2.2	Berechnung von Lärmausbreitung	5
1.2.3	Berechnung zur Ausleuchtung von Funkräumen	6
1.2.4	Raum- und Stadtplanung	7
1.2.5	Stadtmodelle als Datenbasis für Spielumgebungen	9
2.	Verfahren zur Erfassung und Gestaltung von 3D-Stadtmodellen	11
2.1	Stand der Wissenschaft	11
2.1.1	Verfahren zur geometrischen Rekonstruktion von Gebäuden	11
2.1.2	Verfahren zur Baumextraktion	15
2.1.3	Verfahren zur Gestaltung von Geo-Objekte in Städten	16
2.2	Zu bearbeitende Aufgaben	18
3.	Datenquellen, -informationen und Konzepte	20
3.1	GIS-Daten der Vegetation und der Bebauung	20
3.1.1	Daten der Vegetation	20
3.1.2	Daten der Bebauung	21
3.2	Bilddaten, Formate und Komprimierung	21
3.2.1	Joint Photographic Experts Group – Format Jpeg	22
3.2.2	Graphics Interchange Format – Gif	23
3.3	Geometrie und Texturen aus Bilddaten	25
3.4	Regeln und Vorschriften	30
3.5	Detailstufen	30
4.	Modellierung von Vegetationsobjekten	32
4.1	Kugelbaumobjekte	33
4.1.1	Einfache Kugelbaumobjekte	33
4.1.2	Komplexe Kugelbaum-Objekte	34
4.1.3	Texturen für Kugelbaum-Objekte	35
4.2	Bäume aus L-Systemen	35
4.2.1	Das Prinzip	35
4.2.2	Texturierung von L-Bäumen	38
4.3	Bäume mit Ebenentechnik	38
4.3.1	Bäume als Tafelflächen	38
4.3.2	Modellierung von Bäumen als Kreuztafelobjekte	39
4.3.3	Modellanforderung für die Berechnung von Schall- oder Funkwellenausbreitung	40
4.4	Texturen für Vegetationsobjekte	41
4.4.1	Fotografisch erzeugte Texturen	41
4.4.2	Texturen aus L-Systemen	41
4.4.3	Animierte Texturen	43
4.4.4	Mehrfachtexturierung	45
4.4.5	Das Texturmodell	45
4.5	Automatisierte Detailstufen	46
4.6	Aggregierte Vegetationsobjekte	49

5.	Modellierung von Gebäudeobjekten	51
5.1	Modellierungskonzepte	51
5.1.1	Volumenmodell	51
5.1.2	Flächenmodell	53
5.1.3	Gegenüberstellung	54
5.2	Modellierung der Gebäudebestandteile	54
5.2.1	Mauerwerk	54
5.2.2	Dach	57
5.2.3	Die geometrischen Kenngrößen des Daches	64
5.2.4	Texturierung der Gebäudemodelle	68
5.2.4.1	Fassadentexturen	69
5.2.4.2	Dachtexturen	73
5.3.	Automatisierte Vereinfachung von Gebäudeobjekten	74
5.4	Reihenhauszeile und Wohnblock	75
6.	Objektmuster	
6.1	Die Allgemeinform des Baumes	76
6.2	Die Allgemeinform für Gebäude	76
7.	Umsetzung der Ergebnisse in verschiedenen Sprachen	81
7.1	Implementierung zur Speicherung	81
7.1.1	Integration in X3D	84
7.1.2	Integration in GML	86
7.1.3	Vergleich der Modellierungsmöglichkeiten	92
7.2	Implementierung zur Visualisierung	93
7.2.1	Implementierung in VRML	94
7.2.2	Implementierung in Java3D	96
8.	Einsatzmöglichkeit in der Praxis	99
8.1.	Tauglichkeit des Verfahrens	99
8.2	Wirtschaftlichkeit des Verfahrens	103
8.3	Fazit	104
	Anhang	108
	Literaturverzeichnis	

Kapitel 1

EINLEITUNG UND ZIELSETZUNG

Das Ziel dieser Arbeit ist die Entwicklung eines Verfahrens zur Erstellung von 3D-Stadtmodellen aus bestehenden bzw. leicht herzustellenden Datenbeständen. Das Resultat wird keine auf photogrammetrischen Verfahren beruhende High-End-Lösung sein. Das Vorhaben zielt auf ein hinreichendes und zweckmäßiges Ergebnis mit möglichst geringen wirtschaftlichen Mitteln und Zeitaufwenden.

Die Modellierung dreidimensionaler Geoobjekte als Abbildung der Wirklichkeit, erstreckt sich im wesentlichen auf drei Teilbereiche (Abb. 1).

- Objektkontext (insbesondere die Erdoberfläche), der alle nicht als einzelne Objekte modellierten Teile betrifft, beispielsweise ein Höhenmodell texturiert mit einem Orthofoto, aber auch Straßen, Wiesen oder Äcker, sowie Nachbarobjekte
- Objektgeometrie, eine zahlenbezogene Beschreibung, welche die Ausmaße und die Topologie eines Objektes zum Inhalt hat
- Objekterscheinung, die Materialeigenschaften eines Objektes

Arbeiten zum Objektkontext werden schon seit geraumer Zeit betrieben und können aus im wesentlichen als abgeschlossen betrachtet werden. Üblicherweise werden Dreiecksvermaschun-

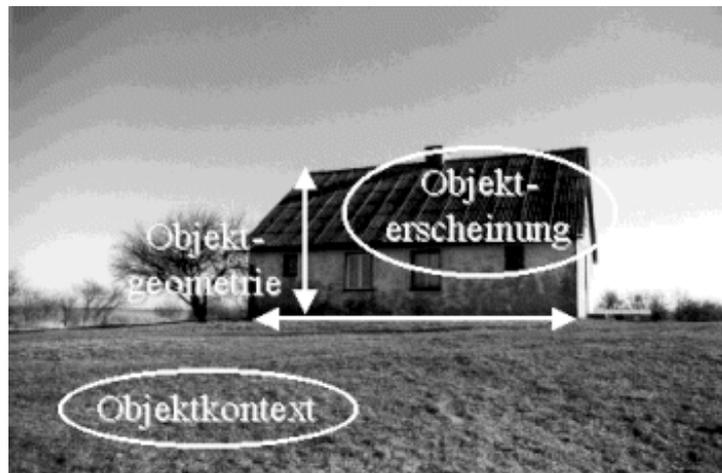


Abb. 1: Objektkontext, Objektgeometrie und Objekterscheinung in der Natur

gen oder Gitter verwendet, um diesen Teilbereich zu modellieren.

Im Teilbereich Objektgeometrie und Objekterscheinung sind jedoch viele Fragen offen oder noch nicht bis ins Detail beleuchtet. Einige dieser Fragen stehen deshalb im Mittelpunkt dieser Arbeit.

1.1 Gliederung

Verfahrensweisen zur Erfassung der groben Strukturen der Objektumrisse sind in ihrer Entwicklung weit fortgeschritten. Hierzu kommen eine ganze Reihe von Verfahren zum Einsatz, die in Kapitel 2 vorgestellt werden. Teilweise haben diese Verfahren die wissenschaftlichen Labore verlassen und sind als kommerzielle Software verfügbar, wie beispielsweise [GO3DM], [INPHO], [SCOP] oder [CCM].

Die Arbeiten zur Entwicklung der Detailbeschreibung und der Objekterscheinung sind nicht so weit fortgeschritten. Dies liegt sicherlich daran, dass die Vielfalt der Formen und Strukturen einer automatischen Bearbeitung schwerer zugänglich ist. Ein anderer Grund ist sicherlich darin zu suchen, dass es für diesen Themenbereich nur wenige, zumeist grobe Informationen gibt oder aber die tatsächliche Realität im Einzelfall fotografisch wiedergegeben wird (Kap.3). Fotografien als Datenquellen sind zwar hochdetailliert, ihre Bearbeitung ist aber auch arbeitsintensiv und finden deshalb nur im besonderen Fällen Anwendung. Ein großflächig ein hochauflösendes, detailgetreues Stadtmodell kann nur mit erheblichen Aufwand errichtet werden. Die vorliegende Arbeit stellt sich das Ziel, ohne Benutzung photogrammetrischer Aufnahmen und unter Beachtung der Aufwandsminimierung (Kosten), möglichst gute, realitätsnahe 3D-Stadtmodelle zu erzeugen. In diesem Sinne leistet die Arbeit einen Beitrag zur Generierung der Objektarten Vegetation (Kap.4) und Gebäude (Kap.5). Die Modelle, die in diesen Kapiteln betrachtet werden, sind gleichzeitig die Basis für verfeinerte Darstellungen, deren Datengrundlage aus realen Objekten gewonnen wird, beispielsweise Unterlagen der Raumplanung.

Auf Basis der in den Kapiteln 4 und 5 erarbeiteten Ansätze müssen die Erkenntnisse in eine für die praktische Verwendung notwendige Form umgesetzt werden. Um dies zu erreichen, werden vom Format unabhängige, allgemein gültige Objektdefinitionen (Kap.6) beschrieben.

Als gemeinsamer Sprachstandard für die Objektdefinitionen in Kap.7 wird XML gewählt. Dabei wird die Umsetzung auf unterschiedliche Art und Weise beschrieben, anschließend verglichen und bewertet. Die Arbeit endet mit einem praktischen Beispiel, das die Funktionsfähigkeit der vorgestellten Modellierung nachweist. Das geschieht mit der zu dieser Arbeit entstandenen Software „Urban Model Generator“.

Insgesamt soll die vorliegende Arbeit als Beitrag ergänzend zu den bekannten photogrammetrischen Verfahren (Kap. 2) verstanden werden. Sie zeigt einen Weg auf, wie kostengünstig aus bestehenden Datenbeständen und leicht integrierbaren weiteren Informationen ein 3D-Stadtmodell entwickelt werden kann.

1.2 Grundkonzept und Einsatzbereiche für 3D-Stadtmodelle

Ein 3D-Stadtmodell besteht aus einer dreidimensionalen Abbildung bebauter Gebiete und deren näheren Umgebung. Gebäudeobjekte sind daher häufig im Mittelpunkt von 3D-Stadtmodellen. Es kann jedoch auch andere Objektarten, beispielsweise Vegetation enthalten. Eine Auswahl weiterer möglicher Objektarten sind Straßen und Wege, Verkehrsschilder, Straßenlaternen und andere. Stadtmodelle werden zu den verschiedensten Zwecken errichtet. Das eigentliche Stadtmodell ist dabei eine Grundlage für unterschiedlichen Anwendungen, die sich grob in die Bereiche Visualisierung und Simulation gliedern lassen.

Die in dieser Arbeit vorgestellte Vorgehensweise ist geeignet, maßvolle Anforderungen an Detailtreue und Genauigkeit zu erfüllen. Es wird nicht jede Einzelheit dargestellt. So kann mit dem Modell eine Darstellung mittlerer Auflösung generiert werden, deren Feinheiten von Hand nachbearbeitet werden muss, wie Abb. 2 zeigt.

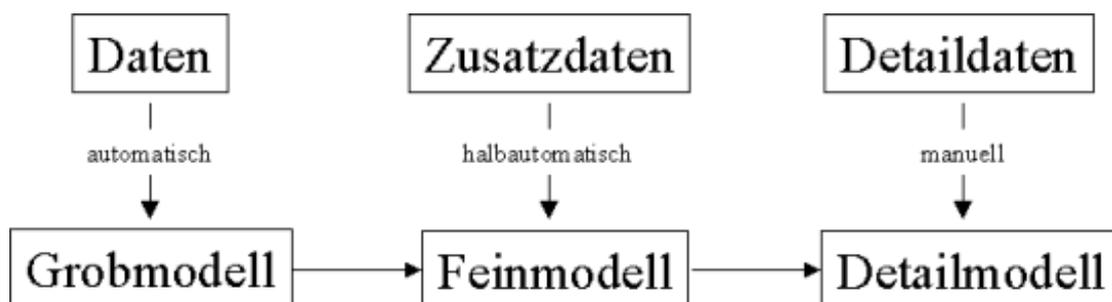


Abb. 2: Arbeitsprozess für die Bearbeitung vom Grob- bis zum Detailmodell

Aus vorhandenen Daten wird automatisch ein Grobmodell erzeugt. Durch zusätzlich eingeführte Informationen entsteht das Feinmodell. Die Erstellung eines Feinmodells kann manuell durch Festlegungen oder durch Zuführung existierender Datenquellen erfolgen. Ein Detailmodell entsteht schließlich auf Basis des Feinmodells, wenn zusätzlich in einem Gestaltungsprogramm Besonderheiten herausgearbeitet werden.

1.2.1 Visualisierung von 3D-Stadtmodellen

Eine wichtige Anwendung von 3D-Stadtmodellen ist die Visualisierung. Bei der Visualisierung wird das Modell zur Betrachtung optimiert. Die Optimierung beinhaltet die standpunktbezogene Steuerung der Darstellung des Detailreichtums.

Eine Möglichkeit der Visualisierung ist der „virtuelle“ Spaziergang. Dabei kann sich der Betrachter frei in einer Szene bewegen. Insbesondere touristische oder historisch interessante Objekte [KOPPERS 00] [[INTERNET3D](#)] sind hervorragend für diese Form der Visualisierung geeignet. Mit den entstehenden Ergebnissen werden häufig Werbemaßnahmen unterschiedlichster Art betrieben. Die Darstellung muss weitgehend detailgetreu sein und einen realistischen Eindruck hinterlassen.

Eine weitere Visualisierungsart ist die Rekonstruktion historisch verschwundener Szenarien, von denen nur Aufzeichnungen existieren. Auf diese Nutzung wird im Beispielprojekt (Kap. 8) näher eingegangen.

In einer weiteren Anwendungsart sind großflächige Bereiche visualisiert. Dabei wird versucht, die Situation realitätsgetreu nachzubilden. So kann man sich in ihr orientieren, Sichtlinien prüfen oder aber diese Darstellung als Basis für Simulationen (Integration beweglicher Objekte) verwenden. Dreidimensionale Karten werden in unterschiedlichen Detaillierungsgraden erzeugt. Diese Anwendung der Visualisierung wird als dreidimensionale Karte [KOPPERS 98] [[PLANET9](#)] bezeichnet.

Ein anderer Visualisierungszweck ist die Verwendung als virtuelles Exposee d.h. als Informationsbasis für den Verkauf von Immobilien. Ein Exposee ist eine Zusammenstellung aller für eine Immobilientransaktion (Verkauf bzw. Miete) notwendigen Informationen. Neben Zahlenangaben über das Objekt beinhaltet ein Exposee auch Pläne und Ansichten des betreffenden Objektes. Für diesen Fall ist es naheliegend, die Unterlagen durch eine künstliche Ansicht zu ergänzen, in der die Pläne und das Umfeld zum Leben erweckt werden.

Stadtmodelle werden auch verwendet, um hochwertige Gebäudeobjekte oder Gebäudeensembles zu präsentieren. Für diese Objekte besteht häufig ein europaweites, oft auch ein weltweites Interesse. Ein Beispiel für ein solches Objekt ist z.B. der Industriepark Höchst, in dem freierwerbende Objekte weltweit operierenden Firmen der Chemie-Branche angeboten werden.

1.2.2 Berechnung von Lärmausbreitung

In die Berechnung von Lärmausbreitungen gehen unterschiedliche Daten ein:

- Geometriedaten aus dem Stadtmodell sowie das Geländemodell
- klimatische Daten, vor allem Angaben zur Luftfeuchte
- Schallquellendaten, gegebenenfalls mit Richtcharakteristik

Auf Basis der in dieser Arbeit entwickelten Vorgehensweise ist es möglich, bisherige Berechnungsverfahren für die Lärmausbreitung zu verfeinern. Die heute üblichen Verfahren beruhen, bedingt durch die eingeschränkte Datenlage, häufig auf stark vereinfachten Modellen. Beispiel-

haft sei hier das aktuelle Straßenlärm-Berechnungsmodell für die Schweiz angeführt [LSV]. Hier wird ein Reflexionszuschlag für die geschlossen bebaute Straße mit einem Tabellenwert

$$\frac{H_F}{W} \quad \text{und} \quad \frac{\sqrt{a^2 + (H_E - 0.8)^2}}{W}$$

angegeben, wobei die Werte W – die mittlere Straßenweite, H_F – mittlere Fassadenhöhe, H_E – Empfangspunkthöhe sowie a – die mittlere Entfernung zwischen Quelle und Empfangspunkt darstellen. Dieses Modell setzt an der Fassadenhöhe an, die wesentlich genauer bestimmt werden kann. Dies fordert genaue Verfahren, wie es beispielsweise der Teil 3 der [EN 12354] (Einheitliche Grundlagen für die Berechnung akustischer Eigenschaften von Gebäuden) fordert.

1.2.3 Berechnung zur Ausleuchtung von Funkräumen

Knappe Finanzmittel und eine steigende ablehnende Haltung der Anwohner gegen die Errichtung neuer Mobilfunksender zwingen die Betreiber zu Flexibilität und sorgfältiger Planung der Funknetze. So verhandeln z.B. die zukünftigen Betreiber der UMTS-Netze über den Aufbau

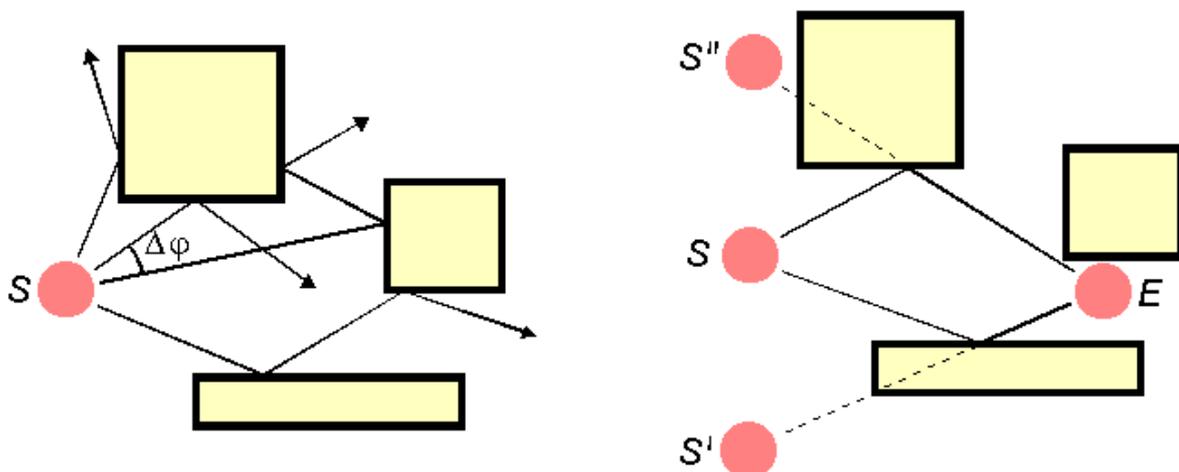


Abb. 3: Verfahren zur Funkraumberechnung: links Ray-Launching, rechts Ray Tracing

gemeinsamer Sendestandorte. Dazu sind genauere Verfahren zur optimalen Funknetzplanung erforderlich, die sich unter anderem auf realistische Stadtmodelle stützen müssen. Weniger Bedeutung haben detailreiche Modelle im Richtfunkbereich. Zur Planung von Funknetzen sind zwei Verfahren üblich. Die Grundzüge beider Verfahren sind in Abb. 3 zu erkennen.

Beim Ray-Launching Verfahren wird der einzelne Strahl verfolgt. Wird der Strahl an z.B. einer Hauswand gebrochen, so wird mit Einfallswinkel gleich Ausfallswinkel die weitere Richtung des Strahls bestimmt. Die Feldstärke entlang des Strahls wird bei bekanntem Weg durch eine Wegstreckenfunktion und durch Dämpfungskoeffizienten (bei Reflexion an einem Material) berechnet. Der Strahl wird nach abgeschlossener Berechnung mit einem festen Winkelinkrement verändert.

Das Ray-Tracing Verfahren entstammt ursprünglich Techniken der 3D-Visualisierung. Die von einer Lichtquelle ausströmenden unendlich vielen Lichtstrahlen werden von Objekten reflektiert und gebrochen, bis einige wenige das Auge des Betrachters treffen. Beim Ray-Tracing verläuft die Berechnung genau anders herum. Da nicht unendlich viele Strahlen (Funkwellen) einer Lichtquelle (Funksender) in einer Berechnung verfolgt werden können, werden nur genau die

Strahlen betrachtet, die das Auge des Betrachters (Empfängers) erreichen. Dies wird möglich, indem Strahlen vom Empfänger in die Objektwelt entsendet werden. Die auftreffende Welle wird an Oberflächen reflektiert bzw. gebrochen. Die Welle wird weiter in die Szene geworfen und verfolgt, bis sie aufgrund der Lauflängenintensität zu schwach wird, die Szene verlässt oder auf einen Sender trifft. Danach wird rückwärts die Intensität nach den gleichen Verfahren wie beim Ray-Launching berechnet.

In Abb. 4 ist ein typisches Ergebnis aus einer Funkraumberechnung nach dem Ray-Tracing Verfahren gegeben. Die Intensität der Feldstärke sinkt dabei von violett über rot, gelb, grün nach blau. Dieses Verfahren wird beispielsweise bei T-Mobile verwendet.

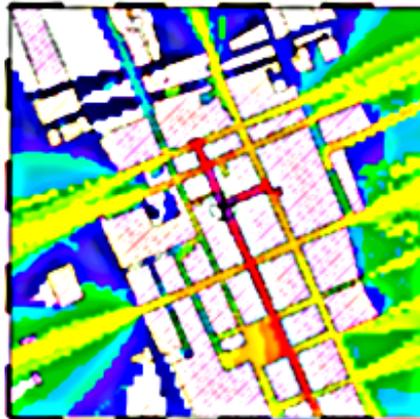


Abb. 4: Ergebnis einer Feldstärke-schätzung mittels Ray-Tracing

In beiden Standardverfahren werden Modelle für die Feldstärkendämpfung bei Reflexionen an Objektoberflächen verwendet. Wichtige Parameter der Modelle sind der Detailreichtum, das Material und die Oberflächenstruktur der reflektierenden Flächen. Das entwickelte Generierungsverfahren basiert zur Fassadenmodellierung auf eben diesen Parametern und ist daher für Funkausbreitungsberechnungen besonders geeignet.

1.2.4 Raum- und Stadtplanung

Im Bereich der Planung haben 3D-Modelle eine lange Tradition. Hier werden allerdings reale Modelle verwendet, die i.d.R. aufwändig aus Gips, Holz, Karton, Kunststoffen gefertigt werden. Bei diesen Modellen wird meist stark abstrahiert sowie auf Farbgebung und Textur verzichtet.

Die Ansprüche, welche die Raumplanung an 3D-Stadtmodelle stellt, sind vielseitig und werden im Folgenden, basierend auf Auskünften der Abteilung Raumplanung und Verkehr des Magistrats der Stadt Salzburg genannt:

- Die Kubatur der Objekte soll ersichtlich sein (Anm.: Kubatur ist die aus dem Volumen resultierende Wirkung)
- Die Raumwirksamkeit (Begrenzung von Plätzen, Straßenraum usw.) von Gebäuden soll erkennbar sein
- Das Umgebungsmodell soll sich neutral verhalten (es wird nur die Farbe Grau verwendet)
- Die Topographie soll erkennbar sein
- Die Begrenzungen und die Gliederung des öffentliche Straßenraumes soll erkennbar sein

- Die Gebäudehöhen sollen ablesbar sein (Anm. an Vergleichsobjekten, wie Personen, Fahrzeuge)
- Markante Objekte (Denkmäler und dgl.) sollen dargestellt werden
- Bäume sollen ab einem Stammumfang von ca. 1m dargestellt werden

Weiterhin kann die Gestalt von Gebäuden soweit vereinfacht werden, dass sie aufgrund ihrer Proportion erkennbar ist:

- Dachformen mit Dachvorsprüngen
- Fenster, Eingänge und Glasflächen abstrakt als Schattierung auf der Fassade
- Balkone, Erker, Durchgänge usw.
- Raumbildende Elemente wie z. B. Säulen

Sämtliche Objektbestandteile sollten von der Seite nicht als Fläche erkennbar sein. (z. B. sollen Dächer durchschnittlich eine Stärke von ca. 10 cm haben, damit ist die Modellierung des Ortanges und der Traufe gemeint).

Die in dieser Auflistung genannten Anforderungen überschreiten bei weitem die Fähigkeiten der bisher bekannten Objektgestaltungen. Die Realisierung derartig detaillierter Forderungen kann nur in Ansätzen halbautomatisch oder gar automatisch gelingen. Deshalb muss eine Gestaltung für diesen Zweck manuelle Änderungen (Abb.2) zulassen.



Abb. 5: Vergleich zwischen einer Fotografie und dem dazugehörigen Modell für die Raumplanung (Situation in Salzburg)

Das Beispiel eines Gebäudes (Abb. 5) mit verschiedenen Grundrissen der einzelnen Stockwerke verdeutlicht wie flexibel ein Gestaltungsverfahren sein muss, um in diesem Fall die Wünsche der Raumplaner wenigstens näherungsweise zu erfüllen. Durch einen zurückgesetzten Eingangsbereich differiert der Erdgeschossgrundriss gegenüber den übrigen Etagen. Hier muss die Verwendung stockwerkbezogener Grundrisse gestattet sein. Dies führt aber auch zu einer komplexen Struktur in der Fassade. So muss die Fassade stockwerksweise aufgebaut werden. Auch die Fenster müssen für jedes Stockwerk neu berechnet werden und zwischen die Etagen müssen Decken eingezogen werden, da bei unterschiedlichen Grundrissen ansonsten in das Gebäudeinnere geblickt werden kann.

Eine solche Gestaltung kann das hier vorgestellte Verfahren nicht leisten. Es kann jedoch eine bereits recht weit gehende Modellierung bieten, bei der die fehlenden Elemente nachträglich gestaltet werden können.

1.2.5 Stadtmodelle als Datenbasis für Spielumgebungen

Ein weiteres Nutzungsfeld für 3D-Stadtmodelle liegt im Bereich der Computerspiele. Spiele finden häufig in realitätsnahen virtuellen Welten statt, die sicherlich für diese Anwendung optimiert wurden. Die Nutzer legen dabei einen hohen Wert auf einen realistischen Eindruck der Szenen. Gesteigert wird dieser Eindruck, wenn nicht nur realistisch wirkende, sondern wirkliche Umgebungen mit gewissen Wiedererkennungsobjekten verwendet werden. Ein Beispiel dafür sind Flugsimulatoren, wie in Abb. 6 ein Ausschnitt des Flugsimulator II von Microsoft verdeutlicht.



Abb. 6: Screenshot Microsoft Flugsimulator 02

Die Visualisierung New Yorks ist derartig realistisch, dass Hersteller Microsoft bereits kurz nach den Terrorattentaten vom 11. September 2001 ein Patch veröffentlichte, um die Türme des WTC aus den Ansichten zu entfernen. Zudem musste man sich gegen heftige Vorwürfe zur Wehr setzen, mit dem Flugsimulator könne das Fliegen erlernt werden. Allerdings werden in Flugsimulatoren bei weitem nicht alle Gebäude, sondern eher die Landschaftserkennungsobjekte (Landmarks) abgebildet, sowie bei der Modellierung auf die fast ausschließliche Sicht von oben geachtet. Feiner modelliert werden Ansichten, welche aus der Nähe und vom Erdboden aus betrachtet werden, insbesondere die Flughäfen. Allerdings ist deutlich zu erkennen, dass alle Gebäude mit Flachdächern dargestellt sind.

Ein weitere Gruppe von Spielen, in denen 3D-Stadtmodelle verwendet werden, sind die sogenannten 3D-Shooter. Dies sind Spiele, in denen allein oder in Gruppen (Netzwerk) auf Avatare (Visualisierungen anderer Mitspieler) oder virtuelle Figuren geschossen wird. Bekannte Vertreter dieser Gruppe sind DOOM oder Quake. Die 3D-Stadtmodelle können natürlich nicht zu Innenraumszenen verwendet werden. Für Szenen im Außenbereich kann diese Gestaltungsweise sehr wohl verwendet werden. In Abb. 7 wird deutlich, dass das in dieser Arbeit vorgestellte Verfahren hier verwendet werden kann.



Abb. 7: Screenshot Map-Editor für Quake

Eine dritte Gruppe von Spielen, in denen 3D-Stadtmodelle eine tragende Wirkung haben wird zur Zeit gerade entwickelt. Es sind Spiele, die für ganz bestimmte Stadt oder Stadtteil werben sollen. So werden in Salzburg bereits seit längerem Überlegungen angestellt, wie der Film „The sound of music“ in ein Spiel umgesetzt werden kann. Hiervon verspricht sich der Fremdenverkehr einen Aufschwung durch Spieler, die sich die Stätten des Spiel im Original anschauen wollen.

Kapitel 2

VERFAHREN ZUR ERFASSUNG UND GESTALTUNG VON 3D-STADTMODELLEN

3D-Stadtmodelle sind bereits einige Zeit Gegenstand der Forschung. Beginnend in den frühen 60er Jahren des 20. Jahrhunderts entstanden die ersten Entwicklungen im Bereich Virtual Reality bereits mit Szenen von im weitesten Sinne 3D Stadtmodellen.

Erste Schritte in Richtung einer virtuellen 3D-Umgebung wurden 1962 unternommen. Morton Heilig entwickelte eine bi-okulare Sichtbrille, die es erlaubte, einen stereoskopischen Endlosfilm einer Motorradfahrt durch New York zu betrachten. ‚Sensorama‘, so der Name des Systems, vermittelte dem Benutzer zusätzlich über Motoren die Vibrationen einer Motorradfahrt und ein kleiner Ventilator blies typische Gerüche wie beispielsweise Abgase in das Gesicht des Benutzers. Sensorama war eines der innovativsten Simulationssysteme der 60er Jahre und stand dort, wo auch heutzutage die öffentlich zugänglichen Cyberspacesysteme stehen: In Spielhallen.

Einem ähnlichen Konzept folgend, nur technologisch weiterentwickelt war das System "Aspen" aus den späten 70er Jahren, das von einem Team am Massachusetts Institute of Technology (MIT) entwickelt wurde. Angesichts der damals noch unzureichenden Leistungsfähigkeit der Computer basierte dieses System auf der Bildplattentechnologie, einem parallel zur Videotechnik entwickelten System zur analogen Speicherung von audio-visuellen Daten.

Auf Bildplatten wurde von dem amerikanischen Wintersportort Aspen jede Straße, jedes Haus und von einigen Häusern die Inneneinrichtung gespeichert. Der Benutzer des Bildplattensystems konnte so durch die Stadt "fahren". Bei "Aspen" saß der Betrachter vor einem Monitor und bestimmte über die Steuerung des Bildplattensystems seinen Weg durch die Stadt. Bereits in den späten 70er Jahren entstanden auf den beschriebenen Wegen erste hochdetaillierte Stadtmodelle, wie Aspen. Weitere Modelle folgten, so dass der heutige Stand das Ergebnis einer nahezu 25-jährigen Entwicklung ist.

2.1 Stand der Wissenschaft

Zur Darstellung des Standes der Entwicklung von 3D-Stadtmodellen ist es notwendig, objektbildende Verfahren näher zu betrachten. Dabei stehen die Objektklassen Gebäude und Vegetation im Vordergrund. Wichtige Verfahren seit 1995 werden näher betrachtet. In diesen Arbeiten wird zum Teil nur in Nebenaspekten auf die Generierung von Objekten eingegangen.

2.1.1 Verfahren zur geometrischen Rekonstruktion von Gebäuden

Beim klassischen photogrammetrischen Verfahren befinden sich die Gebäudegrundrisse auf einer Ebene und die Gebäude „wachsen quasi aus der Geländeoberfläche heraus“. Aus Luftbildern wird ein Stereomodell erstellt. Über aufgenommene Dachpunkte werden die Dachflächen erstellt. Durch das Einspielen der Grundrisse in das Stereomodell liegen Anhaltspunkte vor, wo ein Dach zu messen ist und wie gut Traufekante und Grundriss eines Gebäudes zusammenpassen. Anschließend werden die ermittelten Daten einer Gestaltungssoftware zur Weiterverarbeitung übergeben.

Zusätzlich zu diesem konservativen Ansatz wurden insbesondere in den Jahren seit 1995 verschiedene Verfahren mit unterschiedlichen Ansätzen zur geometrischen Rekonstruktion und teilweise zur Texturierung von Gebäudeobjekten entwickelt. Im folgenden wird eine Auswahl dieser Verfahren näher erläutert (Abb. 8). Die Bezeichnung erfolgt nach den Verfahrensauctoren und dem Veröffentlichungsjahr.

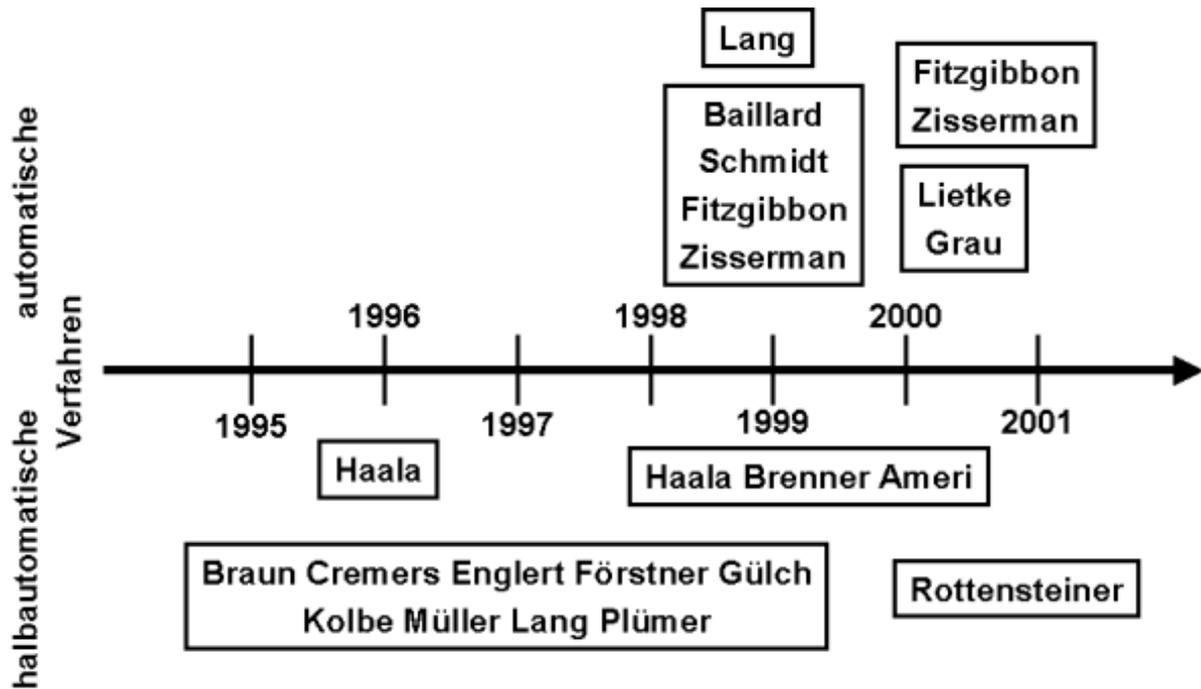


Abb. 8: Zeitliche Einordnung von Verfahren zur Gebäuderekonstruktion

Haala 1996

Das Verfahren, das in [HAALA 96] beschrieben wird, wird durch Filteransätze geprägt. Zunächst wird aus einem Stereomodell ein Digitales Oberflächenmodell (DOM) erzeugt, aus dem einzelne Strukturen als mögliche Elemente von Geländeobjekten extrahiert werden. Zudem wird aus dem DOM eine Tiefenkarte zwischen den Bildern des Stereobildpaares gerechnet. Mit einem Filter werden die 2D-Liniensegmente zu 3D-Segmenten gruppiert. Anschließend werden die dreidimensionalen Segmente zu parallelen oder rechteckigen Primitiven gruppiert.

Die Lage und Form der Rechtecke wird analysiert und zur Erkennung der Dachform genutzt. Dabei werden Rechtwinkleigenschaften, Größen und Neigungswinkel über Kenngrößen eingebracht. Der Ansatz erfordert ein DOM guter Qualität. Er greift in dünn besiedelten Gebieten (Segmentierung). Seine einzige mögliche Dachform ist das Satteldach.

Haala und Brenner 1999

Der Ansatz von [HAALA ET AL 99, BRENNER 00] beginnt ebenfalls mit der Zerlegung in ROI mittels eines DOM, allerdings wird dieses aus Laserscannerdaten gewonnen. Ebenfalls aus den Laserscannerdaten (Standardabweichung $< 1\text{m}$) erhält man grobe Gebäudemodelle. Diese werden gemeinsam mit den Grundrissdaten analysiert um einfache 3D Grundelemente zu erkennen und festzulegen aus denen im nächsten Schritt die Gebäude



Abb. 9: Ergebnis aus dem Ansatz von BRENNER 00 in BRENNER & HAALA 00

zusammengesetzt werden. Mögliche Dachformen sind Flach-, Sattel- und Walmdach. Anschließend wird ein Verfahren zur automatischen Texturgewinnung vorgeschlagen.

Die in Abb. 9 zu sehende Szene stellt ein Ergebnis dieses Verfahrens dar. Die Gebäude wurden mit terrestrisch gewonnenen Texturen versehen, indem im Bild die Textureckpunkte manuell bestimmt, der Bildausschnitt perspektivisch entzerrt und schließlich die so gebildete Textur mit dem korrespondierenden Polygon verknüpft wurde. Erkennbar sind die Schwierigkeiten welche bei terrestrisch gewonnenen Texturen auftreten. Besonders die Ansicht verdeckende Objekte sind hinderlich, da sie (in der Abb. 9 im wesentlichen Vegetation) mit auf die Objektoberfläche aufgetragen werden und damit Objektbestandteil werden.

Ameri 2000

Ein Verfahren zur automatischen Erfassung von Gebäudedächern beschreibt [AMERI 00]: Zunächst wird eine Segmentierung in ROIs nach [BRENNER 00] vorgenommen, um Bereiche mit einzelnen Gebäuden zu erhalten. Durch kleinste Quadrate Schätzung werden ebene Bereiche und damit Dachflächen erkannt. Die Dachflächen werden dann mittels Parameterschätzung in den dreidimensionalen Raum übertragen und es entstehen dreidimensionale Dachpolygone.

Anschließend werden die topologischen Beziehungen bestimmt. Über Voronoi-Diagramme entstehen Nachbarschaftsbeziehungen zwischen den Dachelementen. Die fertigen Dächer dienen dann als Grundlage für ein grobes Gebäudemodell. Eine Texturierung der Gebäude ist nicht vorgesehen. Das Verfahren ist im Programmpaket Poly-Modeler realisiert.

Baillard, Schmidt, Fitzgibbon und Zisserman 1999

Bei dem in [BAILLARD ET AL 99] vorgestellten Ansatz 'Automatische 3D Gebäuderekonstruktion aus Mehrbildansichten', der im Rahmen des IMPACT-Projektes entwickelt wurde, werden mindestens 4, besser jedoch 6 Bilder mit starker Überdeckung betrachtet. Aus den einzelnen Bildern werden mittels Canny-Operator und Linienschätzung Liniestücke extrahiert. Diese werden dann mit Merging- und Growing-Algorithmen verbessert. Es folgt die Bestimmung der Halbebenen links und rechts der Linien.

Die Gruppierung und Vervollständigung der 3D Linien wird über Verschmelzung kollinearer Halbebenen und Gruppierung koplanarer Linien und Halbebenen erreicht. Durch Ebenenschnitte entstehen Linien, die mittels Kantendetektion zuvor nicht erkannt wurden. Die Grenzen der Ebenen werden ermittelt, die Ebenen verbunden und überstehende Liniestücke abgeschnitten.

In einer Überprüfung der Daten werden Ebenen mit nur einer Grenzlinie verworfen, eine radiometrische Ähnlichkeitsprüfung über alle betroffenen Bilder durchgeführt und widersprüchliche Ebenen unterdrückt. Mit diesem Verfahren werden Dachflächen mit einer Standardabweichung von 2° - 5° erkannt. Es werden nur Bilder als Ausgangsdaten benötigt.

Fitzgibbon und Zisserman 2000

In einer Weiterentwicklung des Verfahrens von [BAILLARD ET AL 99] entstand das VHS2VRML Verfahren [FITZGIBBON ET AL 00] zur vollständigen Wiederherstellung einer 3D-Szene. Als Eingangsdaten werden die Aufnahmen einer Videokamera als Bildsequenz von 25-30 Bildern verwendet. Zusätzlich zu Linien werden Ecken detektiert. Die vorher betrachteten Bildtriplets werden als multiple Triplets betrachtet.

Schließlich entsteht ein texturiertes VRML Modell, wobei die Texturen als Bildextrakt mit anschließender Affin-Transformation entstehen. Die Wahl der Textur geschieht durch Auswahl aus allen Bildern, in denen die jeweilige Ebene vollständig zu sehen ist. Als Texturquelle wird dann das Bild gewählt, welches die größte Frontfläche zur Kameraposition bietet.

Braun, Cremers, Englert, Förstner, Gülch, Kolbe, Müller, Läbe, Lang, Plümer 1995-1999

Das Softwarepaket ObeX [GÜLCH ET AL 1999] des Instituts für Photogrammetrie an der Universität Bonn ist für die semi-automatische Generierung eines 3D-Stadtmodells gedacht. Es wurde beginnend mit der Entwicklung von HasE [BRAUN ET AL 95], HasE+ [ENGLERT ET AL 96, GÜLCH 96] und anschließend ObeX [LANG ET AL 96] bis zur Version 1.0 [FÖRSTNER 99] entwickelt und immer weiter verbessert.

Zum Verfahren: Ein Operateur wählt zunächst ein Gebäude aus einem Luftbilddatensatz aus und bestimmt die parametrisierte Grundform. Diese wird dann mittels eindeutiger Punkte quasi über den Datensatz gestülpt und damit das Modell erstellt. Für ein Satteldach sind beispielsweise 3 Messpunkte erforderlich (1 Bodenpunkt, 2 Giebelpunkte). Weitere Merkmale des Systems sind die Nutzung eines intuitiven Drahtgittermodell-Cursors sowie die automatische Höhenmessung und Parameterschätzung der Primitive.

Durch das Verfahren der Unterstützung der manuellen Messung durch automatische Parameterbestimmung wird ein zügiges und somit wirtschaftliches Erfassen von dreidimensionalen Gebäudedaten ermöglicht. Einfach strukturierte Gebäude können in sehr kurzer Zeit (< 30 sec) erfasst werden. Komplex aufgebaute Häuser benötigen die Entwicklung konkreter Vorgehensstrategien.

Das Verfahren ist sehr gut geeignet, um blockförmige Gebäudestrukturen zu erstellen. Ein Fassadenkörper und ein Dachkörper ohne Texturierung entstehen. Das Verfahren ist hauptsächlich zur Geometrierfassung angelegt. Bei Beispielergebnissen mit Textur wird diese aus terrestrischen Aufnahmen gewonnen oder automatisch aus Luftbildern extrahiert. Die Arbeiten sind, durch Initiative von Gülch und Müller, Basis für das Produkt Inject der Fa. Inpho [[INPHO](#)].

Lang 1999

In einem weiteren Verfahren, der komponentenbasierten Gebäuderekonstruktion [LANG 99], wird ein automatischer Ansatz beschrieben. Voraussetzung sind sich mehrfach überlappende Bilder des Extraktionsobjektes. Daraus werden mittels Unterstützung durch Fremddaten (DHM, Farbinformation oder GIS-Daten) Objekte detektiert.

Dem Erkennen schließt sich die Rekonstruktion an. Dazu wird zunächst der zweidimensionale Bildraum betrachtet. Punkte, Kanten und Bereiche sind das Ergebnis der Anwendung der Feature-Extraktionssoftware FEX [FUCHS ET AL 98] auf den Datensatz. Aus den Punkten und Kanten werden unter Verwendung von Kolinearität und Koplanarität Eckpunkte erkannt. Diese Eckpunkte sind unter Ausnutzung einer Gebäudeecken-Struktur bestimmten Gebäudeecken zugeordnet. Aus den Gebäudeecken werden Gebäudebestandteile (Terminatoren und Konnektoren) gebildet, welche schließlich zusammengesetzt die Gebäuderekonstruktion ergeben.

Das Verfahren ist hochautomatisiert und hocheffizient. Allerdings ist es notwendig, dass alle Gebäudeecken in den Bildern eingesehen werden können. Dabei kann es zu Störeinflüssen durch die Fassadentextur kommen [LANG 99]. Die Texturierung der Objekte wird nicht behandelt.

Liedke und Grau 1997-2000

Der Ansatz von [Liedtke 97, Grau 00] basiert auf AIDA, einem automatischen wissensbasierten Bildinterpretationssystem, welches beispielsweise auch in der Fernerkundung Anwendung findet. Bei diesem Ansatz wird zunächst mit einem stereoskopischen Tiefenschätzverfahren eine Tiefenkarte aus den Messbildern geschätzt. Dann werden gerade Konturlinien aus den Kamerabildern extrahiert. Unter Verwendung der Segmentierung wird schrittweise eine Bedeutungszuweisung durchgeführt. Dazu werden Hypothesen erstellt, die anschließend falsifiziert werden. Bei Auftreten konkurrierender Mehrdeutigkeiten werden diese verfolgt, bewertet und schließlich ausgewählt. Dazu wird ein Restkostenabschätzungsverfahren verwendet.

Zur Texturierung werden Texturkarten verwendet, die automatisch aus den Ausgangsbildern gerechnet werden. Zunächst wird die Textur eines Objektteils mittels der erkannten Geometrie aus dem Ausgangsbild entnommen und perspektivisch entzerrt und qualitativ angeglichen. Hieraus entsteht eine für VRML 97 direkt interpretierbare Texturkarte. Durch Akkumulation aus Teilansichten können die Texturkarten auch bei teilweise verdeckten Objektteilen gewonnen werden.

Die vorgestellte Vorgehensweise setzt diffuses Licht für die Aufnahmenvoraus und die Körper müssen eine gewisse Texturierung aufweisen [GRAU 00]. Auch hier gilt, dass alle relevanten Gebäudekanten in den Bildern zu finden sein müssen.

Rottensteiner 2001

Das von [ROTTENSTEINER 01], [IPF Wien] vorgestellte Verfahren ist in seiner Funktionsweise dem ObEx-Verfahren sehr ähnlich, jedoch weitaus vielfältiger. Ein halbautomatischer Ansatz wurde bewusst gewählt, um die Fehleranfälligkeit der automatischen Verfahren zu vermeiden.

Bei diesem Verfahren wählt der Operateur ein geometrisches Primitiv aus einer Vielzahl von Möglichkeiten. Das Primitiv wird mittels Zuweisung eindeutiger Eckpunkte zur Luftbildsituation abgelegt. Mehrere Primitive bilden dann das Gebäude in CSG-Struktur (siehe Kap. 5). Dieses kann dann gespeichert werden. Eine Umwandlung in eine B-Rep.- Struktur (siehe Kap. 5) kann dann durchgeführt werden. Das Verfahren ist Bestandteil des Programmpaketes Orient unter der Oberfläche Orpheus und hat eine Schnittstelle zu SCOP.TDM.

Weitere Verfahren und Vorgehensweisen

Die hier vorgestellte Übersicht erhebt keinen Anspruch auf Vollständigkeit. Zu weiteren Forschungsgruppen und Ergebnissen führen beispielsweise die Internetverweise [CMU], [CVL], [CVRL], [EPFL], [DCS], [IPF KARLSRUHE], [UCL GEOMATIC ENGINEERING] oder [VISICS].

2.1.2 Verfahren zur Baumextraktion

Zur Extraktion von Bäumen aus Luftbildern gibt es einige publizierte Verfahren. In [MAYER 98] ist dazu eine umfassende Übersicht etwas älterer Methoden zu finden. Andere Verfahren sind in [HAALA ET AL 99] dargestellt. Drei weitere, neuere und erfolgversprechende Vorgehensweisen werden hier vorgestellt. Zukünftig ist mit weiteren Fortschritten in der Laserscannertechnologie und ihrer Kombination mit der Photogrammetrie zu rechnen, so dass auf diesem Wege weitere, hochautomatisierte Verfahren zur Vegetationsextraktion und Artenbestimmung zu erwarten sind.

Haala und Brenner 1999

Ein DOM und Multispektraldaten aus Farb-Infrarot Luftbildern werden von [HAALA ET AL 99] zur Extraktion von Baumdaten kombiniert. Zunächst wird durch Reduktion des DOM durch ein Digitales Geländemodell DGM ein normalisiertes DOM erstellt.

Dieses ist Grundlage für eine automatische Klassifizierung. Zunächst werden über Parameter Pixelbereiche gebildet. Bereiche, die weniger als eine vorgegebene Pi-



Abb. 10: Ergebnis der Vegetationsextraktion aus [HAALA ET AL 99]

xelzahl beinhalten, werden aufgelöst. Bereiche werden geteilt, wenn sie einen bestimmten Schwellwert überschreiten oder sie werden verschmolzen, wenn die Entfernung zwischen zwei Bereichen unter eine vorgegebene Grenze sinkt.

Das Ergebnis ist die Baumstammposition als Bereichsmittellage, sowie der Bereichsrand als Kronenumrisspolygon (Abb. 10). Als Baumhöhe wird der höchste Punkt des normalisierten DOM dieses Bereiches angenommen. Über die Baumart, Kronenhöhe bzw. Stammhöhe oder Stammdurchmesser können mit diesem Verfahren keine Angaben gemacht werden.

Mayr, Mayer, Bacher und Ebner 99

In [BACHER ET AL 00] und [MAYR ET AL 99] werden zwei innovative Verfahren zur Identifikation von Vegetationsobjekten, speziell für einzelstehende Bäume vorgestellt.

Im ersten Verfahren werden Bäume als allgemeine, umrissbestimmte Ellipsoide im DOM betrachtet. Dazu werden Erkenntnisse aus einem Vegetationsmodell mit den Informationen aus einem Umgebungskontext (Bodenart, Klima, Höhenzone usw.) kombiniert. Weiterhin gehen Informationen aus der geometrischen und der radiometrischen Eigenschaft in das Modell ein.

Zunächst wird die Position der Baumobjekte aus dem DSM bestimmt. Aus den anderen Informationsquellen wird dann die Baumart (Laub- bzw. Nadelbaum) bestimmt.

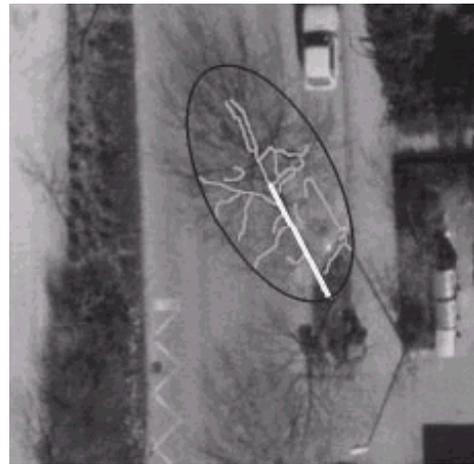


Abb. 11: Bestimmung von Baumparametern laubfreier Bäume aus [MAYR ET AL 99]

Das zweite Verfahren dient zur Bestimmung laubfreier Laubbäume in Luftbildern hoher Auflösung. Zunächst wird die Schattenprojektion des Stammes unter Zuhilfenahme der Kenntnis des Sonnenstandes unter Verwendung der Hough-Transformation detektiert. Anschließend werden die Äste mit dem Stamm als Startpunkt ermittelt. In Abb. 11 sind die durch das beschriebene Verfahren ermittelten Informationen in das Luftbild eingeblendet.

Mit dieser Vorgehensweise werden nicht nur die Position bestimmt, sondern auch Attribute wie Kronenform, Kronenhöhe, Stammhöhe und Stammdurchmesser können abgeleitet werden.

2.1.3 Verfahren zur Gestaltung von Geo-Objekte in Städten

In verschiedenen Verfahren werden Gebäude- bzw. Baumobjekte entweder . Bei diesen Verfahren werden zum Teil bereits bestehende Datenbestände für die Gestaltung verwendet, die Aspekte der Datenerfassung spielen nur eine untergeordnete Rolle.

Gleixner, Kranjec, Ranzinger 1995-1998

Das Verfahren GO3DM (GIS-Objekte in 3D-Modellen) wurde von der Fa. Grintec, Graz [GLEIXNER ET AL 95] im Zuge eines Projektes mit dem Magistrat der Stadt Graz entwickelt und 1998 vorgestellt. Die weitere Entwicklung des Produktes ist zumindest vorläufig eingestellt, da sich das Unternehmen inzwischen mit anderen inhaltlichen Schwerpunkten befasst. GO-3DM basiert auf bestehenden GIS-Datenbeständen. Als wesentliche Datenbestände sind zu erwähnen:

- Katasterdaten
- Naturbestandsdaten
- Dachausmittlungen

In größeren Städten finden sich oft flächendeckend Katasterdaten, des weiteren werden große Anstrengungen unternommen, den Naturbestand im öffentlich zugänglichen Straßenraum terrestrisch zu erfassen. Die übrigen Bereiche werden photogrammetrisch ausgewertet. Dachausmittlungen werden projektbezogen erfasst.

Es handelt sich in der Regel um auf die Ebene projizierte Punkt-, Flächen- und Linienobjekte, wobei die zugrundeliegende Geometrie in vielen Fällen dreidimensional vorliegt. In die Erfassung und Laufendhaltung dieser Daten wurde und wird sehr viel investiert. Es ist daher naheliegend zu versuchen, diese Daten auch als Grundlage für 3D-Stadtmodelle zu verwenden.

Mittels GO-3DM werden aus den oben beschriebenen GIS-Daten (die Konventionen für die Daten in Name, Typ und Layer sind strikt einzuhalten) automatisch in einem an Style-Sheets erinnernden Konvertierungsprozess 3-dimensionale Objekte modelliert. Folgende Objekte können automatisch generiert werden: Höhenmodell, Gebäude mit Dächern, Verkehrszonen, Grünzonen und Bäume sowie Spezialobjekte, wie z.B. Verkehrszeichen oder Zäune. Wie detailliert das Modell ist, hängt von Art und Umfang der vorhandenen Ausgangsdaten ab.

Die aus den GIS-Daten eines Gebietes erzeugten 3D-Objekte werden als 3D-Flächen abgespeichert und können mit Texturen der ganzen Fassade aus terrestrischen Fotografien versehen werden. Die Ergebnisdatei wird dann Animationspaketen oder Werkzeugen zur interaktiven Visualisierung zur Verfügung gestellt. GO-3DM verwendet Standard DXF-Files zur Ein- und Ausgabe. Die Ergebnisdaten können daher mit einer Vielzahl von Standardpaketen weiter bearbeitet werden.

GO-3DM wird nur im Zusammenhang mit Dienstleistungen und Projekten der Fa. Grintec verwendet und ist ansonsten nicht erwerbbar.

Henricsson, Grün, Wang 1996-2000

Mit dem Cybercitymodeller existiert ein umfassendes Verfahren von der Datenerfassung bis zur Objektbildung. Dabei entstehen hochgenaue Abbildungen der Realität.

Der CyberCity Modeller ist ein aus unterschiedlichen Arbeiten an der ETH Zürich entwickeltes Produkt. Begonnen hat die Entwicklung mit dem TOBAGO - (Topology Builder for the Automated Generation of Objects from Pointclouds) [HENRICSSON ET AL 96] und dem ARUBA (Automatic Reconstruction of Sub-Urban Buildings from Aerial Images) - Verfahren [HENRICSSON ET AL 96] als topologisches Modellierungsverfahren. Auf diesem basierend wurde das Programmpaket CyberCityModeller entwickelt [GRÜN ET AL 98, 99].

Der Operateur misst an einem Stereoauswertegerät (Analytischer Plotter oder Digitale Auswertestation) die Hausdächer in Form einer 3-D Punktwolke. Die Reihenfolge dieser Punkte ist beliebig. Ein einzelnes Dach sollte klar als solches gekennzeichnet sein. Die Bildung eines Punktclusters wird durch das Setzen eines Schlusszeichens erreicht. Als Dacheinheit sind ein als abgeschlossene Punktwolke gemessenes volles Dach oder der Teil einer komplexeren Dachkonstruktion, welche als Modell im Objektartenkatalog existiert. Strukturell komplexe Dächer müssen vom Operateur in Untereinheiten aufgebrochen werden.

Die Anzahl der gemessenen Dacheinheiten pro Zeiteinheit wird im wesentlichen von folgenden Faktoren bestimmt:

- Komplexität der Dachformen (Punktzahl)
- Generalisierung- (Detaillierungs-) Niveau
- Verdeckungen durch andere Objekte
- Bildgeometrie und Bildqualität

Die automatische Strukturierung läuft dann in zwei Stufen ab. Zunächst wird ein Klassifizierungsalgorithmus initialisiert, der die Anzahl der Firstpunkte eines Daches erkennt. Der Classifier, der auf einer Histogrammanalyse der Höhenwerte der Dachpunkte beruht, gibt seine Ergebnisse an den K-Parser ab. Dieser bestimmt die Topologie der Firstpunkte und analysiert, ob ein gegebener Datensatz strukturierbar ist.

Verworfenen Punktwolken werden dem Operateur gemeldet, damit dieser Korrekturmaßnahmen einleiten kann. Das erfolgreich analysierte Dach wird einer Klasse von Modellen zugeordnet, wobei sich die Klassenzugehörigkeit eben nach der Anzahl der Firstpunkte richtet. Innerhalb dieser Klassen gibt es wieder eine Anzahl von Hausmodellen. Dachüberstände entstehen durch Verschneiden mit Grundrissdaten. Die Daten werden im eigenen V3D-Format gespeichert.

Die Texturen der Dächer können automatisch unter Verwendung der Dachumringpolygone aus den Luftbildern entnommen werden. Fassadentexturen entstehen durch terrestrische Objektfotos. Generische Texturen sind möglich. Allerdings können nur ganzfassadige Texturen verwendet werden. Als Geländetextur werden Orthofotos eingebracht.

Der CyberCityModeller ist als Softwarepaket erhältlich. Die Entwicklung wird zwischenzeitlich von der CyberCity AG (Spin-Off der ETHZ) weiterverfolgt. Das Programmpaket wird dabei durch die Visualisierungssoftware TerrainView und CityView ergänzt. Der Viewer für den CyberCityModeller, CityView, der Bestandteil des Paketes ist, enthält die Möglichkeit, Gebäude in den drei Detailstufen Bounding Box, Dachformen sowie Detailstrukturen/Texturen darzustellen.

Virtual Terrain Project

Das recht weit gediehene Virtual Terrain Project [[VTP](#)] soll dennoch näher betrachtet werden. VTP dient als Sammelbecken von Programmen und Verfahren, die dazu dienen, künstliche Landschaften zu schaffen. Es wird versucht diese unter einem Dach zu einem Ganzen zu integrieren. Darunter befindet sich auch eine Vorgehensweise, die sich mit der Bildung von Gebäudeobjekten befasst. In der Sammlung ist es unter anderem möglich, einfache Gebäudeobjekte zu definieren. Parametrisiert sind Lage, drei verschiedene Grundrissformen (Rechteck, Kreis, Polygon), Wandfarben, zwei Dachformen (Sattel- und Walmdach) sowie Stockwerkweise Türenzahl, Fensterzahl, Materialart. Weitere Elemente sind in einer Wunschliste aufgeführt.

Parish, Müller 2001

In einem anderen Projekt [[PARISH ET AL 01](#)] wird versucht, Stadtmodelle mittels L-Systemen (siehe Kap. 4) zu erzeugen, indem das Wachstum der Stadt nachvollzogen wird. Dabei werden, aufbauend auf einer Art Nukleus, Straßenachsen bestimmt, an deren Rändern die Bebauung errichtet wird. Einige Objektarten, wie beispielsweise Gewässer, können mit realen Bestimmungsgrößen eingegeben werden. Dieses Verfahren mag vielleicht in anderen Regionen der Welt brauchbare Ergebnisse erzielen. Für europäische Verhältnisse jedoch, wo üblicherweise keine auf dem Reißbrett geplante Strukturen vorherrschen, ist dieses Verfahren sicherlich nicht realistisch. Die Visualisierung der Ergebnisse ist aber dennoch recht beeindruckend.

Es gibt viele verschiedene Werkzeuge, die mehr oder weniger in der Lage sind, dreidimensionale Daten zu präsentieren. In der herstellerneutralen Marktübersicht [[US Army TEC](#)] über die meisten dieser Produkte, sind aktuelle Informationen zusammengestellt. Dort sind Hersteller, Produktbeschreibungen, Bezugsquellen und Preise der gängigen zu finden.

2.2 Zu bearbeitende Aufgaben

In dieser Arbeit geht es um parametrisierte Musterobjekte für Bäume und Gebäude. Mithilfe dieser Objekte lassen sich vielfältige Exemplare erzeugen. Die Exemplare entsprechen zumeist nicht genau ihrem Vorbild, der Wirklichkeit, sondern bilden eine Näherung. Es wird versucht, unter dem Gesichtspunkt der Aufwandsminimierung ein weitgehend allgemeingültiges Muster

zu finden, das jedoch möglichst viele Varianten zulässt. Der Aspekt der Datenerfassung wird dabei vernachlässigt, es wird auf bestehende Datenbestände zurückgegriffen.

Das hier erarbeitete Verfahren bietet 16 Dachtypen an und deckt damit wesentliche Teile und Strukturen ab. Es lassen sich leicht weitere Dachformen durch Parameter beschreiben, um das Verfahren zu erweitern.

Ein weiterer Ansatzpunkt ist die Bedienbarkeit der Software. Sollen 3D Stadtmodelle für den ländlichen Raum geschaffen werden, beispielsweise für Dorferneuerungs- bzw. Flurbereinigungszwecke, so führt der Einsatz vorhandener Softwarepakete, welche durch Experten bedient werden müssen, zu erheblichen Folgekosten. Hier werden Wege gefunden, mit denen z.B. vor Ort ganz leicht oder unter Anleitung aussagefähige kleine Stadtmodelle, sozusagen Dorfmodelle erstellt werden können. Dies gelingt über parametrisierte Dachformen, indem man Formulare für die Dateneingabe z.B. durch die Hausbesitzer erstellt. Das in dieser Arbeit vorgestellte Verfahren versucht diesen kosteneffizienten Weg einzuschlagen.

Bei historischen Ensembles gibt es andere Probleme. Oft existieren nur zeitgenössische Darstellungen als Zeichnungen, oder Bilder, die nicht photogrammetrisch messbar sind. Dennoch lassen sich aus diesen Daten Angaben für ein Modell ableiten, die Eingang in die Objektmuster dieser Arbeit finden.

In der 3D Darstellung ist neben der geometrischen Repräsentation ein weiterer Faktor wichtig: die Textur. Die beschriebenen Verfahren können in diesem Punkt nicht überzeugen. Die Ergebnisse reichen von stark verzerrten Texturen aus Schrägluftbildern bis hin zu aus terrestrischen Fotografien gewonnenen aus Stücken zusammengesetzten Komplettfassaden. Letztere setzen ein rundum begehbare und fotografierbare Objekt voraus, das eher selten vorliegt. Der in dieser Arbeit vorgeschlagene Ansatz verwendet eine Texturbibliothek auf Kachelbasis, die sukzessive erweitert werden kann und für die es bereits ausreicht, wenn kleinere Bereiche der Fassade sichtbar sind.

Nach Aussagen einer der Hauptnutzergruppen von 3D Stadtmodellen, den Funknetzbetreibern, ist eine Lageunsicherheit der Lage von Einzelobjekten im Mittel von 1-2 m zulässig. Diese wird mit den vorgestellten Verfahren leicht erreicht. Dieser Nutzergruppe (persönliche Gespräche mit Herrn Dipl.-Ing. B. Ruff, T-D1 und Herrn Dipl.-Ing. B. Blaschke, Viag Interkom) macht allerdings ein anderes Problem zu schaffen: die Fassadenstruktur. Deren Rauigkeit hat auf die Ausbreitung von Funkwellen insbesondere auf die Reichweite wesentlich mehr Einfluss als etwa eine genauere Objektposition. Gleiches gilt sinngemäß für Schallausbreitungsmodelle.

Im zuletzt Genannten versagen die vorgestellten Verfahren. Die einzigen Quellen, aus denen diese Informationen beziehbar sind, sind die Fassadentexturen, die jedoch in den bereits beschriebenen Verfahren nur eine untergeordnete Rolle spielen. Die im Folgenden vorgestellten Verfahren zur Texturierung bieten Abhilfe, in dem über die Textur die Rauigkeit dargestellt werden kann.

Eine weitere Aufgabe ist die Gestaltung von generalisierten Objektrepräsentationen. Diese sind notwendig, um in weiterer Entfernung liegende Objekte mit geringerem Rechenaufwand darzustellen. Vorteilhaft sind hier festgelegte Regeln, die aus dem vorliegenden Datenbestand automatisch Repräsentationen mit geringem Detailgrad erzeugen.

Kapitel 3

DATENQUELLEN, INFORMATIONEN UND KONZEPTE

Ein erheblicher Anteil der für ein Stadtmodell notwendigen Daten speziell die Geometriedaten, wird mit den in Kap. 2 beschriebenen Verfahren ermittelt. Mit weiteren Daten, Datenquellen und Datenformaten entsteht die Möglichkeit, die Geometrie zu ergänzen oder zu ersetzen.

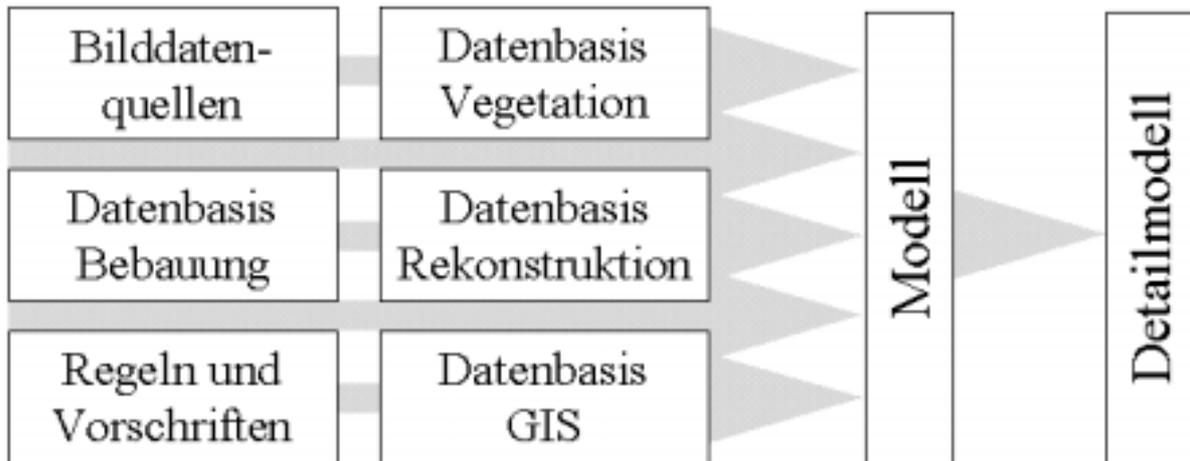


Abb. 12: Übersicht über Daten und Modelle

Abb. 12 gibt eine Übersicht über die Datenquellen aus denen das Modell und die Detailmodelle abgeleitet werden. Die Datenquellen bestehen aus Bebauungs- oder Vegetationsdaten, Rekonstruktionsdaten anderer Verfahrensweisen und aus GIS-Daten. Hinzu kommen die Bilder aus der Texturbibliothek. Diese können in ein File-System oder einer Bilddatenbank abgelegt sein. Ergänzt werden die Daten durch Regeln und Vorschriften, die in einer Wissensbasis abgelegt sind und die dafür sorgen, dass aus den Daten nur sinnvolle Objekte generiert werden.

3.1 GIS-Daten der Vegetation und der Bebauung

Für die Modellierung sind verschiedene Daten notwendig. Eine Möglichkeit der Datengewinnung ist der Neuerfassung. Eine elegantere Möglichkeit besteht darin, bereits vorhandene Daten, wie sie in GIS vorliegen, zu verwenden. Beispiele für vorliegende Daten werden im Folgenden näher betrachtet.

3.1.1 Daten der Vegetation

Die durch ein Gartenamt bzw. Forstamt oder einer ähnlichen Institution erhobenen Daten beinhalten in der Regel, neben Daten über Anlagen, wie Spielplätze insbesondere genaue Daten über die Vegetation. Oft hat ein Gartenamt einige Maße und Attribute von Einzelbäumen für die Durchführung von Arbeiten im Baumkataster gespeichert.

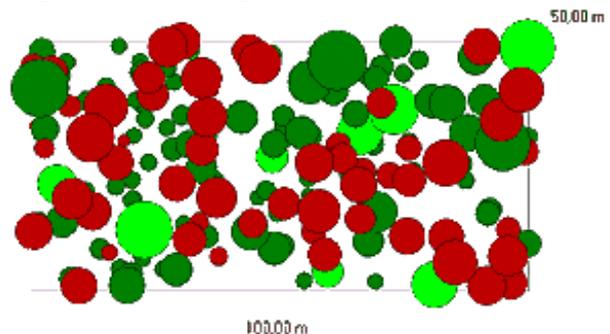


Abb. 13: Baumkronenkarte aus einem GIS nach [WWK]

In Abb. 13 ist ein Beispiel für einen solchen Datenbestand in Form einer Baumkronenkarte dargestellt. Aus diesen Daten lassen sich Baumart (Farbe), Lage der Baumachse (Mittelpunkt der Krone) sowie der Kronendurchmesser und damit in etwa das Alter bestimmen.

Diese Bestimmungsgrößen können für einen Einzelbaum sein:

- Ebene Position des Baumes in Koordinaten
- Baumfamilie, wie Laubbaum oder Nadelbaum, oder noch detaillierter die Baumart, wie Eiche, Buche, Fichte usw.
- Baumhöhe in [m]
- Baumalter in Jahren
- Kronendurchmesser in [m]

oder eher flächige Parameter für einen Wald, wie

- Baumdichte (siehe Abb. 14)
- Baumverteilung
- Mittlere Maße für Alter, Stammdicken oder Kronendurchmesser



Abb. 14: Baumdichte aus einem GIS nach [WWK]

Wesentlich für die Erstellung der Vegetationsdaten ist das Wissen um ihre Verschiedenartigkeit. Das bedeutet, dass je nach gefundener Situation mit unterschiedlichen Daten und Datenstrukturen zu rechnen ist, die jeweils situationsgerecht angepasst werden müssen.

3.1.2 Daten der Bebauung

Ein gewöhnlich vorliegender Datenbestand sind Katasterdaten. In Katasterdaten gibt es einige Datenklassen, die für die dreidimensionale Modellierung interessant sind.

Der Gebäudeumriss ist ein zweidimensionales Polygon. Er beschreibt die ein Gebäude umschließende Linie in Bodenhöhe. Die Information Gebäudeumriss beinhaltet die wesentlichen Umrissformen. Kleinformen unter ca. 10 cm Größe werden vernachlässigt. Verfahren zur Vereinfachung von Gebäudeumrissen sind in [MAYER 98] dargestellt.

Gebäudeumrisse werden vorwiegend durch terrestrische Verfahren bestimmt, da für eine photogrammetrische Auswertung die Dachüberstände hinderlich sind. Die Geschosshöhe ist eine erste Information zur Ermittlung der Gebäudehöhe. Liegen keine weitere Informationen vor, lassen sich mit der Geschosshöhe und dem Gebäudeumriss unter Zuhilfenahme einiger Regeln bereits einfache Modelle erstellen. Trotz der bestehenden Gebäudeeinmessungspflicht ist diese Informationsquelle häufig unvollständig. Auch fehlen oft die Informationen zur räumlichen Ausprägung, wie die Geschosshöhe.

3.2 Bilddaten, Formate und Komprimierung

Dreidimensionale Objekte haben neben geometrischen auch Materialeigenschaften. Diese sind beispielsweise die farbliche Gestaltung, als emittierende, spiegelnde Farben, Transparenz u.a. sein. Eine andere Form der Materialeigenschaft ist die Verwendung von Bilddaten anstelle von Farbe, den sogenannten Texturen. Dabei erfolgt eine Projektion vom Bildraum in den Objekt-raum. Das Bild wird auf diese Weise dem eigentlichen Objekt quasi aufgezogen. Durch geschickte Wahl der Bildkoordinaten können leichte Bildverzerrungen ausgeglichen werden.

Für Texturen werden üblicherweise die Bilddatenformate Jpeg, Gif und PNG verwendet. Die Formate unterliegen aufgrund der spezifischen Formatausprägungen Randbedingungen, auf die im folgenden näher eingegangen wird.

Kaum ein aktuelle Bilddatenformat verzichtet auf eine Reduzierung der Speichervolumens, der Kompression. Bei der Kompression wird zwischen verlustfreien und verlustbehafteten Verfahren unterschieden. Beim verlustfreien Komprimieren bleiben alle Details im Bild erhalten. Hier werden die vorhandenen Daten zumeist durch Verwendung von Lauflängenkodierung [KOMPRESSIONSALGORITHMEN] effizient gespeichert. Dabei wird üblicherweise weniger Speicherplatz als beim verlustbehafteten Komprimieren gespart. Verlustbehaftetes Komprimieren bedeutet, dass Details, die das menschliche Auge nicht oder kaum wahrnimmt, weggelassen werden, um das Datenvolumen zu reduzieren. Je mehr Informationen aus dem Bild entfernt wird, desto stärker leidet aber die Bildqualität.

Verlustbehaftete Kompressionsverfahren arbeiten in drei Schritten: Transformieren, Quantisieren und Kodieren. Wegen des vorgeschalteten Transformierens werden die Programme Transformationskodierer [KOMPRESSIONSALGORITHMEN] genannt. Beim Transformieren werden die Bilddaten in Anteile unterschiedlicher Frequenzen aufgeteilt. Schwingungen niedriger Frequenz stellen unscharfe Bildelemente wie etwa Hintergründe dar. Details werden durch hohe Frequenzen beschrieben. Durch Berechnung der Anteile der Frequenzen am gesamten Signal erhält man eine Liste von Zahlenwerten. Obwohl alle Koeffizienten gleichen Anteil an der Bildqualität haben, tragen niedrige Werte wenig zur Wahrnehmung des Bildinhaltes bei, während hohe Werte wichtige Details repräsentieren. Die Koeffizientenliste stellt das Ausgangsmaterial für das Kodieren dar.

Damit später die komprimierten Daten wieder dekodiert, also die Zahlenwerte zurück in Bildinformationen umgesetzt werden können, muss das Transformationsverfahren umkehrbar sein. Das eigentliche Komprimieren erfolgt durch Quantisieren, das heißt Nähern oder Runden der transformierten Werte. Vereinfacht ausgedrückt werden alle Werte unterhalb einer bestimmten Grenze auf Null gesetzt, die darüber liegenden Werte werden ebenfalls (auf Ganzzahlen) gerundet gespeichert. Dadurch verringert sich die Bildqualität nur geringfügig, da alle Zahlenwerte gleich behandelt werden, vorausgesetzt, man wählt den Schwellwert, ab dem die Werte auf Null gesetzt werden, nicht zu hoch. Bei diesen Verfahren gehen hauptsächlich Hintergrundinformationen verloren. Je höher der Quantisierungsfaktor ist, desto gerundeter sind die Werte, die gespeichert werden müssen und sich dann auch durch weniger Bits darstellen lassen.

Beim anschließenden Kodieren kommt es darauf an, dass eine effiziente Speicherung der gerundeten transformierten Zahlen ermöglicht wird. Dazu gibt es eine Reihe von Verfahren: Das bekannteste ist auch bei der verlustfreien Komprimierung verwendete Lauflängenkodierung (Run-Length Encoding, RLE), die bei BMP- und Jpeg-Komprimierung sowie bei Wavelet-Verfahren eingesetzt wird. Beim Wavelet-Komprimieren können zusätzlich noch Zerotree und Kontextkonditionierung zum Einsatz kommen. Üblicherweise werden Mischformen aus verschiedenen Verfahren angewendet.

3.2.1 Joint Photographic Experts Group – Format Jpeg

Jpeg wurde ab Mitte der 80er Jahre von der Joint Photographic Experts Group erarbeitet und ist seit 1991 als ISO-Standard definiert. Das Format ist nach wie vor das wichtigste für Komprimierung von Bitmaps mit einer Farbtiefe über 256 Farben, da es sehr hohe Komprimierungsgrade anbietet. Es ist Bestandteil des VRML 97 ISO-Standards.

Jpeg arbeitet mit der diskreten Kosinustransformation (DCT). Dabei wird das Bild in quadratische Blöcke von 8 x 8 Pixel aufgeteilt (seltener 16 x 16). Kosinusfunktionen unterschiedlicher Frequenzen stellen jeden Block dar. Durch Aufteilen in kleine Blöcke lassen sich die Berechnungen beschleunigen. Bei größeren Blöcken steigt die Verarbeitungszeit überproportional an.

Ohne Aufteilen in Blöcke kann man den Zahlenwerten zwar noch ansehen, wie detailreich das Bild ist. Der Betrachter kann aber nicht mehr erkennen, an welchen Stellen sich die Details befinden. Das ist damit zu begründen, dass die DC-Transformation zwar eine Aufteilung in Frequenzanteile liefert, aber nicht angibt, an welcher Stelle des Bildes die jeweiligen Frequenzen auftreten. Die Positionsangaben verteilen sich über alle bei der Transformation ermittelten Einzelwerte. Das Ändern eines Wertes hat Folgen für das gesamte Bild.

Für die Kompression werden Quantisierungstabellen verwendet, die Erkenntnisse aus wahrnehmungsphysiologischen Untersuchungen berücksichtigen. Demzufolge nimmt das Auge Störungen in niederfrequenten Bildteilen stärker wahr als in hochfrequenten. Daher sorgt die vom Jpeg-Komitee empfohlene Standard-Quantisierungstabelle dafür, dass die Werte für die niederfrequenten Anteile mit höherer Präzision gespeichert werden. Die genaue Zusammensetzung der Tabelle ist allerdings frei wählbar. Bildbearbeitungsprogramme können sie an verschiedene Bildtypen anpassen. Daher ist die jeweils verwendete Quantisierungstabelle auch Bestandteil des kodierten Datenstroms.

Zur Kodierung nutzt Jpeg die Lauflängenkodierung. Dazu werden aufeinanderfolgende gleiche Werte durch die Angabe der Anzahl und des Wertes in Binärform dargestellt. Zum effizienteren Kodieren sind im Jpeg-Standard Kürzel definiert: So werden 15 aufeinanderfolgende Nullen eines Blocks durch ein Symbol ersetzt.

Das Ergebnis des Kodierens wird üblicherweise nach dem Huffman-Algorithmus nochmals komprimiert. Dabei kommen Codes variabler Länge anstelle einer festen Anzahl von Bits wie bei der statischen Lauflängenkodierung, zum Einsatz. Häufig vorkommenden Werten wird dabei ein kürzerer Code zugeordnet. Anstelle der Huffman-Kodierung kann die arithmetische Kodierung eingesetzt werden, die effizienter komprimiert, aber deutlich höheren Rechenaufwand erfordert. Beim Rücktransformieren werden die unterschiedlichen Kosinus-Schwingungen den transformierten und quantisierten Werten entsprechend überlagert. Stärkere Kompression führt zu sichtbaren überwiegend langwelligen Schwingungen. Das heißt, dass das rücktransformierte Bild nicht nur unscharf ist, sondern auch Blockmuster zeigt.

Nachteilig ist dass Blockartefakte besonders bei hoher Komprimierung sichtbar sind. Dabei weichen die Kanten benachbarter Bildteile deutlich erkennbar voneinander ab. Zu diesem Phänomen kommt es, wenn nur einige wenige langwellige Schwingungen den Bildinhalt eines Blocks darstellen und nebeneinander liegende Blöcke nicht mehr übergangslos aneinander passen. Bei mittleren und hohen Komprimierungsraten sichtbare Blockartefakte vermeidet der künftige Standard Jpeg 2000 dadurch, dass vor der Einteilung des Bildes in Blöcke erst eine Wavelet-Transformation durchgeführt wird. Abweichungen zwischen benachbarten Blöcken gleicht die Wavelet-Transformation aus.

Jpeg 2000 erlaubt auch das unabhängige Kodieren unterschiedlicher Bildbereiche, wodurch progressives Übertragen und Darstellen eines Bildes vereinfacht werden. Einzelne Bildausschnitte mit besonders wichtigen Informationen (engl.: Regions of Interest) lassen sich mit höherer Genauigkeit kodieren. Außerdem wird es bei Jpeg 2000 möglich, Bilder wahlweise verlustfrei oder mit Verlusten zu speichern. Es können Metadaten, beispielsweise Text zum Bildinhalt, Copyright-Vermerke oder Geokodierungen mit den Bildern gespeichert werden.

3.2.2 Graphics Interchange Format – Gif

Gif ist eins der am weitesten verbreiteten Bitmapformate. In Gif können nur 256 Farben aus 16 Mio. (8 Bit RGB)möglichen verwendet werden. Theoretisch ist auch die Verwendung von mehr als 256 Farben gestattet. Dies geschieht, indem das Bild in Regionen gleicher Farbpalette aufgeteilt wird. Praktisch gibt es kaum Bildbearbeitungssoftware, die dieses Konzept unterstützt. Zum Komprimieren dient der LZW-Algorithmus [[Kompressionsalgorithmen](#)], der Redundanzen durch sich wiederholende Pixelketten effizient ausnutzt. Gif ist wie Jpeg Bestandteil des

VRML ISO-Standards. Für dessen nächste Version die Ablösung von Gif durch PNG aus lizenzrechtlichen Gründen propagiert wird.

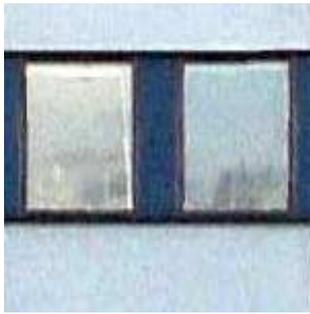


Abb. 15: Fenstertextur im Jpeg Format

Wegen der geringen Farbtiefe von 256 Farben eignet sich Gif nicht für Fotografien und ist deshalb zur Verwendung für Oberflächentextur nicht geeignet. Gif eignet sich jedoch hervorragend für Texturen die eine geringe Farbtiefe haben. Solche Texturen treten besonders bei Gebäuden auf.

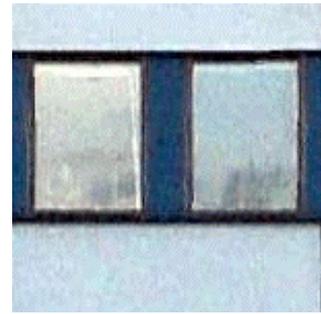


Abb. 16: Fenstertextur im Gif Format

In Abb. 15 ist ein Teil einer Fensterfassade im Jpeg-Format in Abb. 16 das gleiche Bild im Gif-Format dargestellt. In beiden Bildern ist kein Unterschied zu erkennen. Dies

liegt vor allem daran, dass diese Textur im wesentlichen von den Farbtönen schwarz und blau bestimmt wird. Deshalb werden im linken, Jpeg-Bild nur 10833 Farben (von über 16 Mio. möglichen) benutzt. Das bedeutet jeder Farbton im Gif-Bild ersetzt im Mittel weniger als 43 Farbtöne im Jpeg-Bild.

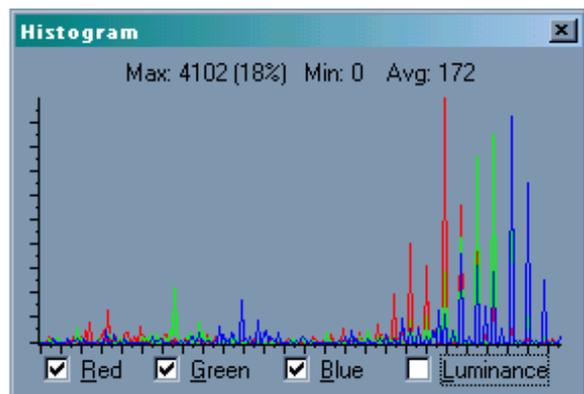
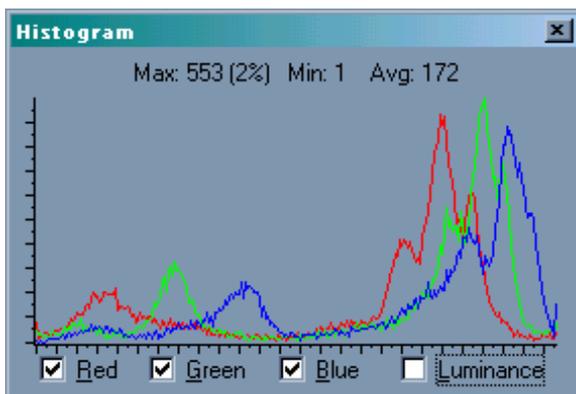


Abb. 17: Histogramme der Fenstertexturen des Jpeg-Bildes (links) und des Gif-Bildes (rechts)

Verwendet man das RGB-Konzept, so lassen sich, variiert man jeden Kanal um maximal 1 lassen sich bereits 26 verschiedene Farbtöne darstellen. Dies gelingt allerdings nur durch die Bündelung der Farben auf 2 Schwerpunkte, wie im Histogramm für das Jpeg-Bild in Abb. 17 links sichtbar. Im Histogramm Abb. 17 rechts, in dem die Werte für das Gif-Bild aufgeführt sind, wird deutlich, dass es sich bei der Umwandlung in das 256-Farben Bild um eine Konzentration um wenige Einzelwerte handelt. Allerdings ist man, unabhängig der die Farbtiefe betreffenden Argumente, auf das Gif-Format angewiesen, wenn transparente Pixel dargestellt werden sollen. In Gif kann nämlich ein bestimmter Farbwert als transparent bestimmt werden. Alle Pixel mit diesem Farbwert im Bild erscheinen dann durchsichtig.

Ein weitere Möglichkeit, die allein Gif bietet, sind Animationen. Dazu werden in einem einstellbaren Zeitabstand einmalig oder endlos eine Reihe von Bildern abgespielt. Im untenstehendem Beispiel (Abb. 18) ist die Bildfolge zur Darstellung der Fontäne eines Springbrunnens dargestellt.

PNG sollte Gif wegen immer wieder aufflammender Lizenzstreitigkeiten mit diesem Format als Standardformat im Internet ablösen, konnte sich aber nicht durchsetzen. Anders, als für das in die Internetbrowser integrierte Gif, benötigt man zur Anzeige von Png-Dateien zusätzliche Pro-

gramme, die seine Verbreitung bremsen. PNG unterstützt Truecolor, das Format eignet sich also für die Speicherung von Fotos. Es verwendet zum Komprimieren den LZ77-Algorithmus und komprimiert etwas effizienter als Gif, allerdings bei weitem nicht wie Jpeg. Auch PNG kann Transparenz darstellen. Hier wird ein eigener Kanal, der α -Kanal angeboten, mit dessen Hilfe die Transparenz in 256 Stufen reguliert werden kann.



Abb. 18: Bildfolge eines Springbrunnens als animiertes Gif in Einzelbildern

Die variable Transparenz ist das wichtigste Argument für die Verwendung des PNG oder Gif Formates. Diese Möglichkeit bietet Jpeg nicht, dafür bietet es Vorteile, wenn hohe Farbtiefe benötigt wird.

Die Vor- und Nachteile der Grafikformate müssen bei der Generierung von 3D-Objekten berücksichtigt werden. Die Texturen für Gebäude erfordern keine Transparenzeigenschaft. Andererseits spricht das schmale Farbspektrum der Gebäudetexturen eher für die Verwendung des Gif-Formates. Dennoch werden bei Gebäuden in der Regel die Texturen im Jpeg Format gespeichert. Für Bäume werden normalerweise Texturen mit Transparenz gefordert. Aus diesem Grund werden für Bäume, obwohl für sie höhere Farbtiefe vorteilhaft wäre, Texturen im Gif Format gespeichert.

3.3 Geometrie und Texturen aus Bilddaten

Bilder, die terrestrisch oder durch Schrägaufnahmen aus der Luft gewonnen werden, liegen normalerweise im Jpeg-Format vor. Aus ihnen werden dreidimensionale Oberflächenmodelle der Geometrie und Texturen ermittelt. Besonders eignen sich Objekte mit glatten Oberflächen, wie beispielsweise Gebäude für die photogrammetrische Gestaltung [DORFFNER ET AL 99].

Die Datengewinnung für die photogrammetrische Modellierung erfolgt vollständig manuell, weshalb sie nicht unter den halbautomatischen und automatischen Verfahren aufgelistet wurde, die unter dem Stand der Wissenschaft (Kap. 2) zu finden sind. Zudem ist es weitgehend ausge-reift, wie weit verbreitete kommerzielle Produkte wie [GEOMETRA] und [PHOTOMODELLER] zeigen. Es wird im Rahmen des in dieser Arbeit vorgestellten Verfahrens ergänzend verwendet, um geometrische Details zu ermitteln und originale Texturen zu gewinnen. Es wird also als Werkzeug benutzt, mit dem das Feinmodell (siehe Abb. 2) weiter ausgearbeitet werden kann oder aber um Formen, die im Modell nicht vorgesehen sind, besser gestalten zu können, z.B. Rundformen (Tonnendächer bzw. Rundtürme). Es dient also zur Ergänzung des Modells, wenn für die geplante Modellnutzung (Kap. 1) ein höherer Detaillierungsgrad benötigt wird.. Allerdings ist dieses Verfahren wegen der hochgradig manuellen Bearbeitung, sehr arbeitsaufwändig und sollte sparsam eingesetzt werden.

Für die photogrammetrische Modellierung ist es notwendig, Voraussetzungen für eine Bündelblockausgleichung zu schaffen. Zunächst benötigt man eine geeignete Anordnung der Aufnahmestandorte. Man kann zwischen Erfordernissen der Geometrie und der Oberflächentextur unterscheiden.

Zur Ermittlung der Geometrie sind Aufnahmestandorte zu wählen, in denen schleifende Schnitte [KRAUS 96] vermieden werden. In Abb. 19 ist eine solche Anordnung zur Modellierung eines Gebäudevorsprunges dargestellt. Es ist darauf zu achten, dass für eine ausreichende Überdeckungsfläche der Fotografien gesorgt wird. Relevant ist dabei nicht die Fläche an sich, son-

dem eine ausreichende Anzahl identischer Punkte in mindestens zwei nebeneinander liegenden Bildern. Voraussetzung ist die Identifizierbarkeit identischer Punkte, was besonders bei langen Gebäuden oder dichter Bebauung problematisch sein kann.

Optimal ist es, wenn in einer Bildfolge noch im übernächsten Bild identische Punkte zu finden sind. Abschattungen durch Vorsprünge oder Erker können dabei hinderlich sein. Es ist auf eine vollständige Ausleuchtung des Objektes zu achten. Sind trotzdem einzelne Punkte nicht sichtbar, lassen sich durch Formvorgaben fiktive Beobachtungen einfügen.

Im Bereich der Luftbildphotogrammetrie lassen sich durch automatische Verfahren [CLAUS 83, BOOCHS 84] homologe Punkte identifizieren und auswerten. Bei terrestrischen Objekten, mit stark konvergenten Aufnahmerichtungen und damit sehr differenten Bildinhalten, ist eine Automatisierung schwieriger. Deshalb wird die Bearbeitung interaktiv durchgeführt.

Zur Modellierung der Geometrie müssen in einem Bildverband jeweils räumlich identische, also homologe Punkte gefunden werden. Objekte mit sichtbaren Ecken und Kanten sind dafür besonders geeignet, da sich diese Stellen in Bildern besonders leicht identifizieren lassen. In Abb. 19 sind grüne Punkte in beiden Bildern zu sehen, die roten Punkte jeweils in nur einer Aufnahme. Für die spätere Modellrechnung müssen alle Kantenendpunkte bearbeitet werden.

Bei einigen Objektarten, wie z.B. historischen Gebäuden sind sehr viele kleingliedrige Objektteile und damit eine hohe Anzahl an Kanten zu finden. Hier kann ein Bearbeiter generalisierend einwirken, indem kleinere Strukturen nicht berücksichtigt werden. Dabei fallen diese Strukturen nicht weg. Vielmehr werden sie durch die Fototextur ersetzt.

Aus den bestimmten Punkten kann nun eine Bündelblockausgleichung gerechnet werden. Sind keine Passpunkte aus einem übergeordneten Koordinatensystem vorhanden, so entsteht ein geometrisch richtiges Modell mit unbekanntem Maßstab.

Für die Erfassung der Textur hingegen ist es wichtig, möglichst orthogonal zum Aufnahmeobjekt zu fotografieren. Für den in Abb. 19 gezeigten Gebäudevorsprung, ist in Abb. 20 die Aufnahmeanordnung für die Erfassung der Textur abgebildet. Bei dieser Anordnung ist der Bildausschnitt, in dem sich das Objekt befindet, nur mit wenigen Verzeichnungen behaftet.

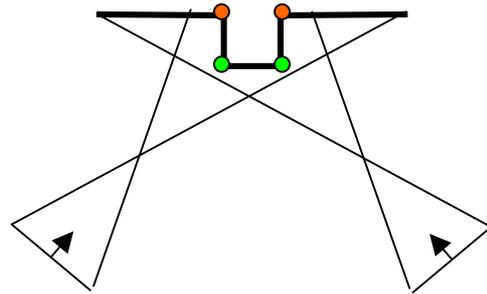


Abb. 19: Aufnahmestandorte zur Ermittlung der Geometrie

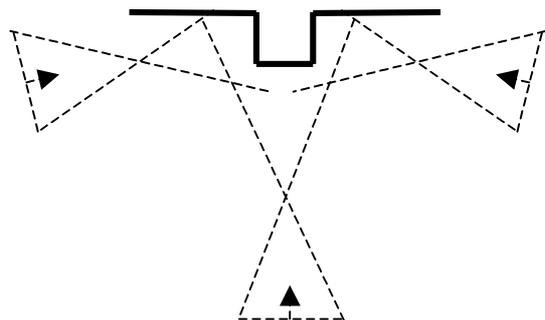


Abb. 20: Aufnahmestandorte zur Erfassung von Texturen

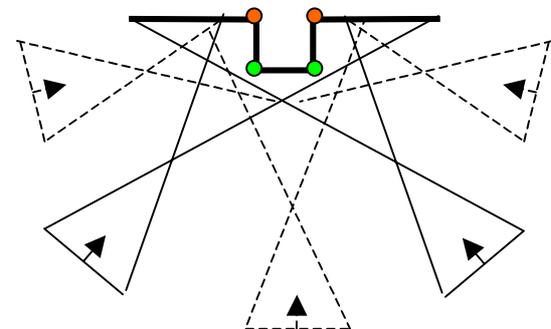


Abb. 21: Kombination von Aufnahmestandorten für Textur- und Geometriezwecke

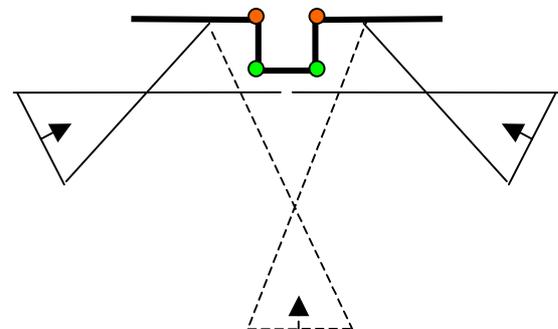


Abb. 22: Optimale Aufnahmeanordnung für Textur und Geometrie

Die Bildinformation ist nur wenig perspektivisch verzerrt, so dass die Informationsdichte über den gesamten Bildausschnitt, den das Aufnahmeobjekt wiedergibt, sehr homogen ist. Dadurch entsteht, nach der Entzerrung, ein realitätsnahes Bild.

Um eine möglichst realitätsnahe Textur und eine richtige und vollständige Geometrie zu erhalten, müssen beide Aufnahmeanordnungen miteinander verknüpft werden. Allerdings ist der Aufwand fast doppelt so hoch, siehe Abb. 21, als wenn nur geometrische Daten modelliert werden sollen. Allerdings ermöglicht die größere Bilderzahl eine höhere Anzahl an Messungen, die das Gesamtmodell verbessern. Zudem ist es oft schwierig, alle Aufnahmestandorte zu realisieren, da oft andere Objekte den Blick versperren.

Zur Reduzierung des Aufwandes auf ein wirtschaftlich vertretbares Maß ist es sinnvoll, eine Kompromisslösung anzuwenden, wie in Abb. 22 dargestellt ist. Dazu werden die beiden äußeren Aufnahmestandpunkte gemittelt und der mittlere Standpunkt bleibt. Mit diesem Vorgehen wird man der Geometrie gerecht und kann zudem noch geeignete Texturen mit geringen Verzerrungen erreichen.



Abb. 23: Gebäude mit rundem Grundriss

Die Verknüpfung zwischen Texturen und Geometrie erfolgt üblicherweise direkt in den Fotografien. Aus der Verbindung zwischen Punkten werden Kanten nachgebildet. Aus den Kanten entstehen durch topologische Verknüpfung Flächen. In unterschiedlichen Aufnahmen kann die gleiche Fläche vollständig vorkommen. In diesem Fall kann zwischen unterschiedlichen Quelltexturen gewählt werden.

Die bereits genannten Softwareprodukte für die photogrammetrische Modellierung [GEOMETRA] und [PHOTOMODELLER] bieten

neben einer komfortablen Benutzeroberfläche und der Bündelblockausgleichung weitere Bearbeitungsmöglichkeiten. So ist es möglich, Formvorgaben, wie Rechtwinkelbedingungen zu machen. Auf diesem Wege lassen sich Schwierigkeiten beheben, die z.B. von schlechter Bildgeometrie verursacht werden. So können auch Objekte modelliert werden, deren Geometrie nur schwer bearbeitet werden kann, wie dies bei Röhren und Rundungen der Fall ist. Solche Objekte werden als Röhrenobjekte gekennzeichnet für die im Bild einfach der Durchmesser ausgemessen wird.

Trotz dieser Möglichkeiten können dennoch einige Details Schwierigkeiten bereiten. Insbesondere historische Gebäude haben gelegentlich abgerundete Grundrisse. Abb. 23 zeigt einen Nebenturm der Orangerie des Schlosses Hellbrunn südlich von Salzburg. Der Grundriss der Außenfassade ist gekrümmt, jedoch gibt es keine messbaren Strukturen, welche eine hybride geometrisch-photogrammetrische Modellierung zulassen. Ein Hilfsmittel sind zusätzlich angebrachte Messmarken. In Abb. 24 rechts sind drei dieser Marken zu erkennen. Sie helfen die Kontur zu erfassen. Dabei reichen sehr kleine Marken aus. Die im Beispiel vorhandenen Marken sind etwa 1 cm groß. Sie sind im übrigen auch in Abb. 23 (Abb. 24 ist ein Auszug aus Abb. 23) vorhanden, aufgrund ihrer Größe aber nicht erkennbar.

Problematisch ist auch der obere Bereich des Gebäudes. Die Traufe befindet sich in über 6m Höhe. Hilfsmittel wie Leitern oder stehen selten zur Verfügung und ihre Handhabung z.T. gefährlich. Daher ist hier ein Weg hilfreich, bei dem ebenerdig gearbeitet werden kann.



Abb. 24: Zusatzmarken

Entlang einer Lotlinie werden zwei Punkte markiert (zur Überbestimmung hier drei Punkte, siehe Abb. 24). Bei der Auswertung wird eine später wieder zu entfernende Linie entlang dieser Punkte gezogen und bis zur Traufe verlängert. Der resultierende Schnittpunkt kann als weiterer Stützpunkt verwendet werden, so dass die gesamte Mauer in Streifen geteilt wird und so der gerundeten Form angeglichen wird. In Abb. 25 ist der fertig modellierte Turm im 3D-Modell zu sehen, der besseren Anschaulichkeit halber von schräg unten. Aus dieser Perspektive sind die einzelnen Streifen zu erkennen.



Abb. 25: Fertig modellierter Turm in Hellbrunn

Zur Texturierung der Objekte werden die einzelnen Teilflächen betrachtet. Jede Teilfläche erhält genau eine Bilddatei als Quelle für die Textur. Die Bilddatei entsteht, indem aus dem Ausgangsphoto ein Bereich entnommen wird. Damit der Bildausschnitt sich gut den orthogonal angeordneten Pixel des Bilddatenformates anpasst, wird es mittels einer Rotation in das Format hineingedreht. So wird die Minimierung der Dateigröße erreicht, da die Bildfläche bestens ausgenutzt wird.

Die Bildgröße der Texturen kann den Erfordernissen der Nutzung angepasst werden. Dabei darf die Qualitätseinbuße jedoch nur bis zu einem bestimmten Maß führen. Der zeitaufwändige Arbeitsschritt vom Erkennen des Informationsgehalts eines Fotos zur entsprechenden Reduktion kann automatisiert werden.

Dazu wird von der realen geometrischen Größe des Ausschnittes innerhalb des Modells ausgegangen. Diesem entsprechend muss die Anpassung vorgenommen werden. Dazu wird für jedes Texturelement das Flächenverhältnis zwischen Modells und Realität bestimmt. Ist das Flächenverhältnis größer als eine entsprechende Kennzahl, kann man davon ausgehen, dass die Textur eine geringe Informationsdichte hat. Eine Verkleinerung dieser Texturfläche ist somit zulässig.

Problematisch bei dieser Beurteilung ist die Annahme, dass eine große Fläche mehr Information enthält als eine kleine. Dies ist nicht zwingend der Fall. Beispielsweise ist die Textur einer glatt verputzten Hauswand gegenüber einem Fenster flächenmäßig groß, aber von weniger Inhalt. Es kann also passieren, dass eine kleine Fläche mit viel Information stark verkleinert wird und sich somit die Textur verschlechtert. Daher ist bei diesem Verfahren eine manuelle Überprüfung notwendig. Es wird aber trotzdem eine signifikante Arbeitszeiteinsparung gegenüber einem rein interaktivem Verfahren erzielt.



Abb. 26: Texturbasis für ein Dach

Eine weitere Möglichkeit der Automatisierung, ist die Kontrolle der Informationsdichte mit dem Histogramm. Weist das Histogramm eines Bildes ein oder zwei deutliche Maxima auf, bedeutet dies, dass eine erhebliche Anzahl an Pixeln gleiche oder ähnliche Farb- oder Grauwerte besitzt. Das Bild wird von diesen Werten dominiert. Wenn dies der Fall ist, kann davon ausgegangen werden, dass in etwa gleichermaßen eingefärbte Bildteile vorliegen, das Bild folglich eine geringe Informationsdichte besitzt. Dieser Eindruck wird um so stärker, je steiler die Flanken der Maxima sind.

Ist das Histogramm relativ flach, bedeutet dies, dass die Anzahl der Pixel gleichmäßig über die Farbwerte verteilt ist. In diesem Fall wird das Bild nicht von bestimmten Werten dominiert, sondern besitzt viele unterschiedliche Farb- und Grauwerte. Das Bild besitzt also eine unregelmäßige Struktur und somit liegt ein Bild mit hoher Informationsdichte vor. Entsprechend diesen

Kriterien erfolgt anschließend die Reduktion der Bildgröße. Dieses Verfahren ist unabhängig vom Bildformat anwendbar.

Ein weiteres Maß für den mittleren Informationsgehalt eines Bildes ist die Entropie H . Über den Wert H bestimmt man die mittlere notwendige Bittiefe für ein Bild, mit der dieses ohne Informationsverlust gespeichert werden kann. H gibt die Anzahl der Bit für die Speicherung an.

$$\text{Entropie } H = - \sum_{g_i=g_{\min}}^{g_{\max}} p(g_i) \cdot \log_2 p(g_i)$$

$p(g_i)$ = Wahrscheinlichkeit für das Auftreten des Farbwertes g eines Farbkanals

Des Weiteren gilt: $\sum_{g_i=0}^{255} p(g_i) = 1$ und $0 \leq p(g_i) \leq 1$.

Aus dem Entropiewert eines Bildes, kann die notwendige Anzahl an Bit pro Pixel zur Speicherung des Bildes ermittelt und dadurch Speicherplatz gespart werden. Je geringer der Informationsgehalt eines Bildes ist desto geringere Bittiefe ist für die verlustlose Speicherung nötig. Der Entropiewert wird getrennt für jeden Farbkanal berechnet. Mit ihm kann das optimale Speicherformat bestimmt werden. Man kann verallgemeinern:

$$\text{Gif}, \forall \sum (H_R, H_G, H_B) \leq 8$$

da das gesamte Bild durch Auswahl von 256 Farben darstellbar ist, und

$$\text{Jpeg}, \forall (\max(|H_R|, |H_G|, |H_B|)) * 3 \geq 16$$

da das Jpeg-Format den 16 Bit Farbraum abdeckt.

Dabei entsteht ein Bereich, in dem keine klare Zuordnung möglich ist, sondern zwischen der möglichst geringen Datenmenge und der notwendigen Farbtiefe entschieden werden muss. Es kommt dann darauf an, in welchem Farbbereich die Farbmaxima zu finden sind. Die Zugehörigkeitsfunktion in Abb. 27 veranschaulicht das Auswahlkriterium für das Speicherformat.

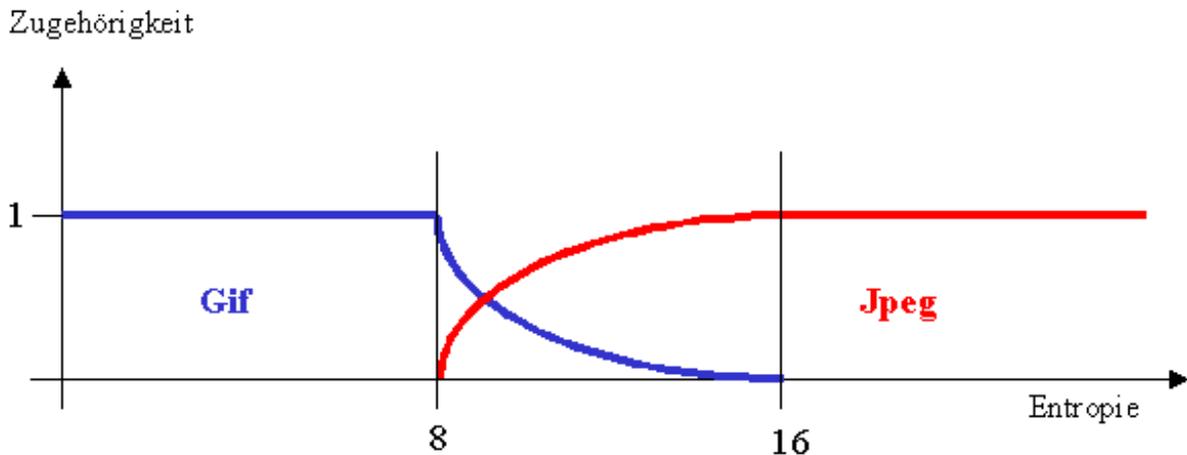


Abb. 27: Verwendung von Jpeg und Gif Bildformat nach Entropie

Leider stehen in der Regel nur das Gif bzw. das Jpeg-Format zur Verfügung. Eine optimale Speicherung aufgrund der notwendigen Bittiefe ist daher nicht immer möglich, wäre aber wünschenswert. Hier besteht noch ein erhebliches Reduktionspotential.

3.4 Regeln und Vorschriften

Neben den genannten Datenquellen gibt es noch andere, die schwieriger erschließen und in eine informationstechnische Form bringen lassen, zumal sie üblicherweise nicht digital vorliegen.

Ein Teil dieser Datenquellen ist auf bestimmte geografische Räume bezogen. Aus ihnen folgen beispielsweise konstante Attribute, die in das Modell eingeführt werden können. Für Gebäude werden Regeln durch Bauvorschriften, Bebauungspläne (Baugesetzbuch § 9) bzw. Gemeinde-satzung oder gestalterische Regeln, aber auch durch Erfahrungswerte bestimmt, sozusagen wissensbasierte Parameter.

Dabei können z.B. folgende Annahmen getroffen, bzw. vorausgesetzt werden (Auswertung des Beispiels einer realen planrechtlichen Festlegung nach Baugesetzbuch § 9 im Anhang):

- Die Dachform ist Satteldach mit Dachneigung 40%
- Die Firstlinie läuft parallel zur längsten Gebäudeseite
- Die Geschosshöhe beträgt 2,80 m
- Die Sockelhöhe beträgt 0,5 m
- Die Kniehöhe beträgt 0,8 m
- Der Dachüberstand giebelseitig ist 0,25 m und traufeseitig 0.5 m
- Die Dachfarbe ist rot, und die Form der Dachziegel strukturiert

Diese Größen lassen sich dann durch die Formulierung von Default-Objekten in die Modellierung einbringen. Änderungen dieser Regeln oder weitere Regeln können sich durch die jeweilige räumliche Lage ergeben.

Andere Datenquellen können über Funktionen, die auch externe Parameter beinhalten, definiert werden. Auch für Bäume gibt es wissensbasierte Regeln, die in Funktionen ausgedrückt werden können (Auswahl) :

- Gravitation f (Geländemodell, Baumposition, Baumunterkante): ein Baumobjekt muss aus der Geländeoberfläche heraus wachsen
- Stammhöhe f (Baumart, Baumalter, Baumhöhe, Kronendurchmesser): die Höhe des Stammes kann aus einem Wachstumsmodell bestimmt werden
- Kronenhöhe f (Baumhöhe, Stammhöhe) : die Höhe der Krone kann aus der Differenz von Baumhöhe und Stammhöhe berechnet werden

3.5 Detailstufen

Daten zur Modellierung von 3D-Objekten sind zweifelsohne eine Datenfamilie in der hohe Datenmengen auftreten. Deshalb sind bei ihrer Verwendung Strategien gefragt, wie eine Optimierung der Handhabung dieser Datenmengen erfolgen kann. Für die Darstellung von Gebäuden und die gleichzeitige Verwendung einer angemessenen Datenmenge ist daher ein Detailstufenmodell mit einer darauf abgestimmten Zugriffsstruktur sinnvoll.

Ein Detailstufenmodell beinhaltet vollständige Repräsentationen ein und desselben Objekts in unterschiedlichen Detaillierungsgraden. Abhängig von der Detaillierung bedeutet es mehr oder weniger Aufwand der für die Darstellung betrieben werden muss. Die Detailstufen können zu

unterschiedlichen Bedingungen aufgerufen werden. Zwei Repräsentationen in einer Anwendung dürfen nicht gleichzeitig aufgerufen werden. Das Detailstufenmodell ist objektbezogen und wird im Zuge der Objektmodellierung erstellt. Automatisch erzeugte Modelle sind möglich.

Für das Detailstufenmodell wird die folgende Zugriffsstruktur vorgeschlagen:

Der einfachste Fall unterschiedlicher Detaillierung ist die Darstellung in Abhängigkeit von der Entfernung. Weitere Bedingungen, zu denen wechselnde Detailstrukturen abgerufen werden sollen sind:

- Richtungsbedingungen: Denn der Fokus der Betrachtung liegt auf der Bildmitte. Deshalb dürfen Objekte außerhalb eines Hauptblicksektors einen geringeren Detaillierungsgrad aufweisen.
- Sichtbarkeitsbedingungen: Da in einer 3D-Szene ein Bild graphikkartenseitig immer von hinten nach vorne aufgebaut wird, werden Objekte dargestellt, obwohl sie hinterher nicht mehr sichtbar sind.
- Thematische Bedingungen: Um besondere Inhalte besonders hervorzuheben, beispielsweise alle Eichen in einer Mischbewaldung, werden die thematisch hervorzuhebenden Objekte detailliert gezeigt.
- Zeitbedingungen: Mit denen kann automatisch zwischen zeitlich variablen Modellierungen gewechselt werden, beispielsweise zwischen der Sommer- und Winterdarstellung eines Laubbaumes (siehe Kap. 4)

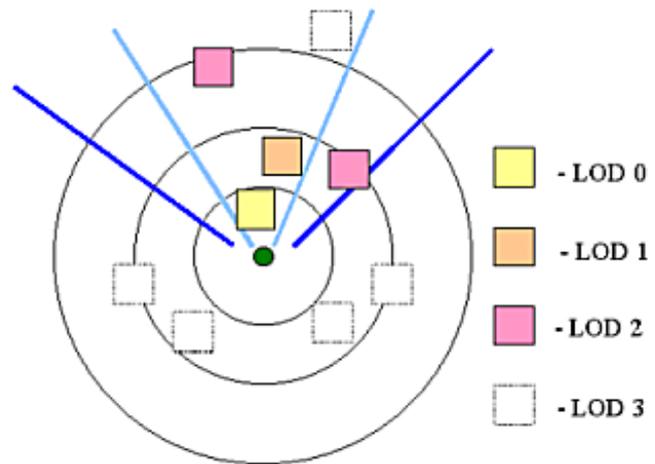


Abb. 28: LoD nach Entfernung-, Richtungs- und Sichtbarkeitsbedingungen

Kapitel 4

MODELLIERUNG VON VEGETATIONSOBJEKTEN

Vegetation (lateinisch *vegetare*: wachsen, beleben) ist ein botanischer Begriff für die aus bestimmten Pflanzengesellschaften aufgebaute Gesamtheit des Pflanzenbewuchses in einem Gebiet [Encarta]. Unter Vegetationsobjekten werden im folgenden die räumlichen Objekte Baum und als Sonderform des Baumes der Busch verstanden.

Die Modellierungsarten, die hier vorgestellt werden, sind keine neuen Entwicklungen. Sie werden bereits erfolgreich besonders in Spiel- bzw. Trickfilmumgebungen, also nicht in 3D-Stadtmodellen eingesetzt. Ergänzt werden diese Darstellungen durch eigene Überlegungen zur Texturgewinnung und zur Verwendung verschiedener Texturen zu unterschiedlichen Zwecken und Jahreszeiten.

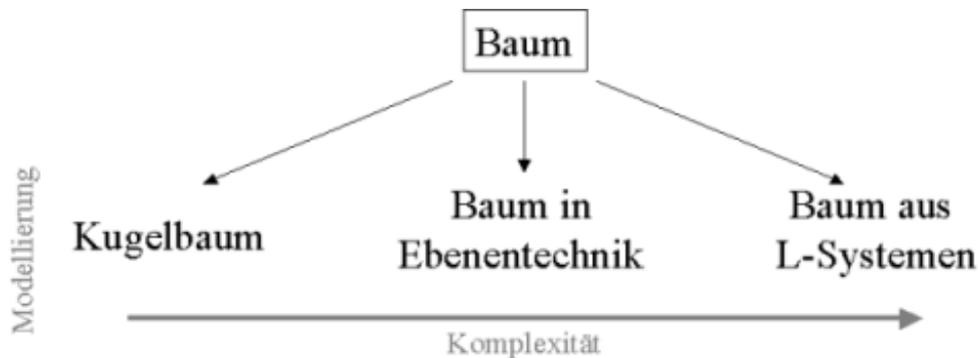


Abb. 29: Überblick über eine Auswahl von Modellierungsarten der Objektart Baum, Anordnung nach Komplexität des Aussehens

Der Baum ist im Sinne der 3D-Modellierung ein auf einen Punkt bezogenes, natürliches mit unregelmäßigen Umrissen versehenes Objekt. Um dieser Charakteristik nahe zu kommen gibt es unterschiedliche Wege zur Modellbildung:

- mit dem Kugelbaummodell wird die voluminöse Ausprägung eines Baumes nachgebildet. Dabei wird die Baumkrone unabhängig vom Stamm als deformierte Kugel (für Laubbäume) und als Konus (für Nadelbäume) betrachtet.
- Bäume in Ebenentexturtechnik werden gebildet indem das Bild eines Baumes auf eine oder mehrere ebene Flächen aufgebracht wird.
- bei Bäumen aus L-Systemen wird das natürliche Wachstum von Bäumen nachgebildet. Diese Bäume bestehen aus einer Ansammlung grafischer Repräsentationen für Stamm, Ästen und Blättern.

In Abb. 29 wird ein Überblick über diese Modelle gegeben, die im folgenden näher betrachtet werden. Unter dem Gesichtspunkt der Komplexität ihrer Geometrie (Kugelbaum, geringste Komplexität – L-Systeme, höchste Komplexität) können auch weitere Modelle, die hier nicht näher betrachtet sind, geordnet werden. Es gibt noch andere Ansätze, als die hier verwendeten, beispielsweise die Betrachtung der Baumkrone als Quader.

4.1 Kugelbaumobjekte

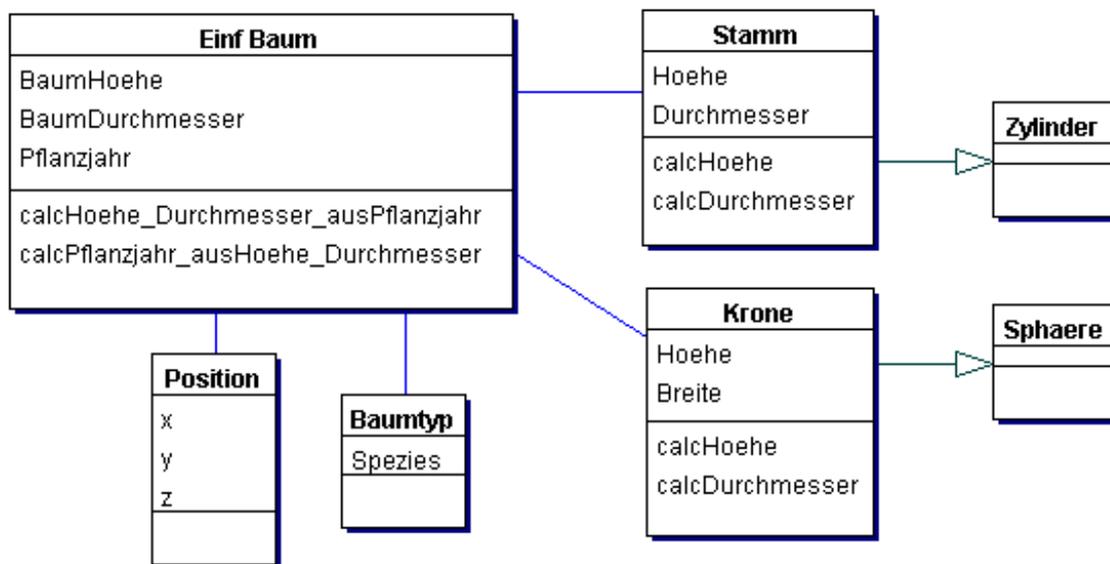
Unter dem Begriff Kugelbaumobjekte werden die aus einfachen geometrischen Volumenkörpern (z.B. Zylinder, Konus, Kugel) gebildeten Vegetationsobjekte subsummiert. Bäume werden in die Bestandteile Stamm und Krone (beim Busch nur Krone) unterteilt.

Laubbäume bestehen aus Zylinder für den Stamm und Kugel für die Krone, Nadelbäume werden durch Zylinder für den Stamm und Konus für die Krone gebildet. Die hier beschriebene Modellierung bezieht sich auf den Typ Laubbaum. Die Erkenntnisse lassen sich jedoch leicht auf Nadelbäume übertragen.

4.1.1 Einfache Kugelbaumobjekte

Mit den in Kap. 3 aufgeführten Daten kann ein einfaches Modell, bestehend aus den Teilen Stamm und Krone erstellt werden. Stamm und Krone werden getrennt betrachtet, da sie unterschiedliche Geometrie und Materialeigenschaften besitzen.

In Modell 1 ist die Analysesicht für ein Kugelbaumobjekt Laubbaum als Klassen-Diagramm in



Modell 1: Strukturmodell einfacher Kugelbaum

UML abgebildet. Der Baumstamm, ist ein Zylinder, die Krone ist eine Sphäre. Zudem wird über die Position die Lage des Objektes bestimmt, der Baumtyp dient der Baumhöhenberechnung aus dem Pflanzjahr.

Die Attributierung ist vom Zweck der Modellierung abhängig. Für die Visualisierung ist der optische Eindruck entscheidend. Dabei kann ein Baum auf unterschiedliche Art und Weise modelliert werden. Je nach Aufwand entsteht ein mehr oder weniger realer Objekteindruck. Die Baumkrone eines Laubbaumes lässt sich vereinfacht als gestreckte oder gestauchte Kugel (Sphäre) darstellen. Dabei geben Kronendurchmesser und Kronenhöhe die Geometrie für die Kugel an. Für jede Baumart ist eine andere Einfärbung vorzusehen. Je nach Jahreszeit kann von einer anderen Transparenz des Objektes ausgegangen werden. Der Zylinder des Stammes erhält baumartspezifische, aber opake Einfärbung.

Dieses sehr einfache Modell benötigt nur eine niedrige graphische Ebene zur Realisierung. Das Beispiel in Abb. 31 wurde mit Hilfe des Gouraud-Shadings und VRML erstellt.

Das einfache Kugelmodell wird erweitert, indem anstelle der Einfärbung eine ebenfalls objektbezogene Textur verwendet wird.

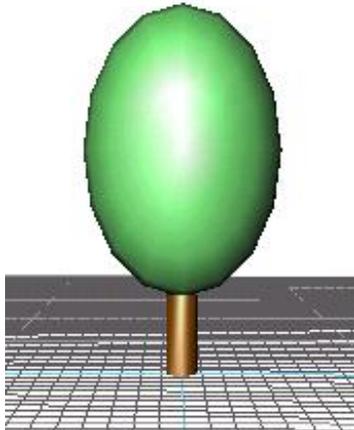


Abb. 30: Visualisierung Modell farbiger Kugelbaum

Abb. 31 zeigt eine solche Textur, die sowohl auf Stamm als auch auf Krone angewendet wurde. Die Kronentextur wird durch die Verwendung von Transparenz etwas realistischer gestaltet.

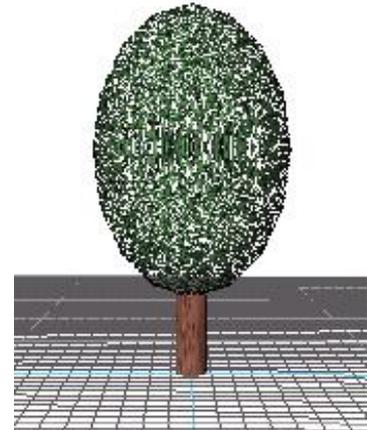
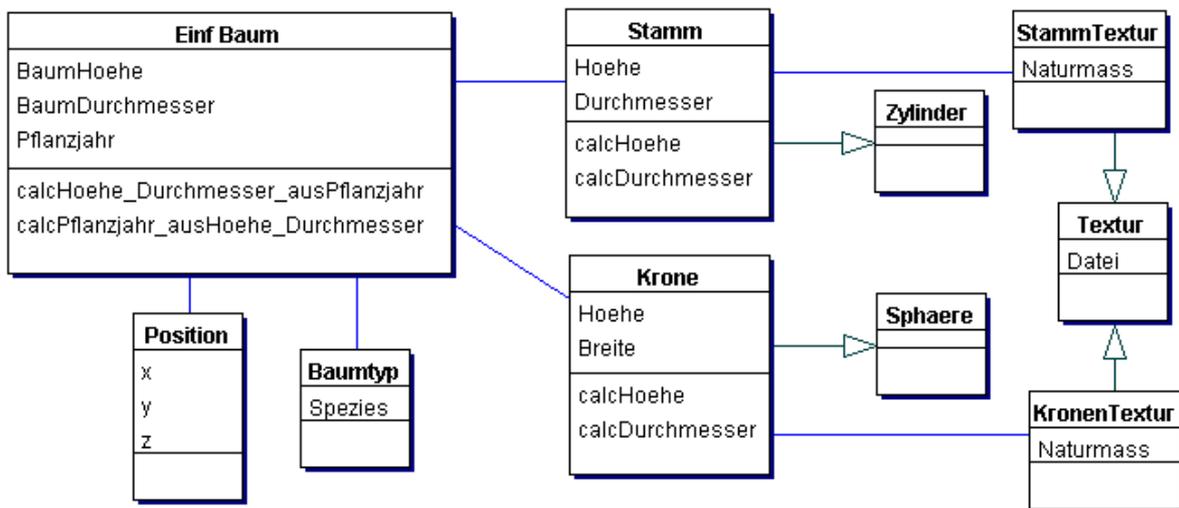


Abb. 31: Kugelbaummodell mit Textur

Für die Texturierung werden Ausschnitte von Bildern als Texturen verwendet. Die Textur muss der Geometrie des Objektes angepasst werden. Die Anpassung geschieht durch mehrfache Anordnung der Texturkachel neben- und übereinander (in der Kachelungstechnik s, t für die Anzahl der Wiederholungen).

Um die richtige Anzahl der Kachelwiederholungen automatisch bestimmen zu können, muss bekannt sein, welcher Ausdehnung die Textur in der Natur entspricht. Der Vergleich dieses Wertes mit der Objektgröße, ergibt die Werte (s, t) für die Wiederholung.

Das um die Textur (im Gegensatz zu Modell 1) erweiterte Modell 2 beinhaltet eine Textur für



Modell 2: UML-Modell des Kugelbaummodells mit Textur

den Stamm und eine solche für die Baumkrone. Diese Texturen sind Bestandteil jeweils des Stammes bzw. der Krone und darüber zum Baum gehörig. Das Modell ist ansonsten unverändert.

4.1.2 Komplexe Kugelbaum-Objekte

Dieses Modell lässt sich erweitern, indem die Kardinalität zwischen Baum und Krone mehr als eine Krone zulässt. Dadurch entstehen Nebenkronen, welche wie Verästelungen wirken (siehe

Abb. 32). Zur Modellierung müssen eine Richtung und ein Abstand von der Hauptkrone angegeben werden, da die Nebenkronen exzentrisch zur Hauptkrone stehen.

Krone
Hoehe
Durchmesser
calcHoehe
calcDurchmesser

Modell 3: Modelldifferenz im Nebenkronenmodell

Zur Realisierung dieser Modellierung müssen weitere Daten vorliegen oder in den Datenbestand eingefügt werden können. Wenn die Realitätsnähe des Eindrucks gesteigert werden soll, kann es genügen, Nebenkronen mit Zufallswerten hinzuzufügen.

Die Aufwände zur Erstellung von Nebenkronen sind im Vergleich zum Ergebnis recht hoch, so dass diese Form der Modellierung eines Baumes recht selten angewendet wird. In Modell 3

sind die Veränderungen gegenüber Modell 2 zu sehen, welche das Modell zur Erstellung von Nebenkronen befähigt.

Komplexe Kugelbaum Modelle sind als Allgemeinfall zu betrachten, da die Nebenkronen nicht zwingend besetzt werden müssen und so auch das einfachere Modell diesen Modellierungstyp abdeckt.

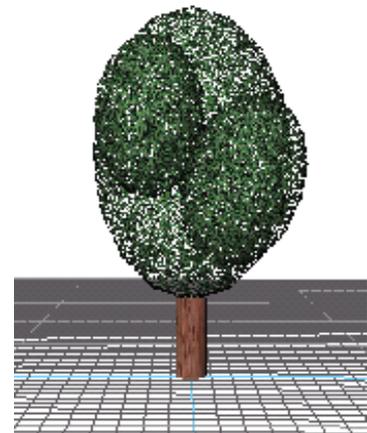


Abb. 32: Visualisierung des Nebenkronenmodells

4.1.3 Texturen für Kugelbaum-Objekte



Abb. 33: Blatt-Textur für Kugelbäume - Originalaufnahme

Texturen für Kugelbaum Objekte werden gewonnen, indem eine Nahaufnahme vom Laub des Baumes gemacht wird (Abb. 33). Dieses Laub wird weiterverarbeitet indem die Farbenzahl reduziert wird, so dass eine höhere Anzahl von Pixeln je Farbe entsteht.

Anschließend wird eine häufig vorkommende aber nicht so leicht dem Objekttyp zuzuordnende Farbe gewählt und diese Farbe als Transparenzkanal belegt, so dass der Eindruck einer lichtdurchlässigen Krone entsteht.

Abb. 34 zeigt die bearbeitete Fototextur, wobei die transparenten Stellen weiß dargestellt sind.

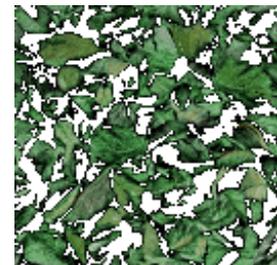


Abb. 34: Blatt-Textur für Kugelbäume – bearbeitete Textur

4.2 Bäume aus L-Systemen

Hinter L-Systemen verbirgt sich der Ansatz, den natürlichen Bildungs- und Wachstumsprozess eines Objektes nachzubilden. Damit soll eine möglichst naturnahe und damit realistische Abbildung erzeugt werden. Ehedem nur für natürliche Objekte entwickelt, werden L-Systeme heute auch für anthropogen entstandene Objekttaggregationen verwendet.

4.2.1 Das Prinzip

Beginnend mit einem Startelement werden mit Hilfe von Produktionsregeln Elementketten generiert. Diese sind dann graphisch darzustellen. Ein solches Element kann ein Blatt, eine Knospe, ein Stück Stängel, etc. sein. Jedes Element kann Attribute enthalten, die Größe, Alter, Farbe, Hormonkonzentration o.ä. beschreiben können. Aufgrund seiner Einfachheit ist das L-System sehr gut die Vegetationsgenerierung für 3D-Stadtmodelle geeignet. Es gibt mehrere

Varianten der L-Systeme, die sich in den in der folgenden Auflistung angegebenen Eigenschaften unterscheiden:

- Deterministische L-Systeme wenden feste Algorithmen und Parameter für das Wachstum an. Sie werden i.d.R. durch ein D gekennzeichnet
- Stochastische L-Systeme verwenden einen Zufallsgenerator, der dafür sorgt, dass die erzeugten Objekte unterschiedlich ausfallen
- Kontextfreie L-Systeme entwickeln die einzelnen Objektelemente unabhängig voneinander. Sie werden durch den Kontextparameter 0 gekennzeichnet.
- Kontextsensitive L-Systeme machen die Entwicklung einzelner Elemente vom Status der Nachbarelemente abhängig. Der Grad der gegenseitigen Abhängigkeit der Elemententwicklung wird durch die Kontextparameter 1, 2, ... ausgedrückt.

Die einfachste Form von L-Systemen sind die 0L-Systeme (sprich Null-L-Systeme), bei denen die Elemente kontextfrei entwickelt werden. Die Definition eines 0L-Systems (nicht deterministisch) und eines D0L-Systems (deterministisch) lautet:

Es sei V ein Alphabet $\{a_1, a_2, a_3 \dots a_n\}$, V^* die Menge aller möglichen Wörter aus V und V^+ die Menge aller möglichen Wörter aus V ohne $\{\emptyset\}$. Ein 0L System ist das geordnete Triplet $[V, \omega, P]$, in dem V das Alphabet, $\omega \in V^+$ ein nichtleeres Wort genannt Axiom und $P \subset V \times V^+$ eine endliche Menge an Produktionsregeln ist. Eine Produktionsregel $(a, \chi) \in P$ wird als $a \rightarrow \chi$ geschrieben.

Der Buchstabe a und das Wort χ werden Vorgänger und Nachfolger genannt. Für jeden Buchstaben a existiert mindestens ein Wort $\chi \in V^*$ so dass $a \rightarrow \chi$. Ist kein spezielles Wort χ für einen bestimmten Vorgänger $a \in V$ gegeben, so wird $a \rightarrow a$ der Menge aller P hinzugefügt. Ein deterministisches 0L-System, genannt D0L-System liegt dann vor, wenn für jedes $a \in V$ genau ein $\chi \in V^*$ existiert, so dass $a \rightarrow \chi$.

Man verwendet zur Darstellung der L-Systeme einen anweisungsbezogenen Zeichenpunkt (turtle = Schildkröte, im deutschen auch Igel genannt), der den folgenden Bewegungsanweisungen gehorcht. Dieser Zeichenpunkt kann auch als Stift eines Plotters betrachtet werden. Der Winkel δ hat über dem gesamten Zeichenvorgang eine feste Größe.

- F Zeichne Element F und bewege den Stift um die Länge des Elements F nach vorne
- G Zeichne Element G und bewege den Stift um die Länge des Elements G nach vorne
- + Drehe um δ gegen den Uhrzeigersinn
- Drehe um δ im Uhrzeigersinn
- [Merke aktuelle Position
-] Kehre zur letztgemerkten Position zurück

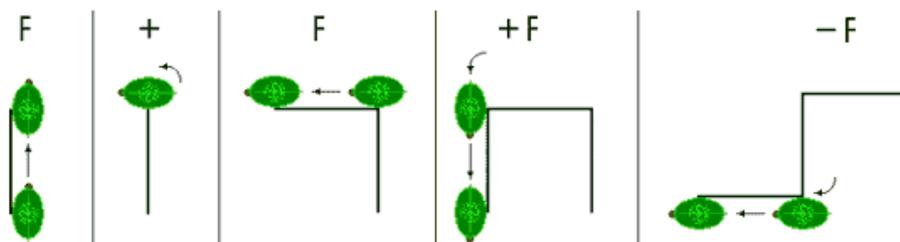


Abb. 35: Das Wort $F+F+F-F$ ($\delta=90^\circ$) interpretiert durch den Igel

Überführt in das Alphabet des L-Systems entspricht dies dem Befehlssatz: $V = \{F, G, \dots, +, -, [,]\}$

Das Wort $F+F+F-F$ mit $\delta = 90^\circ$ interpretiert der Igel wie in Abb. 35 dargestellt. Hier wird die Befehlsfolge in Einzelbilder zerlegt, Das Ergebnis der Igelbewegung ist als Strich dargestellt.

Ein weiteres Beispiel ist das Wort $F[+F][-F-F]F$, mit $\delta = 90^\circ$ bildet der Igel die Interpretation, wie in Abb. 36 gezeigt:

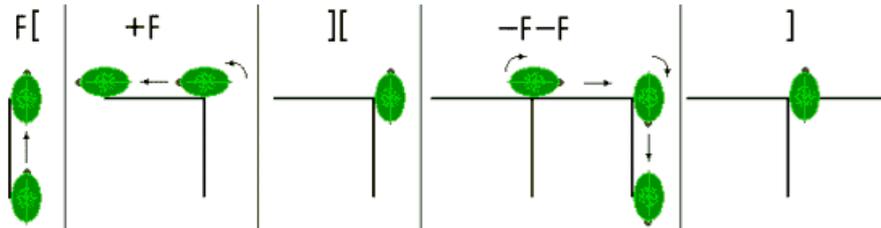


Abb. 36: Das Wort $F+F+F-F$ ($\delta = 90^\circ$) interpretiert durch den Igel

In [L-SYS] werden beispielhaft einige L-Systeme gebildet, die hier zur Erläuterung dargestellt werden. Mit Hilfe des Igels können Verzweigungen jeder Art realisiert werden. Je mehr Iterationsschritte ausgeführt werden, desto verästelter wird das Ergebnis sein.

Das buschige L-System in Abb. 37 enthält nur das Element F. F wird vom Igel als grüne Linie interpretiert. Bereits nach der vierten Iteration mussten 1116 Zeichenbefehle ausgeführt werden. Der Parameter δ beträgt 30° .

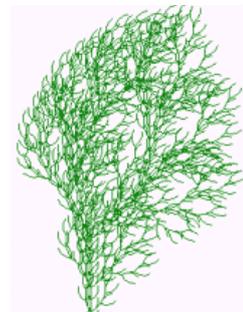


Abb. 37: Buschartig ausgebildete Struktur aus L-Systemen, 4 Iterationen

ω :	F
p_1 :	$F \rightarrow FF[-F+F+F][+F-F-F]$
δ :	30°

Die krautartige Struktur aus Abb. 38 entsteht aus sechs Iterationen und 4607 Zeichen. Allerdings wurden nur alle F gezeichnet; das Element I ist ausschließlich dazu da, um die Verzweigungen zu generieren und besitzt keine physische Länge. Der Igel ignoriert es einfach.

Bemerkenswert sind die verschiedenen Arten der Verzweigung. Ebenfalls gut sichtbar ist hier das Prinzip der Selbstähnlichkeit: Einzelne Äste haben stets dieselbe Form wie die gesamte Pflanze.

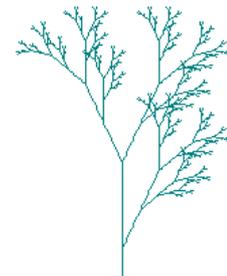


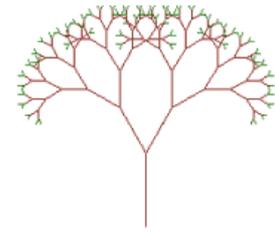
Abb. 38: Krautartig ausgebildete Struktur aus L-Systemen, 6 Iterationen

ω :	I
p_1 :	$F \rightarrow FF$
p_2 :	$I \rightarrow F[+I]F[-I]+I$
δ :	30°

Bei der baumartigen Struktur von Abb. 39 werden zwei Elemente F und G verwendet. F wird als grüne, G als braune Linie interpretiert. Die Länge der Linien hängt vom jeweiligen Parameter ab.

Der Parameter bewirkt, dass die Linien kürzer werden, je mehr sie verzweigt sind. Die Darstellung zeigt die Visualisierung nach dem 7. Iterationsschritt.

$\omega :$	F_1
$p_1:$	$F_n \rightarrow G_n[+F_n*0.7][+F_n*0.7]$
$p_2:$	$G_n \rightarrow G_n$
$\delta:$	30°



Mit diesen Darstellungen sind wesentliche Grundformen für Pflanzen beschrieben.

Abb. 39: Baumartig ausgebildete Struktur aus L-Systemen, 7 Iterationen

4.2.2 Texturierung von L-Bäumen

Neben der natürlichen Aststellung kann über Texturen ein weiterer Realitätsgewinn in der Darstellung von Bäumen erzielt werden. Dazu werden Stamm und Äste einfach mit der passenden Rindentextur überzogen. Abb. 40 zeigt dies für einen Kirschbaum.



Einen weiteres Objektelement für die Verwendung von Texturen sind die Blätter. Diese können als texturierte ebene Fläche dargestellt werden. Die Textur wird von fotografierte Blätter desjenigen Baumtyps übernommen, der dargestellt wird. Dazu wird das einzelne Blatt freigestellt und der restliche Anteil des Bildes transparent belegt. Eine Anwendung dieser Technik ist in Abb 40 dargestellt.

Abb. 40: Texturierter Stamm und Äste sowie Blätter eines Kirschbaumes als L-System, Blick von schräg oben durch die Krone

4.3 Bäume mit Ebenentechnik

Die bisher vorgestellten Modellierungen erzeugen die eher volumenartige Ausprägung eines Baumes. Diese beruht auf einer durch Dreieckspolygone gestalteten, genäherten Umrissform. Dieses Vorgehen hat für die Visualisierung besondere Bedeutung. Je nachdem, wie viele Polygone dargestellt werden, verändert sich die Anzahl der darstellbaren Bilder/Sekunde bis zu einem Grenzwert, bei dem die Animation ins Stocken gerät.

Bei einem der modernsten Geräte, die auf 3D-Grafikvisualisierung optimiert sind, der Playstation 2 (PS2), beträgt dieser Grenzwert laut Herstellerangaben etwa 200.000 texturierte Polygone. Geht man von einer objektbezogenen Polygonzahl von 250 (was eher die Untergrenze darstellt) aus, so lassen sich auf diese Art und Weise bis maximal 800 Objekte darstellen. Diese Anzahl ist aber zur Darstellung weitläufiger Räume und Landschaften nicht ausreichend. Es ist daher nach Wegen der Darstellung zu suchen, die mit weniger hohen Polygonzahlen auskommen.

4.3.1 Bäume als Tafelflächen

Ein Bild des Baumes wird als Textur auf eine rechteckige Fläche aufgebracht. Damit wird nur die Ansicht einer Baumseite abgebildet. Da ein Baum, vereinfacht gesehen, rotationssymmetrisch ist, reicht diese Ansicht als Visualisierung aus, vorausgesetzt, die Fläche befindet sich immer rechtwinklig zum Betrachter, was durch das Tafel- Konzept gewährleistet wird. Man

kann eine oder mehrere Achsen eines Objektes festlegen, die orthogonal zur Visur des jeweiligen Betrachtungspunkt gehalten wird. Im Falle des Baumes ist dies die Z-Achse.

Bäume in Tafel-Technik eignen sich für Objektansammlungen, wie Wald. Für Einzelbaumdarstellung ist die Anwendung weniger geeignet, da die Ansicht statisch ist, wodurch die Szene unnatürlich erscheint. Zudem sind Bäume in Tafel-Technik in der Aufsicht nur eine mehr oder weniger dünne Linie.

4.3.2 Modellierung von Bäumen als Kreuztafelobjekte

Ein Baum kann als Textur mit Transparenzmöglichkeit gebildet werden, indem die Kronentextur mittig auf den Stamm aufgesetzt wird. So erhält man einen recht realistischen Baum, jedenfalls für den Sichtbereich 45° links und rechts orthogonal von der Textur aus.

Um das Defizit des begrenzten Blickwinkels auszugleichen, wird eine zweite Kronentextur rechtwinklig zur ersten mittig auf den Baumstamm gesetzt. In Abb. 41 ist das Prinzip dieser Vorgehensweise, zur Verdeutlichung ohne Texturen, zu sehen. Zur besseren Visualisierung wurde gegenüber den bisherigen Bildern hier die Blickposition leicht nach oben verlegt und nach unten geneigt.

Demgegenüber wird in Abb. 42 die gleiche Situation mit Texturen dargestellt. Der Blickwinkel auf die Texturen beträgt 45° . Bereits diese mit geringem Aufwand hergestellte Darstellung wirkt recht realitätsnah. Bedingt durch die Transparenz in der Textur hat man, wenn man sich im Halbkreis um den Baum bewegt, an jeder Position ein anderes Bild des Baumes, der dennoch füllig wirkt.

Dieser füllige Eindruck kann noch verstärkt werden, indem anstelle von zwei Texturen drei oder gar vier verwendet werden, die ebenfalls in gleichen Winkeln von 60° , bzw. 45° zueinander positioniert werden. Der Winkel verändert sich auf 30° bzw. 22.5° .

Betrachtet man den Baum von oben, so sieht man den zylindrischen Baumstamm und darüber die Ebenen, die aus dem Blickwinkel von oben wie ein Kreuz erscheinen. Diesem Umstand kann begegnet werden, indem eine horizontale Kronentextur gefertigt wird. Die Kronentextur ist im einfachsten Fall die vertikale Textur um 90° gedreht oder eine Photographie einer Krone von unten. Diese wird dann in mittlerer Höhe der Krone eingefügt. So erhält man auch von oben einen fülligen Eindruck der Krone.

Nachteil der Darstellungsart Kreuztafel ist die Beschränkung auf das Gif-Format, durch welches zwar die Transparenz genutzt werden kann, andererseits aber nur die geringe Farbtiefe unterstützt wird. Dieser Nachteil tritt vor allem bei Texturen zutage, die neben den grünen auch starke rote (bei Laubbäumen) oder blaue (bei Nadelbäumen) Farbschläge haben. Das Png-Format ist keine Alternative, da es nur von wenigen Systemen interpretiert wird.

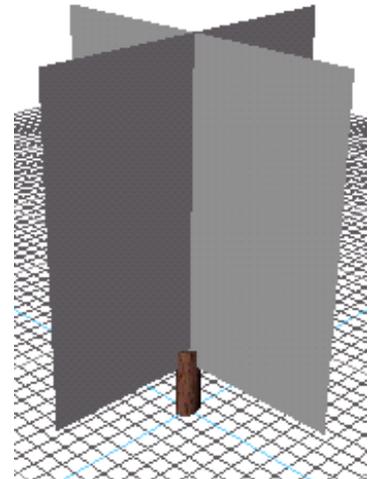


Abb. 41: Kreuzweise angeordnete Flächen ohne Texturen

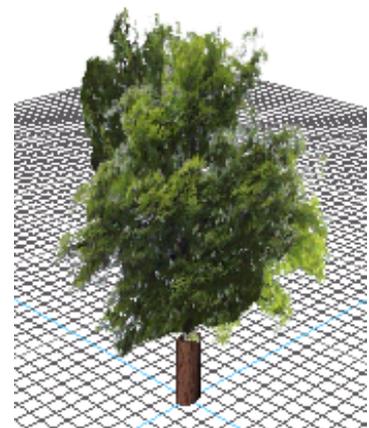
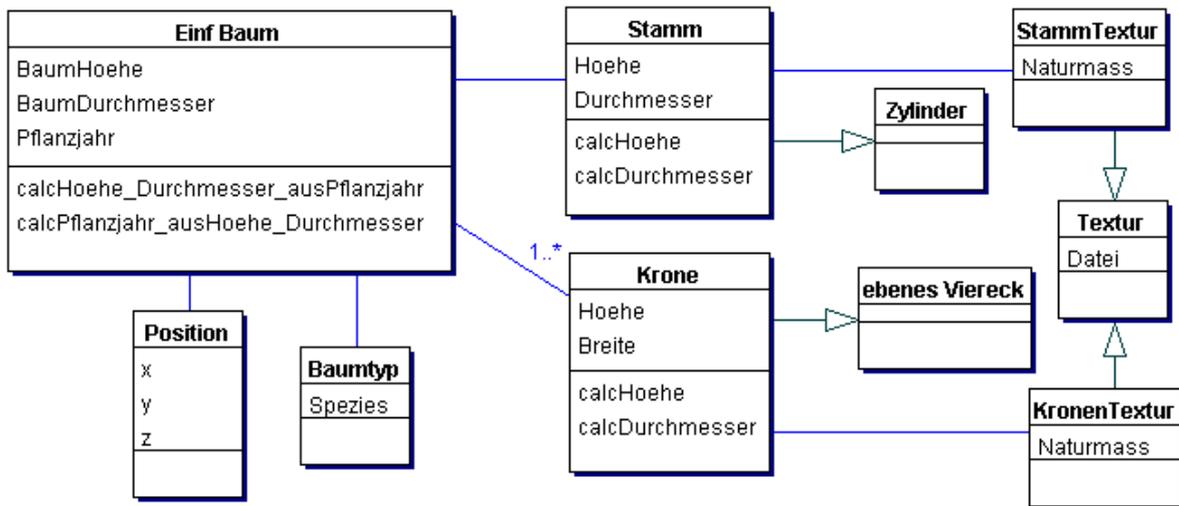


Abb. 42: Beispiel für eine Kastanie in Kreuztafel-technik



Modell 4: Struktur des Kreuztexturmodells

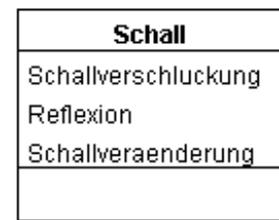
Das zugehörige Modell ist in Modell 4 beschrieben. Es ist gegenüber den vorherigen Modellen nur wenig geändert. Die Klasse Baum wird um die Eigenschaft Kronenanzahl erweitert und die Klasse Krone erbt von ‚ebenes Viereck‘ anstelle von Sphäre und erhält eine Eigenschaft Drehwinkel. Die weiteren Änderungen betreffen die Texturauswahl und sind abhängig von der Baumart, so dass das Datenmodell nicht betroffen ist.

Beim Zweigmodell wird der einzelne Ast modelliert. Hierzu gibt es Programme, wie beispielsweise Treedesigner, mit denen derartige Objekte erstellt werden können. Dabei werden die Bäume Astweise aufgebaut. Am Ende der Aste werden unterschiedliche Laubkörper, in der Regel in Blatt oder Kugelform angebracht. Diese Form der Modellierung hat für den Bereich der automatisierten Darstellung nur sehr geringe Bedeutung und wird daher nicht weiter diskutiert.

4.3.3 Modellanforderung für die Berechnung von Schall- oder Funkwellenausbreitung

Exkurs: Die bis zu dieser Stelle aufgezeigten Gestaltungen dienen der Visualisierung. Es gibt weitere Verwendungen von 3D-Modellen. Betrachtet man beispielsweise die Modellierung zur Berechnung von Schallausbreitung, so tritt die Bedeutung der Textur in den Hintergrund, während die Geometrie in den Vordergrund rückt. Das zuletzt beschriebene Mehrebenenmodell ist, da es eher mit der Textur als mit der Geometrie arbeitet, für diesen Zweck nicht geeignet.

Für solche Anwendungen müssen zusätzliche Eigenschaften modelliert (Modell 5) werden, die bei der Visualisierung keine Rolle spielen. Im Fall der Schallausbreitung sind dies Faktoren wie Schallverschluckung und Reflexion sowie Schallveränderung. Diese lassen sich als Funktion von Baumart, geometrischer Ausprägung und Jahreszeit modellieren. Liegen die Faktoren vor, kann man diese Objekte in Akustiksoftware einführen.



Modell 5: Ansatz für ein Schallmodell

4.4 Texturen für Vegetationsobjekte

Ganz wesentlich für die Qualität der Visualisierung von Vegetationsobjekten ist neben der geometrischen Modellierung die verwendete Textur. Insbesondere wenn die Szene für verschiedene Jahreszeiten modelliert werden soll, kann die Texturerzeugung erheblichen Aufwand erfordern.

4.4.1 Fotografisch erzeugte Texturen

Vegetationsobjekte sind von einem unregelmäßigen Rand gekennzeichnet, sie sind aber nahezu rotationssymmetrisch bezüglich der Z-Achse. Es wird zunächst ein Baum in der XZ-Ebene betrachtet. Dieses Bild wird fotografisch erfasst. Verwendet wird diese Ansicht als Textur des Objekts, so dass, wenn man frontal darauf blickt, eine ebene Fläche mit der Textur als Raum wahrgenommen wird. Um dies zu erreichen, ist es notwendig, die Objektkontur auszuschneiden und innerhalb der Krone „Löcher“ einzufügen und diese nicht zum Objekt gehörenden Bereich mit Transparenzeigenschaften zu versehen. Diese Stellen sind in Abb. 43 rot eingefärbt. Um das Ergebnis automatisch zu erzielen, wird eine Kombination verschiedener Bildbearbeitungsfunktionen verwendet. Dazu ist vorbereitend auf einiges zu achten: Zunächst ist für einen möglichst hellen und weitgehend einfarbigen Bildhintergrund bei der Fotografie zu sorgen. Wichtig ist ein deutlicher hell-dunkel Unterschied zwischen Baum und Hintergrund. Dann bearbeitet man das Bild in der Weise, dass Farbtiefe von 2 entsteht.



Abb. 43: Kastanienkrone mit Transparenzflächen (rot)

Für das Kastanienbeispiel in Abb. 44 ist die so entstandene Form zu sehen. Nun kann ein Maximumoperator auf die beiden Bilder angewendet werden, indem Pixelweise das Graustufenbild mit dem Farbbild verglichen und das jeweils hellste Pixel (bei hellem Bildhintergrund) für ein neues Bild verwendet wird. So entsteht als neues Bild ein Auszug, der zum einen in den Bereichen, die nicht zum Objekt gehören weiß ist und der zum anderen den jeweilige Farb-Ton für den Bereich der Baumkrone beinhaltet. Hier macht sich Unausgewogenheit beim Fotografieren Probleme. Sind dunklere Stellen im Bildhintergrund, werden sie mit diesem Verfahren nicht entfernt und müssen manuell nachbearbeitet werden.



Abb. 44: Differenzmaske für transparente Texturen

Das Verfahren hat den Vorteil, schnell zu einem Ergebnis zu führen. Allerdings sind nicht alle Fotoobjekte geeignet. So kann es bei Objekten, die nicht oder nur selten einzelnstehend in der Natur vorkommen oder aber räumlich nicht verfügbar nicht eingesetzt werden. Weiterhin ist es aufwändig jahreszeitliche Variationen zu erzeugen, ohne den Objektumriss zu verändern, denn dafür muss die gleiche Aufnahmesituation mit dem gleichen Objekt zu unterschiedlichen Jahreszeiten realisiert werden.

4.4.2 Texturen aus L-Systemen

Ein weiterer Weg, um Texturen für Vegetationsobjekte zu gewinnen, ist die Verwendung von Wachstumsmodellen. Das Ziel bei Wachstumsmodellen ist es, Vegetationsobjekte möglichst naturgetreu zu visualisieren. Um dies zu erreichen wächst die virtuelle Pflanze genauso heran, wie die natürliche. Die Regeln für das Wachstum sind in Kap. 4.2.1 dargestellt.

In einer Weiterentwicklung dieses Verfahrens werden zusätzlich Metriken anstelle der Igel-Befehle verwendet, die näher an den natürlichen Parametern liegt und damit den erzeugten Darstellungen der Objekte leichter eine höhere Realitätsnähe gibt [DEUSSEN ET AL. 01].

Dabei werden sowohl Strategien verfolgt, in denen auf bestehende Strukturen Knospen aufgesetzt werden, als auch solche, in denen algorithmisch zu jedem aufgebautem Blatt ein umfangreich gleicher Strang bis zur Wurzel gezogen wird und die Dicke der gesammelten Stränge ein Maß für den Umfang der sie betreffenden Objektteile Ast, bzw. Stamm bildet. Aus diesen Erkenntnissen heraus existiert ein Pflanzenbaukasten mit 11 Bausteinen entwickelt [DEUSSEN ET AL. 01], mit dem fast jede Pflanzenart erzeugt werden kann.

Wenn eine Pflanzenart mittels dieser Wachstumssimulationstechnik erstellt wurde, wird diese mittels bekannter Rendering-Techniken visualisiert und kann so ohne störende Hintergrundbilder weiterverwendet werden.

Dieses Verfahren ist nicht nur zeitlich aufwändig, bei der Erschaffung des Modells einer Baumart. Erfahrung und Kenntnisse der Biologie über Strukturen von Pflanzen, von denen Muster erzeugt werden sollen sind darüber hinaus erforderlich. Ist das Basismodell einmal erzeugt, lassen sich leicht Variationen herstellen. Insbesondere jahreszeitliche Änderungen sind schnell realisierbar, wie Abb. 45 am Beispiel einer Eiche zeigt.



Abb. 45: Jahreszeitliche Variation des gleichen Modells einer Eiche v.l.n.r. : Frühjahr, Sommer, Herbst und Winter

In diesem Beispiel wurde zunächst das Modell Frühling erzeugt. Darauf aufbauend entsteht allein durch Variation der Blättertexturen das Sommer- bzw. Herbstbild der Eiche. Das Winterbild wird generiert, indem die Blätter einfach weggelassen werden.

Ebenso sind unterschiedliche Objektgrößen mit unterschiedlicher Blattdichte und Astzahl durch Variation des Parameters Wachstumsalter schnell zu erstellen. Dies ist in Abb. 46 am Beispiel einer Pappel dargestellt.



Abb. 46: Altersvariation einer Pappel, v.l.n.r. im Alter von 10, 13 und 20 Jahren

Um eine möglichst realistische Darstellung von Objektaggregate zu erreichen, ist es notwendig, wenige sich gleichende Objekttexturen zu verwenden. Dies ist mit dem beschriebenen Wachstumsparameterverfahren leicht umzusetzen, indem die formbestimmenden Größen etwas variiert werden.

Das vollständige Modell entsteht, wenn zu jeder Altervariation die entsprechende jahreszeitliche Veränderung erstellt wird. Auf diese Art und Weise entsteht eine Texturmatrix, aus der je nach Bedarf gewählt werden kann (siehe Kap. 4.4.5).

Vergleicht man die beiden Verfahren, so ergibt sich die folgendes Fazit:

Kriterium	Verfahren	Fotoverfahren	Wachstumsverfahren
Einfachheit		einfach	komplex
Schnelligkeit		hoch	niedrig
Jahreszeitenvariabilität		gering	sehr hoch
Altersvariabilität		gering	sehr hoch
Automatisierungsgrad		sehr gering	Hoch
Kosten		100 € + Kamera	300 €

Tabelle 1: Verfahrensvergleich zur Texturierung von Bäumen

Aus dem Vergleich in Tabelle 2 wird ersichtlich, dass beide Verfahren Vorteile haben. Im Sinne einer durchgängigen Lösung sind Texturen, die mit dem Wachstumsverfahren erzeugt werden vorzuziehen.

4.4.3 Animierte Texturen

Wie in Kap. 3.2.2 beschrieben, besteht die Möglichkeit, Objekte mit animierten Texturen zu versehen. Dies ist vor allem dann sinnvoll, wenn in eine statische Szene etwas Bewegung gebracht werden soll. Vegetationstexturen eignen sich gut für animierte Texturen. Bei ihnen lassen sich Operationen an den Objekten durchführen, wie sie auch natürlicherweise vorkommen, beispielsweise die Bewegung eines Getreidefeldes im Wind, bzw. für die Objektart Baum das Wiegen der Bäume (Abb. 47).

Für diese Effekte sind Fototexturen nicht geeignet. Wird dieses Verfahren um die Aufnahmetechnik Video erweitert, lassen sich die einzelnen Bilder aus Videosequenzen extrahieren. Dennoch ist diese Verfahrensart aus den beschriebenen Gründen sehr aufwändig und schwierig handhabbar.

Wesentlich geeigneter ist das erweiterte Verfahren der Texturen aus L-Systemen. Dabei sind, je nach Windstärke unterschiedliche Texturen vorzusehen. Verwendet man die Technik nach Deussen und Lintermann, so ist im Pflanzenbaukasten unter dem Element Tree die Funktion Gravitropismus vorzufinden, mit der die Neigung der Pflanze mittels einer Verlaufsfunktion bestimmt werden kann.

Bei der Generierung der Bilder ist darauf zu achten, dass sich der Umriss der Textur in der gekreuzten Anbringung über einen möglichst großen Teil der Längen schneidet, da ansonsten der Eindruck von zwei Haupttrieben entstehen kann.

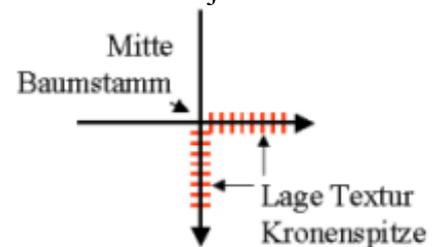


Abb. 47: Einfluss des Windes auf die Lage der Kronenspitze im Bezug auf die Mitte des Baumstammes (Fußpunkt)

Am Beispiel der aus Kap. 4.4.2 bekannten Pappel wird eine Reihe von Texturen in Abhängigkeit von der auf das Objekt wirkenden Windstärke entwickelt (Darstellung verkleinert):

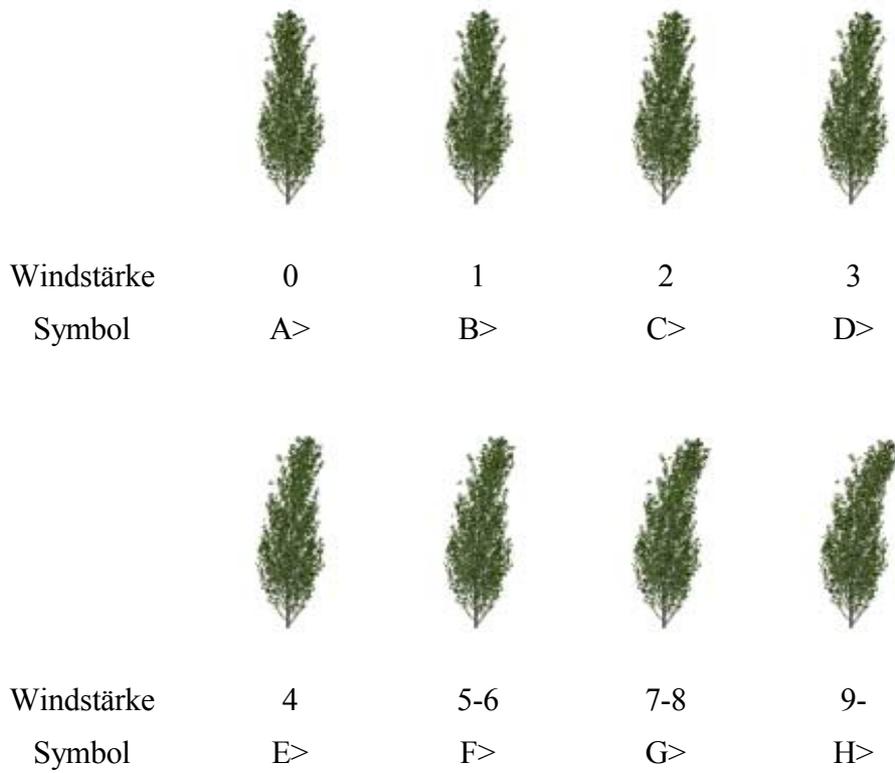


Tabelle 2: Texturierung nach Windstärken

In der Symbolik von Tabelle 2 stehen die Buchstaben für die Neigung der Bäume sowie > für die Richtung der Neigung. Neben der Richtung > existiert noch die Richtung <, welches eine Spiegelung des Bildes um die Vertikalachse bedeutet. Diese Spiegelung ist notwendig, um einerseits die Schwingung in die andere Richtung zu gestalten, andererseits aber auch ein Nachschwingen der Bäume nachzuahmen.. Mit dieser Notation können nun aus den bestehenden Bildern animierte Sequenzen für die jeweiligen Windstärken gebildet werden. Daraus entsteht eine Reihe von Sequenzen im Bezug zur jeweils herrschenden Windstärke (Tabelle 3):

Windstärke	Animation
0	A>
1	A> B>
2	A> B> C> B>
3	A> B> C> D> C> B> A>B<
4	A> B> C> D> E> D> C> B> A>B<
5	A> B> C> D> E> F> E> D> C> B> A>B<
6	A> B> C> D> E> F> F> E> D> C> B> A>B<
7	A> B> C> D> E> F> G> F> E> D> C> B> A>B<C<B<
8	A> B> C> D> E> F> G> G> F> E> D> C> B> A>B<C<B<
9	A> B> C> D> E> F> G> H> G> F> E> D> C> B> A>B<C<B<

Tabelle 3: Anwendung der Texturierungsnotation

Die so bestimmten Daten müssen als Animated Gif prozessiert und dann vorgehalten werden.

4.4.4 Mehrfachtexturierung

Eine weitere Möglichkeit, Texturen zu verwenden, ist die Mehrfachtexturierung. Dabei wird einer Fläche nicht nur eine, sondern zwei oder mehr Texturen zugewiesen. Für die Texturierung der Bäume kann die Mehrfachtexturierung verwendet werden um die Differenzen in der jahreszeitlichen Texturierung zu glätten. Ansonsten müsste die Textur zu einem bestimmten Zeitpunkt jedoch abrupt wechseln.

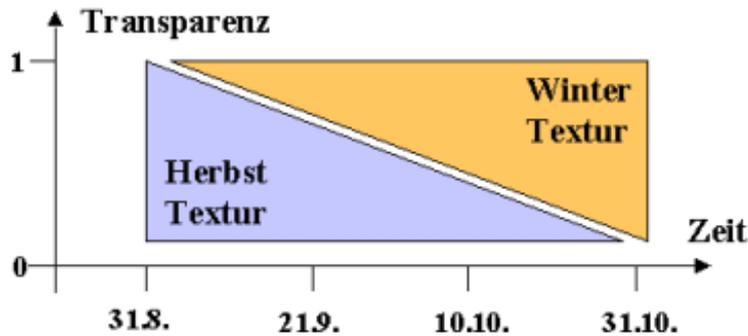


Abb. 48: Beispiel für den Transparenzverlauf zwischen Herbst und Winter

Verwendet man nun alle vier Texturen (Frühling, Sommer, Herbst und Winter), lässt sich über eine zur jeweiligen Textur gehörige Funktion das Auf- und Abschwellen der Transparenz regulieren. Die hier gezeigte Modellierung der Transparenzfunktion beruht auf der kalendarischen Jahreszeitendefinition. Sicherlich besser ist eine Modellierung auf phänomenologischer Basis.

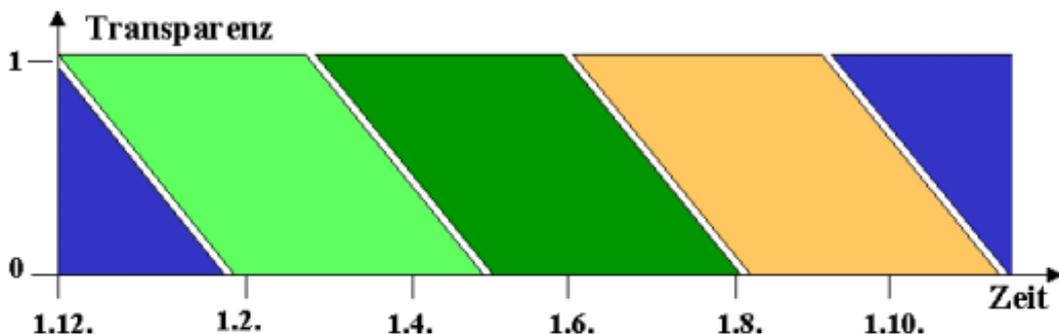


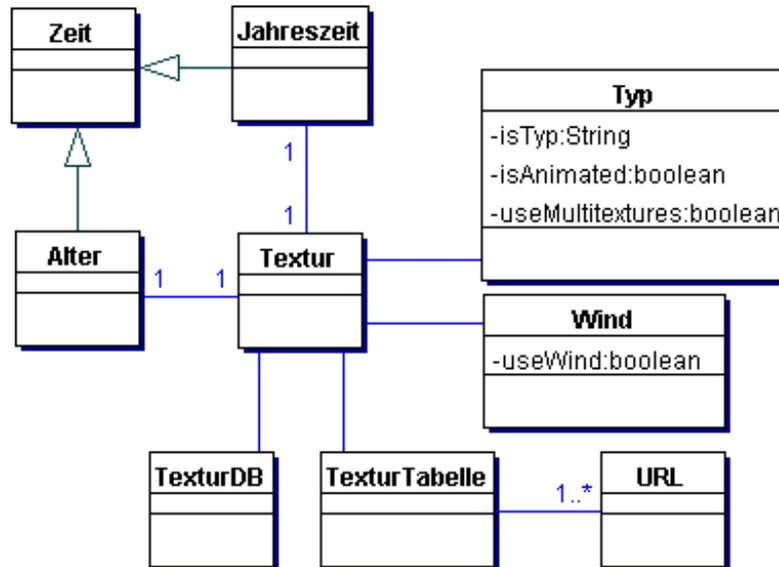
Abb. 49: Funktion für das vierfach Multitexturing für Bäume

Allerdings beinhaltet die Verwendung der Mehrfachtexturierung auch erhebliche Nachteile. So erhöht sich die Anzahl der Texturen und damit der Darstellungsaufwand in der Visualisierung um das mehrfache. Zudem wird Mehrfachtexturierung nicht von allen Implementierungssprachen unterstützt. Daher wird auf die Verwendung von Mehrfachtexturierung verzichtet, obwohl sie das Ergebnis in der Visualisierung verbessert.

4.4.5 Das Texturmodell

Die Erkenntnisse aus den unterschiedlichen Verfahren münden im Texturmodell für Bäume aus ebenen Flächen:

Die Texturen werden durch verschiedene Einflussfaktoren bestimmt. Im wesentlichen sind dies Jahreszeit, Alter und Wind. Diese führen zu Kodierungsparametern, über welche die entsprechend hinterlegten Texturen aus Texturtabellen (Ablageorte in Textur-Filesystemen) bzw. Texturdatenbanken ausgewählt werden.



Modell 6: Texturmodell für Bäume aus ebenen Flächen

Das Modell im Einzelnen: Sowohl Alter als auch Jahreszeit sind Zeiten und Bestandteil des Texturmodells. Zudem hat das Modell weitere Bestandteile. Typ definiert die Baumart und ist damit für die Auswahl aus der Texturtable verantwortlich. Zudem wird über den Typ die Animation und die Mehrfachtexturierung eingeschaltet.

Mit dem Parameter Wind wird bestimmt, ob ein Windmodell verwendet wird und mit welcher Windstärke zu arbeiten ist. Der letzte Bestandteil ist eine Anbindung an ein Speichersystem für Texturen. Dies kann eine Tabelle mit verzeichneten Speicherorten, aber auch eine Bilddatenbank sein.

4.5 Automatisierte Detailstufen

Die darstellbare Bildwiederholrate hängt von der Anzahl der wiederzugebenden Polygone ab. Gleichzeitig nimmt die Wahrnehmung von Objektdetails mit wachsender Entfernung vom Auge ab. Dies macht man sich zunutze, indem man die Repräsentation eines Objektes nach Polygonzahl strukturiert und die notwendige Zahl nach dem gewünschten Detailreichtum auf einer Entfernungsskala festlegt. Die Verteilung auf der Entfernungsskala kann in Abhängigkeit von der Objektdichte variiert werden.

Im Falle der Kugelbäume gibt es die drei verschiedene Repräsentationen, Nebenkronenmodell, Kugelbaummodell mit Textur, Kugelbaummodell mit Farbe (siehe Kap. 4.1), die für je eine Detailstufe stehen. Eine vierte Detailstufe ist durch „keine Repräsentation“ gegeben.

Für das Intervall $[0, \infty]$ z.B. gilt folgende Verteilung (Tabelle 5):

Detailstufe	Ausprägung	Entfernungsintervall
LOD0	Nebenkronenmodell	[0; 100[
LOD1	Kugelbaummodell mit Textur	[100; 300[
LOD2	Kugelbaummodell mit Farbe	[300; 1000[
LOD3	Ohne	[1000, ∞ [

Tabelle 4: Beispiel für eine entfernungsabhängige Detailstruktur mit Kugelbaumstrukturen

Die Anzahl der Polygone ist der wesentliche berechenbare Faktor, mit dem die Leistungsfähigkeit graphischer Anzeigesysteme begrenzt wird.

Die mittels Kugelform erzielten Ergebnisse sind von der Anschauung her akzeptabel, diese Modellierung ist aber in der Regel nicht geeignet, Baumgruppen, Alleen oder gar ganze Wälder darzustellen. Die Ursache dafür ist der hohe polygonale Aufwand, der betrieben werden muss, um eine Kugel zu erstellen, denn die in den Modellen verwendeten Kugeln bestehen aus 224 Polygonen. Für diesen Fall ist eine Modellierung zu finden, deren Berechnungsaufwand nicht so hoch ist.

Eine mögliche Strukturierung mit vernünftigen Ergebnissen bei einer 1024x768 Pixel Bildschirmauflösung und einem 90° Blickfeld und durchschnittlich 10m Baumhöhe ist die folgende:

Entfernung in [m]	Pixel auf dem Bildschirm	LOD	Darstellungsart	Polygonzahl
5 - 40	1024 - 128	0	L-System Modell	bis zu 6000
40 - 100	128 - 50	1	Kreuztextur	8
100 - 1km	50 - 5	2	Billboardmodell	2
1km - 5km	5 - 1	3	Linie als Baumstamm	0
> 5km	< 1 pixel	4	Keine Darstellung	0

Tabelle 5: Detailliertes Modell der LoD Struktur

Diese Strukturierung erfordert bei einer vollständig mit Bäumen bedeckten Fläche näherungsweise folgende Polygonzahl, richtiger ist jedoch eine Berechnung nach der Segmentformel:

- durchschnittlicher Platzbedarf je Baum mit 25 m² Kronenfläche = 0.004 Bäume/m²
- L-System Modell für 40 x 40 m² = 1600 m² = durchschnittlich 2.5 Bäume
- KreuzTafel für 10000 m² - 1600 m² = 8400 m² = durchschnittlich 13.4 Bäume
- Tafeltextur für 100000 m² - 10000 m² = 99000 m² = durchschnittlich 1584 Bäume
- Polygonzahl = (2.5 * 6000) + (13.4 * 8) + (1584 * 2) = 15000 + 107 + 3168 = 18275 Polygone (gerundet)

Moderne PC-Grafikkarten sind in der Lage, typischerweise 100.000 Polygone darzustellen. Für die Vegetationsdarstellung sind somit etwa 1/5 der Gesamtleistungsfähigkeit erforderlich. Der Wert 100.000 Polygone stellt allerdings einen Grenzwert dar, für eine solche Darstellung enthält jedes dargestellte Bild ein Volumen von 15 MByte, was einer Bandbreite von 450 MByte/s bei einer Bildwiederholrate von 30 Bildern/s entspricht. Die Qualität der Darstellung wächst mit zunehmender Polygonzahl und Steigerung der Bildwiederholrate.

Das obige Beispiel für eine LOD Struktur wurde heuristisch gefunden. Um die Leistungsfähigkeit vorhandener Hardware optimal zu nutzen, wird ein Verfahren mit fließenden LOD Intervallen vorgeschlagen. Dies ist möglich, da nur bestimmte Objektarten für die Darstellung zugelassen sind. Eine optimale Auslastung der Grafikkarte wird mit 80% angenommen, dies entspricht 80.000 Polygonen.

In einem ersten Ansatz werden den dargestellten Objektarten Anteile an der verfügbaren Polygonzahl gegeben. Die verfügbare Polygonzahl wird nach dem Engpassverfahren, wie es aus der Kosten- und Leistungsrechnung bekannt ist, optimal ausgenutzt. Dazu wird von dem Prinzip ausgegangen, dass alle Objekte ab einer bestimmten Größe dargestellt werden müssen. Als Beispiel sei dieses Verfahren auf die obige Strukturierung angewendet. Bei einer verfügbaren Polygonzahl von 32.000 liegt dem folgende Berechnung zugrunde:

1. Schritt: Gesamtzahl der darzustellenden Bäume = 1600, dies bedeutet bei einem polygonalem Aufwand von 2 einen Verbrauch von 3200 Polygonen
2. Schritt: Ein Rest von 28.800 Polygonen verbleibt. Auf die nächste LOD Stufe angewendet bedeutet dies eine Gesamtpolygonzahlverbrauch von 12.800, d.h. alle Bäume können mindestens in der LOD-Stufe 1 dargestellt werden. Es verbleiben 19200 Polygone.
3. Schritt: Mit 19200 Polygonen lassen sich drei Bäume der höchsten LOD Klasse darstellen

Das Ergebnis lautet also:

LOD0 [5; Entfernung zum 3. Baum]

LOD1] Entfernung zum 3. Baum; ∞[

Der Nachteil an dieser Vorgehensweise ist die Polygonzahl für Bäume aus dem L-System Verfahren. Die Zahl der Polygone steigt natürlicherweise mit dem Alter der Baumobjekte. Die zuvor angegebene Zahl von 6000 Polygonen gehört zu einem eher jungen Baum. Bereits ein Baum im mittleren Alter wird mit zwischen 125.000 und 160.000 Polygonen dargestellt, bei älteren Bäumen werden schnell 250.000 und mehr Polygone erreicht (zum Vergleich: in animierten Trickfilmen wie beispielsweise „SHREK“ haben die Baumobjekte bis zu 800.000 Polygone und mehr, dort wird allerdings auch kein Real-Time Rendering notwendig).

Die pragmatische Lösung für die Darstellung lautet:

1. Einzelstehende Bäume werden in Kreuztafel-Modellierung dargestellt.
2. Waldbäume werden in Tafel-Technik visualisiert.
3. Ab einer Entfernung in der die Bäume mit maximal 3x3 Pixels dargestellt werden, wandelt die Darstellung in einen Strich

Diese Lösung gewährleistet, dass möglichst wenige Polygone dargestellt werden und dabei ein möglichst realitätsnahe Visualisierung erzeugt wird. Auch bleibt eine ausreichende Kapazität für die Gebäudedarstellung übrig.

4.6 Aggregierte Vegetationsobjekte

Die Modellierung beschränkte sich bisher auf Einzelbäume. Wald kann in der einfachsten Form als Aggregation von Einzelbäumen betrachtet werden. Er ist aber auf der einen Seite mehr, da er mit anderen Objekten, beispielsweise umliegende Wiesen (Sukzessionszone) interagiert, auf der anderen Seite ist es sehr aufwändig und mühselig, innerhalb eines Waldes jeden einzelnen Baum zu modellieren. Der Wald hat also Eigenschaften und Funktionen, die ihn vom einfachen Baum unterscheiden. Weitere Aggregate aus Einzelbäumen sind Baumreihen und Hecken.

Ein Wald besteht für die Modellierung aus zwei Regionen:

- Ein Waldkern stellt den inneren Bereich eines Waldes dar. Er besteht aus einer Sammlung von Einzelbäumen. Im Waldkern existiert üblicherweise ein bestimmtes Höhenintervall, sowie einer Verteilung von Einzelbäumen, die sogenannte Baumdichte. In Abb. 50 ist die Karte eines Waldobjektes mit unterschiedlichen Waldkernexemplaren dargestellt. In rot, grün und blau sind unterschiedliche Kernbereiche mit jeweils gleichen Mischungsverhältnissen an Baumarten, Baumhöhenintervallen und Baumdichten zu sehen.
- Der Waldrand stellt eine Sukzessionszone dar. Natürlicherweise beginnt der Wald am Rand des Waldkerns und wird dann nach und nach in das neben dem Wald liegende Objekt, beispielsweise eine Wiese überführt. Soll ein Wald keinen Waldrand besitzen, dann wird der gesamte Wald als Waldkern erstellt. In Abb. 50 ist die Sukzessionszone gelb dargestellt.

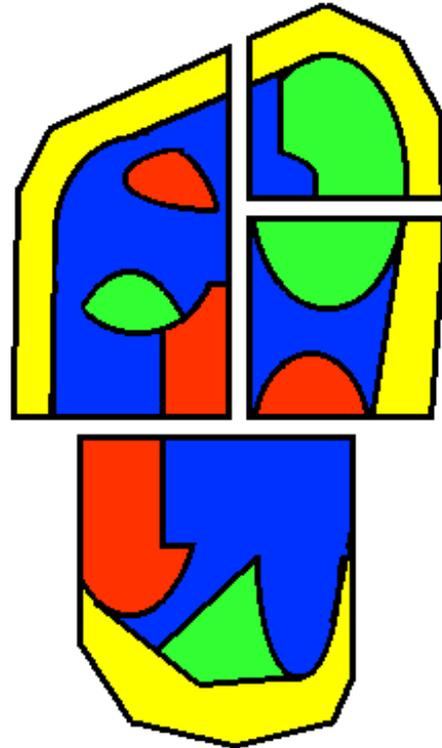


Abb. 50: Waldfläche belegt mit Waldrand und unterschiedlichen Baumarten

Ähnlich wie Wälder können auch Baumgruppen oder Alleen modelliert werden. Die einzelnen Baumexemplare werden mit Hilfe der aufgeführten Bestimmungsgrößen generiert. So gleicht eine Allee einem Waldkern, dessen Breite der unteren Baumverteilung entspricht.

Die Sicht auf einen Wald läßt sich in ihrer Modellierung auf unterschiedlichen Wegen modellieren. Eine Sichtweise ist die Unterteilung der Klasse Wald in die Zonen Innenbereich (Waldkern) und Außenbereich (Waldrand), die jeweils aus dem Objekt Wald abgeleitet werden. Dadurch erben sie Eigenschaften und Verhalten von Wald. Waldkern und Waldrand werden aus einer beliebigen Anzahl von Bäumen gebildet. Waldrand soll die Fähigkeit besitzen, Baumobjekte zu erzeugen, deren Größe in einer bestimmten Richtung abnimmt.

Eine andere Möglichkeit ist die Betrachtung des Baumes als Bestandteil von Baumgruppe und Wald. In Baumgruppe sind die Operationen von Waldkern und Waldrand vereint. Wald kann aus einem oder mehreren Exemplaren von Baumgruppe gebildet werden. Analog gilt dies für Baumgruppe und Baum. Baumverteilung, Waldart und Baumart sind definiert.

Die Operationen der Baumverteilung benötigen für ihre Berechnungen Werte aus den Klassen Wald und Baumgruppe. Mittels eines Zufallszahlengenerators werden die Lagekoordinaten (x , y) der Bäume berechnet. Diese müssen folgende Randbedingungen erfüllen: die Zahlenpaare sollen innerhalb der Ober- und Untergrenze liegen und den minimalen Baumabstand einhalten.

Höhen für die Bäume werden ebenfalls durch den Zufallszahlengenerator in Abhängigkeit von ihrer Verteilung, sowie den Ober- und Untergrenzen ermittelt. Die Variablen für diese Werte sind die durchschnittliche Baumhöhe und die Baumhöhenvarianz.

Wenn für ein Objekt der Klasse Wald ein Waldrand erzeugt werden soll, wird dieser aus einem Objekt Baumgruppe gebildet. Die Werte für die Eigenschaft Waldtyp sind durch die im Aufzählungstyp Waldart enthaltenen Namen vorgegeben. In der Klasse Baum sind einige Werte für die Eigenschaft Baumart durch die im Aufzählungstyp Baumart enthaltenen Namen angegeben. Zu den bereits angegebenen Baumarten können neue Baumarten hinzugefügt werden. Es muss darauf geachtet werden, dass bei Hinzufügen neuer Baumarten auch entsprechende Texturen hinzugefügt werden.

Kapitel 5

MODELLIERUNG VON GEBÄUDEOBJEKTEN

Gebäude sind selbständig nutzbare, überdachte Bauwerke, die auf Dauer errichtet sind und von Menschen betreten werden können sowie geeignet oder bestimmt sind, dem Schutz von Menschen, Tieren oder Sachen zu dienen. Dabei kommt es nicht auf eine Umschließung durch Wände an. Eine Überdachung allein ist hierbei ausreichend. Die Bauwerke müssen jedoch eine selbständige Begrenzung aufweisen und für sich benutzbar sein [CONSTRUCTION].

Auch unterirdische Bauwerke, die von Menschen betreten werden können und dem Schutz von Menschen, Tieren oder Sachen dienen, werden zu den Gebäuden gerechnet, sind aber im folgenden nicht von Belang.

5.1 Modellierungskonzepte

Für die Modellierung von Gebäuden stehen die Konzepte Volumenmodell und Flächenmodell zur Verfügung

Das Volumenkonzept ist eine 3-D-Modellierungstechnik, bei der neben äußeren Kanten und Flächen auch innere Strukturen von Körpern darstellbar sind. Die Beschreibung von raumbezogenen Daten mittels Volumenmodellen ist das aufwändigste Verfahren. Ein Volumenmodell beschreibt z.B. ein komplexes Objekt durch zusammengesetzte Teilkörper. Gängige Modellierungstechnik im Computergraphikbereich ist die Constructive Solid Geometry (CSG) [SCHULZE 96].

Ein Flächenmodell ist die Beschreibung eines Körpers durch Flächen im Raum. Es enthält alle Flächen eines Körpers, sowie deren Beschreibungen und Verknüpfungen zu anderen Flächen. Das Objekt wird also durch seine Randflächenbeschreibung definiert. Es werden i.d.R. nur die von dem Betrachtungspunkt sichtbaren Teile des Objektes gezeigt [SCHULZE 96].

Im folgenden wird eine grobe Modellierung eines einfachen Gebäudes jeweils in beiden Modellvarianten beschrieben.

5.1.1 Volumenmodell

Beim Volumenmodell, CSG-Modell (Constructive Solid Geometry) genannt, wird das Objekt als aus Teilkörpern bestehend betrachtet. Mit Geschosshöhe und Grundfläche werden Quader gebildet. Diese aufeinander gesetzt ergeben den Hauskörper. Der Dachkörper wird gebildet, indem aufsetzend auf den Gebäudeumriss und ausgehend von den definierten Trauflinien ein Körper mit dreieckigem Querschnitt gebildet wird. Die Steigung der Fläche ist durch die Dachneigung vorgegeben. Die Dachhöhe ergibt sich so aus:

$$h_D = \frac{b_G}{2 \cdot \tan(90 - \alpha)}$$

mit h_D - Dachhöhe b_G – Dachbreite
 α – Dachneigung an der Traufe

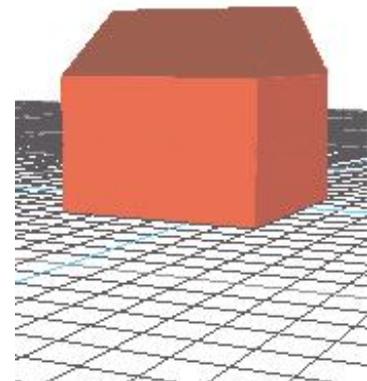
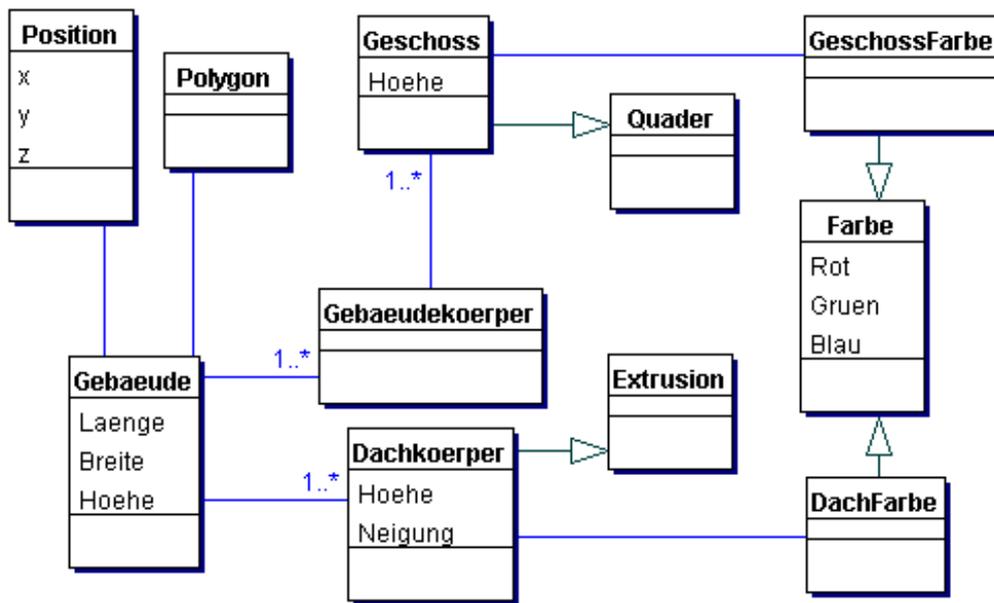


Abb. 51: Einfache Hausgeometrie mit Farben

Das in UML ausgedrückte zugehörige Modell ist in Modell 7 dargestellt.



Modell 7: Einfaches Gebäudemodell in Volumenmodelltechnik

Ein Gebäude nach Modell 7 entsteht aus einem Polygon, und besitzt einem Gebäudekörper sowie einem Dachkörper. Das Polygon steht dabei für den Grundriss, aus diesem müssen die Werte für Breite und Länge des Quaders (entsprechen den Geschossen bzw. dem unbedachten Mauerwerk) und des Daches entnommen werden. Der Dachkörper ist eine Extrusion einer dreieckigen Grundfläche. Die Materialeigenschaften werden durch die Einfärbung modelliert, jeweils eine Farbe für Dachkörper und eine für Mauerwerk.

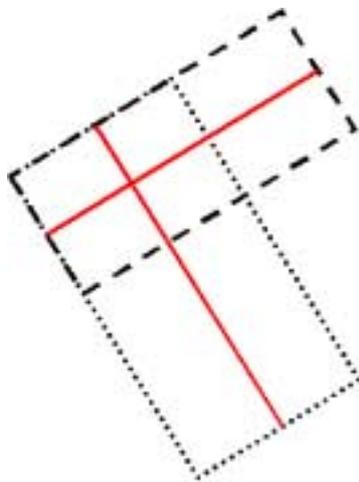


Abb. 52: Rechteckige Grundrisse für Gebäudeteile aus dem Gesamtgrundriss

Mit wenigen Modifikationen dieses Modells sind zusammengesetzte Gebäude möglich. Dabei wird aus dem bisherigen Objekt Gebäude die Objektart Gebäudeteil. Das Gebäude ist dann eine Aggregation aus Gebäudeteilen. Die Grundrisse der Gebäudeteile ergeben sich als Rechteckelemente aus dem Grundriss des Gesamtgebäudes. Abb. 52 zeigt am Beispiel eines L-förmigen Gebäudes wie die Teilgrundrisse entstehen. Diese Teilgrundrisse sind durch gestrichelte bzw. gepunktete Linien dargestellt, die Firstlinien sind in rot gehalten. Dies gilt jedoch nur, soweit die sich überschneidenden Teilgrundrisse nicht über die Projektion der Firstlinie (Gebäudemitte) hinausragt. In solchen Fällen sind Modifikationen des Daches notwendig, da sich die Dächer in der Gebäudemitte ansonsten überschneiden.

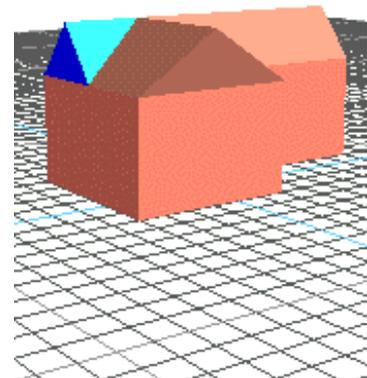


Abb. 53: Komplex aggregiertes Gebäude nach der Anwendung des und-Operators

In solchen Fällen sind Modifikationen des Daches notwendig, da sich die Dächer in der Gebäudemitte ansonsten überschneiden.

Um diesen Fall zu vermeiden, kann man zunächst auf die beiden Teilgebäude logische Operatoren zum Beispiel den und-Operator anwenden. Aus einem Teilgebäude entstehen im Überschneidungsfall drei Teilobjekte, sonst nur zwei. Diese Teilobjekte sind neu zu analysieren, da

solche, die keine Verbindung mit dem Boden mehr haben, gelöscht werden müssen. In Abb. 53 ist dieser Vorgang dargestellt, die zu löschenden Teilobjekte sind blau eingefärbt. Teilobjekte die vollständig in einem zweiten Gebäudeteil (aus dem anderen Teilgebäude) enthalten sind, müssen ebenfalls gelöscht werden.

5.1.2 Flächenmodell

Das unter Verwendung der Volumenmodelltechnik erstellte Gebäudemodell ist auch in Flächen- bzw. Wandmodelltechnik modellierbar, die auch B-Rep (Boundary Representation) genannt wird.

Das Grundrisspolygon wird durch seine Zwischenstücke in einzelne Linien segmentiert. Jede Linie bildet die Basis zu einer Wand. Diese Basis wird in Lotrichtung zu einer Fläche über die Firstlinie hinaus verlängert (Abb. 54).

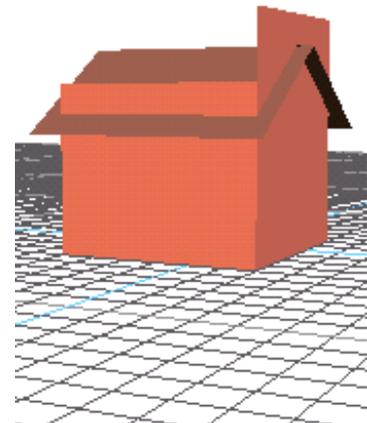


Abb. 54: Konstruktion eines Gebäudes als Flächenmodell

Das Dach wird, im Gegensatz zum Volumenmodell, nicht in Verbindung mit dem Giebel als Dachkörper gesehen, sondern die Dachflächen selbst bilden das Dach. Aufsetzend auf dem Ende der längsten Seite des obersten Geschosses, wird die Dachfläche unter Einbehaltung der Dachneigung bis zur Gebäudemittelachse verlängert.

Anschließend kann die Mauer auf das richtige Maß reduziert werden. Dies wird durch Bildung der Schnittlinie zwischen Dach und Mauerelement und Reduzierung des Mauerelements bis zu dieser Linie realisiert (Abb. 55).

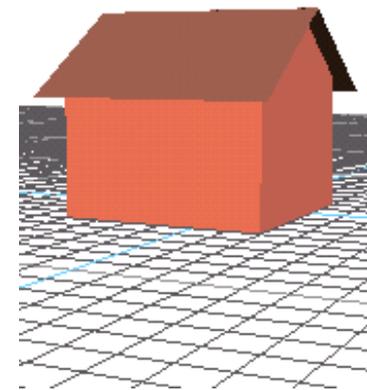
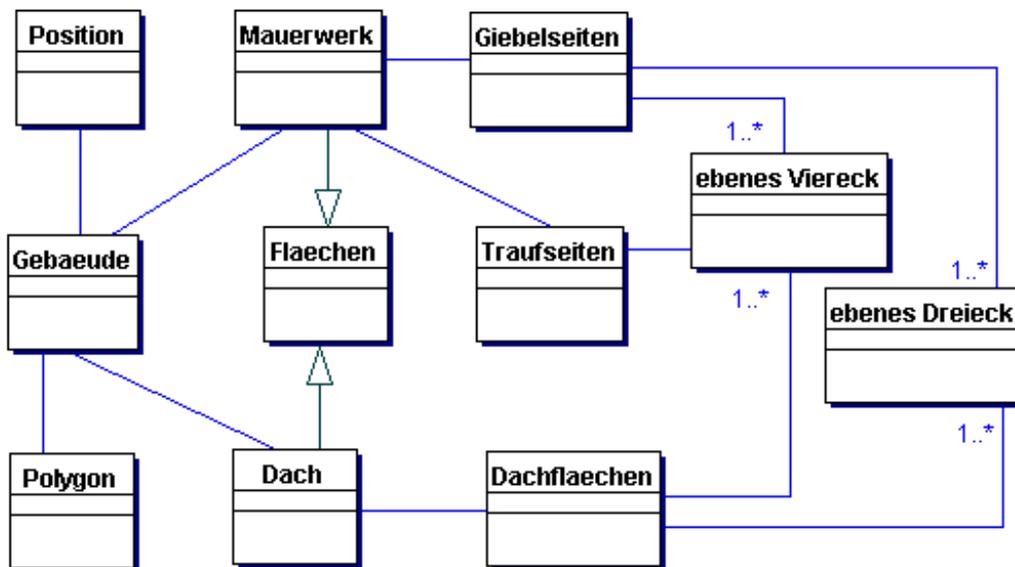


Abb. 55: Gebäude als Wandmodell

Im Modell 8 ist eine Modellierung für ein flächenhaftes Gebäude dargestellt. Im wesentlichen ist die Trennung in Mauerwerk und Dach zu sehen, die aber wiederum beide auf ebenen Dreiecken und Vierecken basieren. Der Gebäudegrundriss wird durch ein Umrisspolygon beschrieben. Jeder einzelnen Fläche können expli-



Modell 8: Grobmodell für ein flächenhaft ausgestaltetes Gebäude

zit Materialeigenschaften zugewiesen werden. Auf eine Beschreibung der Anbindung von Materialeigenschaften wird hier der Übersichtlichkeit halber verzichtet. Diese ist Bestandteil von Modell 10.

5.1.3 Gegenüberstellung

Der Vorteil der Volumenmodelltechnik unter den Aspekten dieser Arbeit ist die einfache Bestimmung des Gebäudeobjekts. Ein wichtiger Nachteil ist die grobe Strukturierung des Modells. Sie erlaubt nur eine Materialzuweisung für einen Volumenkörper und ermöglicht keine Differenzierung, so dass z.B. der gesamte Dachkörper, bestehend aus Giebel und Dachflächen dieselbe Materialeigenschaft (Farbe, Textur) erhält (Abb. 56).

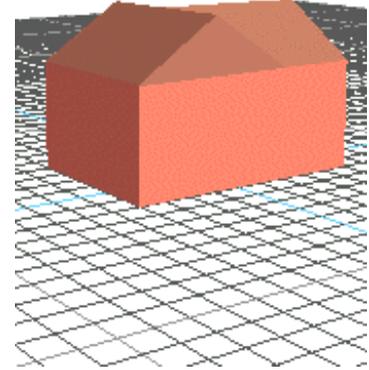


Abb. 56: Gebäude in Volumenmodelltechnik

Nachteil der Flächenmodelltechnik ist, dass die gebildeten Objekte keine gültigen Körper sind. Zudem werden Informationen über das Innere bzw. Äußere nicht direkt definiert sondern müssen explizit berechnet werden. Besonderer Vorteil ist die Zuweisungsmöglichkeit von Materialeigenschaften zu jeder einzelnen Fläche, genauso wie zu Flächenverbänden.

Vergleicht man Volumen- und Flächenmodell, wird die Flexibilität des Flächenmodell-Ansatzes sowie die Variabilität im Detail, besonders bei den Materialeigenschaften deutlich. Dieser Ansatz wird im weiteren verfolgt.

5.2 Modellierung der Gebäudebestandteile

Zur besseren Modellierung, der Materialeigenschaften wird das Gebäude in seine Hauptbestandteile Mauerwerk und Dach zerlegt und getrennt bearbeitet. In dieser Arbeit werden als Mauerwerk alle von außen sichtbaren, senkrecht stehenden Gebäudebestandteile bezeichnet. Üblicherweise besteht das Mauerwerk an der Außenseite aus Stein, Putz, Glas, Kunststoff oder Holz. Im Mauerwerk sind Öffnungen für Fenster und Türen zu finden. Die schmale Mauerwerksseite ist die Giebelseite, die lange Seite wird Traufeseite genannt.

Als Dach wird der Gebäudeteil angesehen, welcher das Gebäude nach oben hin abschließt. Das Dach besteht in der Regel aus einem Dachstuhl aus Holz oder Beton, einer Lattung, sowie dem Deckmaterial aus Ziegel, Kunststoff, Wellblech, Schindeln oder anderen Materialien. Eine Kante, an der das Dach zu beiden Seiten abfällt, ist der Dachfirst. Die unteren Dachkanten sind die Dachtraufen, die Giebelseite wird durch den Ortgang abgeschlossen.

Mauerwerk und Dach sind fest miteinander verbunden und haben eine umlaufende Stoßkante miteinander. Daher stehen einige der Punkte, aus denen Mauerwerk und Dach gebildet werden, in einem geometrischen Zusammenhang zueinander.

5.2.1 Mauerwerk

Um die Geometrie des Mauerwerks zu bestimmen, sind Maße notwendig. Aus dem Grundriss sind Gebäudebreite G_B und -länge G_L ablesbar. Weitere Maße sind die Traufhöhe T_H die Mansardhöhe M_H und die Giebelhöhe G_H (vgl. Abb. 57).

Diese Maße ergeben den in Abb. 58 gezeigten Koordinatensatz. Dieser lässt sich aus den Maßen bestimmen, die in Abb. 57 beschrieben sind. Zu den bereits erklärten Punktbezeichnungen Gr, Tr, M und G kommen nun die Indizes V (vorn), H (hinten), L (links) und R (rechts) hinzu.

Der Einfachheit halber wird ein lokales Koordinatensystem aufgezo-gen. Der Ursprung liegt im Punkt Gr_{VL} , die x-Achse liegt in Richtung von Gr_{VR} , die y-Achse in Richtung Gr_{HL} , die z-Achse in Richtung Tr_{VL} .

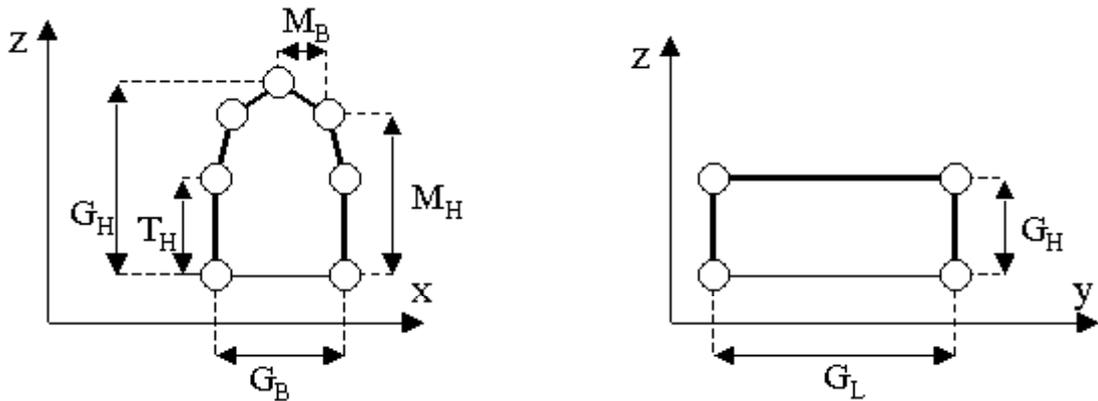


Abb. 57: Maße des Mauerwerks

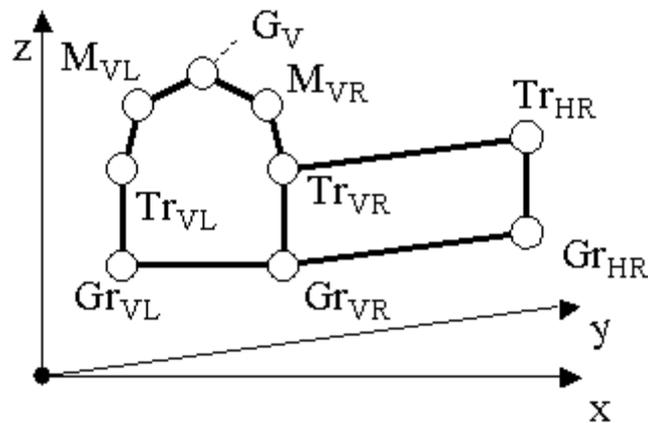


Abb. 58: Gebäudepunkte für das Mauerwerk

Der gesamte Koordinatensatz für das Mauerwerk ist in Tabelle 6 dargestellt. Die Größen für die Bestimmung der Koordinaten x , y , z sind aus Abb. 57 abgeleitet. Aus den Koordinaten sind leicht Zugehörigkeiten ablesbar. So nimmt Y Werte von 0 oder der Gebäudelänge an, wodurch die beiden Giebelseiten unterschieden werden. Die Z -Werte gibt die jeweilige Höhenebene Gr , T , M , G an.

Lfd. Nr.	Pkt. Bez.	X	Y	Z
0	Gr_{VL}	0	0	0
1	Gr_{VR}	G_B	0	0
2	Tr_{VR}	G_B	0	T_H
3	M_{VR}	$G_B + M_B$	0	M_H

4	G_V	$\frac{1}{2} G_B$	0	G_H
5	M_{VL}	$G_B - M_B$	0	M_H
6	Tr_{VL}	0	G_L	T_H
7	Gr_{HL}	0	G_L	0
8	Gr_{HR}	G_B	G_L	0
9	Tr_{HR}	G_B	G_L	T_H
10	M_{HR}	$G_B + M_B$	0	M_H
11	G_H	$\frac{1}{2} G_B$	G_L	G_H
12	M_{HL}	$G_B - M_B$	0	M_H
13	Tr_{HL}	0	G_L	T_H

Tabelle 6: Ermittlung der Punktkoordinaten des Mauerwerks

Die Giebelpunkte 4 und 11 (G-Ebene) treten nicht zwingend auf, genauso wie die Punkte 3, 5, 10, 12 (M-Ebene). Welche Punkte fehlen, ist von der Dachform abhängig. Liegt ein Krüppelwalmdach vor, so liegt Punkt 4 auf der Linie 3-5 und Punkt 11 auf der Linie 10-12. Soll ein Satteldach modelliert werden, liegen diese Punkttupel in einem Punkt.

Aus dem Koordinatensatz sind die Mauerflächen zu bilden. Verwendet man die von VRML [ISO/IEC14772-1:1997] für den Knoten IndexedFaceSet bekannte Notation: {lfd. Nr., lfd. Nr., ... , -1} (-1 für schließen), werden für die verschiedenen Mauerflächen folgende Polygone gebildet:

Mauerflächen	Polygon
Giebelseite vorn	{0, 1, 2, 3, 4, 5, 6, -1}
Giebelseite hinten	{7, 8, 9, 10, 11, 12, 13, -1}
Gebäudeseite links	{0, 6, 13, 7, -1}
Gebäudeseite rechts	{1, 2, 9, 8, -1}

Tabelle 7: Topologie der 4 Mauerwerksflächen

Die so ermittelten Mauerseiten sind jeweils in unterschiedliche Bereiche waagrecht und senkrecht zu unterteilen. In der horizontalen Staffelung ist zunächst der Sockel zu finden, der bei verputzten Mauerwerken vorkommt. Dieser Sockel hat zum restlichen Mauerwerk andere Materialeigenschaften, wie beispielsweise Sockelputz oder Steinriemchen. Als geometrische Eigenschaft für den Sockel gilt die Sockelhöhe. Der Fall des Souterrain bleibt hier außer acht.

Das nächste Element ist das Erdgeschoss (EG). Dieses Geschoss unterscheidet sich schon durch die Haustüre von den übrigen Stockwerken. Oft besteht die Fassade des Erdgeschosses aus anderem Material als der Rest des Gebäudes und die Fenstergrößen unterscheiden sich von denen des übrigen Gebäudes. Dies ist vor allem bei Gebäuden mit Verkaufsfenstern der Fall. Zudem hat das Erdgeschoss oft eine andere Höhe als die anderen Etagen.

Das letzte Element ist das Dachgeschoss (DG). das Geschoss in dessen Bereich das Dach beginnt, wird als Dachgeschoss bezeichnet. Das Dachgeschoss besteht an den Traufen aus einer in

der Regel fensterlosen Fassade, während auf der Giebelseite üblicherweise Fenster zu finden sind. Der Anteil des traufeseitigen Mauerwerks vom obersten Geschoss bis zum Dach wird mit Kniestockhöhe bezeichnet.

Die Giebelfenster sind oft von geringerer Anzahl und kleinerer Größe als die der übrigen Geschosse. Ihre Zahl richtet sich nach der zur Verfügung stehenden Mauerwerksbreite in Höhe der Fensteroberkante. Dabei wird, bei mehreren vorhandenen Fenstern, der Fensterabstand des Gesamtgebäudes beibehalten und die Fensterlage zentriert.

Die vertikale Staffelung dient der Ausgestaltung von Bereichen, die stockwerksübergreifend sind, wie z.B. dem Treppenhaus. Die Fassade von Treppenhäusern ist üblicherweise durch halbstockweise versetzte Fenster oder aber durchgängige Lichtelemente, beispielsweise Glasbausteine, gekennzeichnet.

Aus dieser Strukturierung ergeben sich die Größen, die in Tabelle 8 aufgelistet sind.

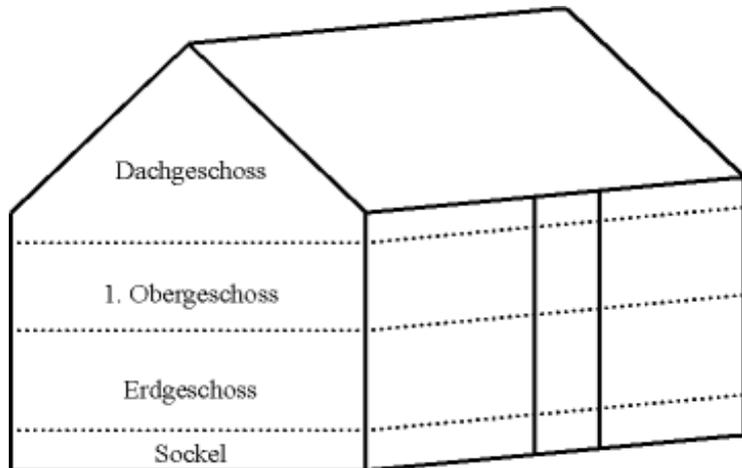


Abb. 59: Struktur der Fassade

Fassadenelement	Bezeichner	Datentyp
Sockel	S	Bool
Sockelhöhe	S_H	Float
Erdgeschoss	EG	Bool
Erdgeschosshöhe	EG_H	Float
Geschosshöhe	G_H	Float
Geschoszahl	GZ	Int
Treppenhaus	Th	Bool
Treppenhausbreite	Th_b	Float
Treppenhauslage	Th_l	Float

Tabelle 8: Bestimmungsgrößen für Fassadenelemente

Der Datentyp Bool = TRUE kennzeichnet das Vorhandensein des jeweiligen Teilobjektes und damit die Aktivierung der zugehörigen Größen. Fassadenelemente vom Typ Float bzw. Int stellen geometrische Größen dar.

5.2.2 Dach

Die Dachform ist für die Modellierung eines Gebäudes bedeutend, da sie giebelseitig die Geometrie des Mauerwerks bestimmt. Die verschiedenen Dachbestandteile (Abb. 60) sind nicht bei jedem Dach vorhanden, vielmehr wird durch ihre Anwesenheit die Dachform bestimmt.

Es existiert eine Vielzahl von Dachformen. Von dieser Vielfalt werden einige übliche Formen nach [BAULEXIKON] [BIW], [SCHUNCK ET AL 01] und [WERNER 93] modelliert. Die Modellierung weiterer Dachformen ist nach den hier angewandten Regeln ohne weiteres möglich.

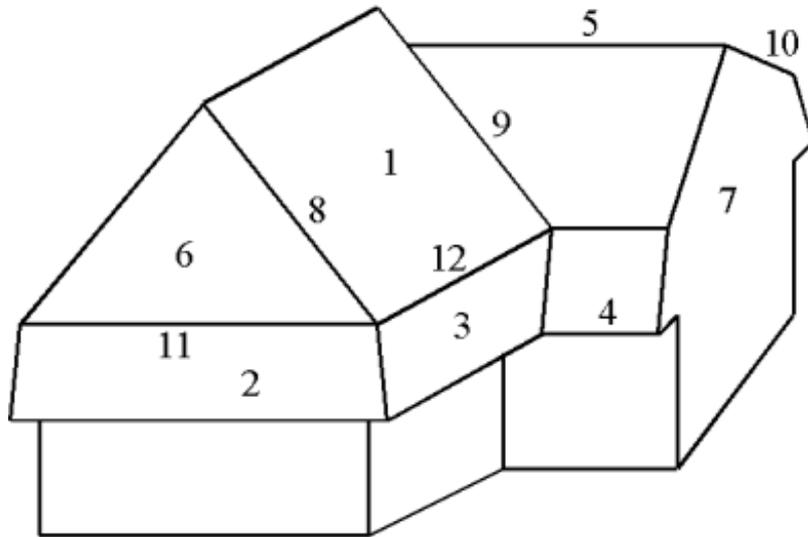


Abb. 60: Bezeichnungen von Dachelementen: 1 – Hauptdach, 2 – Mansarde-Walm, 3 – Mansarde, 4 – Traufe T, 5 – First F, 6 – Walm W, 7 – Giebel, 8 – Grat G, 9 – Kehle, 10 – Ortgang O, 11 – Walmbruch, 12 – Dachbruch Db

Ausgelassen werden beispielsweise Dächer mit rundlichen Formen, wie das Tonnendach. Für weiterführende Literatur zum Thema Dachformen s.u. [DASDACH].

Satteldach

Die in den bisherigen Beispielen verwendete Dachform ist ein am Giebel und den Traufen oft leicht überstehendes Dach mit glatten Dachseiten, variiert durch die im Bebauungsplan bzw. in der Gemeindefestsetzung angegebene oder ortsübliche Dachneigung, ein sogenanntes Satteldach (Abb. 61). Die Länge des Dachfirstes orientiert sich dabei am Maß der längsten Seite des Grundrisses. Satteldächer weisen mehrere, oft regional unterschiedliche Dachneigungen auf [BAULEXIKON]:

1. flaches Satteldach Dachneigung $\leq 30^\circ$
2. als "neudeutsches Dach" oder "Winkeldach" wird es mit 45° Dachneigung bezeichnet
3. ein Satteldach mit 60° Dachneigung wird es als "altfränkisches" auch "altfranzösisches" Dach bezeichnet. Diese Satteldachform findet man überwiegend im fränkischen Raum wieder.
4. mit 62° Dachneigung wird ein Satteldach als "gotisches" oder "altdeutsches" Dach bezeichnet

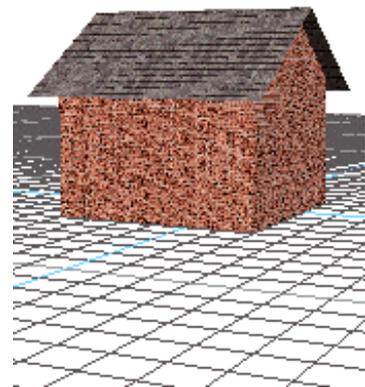


Abb. 61: Satteldach

Flachdach

Die einfachste Dachform ist das Flachdach. Für die Modellierung des Flachdachs wird einfach eine ebene Fläche in Form des Grundrisses gebildet. Diese Fläche wird etwa auf einer Höhe ca. 20cm unter der Gebäudehöhe angebracht.

Die Dachform Flachdach (Abb. 62) hat große Bedeutung bei mehrgeschossigen (>4 Stockwerke) Wohngebäuden, Gewerbe- und Bürobauten sowie Einfamilienhäusern (Bungalows), die Anfang der 70er Jahre gebaut wurden. Flachdächer werden erstellt, indem auf die gegossene Gebäudedecke Schweißbahnen aufgebracht werden, die schließlich mit Kies abgedeckt werden. Daher kann als Textur eine Kiestextur verwendet werden. Maßgeblich für die Anzahl der Texturwiederholung ist, wie in den vorangegangenen Fällen, die Größe der einzelnen verwendeten Kachel in der Natur.

Insbesondere bei höheren Gebäuden sind auf den Flachdächern zusätzliche Einrichtungen angebracht. Das sind beispielsweise quaderförmige Objekte, wie Heizungsaufbauten oder Maschinenräume von Fahrstuhlschächten. Diese Objekte werden, sofern Informationen über ihren Grundriss und die Höhe vorliegen, auf das Flachdach aufgebracht.

Ab einer Geschosshöhe von 8, einem Baujahr nach 1970 und einer Nutzung als Wohn- oder Bürogebäude, kann davon ausgegangen werden, dass sich auf dem Dach Versorgungseinrichtungen befinden, wie dies bei dem in Abb. 63a gezeigten Gebäude aus der Liegenschaft der Fa. Hoechst, Frankfurt der Fall ist. Liegen keine weiteren Daten vor, so lässt sich ein solcher Versorgungsblock (siehe Abb. 63a und 63b) annehmen. Die Parameter für einen Block sind jeweils 66% der Gebäudebreite und -länge. Der Block wird in der Mitte des Grundrisses platziert. Die Höhe des Versorgungsblocks ist eine Geschosshöhe. Ein Beispiel für die Umsetzung dieser Modellannahmen ist in Abb. 63b zu sehen. Als Textur dienen einfarbige Kacheln mit Rand, welche dem Charakter der üblichen Weißblech-, Zink- oder Kupferplatten nahe kommen.

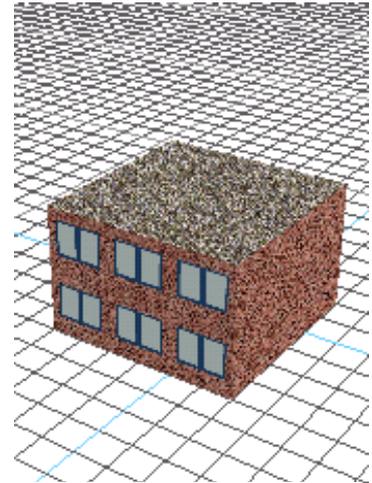


Abb. 62: Gebäude mit Flachdach

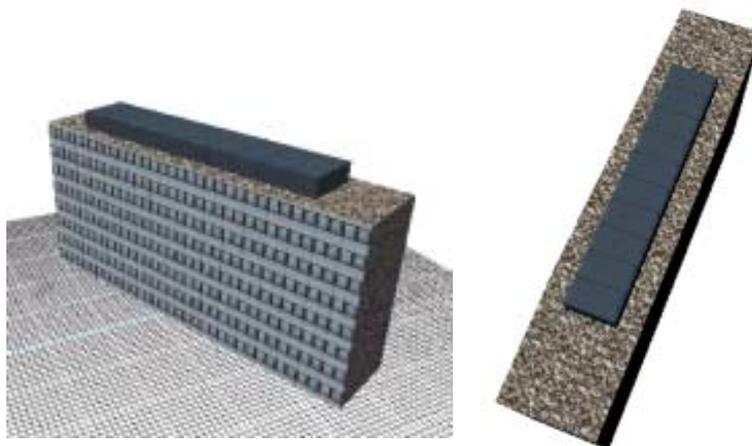


Abb. 63: Größe und Platzierung eines Versorgungsblocks auf einem Hochhaus, in der Schrägansicht (a) und mehr von oben (b)

Des Weiteren sind üblicherweise Antennen- und Blitzschutzanlagen auf Flachdächern zu finden, die jedoch aufgrund von Geringfügigkeit hier keine Berücksichtigung finden.

Pulldach

Diese Dachform ist vorwiegend bei einfachen Gebäuden, wie Schuppen oder Garagen zu sehen. Das Pulldach (Abb. 64) ist die im Mittelmeerraum am häufigsten vorkommende Dachform. Seit der Mitte des 20. Jahrhunderts wird sie auch in Mitteleuropa eine häufig verwendete Dachausführung bei Gewerbe-, Kommunal- und Wohnbauten. Häufig ist das Pulldach auch bei modernen Gebäuden, besonders bei Einzeldachkombinationen oder in Niedrigenergiekonzeptionen vorzufinden.

Die Dachfläche ist nur in einer Richtung geneigt, zumeist mit recht flacher Neigung. Als Bestimmung für ein Pulldach reicht die Kenntnis der Dachneigung oder des Höhenunterschieds zwischen den beiden Pulldachenden aus. Das Pulldach ist also ein Dach mit einem Dachfirst, jedoch nur einer Trauflinie. So gesehen ist das Flachdach eine Sonderform des Pulldaches, bei dem die Neigung = 0 ist.

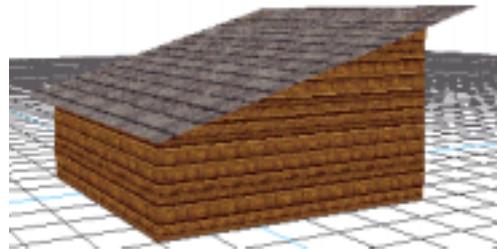


Abb. 64: Pulldach

Bei zwei oder mehrstufigen Pulldachformen größerer, besonders industrieller Bauten sind zwei oder mehr Pulldächer hintereinander angeordnet, gelegentlich auch in der Höhe zueinander versetzt. Vorteil dieser Dachform ist die zusätzliche Belichtungsmöglichkeit des Dachraums durch den senkrechten Außenwandteil. Deshalb wird diese Dachform häufig für Produktionshallen verwendet. Das multiple Pulldach wird auch als Sheddach bezeichnet. In der Modellierung wird jedes Teilpulldach einzeln als Pulldach betrachtet.

Walmdach

Eine weitere Dachform ist das Walmdach. Walmdächer sind in der Regel bei bis zu dreistöckigen Gebäuden zu finden. Bei einem Walmdach ist die Firstlinie gegenüber der langen Gebäudelinie auf beiden Seiten verkürzt. Die Enden der Trauflinien werden dann zum Ende der Firstlinie gezogen. So entsteht ein schräg im Raum liegendes Dreieck, welches die gleiche Textur in gleicher Auflösung wie das übrige Dach erhält.

Es ist ausreichend, die Endpunkte der Firstlinie zu kennen, da sich alle weiteren Gegebenheiten danach richten. Sind in einer Region Walmdächer vorgeschrieben, beispielsweise durch einen Bebauungsplan bzw. eine Gemeindefestsetzung, so ist der Allgemeinfall des Walmdaches (Abb. 65) vorzusehen.

Dieser wird gebildet, indem die Firstlinie als mittlere Linie zwischen den Trauflinien, den längsten Gebäudeseiten, erzeugt wird. Die Firstlinie wird dann um eine halbe Gebäudebreite gekürzt, wodurch der Rand des Walmbereiches eine Winkelhalbierende zwischen Giebelseite und Trauflinie bildet. Damit erreicht man dort die gleiche Dachneigung wie beim übrigen Dach.

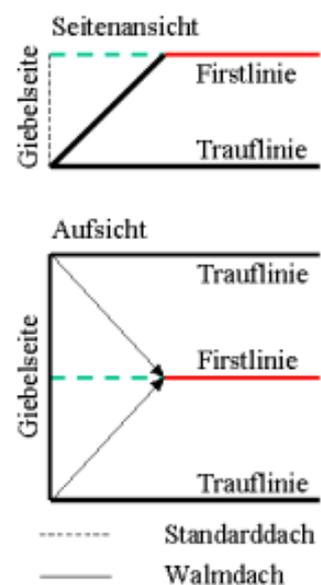


Abb. 65: Bildung eines Walmdaches

Mansardendach

Eine vierte, weit verbreitete Dachform ist das Mansardendach. Das Mansardendach wird durch zwei unterschiedliche Dachneigungen auf jeder Dachseite geprägt, (Abb. 66). Diese Dachneigungen können beschrieben werden, indem zusätzlich zur Trauflinie und zur Firstlinie eine

Dachbruchlinie eingeführt wird. Mittels dieser Dachbruchlinie können zwischen Trauflinie und Dachbruchlinie sowie zwischen Dachbruchlinie und Firstlinie zwei Dachneigungen definiert werden. Auf der anderen Dachseite wird dieses Vorgehen symmetrisch angewendet.

Auch wenn die Informationen über die Dachneigung sowie die Lage der Dachbruchlinie nicht bekannt sind, ist es möglich, ein Mansarddach aus den Informationen Grundriss und Geschosszahl zu erzeugen, wie im Folgenden begründet wird.

Ein Mansarddach dient zur Erstellung von möglichst senkrechten Wänden auch in Räumen, die bereits in das Dach hineinragen. Diese Eigenschaft führt zu einer Neigung von $>60^\circ$, üblicherweise sogar 70° - 80° für den steilwandigen Bereich des Daches. Auf Höhe der Oberkante des letzten Vollgeschosses ist dann die Dachbruchlinie zu finden. Von dort aus wird das Dach mit geringer Neigung zur Firstlinie fortgeführt, üblicherweise von ca. 30° bis zu 10° (Flachdach) Dachneigung.



Abb. 66: Mansarddach

Zeltdach

Bei dieser Dachform wird die Firstlinie auf einen Firstpunkt (Abb. 67) reduziert. Der dazugehörige Grundriss hält üblicherweise von seinen vier oder mehr Eckpunkten gleichen Abstand zum Firstpunkt. Das Zeltdach kann über der Lage und Höhe des Firstpunktes bestimmt werden. In Abb. 67 ist ein Beispiel für ein Zeltdach zu sehen. Zeltdächer findet man häufig auf Türmen, wie Kirchtürmen oder auf Erkern. Bei großer Höhe im Verhältnis zur Traufenlänge heißen sie Turmdach. Vom Zeltdach gibt es einige Variationen. So werden mitunter alle Dächer, die auf einem Punkt zulaufen als Zeltdach bezeichnet. In diesem Sinne sind auch die Sonderformen Mansarde- Zelt-Dach oder Walm-Pult-Zeltdach, aber auch noch einige andere Dachformen, wie Krüppelwalm-Zeltdach und Mansarde-Krüppelwalm-Zeltdach ein Zeltdach.

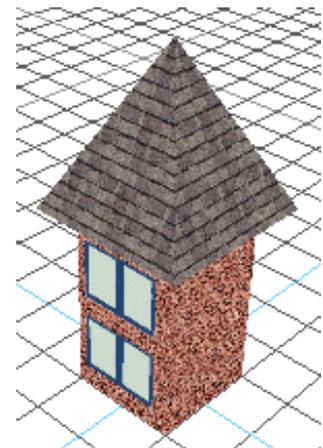


Abb. 67: Zeltdach

Dachkombinationen

Neben eindeutigen Dachkategorien gibt es Mischformen zwischen den Dacharten. Eine Mischform aus Walmdach und Satteldach ist das Krüppelwalmdach (Abb. 68). Bei diesem wird der Walmbereich nicht bis zur Traufe heruntergezogen, sondern endet bereits vorher. Ab dieser Stelle wird die Form des Standarddaches weiterverfolgt.

Bei dieser Dachform ist davon auszugehen, dass der Walmbereich die gleiche Neigung, wie das übrige Dach besitzt. Zudem ist bekannt, dass die den Walmbereich begrenzenden Linien die Firstlinie im Winkel von 90° verlassen. Unter diesen Rahmenbedingungen reicht zur Konstruktion eines Krüppelwalmdaches die Kenntnis der Lage des Firstlinienpunktes aus.

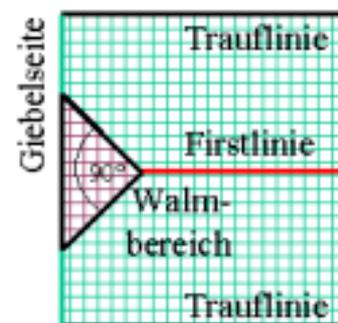


Abb. 68: Krüppelwalmdach in der Aufsicht

Eine andere weit verbreitete Mischform ist die Kombination zwischen Mansarddach und Krüppelwalmdach. Der Walmbereich erstreckt sich ausgehend von der Firstlinie hinunter bis zu den Nebenfirstlinien (Dachbruch), durch deren Verbindung der Walmbereich abgeschlossen wird. Für die Exemplarbildung gelten die zur jeweiligen Dachform gehörigen Regeln.

Zur Modellierung der Krüppelform muss die Firstlänge angegeben werden, deren Differenz zur Dachlänge die Walmlänge ergibt. Des Weiteren ist die Angabe der Walmhöhe notwendig. Weitere Mischformen sind das Walm-Mansarde-Dach und das Pult-Mansarde-Dach.

Die besprochenen Dachformen decken sicherlich nur die häufigsten in Europa gängigen Dachformen ab. Andere Dachformen, z.B. asiatische [UEDA ET AL 99] sind nicht berücksichtigt. Weitere Formen, die durch Kombination der Grundformen entstehen, sind in Abb. 69 dargestellt.

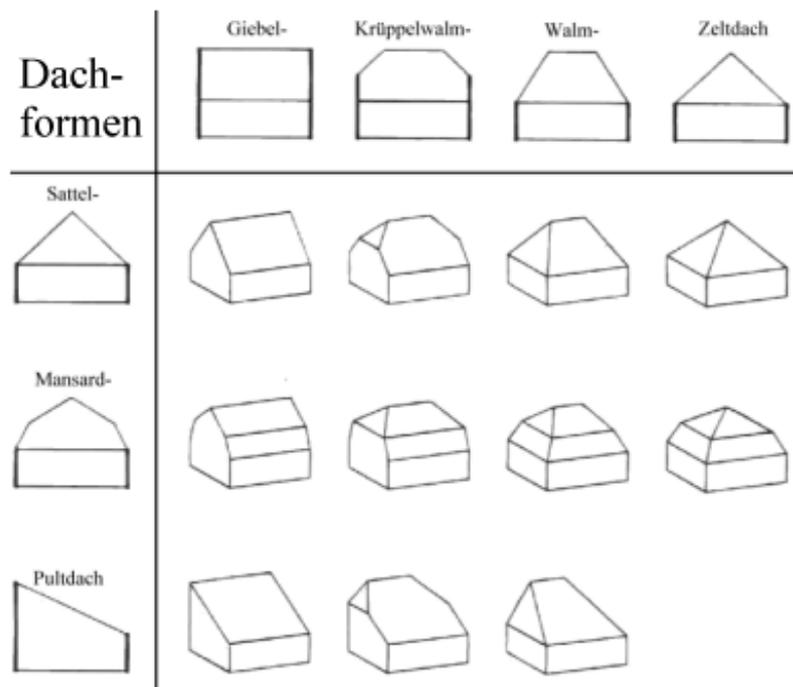


Abb. 69: Kategorien gängiger europäischer Dachformen

Der richtige Dachname ist eine Kombination aus der Dachlängsschnitt (Giebel-, Walm-, Krüppelwalm- und Zeltdach) und der Dachquerschnittbezeichnung (Sattel-, Mansarde- und Pultdach). Üblicherweise werden die Bezeichnungsbestandteile Satteldach und Giebeldach nicht explizit verwendet.

Zum Dach gehören als häufig auftretende Bestandteile Dachgauben. Dachgauben sind kleine Gebäudeteile, die aus dem Dach herausragen und eine eigenständige Dachform besitzen. Üblicherweise will man mit Dachgauben gerade Wände auch in den Räumen erreichen, die in den Dachbereich hineinragen. Dies bedeutet, dass ihr Mauerwerk eine Verlängerung der Außenmauer nach oben bildet, welches von dort ins Dach hineinragt. Die Außenmauer von Dachgauben ist oft nicht mit

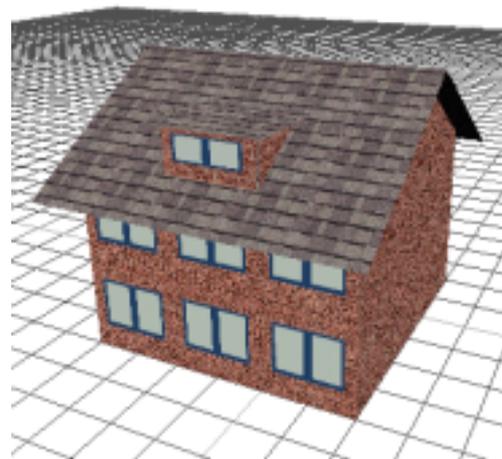


Abb. 70: Gebäude mit Dachgaube

Stein, sondern mit Holz, Kupfer oder Zinkblech verkleidet. Das Dach einer Gaube kann, wie in Abb. 70 dargestellt, ein einfaches Pultdach sein, , aber auch komplexe Dächer, wie ein Krüppelwalmdach. Die Dachgaube wird als unabhängiger Gebäudeanteil modelliert und dann mit dem Hauptgebäude zum Gesamtgebäude zusammengefasst.

Balkone

Ein weiteres wichtiges raumbildendes Element sind Balkone. Allgemeine Daten zu ihrer Modellierung sind schwer zu ermitteln, so dass hier nach fallweiser Beurteilung eine Bearbeitung von Hand vorgenommen werden muss.

Ein Balkon ist eine nach oben offene Box geringer Tiefe mit einer üblichen Höhe versehen. Der Balkonboden liegt i.d.R. auf dem Niveau eines Geschossbodens. Balkone werden im allgemeinen links- bzw. rechtsbündig, in Gebäudeseitenmitte oder über die gesamte Seite hin angeordnet.

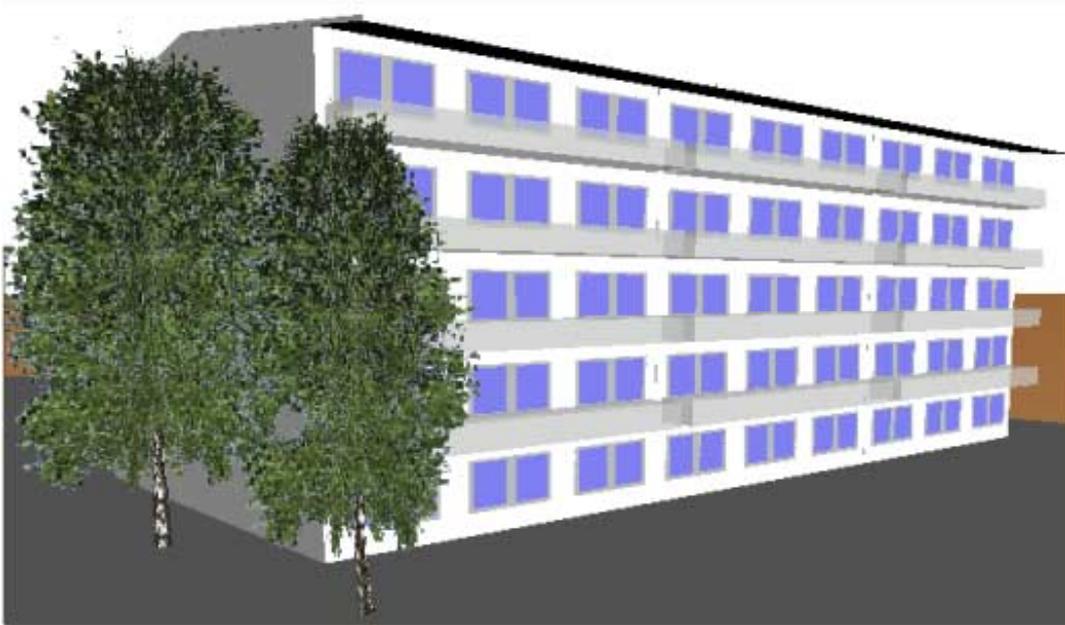


Abb. 71: Darstellung von Balkonen bei einem Mehrfamilienhaus

Die Anordnung der Balkone unterscheidet sich bei Mehr- und Einfamilienhäusern. In Mehrfamilienhäusern ist üblicherweise in jedem Geschoss ein Balkon zu finden, gelegentlich auch im Erdgeschoss. Die Balkone finden sich alle auf der sonnenzugewandten Traufeseite. In Abb. 71 ist das Beispiel eines typischen, mit Balkonen versehenen Mehrfamilienhauses im Salzburger Nonntal zu sehen.

Im Gegensatz dazu findet man bei Ein- und Zweifamilienhäusern Balkone eher auf der sonnenzugewandten Giebelseite. Auf diese Weise kann ein längerer Dachüberstand am Giebel für eine Überdachung genutzt werden. Balkone an den Traufeseiten sind bei Einfamilienhäusern zwar baulich möglich, in Wirklichkeit aber eher selten.

Balkone sind in der Regel an mehreren Stellen wenigstens teilweise durchsichtig. Um dieser Eigenschaft Rechnung zu tragen, wird für Balkone grundsätzlich eine Transparenz von 50% angenommen.

Mit den genannten Eigenschaften lassen sich die folgenden Kenngrößen für Balkone festlegen:

Länge, Breite, Höhe	Geometrie in [m]
Bündigkeit	links, rechts, zentriert, block
Häufigkeit	Ab EG, Ab 1. OG
Materialeigenschaft	Farbe, Transparenz 50%

Tabelle 9: Balkonattribute

5.2.3 Die geometrischen Kenngrößen des Daches

Die Dachform ist für die Modellierung eines Gebäudes besonders wichtig. Dies liegt vor allem an dem großen Volumen und an der Vielfältigkeit der Dachformen. Das Dach hat wesentlichen Anteil am Gesamtgebäude hat.

Die zur Beschreibung eines allgemeinen europäischen Daches notwendige Geometrie lässt sich aus den Ansichten von Dachlängs- und Querschnitt bestimmen, vgl. Abb. 71 zu sehen. Im Dachlängsschnitt ist die Dachform ein Dreieck, Rechteck, Trapez, Fünfeck bzw. Sechseck, deren Grundseite der Traufekante entspricht. Unterschiede sind in der Firstlänge zu finden. Beim Giebeldach entspricht die Firstlänge der Dachlänge. Beim Zeltdach hat der First die Länge null.

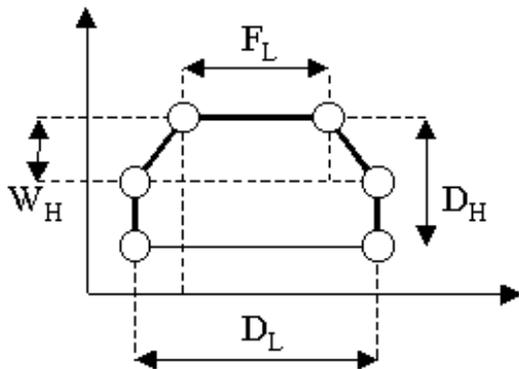
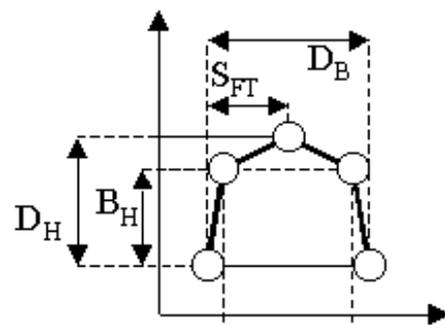


Abb. 71: Parameter im Dachlängsschnitt: Dachlänge D_L - Firstlänge F_L - Dachhöhe D_H - Walmhöhe W_H



Parameter Dachquerschnitt: Dachhöhe D_H - Dachbruchweite B_H - Dachbreite D_B - Abstand First Traufe S_{FT}

Will man ein Walmdach oder ein Krüppelwalmdach modellieren, kann die Firstlänge nicht aus anderen geometrischen Größen hergeleitet werden, sie muss bekannt sein oder sich aus voreingestellten Werten bestimmen lassen. Für das Krüppelwalmdach muss zudem die Lage des Walmbruches ermittelt werden. Abb. 72 zeigt die Punktebezeichnungen für die Punkte eines Daches.

Für den Dachquerschnitt kommen drei Grundformen in Frage. Neben dem fünfeckigen Mansarddach sind sowohl das Pultdach als auch das Satteldach dreieckige Formen, beim Pultdach mit seitlich verschobener Dreiecksspitze. Hier kommt der seitlichen Lage des Firstes eine besondere Bedeutung zu.

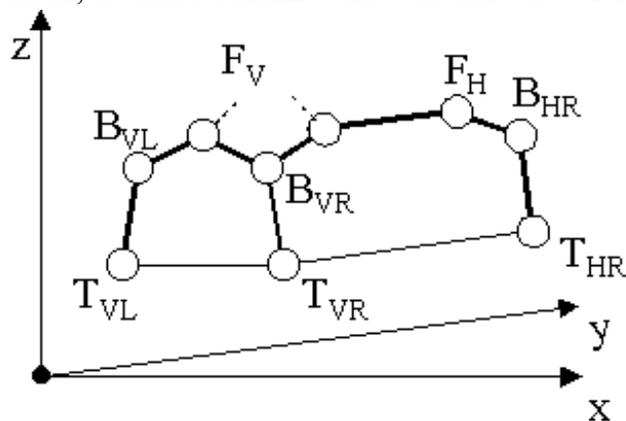


Abb. 72: Punkte an einem Dach

So lassen sich alle Einflussgrößen anhand der komplexesten Dachform Mansarde-Krüppelwalmdach darlegen. Tabelle 10 zeigt, welche Kenngrößen bekannt sein müssen, um die unterschiedlichen Dachtypen realisieren zu können.

Diese Kenngrößen lassen sich in Koordinaten und topologische Beziehungen übersetzen. Es entsteht ein topologisches Modell. Das dazu verwendete Koordinatensystem hat seinen Ursprung im Punkt T_{VL} , die X-Achse wird nach T_{VR} ausgerichtet, die Y-Achse nach T_{HL} , die Z-Achse senkrecht zu diesen beiden.

An den Stellen, an denen die aus den Bestimmungsgrößen gebildeten Linien, die Kanten, zusammenlaufen, werden Knoten gebildet. Daraus entstehen Regeln für die Koordinatenbildung der Komplexform des Daches (Tabelle 11). Alle Koordinaten verstehen sich als Koordinaten eines lokalen Bezugssystems mit einer unteren linken Hausecke als Ursprung (wie beim Mauerwerk).

Kenngröße	Dachform	Pultdach	Flachdach	Satteldach	Mansarddach	Zeltdach	Zelt-Mansarddach	Pult-Walmdach	Walmdach	Mansarde-Walmdach	Pult-Krüppelwalmdach	Krüppelwalmdach	Mansarde-Krüppelwalmdach
Grundriss		X	X	X	X	X	X	X	X	X	X	X	X
Dachhöhe		X		X	X	X	X	X	X	X	X	X	X
Dachbruchweite					X		X			X			X
Höhe Dachbruch					X		X			X			X
Dachbreite		X	X	X	X	X	X	X	X	X	X	X	X
Abstand First Traufe		X		X	X	X	X	X	X	X	X	X	X
Walmhöhe											X	X	X
Dachlänge		X	X	X	X	X	X	X	X	X	X	X	X
Firstlänge		= D_L		= D_L	= D_L	0	0	X	X	X	X	X	X

Tabelle 10: Notwendige Bestimmungsgrößen für verschiedene Dachformen

Lfd. Nr.	Pkt. Bez.	X	Y	Z
0	T _{VL}	0	0	0
1	B _{VL}	$\frac{1}{2} (D_B - D_{bW})$	$\frac{1}{2} (D_B - D_{bW})$	B _H
2	F _V	S _{FT}	$\frac{1}{2} (D_L - F_L)$	D _H
3	B _{VR}	$D_B - \frac{1}{2} (D_B - D_{bW})$	D _L	B _H
4	T _{VR}	D _B	$\frac{1}{2} (D_B - D_{bW})$	0
5	T _{HL}	0	D _L	0
6	B _{HL}	$\frac{1}{2} (D_B - D_{bW})$	$D_L - \frac{1}{2} (D_B - D_{bW})$	B _H
7	F _H	S _{FT}	$D_L - \frac{1}{2} (D_L - F_L)$	D _H
8	B _{HR}	$D_B - \frac{1}{2} (D_B - D_{bW})$	$D_L - \frac{1}{2} (D_B - D_{bW})$	B _H
9	T _{HR}	D _B	D _L	0

Tabelle 11: Koordinatentabelle für die Allgemeinform des Daches

Aus dem Koordinatensatz (Tabelle 11) sind die Dachflächen zu bilden. Verwendet man die von VRML [ISO/IEC14772-1:1997] für den Node IndexedFaceSet bekannte Notation: {lfd. Nr., lfd. Nr., ... , -1} (-1 für schließen), werden für die verschiedenen Dachflächen folgende Polygone gebildet:

Dachfläche	Polygon
Hauptdach	{3, 2, 7, 8, -1} und {1, 2, 7, 6, -1}
Walm (auch Krüppelwalm)	{1, 2, 3, -1} und {6, 7, 8, -1}
Mansarde	{0, 1, 6, 5, -1} und {4, 3, 8, 9, -1}
Mansarde-Walm	{0, 1, 3, 4, -1} und {5, 6, 8, 9, -1}

Tabelle 12: Polygonlisten für Dachflächen

Aus diesem Modell für die komplexe Dachform Mansarde-Krüppelwalmdach werden die anderen Formen abgeleitet. Alle anderen Dächer sind einfacher gestaltet. Dazu werden einzelne Komponenten, die bei diesen Formen keine Bedeutung haben, nicht verwendet.

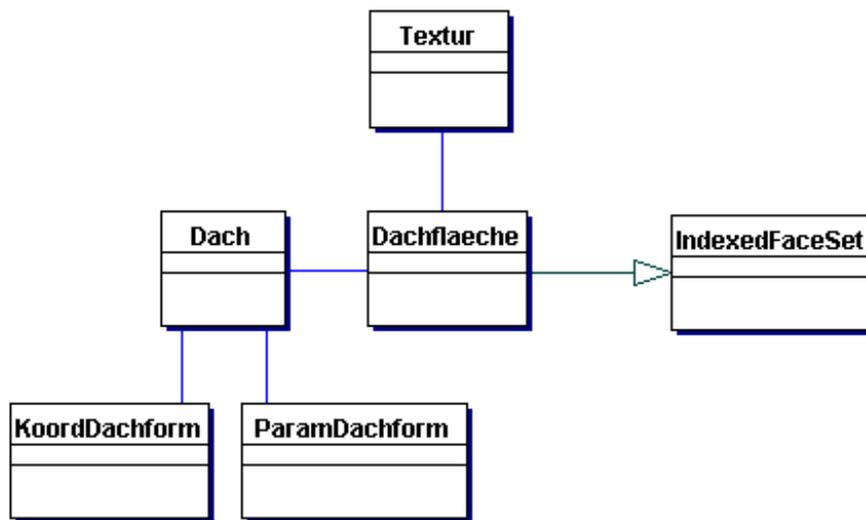
Festzustellen ist also, dass mit der Bildung der geometrischen Bestimmungsgrößen der Dachform Mansarde-Krüppelwalmdach bei geschickter Koordination alle anderen wesentlichen Dachformen abgedeckt werden. Dies bedeutet für die einzelne Dachform (Tabelle 13):

Dachform	Modifikationen	Projektionen, Punktbezeichnungen nach Abb. 72
Pultdach	Verschiebung der Firstlinie aus der Mitte zu einer Traufe hin	B_{VL}, B_{VR} auf B_{HL}, B_{HR} ; T_{VL}, T_{VR} auf T_{HL}, T_{HR}
Flachdach	Nur Verwendung von $T_{VL}, T_{VR}, T_{HR}, T_{HL}$ zu einer Fläche	$B_{VL}, B_{VR}, B_{HR}, B_{HL}$ auf $T_{VL}, T_{VR}, T_{HR}, T_{HL}$; $D_H = 0$
Satteldach	Nicht verwendet: Mansarde, Mansarde-Walm, Walm	$B_{VL}, B_{VR}, B_{HR}, B_{HL}$ auf $T_{VL}, T_{VR}, T_{HR}, T_{HL}$
Mansardedach	Nicht verwendet: Walm, Mansarde-Walm	
Zeltdach	Nicht verwendet: Mansarde, Mansarde-Walm, Walm	$B_{VL}, B_{VR}, B_{HR}, B_{HL}$ auf $T_{VL}, T_{VR}, T_{HR}, T_{HL}$; $F_H = F_V$
Zelt-Mansardedach		$F_H = F_V$
Pult-Walmdach	Nicht verwendet: Mansarde, Mansarde-Walm	$B_{VL}, B_{VR}, B_{HR}, B_{HL}$ auf $T_{VL}, T_{VR}, T_{HR}, T_{HL}$; T_{VL}, T_{VR} auf $\frac{1}{2}T_{HL}, \frac{1}{2}T_{HR}$
Walmdach	Nicht verwendet: Mansarde, Mansarde-Walm	$B_{VL}, B_{VR}, B_{HR}, B_{HL}$ auf $T_{VL}, T_{VR}, T_{HR}, T_{HL}$
Pult-Krüppelwalmdach	Nicht verwendet: Mansarde, Mansarde-Walm	B_{VL}, B_{HL} auf $B_{VL}(x) = F_V(x),$ $B_{HL}(x) = F_H(x)$
Krüppelwalmdach	Nicht verwendet: Mansarde, Mansarde-Walm	
Mansarde-Krüppelwalmdach	Nicht verwendet: Mansarde-Walm	

Tabelle 13: Modifikationen für einfache Dachformen

In der Struktur bestehen alle Flächen aus IndexedFaceSets. Das Objekt hat für die Definition der unterschiedlichen Bestimmungsgrößen zum einen die Koordinatenvariante als auch die Parametervariante zur Verfügung. Die Koordinatenform ist in die Parameterform konvertierbar und umgekehrt. Bei Verwendung der Parameterform wird diese dann in die Koordinatenform überführt.

Aus diesen Überlegungen ergibt sich die Modellierung in UML (Modell 9) für die Dachform:

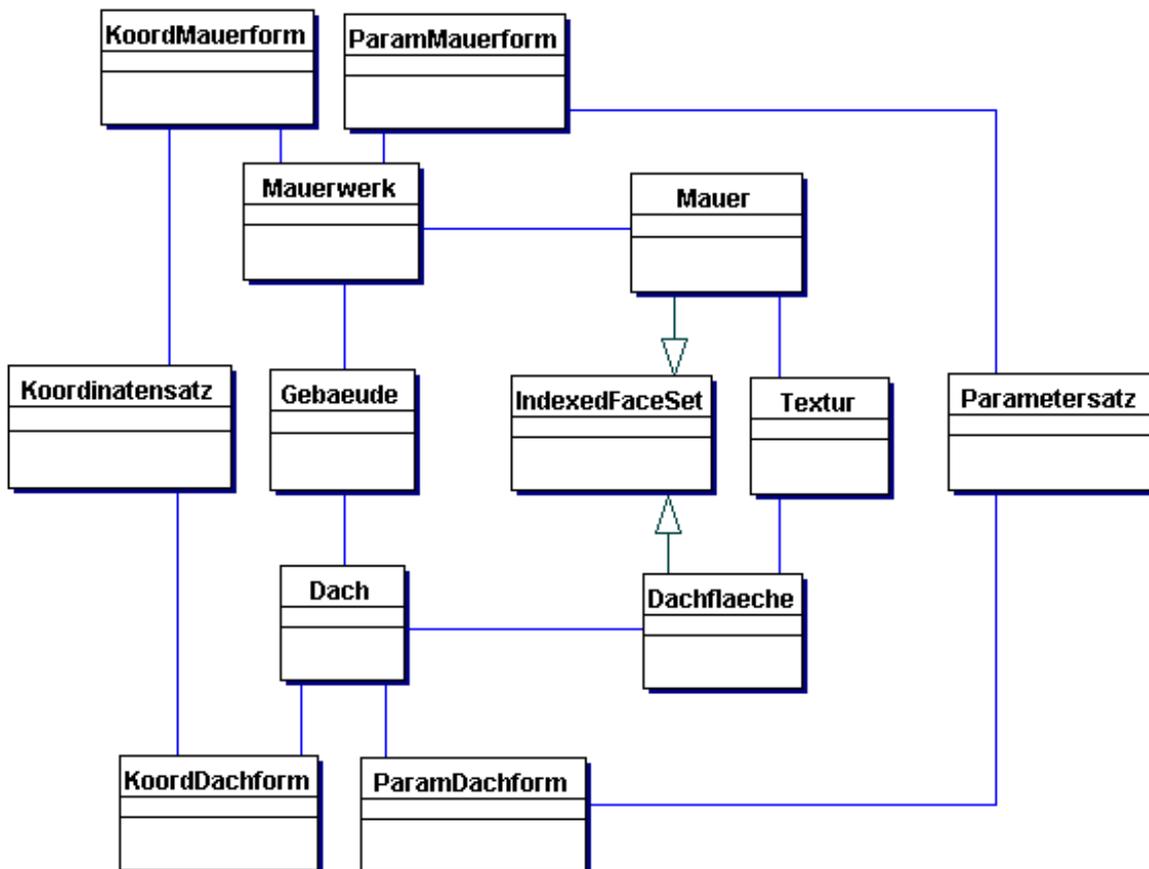


Modell 9: Struktur der Dachform

Das Dach besteht aus einer Geometrieinformation, die durch den Koordinaten- oder Parametersatz dargestellt werden kann. In den Dachflächen, ebenfalls Bestandteil des Daches, befindet sich durch Vererbung die Topologie sowie die Textur.

5.2.4 Texturierung der Gebäudemodelle

Eine Erweiterung des Modells ist die Verwendung von Texturen anstelle der Farben. Wesentli-



Modell 10: Modell für ein texturiertes Gebäude in Wandmodelltechnik

ches Merkmal sind die Textur der Fassade und die Textur des Daches. Diese können aus Regeln, Plänen und Satzungen gewonnen werden, z.B. Backsteinfassade oder Dachschindeln aus dem Bebauungsplan bzw. Gemeindefestsetzung.

Die verwendete Modellierungstechnik benutzt für das Dach und die Fassade, die Kachelungstechnik mit Maßstabsvergleich von Modellgröße zu Originalgröße.

Die Textur für Gebäudeobjekte ist von üblicherweise rechtwinkligen Strukturen, die Einzelobjekte oder bereits Aggregate sein können, gekennzeichnet. Einzelobjekte sind große Einzeloberflächenelemente, wie beispielsweise Fenster oder Türen. Die Aggregate sind demgegenüber Fassadenstrukturen, z.B. Backsteinfassade, oder Dachstrukturen, z.B. Dachziegel.

In Modell 10 wird ein Gesamtgebäudemodell gezeigt. Deutlich wird, wie sich die Textur in das Gesamtbild eingliedert. Jede Fläche, ob Mauerwerk oder Dachfläche hat ihre eigene Textur und ihre eigene festgelegte Topologie und ist damit unabhängig vom Rest des Gebäudes.



Abb. 73: Extraktion von Einzelobjekttexturmustern

Die Texturen werden gewonnen, indem sie manuell aus vorhandenen Fotografien extrahiert werden. Die Fotografien können, qualitativ ausreichend, mit Amateur-Kleinbildkameras erstellt werden. Bei der Fotografie ist das Zielobjekt im Focus zu halten. Der Fotograf sollte möglichst orthogonal zur Objektebene stehen, damit ein möglichst gleichmäßiges Bild entsteht. Der Objektanteil an der Aufnahme darf nicht die gesamte Fotografie abdecken, um Randverzeichnungen zu vermeiden.

Zur Extraktion wird das Bild (Abb. 73 links) in ein Objektkoordinatensystem transformiert, bei dem das Objekt links unten die Koordinaten 0,0 erhält und die Koordinaten der rechten oberen Ecke Breite und Länge des Objektes in der Naturentsprechen. Analog werden die beiden anderen Objektecken behandelt. Nach Durchführung der Transformation ist die Textur rektifiziert. Das Texturelement (Abb. 73 rechts) kann mit einem Rechtwinkelausschnitt entnommen werden.

5.2.4.1 Fassadentexturen

Ein mit den beschriebenen Bestandteilen modelliertes Wohngebäude erzeugt einen recht ansehnlichen Eindruck, wirkt aber eher wie eine Scheune. Es fehlen zusätzliche Details, z.B. Fenster (Abb. 74 im Vergleich zur Abb. 61), welche für den Menschen, auch im Unterbewusstsein, die Assoziation zum Wohnhaus unterstützen.

Bei der Erzeugung der Textur einer Fassade mit Fenstern wird ein Trick angewandt. Zunächst wird die Anzahl der Fenster aus einer mittleren angenommenen Fensterdichte abgeleitet. Die Fensterzahl stimmt üblicherweise nicht mit der Anzahl an Texturkacheln

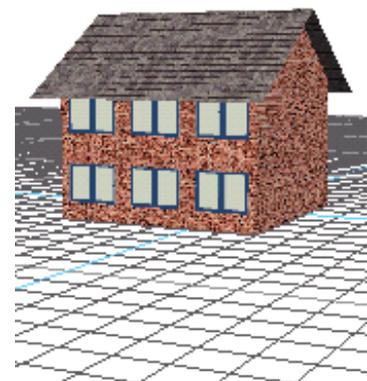


Abb. 74: Texturiertes Gebäude mit Fenstern

für das Mauerwerk überein. Es empfiehlt sich daher, Fenstertextur und Fassadentextur voneinander zu trennen. Eine Objektfläche kann aber nur mit einer Textur belegt werden.

Zur Lösung dieses Modellierungsproblems wird direkt vor die eigentliche Fassade eine zweite, identische Fassade gestellt. Auf diese wird dann die Fenstertextur gelegt. Die Fenstertextur besteht aus einem Fensterteil und ist im übrigen transparent, so dass die Fassadentextur dahinter zum Vorschein kommt.

Mit dieser Vorgehensweise erhält man zwar eine ansprechende Visualisierung eines Gebäudes, aber der Aufwand sowohl an zusätzlicher Geometrie als auch an Textur erhöht sich bei einem vierseitigen Gebäude erheblich. Dies liegt daran, dass einer Oberfläche nur genau eine Textur zugewiesen werden kann und je Fenster eine zusätzliche Fläche entsteht.

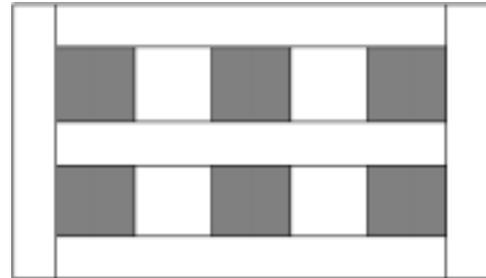


Abb. 75: Aufteilung einer Fassade in Teilflächen

Eine Möglichkeit, diesen Aufwand zu minimieren, ist die Aufteilung der Geometrie der Fassade in Teilflächen (Abb. 75). Die weißen Flächen erhalten z.B. eine Backsteintextur, die grauen Flächen werden mit Fenstertextur belegt (Abb. 76).

Während die Textur der Fenster ohne Wiederholung verwendet wird, ist die Backsteintextur entsprechend ihrer Größe zu wiederholen. Dies kann jedoch zu grafischen Problemen führen, da es an den Stoßkanten der Teilflächen gleicher Textur aufgrund von Rundungsproblemen zur Bildung von Bildfehlern kommen kann. Ursache dafür sind unterschiedliche

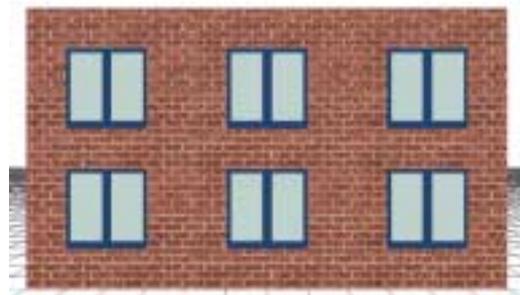


Abb. 76: Fassade mittels Teilflächentextur

Rechenverfahren im physikalischen Layer (Abb. 87).

Diese Fehler treten üblicherweise wie eine gepunktete Linie mit größerem Abstand zwischen den Punkten in Erscheinung. In den hier dargestellten, verkleinerten Abbildungen sind diese Fehler nicht zu erkennen. Über diesen Weg wird der Texturaufwand reduziert, während der geometrische Aufwand noch weiter erhöht wird.

Dieser Weg der Texturierung lässt sich aber automatisieren. Wird die in Abb. 76 dargestellte Fassade als Ganzes betrachtet, kann dieses Ganze mittels eines gleichmäßigen Rasters (Abb. 77) in Kacheln unterteilt werden. Die Rasterweiten orientieren sich dabei an der gewünschten Wiederholung der Kachel.

Zunächst ist eine Grundtextur, z.B. eine Backsteintextur, zu wählen. Diese wird mit Offset und Wiederholungen in Bezug auf die Breite und Länge der zu

texturierenden Fläche zugeordnet. Dadurch kann jede zur Textur gehörige Kachel aufgrund ihrer Lage als die (n,m) Wiederholung eindeutig identifiziert werden. Über dieses Muster können einzelne Kacheln oder Kachelgruppen angesprochen werden.

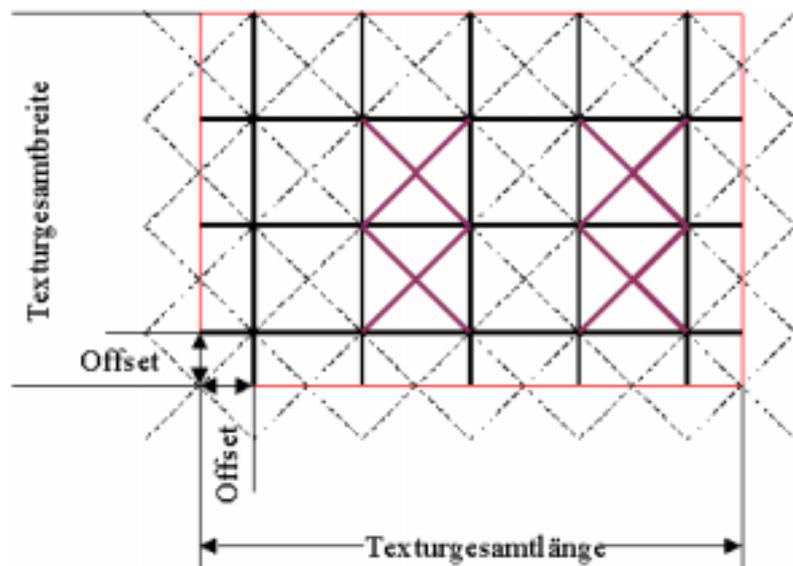


Abb. 77: Raster-Texturmodell

Bei der Zuweisung der Texturen zu den Kacheln können zwei unterschiedliche Wege verfolgt werden. Beim ersten Weg, dem Kachelaustausch, wird die Textur der ausgewählten Kacheln durch eine andere Textur ersetzt, z.B. durch eine Fenstertextur. Der Austausch erfordert eine ungefähre Übereinstimmung der Texturen mit ihrer natürlichen Größe. Deshalb ist es notwendig, bereits bei der Gestaltung der Grundtextur für eine geeignete Größe der Ursprungskachel zu sorgen. Diese kann für den Fall der Fassade z.B. ca. 40 x 50 cm² betragen. Diese Ausdehnung basiert auf im Baubereich üblichen Normgrößen. Fenster und Türen haben Normbreiten von 80 cm bzw. 120 cm, die Normhöhen sind 100 cm bzw. 200 cm. So kann durch Ersatz von 4 bzw. 8 oder 12 Kacheln die jeweilige Darstellung erzeugt werden.



Abb. 78: Textur mittels Integration

Beim zweiten Weg, der Texturintegration (Abb. 78), wird ein einziges Bild für die gesamte Oberfläche einer Fassadenseite verwendet. Zunächst entsteht eine Texturbasis aus entsprechenden Wiederholungen von Fassadentexturkacheln. Die den in die Fassade zu integrierenden Objekte werden in das Bild hineinkopiert. Dies gestattet eine genauere Platzierung der Objekte in der Fassade. Das entstandene Bild wird dann als eine einzige, vollständige Textur einer Wandseite zugeordnet.

Beide Verfahren sind praktikabel. Der Aufwand für das Kachelaustauschverfahren ist gering. Die notwendigen Daten werden bereits in Dateien vorgehalten. Beim Texturintegrationsverfahren ist zur Exemplarerzeugung ein erheblicher Rechenaufwand erforderlich. Dieses Verfahren ermöglicht aber eine präzise Positionierung der Objekte. Da es bei der Erzeugung der Gebäude aus Datenbankbeständen weniger auf Präzision der Details ankommt, als auf ein realistisches Erscheinungsbild, ist das Kachelaustauschverfahren vorzuziehen.

Bei beiden Verfahren ist die verfügbare Auflösung zu berücksichtigen. Sie liegt, je nach verwendeter Darstellungssoftware zwischen 256x256 Pixel und 1024x1024 Pixel. Daher kann es bei besonders langen Gebäuden gelegentlich zu Schwierigkeiten kommen. Bei einem Haus mit einer Länge von 100 m entspricht dies bei einer Texturgröße von 1024 einer Länge von ca. 10 cm/Pixel. Bei einer Texturgröße von 256 sind dies ca. 40 cm/Pixel. Diese Werte sind für eine realistische Darstellung zu hoch. Hier muss deshalb auf Kachelungstechnik in Kombination mit Texturintegration bzw. Kachelaustauschverfahren zurückgegriffen werden.

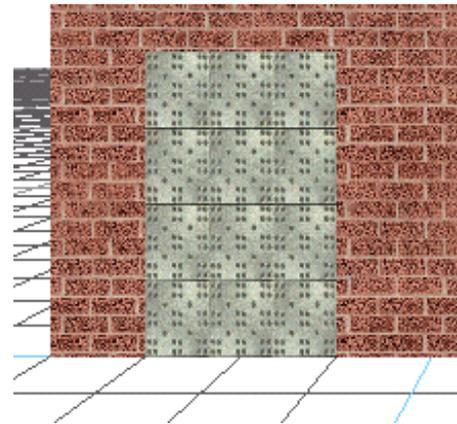


Abb. 79: Einzelobjekt Tür mittels zusätzlicher Objektweiser Aufbringung erzeugt

Ein weiteres Verfahren zur Modellierung von Fassaden ist die zusätzliche objektweise Aufbringung. Dabei werden die Objekte, beispielsweise Fenster oder Türen (Abb. 79), einzeln konfektioniert und vor die Fassade gestellt.

Zunächst werden die Objektgröße sowie die Textur und ggf. die Texturwiederholung festgelegt und ein Exemplar als texturiertes Rechteck erzeugt. Dieses wird dann dem Verteilungsmuster entsprechend am vorgesehenen Ort der Darstellung angebracht. Dabei werden die Objekte 1 cm oder weniger vor die Fassade gezogen, damit sie nicht durch die Fassadentextur überdeckt werden.

Um diese Objekte automatisiert aufbringen zu können, sind einige Formeln notwendig, mit dem die Verteilung bestimmt werden kann. Dabei wird davon ausgegangen, dass die Objekte rechteckig sind, und sich der Lagebezugspunkt an der linken unteren Ecke befindet.

Abb. 80 zeigt das Prinzip. Zunächst ist zu ermitteln, wie viele Fenster eines Stockwerks sich über die Gebäudeseite erstrecken. Hierfür ist die Anzahl der Räume hinter der Fassade eine wichtige Information.

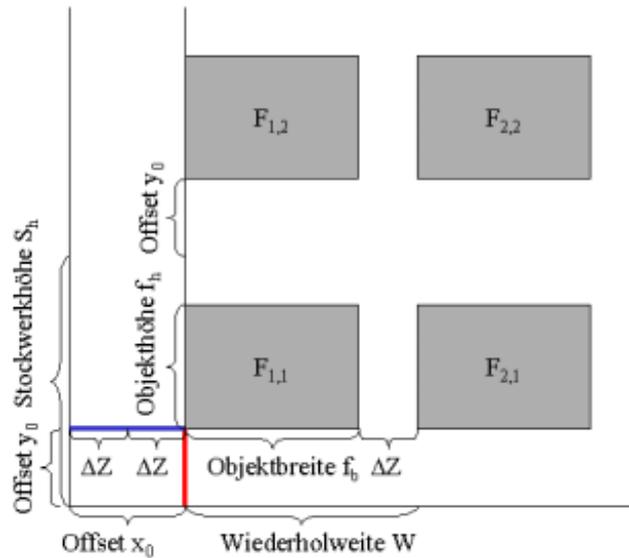


Abb. 80: Parameter für Verteilung von Fenstern

Üblicherweise hat jeder Raum zu jeder Außenseite auf 3m Raumbreite ein Doppelflügel Fenster. Die Raumbreiten betragen in der Regel zwischen 3m und 5m. Bei einer Gebäudebreite von 10m ergibt sich danach eine Fensterzahl zwischen 3,3 und 2, im Mittel also 2,65. Aufgerundet ist die Fensterzahl $n = 3$. Da nun bekannt ist, wie viele Fenster platziert werden müssen, kann die Verteilung ermittelt werden. Der Zwischenraum zwischen den Fenstern wird nach der Formel

$$\Delta Z = \frac{b - n \cdot f_b}{n + 2}$$

ΔZ – Zwischenraum b – Fassadenbreite f_b – Fensterbreite n - Fensterzahl

berechnet. Der zweifache Zwischenraum ist zunächst der Offset x_0 bis zum Beginn des ersten Fensters. Das nächste Fenster folgt in der Wiederholweite $W = f_b + \Delta Z$. Daraus ergeben sich die horizontalen Fensterpositionen eines Stockwerkes aus:

$$F_i(x) = x_0 + 2 \cdot \Delta Z + (i - 1) \cdot W \text{ mit } i = 1..n$$

$F_{i,j}(x, y)$ – Fenstereckpunkt links unten x_0, y_0 – Offset W – Wiederholweite
 i – Index der Fenster horizontal n – Anzahl der Fenster

Die Höhe der Fenster ergibt sich aus der gängigen Fensterbankhöhe von 0,70 m über dem Boden. Diese Höhe entspricht dem Offset y_0 . Die Wiederholung erfolgt stockwerkweise, so dass sich die Position in y ergibt als:

$$F_j(y) = y_0 + (j - 1) \cdot S_h \text{ mit } j = 1..m$$

S_h – Höhe Stockwerk j – Index der Fenster vertikal

Somit ergibt sich insgesamt für die Fensterposition:

$$F_{i,j}(x, y) = (x_0 + (i-1) \cdot W); (y_0 + (j-1) \cdot S_h) \text{ mit } i = 1 \dots n; j = 1 \dots m$$

m – Zahl der
Stockwerke

Die Position einer Tür in einer Fassadenseite lässt sich einfach durch Ersatz eines Fensters im Erdgeschoss ($n = 1$) durch eine Tür ermitteln, indem bei dieser Position der Offset y_0 nicht berücksichtigt wird.

5.2.4.2 Dachtexturen

Dachtexturen sind ähnlich zu betrachten wie Fassadentexturen. Hier finden sich regelmäßig wiederholende Texturmuster (Dachschindeln) in Kombination mit Einzeltexturen (Dachfenster, Solaranlagen, etc.).

Allerdings liegt eine Besonderheit vor. Texturen haben bei modernen Darstellungsverfahren (Backface Culling) ein Seitenverhalten. Das bedeutet, sie werden nur dann angezeigt, wenn sie von der Seite der Fläche betrachtet werden, wo die Reihung ihrer Punkte im Uhrzeigersinn erscheint.

Von der anderen Seite ist keine Textur zu sehen (Abb. 81). Im Mauerwerk ist dies bedeutungslos, da es nicht von innen betrachtet werden kann.

Bei Dachüberständen ist, von unten betrachtet, der Dachüberstand nicht sichtbar. Um dieses Manko zu umgehen, wird eine zweite, nahezu identische Fläche wie die erste, hinzugefügt. Sie unterscheidet sich gegenüber der ersten allein in der Reihenfolge der Punkte, welche von oben betrachtet gegen den Uhrzeigersinn erscheint. Damit entstehen eine Ansicht von oben und eine Ansicht von unten. Für die Ansicht von unten lässt sich so auch eine andere Textur verwenden, als für die aus Ziegel bestehende Dachoberseite, denn die Dachunterseite besteht üblicherweise aus Holz.

Um ein Gebäude jedoch wirklich realistisch erscheinen zu lassen, ist es notwendig eine weitere Eigenschaft zu etablieren. Ein Dach besteht neben der Dachfläche und der Dachunterseite aus Ortgang und Traufekante. Diese können gebildet werden, indem die zweite, unten liegende Dachfläche mit der Textur der Dachunterseite um die Stärke des Ortganges nach unten verlegt wird.

Verbindet man die zwei Dächer durch Ortgang-Flächen, so entsteht, wie Abb. 82 zu sehen, das Dach als Körper. Die Flächen werden mit dem Material der Dachunterseite texturiert.

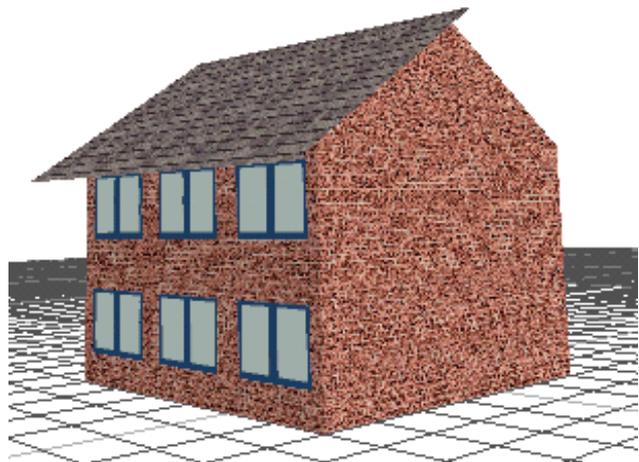


Abb. 81: Gebäude ohne Dachunterseitentextur (rechts)



Abb. 82: Dach mit Ortgang

5.3. Automatisierte Vereinfachung von Gebäudeobjekten

Ähnlich den bei den Baummodellen gefundenen unterschiedlichen Detaillierungsgraden sind auch bei den Gebäuden Detailstufen automatisch erzeugbar. Die Modelle können an den Ansatzpunkten Textur und Geometrie reduziert werden. Dabei wird die in Kap. 5.2 vorgestellte Modellierung als Detailmodell (Abb. 83) betrachtet und dient als Ausgangspunkt für in Detail veränderte Modelle.

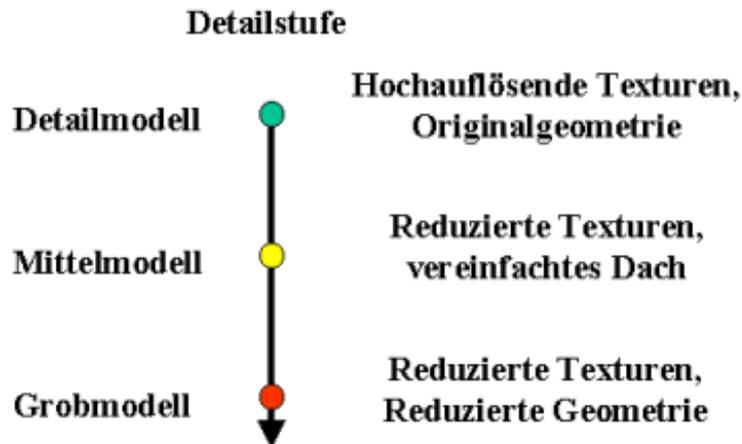


Abb. 83: Struktur des Detailstufenmodells für Gebäude

Für das Mittelmodell (Abb. 83) können auf der Texturebene die gleichen Texturen unter Reduktion der Pixelfläche durch Resampling weiterverwendet werden. Dieser Prozess kann voll automatisiert durchgeführt werden. Geometrisch wird das Dach um Ortgang und Unterdach reduziert.

Diese Vereinfachung verringert die Anzahl der darzustellenden Flächen erheblich, beispielsweise bei einem einfachen Giebeldach von 10 Polygonen auf 2 Polygone. Auf andere Dachformen bezogen, die aus mehr Teilflächen bestehen, ist die Reduktion noch stärker. Zudem kann bereits ab einer Entfernung von ca. 500 m auf Fenster verzichtet werden.

Ein Ansatzpunkt sind Kleingeometrien, wie Dachgauben und Dachmauern. Diese fallen abhängig von ihrer Größe als erste einfach weg. Ein weiterer Ansatzpunkt bietet sich beim Dachmodell. Hier sind bei den unterschiedlichen Dachformen verschieden aufwändige Geometrien zu finden. Die Grundform des Daches ist das Satteldach. Auf diese Form lassen sich die Dachformen mit Mansarddach leicht zurückführen. Dazu werden einfach die zur einzelnen Dachformen gehörigen geometrischen Größen reduziert und so das erforderliche Modell erzeugt. Walm- und Krüppelwalm-Variationen sind demgegenüber für eine Rückbildung weniger geeignet, da sie Einfluss auf die Farbgebung der Giebelseite haben.

Beim Grobmodell (Abb. 83) ab ca. 1000 m Entfernung, wird die Anzahl der Farben eines Textur-elementes reduziert, indem Bereiche ähnlicher Farbe zu einer Farbe zusammengefasst werden. Das Ergebnis bringt neben der Reduktion des Aufwandes bei der Darstellung eine Hervorhebung von Strukturen in größerer Entfernung mit sich. Die Darstellung der Fenster ist selbst noch in größerer Entfernung wichtig, da sie aufgrund ihres farblichen Unterschieds zur restlichen Fassade oft noch aus weiter Entfernung sichtbar sind.

Geometrisch wandelt sich das Gebäudeobjekt beim Grobmodell in einen einfachen Quader mit den Maßen des Grundrisses als Grundfläche und der Gebäudehöhe als Quaderhöhe.

5.4 Reihenhausezeile und Wohnblock

Die Reihenhausezeile entsteht durch eine Aneinanderreihung einzelner Gebäude, so dass die Dachfirste üblicherweise eine Linie bilden. Diese Regel wird allerdings recht häufig durch leichten Quer- und Höhenversatz gebrochen. Was jedoch auf die Modellierung keinen wesentlichen Einfluss hat. Ein Reihnhaus ist eine Aggregation von Häusern in linearer Folge und in dieser Eigenschaft den Bäumen einer Allee ähnlich.

Demgegenüber zeichnen sich Wohnblöcke durch eine Aneinanderreihung einzelner Gebäude zu einer eckigen, üblicherweise rechteckigen Grundform aus. Ein Wohnblock kann also auch als eine Aggregation von Reihenhausezeilen betrachtet werden. Um die rechteckige Grundform zu erzeugen werden die Hauszeilen entweder L-förmig oder rechtwinklig zueinander gebaut.

Insgesamt sind hier insbesondere Gebäudetypen erkennbar, die in Abb. 84 farbig markiert sind.

- das Reiheneckhaus (blau)
- das Reihen- und Wohnblockmittelhaus (gelb)
- das Wohnblockeckhaus (rot)

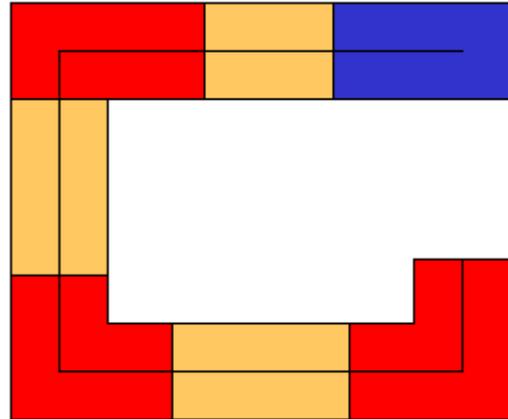


Abb. 84: Struktur eines Wohnblocks

Ein Reiheneckhaus (siehe Abb. 84, blaues Element) hat zwei Giebelseiten und zwar einen Giebel, der wie ein einzelstehendes Haus modelliert wird und einen Giebel, der einem Satteldach mit einem der drei Dachquerschnitte entspricht. Eine Giebelseite grenzt an die weiteren Häuser. Das Reiheneckhaus ist also nichts anderes als eine Sonderform des Einzelhauses.

Ein Mittelhaus (siehe Abb. 84, gelbe Elemente) hat ein Satteldach mit zwei nur wenig oder nicht sichtbaren Giebelseiten. So ist auch das Mittelhaus eine nur in seiner Attributierung beschränkte Sonderform des Einzelhauses.

Ein Eckhaus (siehe Abb. 84, rote Elemente) entspricht seiner Art nach einem Gebäude mit L-förmigem Grundriss. Damit kann auch das Eckhaus auf die Grundform des Einzelhauses zurückgeführt werden.

Für die Gestaltung von Wohnblöcken ist es notwendig, von der bisherigen Verfahrensweise der Zwillingsgiebel (jeder Giebelseite entspricht spiegelverkehrt der gegenüberliegenden) abzuweichen.

Während an den Stoßkanten der Objekte vorwiegend Giebeldachformen zu finden sind, kann man an den Wohnblockenden sowohl Giebel als auch Walm- und Krüppelwalmdachformen finden. Daher ist hier jede Giebelseite separat zu gestalten. Dieser Vorgang kann automatisiert werden. Während die freistehende Seite des Giebels in üblicher Form bestimmt wird, kann für die Stoßkante die Giebelform sowie ein Dachvorsprung der Länge 0 angenommen werden. Damit wird die Zahl der Kenngrößen lediglich um 1 erhöht.

Kapitel 6

OBJEKTmuster

In den Kapiteln 4 und 5 wurden die Teilobjekte modelliert, aus denen sich Vegetations- und Gebäudeobjekte zusammensetzen. Verschiedene Varianten wurden dabei dargestellt, die geeignet sind, Objekte genau zu beschreiben. In diesem Kapitel werden die gefundenen Elemente zusammengefasst und für die Strukturierung der Gesamtobjekte aufbereitet.

6.1 Die Allgemeinform des Baumes

Ein Baum wird als nahezu rotationssymmetrischer Körper um eine vertikale Achse betrachtet. Als Lageparameter reicht daher ein Koordinatentupel (x, y) aus. Die Darstellung wird durch nur wenige geometrische Größen bestimmt (Abb. 85), nämlich:

- Baumhöhe, unterteilt in
 - Stammhöhe und
 - Kronenhöhe

und die horizontalen Größen

- Stamm sowie
- Kronendurchmesser.

Baumhöhe und Kronendurchmesser sind die Maße für die äußere Umgebungslinie, den umgebender Quader (engl. bounding box).

Da ein komplexes Gebilde, wie ein Baum, hier mit nur mit wenigen Kenngrößen nachgebildet wird, muss zur Ausgestaltung von Details ein aufwändiges Texturmodell eingesetzt werden. Hierfür werden folgende Einflussgrößen verwendet:

- Baumtyp
- Baumalter
- Jahreszeit
- Windstärke

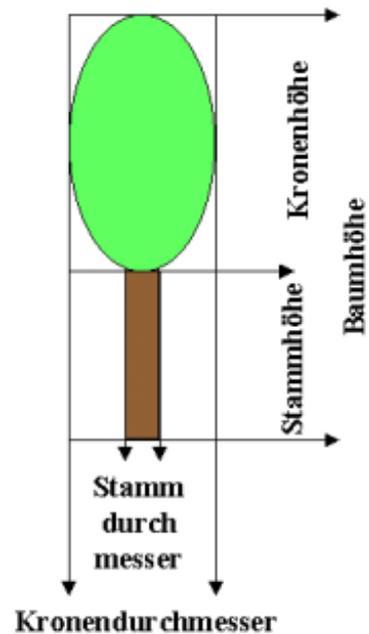


Abb. 85: Geometrische Größen eines Baumes

Diese Parameter bestimmen die Textur, die dem Objekt aus einem einem Texturverzeichnis, bzw. einer Texturdatenbank zugeordnet wird.

6.2 Die Allgemeinform für Gebäude

Das Objekt besteht aus einem Koordinatensatz, wie für das Dach und das Mauerwerk bereits beschrieben. Nun kommen einige Bedingungen zum Tragen, die Dach und Mauerwerk richtig verbinden.

- Alle Punkte eines Unterobjektes sind coplanar.
- Die Punkte T_{VR} , T_{HR} liegen in der Fläche F_V , F_H , B_{HR} , B_{VR} .
- Die Punkte T_{VL} , T_{HL} liegen in der Fläche Fläche F_V , F_H , B_{HL} , B_{VL} .

- Die Punkte F_V, G_V, G_H, F_H sind colinear
- Die Punkte $B_{VL}, M_{VL}, M_{HL}, B_{HL}$ sind colinear
- Die Punkte $B_{VR}, M_{VR}, M_{HR}, B_{HR}$ sind colinear
- Die folgenden Linien (a) sind parallel: $F_V - F_H \parallel B_{HL} - B_{VL} \parallel B_{HR} - B_{VR} \parallel Tr_{VR} - Tr_{HR} \parallel Tr_{VL} - Tr_{HL} \parallel Gr_{VR} - Gr_{HR} \parallel Gr_{VL} - Gr_{HL} \parallel M_{VR} - M_{HR} \parallel M_{VL} - M_{HL}$
- Die folgenden Linien (b) sind parallel: $Tr_{VL} - Tr_{VR} \parallel Tr_{HL} - Tr_{HR} \parallel B_{VL} - B_{VR} \parallel B_{HL} - B_{HR} \parallel Gr_{VL} - Gr_{VR} \parallel Gr_{HL} - Gr_{HR} \parallel M_{VR} - M_{VL} \parallel M_{VLR} - M_{HL}$
- Die Linien (a) sind senkrecht zu den Linien (b)

Mit diesen Bedingungen wird die Anzahl an Kenngrößen reduziert, indem Ersatzparameter eingeführt werden und die Koordinatenliste zu einer Gesamtliste (Abb. 86, Tabelle 14) zusammengeführt wird. Ersatzparameter sind die Dachüberstände (Dü), die vorne, hinten, rechts, links auftreten. Die weitere Größe $m(x, y)$ ist die Steigung der Gerade, die durch x und y verläuft.

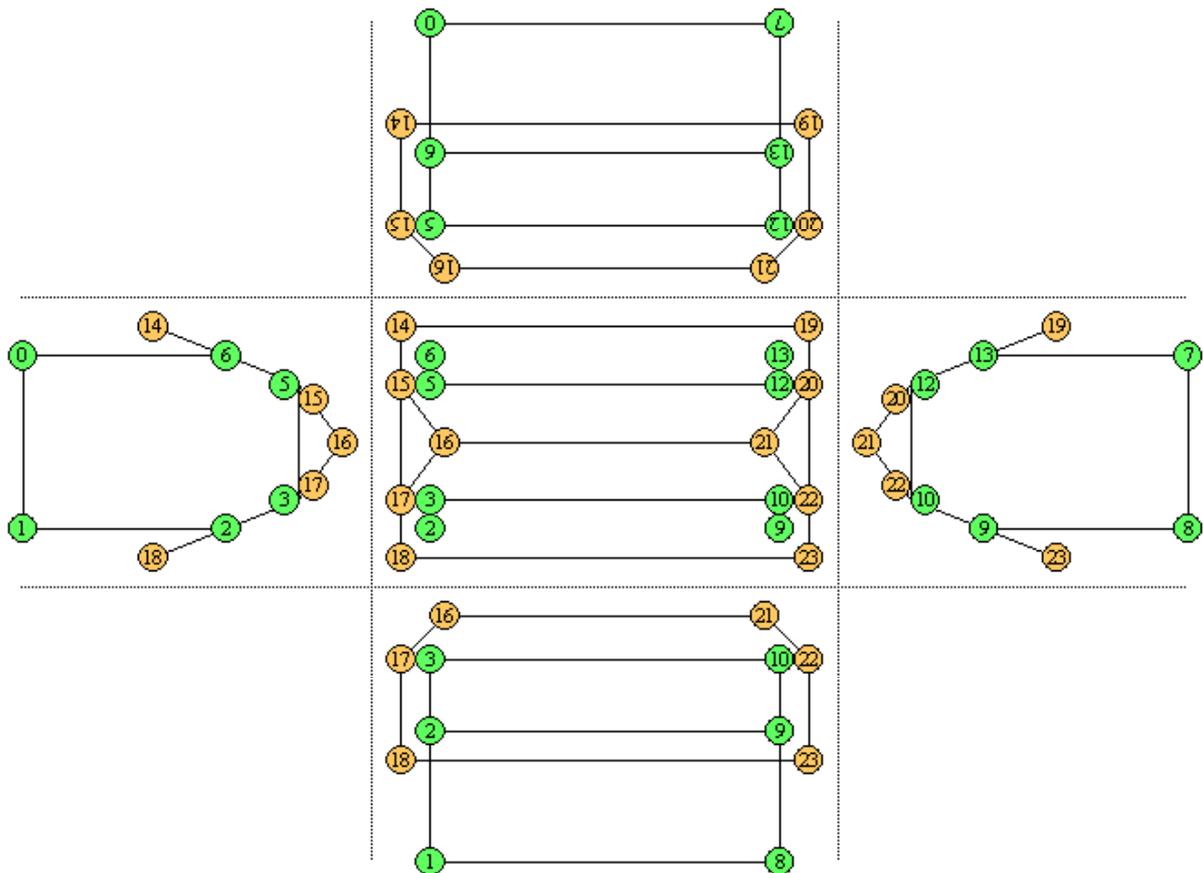


Abb. 86: Gemeinsame Punktstruktur von Mauerwerk und Dach

Eine weitere Hilfsgröße die verwendet wird, ist die Dachneigung. Diese wird für jede Dachhälfte getrennt berechnet. Grundlage sind die bereits bestimmten Punkte Tr_{VL}, M_{VL} bzw. Tr_{HL}, M_{HL} aus denen die Dachneigung α_L der einen Dachhälfte sowie Tr_{VR}, M_{VR} bzw. Tr_{HR}, M_{HR} aus denen die Dachneigung α_R der anderen Dachhälfte jeweils in der zx -Ebene berechnet wird. Mit Dachneigung und Dachüberstand kann über ein einfaches Anhängen der überstehende Dachfläche berechnet werden.

Lfd. Nr.	Pkt. Bez.	X	Y	Z
0	Gr _{VL}	0	0	0
1	Gr _{VR}	G _B	0	0
2	Tr _{VR}	G _B	0	T _H
3	M _{VR}	G _B + M _B	0	M _H
4	G _V	½ G _B	0	G _H
5	M _{VL}	G _B - M _B	0	M _H
6	Tr _{VL}	0	G _L	T _H
7	Gr _{HL}	0	G _L	0
8	Gr _{HR}	G _B	G _L	0
9	Tr _{HR}	G _B	G _L	T _H
10	M _{HR}	G _B + M _B	0	M _H
11	G _H	½ G _B	G _L	G _H
12	M _{HL}	G _B - M _B	0	M _H
13	Tr _{HL}	0	G _L	T _H
14	T _{VL}	Dü _L , m(Tr _{VL} , M _{VL})	- Dü _V	Dü _L , m(Tr _{VL} , M _{VL})
15	B _{VL}	G _B - M _B	- Dü _V	M _H
16	F _V	½ G _B	- Dü _V	G _H
17	B _{VR}	G _B + M _B	- Dü _V	M _H
18	T _{VR}	Dü _R , m(Tr _{VR} , M _{VR})	- Dü _V	Dü _R , m(Tr _{VR} , M _{VR})
19	T _{HL}	Dü _L , m(Tr _{HL} , M _{HL})	G _L + Dü _H	Dü _L , m(Tr _{HL} , M _{HL})
20	B _{HL}	G _B - M _B	G _L + Dü _H	M _H
21	F _H	½ G _B	G _L + Dü _H	G _H
22	B _{HR}	G _B + M _B	G _L + Dü _H	M _H
23	T _{HR}	Dü _R , m(Tr _{HR} , M _{HR})	G _L + Dü _H	Dü _R , m(Tr _{HR} , M _{HR})

Tabelle 14: Die Modellierung eines Gebäudes notwendigen Punkte

Folgende Kenngrößen werden benötigt. Aus diesen werden die notwendigen Punktkoordinaten berechnet:

- Gebäudebreite G_B zu Teilobjekt Grundrisspolygon G_P
- Gebäudelänge G_L zu Teilobjekt Grundrisspolygon G_P
- Dachhöhe D_H
- Traufhöhe $T = (T_{VR}, T_{HR}, T_{HL}, T_{VR})$

Weitere Parameter dienen vor allem der Verfeinerung des Dachmodells:

- Firstlänge F_L zu Teilobjekt Walm W
- Walmhöhe W_H zu Teilobjekt Walmobjekt W
- Dachbruchweite Db_W zu Teilobjekt Mansarde M
- Höhe Dachbruch B_H zu Teilobjekt Mansarde M
- Dachbreite D_B zu Teilobjekt Dachpolygon D_P
- Dachlänge D_L zu Teilobjekt Dachpolygon D_P
- Abstand First-Traufe S_{FT}

Kenngrößen stehen dabei in Beziehung zueinander:

- wenn die Dachbruchweite gegeben ist, muss auch die Höhe des Dachbruchs gegeben sein und umgekehrt

Dies ergibt die folgenden Konstruktoren:

Haus ($G_P (G_B, G_L), D_H, T, D_P (D_B, D_L), W(W_H, F_L), M(Db_W, B_H), S_{FT}$)

Haus ($G_P (G_B, G_L), D_H, T, D_{\bar{U}} (D_{\bar{U}V}, D_{\bar{U}H}, D_{\bar{U}R}, D_{\bar{U}L}), W(W_H, F_L), M(Db_W, B_H), S_{FT}$)

Werden alle Bezugsgrößen belegt, so entsteht ein Mansarde-Krüppelwalmdach, die weiteren Dachformen ergeben sich aus besonderen Parameterkonstellationen:

Pultdach:	$S_{FT} = 0, F_L = G_L, Db_W = 0, B_H = D_H$
Flachdach:	$D_H \leq 0$
Satteldach:	$F_L = G_L, Db_W = 0, B_H = D_H$
Mansardedach:	$F_L = G_L$
Zeltdach:	$F_L = 0, Db_W = 0, B_H = D_H$
Zeltmansardedach:	$F_L = 0, Db_W \neq 0, B_H < D_H$
Pultwalmdach:	$S_{FT} = 0, F_L < G_L, Db_W \neq 0, B_H = D_H$
Walmdach:	$W_H = D_H, F_L < G_L$
Mansarde-Walmdach:	$F_L < G_L, Db_W \neq 0, B_H < D_H$
Pult-Krüppelwalmdach:	$S_{FT} = 0, F_L < G_L, Db_W \neq 0, B_H < D_H$
Krüppelwalmdach:	$F_L < G_L, Db_W \neq 0, B_H < D_H$

Weiterhin sind einige Zwangsbedingungen durch definitionslose Zustände nicht zugelassen:

- $B_H > D_H$
- $Db_W > D_B$
- $W_H > D_H$
- $F_L > D_L$

Wird auf dem eigentlichen Dach ein zweites Dach verwendet, so ist dies möglich, indem das untere Dach als Mansarddach mit $B_H = D_H$ definiert wird und die Größen für das obere Dach wie üblich bestimmt werden und das Ergebnis oben aufgesetzt wird. Großflächige Bereiche dieses Modelltyps sind jedoch sehr selten.

Soll das Modell der Fassade höher detailliert werden, so sind weitere Kenngrößen notwendig:

- Sockelhöhe
- Geschosshöhe
- Fensterdichte

Für die Modellierung eines oder mehrerer Balkone sind die folgenden Größen notwendig:

- Länge, Breite, Höhe
- Bündigkeit
- Häufigkeit

Der Texturierung dienen die folgenden Parameter

- Material Fassade
- Material Fenster
- Material Dach
- Material des Unterdaches

Kapitel 7

UMSETZUNG DER ERGEBNISSE IN VERSCHIEDENEN SPRACHEN

Die bisher dargestellte Sicht auf die Objekte Vegetation und Gebäude war die Analysesicht. Auf sie folgt Designsicht. Dazu werden die Erkenntnisse in eine allgemeingültige, der Interpretation nahe Form gebracht. Der Standard Extensible Markup Language XML wird für diesen Schritt gewählt. XML erlaubt beliebige Elemente zu verwenden. In der Objektorientierung heißt das Äquivalent abstrakte Datentypen. Das Ergebnis dient auch dazu, Szenarien im XML-Format zu speichern.

Ein XML-Dokument enthält im wesentlichen Daten, deren Struktur eine Definition beschreibt. Außerdem kann das Quelldokument eine sogenannte Processing Instruction (PI = Verarbeitungsanweisung) enthalten, die festlegt, welche Teile des Dokuments wie darzustellen sind und welche nicht.

Mit dem Code `<?xml version="1.0" encoding="ISO-8859-1"?>` ist beispielsweise in der ersten Zeile die PI zu finden, die am Anfang des Dokuments besagt, dass es sich hier um XML in der Version 1.0 handelt und sich der Zeichensatz an den ISO-Richtlinien 8859-1 orientiert. Die Buchstabenfolge xml ist in der Spezifikation als Schlüsselwort festgelegt worden.

7.1 Implementierung zur Speicherung

Eine Document Type Definition (DTD) beschreibt den strukturellen Aufbau und die logischen Elemente einer bestimmten Klasse von XML-Dokumenten. Sobald man mehrere gleichartige Texte schreibt, wie zum Beispiel eine Reihe von E-Mails, dann dient eine DTD als Mustervorlage. Ein Editor kann dieses Muster schon beim Schreiben mit dem aktuellen Dokument vergleichen und auf strukturelle Fehler hinweisen. Dies könnte etwa ein fehlender Eintrag für einen Empfänger sein. Wenn keine DTD benutzt wird, kann von keinem Programm erwartet werden, dass es erkennt, aus welchen Bestandteilen ein Dokument besteht, es sei denn, es wäre für eine ganz bestimmte Anwendung proprietär und damit explizit implementiert.

Wenn sich ein Dokument an die Regeln der XML Spezifikation hält, so nennt man es wohlgeformt. Neben der Wohlgeformtheit gibt es mit der Gültigkeit eine zweite Eigenschaft, die ein Dokument auszeichnen kann. Damit ein Dokument auch das Gütesiegel gültig erhält, müssen die Inhalte der XML-Datei den vorgegebenen Strukturen der DTD entsprechen. Dazu muss in der Definition unbedingt eine DTD angegeben werden, und zwar in der Dokumenttyp-Deklaration. Sie kann entweder extern oder intern vorhanden sein [[W3](#)].

Der erste Schritt auf dem Weg zur DTD besteht in der genauen Analyse der zu beschreibenden Dokumentklasse. Diese Beschreibung ähnelt der objektorientierten Analyse und dem Design in der objektorientierten Programmierung (OOP). Als Einstieg soll die Frage dienen, was eine DTD und der Entwurf einer solchen eigentlich sind. Die DTD beschreibt eine Klasse von gleichartigen Dokumenten. Demnach besteht der erste Ansatz beim DTD-Entwurf darin, eine objektbezogene Beschreibung zu finden.

Bei den hier vorgestellten Modellierungen wird jeweils auf die Darstellung des Informationskopfes (Header) und des Wurzelementes (Root) verzichtet, da dort keine für die Modellierung wichtige Information vorzufinden ist. Im Mittelpunkt steht die jeweilige Objektart, deren Elemente vollständig dargestellt werden.

```

<!ELEMENT Tree (Position,
    TreeDimension, Age, TreeType)>
<!ELEMENT TreeType (#PCDATA)>
<!ELEMENT Position (Coordinate,
    Angle?)>
<!ELEMENT Coordinate (X, Y)>
<!ELEMENT TreeDimension (Year |
    TreeGeometry)>
<!ELEMENT TreeGeometry (
    Diameter, Height)>
<!ELEMENT X (#PCDATA)>
<!ELEMENT Y (#PCDATA)>
<!ELEMENT Diameter (#PCDATA)>
<!ELEMENT Height (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT Angle (#PCDATA)>
    
```

Tabelle 15: Einfache Implementierung des Baummodells als Ausschnitt aus einer DTD für Objekte in 3D-Stadtmodellen

TreeDimension stellt die geometrischen Informationen dar. Dafür stehen zwei Möglichkeiten zur Verfügung. Ein Weg verläuft über die explizite Angabe von Baumdurchmesser und Baumhöhe. Der andere Weg ist die Angabe des Baumalters. Ist für diese Baumart das Wachstumsmodell bekannt, so können aus dieser Information Höhe und Durchmesser abgeleitet werden.

Umgekehrt kann von Höhe und Durchmesser mit Hilfe des Wachstumsmodells auf das Alter

```

<!ELEMENT Tree (Position,
    TreeDimension, TreeType)>
<!ELEMENT TreeType EMPTY>
<!ATTLIST TreeType
    isType (Birch | Cherry | Oak |
    Poplar | Spruce | ... )
    #REQUIRED>
<!ELEMENT Position EMPTY>
<!ATTLIST Position
    X CDATA #REQUIRED
    Y CDATA #REQUIRED
    Angle CDATA >
<!ELEMENT TreeDimension (Age |
    TreeGeometry)>
<!ELEMENT TreeGeometry EMPTY>
<!ATTLIST TreeGeometry
    Diameter CDATA #REQUIRED
    Height CDATA #REQUIRED >
<!ELEMENT Age EMPTY>
<!ATTLIST Age
    Year CDATA #REQUIRED >
    
```

Tabelle 16: Optimierte Implementierung des Modells eines Baumes als Ausschnitt aus einer DTD für Objekte in 3D-Stadtmodellen

Der Einbau dieser Möglichkeiten geschieht in einem zweiten Modellierungsschritt.

In Tabelle 16 wird die Modellierung auf Attributlisten umgestellt. Damit wird die Attributierung aus einer Auswahl zugelassener Werte möglich, wie dies für das Element *TreeType* der Fall ist. Es macht keinen Sinn einen Baumtyp zu wählen, der weder in Wachstumsmodell noch

In der ersten Modellierungsstufe werden die einzelnen Kenngrößen identifiziert und benannt. In dieser Phase wurde das Hauptaugenmerk auf die Strukturierung und Gruppierung der Daten gelegt. So ist das Objekt *Tree* (Tabelle 15) in die Unterelemente *Position*, *TreeType* und *TreeDimension* unterteilt. Damit werden die Elemente Lage, Geometrie und Baumart bestimmt.

Das Element *Position* beinhaltet zwei Lagekoordinate sowie fakultativ eine Richtung. Eine Richtungsangabe ist nicht zwingend vorgeschrieben, da es sich bei einem Baum um ein annähernd rotationssymmetrisches Objekt handelt.

TreeType steht für die Baumart. Damit werden Textur und Wachstumsmodell bestimmt.

des Baumes geschlossen werden. Das Alter ist definiert über das Jahr, in dem der Baum gesetzt wurde. Damit kann die Darstellung dem Wachstumsfortschritt angepasst werden. So wird die Darstellung zu einem bestimmten Zeitpunkt und damit der Schritt zur Simulation ermöglicht. Diese Form hat außerdem den Vorteil, dass für eine aktuelle Darstellung eines Baumes nur die Existenz an sich geprüft werden muss, nicht aber die Geometrie.

Betrachtet man die obige Modellierung genau, ist zu bemerken, dass hier sehr viele Gestaltungsmöglichkeiten bleiben, wo eigentlich Restriktionen notwendig wären. Das Format ist noch nicht konkret genug. Restriktionen können in XML am besten durch Attributlisten bestimmt werden. Eine Attributlisten-Deklaration legt fest, welche Attribute für einen Elementtyp existieren und von welchem Typ die Attributwerte sind. Gegebenenfalls enthält sie einen Vorgabe-Wert.

in Textur zur Verfügung steht. Deshalb wird das Attribut *isType* nach und nach mit mehr Möglichkeiten gefüllt.

```

<!ELEMENT House (Position,
  Wall, Roof, Floors?,
  Facade?)>
<!ELEMENT Position EMPTY>
<!ATTLIST Position
  X CDATA #REQUIRED
  Y CDATA #REQUIRED
  Angle CDATA >
<!ELEMENT Wall
  (HouseDimensions,
  WallAppearance)>
<!ELEMENT HouseDimensions
  EMPTY>
<!ATTLIST HouseDimensions
  Height CDATA #REQUIRED
  Length CDATA #REQUIRED
  Width CDATA #REQUIRED>
<!ELEMENT WallAppearance
  (Texture? | Color?)>
<!ELEMENT Texture EMPTY>
<!ATTLIST Texture
  File NMTOKENS #REQUIRED>
<!ELEMENT Color EMPTY>
<!ATTLIST Color
  R CDATA #REQUIRED
  G CDATA #REQUIRED
  B CDATA #REQUIRED>
<!ELEMENT Roof (Overshot,
  Mansarde?, Walm?,
  RoofAppearance,
  SubRoofAppearance)>
<!ATTLIST Roof
  RidgeLength CDATA #REQUIRED
  Roofthickness CDATA "0.20"
  Shed CDATA "0.0">
<!ELEMENT RoofAppearance
  (Texture | Color)>
<!ELEMENT SubRoofAppearance
  (Texture | Color)>
<!ELEMENT Overshot EMPTY>
<!ATTLIST Overshot
  OVFront CDATA "0.0"
  OVRear CDATA "0.0"
  OVLeft CDATA "0.0"
  OVRight CDATA "0.0">
<!ELEMENT Mansarde EMPTY>
<!ATTLIST Mansarde
  MLength CDATA "0.0"
  MWidth CDATA "0.0"
  MHeight CDATA "0.0">
<!ELEMENT Walm EMPTY>
<!ATTLIST Walm
  RoofBreak CDATA "0.0">
<!ELEMENT Floors EMPTY>
<!ATTLIST Floors
  Socket CDATA "0.0"
  GrondFloor CDATA "2.5"
  GeneralFloor CDATA "2.5">
<!ELEMENT Façade
  (WindowAppearance?,
  Balcon?)>
<!ATTLIST Façade
  WindowHeight CDATA #REQUIRED
  WindowWidth CDATA #REQUIRED
  WindowDistanceBetween CDATA
  #REQUIRED
  WindowHeightOffset CDATA
  "0.7">
<!ELEMENT WindowAppearance
  (Texture | Color)>
<!ELEMENT Balcony
  (BalconyAppearance,
  Alignment)>
<!ATTLIST Balcony
  BLength CDATA
  BDepth CDATA
  BHeight CDATA
  Frequency CDATA>
<!ELEMENT BalconyAppearance
  (Texture | Color)>
<!ELEMENT Alignment EMPTY>
<!ATTLIST Alignment
  align ( LEFT | RIGHT |
  CENTER | BLOCK ) >

```

Tabelle 17: Implementierung des Modells eines Hauses als Ausschnitt aus der DTD 3D-Stadtmodell

Weiterhin ist es notwendig, dass zumindest ein Teil der Attribute respektive der Elemente mit Werten belegt sind. Dazu zählt *Position*, *TreeType* und entweder *TreeGeometry* oder *Age*.

In *TreeDimension* liegt die Wahl der Unterelemente entweder auf der geometrischen Repräsentation *TreeGeometry* oder der zeitlichen Repräsentation *Age*. Eines von beiden muss jedoch gewählt sein und wenn es gewählt ist, müssen die Felder der jeweiligen Attributliste belegt sein.

Eine weitere Variante der Zwangsbelegung ist die Vorbelegung mit Werten. Diese macht in Geometrie und Alter kaum Sinn, beim Baumtyp schon eher. Zunächst ist festzustellen, welches die Baumart ist, die am häufigsten vorkommt, und diese Größe ist dann mit dem zugehörigen Wert vorzubelegen. Dieser Schritt lässt sich jedoch besser im Zuge der Implementierung realisieren, wenn zur Exemplarbildung ein zum Erfassungsraum passendes Defaultobjekt gebildet wird.

Für das Hausmodell (Tabelle 17) wird bei der Beschreibung als DTD auf die alleinige Modellierung über Elemente verzichtet und sofort die Modellierung mit Attributlisten angewendet.

Ziel ist die vielen objektbestimmenden Größen als Gruppen zusammenzufassen. Zur besseren Strukturierung wird die Objektart Haus neben den Angaben zu Lage zunächst in vier Teilobjekte unterteilt. Diese haben die Bezeichnungen: Mauerwerk (wall), Dach (roof), Geschosse (floors), Fassade (facade). Eine andere, ebenso gültige Struktur ist Mauerwerksgeometrie, Dachgeometrie, Geschossstruktur, Materialwahl. Da keine der beiden Strukturen wesentliche Vor bzw. Nachteile aufweist und auch weitere Strukturen denkbar sind, wird die erstere gewählt, denn in dieser Situation kommt es nicht auf Namen, sondern nur auf die Strukturierung an.

7.1.1 Integration in X3D

Um möglichst alle Aspekte der dreidimensionalen Darstellung im Internet umfassend zu berücksichtigen, erweiterte sich das VRML-Konsortium 1998 zum Web3D-Konsortium. Dieses verkündete im Februar 1999 die Einführung von XML zur Definition einer Beschreibungssprache für räumliche Objekte. Ergebnis dieser Bemühungen ist die DTD eXtensible 3D (X3D), die im August 2001 auf der SIGGRAPH in Los Angeles vorgestellt wurde. X3D ist modular konzipiert und besteht aus verschiedenen Komponenten [[WEB3D](#)]:

```
<!ELEMENT Tree (Position,
  TreeDimension, TreeType)>
<!ELEMENT TreeType EMPTY>
<!ATTLIST TreeType
  isType (Birch | Cherry | Oak
  | Poplar | Spruce | ... )
  #REQUIRED>
<!ELEMENT Position EMPTY >
<!ATTLIST Position
  Coordinate % MFVec3d
  #REQUIRED
  Angle %SFDouble>
<!ELEMENT TreeDimension (Age |
  TreeGeometry)>
<!ELEMENT TreeGeometry EMPTY>
<!ATTLIST TreeGeometry
  Diameter %SFDouble #REQUIRED
  Height %SFDouble #REQUIRED >
<!ELEMENT Age EMPTY>
<!ATTLIST Age
  Year %SFInt #REQUIRED >
```

Tabelle 18: Objekt Baum, integriert in X3D

- Core kernel (X3D-1): Diese Komponente definiert die für gängige 3D-Darstellungen unabdingbaren Elemente wie Geometrie, Beleuchtung und Animation.
- VRML97 Profile (X3D-2): Dieser Teil der Beschreibung garantiert eine Rückwärtskompatibilität von X3D zu VRML97. Mit entsprechenden XSL-Stylesheets lassen sich X3D-Dateien problemlos zu VRML97-Dateien konvertieren und so bereits mit aktuellen Browser-Plugins betrachten [[X3D](#)].
- Erweiterungen: Zahlreiche VRML-Erweiterungen können in das X3D-Format übernommen werden. Zusätzliche Erweiterungen lassen sich ebenfalls problemlos einbinden.

<pre> <!ELEMENT House (Position, Wall, Roof, Floors?, Facade?)> <!ELEMENT Position EMPTY> <!ATTLIST Position Coordinate%MFVec3d #REQUIRED Angle %SFDouble> <!ELEMENT Wall (HouseDimensions, WallAppearance)> <!ELEMENT HouseDimensions EMPTY> <!ATTLIST HouseDimensions HouseBox %SFVec3f; # REQUIRED> <!ELEMENT WallAppearance (Texture? Color?)> <!ELEMENT Texture EMPTY> <!ATTLIST Texture File NMTOKENS #REQUIRED> <!ELEMENT Color EMPTY> <!ATTLIST Color RGB %MFVec3d #REQUIRED <!ELEMENT Roof (Overshot, Mansarde?, Walm?, RoofAppearance, SubRoofAppearance)> <!ATTLIST Roof RidgeLength %SFFloat #REQUIRED Roofthickness %SFFloat "0.20" Shed %SFFloat "0.0"> <!ELEMENT RoofAppearance (Texture Color)> <!ELEMENT SubRoofAppearance (Texture Color)> </pre>	<pre> <!ELEMENT Overshot EMPTY> <!ATTLIST Overshot OV %MFFloat "0.0, 0.0, 0.0, 0.0" <!ELEMENT Mansarde EMPTY> <!ATTLIST Mansarde MansardeBox %SFVec3f; # IMPLIED> <!ELEMENT Walm EMPTY> <!ATTLIST Walm RoofBreak %SFFloat "0.0"> <!ELEMENT Floors EMPTY> <!ATTLIST Floors FloorStructure Socket %MFFloat "0.0, 2.5, 2,5" <!ELEMENT Façade (WindowAppearance?, Balcon?)> <!ATTLIST Facade WindowHeight %SFFloat #REQUIRED WindowWidth %SFFloat #REQUIRED WindowDistanceBetween %SFFloat #REQUIRED WindowHeightOffset %SFFloat "0.7"> <!ELEMENT WindowAppearance (Texture Color)> <!ELEMENT Balcon (BalconAppearance, Alignment)> <!ATTLIST Balcon BalconBox %SFVec3f; # REQUIRED> Frequency %SFInt #IMPLIED> <!ELEMENT BalconAppearance (Texture Color)> <!ELEMENT Alignment EMPTY> <!ATTLIST Alignment align (LEFT RIGHT CENTER BLOCK) # REQUIRED > </pre>
--	---

Tabelle 19: Implementierung des Haus Modells auf X3D Basis (Ausschnitt aus der DTD)

X3D lässt sich nahezu unbegrenzt erweitern. Um den Kern X3D-1, der nur die notwendigsten Spezifikationen enthält, werden weitere Profile angelegt, die auf die speziellen Bedürfnisse der einzelnen Anwender eingehen. So wird etwa ein Biologe ganz andere Anforderungen an eine dreidimensionale Darstellung haben als ein Geograph. Eines dieser Profile ist GeoVRML, welches dazu dient, geographische Eigenheiten (beispielsweise Koordinatenwerte vom Typ double und unterschiedliche geodätische Bezugssysteme) in diesem Standard zu etablieren. Neue Knoten werden definiert, die speziell auf die Visualisierung von Geodaten zugeschnitten sind.

Erweiterungen zu VRML/Web3D werden seit einiger Zeit von den Arbeitsgruppen des Web3D-Konsortiums spezifiziert. Die Erweiterbarkeit von X3D hat in der Praxis Auswirkungen in zwei Richtungen. Zum einen muss für die Codierung eines X3D-Browsers nicht die komplette Spezifikation berücksichtigt werden, das heißt: für spezielle X3D-Anwendungen kann man auch spezielle Browser konzipieren, die in Größe und Laufzeitverhalten für den entsprechenden Anwendungsfall optimiert sind. Zum anderen sind die Hauptkomponenten jeder-

zeit durch beliebige Module zu ergänzen. Erweiterungen liegen a priori in der Zielsetzung der Spezifikation und sind nicht wie bei VRML97 proprietär. Neue Anwendungen lassen sich damit in die X3D-Spezifikation aufnehmen, ohne mit den bereits existierenden Modulen in Konflikt zu geraten.

Die Integration der Modellierung in X3D (Tabelle 18 und 19) nutzt im wesentlichen die in X3D und ihrer Erweiterung GeoVRML definierten Datentypen:

- MFVec3d, einen Vektor mit 3 Double Werten
- SFDouble, einem Double Wert aus der GeoVRML Spezifikation
- SFInt als einzelner Wert aus der X3D Spezifikation.

Neue Datentypen sind nicht erforderlich. Damit lassen sich alle notwendigen Größen bilden.

7.1.2 Integration in GML

GML ist eine Erweiterung von XML, die den Transport und die Speicherung von geographischen Informationen unterstützen soll. Das Ziel der GML ist es, einen offenen und herstellerunabhängigen Rahmen zur Definition von Geodaten und deren Anwendungen zu geben, was zwar durch die Spezifizierung der GML durch das OGC erreicht werden soll. Weitere Entwicklungsziele sind die Erweiterbarkeit auf beliebige geographische Aufgabenbereiche, die effiziente Erfassung von räumlichen Geometriemodellen, die Integration von räumlichen und nicht-räumlichen Informationen einer Menge von Geodaten sowie die „Linkfähigkeit“ von Geodaten zu anderen Geodaten oder Dokumenten und somit deren Interoperabilität. Für die Realisierung wird GML Version 2.0 verwendet.

In GML 2.0 werden sogenannte Basisschemata verwendet, die in XML schema spezifiziert sind:

- geometry.xsd: stellt ein Schema für den geometrischen Bereich zur Verfügung, so z.B. Point, Curve, LineString oder eine GeometryCollection mit Multipoint, MultiSurface usw.
- feature.xsd: stellt ein Schema zur Featureverwaltung bereit. Es enthält die Typen featureMember und featureCollection (mehrere Features zusammengefasst), die jeweils den eindeutigen Identifikator fid (featureidentifier) besitzen können.
- xlink.xsd: Dieses Schema enthält eine Bibliothek von Xlink-Attributen zur Unterstützung von Link-Verweisen.

Außerdem enthalten diese Basisschemata bereits in XML eine große Zahl primitiver Datentypen (string, boolean, float, month). Mit ihnen werden Datentypen für Geoinformationen festgelegt, aus denen ein anwendungsspezifisches Schema geschaffen werden kann. Um diese GML-Basisschemen aufzurufen, benutzt man im Dokument die <include>-Funktion, so z.B. <include schemaLocation=" geometry.xsd">.

Um den schon beschriebenen Baum unter Verwendung von GML-Datentypen zu modellieren, hat das dazu gehörende *feature scheme* folgende Gestalt:

```
<element name="Tree" type="ex:SingleTree"
substitutionGroup="gml:_Feature" />
<complexType name="SingleTree">
  <extension base="gml:AbstractFeatureType">
    <sequence>
      <element name="TreeType" type="string" />
      <element ref="gml:location"/>
      <element name="Diameter" type="float" />
    </sequence>
  </extension>
</complexType>
```

```

        <element name="Height" type="float" />
        <element name="Year " type="integer" />
        <element name="Angle" type="float" />
    </sequence>
</extension>
</complexType>

```

Durch das Kommando *substitutionGroup*="gml:_Feature" wird das neu definierte GML-Element als Feature deklariert. Dabei wird der abstrakte Typ *Feature* um den Typen *SingleTree* erweitert. Alle folgenden Elemente sind dann dazugehörige Attribute. Theoretisch kann man diese Kommandos auch weglassen, aber um ein den Regeln entsprechendes Dokument zu erhalten, sollte man nicht auf sie verzichten. Eine Exemplar dieses Typs sieht folgendermaßen aus:

```

<Tree fid="xxxxx">
  <gml:description>A single tree</gml:description>
  < TreeType >birch</ TreeType >
  <gml:location>
    <gml:Point
      srsName="http://www.opengis.net/gml/srs/ epsg.xml#32632>
      <gml:coord>
        <gml:X>5645345.33</gml:X>
        <gml:Y>345232.54</gml:Y>
      </gml:coord>
    </gml:Point>
  <gml:location>
    < Diameter >12.5</ Diameter >
    < Height >8.3</ Height >
    < Year >1989</ Year >
    < Angle > 234.345</ Angle >
  </Tree>

```

Das Property *gml:description* gehört zum erweiterten Typ *gml:_Feature*. Es werden einige allgemeine Properties wie z.B. *description* *location* bereitgestellt. Diese Properties sind optional verwendbar.

Die Beschreibung für die Objektart punkthaftes Haus als feature schema hat dann folgendes Aussehen (Auszug):

```

<element name="House" type="ex:SingleHouse"
          substitutionGroup="gml:_Feature" />
<complexType name="SingleHouse">
  <extension base="gml:AbstractFeatureType">
    <sequence>
      <element ref="gml:location"/>
      <element name="Angle" type="float" />
      <element name="HouseHeight" type="float"/>
      <element name="HouseLength" type="float"/>
      <element name="HouseWidth" type="float"/>
      <element name="WallTexture" type="string"/>
      <element name="WallColor" type="string"/>
      <element name="RidgeLength" type="float"/>
      <element name="Roofthickness" type="float"/>
    </sequence>
  </extension>
</complexType>

```

```

<element name="Shed" type = "float"/>
<element name="RoofTexture" type = "string"/>
<element name="RoofColor" type = "string"/>
<element name="SubRoofTexture" type = "string"/>
<element name="SubRoofColor" type = "string"/>
<element name="OvershotFront" type = "float"/>
<element name="OvershotRear" type = "float"/>
<element name="OvershotLeft" type = "float"/>
<element name="OvershotRight" type = "float"/>
<element name="MansardeHeight" type = "float"/>
<element name="MansardeLength" type = "float"/>
<element name="MansardeWidth" type = "float"/>
<element name="RoofBreak" type = "float"/>
<element name="Socket" type = "float"/>
<element name="GroundFloor" type = "float"/>
<element name="GeneralFloor" type = "float"/>
<element name="WindowWidth" type = "float"/>
<element name="WindowHeight" type = "float"/>
<element name="WindowDistanceBetween" type = "float"/>
<element name="WindowHeightOffset" type = "float"/>
<element name="BalconTexture" type = "string"/>
<element name="BalconColor" type = "string"/>
<element name="BalconLength" type = "float"/>
<element name="BalconDepth" type = "float"/>
<element name="BalconHeight" type = "float"/>
<element name="BalconFrequency" type = "integer"/>
<element name="BalconAlignment" type = "string"/>
</sequence>

```

```
</extension>
```

```
</complexType>
```

dazu ein Gebäudeexemplar als Beispiel:

```

<Tree fid="xxxxx">
  <gml:description>A single house</gml:description>
  <gml:location>
    <gml:Point
      srsName="http://www.opengis.net/gml/srs/ epsg.xml#32632>
      <gml:coord>
        <gml:X>5645365.33</gml:X>
        <gml:Y>345282.54</gml:Y>
      </gml:coord>
    </gml:Point>
  </gml:location>
  < Angle > 234.345</ Angle >
  < HouseHeight >12.3</ HouseHeight >
  < HouseLength >14.5</ HouseLength >
  < HouseWidth >9.5</ HouseWidth >
  < WallTexture >wall.jpg</ WallTexture >

```

```

< WallColor >0.5 0.5 0.5</ WallColor >
< RidgeLength >10.2</ RidgeLength >
< Roofthickness >0.3</ Roofthickness >
< RoofTexture >roof.jpg</ RoofTexture >
< RoofColor >0.2 0.2 0.2</ RoofColor >
< SubRoofTexture >subroof.jpg</ SubRoofTexture >
< SubRoofColor >0.2 0.2 0.2</ SubRoofColor >
< OvershotFront >0.5</ OvershotFront >
< OvershotRear >0.5</ OvershotRear >
< OvershotLeft >0.3</ OvershotLeft >
< OvershotRight >0.3</ OvershotRight >
< MansardeHeight >2.3</ MansardeHeight >
< MansardeLength >14.5</ MansardeLength >
< MansardeWidth >5.1</ MansardeWidth >
< RoofBreak >4.3</ RoofBreak >
< Socket >0.7</ Socket >
< GroundFloor >2.5</ GroundFloor >
< GeneralFloor >2.5</ GeneralFloor >
< WindowWidth >1.3</ WindowWidth >
< WindowHeight >0.8</ WindowHeight >
< WindowHeightOffset >0.7</ WindowHeightOffset >
< BalconyTexture >balcon.Gif</ BalconTexture >
< BalconyColor >0.2 0.2 0.2</ BalconColor >
< BalconyLength >10.2</ BalconLength >
< BalconyDepth >1.2</ BalconDepth >
< BalconyHeight >0.8</ BalconHeight >
< BalconyFrequency >1</ BalconFrequency >
< BalconyAlignment >CENTER</ BalconAlignment >
</House>

```

GML besitzt die folgenden Geometriemodelle:

Point	LineString	LinearRing	Polygon
MultiPoint	MultiLineString	MultiPolygon	MultiGeometry

Zusätzlich existieren Koordinatenelemente zur Beschreibung von Positionen und ein *Box*-Element zur Beschreibung eines Intervalls. Die folgende kurze Beschreibung dient dem für diese Arbeit relevanten Elemente aus GML:

Koordinaten

Die Koordinaten einer Geometrietypeninstanz sind Sequenzen von `<coord>` Elementen, die als Tupel angesehen werden können, oder ein String innerhalb eines `<coordinates>` Elementes, `<coord><X>5.0</X><Y>40.0</Y></coord>` oder `<coordinates>5.0, 40.0</coordinates>`. GML unterstützt drei Dimensionen. Dies ist für die Aufgabenstellung ausreichend.

Point

Ein Point-Element beschreibt ein Exemplar von Typ PointGeometry. Jedes Element besitzt genau ein `<coord>` oder `<coordinates>` Element.

Es kann wahlweise ein *gid* (geometry identifier) eingeführt werden, der dieses Exemplar eindeutig beschreibt

```
<Point gid="p1" srs...>
  <coord><X>56.1</X><Y>0.45</Y></coord>
</Point>
```

LinearRing

Ein LinearRing ist ein geschlossenes Polygon mit linearer Interpolation, das durch mehrere Koordinatentupel beschrieben wird. Die Koordinaten des Anfangspunktes und des Endpunktes müssen übereinstimmen.

Die Angabe eines srs (spatial reference system) wird hier und bei allen weiteren Elementen empfohlen. Hierbei handelt es sich um das geodätische Referenzsystem, auf das sich die Koordinatenangaben beziehen. Dies kann bei einem Vergleich zweier Referenzsysteme wichtig sein oder beim Umwandeln von Geodaten von einem System in ein anderes.

Polygon

Die Grenzen eines Polygons werden durch eine Menge von LinearRing-Elementen beschrieben. Es gibt eine Unterteilung in eine äußere und eine innere Grenze. Die LinearRing-Elemente der inneren Grenzen dürfen sich dabei nicht überschneiden. Außerdem muss es mindestens ein LinearRing-Element geben, das die äußere Grenze beschreibt.

Geometry collections

Einige der Elemente einer Geometry collections sind einfache Erweiterungen der primitiven Geometrietypen. So ist *Multipoint* eine Sammlung mehrerer Point-Elemente, *MultiLineString* eine Sammlung von LineString-Elementen und *MultiPolygon* entsprechend eine Sammlung mehrerer Polygone. Die Definition einer Sammlung verschiedener Geometrietypen ist mit *MultiGeometry* möglich. Ein MultiGeometry-Element enthält als Property ein geometryMember, mit dem man alle anderen Geometrietypen (auch Geometry collections) einbetten kann. Die Anzahl der verschiedenen Typen ist nicht festgelegt.

Mit diesen vordefinierten Geometrietypen kann man nun eigenen Featuretypen räumliche Informationen zuordnen. Auf diese Art kann das Schema, basierend auf GML 2.0, folgendermaßen umgeändert werden:

```
<element name="Tree" type="SingleTree"
substitutionGroup="gml:_Feature />
<complexType name="SingleTree">
  <extension base="gml:AbstractFeatureType">
    <sequence>
      <element name="TreeType" type="string"/>
      <element name="Diameter" type="float"/>
      <element name="Height" type="float"/>
      <element name="Year" type="gYear"/>
      <element name="Angle" type="float"/>
      <element ref="gml:location"/>
    </sequence>
  </extension>
</complexType>
```

Hierdurch wird das folgende Exemplar des Feature Trees möglich:

```
<Tree fid="xxxxx">
  <gml:description>A single tree</gml:description>
```

```

<TreeType>birch</TreeType>
<Diameter>12.5</Diameter>
<Height>8.3</Height>
<Year>1989</Year>
<Angle>234.345</Angle>
<gml:location>
  <gml:Point
    srsName="http://www.opengis.net/gml/srs/epsg.xml#32632">
    <gml:coord>
      <gml:X>5645345.33</gml:X>
      <gml:Y>345232.54</gml:Y>
    </gml:coord>
  </gml:Point>
</gml:location>

```

<Tree>

Die Stellen im unteren Teil, die mit ‚gml:‘ beginnen sind Funktionalitäten aus dem Geometrieschema *geometry.xsd*. Mit *gml Point srsName* wird auf ein Bezugssystem verwiesen.

```

<element name="House" type="ex:SingleHouse"
substitutionGroup="gml:_Feature />
<complexType name="SingleHouse">
  <extension base ="gml:AbstractFeatureType">
    <sequence>
      <element ref="gml:location"/>
      <element name="Angle" type="float" />
      <element name="HouseHeight" type="float"/>
      <element name="HouseLength" type ="float"/>
      <element name="HouseWidth" type ="float"/>
      <element name="WallTexture" type = "string"/>
      <element name="WallColor" type = "string"/>
      .
      ...(gleich der vorherigen Modellierung)
      .
    </sequence>
  </extension>
</complexType>

```

</complexType>

Eine weitere Möglichkeit der Modellierung besteht in der Auflösung der vorherigen, parameterbezogenen Darstellung zugunsten einer Verwendung von Koordinaten für alle geometrischen Elementen. Damit wird die Geometrie in einer Geomety collection zusammengefasst Die weiteren Elemente beschreiben die Erscheinung der Gebäudeteile.

```

<element name="House" type="ex:SingleHouse"
substitutionGroup="gml:_Feature />
<complexType name="SingleHouse">
  <extension base ="gml:AbstractFeatureType">
    <sequence>
      <element ref="gml:location"/>
      <element name="Angle" type="float" />
      <element name="WallTexture" type = "string"/>

```

```

    <element name="WallColor" type = "string"/>
    <element name="RoofTexture" type = "string"/>
    <element name="RoofColor" type = "string"/>
    <element name="SubRoofTexture" type = "string"/>
    <element name="SubRoofColor" type = "string"/>
    <element name="BalconTexture" type = "string"/>
    <element name="BalconColor" type = "string"/>
  </sequence>
</extension>
</complexType>

```

Diese Modellierung ist kompakter und übersichtlicher als die vorhergehende. Sie ist zudem wesentlich näher an der späteren Implementierung. Allerdings hat diese Form auch erhebliche Nachteile. So ist es bei der Modellierung mit Koordinaten wesentlich schwieriger, unzulässige Formen zu detektieren, die bei der Bezugsgrößenmodellierung sofort auffallen.

7.1.3 Vergleich der Modellierungsmöglichkeiten

Mit Hilfe der GeoVRML Spezifikation können die geodätischen Belange der Implementierung umgesetzt werden. Leider beruht GeoVRML nicht auf GML, sondern basiert vielmehr auf Teilen der SEDRIS [[SEDRIS](#)] Spezifikation. SEDRIS ist eine Spezifikation für die Darstellung und den Austausch von Umweltdaten und befindet sich im ISO Normungsprozess bei ISO/IEC JTC 1/SC 24 WG 8. Die SEDRIS Technologie stellt Mittel zur Verfügung, um Umgebungsdaten, wie Gelände, Ozeane, Atmosphäre oder Raum darzustellen und unterstützt den eindeutigen, verlustfreien und standardisierten Austausch dieser Daten.

Der wesentliche Vorteil der Implementierung in X3D liegt in der Nähe zur Interpretation. Mit relativ wenig Aufwand kann aus einer X3D nahen Implementierung ein Werkzeug für die Visualisierung erstellt werden. Eine Komponente zu X3D ist Xj3D, eine 3D Runtime Engine auf Java Basis mit Rendering-Funktionen.

Eine enge Anbindung an GML eröffnet andere Möglichkeiten. Zunächst gewährleistet GML die Bereitstellung der notwendigen Datentypen. Darüber hinaus ist zu erwarten, dass bereits in relativ kurzer Zeit GML das bestimmende Datenformat zum Austausch von Geodaten wird. Daten, welche Basis für die Modellierung sind, beispielsweise Gebäudeumrisse, werden also in naher Zukunft schon in GML geliefert werden.

Conclusio: Während X3D als DTD konstruiert wurde (der Übergang zu XML Schema ist allerdings vorgesehen), wird für GML das XML Schema verwendet. Um eine Entscheidung zu fällen muss daher der Implementierungsrahmen beleuchtet werden. Eine alleinstehende DTD ist unabhängig von anderen Spezifikationen und so einfacher fortzuführen. Daher einige Gedanken zur DTD:

- DTD's haben eine eigene (nicht XML konforme) Syntax.
- Das Prinzip der DTD ist sehr leicht erlernbar und kann daher leicht Verbreitung finden
- Namensräume werden nicht unterstützt. Setzen sich die Beschreibungsregeln eines XML Dokuments aus mehreren DTDs aus verschiedenen Namensräumen zusammen, lässt sich die Mehrfachverwendung von Bezeichnern nicht explizit ausschließen. Die so auftretenden Namenskollisionen verletzen die Bedingung der Wohlgeformtheit von XML Dokumenten.
- Stark eingegrenzter Typvorrat. Außer einiger Stringtypen und expliziter Aufzählung verfügen DTD's nicht über die Möglichkeit weitere Datenformate (wie z.B. color, date, vector, etc.) zu beschreiben. Das Definieren eigener Datentypen ist ebenfalls nicht möglich. Damit geht ein großer Vorteil von XML, die Flexibilität, verloren.

- Rudimentäres Konzept zur Erweiterbarkeit: Die Erweiterungsmöglichkeiten von DTD's lassen sich im Wesentlichen auf einfache String-Ersetzung (über Parameter Entities) reduzieren. Eine systematische und von außen nachvollziehbare Modularisierung lässt sich damit nur schwer durchführen.

Demgegenüber steht das XML Schema:

- Im Vergleich zu DTD ist XML Schema kompliziert zu lesen und zu erlernen. Dies steht einer weiteren Verbreitung entgegen.
- XML Schema ist in XML formuliert, d.h. es existierten sowohl eine DTD zur Beschreibung des XML Schema als auch ein selbstbeschreibendes XML Schema. Ein gegen ein XML Schema geprüfetes XML Dokument (document instance) kann wohlgeformt (well-formed) sein oder gültig (valid) sein. Der Test auf Gültigkeit beinhaltet neben der Überprüfung auf korrekte Reihenfolge und Struktur der Tags auch die Überprüfung auf Einhaltung der Wertebereiche und gültige Datentypen.
- Umfangreicher Vorrat an vordefinierten, einfachen Datentypen: Das XML Schema bietet zahlreiche Datentypen zusammen mit der Möglichkeit, den Wertebereich explizit anzugeben. Mit diesen Basistypen können weitere, komplexe Elementtypen definiert werden. Die einmalige Definition von Daten- oder Elementtypen bei u.U. häufiger Verwendung fördert zum einen die Lesbarkeit des Schemas, zum anderen auch die spätere Verarbeitung durch einen Parser/Validator, etc.
- Das XML Schema erlaubt explizites Gruppieren von Attributen. Attribute, die wiederkehrend in Gruppen verwendet werden (z.B. HTML: width / height) können als Attributgruppe einmalig definiert werden. Dieser Mechanismus, der bei DTD's mit Parameter Entities realisiert wurde, kann explizit angegeben und damit auch von einem Parser/Validator genutzt werden.
- Das XML Schema ermöglicht die Definition neuer Elementtypen auf der Basis vorangegangener Definitionen (Vererbung). Durch die Erweiterung / Einschränkung bestehender komplexer Elementtypen um weitere Elemente oder Attribute lassen sich gleichartige Elementfamilien deklarieren .
- Unterstützung von Namensräumen (namespaces): Durch die Berücksichtigung von Namensräumen lassen sich document instances erstellen, die auf Elemente in verschiedenen XML Schemata zugreifen. Elemente mit gleichen Namen aber unterschiedlicher Struktur können, sofern durch Namensräume getrennt, in einer document instance erscheinen.

Eine sehr gute Lösung ist in dem Ansatz zu sehen, die Modellierung sowohl an X3D und an GML 2.0 anzubinden. Wie sieht ein solcher Weg aus? Während es in X3D eine Spezifikation für Geodaten gibt, ist die Verwendung von 3D-Daten in GML weniger stark vorgesehen, abgesehen von der Verwendung von 3D Koordinaten. Daher wird vorgeschlagen, die GeoVRML Spezifikation derart zu erweitern, dass sie GML konform ist und dann als auf X3D beruhende Modellierung verwendet werden kann.

7.2 Implementierung zur Visualisierung

Die gefundenen Objektausprägungen und ihre Kenngrößen lassen sich auf unterschiedliche Art und Weise realisieren. Die hier dargestellte, beispielhafte Realisierung wird im ISO-Standard VRML 97 (ISO/IEC14772-1:1997) durchgeführt.

Die Realisierung im VRML Umfeld ist nicht bindend. Wie Abb. 87 zeigt, verläuft der Prozess ausgehend von verschiedenen Datenquellen mittels einer Transformation in urbane Objektmuster. Diese komplexen Objektmuster werden in einer Implementierungssprache (z.B. VRML, Java3D, C++) ggf. unter Einbeziehung einer für diese Sprache verfügbaren 3D Engine oder mit

einer der anderen der z.Zt. mehr als 315 Implementierungssprachen in einfachere Objekte umgewandelt.

Für diese Umbildung kann eine Datenbank als Exemplarpuffer eingesetzt werden. Die Implementierungssprachen setzen wiederum auf einen eher physikalisch zu nennenden Layer wie OpenGL, DirectX oder andere auf. Diese Layer wiederum gestatten den direkten Hardwarezugriff, beispielsweise auf die Grafikkarte. Sicherlich ist eine von der Implementierungsseite her einfachere Lösung möglich, indem mehr oder weniger unabhängige oder lose gekoppelte Einzelflächen gebildet werden. Eine solche Vorgehensweise, ist auch in CAD-Systemen üblich. Die Bildung expliziter Objekte hat demgegenüber Vorteile. Nur so ist es möglich, auch objektbezogene Operationen durchzuführen, wie es in GIS üblich ist. Als beispielhaft für eine solche Operation sei die Fragestellung ‚Objektzahl innerhalb einer bestimmten Fläche‘ genannt.

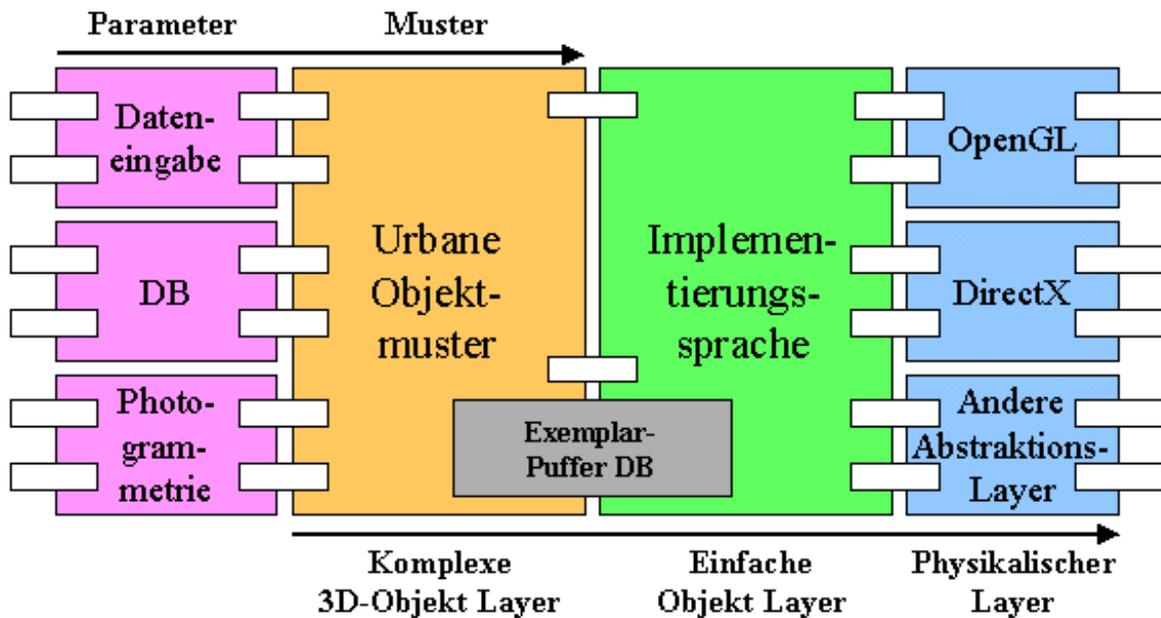


Abb. 87: Einbindung urbaner Objektmuster

7.2.1 Implementierung in VRML

In VRML 97 existiert ein Erweiterungsmechanismus, der es erlaubt, aus bestehenden Objektarten eine neue Objektart zu erzeugen. Dieser Prototyp genannte Mechanismus verwendet vor allem die ‚HAT EIN‘ Aggregationsbeziehung. Ein Prototyp wird innerhalb einer VRML-Datei definiert und kann nur innerhalb der Darstellung dieser Datei genutzt werden.

Die Gestaltung eines Prototypen vollzieht sich in zwei grundlegenden Schritten, die an das Prinzip der Klassendefinition mit privaten und öffentlichen Funktionen in der objektorientierten Programmierung erinnern. Im ersten Schritt wird der Prototyp in der von den Standardknoten bekannten Weise vollständig spezifiziert.

Jede Spezifikation setzt sich aus dem Namen des Knotens, den Feldtypen, den Datentypen, den Feldbezeichnern und im Falle der Fields und ExposedFields zusätzlich aus Default-Werten zusammen. Entsprechend ist bei der Gestaltung eines Prototypen im Knoten PROTO eine neue Kombination von Feldern anzugeben, wobei man vom Vorgang der Deklaration spricht. Über die öffentlichen Felder und Ereignisse lassen sich dann die Eigenschaften des durch den Knoten erzeugten Objekts manipulieren.

Im Gegensatz dazu ist der als Definition bezeichnete Teil der Prototypengestaltung privat, d.h. die entsprechenden Daten sind im Knoten gekapselt. In diesem der Deklaration folgenden Schritt wird das eigentliche Objekt mit seinen Eigenschaften definiert. Hier werden die zu erzeugende Geometrie und gegebenenfalls interaktive Verhaltensformen beschrieben, wobei Standardknoten, bereits gestaltete oder weitere untergeordnete prototypische Knoten eingesetzt werden können. Die Eigenschaften des definierten Prototyps sind somit weitgehend festgelegt und können innerhalb des vorgegebenen Rahmens über die Felder manipuliert werden.

Mit der Gestaltung eines Prototypen wird keine Instanz desselben erzeugt, sondern ein neuer Knotentyp, ein Muster für den Einsatz in einer VRML-Datei, bereitgestellt. Erst wenn dieser an entsprechender Stelle im Quellcode Verwendung findet, entsteht ein Objekt korrespondierenden Typs. An welcher Stelle ein Prototyp im Quellcode eingesetzt werden darf, hängt von dem ersten im Definitionsteil aufgeführten Knoten ab, der den Knotentyp des gesamten Prototyps bestimmt. Steht an dieser entscheidenden Stelle beispielsweise ein Geometrieknoten Box, kann der entsprechende Prototyp ausschließlich überall dort eingesetzt werden, wo Geometrieknoten erlaubt sind, wie beispielsweise im Feld geometry des Knotens Shape.

In VRML 97 besteht neben der ursprünglichen Proto-Form noch ein etwas erweitertes Konzept, der Externproto. Dieser enthält in der Regel einen vollständigen Deklarationsteil, wobei den Feldern keine Standardwerte zugeordnet werden können. Der Definitionsteil beschränkt sich auf die Angabe einer oder mehrerer URLs. Die aufzurufende VRML-Datei enthält dann die vollständige Deklaration und Definition des Prototypen innerhalb eines Knotens PROTO, wobei die Namen des Prototypen und der Felder im Knoten EXTERNPROTO nicht mit denen im Knoten PROTO übereinstimmen müssen.

Um einen Prototypen für eine Objektart zu modellieren, ist es hilfreich, die explizite Modellierung eines Exemplars dieser Objektart zu erzeugen. Diese Modellierung erfolgt bereits unter dem Aspekt der Typisierung.

Grundsätzlich bleibt das Objekt während der Laufzeit unverändert. Deshalb kommen hier Ansätze zum Tragen, die vor oder zum Zeitpunkt der Exemplarbildung (Abb. 88) greifen. Zum Zeitpunkt der Exemplarbildung erfolgt die Implementierung in VRML 97 mittels der Script-Knoten. Diese ermöglichen es, mit den Scriptsprachen Javascript oder VRML-Script oder aber mit Codierungen in Java zu interagieren. Der Script-Knoten ist eigentlich dazu gedacht, ereignisbezogenen Funktionen in externem Programmcode auszulösen. Man kann ein Start-Ereignis auslösen, so dass zur Exemplarentstehung das angesprochene Programm ausgeführt wird. So erhält man ein Verhalten ähnlich einem Konstruktor. Dies kann zum Beispiel in einer Javascript-Datei codiert werden.

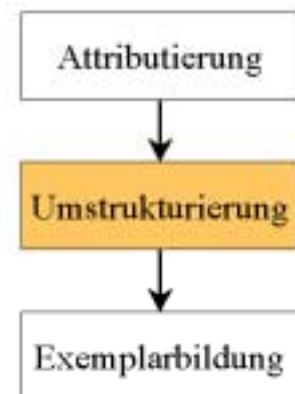


Abb. 88: Positionierung der Umstrukturierung im Exemplarbildungsprozess

Ein anderer Weg der Implementierung liegt im Eingriff vor der Exemplarbildung (Abb. 88). Hierbei wird das Gesamtobjekt mit redundanzfreien und demzufolge nur den notwendigen Attributen beschrieben. Aus dieser Beschreibung werden die für die Unterobjekte notwendigen Anpassungen vorgenommen. Dazu werden in einem separaten Programm die Objektdaten eingelesen, umstrukturiert und ummodelliert wieder ausgegeben. Es hat dabei keine Bedeutung, in welcher Programmiersprache dieses ‚mapping‘ durchgeführt wird, da die notwendigen Funktionen von allen gängigen Programmiersprachen zur Verfügung gestellt werden.

Alle hier aufgeführten Implementierungsverfahren leiden unter hoher Komplexität und wenig Übersichtlichkeit, so dass bei größeren Objektarten, wie zum Beispiel dem Haus, die Implementierung sehr schwierig erscheint. Das Haus in seiner Allgemeinform lässt sich nicht als Prototyp umsetzen. Jede Dachform, die beschrieben wurde hat eine eigene, sich von den anderen

Dachformen unterscheidende Topologie. VRML gestattet nicht, eine variable Topologie zu bilden, wie sie für jeden Dachtyp notwendig ist.

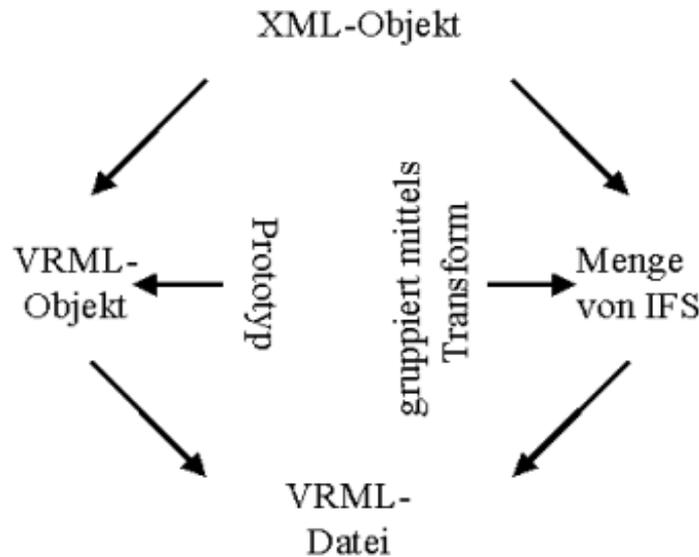


Abb. 89: Formen der Implementierung in VRML über VRML-Objekte links (a) und Indexed Face Sets rechts (b)

Um dieses Problem zu lösen, muss für jede Dachform, also für jede explizite Topologie, ein Prototyp erstellt werden. Dies ist machbar, da aufgrund von Identifikationsgrößen aus den Werten eines Exemplars der Dachtyp bestimmt werden kann. Dies bedeutet aber auch, 15 Prototypen allein für die Objektart Haus zu erstellen. Sicherlich ist dies die nicht wirklich notwendige Prototypenzahl. So ist für die Zeltdach-Formen kein eigener Prototyp erforderlich, indem hier weiterhin zwei Punkte mit den gleichen Koordinaten für den Dachfirst verwendet werden, so dass sich die Zahl der notwendigen Prototypen auf 11 reduziert. Dies ist eine hohe Zahl, die dem Objektgedanken widerspricht. Aus diesem Grunde wird auf eine Verwendung von Prototypen verzichtet. Ein weiterer Nachteil ist die fehlende Universalität dieses Ansatzes. Die Lösung lautet daher: Verwendung von einzelnen IndexedFaceSet Knoten (Abb. 89b), die mittels des Transform Knotens zu einem Objekt zusammengefasst werden. Diese Lösung ist sicherlich nicht sehr elegant, wird aber den Ansprüchen gerecht und ist vor allen auch in anderen 3D Sprachkonzepten anwendbar.

7.2.2 Implementierung in Java3D

Java3D ist Teil der Java Media Suite API. Java3D entstand aus der Zusammenarbeit der Firmen Intel, Silicon Graphics, Apple und Sun. Eine erste Version von Java3D wurde im 4. Quartal 1997 veröffentlicht und seitdem ständig weiterentwickelt. Zielmarkt von Java3D sind alle jene, die ein leistungsfähiges und Plattform-unabhängiges System zur Darstellung von 3-D Welten jeder Größe suchen.

Als Teil der Java-"Welt" ist auch Java3D für viele Systeme erhältlich. Java3D baut dabei auf low-level Technologien wie OpenGL und Direct3D (siehe Abb. 87). Bei der Entwicklung wurde besonderes Gewicht auf die Leistungsfähigkeit gelegt, was mit parallelen Rendertechniken verwirklicht wurde [[SUN-JAVA](#)]. Außerdem ist Java3D bestens gerüstet für zukünftige Entwicklungen in Sachen Hardware, da man bei der Programmierung auf einer abstrakten Stufe bleibt und die Optimierung Java3D überlässt, welches wiederum auf physikalische Layer aus Abb. 87 zurückgreift (DirectX und OpenGL sind implementiert).

Java 3D ist ein API mit einer Klassenhierarchie, um auf einem abstrakten Level dreidimensionale Objekte zu konstruieren, zu manipulieren und zu rendern. Diese Objekte (graphische und Sounds) werden in einem "Virtual Universe" platziert. Dieses wird dann dargestellt. Trotz des großen Funktionsumfangs ist Java3D einfach einzusetzen.

Das Rendern, ein sehr zeitaufwändiger Prozess, wird automatisch gehandhabt und mit dem Einsatz der Thread-Technik von Java ist es möglich, verschiedene Aufgaben parallel zu bearbeiten. Ein Java3D Programm erschafft Java3D Objekte und verstaut diese in einer baumartigen Datenstruktur, die aus verschiedenen 3D Objekten besteht, dem "Scene Graph".

Ein Scene Graph dient der Anordnung von Objekten zur Definition der Grafik-Szene. Er ist eine Baumstruktur, die von der Wurzel ausgehend alle Elemente der Grafik enthält und ist an die Objekthierarchie angelehnt.

Jedes Java3D Programm besteht zumindest teilweise aus Objekten aus der Java3D Klassen-Hierarchie. Diese Objekte beschreiben das "virtual universe". Das API besteht aus über 100 Klassen, die im Paket javax.media.j3d zu finden sind. Diese Klassen werden allgemein als Java3D core Klassen bezeichnet. Zusätzlich zu diesen Klassen gibt es noch weitere, welche oft in Java3D Programmen gebraucht werden. Ein Paket mit solchen Klassen die als Java3D utility Klassen bezeichnet werden ist com.sun.j3d.utils.

Während die core Klassen nur grundlegende Klasse für Java3D enthalten, findet man in den utility Klassen nützlich Funktionalitäten, welche das Programmieren erheblich erleichtern können. Des weiteren gib es noch Packages zur Bearbeitung von räumlichen Ton (com.sun.audioengines). Zukünftige neue Klassen werden in späteren Versionen von Java3D in die utility Klassen eingefügt, während andere utility Klassen in das core-package aufgenommen werden.

Zusätzlich zu den core und utility Klassen braucht jedes Java3D Programm Klassen aus java.awt und javax.vecmath. Die awt-Klassen generieren das Fenster in welchem das gerenderte Bild angezeigt wird. Die Klassen aus javax.vecmath definieren Vektorklassen für Punkte, Vektoren und andere mathematische Objekte.

Java3D hat ganz erhebliche Vorteile und unterscheidet sich von der Lösung mit VRML:

- Der Source-Code und damit die Datengrundlage ist im Gegensatz zu VRML nicht für jedermann lesbar, was zum Schutz der Daten beiträgt. Dies ist allerdings gleichzeitig auch ein Nachteil, für die Verbreitung der Daten.
- Wird Java3D verwendet, so stehen dem Entwickler gleichzeitig alle weiteren Möglichkeiten dieser Sprache zu Verfügung. Dies bedeutet im einzelnen beispielsweise die Verwendung von Input/Output Klassen, dem Graphical User Interface (GUI, Abb. 90) aber auch der XML-Klassen beispielsweise aus dem Java XML Pack [[SUN-JAVA](#)].
- Innerhalb von Java werden für die Objekte tatsächliche Klassen verwendet. Jedes Exemplar ist eine Instanz genau dieser Klassen und ein Teil des Scene Graph

Werden die Vorteile von Java genutzt, entsteht ein vollständiger Browser für die gefundenen Objekte mit einer XML-Schnittstelle. Die Objekte werden dabei in einer der VRML Lösung ähnlichen Struktur in den Scene Graph (Abb. 90) eingeführt. Mit dieser Lösung können die Objekte sogar verändert und gespeichert werden.

Java 3D hat aber auch einige Nachteile. So ist der Zugriff auf die unterschiedlichen Rendering Methoden nicht in einer Weise realisiert, die ein exzellentes Ergebnis liefert. Insbesondere das

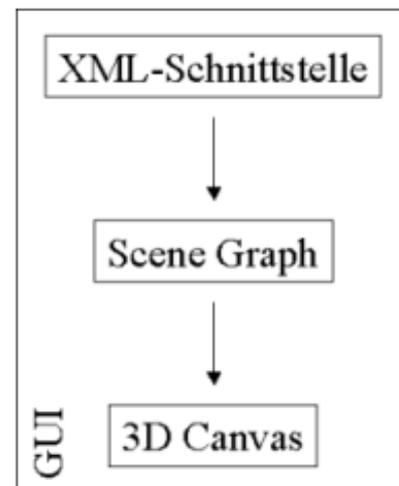


Abb. 90: Struktur der Implementierung in Java 3D

Anti-Aliasing funktioniert nicht wunschgemäß, da noch nicht alle möglichen Rendering Modi des physikalischen Layers und hier insbesondere des Direct X (Abb. 88) bedient werden. Laut Aussage des Java3D Entwicklungsteams [[SUN-JAVA](#)] ist jedoch vorgesehen, dass im Laufe der weiteren Entwicklung von Java3D dieses Manko behoben wird.

Kapitel 8

EINSATZMÖGLICHKEIT IN DER PRAXIS

Bei der Entwicklung des vorgestellten Verfahrens zur Generierung von Objekten für 3D-Stadtmodelle wurde eine Lösung angestrebt, die weitgehend automatisiert, leicht zu handhaben und kostengünstig ist. Diesen Kriterien steht die Forderung gegenüber, dass das Modell möglichst realitätsnah sein muss um in der Praxis akzeptiert zu werden. Er war also ein ausgewogenes Verhältnis zwischen Aufwand und Detailgenauigkeit zu finden, um im Vergleich mit den konkurrierenden photogrammetrischen Lösungen bestehen zu können.

Einsatzbereiche (siehe Kap. 1)	Photogrammetrische Lösung	Datenbasierte Lösung
Spaziergang	ja	ja
Historische Objekte	selten möglich (i. d. R. fehlende oder schlechte Datengrundlage)	ja
3D-Karte	Ja	ja
Exposee	Ja	ja
Lärmausbreitung	Ja	ja
Funkräume	Ja	ja
Raum- und Stadtplanung	Nicht möglich	ja

Tabelle 20: Vergleich der Verwendbarkeit photogrammetrischer und datenbasierter Verfahren nach Einsatzbereichen

In Tabelle 20 wird die Verwendbarkeit der photogrammetrische Verfahren der datenbasierten Lösung gegenübergestellt. Deutlich ist die Flexibilität der in dieser Arbeit vorgestellten datenbasierten Lösung zu erkennen, die alle Einsatzbereiche abdeckt.

8.1. Tauglichkeit des Verfahrens

Die Einsatzfähigkeit des Verfahrens wird an den zwei Ansatzpunkten Realitätsnähe und Projekteinsatz überprüft.

Realitätsnähe

Die Übereinstimmung mit der Wirklichkeit ist Untersuchungsmerkmal der Realitätsnähe. Wie bereits beschrieben handelt es sich bei dem Verfahren um ein approximatives Verfahren, die Wirklichkeit wird demzufolge nicht 1:1 abgebildet. Es wird also eine Darstellung angeboten, die unter dem Gesichtspunkt der Aufwandsminimierung ein möglichst akzeptables, d.h. wiedererkennbares Ergebnis leistet. Hierbei spielen neben der eigentlichen Objektgestaltung auch Aspekte der Bildschirmvisualisierung [BRUNNER 02A, BRUNNER 02B] eine erhebliche Rolle. Ein weiterer Einfluß sind die zur Verfügung stehende Zeit für Bildberechnung, durch welche die Qualität des entstehenden Bildes maßgeblich beeinflusst wird.

In Abb. 91 ist ein Gebäudeobjekt vergleichend als Fotografie, sowie daneben eine Bildschirmkopie aus ähnlicher Perspektive der parametrisierten Abbildung aus dem hier vorgestellten Verfahren. Als Werkzeug zur Visualisierung wurde der VRML-Browser Cortona

von [[PARALLELGRAPHICS](#)] verwendet. Bei diesem ist es möglich, über den Weg kurzzeitiger Verzögerung, die Qualität des entstehenden Bilder entscheidend zu beeinflussen.

Bei der Beurteilung der Bilder ist zu berücksichtigen, das das künstlich erstellte Bild einzelstehend ist, während die Fotografie in einem Kontext mit anderen Objekten (andere Gebäude, Schilder usw.) eingebettet ist.

Deutlich auffallend ist die geringere Detaildichte des künstlich erzeugten Objekts. Dies liegt daran, dass nicht alle denkbaren Details in dem Modell enthalten sind, beispielsweise Kamine, Dachfenster oder Briefkästen, um nur einige zu nennen.

Ein schwierig zu behandelnder Einflußfaktor ist die Farbgebung. Um nicht für jedes Gebäude mit roten Dachpfannen eine erneute Textur zu erzeugen, wurde bei der Visualisierung auf eine bereits bestehende Textur zurückgegriffen. Deren Farbe von unterschiedlichen Faktoren beeinflusst, beispielsweise den Lichtverhältnissen zum Zeitpunkt der Fotografie, der Richtung des Lichteinfalls (dies gilt im Übrigen auch für das Vergleichsfoto), Feuchteverhältnisse, Aufnahme-Sensor, Wiedergabeverhältnisse des Monitors. Die Geometrie der dargestellten Objekte werden zudem von der Perspektive beeinflusst.

Insgesamt gesehen ist eine akzeptable Näherung an das Objekt erkennbar. Dabei ist ein erheblicher Anteil dieses Eindrucks dem Ortgang und der Traufe anzurechnen. Die Verwendung von Fenstern, in denen sich andere Objekte schemenhaft spiegeln ist ebenfalls dem natürlichen Eindruck förderlich.

Projekteinsatz

Das in dieser Arbeit gefundene Verfahren wird an einem Beispielprojekt verifiziert. In der nördlichen Oberpfalz,



Abb. 91 Vergleich Fotografie - Modell

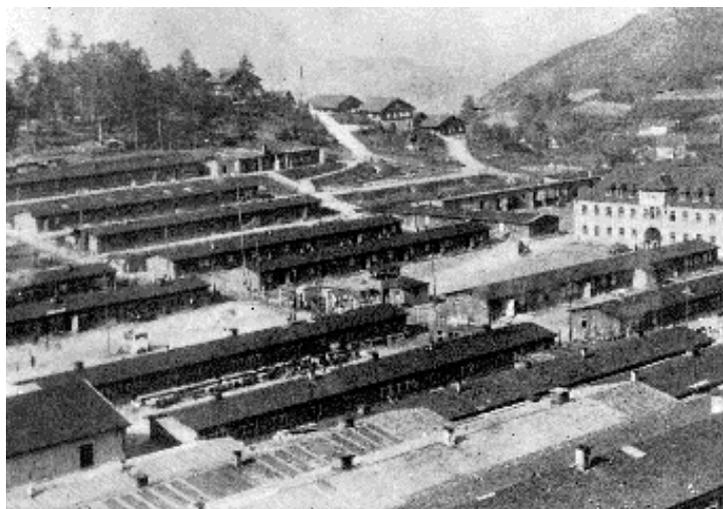


Abb. 92 Blick auf den Lagerbereich

nahe der Grenze zur Tschechischen Republik liegt der Ort Flossenbürg. Die Ortsansicht wird beherrscht von einer Hohenstaufenburg aus dem 12. Jahrhundert. Bekannter ist der Ort jedoch wegen des sich am Ortsrand befindlichen ehemaligen Konzentrationslagers Flossenbürg (Abb. 92).

Nach der Befreiung des KZ Flossenbürg 1945 wurde der Lagerbereich zügig anderer Nutzung zugeführt. Bis 1947 wurde eine Gedenkstätte errichtet, und das Gelände diente als Lager für „displaced persons“. Anschließend wurden die Baracken mit Flüchtlingsfamilien aus Schlesien, Ostpreußen und dem Sudetenland belegt.

Ab 1950 entstanden auf einem Teil des Lagergeländes Wohnungen für Vertriebenenfamilien, ein anderer Teil wurde alsbald Produktionsstätte für Betriebe der Metall- und Textilindustrie. Ein kleinerer Teil des ehemaligen Lagerbereiches blieb erhalten und wurde fortan bewohnt. Die übrige, unbebaute Lagerfläche war ungenutzt und verwilderte.

Seit 1996 befindet sich ein Forschungs- und Dokumentationszentrum als Außenstelle der Universität Nürnberg mit Archiv und Bibliothek für die Gedenkstätte im Aufbau. Durch eine Übertragung des Eigentums am ehemaligen Appellplatzes mit zwei Originalgebäuden, der früheren Häftlingsküche und der Lagerwäscherei, ist die Gedenkstätte 1998 um einige Kernbereiche des ehemaligen Konzentrationslagers erweitert worden.

Trotz des hohen Aufwandes bleibt ein Defizit gegenüber anderen bekannteren Gedenkstätten wie Buchenwald oder Dachau. Der abgerissene Lagerbereich kann zwar rekonstruiert, aufgrund der nachfolgenden Bebauung aber nicht wiederhergestellt werden. Deshalb erscheint es sinnvoll, das KZ nicht nachzubauen, sondern ein Modell der Lagerbebauung zu errichten, um auf diese Art und Weise den Besuchern der Gedenkstätte, aber auch anderen Interessenten via Internet [[GEDENKSTÄTTE](#)], einen Eindruck des Lagers zu präsentieren, nicht ohne den Hintergedanken, evtl. auch über diesen Weg Jugendliche mehr für diese Thematik zu interessieren. Das Ergebnis ist in Abb. 93 rechts zu sehen.

Datenbasis für das Projekt ist ein Plan (Abb. 93, links), der auf Grund von nicht georeferenzierten Luftbildern, aufgenommen nach der Lagerbefreiung, erstellt wurde. Ein Lagerplan existiert in den Lagerunterlagen nicht. Demgegenüber sind jedoch Konstruktionszeichnungen der unterschiedlichen Gebäudetypen erhalten, aus denen die Gebäudeparameter ermittelt werden können. Insgesamt sind 4 Gebäudetypen zu identifizieren:

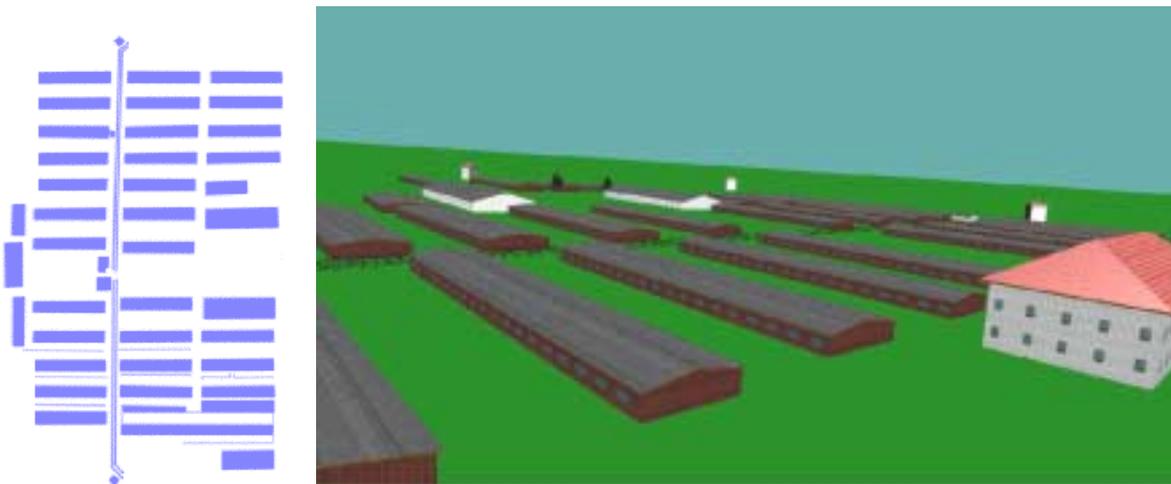


Abb. 93 Plan des Lagerbereiches KZ Flossenbürg (links) und Visualisierung des KZ Flossenbürg aus der Blickrichtung Kommandantur (rechts)

- Kommandantur
- Wäscherei und Küche
- Wachtürme
- Unterkunftsbarracken, baugleich für Häftlinge und SS-Mannschaften

Am Beispiel der Wachtürme wird die Erstellung der Gebäude demonstriert. Nachdem der Grundriss digitalisiert wurde, müssen die Objektattribute eingegeben werden. Dazu wird als erstes die Dachform angegeben, da sich dadurch die Zahl der Bestimmungsgrößen einschränken lässt (Kap 6.2). Daran anschließend kann die Geometrie und die Erscheinung (Texturierung bzw. Farbe) eingegeben werden. Für die Wachtürme sind die folgenden Daten vorhanden:

Gesamthöhe	11,55 m	Sockel	6,50 m
Dachüberstand	0,75 m	Geschosshöhe	2,55 m
Fensterhöhe	0,75 m	Dach	dunkel rot-orange
Fensterbreite	0,90 m	Unterdach	dunkelbraun
Fensterabstand	0,10 m	Mauerwerk	Granit

Tabelle 21: Gebäudeattribute für die Wachtürme des KZ Flossenbürg

Sind mehrere baugleiche Objekte vorhanden, wie es bei den sechs Wachtürmen der Fall ist, wird ein Standard-Objekt gebildet. Die Attributierung kann dann durch Zuweisung oder durch Bildung von Default-Objekten auf andere Objekte übertragen werden. In Abb. 94 wird ein einzelner Wachturm mit Granittextur gezeigt, der mit den Attributwerten aus Tabelle 21 realisiert wurde.

Sind alle Gebäude intern gebildet, erfolgt der Export nach VRML. Die dabei gebildete Struktur enthält eine Wurzeldatei, in der alle weiteren Dateien aufgerufen werden. Für jedes Gebäude und für jeden Baum wird eine gesonderte Datei erzeugt. Durch Aufruf der Wurzeldatei wird die Visualisierung mit dem Browsers angestoßen. Vorteilhaft ist diese Form der Realisierung, da auf diese Weise die einzelnen Objekte leichter identifiziert und dann einfacher als Detailmodell (Abb. 2) bearbeitet werden können.

Die verwendete Software ist nur eine der möglichen Mensch-Maschine-Schnittstellen, die auf diesem Verfahren beruhen. Aufgrund der Objektstruktur ließe sich eine Benutzeroberfläche für weniger gut ausgebildete Anwender erstellen, indem einzelne Eingabemöglichkeiten fragebogenartig erläutert werden.

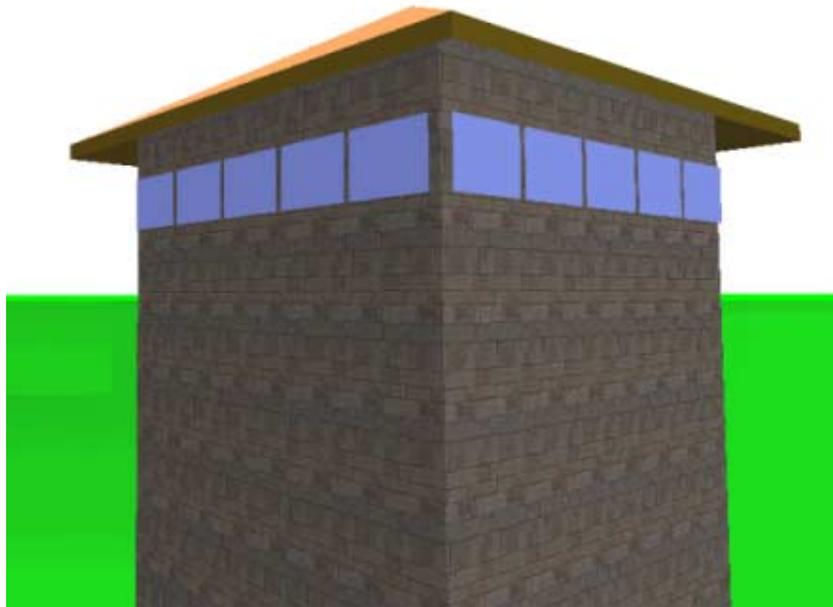


Abb. 94: Wachturm mit Mauertextur

8.2 Wirtschaftlichkeit des Verfahrens

Bestimmend für die ökonomische Betrachtung ist der zu erwartende zeitliche, maschinelle und personelle Aufwand, welcher insgesamt die Kosten für das Verfahren ergibt. Der zeitliche Aufwand hängt bisherigen Erfahrungen nach von verschiedenen Faktoren ab. So sind je nach Dachform verschiedene Anzahlen von Eingaben zu machen. Haben ganze Bereiche gleiche Attribute, so lassen sich diese kopieren (auch als default-Wert) bzw. durch Gebietsauswahl gleichzeitig mehreren Objekten zuordnen.

Die Kosten für die Übernahme von Daten (Grundrisse usw.) hängt vom Datenbestand selbst ab. Hier sind u.U. auch noch Konvertierungen in einlesbare Formate und Plausibilitätskontrollen notwendig. Daher werden diese Kosten wie Abschreibungskosten für Software auf das jeweilige Projekt behandelt.

Die Ermittlung des Wertes der Erstellung von Texturen ist objektabhängig. Einmal erstellt, ist der Aufwand mit einer Attributeingabe gleichzusetzen. Die Texturgewinnung an sich ist demgegenüber recht aufwändig und sollte daher ebenfalls den Kosten für die Software zugeordnet werden.

So ergibt sich als Berechnungsgrundlage für die Kosten- Leistungsrechnung:

$$K_G = A + D + T + \frac{GAZ \cdot X}{O_z} + AZ \cdot X$$

mit

K_G

Kosten für ein Gebäude

A

Anteil der Software und der Hardware
am Projekt (Abschreibung)

AZ * X	Anzahl der Attribute AZ, die direkt eingegeben werden muss, multipliziert mit den Kosten für die Attributeingabe
(GAZ* X)/Oz	Gruppenattributzahl GAZ fasst Objektattribute zusammen, die mehrfach verwendet werden können, Oz ist die Zahl der Objekte mit gleichem Attribut
Datenübernahme D	Grundrisse und andere Daten, die aus Fremddatenquellen übernommen werden können und deren Bearbeitung, projektbezogen
Texturerstellung T	Aufwand zur zusätzlichen Erstellung von Texturen

Tabelle 22: Erklärung der Einzelposten für die Kosten-Leistungsrechnung

Setzt man nun die folgenden eher großzügig gewählte Kosten an:

Kosten/Gebäude	Kosten / Vegetationsobjekt	Kostenart
0,40 €	0,40 €	Aufwand für die Attributierung eines Attributes = Eingabe eines Wertes, geschätzte Dauer 20 sec.
0,40 €	0,40 €	Aufwand für Gruppenattribute
2,00 €	2,00 €	Datenübernahme
2,00 €	1,00 €	Texturierung

Tabelle 23: Kosten für die Generierung eines Gebäudes, bzw. eines Vegetationsobjektes

So ergeben sich nach Tabelle 23 bei durchschnittlich 6 einzugebenden Attributen (Erfahrungswert) Kosten von 6,80 € je Gebäude mit Textur ohne Abschreibungskosten für Hard- und Software.

Der Aufwand für Vegetationsobjekte lässt sich ähnlich wie für Gebäude berechnen. Ein Vegetationsobjekt hat weniger Attribute als ein Gebäude. Meistens werden nur 2 Attribute eingegeben und der Aufwand für Texturen ist geringer, da weniger Texturtypen verwendet werden. So ergeben sich Gesamtkosten von 4,20 € je Objekt.

8.3 Fazit

Für das Verfahren wurde der Nachweis der prinzipiellen Tauglichkeit erbracht. Die Verwendbarkeit in der Praxis muss sich noch erweisen. Die Bedienung des Verfahrens ist leicht erlernbar. Es zeigt gute Ansätze in Richtung Flexibilität und ist daher vielseitig anwendbar. Die Speicherung der Daten im XML-Format trennt die Daten von der Visualisierung. Zudem ist das

Format durch die Verwendung von ASCII-Zeichen menschenlesbar und damit leicht manipulierbar. Es ist lizenzfrei, plattformunabhängig und wird auf breiter Basis unterstützt. [W3].

Wirtschaftlich gesehen ist der Vergleich zu anderen Verfahren schwierig, da sie nicht ohne weiteres verglichen werden können und häufig auch keine wirtschaftlichen Vergleichswerte vorliegen, ausgenommen für den CyberCityModeller. Neben dem Inpho ist CCM das einzige Produkt im kommerziellen Einsatz. Für den CyberCityModeller werden die Kosten pro generiertem Haus mit ca. 10 € für die Geometrie beziffert, der Aufwand für die Texturierung wird mit 80-90 € angegeben, insgesamt also bis zu 100 € je Gebäude ohne Kosten für die Software und Korrekturen (lt. Herstellerangaben).

Die Kosten je Gebäude betragen etwa 7,5 % im Vergleich zum CCM. Allerdings muss die Dachform im Formenkatalog enthalten sein. Dennoch scheint das Verfahren eine Alternative zu bieten, denn mit den insgesamt 16 verschiedenen Dachformen wird eine hohe Anzahl an Gebäudeformen abgedeckt. Eine Erweiterung des Dachkatalogs ist ohne weiteres möglich. Für die Generierung von Vegetationsobjekten liegen keine Vergleichswerte anderer Verfahren vor.

Das vorgestellte Verfahren hat seine Stärken besonders dort, wo andere Verfahren zu kostspielig sind. Hier sind Flurbereinigungsgebiete und insgesamt eher Gebiete des ländlichen Raumes anzusprechen. Besondere Effizienz ist für Gebiete zu erwarten, in denen gleichartige Attribute in großer Zahl vorkommen. Vegetation gleichen Alters, vom Bebauungsplan bestimmte Areale, Bebauung gleichen Alters usw. sind mögliche Merkmale hierfür.

Schwächen hat das Verfahren, wo die Dachform eines Gebäudes nicht mit der Realität in Übereinstimmung gebracht werden kann, obwohl das Verfahren bereits eine große Anzahl verschiedener Dachformen berücksichtigt. Aus diesem Grund sollte es vor allem dort eingesetzt werden, wo diese Dachformen vorliegen.

Kapitel 9

ZUSAMMENFASSUNG UND AUSBLICK

Der Bedarf an 3D-Stadtmodellen wächst stetig. Sie werden für unterschiedlichste Anwendungen benötigt. Darunter fallen beispielsweise Funknetzplanungen, Simulationen für die Raumplanung aber auch als Hintergrund für Spielumgebungen und andere Anwendungen. Die Verfahren und die darauf beruhende Software, die bisher für die Erzeugung von Objekten für 3D-Stadtmodellen verwendet werden, haben überwiegend photogrammetrischen Bezug. Sie müssen von Fachleuten bedient werden, benötigen hochwertige Eingangsdaten und verursachen relativ hohe Kosten mit einem ansprechendem Ergebnis. Zudem rechnen sich die Verfahren nur, wenn relativ große Gebiete bearbeitet werden, da bereits in der Verfahrensvorbereitung hohe Fixkosten entstehen.

Mit dem in dieser Arbeit entwickelten Verfahren wird versucht, Kosten zu reduzieren, indem auf ein photogrammetrisches Verfahren verzichtet wird und als Eingangsdaten bereits bestehende Datensätze verwendet werden. Die Daten, beispielsweise Grundrissdaten, Bebauungspläne, Katasterunterlagen, wurden also eigentlich zu anderen Zwecken ermittelt und erhalten so eine Mehrfachnutzung. Die Mehrfachnutzung der Daten, macht diese besonders kosteneffizient. Basis für das Verfahren ist eine umfassende Definition der Objektklassen, die in 3D-Stadtmodellen häufig vorkommen. Als Objektklassen wurden Gebäude und Vegetation ausgewählt.

Die Klasse der Vegetationsobjekte wird beschrieben, indem Attributwerte zu Objekthöhe und Kronendurchmesser, sowie Alter und Baumart gemacht werden. Daraus werden die drei Modellrichtungen Kugelform, L-System und Kreuztextur entwickelt. Kugelbaum-Modelle vereinfachen die Umrissform der Baumkrone zu einem in alle Richtungen skalierbaren Sphäroid. Der Baumstamm wird als Zylinder dargestellt. Mit L-Systemen wird versucht, den Wachstumsprozess von Pflanzen nachzuvollziehen. Es entsteht je Baum ein Baumstamm, sowie Äste und Blätter als Baumteile. Je nach Baumalter entstehen weitere Verästelungen. Für jede Baumart wird ein eigenes Wachstumsmodell erstellt. Dieses bildet die Eigenarten der jeweiligen Baumart ab, besonders in den formgebenden Faktoren (Astwinkel, Astdichte, Astlängen, usw.) und jahreszeitlichen Änderungen. Kreuztexturmodelle verwenden ebene bildliche Projektionen der jeweiligen Baumart. Mit dem Bild werden zwei Flächen texturiert, welche dann kreuzweise zueinander aufgestellt werden. Texturen für Vegetationsobjekte können aus Fotografien aber auch aus Abbildungen künstlich erstellter Baumvisualisierungen, hervorgegangen aus L-Systemen, gewonnen werden. Die vorgestellten Modelle können mit unterschiedlichem Detaillierungsgrad realisiert werden, um die Vereinfachungsmöglichkeiten bei zunehmendem Abstand vom Betrachter (Level of Detail) nutzen zu können.

Zur Modellierung der Gebäude sind unterschiedliche Dachformen bereitgestellt worden. Je nach Dachform sind unterschiedliche Anzahlen von Parametern vorgegeben, die notwendig sind, um Mauerwerk und Dach zu gestalten. Insgesamt werden 15 Dachformen unterstützt. Die in Europa üblichen Dachformen werden gebildet aus Kombinationen von traufseitige Formen (Giebel, Walm, Krüppelwalm) und giebelseitigen Formen (Sattel, Mansarde, Pult). Hinzu kommen noch Sonderformen wie das Flachdach. Die Dächer haben ein Volumen, d.h. es gibt eine Dach ober- und Dachunterseite, die durch Ortgang und Traufeseite verbunden sind. Auf das Mauerwerk werden Fenster aufgebracht. Das Gebäudemodell wird durch ein Texturmodell verfeinert. Es sind unterschiedliche Texturen für Mauerwerk, Fenster, Dachunterseite und Dachoberseite möglich.

Die Speicherung der Daten erfolgt im XML-Format. Dafür werden Lösungen als unabhängige DTD, in Anlehnung an X3D und in Anlehnung an GML diskutiert. Während X3D nur eine geringe Auswahl an Geo-Elementen bieten, liegen die Defizite bezüglich GML im 3D-Bereich. Ideal ist eine Integration von X3D in GML oder umgekehrt. Es gibt zwar Überlegungen, insbesondere zur Integration von GML in X3D, aber auch zu GML gibt es Überlegungen, die dritte Dimension einzuführen. Bis dies jedoch so weit ist, wird eine unabhängige DTD als Grundlage für eine ‚3D-Stadtmodell Sprache‘ vorgeschlagen.

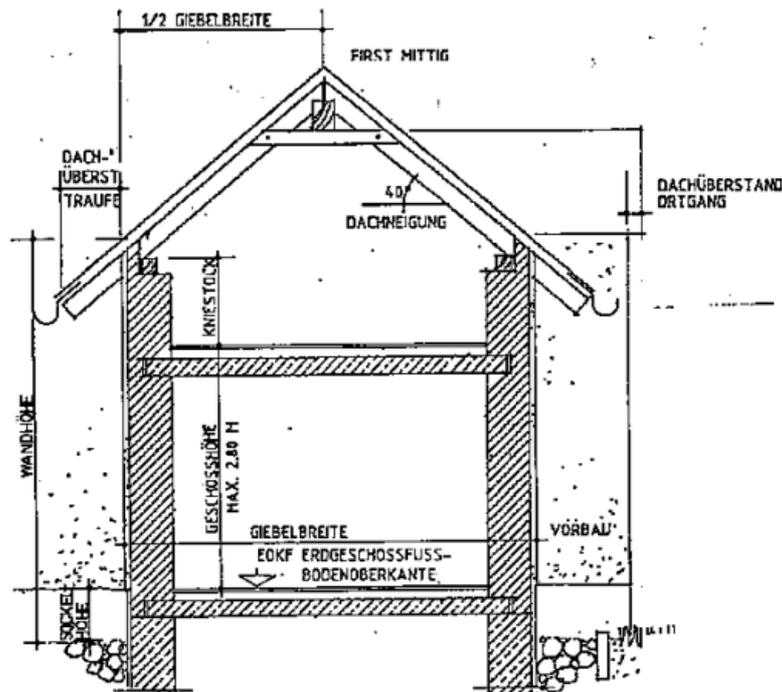
Als Anwendung werden die Daten in VRML und Java3D visualisiert. Damit wird die Universalität des Ansatzes unterstrichen. An einem praktischen Beispiel wird die Funktionsfähigkeit des Verfahrens nachgewiesen. Aus diesem wird ein Kostengerüst ermittelt. Dieses zeigt im Vergleich mit anderen Verfahren, wie effektiv mit dem Verfahren gearbeitet werden kann.

Der nächste Schritt in der Bearbeitung der Modellierung für 3D-Stadtmodelle liegt in der Bereitstellung weiterer Objekte, wie Schild, Straßenbeleuchtung, Bordsteinkante, Straße oder Zaun. Hier kann auf die Erfahrungen mit Baum und Gebäude zurückgegriffen werden.

A n h a n g

BEISPIEL FÜR EINE PLANRECHTLICHE FESTLEGUNG

Auszug für die Bestimmungen des Baugebiets Am Zehenthof, Floss
 PLANUNGSRECHTLICHE FESTSETZUNGEN gem. § 9 (1) BauGB



1. Stellung der baulichen Anlagen gem. § 9 Abs. 1 Nr. 2 BauGB
 - 1.1. Die Richtung der Außenseiten und die Firstrichtung der Gebäude sind entsprechend der Einzeichnung im Bebauungsplan anzuordnen.
2. Flächen für Nebenanlagen sowie Stellplätze und Garagen mit ihren Einfahrten, gem. § 9 Abs. 1 Nr. 4 BauGB
 - 2.1. Im Bebauungsgebiet sind Garagen in einer Länge bis max. 8,0 m und einer Traufhöhe von max. 3,0 m an den durch einen Hinweispfeil (Zufahrt auf die Baugrundstücke) gekennzeichneten Grundstücksgrenzen anzuordnen. Dabei ist ein Garagenvorfeld von mind. 5,0 m einzuhalten.
 - 2.2. Nebenanlagen sind gem. § 14 Abs. 1 BauNVO auf den nicht überbaubaren Grundstücksflächen unzulässig, soweit sie nicht dem Nutzungszweck der im Baugebiet gelegenen Grundstücke oder dem Nutzungszweck des Baugebietes selbst dienen.
3. Grünflächen gem. § 9 Abs. 1 Nr. 15 BauGB
 - 3.1. In dem im Bebauungsplan als öffentliche Grünfläche festgesetzten Bereich sind nur der Nutzung entsprechende Anlagen (Kinderspielplatz, etc.) sowie Anlagen für Zwecke der Versorgung allgemein zulässig.
4. Pflanz- und Pflanzgebot gem. § 9 Abs. 1 Nr. 25 Buchst. a) und b) BauGB
 - 4.1. Für die Ein- und Durchgrünung sollen standortheimische Laubgehölze Verwendung finden. Bepflanzungsvorschlag:

Malus	Apfel
Pyrus	Birne
Fagus	Buche
Acer pseudoplatanus	Bergahorn
Tilia cordata	Winterlinde
Tilia platyphynus	Sommerlinde
einheimische Feldgehölze, Hecken	

4.2. Die im Bebauungsplan festgesetzten Einzelbäume und Baumgruppen sind dauernd zu unterhalten. Insbesondere ist während der Bauzeit jegliche Beeinträchtigung zu vermeiden, Sicherungsmaßnahmen sind zu treffen.

5. Höhenlagen der baulichen Anlagen gem. § 9 Abs. 2 BauGB

Die Erdgeschossfußbodenoberkante (EGFOK) der baulichen Anlagen wird mit höchstens 0,5 m über der Höhe der Verkehrsfläche gem. § 9 (1) 11 BauGB festgesetzt. Gemessen wird von der Mittelachse der Fahrbahn bis zur Erdgeschossfußbodenoberkante (EGFOK). In den Bauvorlagen sind die Geländeprofile einzutragen, die sowohl den derzeitigen als auch den künftigen Geländeverlauf erkennen lassen

6. Abstandsflächen Für die Mindestabstandsflächen gilt die BayBO (4,0 m) soweit in dem Bebauungsplan durch eine Vermessung (3,0 m) nichts anderes festgesetzt.

BAUORDNUNGSRECHTLICHE FESTSETZUNGEN

1. Haus- und Dachform, Dacheindeckung

1.1. Doppelhäuser und Hausgruppen sind mit gleichem Querschnitt, mit gleicher Dachneigung und mit gleicher Dachneigung und mit gleicher Farbe und Material der Eindeckung zu errichten.

1.2. Übersicht über Dachform, Kniestockhöhen und Dachneigung

Gebiet	Geschlosszahl	Dachform	Kniestockhöhe max.	Dachneigung
WA 1	I	Satteldach	max. 0,80 m	Max. 40 Grad
WA 2	II	Satteldach	max. 0,40 m	Min. 40 Grad

1.3 Der Dachüberstand darf an der Traufe waagrecht gemessen 50 cm, am Ortgang 25 cm nicht überschreiten

1.4. Garagen sind mit Satteldächern auszubilden, die Dachdeckung und die Dachneigung ist der des Hauptbaukörpers anzugleichen.

1.5. Als Dacheindeckungsmaterial für Hauptgebäude und Garagen wird nur profiliertes Material in naturroter oder rotbrauner Farbgebung zugelassen.

1.6. Dachgauben Als Dachgauben sind nur einzelne Gauben zulässig, die nicht breiter als 1,20 m (Ansichtsbreite) sein sollen. Sie sind in gleicher Art wie das Hauptdach einzudecken. Lage der Dachgauben im inneren Drittel der Dachfläche.

2. Fassadengestaltung

2.1 Die Verwendung von grellen Farben für Fassadenputz und Außenanstrich sowie das Verkleiden der gesamten Außenfassade mit Asbestzement- oder Kunststoffplatten ist unzulässig.

2.2. Doppelhäuser sind in Farbe und Putzstruktur der Fassade aufeinander abzustimmen.

LITERATURVERZEICHNIS

- Ameri, B., 2000: Automatic Recognition and 3D Reconstruction of Buildings from Digital Imagery, DGK C 526, München, [ISBN 3 7696 9565 8]
- Ammann, E., 1997: Programmierung animierter Welten. Java, JavaScript und VRML, International Thomson Publishing, Bonn, [ISBN 3-8266-0329-X]
- Baillard, C.; Schmidt, C.; Zisserman, A.; Fitzgibbon, A., 1999: Automatic line matching and 3D Reconstruction of Buildings from multiple views, in International Archives of Photogrammetry and Remote Sensing, 32, S. 69-80, München
- Bacher, U.; Mayer, H., 2000: Automatic Extraction of Trees in Urban Areas from Aerial Imagery, International Archives of Photogrammetry and Remote Sensing, 33, p. 51-57
- Bartel, S.; Bill, R.; Boytscheff, C.; Köninger, C., 1997: Datenfusion zur Erstellung realitätsnaher 3D-Geoinformationssysteme für städtebauliche Planungen, Zeitschrift für Photogrammetrie und Fernerkundung (ZPF), Jg. 1997, S. 129-137, [ISSN 0937-9800]
- Bartels, R.; Beatty, J.; Barsky, B., 1987: An Introduction to Splines for Use in Computer Graphics and Geometric Modeling, Morgan Kaufmann, Los Altos, [ISBN 0-934613-27-3]
- Bill, R.; Fritsch, D., 1994: Grundlagen der Geo-Informationssysteme, Band 1, Hardware, Software und Daten, Wichmann, Heidelberg, [ISBN 3-87907-325-2]
- Bill, R.; Fritsch, D., 1996: Grundlagen der Geo-Informationssysteme, Band 2, Analyse, Anwendungen und neue Entwicklungen, Wichmann, Heidelberg, [ISBN 3-87907-326-0]
- Booch, G., 1994: Objektorientierte Analyse und Design, Addison-Wesley, Bonn, [ISBN 3-89317-673-0]
- Boochs, F., 1984: Ein Verfahren zur Herstellung digitaler Höhenmodelle aus photogrammetrischen Stereomodellen mit Hilfe der flächenhaften Korrelation in digitalen Bildern., DGK C 299, München, [ISBN 3 7696 9449 3]
- Bott, O.; Terstappen, A., 1997: Objekte im Web, iX-Magazin, Heise Verlag, Jg. 1997, Heft 11, S. 125-127, [ISSN 0935-9680]
- Braun, C.; Kolbe, Th.; Lang, F.; Schickler, W.; Steinhage, V.; Cremers, A. B.; Förstner, W.; Plümer, L., 1995: On the Models for Photogrammetric Building Reconstruction, Computer & Graphics, 19, pp. 109-118
- Brenner, C., 2000: Dreidimensionale Gebäuderekonstruktion aus digitalen Oberflächenmodellen und Grundrissen, DGK C 530, München, [ISBN 3 7696 9569]
- Brenner, C.; Haala, N., 2000: Erfassung von 3D Stadtmodellen, Photogrammetrie, Fernerkundung, Geoinformation (PFG), Jg. 2000, S.109-118, [ISSN 1432-8364]
- Brunner, K., 2002: Elektronische Bildanzeige von Geoinformation, Kartographische Schriften, Jg. 2002, S. 138-144 [ISSN 0022-9164]
- Brunner, K., 2002: Limitierungen bei der Visualisierung von Geoinformation an Monitoren, Vermessung, Photogrammetrie, Kulturtechnik, 100. Jg., S. 142 – 145 [ISSN 0252-9424]
- Bronstein, N.; Semendjajew, K.; Musiol, G.; Mühlig, H., 1995: Taschenbuch der Mathematik, Harri Deutsch, Thun, [ISBN 3 87144 016 7]

- Burkhardt, R., 1997: UML-Modified Modeling Language, Objektorientierte Modellierung für die Praxis, Addison Wesley, Bonn, [ISBN 3-8273-1226-4]
- Carey, R., Bell, G., 1997: The Annotated VRML 2.0 Reference Manual, Addison Wesley Developers Press, Reading MA, [ISBN 0-201-41974-2], Online Version: "The Annotated VRML97 Reference Manual", auch <http://www.best.com/~rikk/Book/>
- Cartwright, W.; Peterson, P.; Gartner, G., 1999: Multimedia Cartography, Springer Verlag, Berlin, [ISBN 3-540-65818-1]
- Daemisch, K.-H., 1999: .Der Alltag wird unwirklich, in Computerwoche, Die Zukunft der Informationstechnik, Jubiläumsausgabe, S. 180-182, [ISSN 0170-5121]
- Däßler, R.; Palm, H., 1998: Virtuelle Informationsräume mit VRML, dpunkt Verlag, Heidelberg, [ISBN 3-920993-78-0]
- Deussen, O.; Lintermann, B., 2001: .Computerpflanzen, Spektrum der Wissenschaft, 2/2001, S. 58-65, [ISSN 0170-2971]
- Dichter, J., 1996: .Photogrammetric Method for Generating High Resolution Data Bases for Simulation and Virtual Reality, The Workshop of 3D-City Models, Bonn
- Dorffner, L., Forkert, G., 1999: .Photomodelle als Basis eines interaktiven dreidimensionalen Gebäudeinformationssystems, Tagungsband CORP 99, Wien, S. 413-420
- EN 12354 Einheitliche Grundlagen für die Berechnung akustischer Eigenschaften von Gebäuden, Teil 3
- Englert, R.; Gülch, E., 1996: One-Eye Stereo System for the Acquisition of Complex 3D Building Descriptions, GIS,4/1996, [ISSN 1430-3663]
- Fitzgibbon, A.; Zisserman, A., 2000: Automatic 3D Model Acquisition and Generation of new Images from Video Sequences, University of Oxford, unter www.robots.ox.ac.uk/~vgg/vggpapers/Fitzgibbon98b.ps.gz
- Fowler, M., 1997: UML Distilled, Applying the Standard Object Modeling Language, Addison Wesley, Bonn, [ISBN 0-201-32563-2]
- Fritsch, D., 1991: Raumbezogene Informationssysteme und digitale Geländemodelle, DGK C 369, München, [ISBN 3 7696 9416 3]
- Förstner, W., 1999: 3D-City Models: Automatic and Semiautomatic Acquisition Methods, Photogrammetrische Woche, Stuttgart
- Fuchs, C.; Heuel, St.: Feature Extraction in Förstner, Wolfgang (ed.) Proc. of Third Course in Digital Photogrammetry, Bonn, 1998
- Gamma, E., Helm, R.; Johnson, R.; Vlissides, J.; Booch G., 1997: Design Patterns: Elements of Reusable Object-Oriented Software, Reading MA, Addison-Wesley, [ISBN 3-8273-1407-0]
- Gleixner G.;Kranjec, E.; Ranzinger, M., 1995: GIS-Daten als Grundlage für 3D Stadtmodelle in Proceedings zur AGIT 95, Salzburg
- Göpfert W., 1991: Raumbezogene Informationssysteme, Wichmann Verlag, Heidelberg, [ISBN 3-87907-232-9]
- Grau, O., 2000: Wissensbasierte 3D-Analyse von Gebäudeszenen aus mehreren frei gewählten Stereofotos, Universität Hannover, Hannover

- Grün, A.; Wang, X., 1998: C-Modeler: A Topology Generator for 3-D City Models, in International Archives of Photogrammetry and Remote Sensing 32, S. 188-196, Stuttgart
- Grün, A.; Wang, X., 1999: Urban 3D Mapping for a hybrid GIS, Proceedings UM3'99 Urban 3D/Multimedia Mapping, S. 69-78, Tokio
- Gülch, E.; Müller, H., Läbe, T., 1999: Integration of Automatic Processes into Semi-Automatic Building Extraction, in International Archives of Photogrammetry and Remote Sensing, 32, S. 177-186
- Gülch, E., 1996: Extraction of 3D objects from aerial photographs, Proceedings COST UCE Action 4 Workshop "Information Systems and Processes for urban civil engineering", Rom
- Haala, N., 1996: Gebäuderekonstruktion durch Kombination von Bild- und Höhendaten, 70 S., DGK C 460, München, [ISBN 3 7696 9502 X]
- Haala, N.; Brenner, C., Anders, K-H., 1999: 3D Urban GIS from Laser Altimeter and 2D Map Data, in International Archives of Photogrammetry and Remote Sensing, 32, S. 339-346
- Haala, N.; Brenner, C., 1999: Extraction of buildings and trees in urban environments, in International Archives of Photogrammetry and Remote Sensing 54, S. 130-137
- Hartman, J.; Wernecke, J., 1996: VRML 2.0 Handbook, The: Building Moving Worlds on the Web, 448 S., Silicon Graphics/Addison-Wesley, [ISBN 0-201-47944-3]
- Hase, H.-L., 1997: .Dynamische virtuelle Welten mit VRML 2.0, Dpunkt Verlag, Heidelberg, [ISBN 3-920993-63-2]
- Heuer, A., 1997: Objektorientierte Datenbanken, Addison-Wesley, Bonn, [ISBN 3-89319-800-8]
- Heuer, A., Saake, G., 1995: Datenbanken. Konzepte und Sprachen, Thomson, Bonn, [ISBN 3-8266-0619-1]
- Hake, G.; Grünreich, D., Meng, L., 2002: Kartographie, 8. Aufl., Walter DeGruyter, Berlin, [ISBN 3-11-016404-3]
- Henricsson, O.; Grün, A., 1996: Overview of research activities at ETH-Zürich in automated 3-D reconstruction of buildings from aerial images, Tagungsband zur Jahrestagung der Deutschen Gesellschaft für Photogrammetrie und Fernerkundung, Oldenburg, 18.- 20. September 1996
- Hoffmann, C., 1989: Geometric & Solid Modeling - An Introduction, Morgan Kaufmann, San Mateo CA, [ISBN 1-55860-067-1]
- ISO/IEC14772-1:1997 Virtual Reality Modeling Language
- Jacobson, I., 1992: Object-Oriented Software Engineering, A Use Case driven Approach, Addison Wesley, Bonn, [ISBN: 0-201-54435-0]
- Johann, M., 1999: .Mehr Qualität und zwar ganz fix, in Computerwoche, Die Zukunft der Informationstechnik, Jubiläumsausgabe 1999, S. 170-172, München, [ISSN 0170-5121]
- Johnigk, S., 2000: Formale Beteiligungsverfahren im Web, Proceedings CORP 2000, Wien
- Joos, G., 2000: Zur Qualität von objektstrukturierten Geodaten, Schriftenreihe des Studienganges Geodäsie und Geoinformation, Nr. 67, Univ. der Bundeswehr München, München
- Kelch, J., 1999: Immer in Reichweite, in Computerwoche, Die Zukunft der Informationstechnik, Jubiläumsausgabe 1999, S. 236-238, München, [ISSN 0170-5121]

- Klaus, M., 1997: Erstellung von 3D-Stadtmodellen für interaktive virtuelle Begehung, Münchner Fortbildungsseminare Geoinformationssysteme, TU München, München
- Klein, R., 1997: Algorithmische Geometrie, Addison-Wesley, Bonn, [ISBN 3-486-24382-9]
- Kloss, J.; Rockwell, R.; Szabo, K.; Duchrow, M., 1998: VRML 97, der neue Standard für interaktive 3D-Welten im WWW, Addison-Wesley, Bonn, [ISBN 3-8273-1187-X]
- Koppers, L., 1998: 3DMap – Virtual Reality and Geodata, EOGEO 98, Salzburg, ISBN 3-85283-014-1
- Koppers, L., 2000: Dynamische Visualisierung von 3D-GIS-Daten auf Basis der Internet – Technologie, Mitteilungsblatt des DVW Landesverein Bayern Jg. 52, S. 441-458, [ISSN 0723-6336]
- Kraus, K., 1999: Beiträge der Photogrammetrie für 3D-Anwendungen, Fortbildungsseminar Geoinformationssysteme 1999, München
- Kraus K., 1996: Photogrammetrie, Band 2, 3. Auflage, Dümmler Verlag, Bonn [ISBN 3-427-78653-6], mit Beiträgen von J. Jansa und H. Kager
- Mundle, H.; Hurni, L., 2001: Kartographische Online-3D-Plattform mit Datenbankanbindung der Halbinsel Methana (GR), S. 68-79 in C. Hermann, L. Hurni (Hrsg.), Web.Mapping, Wichmann Verlag, Heidelberg, [ISBN 3-87907-353-8]
- Lang, F., 1999: Geometrische und semantische Rekonstruktion von Gebäuden durch Ableitung von 3D Gebäudeecken, Shaker Verlag, Bonn, [ISBN 3-8265-6619-X]
- Lang, F.; Foerstner, W., 1996: Surface Reconstruction of Man-Made Objects using Polymorphic Mid-Level Features and Generic Scene Knowledge ZPF 6/96, S. 193-201, [ISSN 1432-8364]
- Landes, S., 1999: Funktionalität des internetbasierten 3D-Campus-Informationssystems der Universität Karlsruhe, DGK C 513, München, [ISBN 3-7696-9551-8]
- Liedtke, C.-E., 1998: "Strategien der wissensbasierten Bildinterpretation" 44. DVW- Fortbildungsseminar "Hybride Vermessungssysteme", 1997, Band 29 der Schriftenreihe des DVW im Verlag Konrad Wittwer, Neubiberg, [ISBN 3-87919-257-X]
- Mayer, H., 1998: Model-Generalization of Building Outlines Based on Scale-Spaces and Scale-Space Events, in International Archives of Photogrammetry and Remote Sensing, 32, S. 530-536
- Mayer, H., 1999: Scale-Space Events for the Generalization of 3D-Building Data, International Archives of Photogrammetry and Remote Sensing, 33, S. 639-646
- Mayer, H., 1998: Automatische Objektextraktion aus digitalen Luftbildern, DGK C 494, München, [ISBN 3 7696 9534 8]
- Mayr, W.; Mayer, H.; Bacher, U.; Ebner, H., 1999: Automatic Extraction of Trees from Aerial Imagery Semantic Modeling for the Acquisition of Topographic Information from Images and Maps, SMATI '99, S. 155-166, München
- Maentylae, M.: Solid Modelling, Computer Science Press, College Park, 1988 [ISBN 0-88175-109-1]
- Mori, T.; Koiso, K; Tanaka, K., 1999: Spatial Data Presentation by LOD Control, Proceedings UM3'99 Urban 3D/Multimedia Mapping, S. 49-56, Tokio
- Oestreicher, B., 1998: Objektorientierte Softwareentwicklung; Analyse und Design mit der Unified Modeling Language, Oldenbourg, [ISBN 3-486-24787-5]

- Parish, Y.; Müller, P., 2001: Procedural Modeling of Cities, Proceedings Siggraph 2001
- Prusinkiewicz, P.; Lindenmayer, A., 1990: The Algorithmic Beauty of Plants, Springer, Heidelberg, [ISBN 0-387-97297-8]
- Pomaska, G., 1983: Zur Anwendung räumlicher Vektorgraphik im Vermessungswesen, Geodätische Schriftenreihe der Technischen Universität Braunschweig Nr. 4, Braunschweig
- Reinhardt, W., 1998: On-Line Generation and Verification of Digital Terrain Models, in International Archives of Photogrammetry and Remote Sensing 27, S. 546-555, Kyoto
- Reinhardt, W., 1991: Interaktiver Aufbau hochqualitativer digitaler Geländemodelle an photogrammetrischen Stereosystemen., DGK C 381, München, [ISBN 3 7696 9427 9]
- Roehl, B.; Couch, J., 1997: Late Night VRML 2.0 with Java, Ziff Davis, [ISBN 1-56276-504-3]
- Rogers, D.F.; Adams, J. A., 1990: Mathematical Elements for Computer Graphics, 2. Aufl., Mc Graw-Hill, Hamburg, [ISBN 0-07-053529-9]
- Roller, D., 1995: Effiziente Anpassungs- und Variantenkonstruktion (Grundlagen CAD Systeme), Springer Verlag, Heidelberg, [ISBN 3-540-58779-9]
- Rottensteiner, F., 2001: Semi-automatic extraction of buildings based on hybrid adjustment using 3D surface models and management of building data in a TIS, Dissertation, IPF Wien, Selbstverlag, Wien
- Rumbaugh, J., 1993: Objektorientiertes Modellieren und Entwerfen, Hanser Verlag, [ISBN 3-446-17520-2]
- Schäfer, S., 1994: Objektorientierte Entwurfsmethoden, Addison-Wesley, Bonn, [ISBN 3-89319-692-7]
- Sester, M.; Klein, A., 1999: Rule based Generalization of Buildings for 3D-Visualization, in Proceedings of the 19th International Cartographic Conference of the IAC, Ottawa, Canada, S. 214-224
- Schlüter, O., 1998: VRML-Sprachmerkmale, Anwendungen, Perspektiven, O'Reilly Verlag, Köln, [ISBN 3-89721-121-1]
- Ueda, M.; Tokumaru, K.; Sibasaki, R., 1999: 3D Modeling of Urban Landscape for Monsoon Asia Environment, Proceedings UM3'99 Urban 3D/Multimedia Mapping, S. 95-101, Tokio
- Uhlenkücken, C.; Schmidt, B., 2000: Sichtbare Verbesserung, Trends in der 3D-Geodatenvisualisierung, GeoBit 3/2000, S. 22-24, [ISSN 1430-3663]
- Lorber G., 1996: Aufbau eines 3D-Stadtmodells Graz, in Ingenieurvermessung 96, Band 2, D8, Dümmler-Verlag, Bonn, [ISBN 3-427-78111-9]
- Vince J., 1995: Virtual Reality Systems, Addison-Wesley, Bonn, [ISBN 0-201-87687-6]
- Weber, A.; Elfes, H., 1999: Untertauchen und begreifen, in Computerwoche, Die Zukunft der Informationstechnik, Jubiläumsausgabe 1999, S. 294-295, München, [ISSN 0170-5121]
- Werner, G., 1993: Holzbau, Teil 2, Dach- und Hallentragwerke, Werner (WIT), Düsseldorf, [ISBN 3-8041-4041-6]
- Weidner U., 1997: Gebäudeerfassung aus Digitalen Oberflächenmodellen, DGK C 474, München, [ISBN 3-7696-9514-3]

Weidner, U.; Foerstner, W., 1995: Towards automatic building extraction from high resolution digital elevation models, ISPRS Journal of Photogrammetry and Remote Sensing, 50, S. 38-49

Zerbe, K. , 1999: Bauplan für Objekte, c't, Heft 21, Seite 338-352, ISSN

INTERNETADRESSEN

3D Kroatien	http://vrml.zesoi.fer.hr/croatia/
AIDA	http://www.tnt.uni-hannover.de/project/kbs/aida/
ART+COM	http://www.artcom.de
Baulexikon	http://www.baulexikon.de
BIW	http://www.fh-deggendorf.de/biw/
Blaxxun Interactive	http://www.blaxxun.com
BOTANIK	http://www.biologie.uni-ulm.de/lehre/botanik/
Beratung zu Hardware	http://www.tomshardware.de
CCM	http://www.cybercity.ethz.ch
CMU	http://www.maps.cs.cmu.edu
CVL	http://www.cfar.umd.edu/cvl
CVRL	http://vis-www.cs.umass.edu
DCS	http://www.dcs.uky.edu
Digital City Project	http://www.digitalcity.gr.jp
EPFL	http://www.epfl.ch
Encarta	http://www.encarta.de
Flashpix	http://www.flashpix.com
Flatland	http://www.flatland.com
GisCluster	http://www.giscluster.at
Gedenkstätte	http://www.flossenbuerg.de
Internet3d	http://www.internet3d.at
Intervista	http://www.intervista.com
IPF Karlsruhe	http://www-ipf.bau-verm.uni-karlsruhe.de
IPF Wien	http://www.ipf.tuwien.ac.at
German VRML User Group	http://homepages.msn.com/NonProfitBlvd/web3dev/
Geometra	http://www.geometra-3d.com
Go3DM	http://www.grintec.at
GTOPO30	http://edcdaac.usgs.gov/gtopo30/gtopo30.html
Inpho	http://www.inpho.de
Internet3D	http://www.internet3d.at
Karmanaut Cosmo Player	http://www.karmanaut.com/cosmo/player/
Kompressionsalgorithmen	http://www.ztt.fh-worms.de/de/sem/ws95_96/
Lehrstuhl für Waldwachstumskunde	http://www.wwk.forst.tu-muenchen.de

LSV	http://www.empa.ch/deutsch/fachber/abt177/laerm
MPEG	http://www.csel.it/mpeg/
NewTek	http://www.newtek.com
Quick Time VR	http://www.apple.com/de/quicktime/authoring/
O ₂ -Technology	http://www.o2tech.com/
OOSE	http://www.oose.de
OWiS	http://www.otw.de/
Parallelgraphics	http://www.parallelgraphics.com/
Photomodeller	http://www.photomodeller.com
Planet9	http://www.planet9.com
Platinum	http://www.cai.com/cosmo/
Rational Rose	http://www.rational.com/
SCOP	http://www.ipf.tuwien.ac.at/produktinfo/scop/
Shout3D	http://www.shout3d.com
Sun – Java	http://java.sun.com
UCL Geomatic Engineering	http://www.ps.ucl.ac.uk
US Army TEC	http://www.tec.army.mil/TD/tvd/survey/survey_toc.html
Virtual Helsinki	http://www.arenanet.fi/helsinki/eng/
VTP	http://vtterrain.org/Implementation/Libs/buildings.html
VISICS	http://www.esat.kuleuven.ac.be/psi/visics/
W3	http://www.w3.org/XML
Web3D	http://www.web3d.org

Alle Zugriffe erfolgten vor dem 1.6.2002