Diplomarbeit

Automatische Generierung und Visualisierung von 3D-Stadtmodellen

Marcel Lancelle, 2003/2004

Technische Universität Braunschweig
Institut für Computergraphik
Prof. Dr. D. Fellner

Inhaltsverzeichnis

Kapitel 1

Einleitung

Das Thema der 3D-Stadtmodelle ist sehr vielfältig und interdisziplinär. Dieses Kapitel gibt einen Überblick über den behandelten Stoff und die Ziele dieser Diplomarbeit.
Neben theoretischen Überlegungen sollte auch die konkrete Realisierung einzelner Aspekte unter den gegebenen Einschränkungen erfolgen. Ein schnelles Ergebnis ohne zusätzliche Kosten erforderte die automatische Verarbeitung verfügbarer bereits vorhandener digitaler Daten. Dadurch wurde es auch möglich, die Daten für eine ganze Stadt zu verarbeiten. Mit Hilfe der Implementierung können Forschung und Praxis verknüpft werden und zahlreiche weitere Forschungsprojekte bieten sich an, die auf dieser Grundlage aufbauen können. Es stand kein Komplett-System zur Verfügung, auf das aufgebaut werden konnte, so dass von Grund auf neu angefangen und vieles neu implementiert werden musste. Dadurch hielten sich zwar die Einschränkungen der externen Komponenten in Grenzen, doch war nur Zeit für einfache Realisierungen; es gibt für alle Teilbereiche der Erzeugung und Visualisierung bereits jeweils bessere Implementierungen. Der entstandene „CityModelViewer“ ist dennoch bereits ein nützliches Tool, das diverse Arten von Daten aus ganz unterschiedlichen Quellen nutzen und bereits auf einem durchschnittlich ausgestatteten PC interaktiv visualisieren kann.
Mit Hilfe von Daten der Stadt Braunschweig wurden nach und nach immer mehr Programmteile entwickelt. So wurden hauptsächlich die für diese Stadt relevanten Probleme behandelt. Die konkrete Bearbeitung von Problemen ist ohne vorhandene realistische Daten kaum sinnvoll und wurde daher wenig berücksichtigt. Für Braunschweig existierte noch gar kein digitales 3D-Stadtmodell, so dass dies eine gute Untersuchung für später geplante professionelle Projekte darstellt. So können auch die Mitarbeiter der Stadt schon jetzt erste praktische Erfahrungen auf diesem Gebiet sammeln. Eine Nutzung außerhalb der genannten Bereiche, z.B. im privaten Umfeld, ist leider aus rechtlichen Gründen wegen der benutzten Daten nur sehr eingeschränkt möglich.

1.1 Motivation

Es gibt viele sehr differenzierte Einsatzgebiete und somit auch viele unterschiedliche Anforderungen an ein Stadtmodell. So sind beispielsweise zur Visualisie-

rung ganz andere Informationen wichtig als für eine Simulation der Schallausbreitung. Daher muss die Verwaltung der Daten sehr flexibel sein. Außerdem fallen z.T. sehr große Mengen von Daten an. Eine Akquisition dieser Daten ist oft teuer und personalaufwendig. Die Benutzung bereits vorhandener Daten ist dabei sehr hilfreich, insbesondere, wenn neue Informationen durch Kombination mehrerer Datenquellen automatisch erzeugt werden können. Leider sind die Qualität und die Art der Quellen von Stadt zu Stadt unterschiedlich und manchmal kaum für eine automatische Verarbeitung zu gebrauchen. Die Vielzahl der möglichen Anwendungen macht solch ein Modell sehr attraktiv für Städte und auch größere Firmen oder Messen.
Es gibt bereits eine verwirrende Vielzahl von einzelnen Projekten, sowohl anwendungs- als auch forschungsorientiert, doch trotz Bemühungen zu internationalen Standards sind die bereits existierenden Modelle meist Einzellösungen. Allgemein akzeptierte Standards werden dringend benötigt, doch schon für den 2D-Fall gibt es sie nicht, in 3D ist das Problem noch komplexer. Nur damit kann jedoch erreicht werden, dass nicht viel Arbeit für jede Stadt neu aufgewandt werden muss sondern Verfahren perfektioniert werden können, die auf sehr viele Städte anwendbar sind.

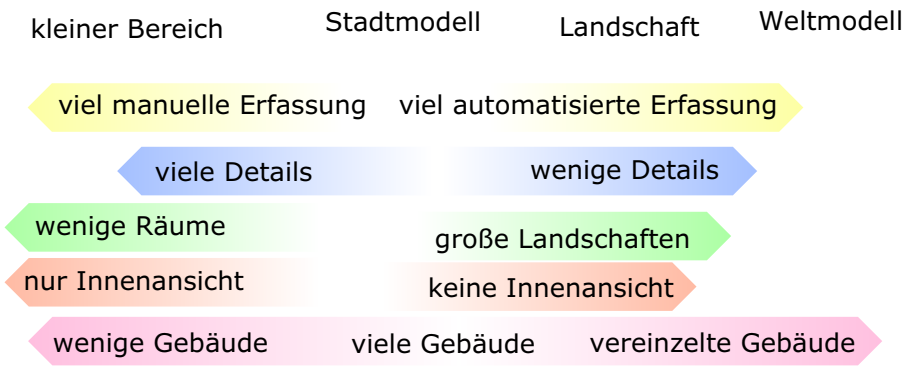

Abbildung 1.1: Eigenschaften verschiedener Modelle von Teilen der Erde

Noch gibt es keine optimale Lösung, die allen Anforderungen gerecht wird.

1.2 Ziele

In dieser Arbeit wurden kurzfristige Ergebnisse angestrebt, um ein Model vor allem zur Forschung bei der Visualisierung zu erhalten. Leider standen keine vorhandenen Implementierung zur Verfügung, so dass dieses System von Grund auf neu entworfen werden musste. Allerdings gab es so auch keine externen Einschränkungen. Die Konvertierungs-Routinen importieren vorhandene Daten zur Visualisierung. Es erstellt jedoch keine professionell nutzbaren Stadtmodell-Datenbanken oder stellt Editoren für die Objekte zur Verfügung, dies sind Aufgaben eines GIS und weiteren Zusatzlösungen. Für den professionellen Einsatz sind auch weitere Aspekte wie User-Interfaces, Benutzerschulungen und Support von Bedeutung, auf die hier nicht eingegangen wird. Vielmehr stand das Austesten der Möglichkeiten vor allem im Bereich der Visualisierung im Vordergrund, ohne konkrete Kunden und Anforderungen zu berücksichtigen.

1.3 Probleme

Eine große Schwierigkeit war auch das Zusammentragen der Daten aus unterschiedlichen Quellen. Diese sind teilweise nur direkt an den Stellen und Ämtern verfügbar, wo sie auch bearbeitet werden. In vielen Stadtverwaltungen gibt es keinen zentralen Datenbestand.
Visionäre sehen in zukünftigen Anwendungen für 3D-Stadtmodelle eine alltagsbegleitende, von vielen benutzte Applikation, doch praktische Probleme wie Finanzierung und Art der Datenformate sind z.Z. überwiegend, so dass solche Nutzungen für die nahe Zukunft unwahrscheinlich sind.

1.4 Wichtige Begriffe

3D-Modelle: Art und Umfang von 3D-Stadtmodellen können sehr unterschiedlich sein. Manchmal bestehen die sog. 3D-Modelle nur aus 2,5D Daten wie Höhendaten. Andere Modelle enthalten detailliert modellierte Gebäude und Texturen. Oft handelt es sich nur um kleine Stadtbereiche.

GIS: Ein Geo-InformationsSystem besteht aus Daten und Softwarepaketen. Damit werden raumbezogene Informationen erfasst, verwaltet, analysiert und graphisch dargestellt.

LoD 1, 2, 3, 4: Level of Detail. Je nach Literatur bedeuten die Zahlen unterschiedliche Detaillierungsgrade von Gebäuden.

Maßstab: Eine Maßstabsangabe im Zusammenhang mit digitalen Karten oder Bildern bezieht sich auf den Deutschen Städtetag, der unterschiedliche Raumbezugsebenen (Generalisierungsstufen) mit Maßstabsangaben benannt hat. Eine Karte „im Maßstab 1:5000“ bezeichnet eine Karte mit mittlerer Generalsierung.

LIDAR: LIght Detection And Ranging: Akquisition von Punktewolken durch Laufzeitmessung eines an der Oberfläche reflektierten Laserstrahls.

DEM, DSM, DOM: Digital Elevation Model, Digital Surface Model, Digitales OberflächenModell: Es handelt sich hierbei allgemein um eine Oberfläche, bei der auch Gebäude und Pflanzen erfasst sein können.

DTM, DGM, Ground DEM: Digital Terrain Model, Digitales GeländeModell: DOM der Erdoberfläche.

Building DEM, normalisiertes DOM: Die Höhen dieses Oberflächenmodells sind bezogen auf die Höhe der Erdoberfläche (DOM - DGM)

GK, WGS-84: Gauß-Krüger-System: eine verzerrungsarme, meridianstreifenbasierte 2D-Abbildung der Erdkugel

Kapitel 2

Anwendungen

Viele Anwendungen für die Nutzung von Stadtdaten werden durch GIS abgedeckt. In dieser Arbeit werden nicht solche klassischen GIS-Anwendungen wie die Nutzung von automatisierten Liegenschaftskatastern behandelt, sondern eher die neuen Anwendungen, die erst mit einem 3D-Stadtmodell richtig genutzt werden können.
Für 3D-Stadtmodelle gibt es eine Vielzahl von interessanten Einsatzgebieten. Viele raumbezogene Informationen in einer Stadt können sinnvoll benutzt werden. Allerdings wären viele dieser Ideen auch in 2D-Karten realisierbar, doch in der Praxis ist kaum etwas implementiert. Die Erwartung an ein 3D-Stadtmodell ist neben dem reinen Hinzufügen der dritten Dimension offenbar, dass auch diese Funktionalität automatisch vorhanden sein soll.
Der kurzsichtige Ansatz, die Daten zu digitalisieren, indem einfach digitale Zeichnungen ohne viel Semantik erzeugt werden, kann spätestens bei einem 3D-Stadtmodell nicht mehr verfolgt werden. Besser ist die Erfassung von Daten, aus denen u.a. solch eine Zeichnung automatisch generiert werden kann. Dadurch wären auch viele andere Arten von zusätzlichen Applikationen einfacher zu realisieren.
Es gibt Annahmen, die auf fast alle Einsatzgebiete zutreffen. So kann oft von einem 2,5D-Modell ausgegangen werden, bei dem die Erdkugel lokal durch eine Ebene angenähert wird. Andere Anforderungen widersprechen sich jedoch vorerst. So ist es in einer Applikation wichtig, dass alle Daten möglichst der Realität entsprechen, in einer anderen sollen aber zufällige Details hinzugefügt werden, um ein visuell schöneres Ergebnis zu erhalten.
Die Analyse und Planung von Verkehr oder Landnutzung sind ebenfalls interessant, aber nicht unbedingt an ein 3D-Stadtmodell gebunden. Die Visualisierung der dafür notwendigen Informationen kann natürlich trotzdem auch im 3D-Stadtmodell erfolgen.
Für spezielle Anwendungen können detaillierte Modelle integriert werden, die auch von Innen besichtigt werden können, wie z.B. bei einem Museum oder einem noch nicht realisierten Projekt.
Oft ist für die Objekte vor allem die äußere Hülle wichtig. Bei der Erstellung der Daten darf aber nicht der Fehler gemacht werden, nur an eine spezielle Anwendung zu denken.

Die Anwendungen sind so verschieden, dass eine Gruppierung schwer fällt, die

Grenzen der folgenden Unterkapitel sind fließend.

2.1 Information und Navigation

Die interaktive Visualisierung eines 3D-Stadtmodells kann als Ersatz oder Ergänzung einer 2D-Karte dienen. Die Integration aller Arten von raumbezogener Information ist möglich. So könnten Daten ähnlich denen in den *Gelben Seiten* mit dem Modell verknüpft werden. Für Geschäfte interessieren z.B. Telefonnummern, Öffnungszeiten, Homepages und evtl. aktuelle Angebote. Eine Suche oder Ergebnisanzeige ist nun auch raumbezogen möglich. Die Lage von Suchergebnissen kann schnell überblickt werden. Zusätzlich zu einer textbasierten Suchmaschine kann so auch anhand der Position gesucht werden. Sehr nützlich können diese Daten auch sein, falls sie mobil zur Verfügung stehen. Viele dieser Anwendungen sind auch für den 2D-Fall möglich, wurden jedoch auch hier kaum realisiert. Seit kurzer Zeit gibt es für Nordamerika eine ortsbezogene Suche von Google, die sich noch in der Testphase befindet. Eine Verbindung von digitaler Stadt und 3D-Stadtmodell ist sehr sinnvoll. Das Konzept von digitalen Städten wird gut in [Ish00] erklärt. Auch die Nutzung aktueller Informationen über Veranstaltungen und Neuigkeiten sowie die Möglichkeit sozialer Interaktion werden dort beschrieben.
Das Modell kann auch zur geographischen Bildung und als allgemeine Referenz dienen. Statistische Informationen und andere übliche Karten können ebenfalls visualisiert werden.
Die Informationen sind oft auch für ortsfremde Personen von Interesse, die sich nicht erst sehr viele Daten über das Internet holen wollen. Ein Streaming der Daten in entsprechenden LoDs ist hier sinnvoll. Progressiv können immer mehr relevante Details übertragen werden. Diese Anforderungen verkomplizieren die notwendige Architektur allerdings erheblich.

2.1.1 Tourismus, E-Commerce

Für Touristeninformation, virtuellen Tourismus und Reiseplanung ist ein 3D-Modell gut geeignet, um sich mit Plätzen von besonderem Interesse bekanntzumachen. Hotels, Sehenswürdigkeiten, Verkehrsanbindungen und Restaurants können für diese Zwecke besonders hervorgehoben werden.
Schon seit einigen Jahren gibt es die Möglichkeit, sich bei großen Veranstaltungen vor dem Kauf von Tickets im Internet die Sicht vom jeweiligen Platz auf die Bühne bzw. das Spielfeld in einem 3D-Modell anzusehen. Dieses Konzept ist auch für Platzreservierung im Theater, Kino oder Restaurant vorstellbar.

2.1.2 Navigationssysteme

Viele Personen haben Probleme, sich mit Hilfe einer zweidimensionalen Karte zu orientieren. Von Fußgängernavigation und Systemen für Autos bis hin zu gedruckten Plänen können die dreidimensionalen Darstellungen viel anschaulicher wirken. Intermodale Navigationshilfen sind Systeme, die unabhängig von dem benutzten Verkehrsmittel Routen berechnen können. Eine 3D-Visualisierung kann hier besonders helfen.

Ein 3D-Landschaftsmodell wird bereits für eine Wanderroutenplanung eingesetzt. Zu erwartende Aussichten und die Streckenführung können so vor der Wanderung betrachtet werden und auch helfen, bei der späteren Wanderung den richtigen Weg zu finden.

2.2 Immobilienmarkt

Neben der Liegenschaftsverwaltung sind 3D-Modelle auch für Standortmarketing bei Verkauf oder Vermietung von Objekten interessant. Hier sind ansprechende Bilder gefragt, die die Kunden überzeugen sollen.

2.3 User Interface für GIS

Das 3D-Stadtmodell kann für das GIS als Interaktionsplattform dienen. Klassische GIS-Userinterfaces arbeiten mit 2D-Linien. Für manche Operationen ist es sinnvoll, alle verfügbaren Daten auch in einer dreidimensionalen Ansicht betrachten zu können. Sichtbarkeiten, Verdeckungen und andere Aspekte der Höheninformation können dadurch besser begriffen werden als durch eine rechnerische Auswertung der Informationen.

2.4 Stadtplanung

Für den Neu- und Umbau ist es wünschenswert, schon beim Entwurf oder Genehmigungsverfahren einen möglichst guten Eindruck vom fertigen Objekt zu erhalten. Dabei kann es sich um Gebäude, Spielplätze oder auch ganze Stadtviertel handeln. Zudem besteht bei einer eigenen interaktiven Visualisierung nicht mehr die Gefahr der Manipulation der 3D-Ansichten durch die Architekturbüros. Eine große Bedeutung hat auch die Ermittlung der Sichtbarkeit bzw. Verdeckung (line-of-sight) durch das geplante Objekt. So soll für ein Projekt z.B. das Landschaftsbild erhalten bleiben und die Sicht von einem zum anderen Ort aus kulturellen Gründen nicht verbaut werden. Auch die präzise Simulation von Sonnenständen und Schattenwurf ist möglich.
Für Entscheidungen bei großen Projekte ist es auch wichtig, dass mehrere Betrachter gleichzeitig aus jedem Winkel sehen, zeigen und diskutieren können, daher werden oft noch Holzmodelle angefertigt. Sollen die Vorteile von den vielen Möglichkeiten der Softwarelösung und der guten Interaktion der physischen Modelle kombiniert werden, können eine CAVE oder VR-Brillen zum Einsatz kommen.
Ein Stadtmodell, das von vielen unterschiedlichen Benutzergruppen verwendet wird, sollte Planungen ohne Veränderung des aktuellen Zustands ermöglichen. Wichtig ist die Berücksichtigung des Existenzzeitraums der einzelnen Objekte. So kann parallel zur Planung auch das aktuelle Modell weiter genutzt werden, hier ist also die 4. Dimension sehr nützlich. Zum Ausprobieren von verschiedenen Alternativen könnten noch weitere Attribute eingefügt werden. Um die ungenaue Planung von Details zu visualisieren, bieten sich Nicht-Photorealistische (NPR) Darstellungsmethoden an. So wird schnell deutlich, dass noch nichts ohne evtl. Zustimmung anderer genau geplant und beschlossen wurde, sonst ensteht vielleicht der falsche Eindruck, nichts mehr ändern zu können.

Zum Bereich der Stadtplanung gehört auch die Baugeschichte. Anstatt zukünftige Projekte darzustellen, können auch historische Begebenheiten rekonstruiert werden. Prinzipiell ist ein beliebiges Ändern des Parameters *Zeit* vorstellbar. Weitere Planungsgebiete, in denen ein 3D-Stadtmodell eingesetzt werden kann, sind u.a.

- Infrastrukturplanung und Stadttechnik
- Umweltplanung
- Verkehr, Logistik, Standorte für Unternehmen
- Stadterneuerung und -entwicklung
- Flächennutzung und Landschaftsplanung

2.5 Analysen und physikalische Simulationen

2.5.1 Funknetzplanung

Neben der eher visuellen Planung eines erfahrenen Funknetzplaners kann inzwischen auch interaktiv die Ausbreitung von Funkwellen simuliert werden. Die Platzierung der Antennen kann so optimiert werden (siehe Abb. 2.1).

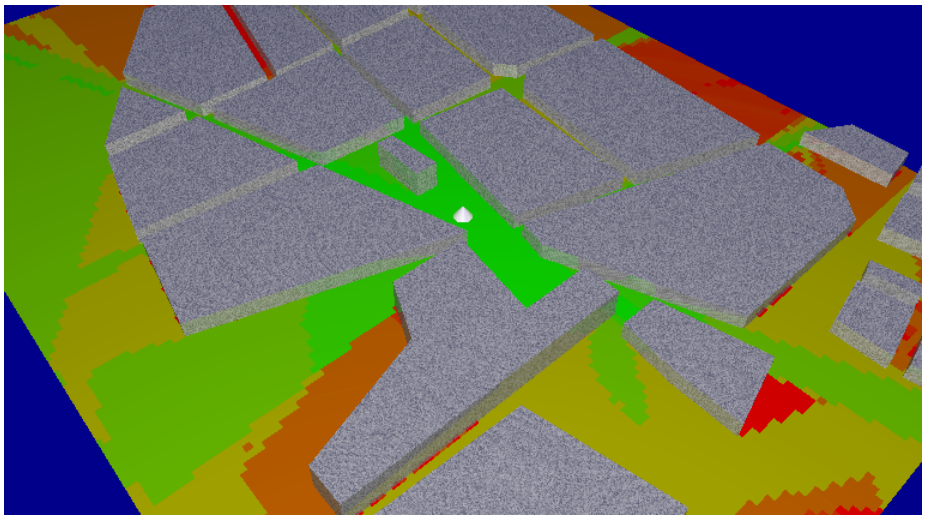

Abbildung 2.1: Simulationsergebnis einer Funkwellenausbreitung. Die Basisstation befindet sich an der Stelle des hellen Kegels in der Bildmitte. (Quelle: CARPET)

2.5.2 Überflutung

Hochwassersituationen können mit Hilfe der heutzutage sehr genauen DGMs recht gut simuliert werden (siehe Abb. 2.2).

Abbildung 2.2: Ausschnitt einer Überflutungssimulation von Houston (Quelle: TerraPoint)

2.5.3 Feuerausbreitung

Simulationen können präventiv durchgeführt werden, um an Schwachpunkten geeignete Maßnahmen zu ergreifen.

2.5.4 Gas- und Luftbewegung

In einer groben Skalierung können Verbreitung von Abgasen und die Luftqualität simuliert werden. Für kleinere Bereiche können z.B. für ein geplantes Projekt mögliche Turbulenzen untersucht werden (siehe Abb. 2.3).

2.5.5 Schallausbreitung

Die Untersuchung zum Lärmschutz ist wichtig, um Grenzwerte einzuhalten. Gegen zu hohe Schallimmission durch Straßenverkehrslärm müssen ggf. Lärmschutzmauern errichtet werden. Eine Simulation der Schallimmision durch Verkehrslärm zeigt Abb. 2.4.

2.6 Militär und Sicherheit

Für das Training von Sicherheitskräften können virtuelle Umgebungen benutzt werden. Hier können Situationen nachgestellt werden, die in der Realität zu gefährlich wären. Sie können beliebig oft wiederholt und deren Parameter variiert werden (siehe Abb. 2.5 und Abb. 2.6).

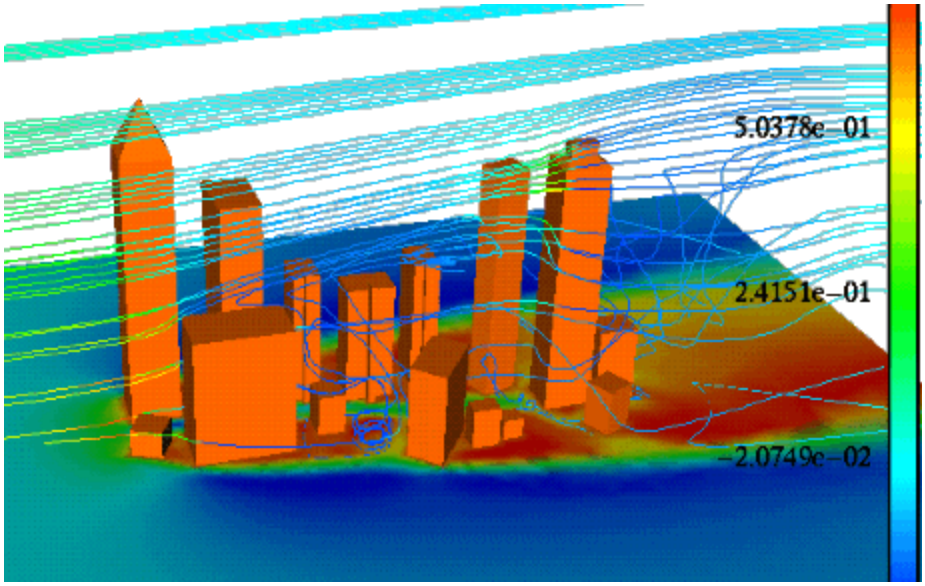

Abbildung 2.3: Luftdruck und Partikelpfade einer Luftströmungssimulation (Quelle: [Muh96])

Abbildung 2.4: Straßenlärmkarte, erstellt mit *CadnaA* (Quelle: Datakustik)

Für Einsatz- und Notfallplanungen bei Großveranstaltungen oder Katastrophen kann das Modell eingesetzt werden. Simulationen können hier zusätzlich helfen, die Rettungs- und Löscheinsätze zu koordinieren. Gerade bei größeren Katastrophen oder vielen gleichzeitigen Löscheinsätzen sind solche Systeme hilfreich, um vorhandene Ressourcen optimal verteilen zu können.
Für Rettungshubschrauber ist eine Landeplatzplanung vorstellbar. So könnte das Bodenteam z.B. schon eine Vorauswahl möglicher Positionen treffen und Hinweise und Bilder an die Piloten des Hubschraubers übermitteln.

Abbildung 2.5: Virtuelles Training zur Feuerbekämpfung (Quelle: Virtual Worlds Lab, Georgia Institute of Technology)

2.7 Multimedia

2.7.1 Spiele und Unterhaltung

Für die Visualisierung auf Consumer-PCs sind Beispiele aus der Spiele-Industrie besonders interessant. Aufwendige Erstellung von Modellen durch professionelle Künstler und Techniken wie Beleuchtungsberechnugnen werden eingesetzt, um sehr ansprechende Welten zu erzeugen und die Grafik-Hardware optimal auszunutzen. Die Modelle sind groß und detailreich, interaktionsfähig und bieten auch Innenansichten von vielen Gebäuden (siehe Abb. 2.7). Navigation und Bedienung sind ebenfalls sehr benutzerfreundlich.

2.7.2 Medien und Werbung

Fotorealismus ist hier besonders wichtig. Häufig werden die Modelle z.B. in Kinofilmen eingesetzt, um Special Effects zu realisieren. So können real nicht existierende Städte geschaffen werden. Oft ist aber auch die Nachbildung bekannter Großstädte von Bedeutung, um große Zerstörungen und Katastrophen darzustellen (siehe Abb. 2.8). Ebenfalls können Städte aus vergangener Zeit nachgebildet werden.

Abbildung 2.6: Szenario: Soldat in einem VR-Kampf. Auf Projektionswänden sind weitere, virtuelle Soldaten auf der linken Seite und die Stadt auf der rechten Seite zu sehen. (Quelle: U.S. Army Public Affairs)

Hierbei können auch Parameter wie Wetter, Tages- und Jahreszeit in die Visualisierung einfließen. Für Illustrationen in Zeitschriften und Zeitungen oder in den Fernsehnachrichten werden z.Z. noch sehr selten 3D-Stadtmodelle benutzt, doch in der Wettervorhersage vom Fernsehen gibt es schon kurze Animationen, bei denen ein Flug durch eine Region die vermuteten Wetterverhältnisse zeigt. Teilweise werden dafür auch schon sehr grobe Stadtmodelle benutzt. Für einige Sportübertragungen existieren 3D-Landschaftsmodelle z.B. einer Rennstrecke. Für eine Rallye durch die Alpen wurde beispielsweise ein sehr realistisch aussehendes Landschaftsmodell benutzt, um den Streckenverlauf und die aktuellen Positionen der teilnehmenden Fahrzeuge darzustellen (siehe Abb. 2.9).
Für Plakatwerbung ist die Sichtbarkeit der angebotenen Flächen entscheidend und kann für Werbekunden mit einem 3D-Stadtmodell einfach und bequem geprüft werden.

2.8 Andere Nutzungen

Weitere Anwendungen u.a. in den Bereichen Umwelt, Versorgung, Versicherung, Bauwesen, Architektur, Analyse, Geographie und Denkmalschutz sind möglich. Für manche Anwendungsideen muss erst noch gezeigt werden, ob diese überhaupt erfolgversprechend sind. *GeoSim Systems* schlägt z.B. das Stadtmodell als Plattform für eine Mensch zu Mensch - Kommunikation vor, bei der die

Abbildung 2.7: Screenshots eines aktuelles 3D-Spiels (Quelle: *Half Life 2*)

Abbildung 2.8: Naturkatastrophe in einer Großstadt im Film *The Day After Tomorrow* (Quelle: comingsoon.net)

Abbildung 2.9: Darstellung einer Rallye; im Hintergrund rechts ist auch ein Gebäude zu sehen (Quelle: VirtualSpectator)

Teilnehmer durch Avatare im Modell repräsentiert werden.
Eine Sekundäranwendung ist die Werbung innerhalb eines öffentlich benutzten 3D-Stadtmodells.

Kapitel 3

Existierende 3D-Stadtmodelle

In diesem Kapitel werden einige wenige Beispiele interessanter bereits existierender 3D-Stadtmodelle vorgestellt. Weitere Beispiele folgen in den nächsten Kapiteln.
GeoSim Systems unterhält u.a. das Projekt „Virtual Philadelphia“ und hat eine breite Palette von Anwendungsvorschlägen. Das gelungene User-Interface, das an die jeweilige Anwendung angepasst werden kann und die hohe Detailtreue, die sogar einen Vergleich mit einem Foto wie in Abb. 3.1 zulässt, sind hier besonders hervorzuheben.

Abbildung 3.1: Philadelphia: Rendering und Foto (Quelle: [Sys])

Aus dem Projekt TRIDENT (2000-2002), bei dem Modelle von Helsinki, Madrid und Rom entstanden sind, wurde ein Gebiet im Stadtbereich von Helsinki weiterentwickelt. Die Stadt Helsinki betreibt die Internetseite VirtualHelsinki, auf der man sich neben aktuellen und historischen Fotos und Panoramabildern auch ein VRML-Modell der Innenstadt und des Hafens ansehen kann. Die Darstellung ist trotz oder gerade wegen des Verzichts von Texturen gut gelungen (siehe Abb. 3.2).
Das VRML-Modell der Stadt Bremen hat graphisch nichts besonderes zu bieten, doch enthält es eine animierte Straßenbahn, die durch das Modell fährt.
Für die kleine Stadt Mühlacker ist mit Hilfe des Viscape-Plugins ebenfalls ein

Abbildung 3.2: Helsinki (Quelle: Virtual Helsinki [kSH])

Modell über das Internet erreichbar. Erwähnenswert ist eine animierte Einwohnerin. Insgesamt wirkt das Modell jedoch nicht sehr ansprechend.
Das Projekt München4D enthält auch die zeitliche Dimension. Die hervorragende Idee ist allerdings noch etwas enttäuschend realisiert, da man nur zwischen zwei Zeitpunkten wählen kann. Das optische Erscheinungsbild vom Modell von heute und von vor ca. 100 Jahren ist gut gelungen.
Modelle von Stuttgart und Heidelberg wurden eher zu Forschungszwecken erstellt. Abb. 3.3 zeigt aus LIDAR-Daten automatisch generierte Gebäude in der Schweiz.

Abbildung 3.3: Ein Dorf in der Schweiz (Quelle: Swissphoto Group)

3.1 Prozedural generierte Städte

Die vielen Probleme mit real existierenden Daten können vermieden werden, wenn die notwendigen Daten generiert werden. Einzelne Objekte oder auch komplette Städte können per Zufallszahlengenerator gebaut werden.

Die *Descensor Engine* [Wor] erzeugt ganze Landschaften und Städte bis hin zu Details wie Autos oder Innenräumen (siehe Abb. 3.4). Dabei können auch Parameter beeinflusst werden, um lokale Strukturen den eigenen Wünschen anzupassen. Ein anderes Beispiel ist in Abb. 6.4 auf S. 49 zu sehen.

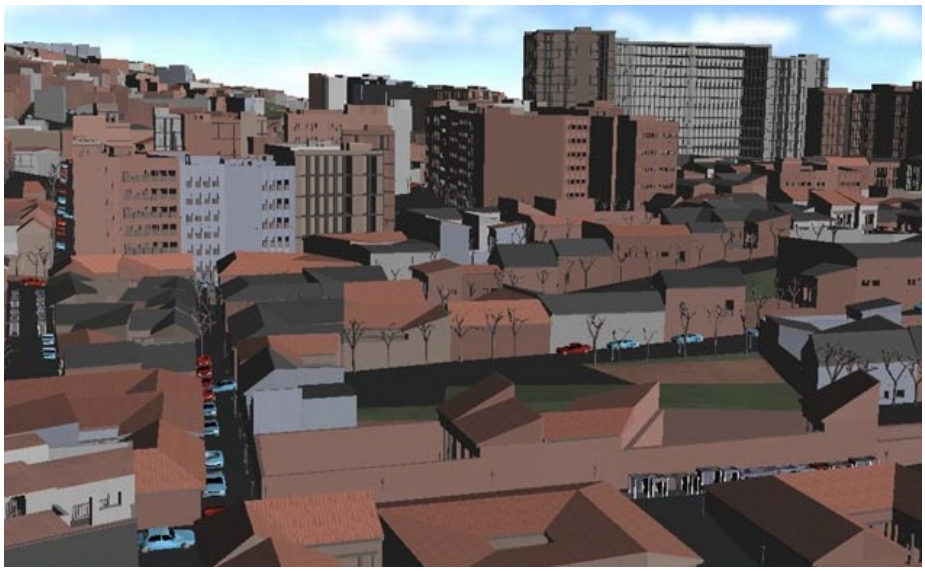

Abbildung 3.4: prozedural generierte Stadt (Quelle: Descensor Engine [Wor])

Kapitel 4

Datenquellen

Im Rahmen dieser Arbeit konnte für die flächendeckende Datennutzung aus Kostengründen nur auf bereits vorhandene digitale Daten zurückgegriffen werden, die von der Stadt Braunschweig zur Verfügung gestellt wurden. Objekte, die manuelle Bearbeitung erfordern, wurden teilweise exemplarisch für einen kleinen Bereich erstellt. In diesem Kapitel werden wichtige Datenquellen beschrieben, aus denen sich die gewünschten Informationen über die Objekte des Stadtmodells extrahieren lassen. Für raumbezogene Informationen kann man zwischen Rasterdaten und Vektordaten unterscheiden.
Bedauerlicherweise enden viele Datensätze oft unmittelbar an der Stadtgrenze. Dadurch kann das Modell in der Nähe der Stadtgrenzen für viele Anwendungen nicht oder nur stark eingeschränkt genutzt werden.
Manche Rasterdaten sind überwiegend zweidimensional, wie die Bodennutzungskarte. Brücken stellen hier ein echtes Problem dar, evtl. sollten für diesen Fall mehrere Layer angelegt werden. Orthofotos könnten immer die unterste Ebene enthalten, z.B. das Bild der Oberfläche unter einer Brücke. Die Brücke selbst wird als Bauwerk zusätzlich eingefügt und texturiert. In anderen LoDs sollte jedoch wieder die originale Textur des Luftbildes gezeigt werden, um bei großer Entfernung beispielsweise die Brückenmodelle weglassen zu können.
Beim LoD von Stadtmodellen handelt es sich um den Detaillierungsgrad bei der Datenerfassung. Grobe Modelle sind einfache Blockmodelle von Gebäuden, detailliertere Modelle enthalten genauer modellierte Gebäude mit korrekten Texturen und auch Pflanzen. LoD-Techniken zum Rendern werden in Abschnitt 6.2 besprochen.

4.1 Digitales Oberflächenmodell

Für die recht genaue Erfassung der Höhendaten gibt es z.Z. zwei wichtige Verfahren, die in den folgenden Abschnitten beschrieben werden. In beiden Fällen werden die Ausgangsdaten durch Befliegungen gewonnen. Satellitendaten sind zumindest für zivile Zwecke heutzutage noch nicht in ausreichender Auflösung verfügbar. Ein weiterer Ansatz wurde für ein Modell von Liverpool benutzt, bei dem ein bereits vorhandenes Holzmodell eingescannt wurde.
Ein übliches, sehr allgemeines Dateiformat enthält Koordinaten aller Punkte im ASCII-Format. Vorteilhaft sind die gute Portabilität und die Flexibilität,

da die Punkte weder auf 2,5D noch auf ein regelmäßiges Gitter beschränkt sind. Für rechtwinklige 2,5D Gitter wird auch das Rohdatenformat häufig benutzt, bei dem die Zahlen zeilenweise binärcodiert abgelegt werden. Für geringe Höhenauflösungen eignen sich auch gewöhnliche 8-Bit Graustufenbilder. Dank der großen Anzahl von vorhandenen Bildverarbeitungsprogrammen lassen sich diese besonders einfach bearbeiten.

4.1.1 Photogrammetrie

Flächendeckend vorhandene Stereobildpaare ermöglichen mit Hilfe von Korrelationsverfahren die automatische Erstellung eines DOM. Gleichzeitig können die Bilder zur Erzeugung von Orthofotos genutzt werden. In [Tri] wird eine Reihe von Problemen des Verfahrens aufgezählt. So können z.B. bei engen Straßen und hohen Gebäuden Teile der Erdoberfläche verdeckt werden. Auch ungünstige Belichtung, also sehr helle oder schattige Bereiche im Bild, können den Erfolg des Verfahrens negativ beeinflussen. Das Korrelationsverfahren hat zudem Schwierigkeiten bei regelmäßigen, selbstähnlichen Strukturen oder schwach strukturierten Oberflächen.
Vermutlich sind auch Wasserflächen problematisch. Mit Hilfe verschiedener Strategien und Annahmen, z.B. über die zu erwartenden Frequenzsprektren, können jedoch größtenteils korrekte DOMs erstellt werden.

4.1.2 LIDAR

Dieses oft auch als *airborne laser scanning* bezeichnete Verfahren gewinnt dank starker Verbesserungen der Hardware in den letzten Jahren immer mehr an Bedeutung. Die Auflösung ist inzwischen sehr hoch und viele der Probleme des photogrammetrischen Ansatzes sind nicht vorhanden. Gerade im Bereich von urbanen Bruchkanten bietet dieses Verfahren deutliche Vorteile (siehe Abb. 4.1).

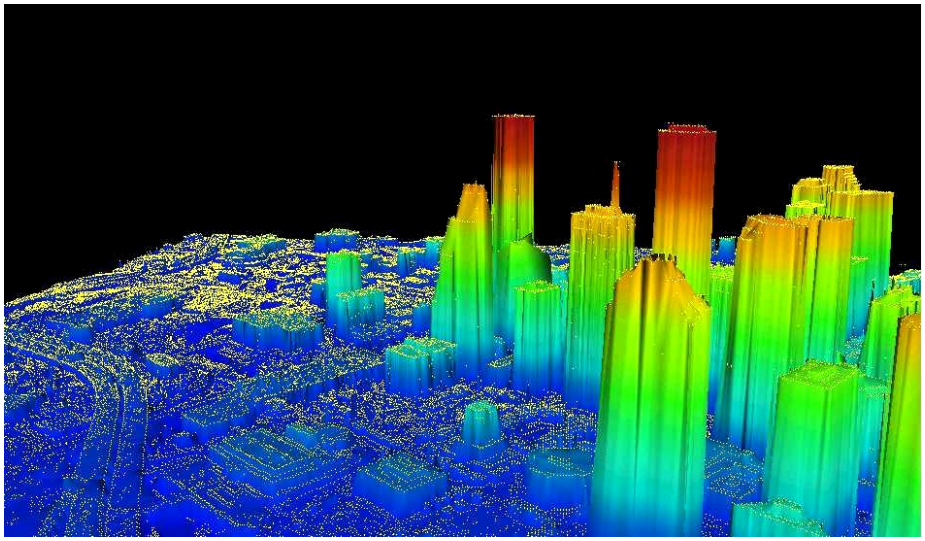

Abbildung 4.1: LIDAR-Scan von Houston (Quelle: TerraPoint)

Es ist sogar möglich, die erste und die letzte Reflexion aufzuzeichnen. Mehrfachreflexionen treten z.B. bei Baumkronen oft auf. Auch für die Erkennung von Hochspannungsleitungen ist das Mehrfachecho hilfreich.

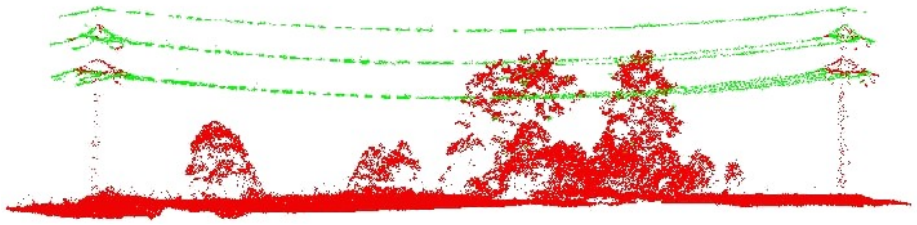

Abbildung 4.2: LIDAR-Scan einer Hochspannungsleitung (Quelle: TopoSys)

Ein leicht verrauschtes Signal trotz korrekt gemessener Punkte wird durch kleine Objekte wie z.B. Ampeln verursacht, die kleiner sind als das Abtastraster. Oft wird auch ein regelmäßiges 2,5D Gitter berechnet. Punkte, die beispielsweise durch leicht schräge Aufnahmen unter einem Dachüberhang liegen, führen hier zu Problemen. Mit den Daten können DGMs erstellt werden. Dazu werden vor allem morphologische Filter benutzt, um nicht zur Erdoberfläche gehörende Erhebungen wir Bauwerke und Pflanzen wegzufiltern. Prinzipiell kann das Morphologische Schließen benutzt werden, doch ist hier die Form des Strukturierenden Elements kritisch, dies zeigen auch eigene provisorische Tests. Eine adaptive Größenanpassung ist daher neben weiteren Maßnahmen unbedingt erforderlich (siehe Abb. 4.3).

Abbildung 4.3: Größenvergleich: Bei der Erstellung eines DGMs soll der Gaußberg (links) erhalten bleiben, das große Gebäude rechts jedoch weggefiltert werden.

Die Erstellung dieser DGMs mit Hilfe von DOMs wird meist von den Firmen angeboten, die auch die Akquisition der LIDAR-Daten durchführen.
Auch die Erkennung großer Wasserflächen ist möglich, vermutlich werden dafür große Flächen gleicher Höhe in lokalen Tiefpunkten gesucht.

4.2 Karten und Pläne

Häufig sind für Städte bereits zweidimensionale digitale Katasterpläne vorhanden, die zur Extraktion von Gebäudegrundrissen und weiteren Daten genutzt werden können. Leider gibt es kein allgemein benutztes Standardformat und die Daten werden oft nur zur 2D-Visualisierung benutzt, so dass eine direkte Benutzung zur automatisierten Verarbeitung nur eingeschränkt möglich ist. Praktische Probleme am Beispiel von Braunschweig werden genauer in Abschnitt 7.3.3 beschrieben.

4.2.1 Liegenschafts- und Katasterkarten

Die Digitalisierung erfolgt überlicherweise manuell mit Hilfe von Orthofotos und den bereits vorhandenen Papierkarten. U.a. wegen fehlender Unterstützung der Software wurde dabei in vielen Städten wie auch im Falle Braunschweigs leider eher auf visuelle Form als auf Semantik geachtet. Mit geringem Mehraufwand hätte anstatt einer digitalen Karte auch die für automatisierte Verarbeitung sehr wichtige Semantik eingegeben werden können.

4.2.2 Stadtkarte

Stadtkarten eignen sich wenig für die Benutzung eines 3D-Stadtmodells. Problematisch ist die Generalisierung. Straßen sind z.B. oft viel breiter eingezeichnet als in der Realität. Evtl. lassen sich manche Daten wie Beschriftungstexte für eine NPR-Visualisierung nutzen.

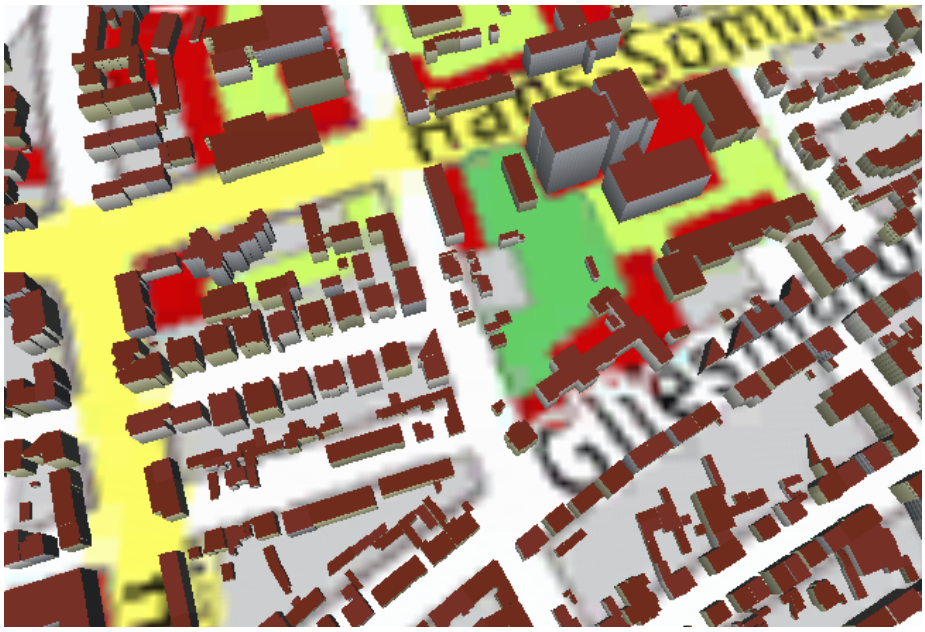

Abbildung 4.4: Die Nutzung einer Stadtkarte zusammen mit Gebäuden in unterschiedlichen Generalisierungsstufen ist nicht sinnvoll.

4.2.3 Bodennutzungskarte

Bodennutzungskarten sind oft anwendungsspezifisch und manuell erstellt. Es existieren jedoch auch Ansätze zur automatischen Klassifizierung mit Hilfe verschiedener anderer Daten, wie multispektralen Orthofotos und Höhendaten. Während letztere nur 2D-Rasterdaten sind, könnten bei manuell erstellten Vektorkarten prinzipiell für eine Position auch verschiedene Nutzungen zutreffen, z.B. für eine Brücke über einen Fluss. Abb. 4.5 zeigt Datenquellen und Ergebnis einer automatischen Klassifizierung von den Klassen 'Schatten', 'Gebäude', 'Baum', 'grasbedeckte Fläche' und 'Straße'. Dieses Verfahren stammt aus dem Gebiet der Mustererkennung.
Einen Ausschnitt der manuell erstellten Bodennutzungskarte von Braunschweig zeigt Abb. 4.6.

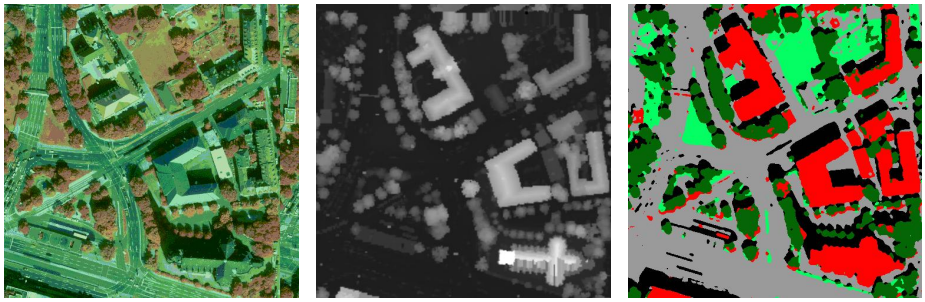

Abbildung 4.5: CIR-Bild, DOM und Ergebnis einer automatischen Klassifizierung (Quelle: [NH99])

Abbildung 4.6: Bodennutzungskarte mit Vektorinformationen

4.3 Fotos

Oft ist das sichtbare Spektrum von Interesse, doch weitere Spektren können für Klassifizierungsverfahren sehr hilfreich sein. Für Erkennung von Vegetation ist vor allem der nahe Infrarotbereich (NIR, Near InfraRed) von großer Bedeutung. Abb. 4.5 zeigt ein CIR-Bild (Color-InfraRed), das aus Rot-, Grün- und NIR-Kanal besteht und in falschfarben dargestellt wird.
Üblicherweise werden die Fotos bei sonnigem Wetter etwa zur Mittagszeit aufgenommen. Dadurch ist eine gute Belichtung sichergestellt, doch enthalten die Fotos auch Schatten, was je nach Anwendung Vor- oder Nachteile hat. Generell sind mehrere Aufnahmen zu unterschiedlichen Zeitpunkten oder aus anderen Blickwinkeln oft hilfreich. Hierbei sind gerade bei unterschiedlichen Bildquellen Farbanpassungen und weiche Übergänge notwendig, um Bilder nahtlos aneinanderzufügen.

4.3.1 Luftbilder

Orthofotos

Professionell erstellte Orthofotos werden aus vielen einzelnen Luftbildern zusammengesetzt, ohne dass stark sichtbare Kanten entstehen. Die Fotos werden so entzerrt, dass das Bild auf der Erdoberfläche genau den jeweiligen Koordinaten entspricht. Durch die Perspektive des recht tief fliegenden Flugzeugs sind gerade an den Randbereichen der zusammengefügten Fotos die Gebäude nicht mehr senkrecht projiziert (siehe Abb. 4.7).
Auch die Aufnahme mit Zeilenkameras ist üblich. Diese Bilder können gleichzeitig mit der LIDAR-Messung aufgenommen werden, um eine gute Übereinstimmung zu erhalten und Kosten mehrfacher Befliegungen zu sparen.

True-Orthofotos

Mit Hilfe eines hoch aufgelösten DOMs und etwas mehr Aufnahmen lassen sich recht einfach „echte“ Orthofotos generieren, bei denen eine exakt senkrechte Projektion generiert wird. So können die eben genannten Artefakte vermieden werden.

Schräge Aufnahmen

Um senkrechte Wände wie Fassaden mit den richtigen Bildern zu texturieren, können Luftbilder aus der „Schrägperspektive“ benutzt werden. Die Registrierung kann entweder durch Bediener erfolgen, die manuell mehrere Punkte anwählen müssen, oder mit Hilfe von GPS vorgenommen werden.

4.3.2 Terrestrisch aufgenommene Bilder

Es gibt Firmen, die mit Fahrzeugen alle Straßen einer Stadt abfahren und im Abstand von wenigen Metern Fotos in verschiedene Richtungen aufnehmen. Die Position der Aufnahmen wird mit GPS bestimmt und mit abgespeichert. Ähnliche Erfassungen mit hochauflösenden Videoaufnahmen sind denkbar. Die kombinierte Aufnahme von Bildern und Laserscan-Daten wurde schon an einzelnen Straßenzügen getestet. Dabei entstehen enorme Datenmengen, doch ist es schon

Abbildung 4.7: Artefakte von Orthofotos: *links:* Trotz guter Übereinstimmung des Grundrisses auf Höhe des Erdbodens ist das Bild des Daches um mehrere Meter verschoben; *rechts:* An der Grenze zwischen zwei zusammengefügten Fotos sieht man Hauswände aus unterschiedlichen Perspektiven und einen wiederholten Dachrand

möglich, die Erfassung im normalen Straßenverkehr vorzunehmen. Die Positionsbestimmung in diesem Beispiel wird mit Hilfe eines zweiten Laserscanners vorgenommen (siehe Abb. 4.8).

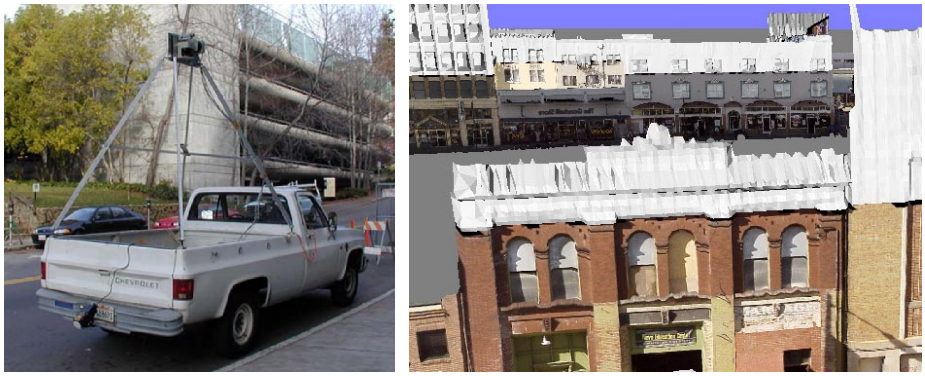

Abbildung 4.8: *links:* Aufnahmefahrzeug mit Laserscannern und Kamera. *rechts:* Ergebnisse. Wegen des beschränkten Sichtfeldes der Kamera ist der obere Teil der Fassaden nicht texturiert. (Quelle: Video & Image Processing Lab, Berkeley)

4.3.3 Satellitenfotos

Satellitenfotos sind für die ganze Erdoberfläche verfügbar. Momentane Fotos für nicht-militärische Zwecke haben jedoch noch eine geringe Auflösung von etwa

1m.

4.4 Andere Datenbanken

Alle raumbezogenen Datenbanken können genutzt werden. Im Folgenden werden einige Beispiele genannt.

Straßennamen und Hausnummern:
Für den Fall Braunschweig existiert eine Datenbank mit Hausnummern, die zusätzlich die Geoposition des Gebäudes und eine ID der Straße enthält. Eine weitere Datenbank enthält die Straßennamen der betreffenden IDs.

Lichtsignalanlagen:
Datenbanken über Verkehrsampeln können einige Attribute wie Position, Ausrichtung, Masthöhe, Hersteller- und Wartungsinformationen enthalten. Auch die Informationen über Steuerungsphasen, Straßen und Fahrspuren sind interessant.

Verkehrsnetze:
Daten über Verkehrsnetze können umfangreich sein und z.B. für Routenberechnungen oder Verkehrsflusssimulationen genutzt werden. Auch Informationen über Straßenbahnhaltestellen, Fahrpläne oder Geschwindigkeitsbegrenzungen für Straßen gehören dazu.

4.5 Manuelle Erfassung

Objekte können durch klassische Vermessung vor Ort aufgenommen und in das GIS eingegeben werden. Die Erfassung kann auch durch Erkennen und Markieren in Ortofotos erfolgen. Die Grenzen zwischen automatischer und manueller Datenakquisition sind fließend. Die manuelle Eingabe kann teilweise gut durch Algorithmen unterstützt werden. Andererseits sind je nach gewünschter Konfidenz der Daten Kontrollen durch Bediener und evtl. manuelles Eingreifen in einen automatischen Erkennungsprozess notwendig. Auch das Antrainieren neuronaler Netze mit Hilfe dieser Ergebnisse ist möglich, um automatische Verfahren zu verbessern.
Eine manuelle Eingabe von Daten ist manchmal mit verhältnismäßig wenig Aufwand zu bewerkstelligen. Allerdings kann die regelmäßige Aktualisierung viel Zeit in Anspruch nehmen. Handelt es sich um Objekte, die Mitarbeiter gezielt verändern, sollten auch diese die Aktualisierung der für sie zuständigen Daten durchführen.
Ist die Erfassung benötigter skalarer Daten gar nicht in einer hohen Genauigkeit nötig und auch zu aufwendig, ist es denkbar, eine grob approximative Rasterkarte zu erstellen, die z.B. als Bild graustufencodiert die Informationen enthält. Um ein ungefähres Gebäudealter für jedes Haus angeben zu können, könnte beispielsweise eine grobe Karte erstellt werden, die das bekannte Entstehungsdatum des jeweiligen Stadtviertels repräsentiert. Sind in manchen Stadtteilen die Informationen genauer bekannt oder sind diese Daten dort wichtiger, können leicht lokal Daten einer höheren Genauigkeit eingegeben werden.

4.6 Datenstrukturen

Für eine automatisierte Nutzung von Daten ist es essentiell, möglichst viele der bekannten semantischen Informationen zusammen mit der rein visuellen Gestalt anzugeben. Gerade im Hinblick auf die Finanzierung muss man sich vermutlich immer auf die gerade wichtigsten Informationen beschränken, doch der Mehrwehrt ist im Vorfeld oft nur schwer zu erfassen. Wird beispielsweise die zeitliche Dimension aller Objekte mit berücksichtigt, obwohl diese Daten auf den ersten Blick von geringer Bedeutung zu sein scheinen, kann mit ihnen das Modell auf einfache Weise für historische und planerische Zwecke genutzt werden. Mit dem gleichen Datenbestand kann so, abhängig von der gewählten Zeit, mit aktuellen Daten oder mit der momentanen Planung beispielsweise einer zukünftigen Autobahnbrücke gearbeitet werden. Für diese Überlegungen ist es jedoch zwingend notwendig, geeignete Programme einzusetzen, die so flexibel sind, die Anwender in dieser Aufgabe gut unterstützen zu können.
Als vor vielen Jahren die Katasterdaten für die Stadt Braunschweig digitalisert wurden, trat dieses Problem auf. Die Daten wurden vorrangig für die Anwendung der Visualisierung eingegeben. Erst durch meinen Versuch der automatischen Nutzung wurde es offensichtlich (siehe Abschnitt 7.3.3).
Für zukünftige 3D-Stadtmodelle werden wahrscheinlich weitere Daten manuell erfasst. Sinnvoll für die Visualisierung ist es, bestimmte Wahrzeichen detaillierter zu modellieren, als es heutige großflächige automatische Verfahren schaffen. Momentan scheint nur die Oberflächenrepräsentation dieser Objekte von Bedeutung zu sein. Doch genau hier könnte sich in 10 Jahren wieder ein Problem zeigen, wenn festgestellt wird, dass diese Modelle kaum weiter benutzt werden können, da z.B. die Oberflächen nicht geschlossen sind und keine Entscheidung getroffen werden kann, ob sich ein beliebiger Punkt innerhalb oder außerhalb des Objekts befindet. Dies könnte z.B. für Kollisionsdetektionen, Erstellung von Voxelräumen oder physikalische Simulationen von Bedeutung sein. Ebenfalls denkbar sind neue Renderingverfahren oder -effekte, für deren Nutzung mehr semantische Informationen sehr hilfreich sein könnten. So sollten z.B. beim Modellieren Glasfenster speziell mit dem Attribut *Glas* markiert werden, anstatt einfach ein semitransparentes Material zu benutzen. Wird eine modellierte Tür als solche markiert und mit weiteren Informationen über diese Tür versehen, ist es denkbar, diese Daten z.B. für eine spätere Simulation von Fußgängern zu benutzen. Beim Erstellen ist der Zusatzaufwand für das Hinzufügen dieser Informationen relativ gering und ermöglicht multiple Nutzung. Nachträglich wären solche Informationen wesentlich aufwendiger nachzubessern. Es ist ebenfalls möglich, dass einige öffentliche Gebäude in Zukunft auch von Innen modelliert werden sollen. Es wäre sinnvoll, ein Modell so flexibel zu gestalten, dass es später für diese Zwecke genutzt werden kann, und nicht wieder ein von Grund auf neues Modell erstellt werden muss. Für die jeweilige Anwendung können dann die relevanten Informationen genutzt bzw. exportiert werden. Die konkreten später benötigten Informationen sind andererseits nur schlecht vorherzusagen. Ein guter Kompromiss zwischen der Aufwendung von finanziellen Mitteln und dem Weitblick für zukünftige Nutzungen muss gefunden werden.
Eine allgemein anerkannte Repräsentation der betreffenden Daten existiert nicht einmal für den 2D-Fall. Für die dritte Dimension sind die Anforderungen noch viel komplexer. Selbst für die visuelle Beschreibung der Oberfläche gibt es keine optimale Lösung.

Eine auf XML basierende Lösung scheint am sinnvollsten, da diese Art der Datenbeschreibung universell, flexibel und einfach zu benutzen ist. So könnten z.B. auch je nach Bedarf unterschiedliche Daten über die gleichen Gebäudebestandteile abgespeichert werden. Der Grundriss, First- und Trauflinien oder die Dachfläche eines Gebäudes können je nach Anwendung verschiedene Definitionen haben und daher jeweils für die unterschiedlichen Nutzungen abgespeichert werden. Wichtig ist auch die einfache Erweiterbarkeit der Formate für neue Objekte oder Eigenschaften. Gebäude, Straßen usw. bekommen eine ID, die dann für alle Datenbanken verbindlich ist. Dieses Konzept ist universell und dürfte von allen Datenbanken unterstützt werden. Ein Teil der ID könnte die grobe Geoposition codieren, wodurch sich manche Suchanfragen beschleunigen lassen. Das *Open GIS Consortium (OGC)*[Conb] bemüht sich, die Standardisierung der GIS-Daten voranzutreiben. Dort gibt es viele Dokumente mit Spezifikationen und Richtlinien, wie Spezifikationen geschrieben werden sollen. Die *Geography Markup Language (GML)* [Cona] gibt es derzeit in der dritten Version und wird vermutlich immer mehr an Bedeutung gewinnen. Mit ihr lassen sich diverse Arten von Objekten mit Attributen, Koordinatensystemen, Topologie, Zeit und Maßeinheiten beschreiben.

Kapitel 5

Objekte im Stadtmodell

Prinzipiell könnten alle Objekte in oder in der Nähe einer Stadt auch in das entsprechende Stadtmodell integriert werden. Als Objekt im weiteren Sinn wird alles bezeichnet, was zum Modell gehört und abgespeichert oder angezeigt wird, wie z.B. die Erdoberfläche. In der Praxis kann nur ein Teil der Objekte in das Modell aufgenommen werden, wobei deren Größe normalerweise ein gutes Indiz für die Wichtigkeit ist. Besonders wichtige Objektgruppen werden in diesem Kapitel vorgestellt.
Von großer Bedeutung sind die Erfassung bzw. der Import der Objekte, deren Repräsentation im Speicher und eine geeignete Technik für deren Visualisierung, einschließlich dem Erzeugen von LoDs.
Je nach Anwendung und Anforderungen können nicht bekannte Objekte oder Objekteigenschaften mit Hilfe der vorhandenen Daten plausibel generiert werden. Dies kann beispielsweise Grashalme und Pflanzen oder Detailtexturen beinhalten. Es ist möglich, einige dieser Details während der Laufzeit nur an den benötigten Stellen zu erzeugen. Auch ganze Landschaften und Städte können ausschließlich prozedural generiert werden (siehe Kapitel 3.1). Unter Benutzung möglichst vieler bekannter Daten können diese Details sehr plausibel erzeugt werden. Allerdings ist darauf zu achten, die Anwender entsprechend über die eingesetzten Techniken zu informieren, um nicht zu große Erwartungen zu erwecken und Enttäuschungen zu vermeiden.
Neben brep-Darstellungen sind auch z.B. csg oder bildbasierte Techniken vorstellbar. Einigen der unten genannten Objekte können auch u.a. akustische oder haptische Eigenschaften zugeordnet werden. So könnten beispielsweise Verkehrsgeräusche oder Glockenläuten abgespielt werden.

5.1 Erdoberfläche und Umgebung

Erstaunlicherweise wird bei einigen professionellen Lösungen die Darstellung der Umgebung stark vernachlässigt. Dies kann daran liegen, dass dafür andere Techniken notwendig sind. Statische Objekte, die sich immer in großer Entfernung befinden, können einfach durch eine Skybox realisiert werden. Dies ist ein großer Würfel, der die gesamte Szene umschließt und auf dessen Wänden eine Textur der Umgebung liegt. Hügel und Berge, bei vielen Anwendungen auch Sonne, Atmosphäre und Wolken können hier realistisch dargestellt werden. Um

Verzerrungen zu vermeiden, kann der Mittelpunkt des Würfels immer auf die aktuelle Kameraposition gelegt werden.
Um eine Reflexion an der Wasseroberfläche zu simulieren, wurde die Skybox am Horizont gespiegelt und unten mit der Farbe des Wassers getönt. Ein senkrechter Blick nach unten zeigt also die Wasserfarbe wie auf einem Luftfoto, ein Blick ins Wasser in Richtung Horizont scheint den Himmel zu reflektieren. An Stellen, wo sich Wasser befindet, ist die Textur semitransparent und der gespiegelte Teil der Skybox ist sichtbar, wodurch der gewünschte Eindruck entsteht. Diese einfache und schnelle Methode ist noch sehr verbesserungsbedürftig, aber schon wesentlich besser, als die Reflexion komplett zu ignorieren. Weder Wellen noch gespiegelte Objekte werden dargestellt, doch lassen sich Wasserflächen so bereits deutlich besser erkennen. Diese Technik wurde u.a. eingesetzt, da weder der Grundwasserpegel noch das DGM unterhalb der Wasseroberfläche bekannt sind. Wünschenswert sind getrennte Informationen über Erdoberfläche ohne Wasser und die Wasseroberfläche. Damit kann eine wesentlich realistischere Darstellung der Wasseroberfläche erfolgen. An der Wasseroberfläche können auch andere Objekte gespiegelt nochmals dargestellt werden, um mit OpenGL eine Reflexion zu simulieren. Mit Hilfe erweiterter Features wie Pixel-Shadern können auch Wellen überzeugend dargestellt werden. Wasser von schnell fließenden Gewässern und Wasserfällen können mit animierten Texturen oder Partikelsystemen realisiert werden.

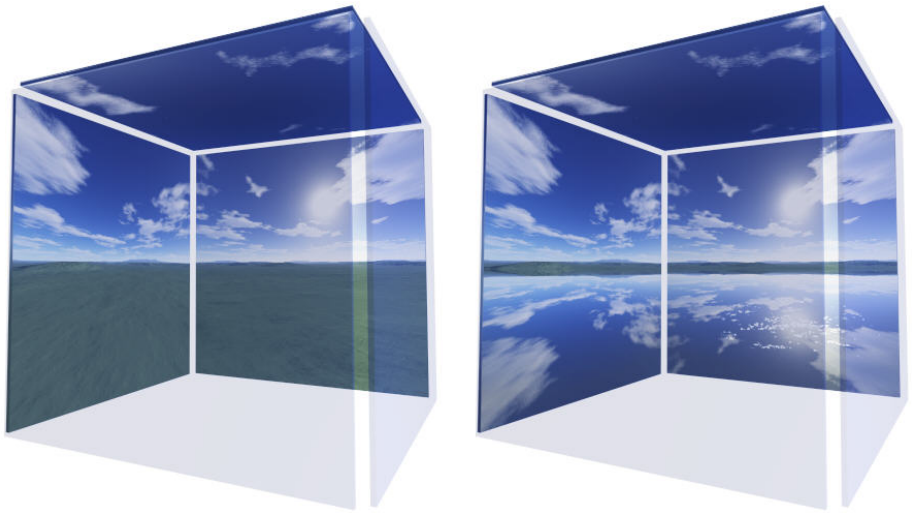

Abbildung 5.1: Die für Braunschweig erstellte Skybox: Normal und mit reflektierender Wasseroberfläche

Selbst bei großem Öffnungswinkel der Kamera muss die Auflösung der Texturen jedoch relativ hoch sein, da immer nur ein kleiner Ausschnitt auf voller Bildschirmgröße sichtbar ist. Vor allem bei der Landschaft am Horizont fällt dies auf, ein zusätzliches Texturlayer in höherer Auflösung nur für diesen Streifen ist sinnvoll. Für eine realistische Darstellung animierter Wolken oder für einen Flug durch Wolken existieren andere Techniken. Ebenso sind für Simulation sich ändernder Tages- oder Jahreszeiten weitere Techniken notwendig.

Für die Darstellung der Erdoberfläche sind vor allem Höhendaten und Luftfotos wichtig. Die Erdoberfläche soll in der Nähe des Betrachters sehr detailliert dargestellt werden, doch auch die größeren Erhebungen einige Kilometer entfernt sollen sichtbar sein. Bei den heutzutage verfügbaren recht genau aufgelösten DGM's wird schnell deutlich, dass LoDs hier unverzichtbar sind. Allerdings ist das Problem nicht trivial. Es gibt viele Ansätze für Terrain Engines wie z.B. ROAM[Ducb], die abhängig von der Krümmung der Landschaft die Triangulierung vornehmen. Im Gegensatz zu statischen Ansätzen wird es bei einer dynamischen Triangulierung zur Laufzeit zusätzlich möglich, den Blick des Betrachters mit einzubeziehen. Für eine gute Ausnutzung der Hardware werden üblicherweise Triangle Strips erzeugt. Herausforderungen sind die Vermeidung von Lücken (cracks) an LoD-Grenzen und von Popping-Artefakten beim Wechsel der LoDs. Gegen letzteres hilft die Technik des Geomorphings, bei der die Vertices der höheren Auflösung während eines Übergangs langsam in Richtung des interpolierten Wertes der gröberen Auflösung verschoben werden. Bei Verwendung von Texturkacheln muss aus Geschwindigkeitsgründen auch die Oberfläche an den gleichen Grenzen gekachelt werden. Übliche Techniken arbeiten meist nur mit 2,5D-Höhenfeldern und können weder Löcher noch Tunnel bzw. Höhlen o.Ä. darstellen. Allerdings kann dieses Problem auch mit transparenten Stellen in der benutzten Textur umgangen werden.
Für eine detaillierte Modellierung sollten aber auch Bordsteinkanten in das Erdoberflächennetz integriert werden können. Auf nicht versiegelten Flächen können Details fraktal generiert werden, um die Oberfläche etwas mehr zu strukturieren und für ein natürlicheres Erscheinungsbild sorgen. Ein anderes Problem tritt vor allem im Zusammenhang mit Gebäuden auf: Bei einem nicht ebenen Erdboden stehen Teile des Gebäudes etwas in der Luft oder tauchen in den Boden ein. Ein simpler Lösungsansatz ist die Bildung von Plattformen im DGM, so dass der komplette Grundriss auf ebenerdig gleicher Höhe liegt.
Anstatt der Erdoberfläche können auch direkt die LIDAR-Messpunkte bzw. das mit ihnen regelmäßig erzeugte Gitter angezeigt werden (siehe Abb. 5.2).

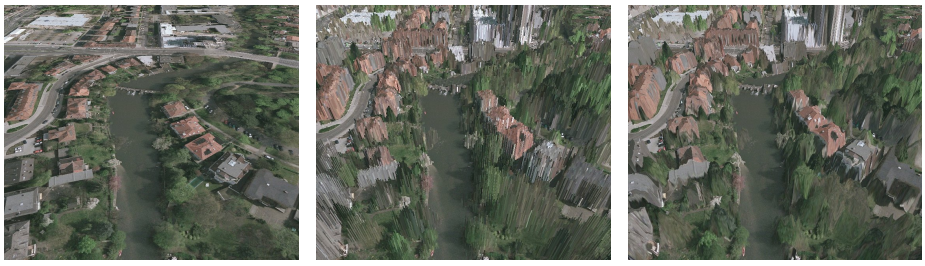

Abbildung 5.2: Nur DGM, DGM und DOM als senkrechte Linien, DOM als Oberfläche

Bei naher Betrachtung der Erdoberfläche wirken selbst relativ hoch aufgelöste Texturen unnatürlich und unscharf. Mit Hilfe einer Detailtextur kann mit Multitexturing effizient in einer anderen Skalierung die Helligkeit variiert werden, um der Oberfläche eine Struktur zu verleihen (siehe Abb. 5.3). Die ursprüngliche Idee, unterschiedliche Detailtexturen abhängig des jeweiligen Bodentyps zu wählen, eignet sich in der vorliegenden Implementierung aus Geschwindigkeitsgründen jedoch schlecht für eine interaktive Darstellung.

Abbildung 5.3: Ohne und mit DetailTexturing

Um die Höhenstrukturen der Landschaft zu verdeutlichen, können die Flächen mit Hilfe einer Beleuchtungsgleichung eine unterschiedliche Helligkeit erhalten. Bei der Texturierung mit Luftfotos ist dies nicht notwendig, da die richtigen Helligkeiten bereits in der Textur enthalten sind. Für die Texturierung mit Plänen und Karten kann dies jedoch sinnvoll sein (siehe Abb. 5.4).

Abbildung 5.4: 3D-Ansicht des Top50-Viewers, 2-fache Überhöhung (Quelle: Landesvermessung und Geobasisinformation Niedersachsen)

Bei Verwendung der Luftfotos gibt es aber auch Probleme. Die aus der Vogelperspektive gut wirkenden Texturen auf der Erdoberfläche wirken gerade bei sich ändernden Blickpositionen mit Blickrichtung zum Horizont sehr unnatürlich, da alle Objekte offensichtlich flach auf dem Boden kleben. Es ist wünschenswert, Autos und Bäume möglichst aus der Textur zu entfernen und durch echte 3D-Objekte zu ersetzen. Eine Lösungsidee ist die Kombination mehrerer Luftbilder zu unterschiedlichen Zeitpunkten. Dadurch können z.B. einige Autos entfernt werden. Bäume sind viel schwieriger zu entfernen, doch Laubbäume beispielsweise würden auf Fotos aus dem Winter weit weniger störend auffallen. Die Nutzung mehrerer Aufnahmen bietet viel Potenzial, ist jedoch auch teuer und aufwendig. Unerwünschte Objekte können auch durch Image Completion Algorithmen gut automatisch entfernt werden, doch müssen dazu die zu ersetzenden

Regionen recht genau bekannt sein. Die automatische Bestimmung dieser Regionen ist ein großes Problem. Abb. 5.5 zeigt ein Beispiel der Image Completion des Resynthesizer Filters von GIMP, der trotz der kleinen Region recht viel Zeit zur Berechnung benötigte.

Abbildung 5.5: Original und automatisch ersetze Regionen, die hier manuell markiert wurden

Zudem können Höhenlinien, anstatt sie direkt in der Textur zu speichern, auch per Multitexturing und automatischer Texturkoordinatengenerierung eingeblendet werden (siehe Abb. 5.6). Auch können so verschiedene Höhenschichten bunt eingefärbt werden. Allerdings führen bei dieser Methode die LoD-Techniken der Oberfläche zu ungewünschten Artefakten.

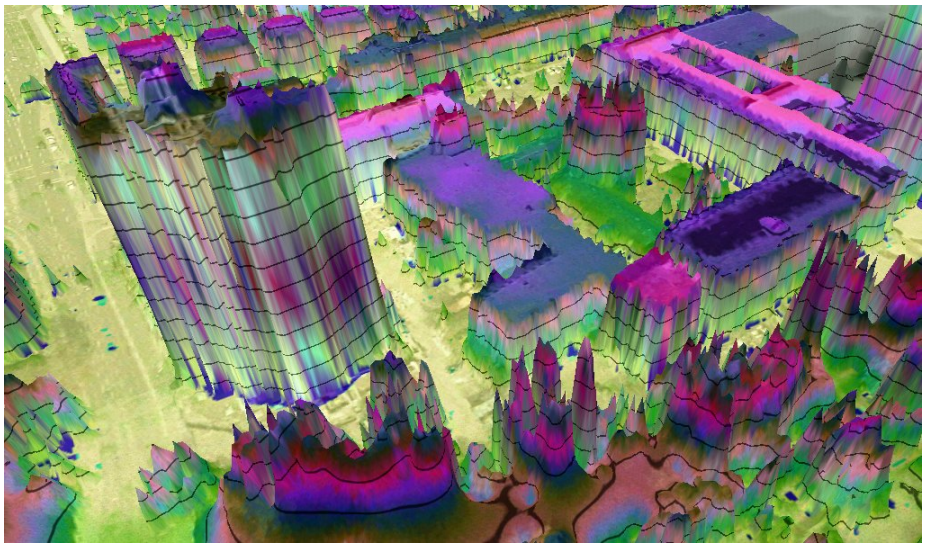

Abbildung 5.6: Textur: Kombination aus Luftfoto und normiertem DOM sowie per Multitexturing zusätzlich eingeblendeten Höhenlinien

Prinzipiell können alle vorliegenden Rasterdaten als Textur der Erdoberfläche

angezeigt werden. Abb. 5.7 zeigt so ein Beispiel. Ebenso können 2D-Vektordaten auf der Erdoberfläche angezeigt werden, indem einfach jeweils die Höhe der Erdoberfläche als dritte Ordinate benutzt wird (siehe Abb. 5.8).

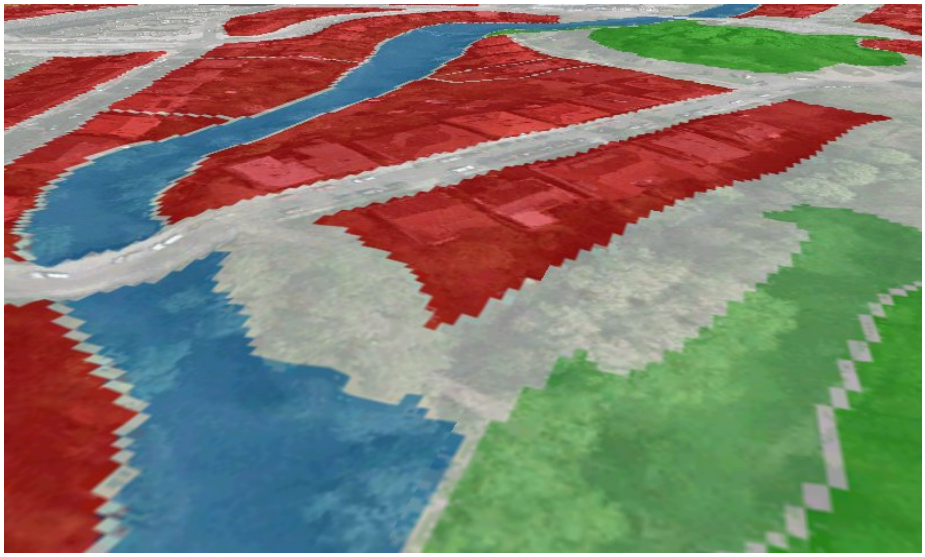

Abbildung 5.7: Textur: Kombination aus Luftfoto und grob aufgelöster Bodennutzungskarte

5.2 Bauwerke

Vor allem in den letzten 5 Jahren sind viele Publikationen zur semiautomatischen oder vollautomatischen Gebäude-Modellierung entstanden. Gebäude sind für viele Anwendungen der wichtigste Bestandteil des Modells. Für viele Einzellösungen werden Gebäude in einem kleinen Gebiet mit hohem Aufwand manuell nachmodelliert, doch für ganze Städte ist der Aufwand mit dieser Methode zu groß. Terrestrische Laserscanner können diesen Prozess unterstützen. Wegen der hohen Anzahl der Gebäude ist eine größtmögliche Automatisierung wünschenswert. Dachgauben, überhängende Dächer, Balkone, Treppen zu Eingangstüren usw. sind schwer automatisch zu erfassen. Viele weitere Informationen z.B. über Keller, Fundamente und Versorgungsanschlüsse können nur manuell angegeben werden.

5.2.1 Umrisse und Dachformen

Existieren als Datenquellen für die Gebäude nur Grundrisse und Stockwerkszahl, können keine korrekten Dachformen erstellt werden. Üblicherweise wird in diesem Fall ein sogenanntes Blockmodell generiert, bei dem alle Gebäudemodelle mit flachen Dächern versehen werden. Die Gebäudehöhe wird dabei mit Hilfe einer durchschnittlichen Stockwerkshöhe geschätzt, bei der auch das Gebäudealter mit einbezogen werden kann. Besser gelingen die Rekonstruktionen, wenn

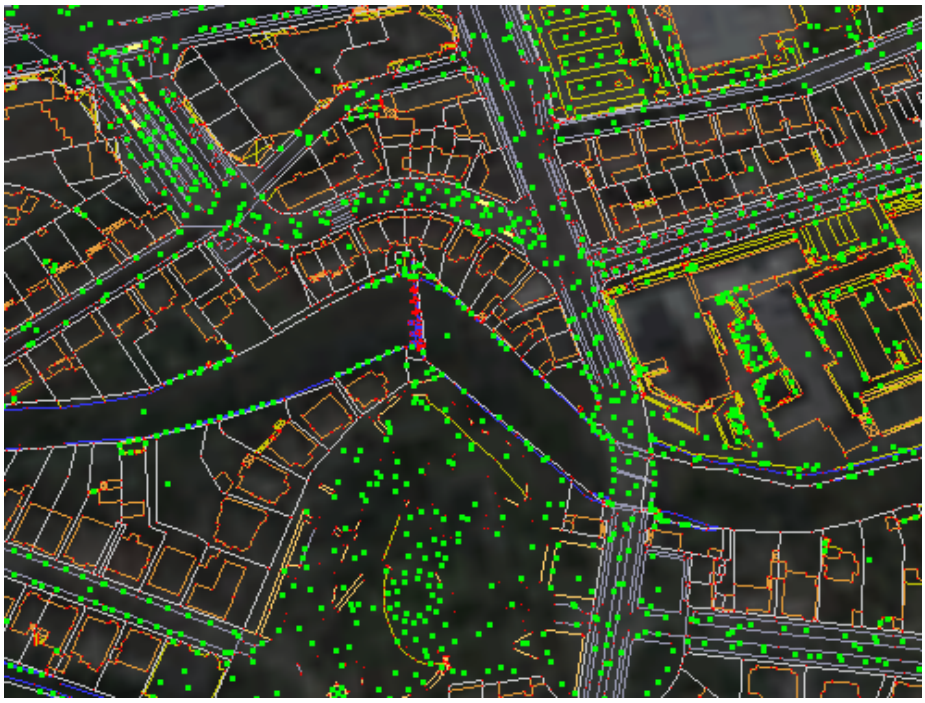

Abbildung 5.8: Abgedunkeltes Luftbild und CAD-Daten

LIDAR-Daten oder photogrammetrische Informationen verfügbar sind. Zur Rekonstruktion von Dachflächen existieren verschiedene Ansätze.
Gebäude können prinzipiell auch ohne Grundrissdaten nur mit Hilfe des DOMs erkannt werden, doch ist diese Methode noch recht fehleranfällig. Vorhandene Grundrisse stellen eine wesentliche Verbesserung dar, da es sich hier um manuell erstellte, bereits generalisierte Daten handelt, die eine hohe Konfidenz haben.
Für die Erkennung von Dachformen gibt es verschiedene Strategien. Beim parametrischen Ansatz wird von wenigen generischen Dachformen ausgegangen und versucht, die optimalen Parameter für die jeweils am besten passende Form zu schätzen. Je weniger Dachformen zugelassenen werden, desto robuster ist das Verfahren (siehe Abb. 5.9).
Der überwiegende Teil mitteleuropäischer Hausdächer kann gut mit mehreren ebenen Dachstücken approximiert werden. Die naheliegende Idee, die ersten beiden Gradienten der Höhenwerte zu benutzen und mittels Hough-Transformation die Grenzen der verschiedenen ebenen Regionen zu finden, führt zu ungenauen und fehleranfälligen Ergebnissen. Besser ist es, die Gleichungen dieser Ebenen zu bestimmen und für die Schnittkantenbestimmung zu nutzen. Die Eliminierung von Übersegmentierung, doppelten Kanten oder sehr nah aneinander liegenden Eckpunkten ist nötig. Auch die Zuhilfenahme bestimmter Annahmen wie der Rechtwinkligkeit führt zu guten Ergebnissen.
Auffällig bei den genannten Verfahren ist die Tatsache, dass Luftfotos oft nur für die visuelle Kontrolle genutzt werden. Die viel höhere Auflösung der Fotos könnte aber benutzt werden, um Kanten genauer zu positionieren, da sich bei Höhenkanten üblicherweise auch die Farbwerte im Bild sprunghaft ändern.

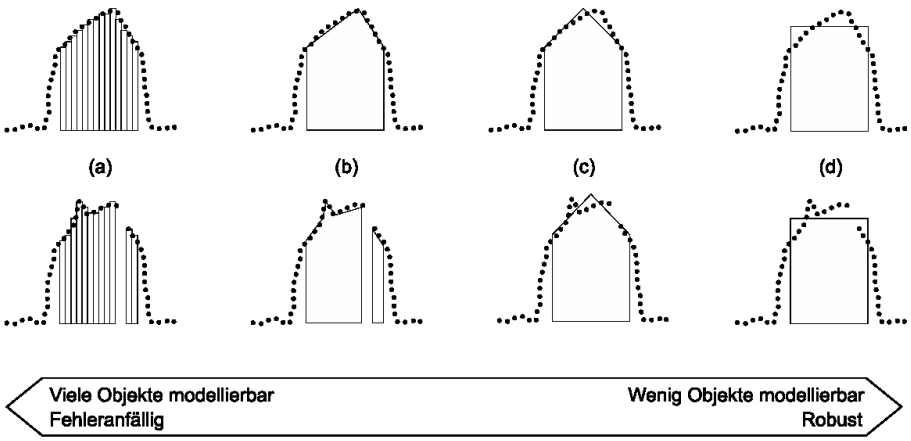

Abbildung 5.9: Vergleich von polygonalen und parametrischen Modellen (Quelle: [Bre00])

Eventuell können sogar Helligkeitsinformationen der Dachregionen genutzt werden, um die Neigung robuster zu bestimmen. Auch Dachüberhänge über den Grundriss können mit den Luftfotos besser bestimmt werden.
[Bre00] bietet einen guten und detaillierten Überblick über diverse vorhandene Verfahren und beschreibt auch bekannte existierende Programme zur semiautomatischen Erstellung von Dachformen.

Abbildung 5.10: Ergebnis einer automatischen Gebäuderekonstruktion (Quelle: [Sha00])

Ein weiterer Ansatz, der einfach zu implementieren ist, benutzt direkt die im Grundriss enthaltene Oberfläche zur Erstellung eines Netzes, das bereinigt und anschließend mit den üblichen Verfahren simplifiziert wird. Genauigkeit und

Abbildung 5.11: Heidelberg: Automatisch rekonstruierte Gebäude mit Luftfotos und teilweise manueller Fassadentexturierung (Quelle: [Bre00])

Qualität sind hier jedoch nicht sehr gut. Kritisch sind u.a. Interpolationen der Oberfläche nahe der Hauswände. So können fälschlicherweise fast senkrechte Dachflächen an den Stellen entstehen, wo sich in der Realität die Hauswand befindet. Die Punkte in der Nähe der Dachkante, also meist die Punkte in der Nähe des Grundrisses, sind wegen der Vorverarbeitung und Interpolation unzuverlässig und sollten nicht benutzt werden.

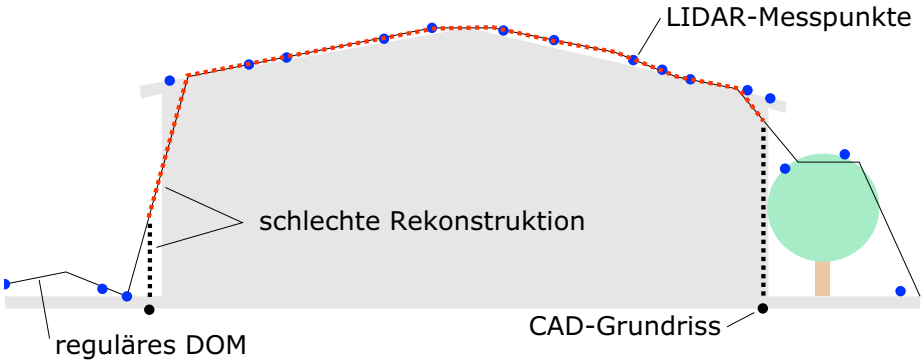

Abbildung 5.12: Einfache Verfahren führen zu schlechten Rekonstruktionsergebnissen.

Falls True Orthophotos zur Verfügung stehen, können die Dächer mit dieser Textur angezeigt werden. Allerdings wirken nicht-texturierte Fassaden bei diesem direkten Vergleich sehr störend.

5.2.2 Fassaden

Für die Texturierung von Fassaden können schräg aufgenommene Luftbilder benutzt werden. Der Vorteil gegenüber terrestrisch aufgenommenen Bildern ist hier, dass Verdeckungen viel seltener vorkommen. Andernfalls müssen viele Aufnahmen wie Video-Sequenzen benutzt werden, damit in manchen Bildern verdeckte Bereiche abgedeckt werden. Bei terrestrischen Aufnahmen können gleichzeitig Laserscanner benutzt werden, um auch die Geometrie der Fassaden aufzunehmen.
Wie in [FZ03] beschrieben, ist für Ansichten aus der Fußgängerperspektive auf Straßenniveau ein rein aus der Luft gewonnenes Modell zu ungenau. In diesem Bericht wird ein Verfahren vorgeschlagen, in dem Aufnahmen aus der Luft und vom Boden kombiniert werden. Dazu werden schräg aufgenommene Luftbilder, ein DOM sowie von einem Fahrzeug aufgenommene Bild- und Laserdaten jeweils aufgenommen, vorverarbeitet und dann kombiniert (siehe Abb. 5.13).

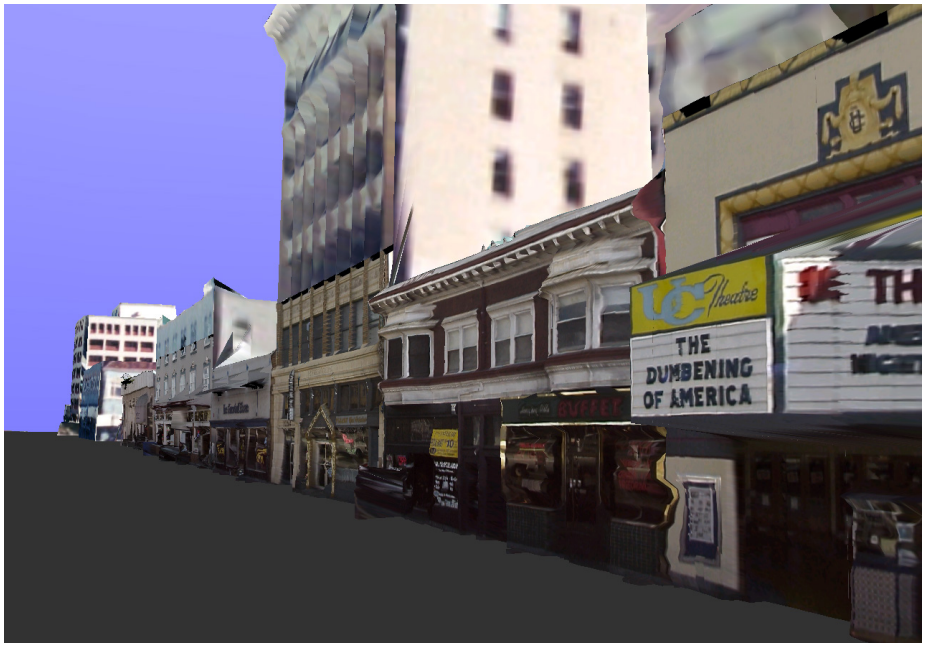

Abbildung 5.13: Automatisch verknüpfte, terrestrisch und luftgetragen aufgenommene Daten (Quelle: [FZ03])

Auch hier könnte ähnlich wie bei den Dachformen anstatt der Oberflächenbeschreibung versucht werden, ein parametrisches Modell automatisch zu extrahieren, in dem Positionen von Fenstern und Türen erkannt werden.
Die Geometrie kann auch ohne Laserscanner nur aus Videoaufnahmen erzeugt werden, wie in [Pol99] beschrieben.
Ohne viele geometrische Details kommt ein bildbasierter Ansatz aus, bei dem abhängig vom Blickwinkel verschiedene Texturen benutzt werden [Deb96].
Stehen für eine korrekte Texturierung der Fassaden keine Daten oder Ressourcen bereit, können diese aus einer kleinen Menge von Standard-Fassadentexturen ausgewählt werden. Viele Informationen können benutzt werden, um plausible

Texturen auszuwählen. Das Verhältnis von Gebäudehöhe zur Grundfläche, das Gebäudealter und umgebende Bodennutzungsdaten sind solche Merkmale. Diese Technik wird vor allem auch in Computerspielen eingesetzt. Dort sieht man, das es sich manchmal lohnt, für das Erdgeschoss eine weitere, spezielle Textur zu benutzen. Die Fassade wirkt so gleich viel realistischer als mit nur einem sich wiederholenden Bild.
Genau wie bei der Detailtextur können auch zufällige Geometriedetails hinzugefügt werden. So ist vorstellbar, dass eine Hauswand mit Fensterbänken und Balkons ausgestattet wird, um eine schönere Visualisierung zu erhalten.
Es gibt sogar den Ansatz, Fassadenstücke mit Hilfe von komplexen Grammatiken zu generieren, um plausible Darstellungen zu erzielen (siehe [WWSR03]).
Bei interaktiven Darstellungen werden Fenster normalerweise mit einer normalen, diffusen Textur oder mit semitransparenten Flächen gerendert. Reflexionen sind für den Fotorealismus gerade bei Glasfassaden wichtig. Bei einem Standbild ist das Benutzen eines Fotos ausreichend, doch wirkt dies bei bewegter Kamera eher unnatürlich. Besser aussehende Verfahren sind meist sehr zeitaufwendig, auch hier hilft das View-Dependent Texture Mapping.

5.2.3 Metadaten

Weitere sinnvolle Informationen über Gebäude sind die Nutzungsart, die Hausnummer, der Straßenname und Informationen über die Eigentümer und Bewohner. Durch Verknüpfung mit weiteren Datenbanken ist es vorstellbar, auf diese Weise weitere Informationen zu erhalten, wie z.B. die Kontaktmöglichkeit zu Bewohnern oder Weblinks zu privaten oder geschäftlichen Webseiten. Eine Gewinnung der Daten z.B. aus Telefonbuchdatenbanken ist aus Datenschutzgründen vermutlich nicht erlaubt, doch denkbar ist die Aufnahme freiwilliger Angaben.

5.3 Wahrzeichen und Kunstobjekte

Besondere, auffällige Objekte tragen zur Wiedererkennung des jeweiligen Ortes erheblich bei, da sie oft einzigartig sind. Wegen eben dieser Eigenschaft sind sie jedoch auch schlecht durch automatische Verfahren zu erfassen, denn gerade die Erkennungsmerkmale sind bei jedem Objekt anders. Denkmale und Springbrunnen gehören zu dieser Gruppe von Objekten. Für Touristeninformationsanwendungen können hier auch zusätzliche Textinformationen und Hyperlinks angefügt werden. Bei Springbunnen kann animiertes Wasser beispielsweise mit Partikelsystemen oder animierten Texturen dargestellt werden und erhöht die Attraktivität deutlich.

5.4 Vegetation

Pflanzen, Gras, Wald, Parkanlagen und Felder sind für die Darstellung ebenfalls von großem Interesse.
Pflanzentypen können sehr grob mit Hilfe von multispektralen Bilddaten und DOMs erkannt und rekonstruiert werden. Eine genaue Rekonstruktion ist kaum von Bedeutung, die Nutzung ähnlicher Modelle mit etwas variierenden Parametern ist meist ausreichend. Für Bäume sind solche Parameter die Unterscheidung

von Laub- und Nadelbäumen sowie Informationen über Größe und Umfang von Stamm und Baumkrone. Durch den bekannten Zeitpunkt der DOM-Aufnahme können Pflanzenhöhen mit Hilfe des Pflanzentyps zu anderen Zeitpunkten grob geschätzt werden, d.h. die Bäume wachsen virtuell im Modell weiter. Es gibt auch Verfahren, die nur mit Farbluftfotos arbeiten, um Baumkronendurchmesser oder gar diverse weitere Informationen wie Baumhöhe zu bestimmen. Letzteres wird mit Aufnahmen laubfreier Bäume im Winter erreicht, von denen der Schatten analysiert wird.
Wald kann evtl. anders behandelt werden als einzelne Stadtbäume. Für die Visualisierung können hier oft Vereinfachungen vorgenommen werden, um die Komplexität und damit die benötigte Darstellungszeit in Grenzen zu halten.

5.5 Straßenmobiliar

Laternen, Ampeln und Schilder sind alltäglicher Bestandteil des Straßenbildes einer Stadt. Auch Parkbänke, Blumenkübel, Mülleimer, Fahrradständer usw. gehören zur normalen Ansicht aus der Perspektive eines Fußgängers. Sofern diese Objekte momentan von der Stadt verwaltet werden, ist die georeferenzierte Datenhaltung vermutlich kein großer Zusatzaufwand. Informationen über Verkehrszeichen und Ampeln können für Anwendungen wie Verkehrssimulation inklusive korrekter Ampelphasen benutzt werden. Es ist leicht vorstellbar, dass für das manuelle Einfügen pro Objekt nur sehr wenig Zeit benötigt wird, sofern der Benutzer diese auf dem Luftbild erkennen kann. In manchen Fällen ist auch eine automatische Erkennung mit Hilfe von Luftbildern denkbar, evtl. kann dabei auch der charakteristische Schattenwurf der Objekte mitbenutzt werden. Fehlerraten lassen sich mit Plausibilitätstests minimieren, so sollten sich Verkehrsschilder oder Ampeln immer sehr nahe einer Straße befinden.

Abbildung 5.14: Manuell platzierte Verkehrsschilder

5.6 Bewegte Objekte

Fahrzeuge, Menschen und Tiere können die Stadt beleben und die Attraktivität des Modells deutlich steigern. Gerade in Spielen wird diese Erkenntnis umgesetzt. Außerdem helfen diese Objekte, die Größenverhältnisse richtig einschätzen zu können. Neben einer realistischen Simulation können sie auch zu reinen Visualisierungszwecken benutzt werden.
Die folgende Liste nennt einige Beispiele.

- Flugzeuge, Hubschrauber, Heißluftballons
- Eisenbahnen, Straßenbahnen, Seilbahnen, Zahnradbahnen, Untergrundbahnen, Fahrstühle an der Außenseite von Gebäuden
- Autos, Busse, Motoräder, Fahrräder
- Personen: Fußgänger, sitzende, laufende, Sportler
- Tiere: Vögel, Schmetterlinge, Pferde, Kühe, Hunde, Katzen, diverse Tiere in einem Zoo, Fische nahe unter der Wasseroberfläche, ...

In [Duca] befinden sich beispielsweise auf ländlichem Gebiet die für die jeweilige Region typischen Tiere. Menschen können sehr realistisch und einfach mit aufgenommenen kleinen Videosequenzen repräsentiert werden. Dafür ist jedoch der Blickwinkel eingeschränkt. Sport treibende Personen wie Fahrradfahrer können bei einer Touristeninformationsanwendung mögliche Freizeitaktivitäten repräsentieren.

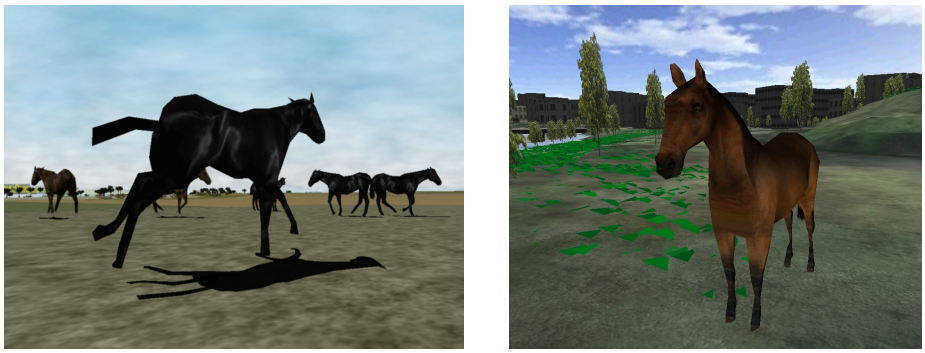

Abbildung 5.15: Automatisch platzierte Tiere in *Eingana* (Quelle: [Duca]) und manuell eingefügtes Pferd im Stadtmodell von Braunschweig

Alle Objekte haben Einschränkungen, wo sie sich bewegen können. Starke Einschränkungen wie Schienen (1,5D) erleichtern die Simulation der Bewegung und sind besonders einfach zu integrieren. Je nach Anwendung und Verfügbarkeit der Daten ist z.B. eine Echtzeit-Ansicht aller Züge und Straßenbahnen einer Stadt vorstellbar. Doch auch die Simulation mit Hilfe künstlicher Intelligenz ist interessant und kann sehr komplex sein, beispielsweise für tiefffliegende Vögel im Schwarm oder für Autos, die sich realistisch im Verkehr bewegen sollen.

5.7 Verkehrs- und Transportnetze

Bei diesen Netzen kann prinzipiell zwischen deren vereinfachten Graphen und deren geometrisch exakt repräsentierten Streckenführungen unterschieden werden. Erstere sind für Simulationen und Suchalgorithmen interessant, während letztere zur Visualisierung eingesetzt werden. Für einen Routenplaner ist zur Berechnung einer Route nur der Graph von Bedeutung, die Visualisierung benötigt mehr Informationen über den konkreten Verlauf der Straßen.

- Straßen mit Fahrbahnen, Rad- und Fußwegen
- Hochspannungsleitungen
- Kommunikationsnetze
- Gas- und Wasserversorgung, Abwasserkanalisation

Teilweise können die Netze oder Zusatzinformationen, sofern noch nicht vorhanden, semiautomatisch mit Hilfe von Luftbildern und LIDAR-Daten erzeugt werden. Sind beispielsweise nur Straßenmittelachsen vorhanden, könnten vermutlich für viele Straßen automatisch die Fahrspuren und deren Grenzen erkannt werden.

5.8 Untergrund-Objekte

Objekte unter der Erdoberfläche können beispielsweise angezeigt werden, indem die Erdoberfläche etwas transparent dargestellt wird. Röhren können komplex in allen drei Dimensionen verlegt sein, eine 3D-Visualisierung kann hier sehr zum Verständnis beitragen. Tunnel und Höhlen stellen eine weitere Anforderung an die benutzte Terrain Engine: Sie muss Löcher enthalten können. Dies kann allerdings auch mit Hilfe einer transparenten Stelle in der Textur geschehen. Auch die Visualisierung anderer unterirdischer Strukturen wie Bergwerksschächte und Bodenschichten ist vorstellbar.

5.9 GUI-Objekte

Rein virtuelle Objekte sind vorstellbar, die nur der Interaktion dienen und z.B. durch Anklicken bestimmte Aktionen auslösen können. Es handelt sich sozusagen um GUI-Elemente, die im Raumbezug stehen. Oft können diese Interfaces aber direkt an betreffende Objekte angebracht werden. Auch Markierungen eines bestimmten Ortes oder die Visualisierung eines Weges gehören dazu (siehe auch Abb. 7.18 auf S. 71).

Kapitel 6

Interaktive Darstellung

Zur Visualisierung einer Stadt fallen große Datenmengen an. Die Benutzung unterschiedlicher Levels of Detail ist besonders wichtig. Die gerade am häufigsten gebrauchten Daten müssen dabei im schnellen Arbeitsspeicher gehalten werden. Mit der richtigen Mischung von bildbasierten Techniken und Geometrie je nach Art, Größe und Orientierung der Objekte auf dem Bildschirm kann eine schnelle und realistische Anzeige auf PCs erzielt werden. Eine der Herausforderungen ist das große Datenvolumen, das im vorliegenden Beispiel für die Stadt Braunschweig komprimiert etwa 5 Gigabyte beträgt. Mit zunehmendem Detaillierungsgrad der Objekte sowie Fassadentexturen kann sich die Datenmenge leicht vervielfachen.
Gerade im Hinblick auf zukünftige mobile Anwendungen im Navigations- und Informationsbereich lohnt sich die Weiterentwicklung

6.1 Culling

Das Culling bezeichnet eine Technik, Objekte nicht zu rendern, die sowieso nicht sichtbar sind. Dies ist bei komplexen Szenen notwendig, um eine hohe Framerate zu erzielen. Es handelt sich hier also ausschließlich um eine Geschwindigkeitsoptimierung.
Ein hierarchisches Clustering macht oft Sinn, so dass nicht zu viele Culling-Tests durchgeführt werden müssen. Einige oft genutzte Culling-Verfahren, die in dieser Reihenfolge angewandt werden, sind:

Contribution culling oder Detail Culling:
Objekte, deren projizierte Fläche sehr klein ist.

View Frustum Culling:
Objekte außerhalb des für die Kamera sichtbaren Volumens.

Occlusion Culling:
komplett durch andere Objekte im Vordergrund verdeckte Objekte.

Backface Culling:
Rückseiten von Polygonen.

Für das Frustum Culling ist ein Test mit Bounding Spheres einfach und meist recht effizient. Occlusion Culling wird gerade bei Innenansichten benutzt. Hier existieren diverse Techniken wie die Vorberechnung von Sichtbarkeitsgraphen.

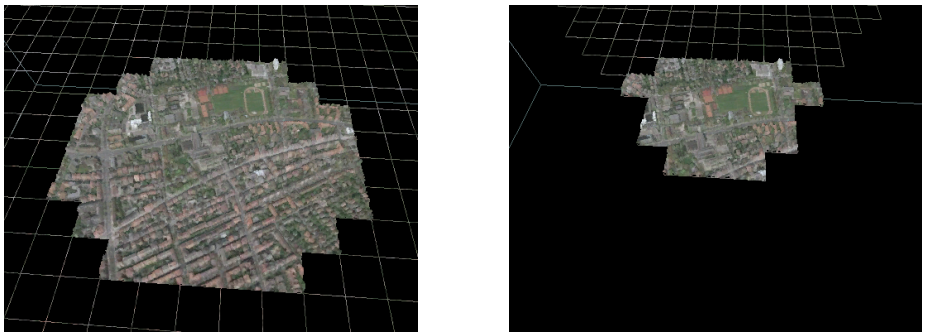

Abbildung 6.1: Das benutzte View Frustum Culling wird erst sichtbar, wenn das View Frustum einer anderen Kamera beobachtet wird.

Nach jedem Culling-Test reduziert sich die Anzahl der zu zeichnenden Objekte meist deutlich. Daher wird das Culling bei komplexen Szenen nicht der Hardware überlassen, sondern schon frühzeitig per Software erledigt, um die Datenmenge deutlich zu reduzieren (siehe Abb. 6.2).

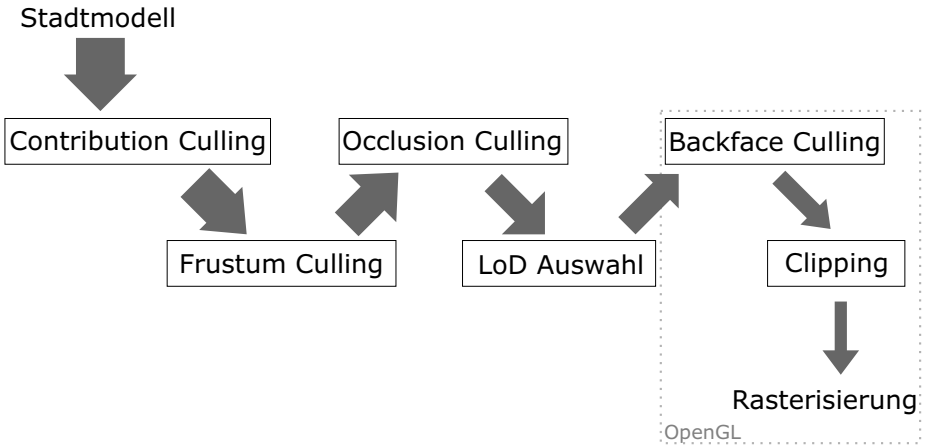

Abbildung 6.2: Wichtige Reduktionsschritte der Menge der potentiell zu zeichnenden Daten. Die Dicke der Pfeile soll die Größenordnung der Datenmenge andeuten.

6.2 Level of Detail

LoD-Techniken beim Rendern werden hauptsächlich zur Geschwindigkeitsoptimierung genutzt. Der Betrachter sollte dabei möglichst keinen Unterschied zur Szene in voller Detailauflösung bemerken können. Andere Anwendungen sind das NPR, die Vermeidung von Aliasingartefakten und die geringe benötigte Bandbreite, was vor allem für die Übertragung der Daten wichtig ist.

Die Erzeugung diskreter Detailstufen erfolgt meist in einer Vorberechnungsphase. Auch dynamische, nicht diskrete Detaildarstellungen sind möglich. Eine Kombination verschiedener Techniken ist sinnvoll. Hautsächlich bei der Benutzung diskreter Detailierungsgrade sollten Vorkehrungen zur Vermeidung von Popping-Artefakten beim Wechsel zwischen unterschiedlichen Stufen getroffen werden.
Von großem Vorteil ist eine progressive Repräsentation der Daten im Speicher, bei der vorerst nur die benötigten Daten geladen werden müssen und bei Bedarf weitere Details dazugeladen werden können. Im Vergleich zum kompletten Ablegen aller Daten für jede Auflösungsstufe kann auch Speicherplatz gespart werden, wodurch weniger Daten von der langsamen Festplatte geladen werden müssen und der schnellere Arbeitsspeicher besser ausgenutzt werden kann. Ein gutes Beispiel ist die Repräsentation mit Hilfe von Wavelets. Neben Volumenbeschreibungen wird diese Technik vor allem zur Kompression von 2D-Rasterbilddaten benutzt.
Bei 3D-Stadtmodellen ist es zweckmäßig, LoDs nicht nur jeweils für einzelne Objekte zu benutzen, sondern auch für Gruppen bzw. Bereiche mit mehreren Objekten. So stellt auch die Skybox solch ein LoD der ganzen umgebenden Landschaft dar.
Die Bestimmung der darzustellenden Detailstufe kann von der projizierten Größe und dem Blickwinkel abhängen. Bei Objekten, die auf nur wenige Pixel abgebildet werden, kann ein sehr grobes Modell benutzt werden. Doch auch bei Blickwinkeln ähnlich den benutzten Fotos kann eine sehr grobe Geometrie auch bei größeren Objekten ausreichen.
Das Vorgehen bei der Generierung geringerer Detailstufen bei Rasterdaten ist das Downsampling mit Hilfe der üblichen Filterkernels und anschließender Abtastung. Bei Bildern ist dies offensichtlich, bei Höhendaten sind jedoch andere Techniken zu diskutieren. Evtl. machen hier z.B. Medianfilter mehr Sinn als einfache Faltungsoperationen.
Gebäude oder andere 3D-Polygonmodelle erfordern aber grundsätzlich andere Arten von geringen Detailstufen. Neben der Simplifizierung der Dreiecksnetze gibt es noch ganz andere Ideen, die zusätzlich genutzt werden können. Einige werden in den folgenden Unterkapiteln beschrieben.

6.2.1 Point based rendering

Für das Rendern des Objekts werden einzelne Punkte der Oberfläche ausgewählt und in der entsprechenden Farbe dargestellt. Je nach Detailierungsgrad und verfügbarer Rechenzeit kann die Anzahl der Punkte variieren. Die Implementierung ist für eine beliebige Geometrie sehr einfach und effizient. Für Pflanzen ist diese Technik besonders gut geeignet.
Anstatt Punkte von den Stadtmodell-Objekten zu extrahieren können auch direkt die LIDAR-Messpunkte benutzt werden. Bei einem horziontalen Blick entstehen aber große Lücken, so dass ich breite, vertikale Linien anstatt nur Punkte benutzt habe (siehe Abb. 5.2, S. 32).
Für weiter entfernte Linien kann bei diesem *vertical line based rendering* eine Auswahl bestimmter, wichtiger Linien eine Rolle spielen. Verschiedene Auswahlkriterien können getroffen werden. Gerade an Kanten oder Eckpunkten im DOM ist der lokal höchste Punkt wichtig. Um Lücken zu vermeiden, sollten aber keine großen, nicht abgetasteten Flächen übrig bleiben. Nach der Auswahl können

sie innerhalb einer Gruppierung noch nach der Höhendifferenz von DOM und DGM sortiert werden. Abhängig von Blickwinkel, Entfernung und verfügbarer Renderzeit braucht so jeweils nur der erste Teil der Linien gerendert werden. Überhängende Strukturen können nicht dargestellt werden und bei der Verwendung von Orthofotos sind die Linien jeweils nur einfarbig. Neben einigen Nachteilen sind die einfache Implementierung und die hohe Geschwindigkeit sehr vorteilhaft. Vermutlich kann diese Technik selbst bei extrem schnellen Kamerabewegungen über mehrere Quadratkilometer gut benutzt werden, da Informationen über die wichtigsten Linien wenig Speicherplatzbedarf haben und schnell geladen werden können.

6.2.2 Sprites und Billboards

Zur Vereinfachung von Geometrie können diese bildbasierten Techniken benutzt werden. Gerade bei Pflanzen kann so sehr viel Zeit eingespart werden. Inzwischen können für beliebige Objekte automatisch Billboard-Wolken erzeugt werden. Einen guten Überblick über die Möglichkeiten bietet [DDSD03].

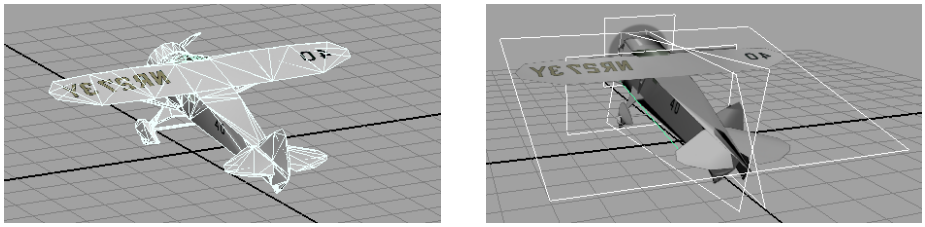

Abbildung 6.3: Ein Flugzeugmodell wurde mit Hilfe von Maya und Gimp von über 1600 auf 6 Polygone reduziert.

Das bereits erwähnte View-Dependent Texture Mapping gehört ebenfalls zu dieser Kategorie.

6.2.3 Impostors

Wie in [DSSD99] ausführlich beschrieben, bieten sich Impostors an, um das Rendering entfernter Geometrie wesentlich zu beschleunigen, ohne die Bildqualität merklich zu beeinflussen. Prinzipiell werden dabei entfernte Objekte in eine Textur gerendert, die auf eine grobe Geometrie gelegt wird. Das passiert aber nicht bei jedem Frame, sondern nur, wenn sich der Blickwinkel merklich ändert. Da die meisten Objekte einer Stadt statisch sind, können viele lokal gültige Impostors innerhalb einer Viewpoint-Zelle in einer Vorberechnungsphase erzeugt werden. Bei der Anzeige müssen nur noch die entsprechend passenden Impostors gerendert werden.

6.3 Dateistrukturen

Um effizient nur die lokal benötigten Daten in den Speicher laden zu können, müssen die Datenbestände in Kacheln aufgeteilt werden. Während sich meist rechteckige Kacheln anbieten, sind gerade für die Gruppierung von Vektordaten auch secheckige Kacheln vorstellbar, womit vermutlich ein etwas effizienteres

Frustum Culling möglich wäre. Um die Hardware gut auszunutzen, ist es für Rasterdaten oft notwendig, quadratische Kacheln mit einer Zweierpotenz als Breite und Höhe zu benutzen.
Für Vektordaten und Objekte sind auch unregelmäßige Kacheln mit adaptiver Größe vorstellbar, so dass sie jeweils ungefähr die gleiche Menge an Daten beinhalten.
Optimal sind vermutlich Dateien, die Informationen einer Kachel progressiv enthalten. Nach einigen Größenordnungen sollten andere Kacheln in der passenden Größe benutzt werden, denn es wäre nicht effizient, beispielsweise für Bilddaten für eine Übersicht aus großer Höhe die gleichen Dateien zu benutzen, die auch die höchste verfügbare Auflösung enthalten.
Bei der Benutzung von verlustbehafteten Bilddateiformaten für Rasterdaten ist Vorsicht geboten, da diese für das menschliche Sehsystem optimiert sind. Für eine Nutzung für Höhendaten oder eine Signalverarbeitung sollten evtl. andere Kompressionsverfahren benutzt werden.
Bei einem Streaming der notwendigen Daten über das Internet können nach und nach immer mehr Objekte geladen werden. Ein dynamisches Einfügen in den Szenengraph und das progressive Verfeinern dieser Objekte sind wünschenswert.

6.4 Rendering-Architektur

Für eine schnelle Darstellung ist OpenGL gut geeignet. Für Webanwendungen könnte auch Java3D in Zukunft an Bedeutung gewinnen.
Für realistische Darstellungen mit Reflexionen und Schatten sind viele Maßnahmen notwendig, die sich ständig weiterentwickeln. Ansätze für Realtime Raytracer sind interessant, im Moment jedoch nicht von Bedeutung.
Game-Engines bauen auf OpenGL oder DirectX auf und nutzen die Fähigkeiten der Grafikhardware gut aus. Werden hohe Ansprüche an Geschwindigkeit und Optik gestellt und können die Lizenzgebühren in Kauf genommen werden, ist die Benutzung einer aktuellen Game-Engine empfehlenswert.

6.5 Interaktion

Eingesetzte LoD und Culling-Techniken werden oft für einen speziellen Kamerabereich ausgelegt. Die Kameraposition sollte entsprechend eingeschränkt werden. Verschiedene Navigationsmöglichkeiten werden im folgenden Kapitel besprochen. Als Feedback für die aktuelle Position eignet sich eine zusätzlich eingeblendete Übersichtskarte.
Es gibt immer einen Kompromiss zwischen Darstellungsgeschwindigkeit und -qualität. Dem User sollte es möglich sein, diese Parameter zu beeinflussen.

6.6 Darstellungsoptionen

6.6.1 Fotorealismus

Eine möglichst realistische Darstellung ist oft das Ziel dreidimensionaler Visualisierungen. Weiche Schatten, Spiegelungen und weitere Effekte sind verhältnismäßig aufwendig zu realisieren und stellen noch eine Herausforderung dar.

Praktischerweise sind die Schatten schon in den Fotos enthalten, so dass bei deren Benutzung eine aufwendige Schattenberechnung entfallen kann. Dieser Vorteil wird aber zu einem Problem, wenn der Sonnenstand geändert werden soll. Das Entfernen des Schattens ist sehr schwierig. Aufnahmen bei bewölktem Wetter könnten hier helfen.
Die Darstellung von unterschiedlichen Jahreszeiten, Wetter oder Tageszeiten kann sehr interessant sein (siehe Abb. 6.4). Die Simulation von Nebel ist sehr einfach, fotorealistischer Schnee hingegen ist viel schwieriger zu realisieren.

Abbildung 6.4: Ein Bild eines Raytracing-Videos einer prozedural generierten Stadt (Quelle: Procedural City Modeling and Night City Lighting)

Vor allem in Spielen werden sogar Effekte simuliert, die durch helle Lichtquellen in der Kameraoptik und den Aufnahmesensoren entstehen. Lens Flares, Glares, Tiefen- und Bewegungsunschärfe gehören zu solchen nachgebildeten Artefakten (siehe Abb. 6.5).

6.6.2 Non-Photorealistic Rendering

Für künstlerisch wirkende Darstellungen können NPR-Techniken benutzt werden. Silhouetten und Schraffuren werden benutzt, um eine Zeichnung zu simulieren. NPR kann für Planungen benutzt werden, damit deutlich wird, dass das entgültige Resultat noch nicht feststeht und noch geändert werden kann. Doch auch aus ästhetischen oder informellen Gründen kann vom Fotorealismus abgewichen werden.
Schließlich müssen nicht so viele Details zur Verfügung stehen. Viele Techniken werden in [AG01] behandelt.
Um einen besseren Überblick zu erhalten, könnten Gebäude und andere Objekte z.B. halbtransparent gezeichnet werden, damit dahinterliegende Objekte auch noch sichtbar sind. Ergebnisse von GIS-Suchanfragen können optisch hervorge-

Abbildung 6.5: Grelle Lichtblitze in *need for speed underground* (Quelle: EA Games)

hoben werden, in dem andere Farben, Leuchteffekte oder Ähnliches benutzt werden. Eine farbliche Codierung der Gebäudenutzung ist eine gebräuchliche Art der Darstellung. Können z.B. wegen geringer zur Verfügung stehender Bandbreite keine hochauflösenden Texturen benutzt werden, macht es evtl. Sinn, vollkommen auf Texturen zu Verzichten und NPR zu benutzen.
Des Weiteren ist es möglich, analog zur generalisierten Karten, Objekte in einer vereinfachten oder vergrößerten Form darzustellen. So ist z.B. eine Übersicht über alle wichtige Straßen einer Stadt möglich, die bei einer realistischen Darstellung wegen ihrer geringen Breite kaum zu sehen sein würden. Die Größe der Straße hängt hier eher vom Verkehrsfluss als von der wirklichen Größe ab. Im Gegensatz zu manuell erzeugten, diskreten Generalisierungsstufen ist die Idee einer dynamischen Generalisierung sehr interessant.
Die Färbung von Gebäuden kann auch zur besseren Wiedererkennung generell so erfolgen, dass z.B. alle Dächer rötlich und alle Wände gelblich gefärbt werden. Dieses Konzept wird zusammen mit Silhouetten in den seit 1950 existierenden Bildkarten wie in Abb. 6.6 genutzt.

6.6.3 Zusatzinformationen

Weitere räumliche Informationen, die in der realen Welt nicht sichtbar sind, können nützlich für die Ansicht sein. Koordinatengitter und Höhenlinien sind Beispiele dafür (siehe Abb. 6.7). Auch die Visualisierung von anderen Daten wie Mess- oder Simulationsergebnisse ist von Bedeutung.

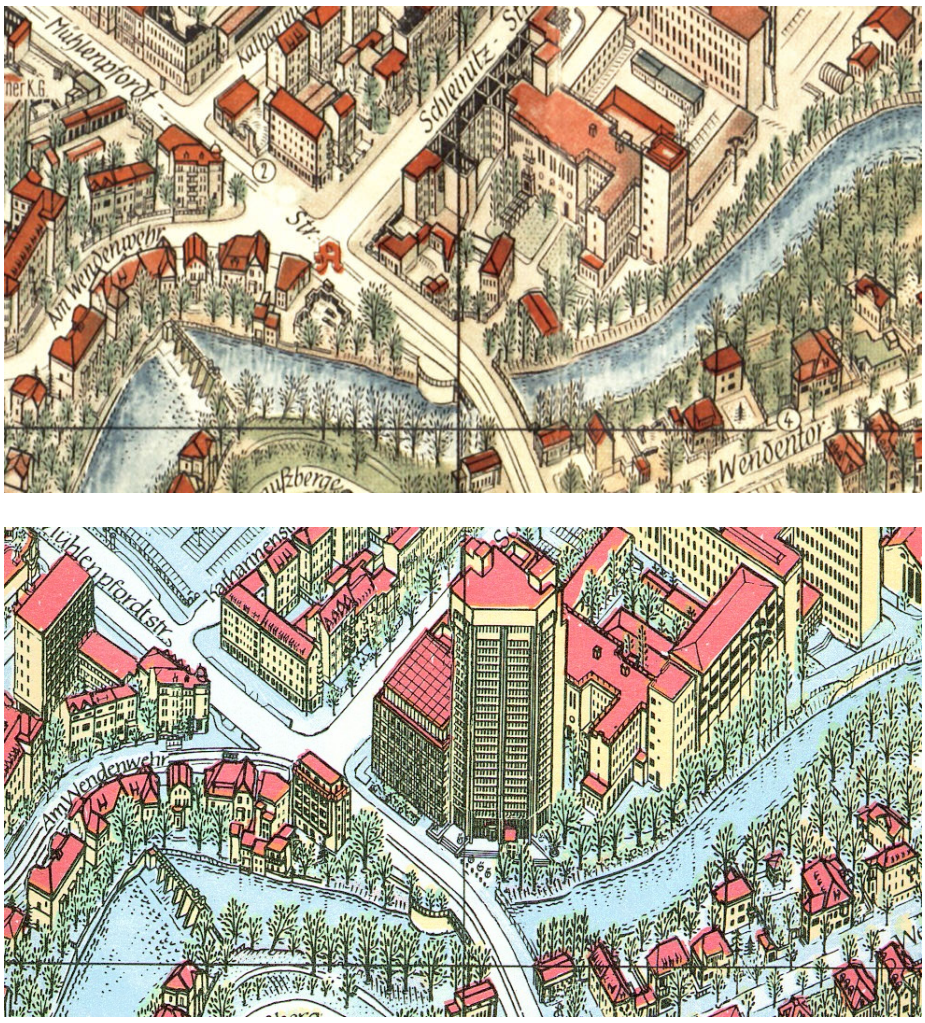

Abbildung 6.6: Die Farbe entspricht dem Objekttyp, nicht der realen Farbe. Die Karten stammen aus den Jahren 1954 und 2000. (Quelle: ©Bollmann-Bildkarten-Verlag Braunschweig, Abdruck mit Freundlicher Genehmigung)

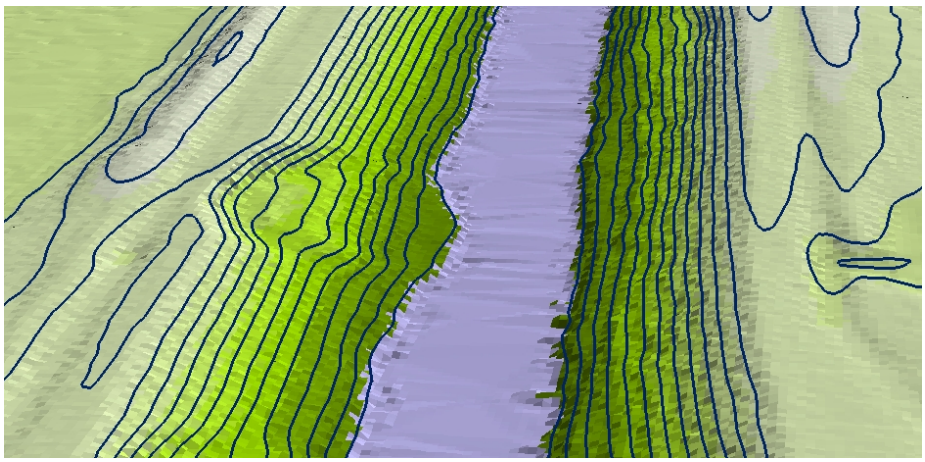

Abbildung 6.7: Höhenlinien (Quelle: TerraPoint)

Kapitel 7

Implementierung

Während der konkreten Realisierung der Konvertierungstools und dem *CityModelViewer* sind viele Erkenntnisse gewonnen worden. Die Notwendigkeit von guten LoD-Techniken und das dynamische Laden wurden bei den großen Datenmengen offensichtlich. Die Implementierung wurde stark im Hinblick auf die Daten der Stadt Braunschweig vorgenommen. Ich habe mich kaum mit Problemen befasst, die sich beim Beispiel Braunschweig gar nicht gestellt haben, weil die Daten *nicht* existieren oder weil sie *schon* existieren. Vektorbasierte Flächeninformationen der Straßen und Bürgersteige z.B. wären für ein Dreiecksnetz der Erdoberfläche nützlich gewesen, wurden aber nicht berücksichtigt, da diese Daten nicht für Braunschweig existieren. Die automatische Extraktion von Gebäudegrundrissen aus Höhendaten wurde nicht behandelt, da diese Informationen bereits vorhanden waren.
Allerdings werden alle Daten mit Konfigurationsdateien geladen, so dass durch die Nutzung einer anderen Stadtbeschreibungsdatei der *CityModelViewer* unverändert für Daten anderer Städte benutzt werden kann. Die Konvertierungstools müssen typischerweise abhängig von den Eingangsdaten angepasst werden. Einige Daten der Stadt Köln wurden schließlich auch konvertiert und können nun mit dem *CityModelViewer* betrachtet werden. Andere Daten wie Straßennamen und Hausnummern wurden direkt für den Fall von Braunschweig implementiert und nicht extra in ein allgemeines Format konvertiert, so dass diese Informationen für andere Städte erst in ein ähnliches Format konvertiert werden müssten.
Bei der Implementierung stand weniger die Softwaretechnik im Vordergrund. Vielmehr wurde einiges an Darstellungsmöglichkeiten erprobt und im Programmcode gelassen, auch wenn der Quelltext dadurch etwas unübersichtlicher wurde.
In Abb. 7.1 ist ein Schema des jetzt benutzten Systems zu sehen. Spezielle Editoren für die Daten stehen nicht zur Verfügung. Bei neuen Importen der Stadtmodell-Daten gehen auch manuell geänderte Informationen verloren, da diese Änderungen nicht an der eigentlichen Datenquelle vorgenommen werden können, sondern nur direkt an den importierten Dateien. Diese Importe sind teilweise recht komplex, da die Daten nur mit begrenzten semantischen Informationen verknüpft sind.

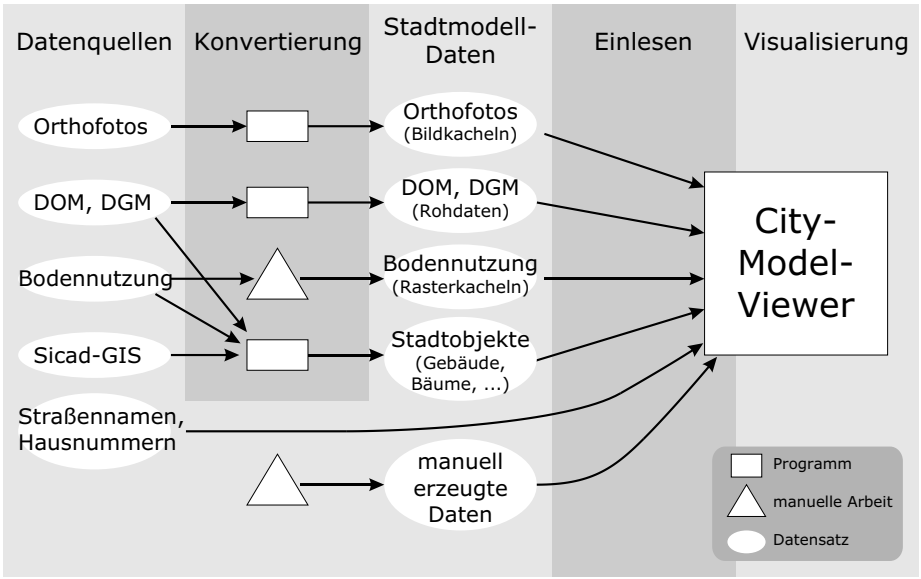

Abbildung 7.1: Vereinfachte Übersicht der Implementierung

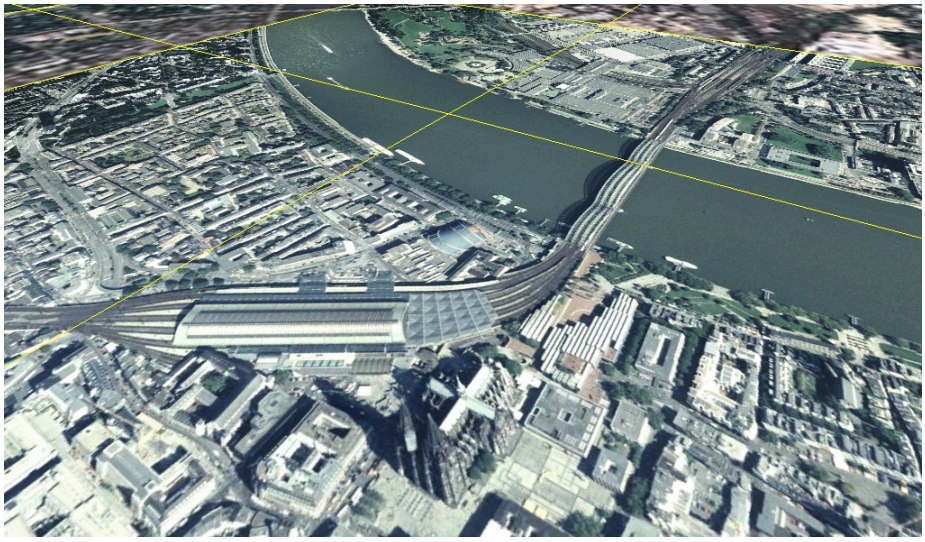

Abbildung 7.2: Köln: Grobe und detaillierte Luftfotos auf einem groben Höhenmodell

7.1 Daten

Das Alter der Datenquellen ist unterschiedlich und es kommt an einigen Stellen zu Unstimmigkeiten. Die Bodennutzungskarte von Braunschweig basiert auf einer Bestandsaufnahme von Biotoptypen aus dem Jahr 1989.
Bei der Implementierung ergaben sich einige Einschränkungen bezüglich der Kachelgrenzen und -größen der Daten. Durch eine Codierung der Südwestecke im Dateinamen sind nur ganzzahlige Koordinaten vorgesehen. Da die Originalkacheln an ganzahligen Kilometergrenzen, Vielfachen oder Unterteilungen orientiert sind, wurden die gleichen Grenzen benutzt, um die Konvertierung nicht zu verkomplizieren. Die Breite und Höhe einiger Daten müssen im Speicher aber Zweierpotenzen sein, um Bibliotheken oder Grafikhardware nutzen zu können. Dafür wurden die Daten teilweise etwas skaliert. Zur Zeit müssen die Kachelgrenzen der Orthofotos mit den Kachelgrenzen der Höhendaten übereinstimmen, damit während des Renderns einer Kachel des Erdbodens nicht laufend die Textur gewechselt werden muss.

7.1.1 Koordinatensysteme

Allgemein verfügbare, grob aufgelöste Daten sind oft in unterschiedlichen Koordinatensystemen vorhanden. Die Umrechnung zwischen den diversen benutzten Koordinatensystemen ist für einen geringen Fehler von deutlich unter einem Meter nicht trivial. Die Vielzahl der Koordinatensysteme verdeutlicht Abb. 7.3.

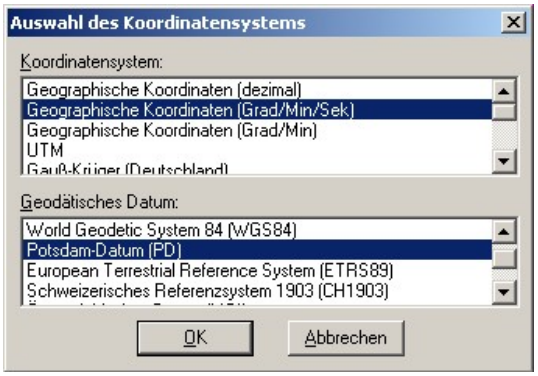

Abbildung 7.3: Teil einer Dialogbox zur Auswahl des Koordinatensystems (Quelle: Geogrid Viewer, EADS/Dornier)

Die Daten der Stadt Braunschweig sind glücklicherweise alle im gleichen Koordinatensystem und müssen daher nicht umgerechnet werden. Da beim benutzten Gauß-Krüger-System gerade ein Wechsel der Meridianstreifen mitten in der Stadt erfolgt, wurde zudem die Konvention eingehalten, den 4. Meridianstreifen zu benutzen.
Die Andreaskirche in Braunschweig hat z.B. die folgenden Koordinaten:

- Geographische Koordinaten:
  geogr. Länge: 10° 31'15,841"; geogr. Breite: 52° 16'09,442"
- Gauß-Krüger Koordinaten im 3. Meridianstreifen (9°):
  Y = 3 603 820,13 m; X = 5 793 801,08 m

- Gauß-Krüger Koordinaten im 4. Meridianstreifen (12°):
  Y = 4 399 055,56 m; X = 5 793 741,52 m

Die Nordrichtung des Gauß-Krüger Gitters weicht von der geographischen Nordrichtung in Braunschweig etwa um 2,4° ab.
Die Zahlenbeispiele zeigen auch die benötigte Genauigkeit von über neun Dezimalstellen, wofür viele Grafikprogramme nicht ausgelegt sind und damit nicht benutzt werden können, da zu große Rundungsfehler auftreten oder zu wenige Stellen exportiert werden können.

7.1.2 Dateiformate

Um die Konvertierungsergebnisse abzuspeichern, wurden eigene Dateiformate benutzt, die teilweise die notwendigen semantischen Informationen enthalten können und nicht rein zur Visualisierung optimiert sind. Es wurden auch Standardformate benutzt, soweit dies möglich war. Rasterdaten können im JPG- und PNG-Format eingelesen werden, wofür die kapselnde Bibliothek CxImage eingesetzt wurde. Höhendaten werden teilweise im Rohdatenformat gehalten, da das Einlesen ASCII-kodierter Formate hier zu lange dauert. Beliebige 3D-Polygonmodelle können im OBJ-Format in das Modell eingefügt werden. Die aktuelle Implementierung unterstützt nicht die komplette OBJ-Formatspezifikation, doch können texturierte Dreiecksmodelle geladen werden.

Alle Daten werden für die Visualisierung in einem Radius um einen Punkt, das *CoI* (center of interest), geladen und ab einer gewissen, größeren Entfernung auch wieder freigegeben. Diese Grenzen lassen sich für jeden Datensatz in der Konfigurationsdatei einstellen und bieten somit eine gute Anpassung an unterschiedliche Geschwindigkeiten von verschiedenen Computern. Diese simple Idee war einfach zu implementieren, hat jedoch den Nachteil, dass die Region, in der Daten geladen werden, immer kreisförmig ist. Eine genauere Anpassung der Region an die Projektion des View Frustums auf die Erdoberfläche sollte bessere Ergebnisse liefern.
Für die Konvertierung werden die Daten innerhalb eines recheckigen Bereichs geladen, da auch alle Konvertierungsbereiche rechteckig sind.
Für die Konfigurationsdateien (*.cm) und auch für die Stadtobjektdateien (*.ca) wird ein einfaches, XML-angelehntes ASCII-Format benutzt. Hier folgt ein Ausschnitt der Datei ' `Braunschweig.cm` ':

```
<CityModel>
<Sun>
x=-0.23
y=-1
z=0.47
</Sun>

streetsFilename=..\data\BS\strassen
houseNumbersFilename=..\data\BS\bhnr

TexturePath=textures\
ModelPath=..\data\BS\models\

<SkyBox>
north=bs skybox4 north.jpg
south=bs skybox4 south.jpg
east=bs skybox4 east.jpg
west=bs skybox4 west.jpg
top=bs skybox4 top.jpg
bottom=bs skybox4 bottom.jpg
</SkyBox>

<TerrainHeightMapArray>
south=5794000.0
north=5796000.0
west=4398000.0
east=4402000.0

width=32
height=16

MapFiles=..\data\BS\Terrain 125x125 *.float

MinLoadDist=10
MaxKeepDist=500
</TerrainHeightMapArray>

...

</CityModel>
```

Der Sonnenstand (als Vektor), die Verzeichnisse und Namen von Datenbanken und Kacheln und die Grenzen der Rechteckigen Gebiete in Gauß-Krüger Koordinaten werden definiert. Für die Höhendaten der Erdoberfläche wird die Breite und Höhe des Feldes mit Kacheln angegeben. Die anfänglichen Einstellungen, ab welcher Distanz zum *CoI* die einzelnen Kacheln geladen oder wieder freigegeben werden sollen, sind ebenfalls enthalten.
Da sich fast alle Daten in rechteckigen Kacheln befinden, wurde der Dateiname benutzt, um die Position der Kachel zu beschreiben. Dabei stehen am Ende des Dateinamens vor der Dateiendung der Rechts- und der Hochwert des südwestlichen Eckpunkts. Benötigte Dateinamen können so automatisch erzeugt werden. Die Koordinaten werden an der Stelle des `*` eingefügt.
Hier folt ein Ausschnitt der Datei ' `BS 1000x1000 4399000 5794000.ca` ':

```
<CityArea>
north=5795000.000000
west=4399000.000000
south=5794000.000000
east=4400000.000000

<div>
filename=..\models\Parkbank.obj
tx=4399403.28
ty=5794245.62
tz=75.36
rx=-90
</div>

<building>
x=4399406
y=5794258
maxHeight=3.752
v=4399400.9 5794283.0
v=4399392.2 5794244.9
v=4399408.9 5794236.7
v=4399419.1 5794270.5
i=1 2 3
i=1 3 4
</building>

<plant>
x=4399999.350000
y=5794067.950000
height=4.060000
width=2.020000
</plant>

...

</CityArea>
```

Die Parkbank in diesem Beispiel wurde manuell eingefügt. Die Übernahme dieser Informationen in einen neuen Import ist nicht vorgesehen und muss manuell erfolgen. Für eine spätere Anwendung müssten diese Objekte wie Parkbänke in der Datenbank des GIS abgelegt werden und können dann ohne manuelle Arbeit importiert werden.

7.2 Interaktion

Zum Bewegen der Kamera im Modell wurden verschiedene Navigationsmodi implementiert. Im Karten-Navigationsmodus kann man die Welt um einen Punkt auf der Oberfläche, das CoI, drehen, zoomen und verschieben. Andere Navigationsmöglichkeiten sind die sehr ähnlichen Modi *Fliegen* und *Gehen*, die einigen Computerspielen nachempfunden wurden. Beim letzteren Modus wird als Unterschied nur die Z-Koordinate knapp über das Bodenniveau gelegt. Durch Drücken der Maustaste wird die Kamera sanft in Blickrichtung beschleunigt.

Während sich das CoI beim Fliegen und Gehen an der senkrechten Projektion der Kameraposition befindet, wird im Karten-Navigationsmodus auch der Schnittpunkt der Blickrichtung mit der Erdoberfläche mitbenutzt, so dass es hier auch bei horizontalem Blick sichtbar ist. Diese Wahl des CoI wäre bei den anderen Navigationsmodi nicht sehr sinnvoll, da dort häufiger Kameraschwenks auftreten, die zu einem dauernden Nachladen von Daten führen könnten.
Die automatische Positionierung des *CoI* kann angehalten werden, so dass die eingesetzten LoD-Techniken gut untersucht werden können.
Die Nutzung von Kamerapfaden ist auch möglich, befindet sich jedoch eher in einer experimentellen Phase. Wird nur ein Pfad für die Kameraposition angegeben, kann die Blickrichtung noch frei mit Hilfe der Maus bestimmt werden. Wird zusätzlich ein Pfad für das *CoI* angegeben, wird die Kamera vollkommen automatisch bewegt. Dies ist auch für die Erzeugung von kleinen Filmsequenzen hilfreich, deren Einzelbilder automatisch abgespeichert werden können. Für die Animationszeit wird hierbei die Nummer des Bildes benutzt, so daß beliebig viel Zeit für das Rendern eines Bildes zur Verfügung steht und es nicht zum Ruckeln durch nachzuladende Daten kommt. Abb. 7.4 zeigt diese Kamerapfade.

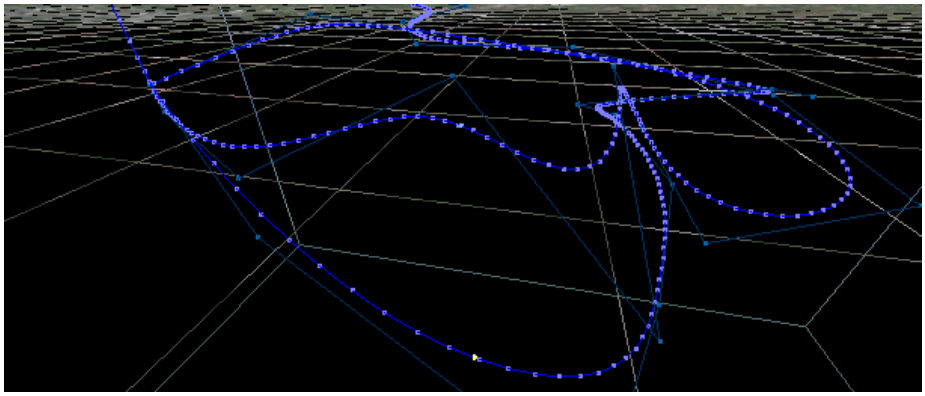

Abbildung 7.4: Nurbs-Kurven und deren Kontrollpunkte für Kameraposition und -blickrichtung

Für die Wechsel zwischen den verschiedenen Navigationsmodi wird versucht, die Parameter entsprechend umzurechnen, so dass sich die Sicht möglichst nicht verändert.
Das dynamische Laden oder Freigeben des Speichers stört manchmal und kann jeweils einzeln ausgeschaltet werden. Zur Untersuchung der Culling-Algorithmen kann auch hier die automatische Anpassung an die Kamera abgeschaltet werden.
Zum Springen zu bestimmten Positionen wird eine Textdatei ausgelesen, in der die Koordinaten oder Straße und Hausnummer der jeweiligen Position eingetragen werden:

```
...
<cam2>
x=4399407.0
y=5794256.0
</cam2>
<cam3>
street=Mühlenpfordtstraße
houseNumber=23
</cam3>
...
```

Einige Visualisierungsparameter können an- und ausgeschaltet werden. Dazu gehören Nebel, Detailtexturen, Höhenlinien, verschiedene Texturen der Erdoberfläche, verschiedene Datensätze, ein Gitter der Texturkacheln und der Wireframe-Modus.
Per Tastendruck kann ein Eckpunkt eines Pfades an der Stelle des Mauscursors eingefügt werden. Dieser kann beliebig im Raum verlaufen, wird jedoch standardmäßig knapp über der Erdoberfläche angelegt. Der Pfad wird mit animierten Kugeln dargestellt, die sich in Pfadrichtung bewegen. So kann ein Weg als Navigationshilfe eingezeichnet werden. Dieser Pfad kann auch zur Distanzmessung benutzt werden. Zudem kann dieser auch abgespeichert werden, so dass dessen Eckpunkte die Kontrollpunkte einer Bézierkurve definieren, die als Animationspfad für Objekte oder die Kamera dienen kann. Dafür kann auch der z-Wert des jeweils letzten Knotenpunkts mit der Maus geändert werden.
Besonders wichtige Informationen für Benutzer sind die Koordinaten an der Position des Mauszeigers, die direkt im Fenster angezeigt werden. Auf Wunsch können auch Informationen über das sich an der Stelle des Mauszeigers befindliche Gebäude angezeigt werden, wie Straßenname und Hausnummer.
Das Programm kann auch im Stereo-Modus betrieben werden und kann so gerade auf großen Rückprojektionsflächen eindrucksvolle, immersive Bilder erzeugen. Da *Glut* (OpenGL Utility Toolkit) benutzt wird, kann die Visualisierung ohne viel Aufwand auf verschiedenen Plattformen stattfinden. Eine Portierung für die *DAVE* (Definitely Affordable Virtual Environment) ist ebenfalls erfolgt. Hier ist auch die Navigation mit Hilfe eines Game-Controllers möglich (siehe Abb. 7.5).
Für weitere Nutzung der erzeugten Bilder sind große Renderings mit vielen Megapixels möglich. Dazu werden viele Kacheln einzeln gerendert und automatisch zu einem großen Bild zusammengesetzt. Dazu wird die *Tiled Rendering Library* benutzt, in der für die korrekte Funktionsweise noch ein Fehler behoben werden musste.

7.3 Objekte

7.3.1 Grobes DGM und Skybox

Für die konkreten Fälle von Braunschweig und Köln wurde jeweils eine Skybox mit dem Programm Terragen erstellt. Die Höhendaten wurden dazu in ein Bildformat konvertiert, um die Hügel in der Umgebung darzustellen. Da

Abbildung 7.5: Der Autor in der *DAVE*

die groben Höhendaten von der Braunschweiger Umgebung nur in einem proprietären Binärformat vorlagen, wurden mehrere Screenshots mit benutzerdefinierter Färbung von Höhenschichten manuell mit Hilfe eines Bildbearbeitungsprogramms übereinander gelegt (siehe Abb. 7.6).

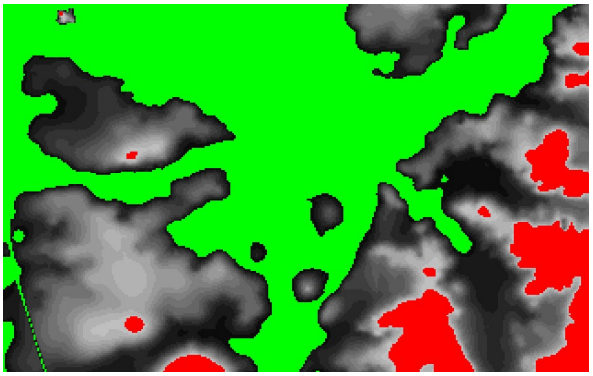

Abbildung 7.6: Ein Schritt bei der Konvertierung der Höhendaten

Die Position der Sonne in der Skybox wurde so gewählt, dass sie zu den Schatten auf den Luftfotos passt.
In einer früheren Version wurde die Silhouette der umliegenden Hügel noch bei Programmstart mit dem DGM berechnet und in hoch aufgelösten Texturen abgespeichert. Allerdings waren diese Hügel nicht texturiert und die Lösung, diese in die Skybox zu integrieren, erwies sich als praktischer.

7.3.2 Terrain Engine

Für die Darstellung der Erdoberfläche wird die Notwendigkeit von LoD-Techniken leicht deutlich, zumal sich im Verlauf dieser Arbeit die Verfügbarkeit eines DGMs im 1m-Raster ankündigte. Wegen Geomorphing und der Vermeidung von

Abbildung 7.7: Darstellung des Horizonts in einer früheren Version

Cracks ist die Realisierung nicht trivial. Ein eigener Implementierungsversuch zeigte viele Nachteile und wurde wegen zu großem noch notwendigen Arbeitsaufwand verworfen (siehe Abb. 7.8 und Abb. 7.9).

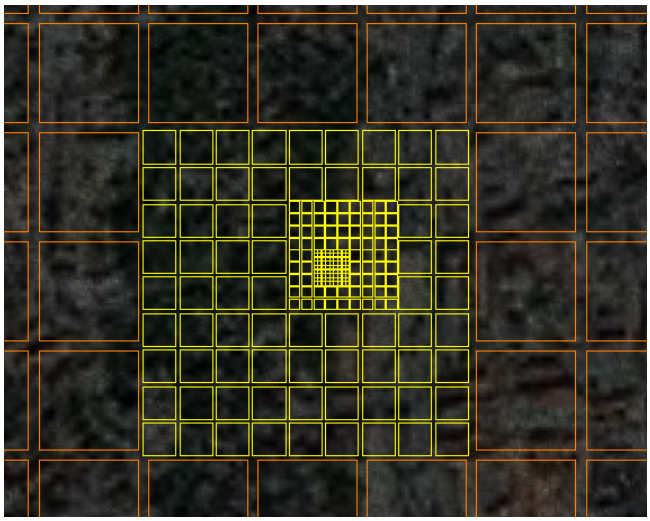

Abbildung 7.8: Eigener Test für ein dynamische LoD-Aufteilung der Erdoberfläche. Es befinden sich immer mindestens zwei Quadrate des selben LoDs nebeneinander, bevor das nächste beginnt.

Auf der Suche nach einer frei benutzbaren Terrain Engine stieß ich auf die *Mini*-Engine, die technisch weit besser ist. Diverse Modifikationen vor allem im Bereich des Renderings mussten vorgenommen werden, wie z.B. das Drehen des Koordinatensystems. Erst spät wurde mir bewusst, dass sich diese Engine nicht für dynamisches Nachladen weiterer Details oder dem Freigeben nicht mehr benutzter Kacheln eignet, da sie darauf ausgelegt ist, alle benötigten Daten bei Programmstart zu laden und gemeinsam am Programmende den Speicher freizugeben.
Auch langwierige Versuche führten nicht zu einem Erfolg. Sobald Kacheln wieder freigegeben werden, kommt es zu einem Absturz des Programms.
Die Terrain Engine sollte in Zukunft auch benutzt werden, um das DOM anzuzeigen, um hier eine höhere Geschwindigkeit zu erreichen.

7.3.3 CAD-Daten

Da das DXF-Format von AutoCad weit verbreitet ist und die Katasterdaten bereits in diesem Format in der Universität vorhanden waren, versuchte ich zuerst, diese Daten zu nutzen. Abb. 7.10 zeigt einen Screenshot einer sehr frühen

Abbildung 7.9: Eigene Terrain-Engine im Einsatz. Das DGM ist hier mit Hilfe des im Hintergrund sichtbaren DOMs selbst generiert. Die Gebäude sind zu Testzwecken von ihren grob approximierten Bounding Spheres umgeben.

Version des *CityModelViewers*, die auf viele brauchbare Datenstrukturen hoffen ließ.
Nach Benutzen externer Bibliotheken zum Lesen der DXF-Dateien und der Überprüfung der Dateien mit Hilfe von AutoCad wurde deutlich, dass für keine der Linien weitere Zusammenhangsinformationen vorhanden sind. Die Gebäudegrundrisse liegen in diesem Export nicht wie erwartet als Polygon vor, sondern als einzelne Linien. Auch wurden immer nur die Linien höchster Priorität exportiert. Da die Liegenschaftsgrenzen einer höheren Priorität zugeordnet sind, wurden häufig darunter liegende Gebäudelinien nicht mit exportiert. Die Gebäude sind nur an der Schraffur zu erkennen, ebenfalls bestehend aus einzelnen unzusammenhängenden Linien. Die Daten für die gesamte Stadt wurden importiert, waren jedoch zu einem hohen Grad fehlerhaft (siehe Abb. 7.11), auch wegen weiterer Probleme, die weiter unten beschrieben werden.
Nach diesen Erkenntnissen fragte ich die Stadt Braunschweig nach den Originaldaten, woraufhin ich verschiedene Exporte aus der GIS-Datenbank im Sicad-Format erhielt. Auch im ersten Export waren jedoch keine Flächeninformationen über die Gebäudegrundflächen enthalten. Zudem bemerkte ich, dass am Rand der 1 km^2 großen Kacheln die Grundrisse teilweise abgeschnitten sind. Das Studium von Formatbeschreibungen, Rückfragen mit der Firma Sicad und Benutzung des anderen Exportformats ergaben bessere Ergebnisse bezüglich der Gruppierung der Linien.

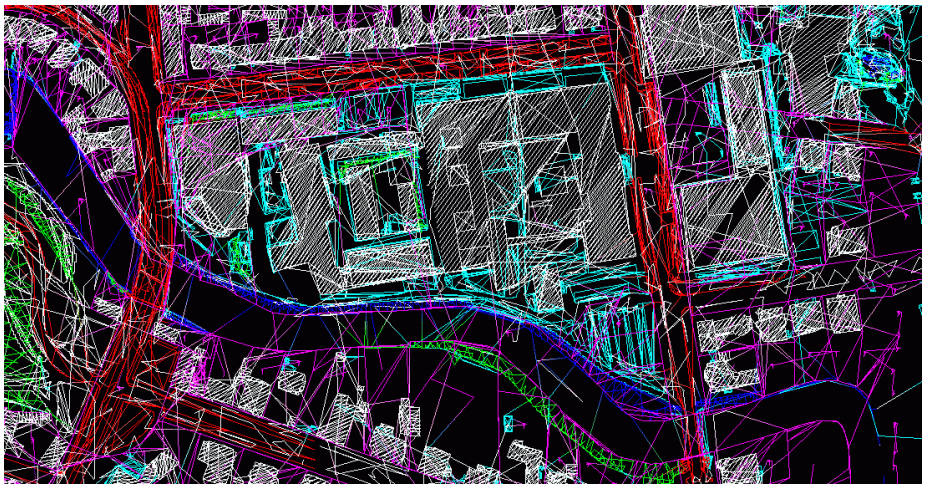

Abbildung 7.10: Erste Versuche zum Einlesen der DXF-Daten: Aufeinanderfolgende Punkte wurden mit Linien verbunden, entsprechend ihrem Layer unterschiedlich gefärbt

Abbildung 7.11: Konvertierungsergebnisse der Gebäudegrundrisse aus den DXF-Daten. Die Gebäudehöhen sind Zufallswerte.

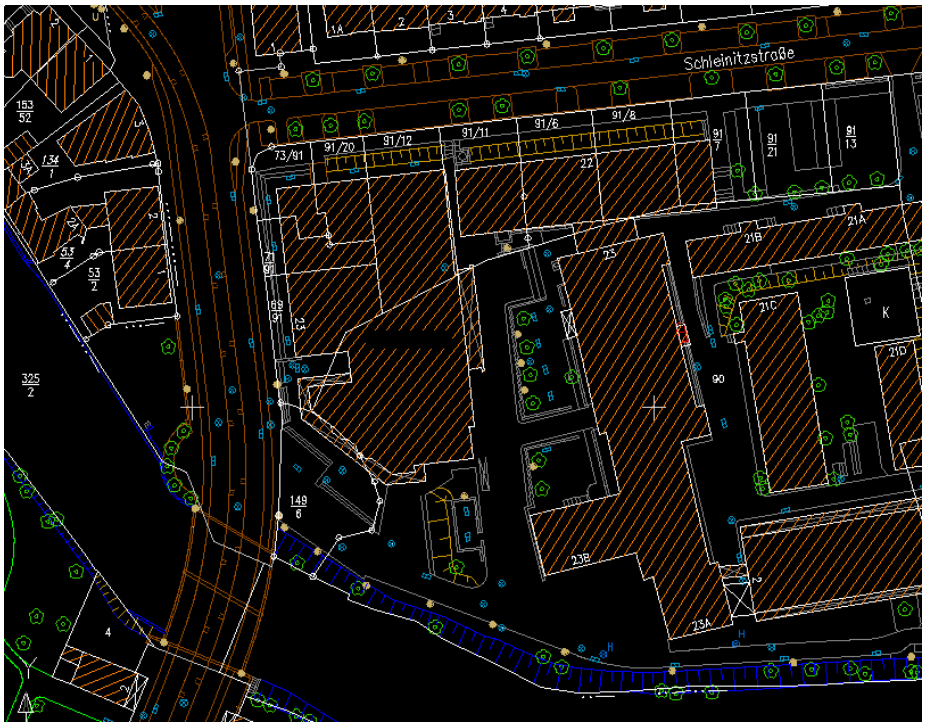

Abbildung 7.12: Auch die Ansicht eines Ausschnitts in AutoCad lässt eine umfassende Nutzungsmöglichkeit der Daten erhoffen.

Doch gibt es noch immer diverse Probleme. Die zu Gebäuden gehörenden Linien liegen auf unterschiedlichen Hierarchie-Stufen. Auch liegen sie in der Datei nicht in der richtigen Reihenfolge vor, so dass ein direktes Anhängen an bereits vorhandene Linienzüge nicht möglich ist. Einige der Linien gehen quer durch die Gebäude, um unterschiedlich hohe Gebäudeteile oder überdachte Flächen zu markieren. Letzteres wird durch Einzeichnen eines Kreuzes markiert, wodurch einige importierte Gebäudeformen fälschlicherweise einer Linie dieses Kreuzes folgen. Die Flächendefinition im Sicad-Format erfolgt durch einen Punkt, der innerhalb der Fläche liegt. Die meisten zugehörigen Linien werden in der Datei direkt anschließend definiert oder referenziert, sofern die Linie bereits vorher angegeben war. Leider liegen Parkplatz- und Gebäudegrundrisse auf dem gleichen Layer und sind nur dadurch zu unterscheiden, dass sich innerhalb der Parkplatzflächen ein Parkplatzsymbol befindet und innerhalb der Gebäudeflächen ein Schraffurobjekt vorhanden ist, bei dem alle Schraffurlinien angegeben werden. In Gebäudeumrissen enthaltene Kreisbögen wurden vorerst durch wenige Linien grob ersetzt, da sie nur selten vorkommen. Insgesamt ist ein sehr hoher Verarbeitungsaufwand notwendig, nur um die Gebäudegrundrisse einzulesen (siehe Abb. 7.13).
Die nun eingesetzten Suchalgorithmen können einen großen Teil der Gebäudegrundrisse fehlerfrei einlesen (siehe Abb. 7.14). Dabei werden geschlossene Polygonzüge einmal im und einmal gegen den Uhrzeigersinn gesucht und jeweils geprüft, ob sich der Flächendefinitionspunkt innerhalb der gefundenen Umran-

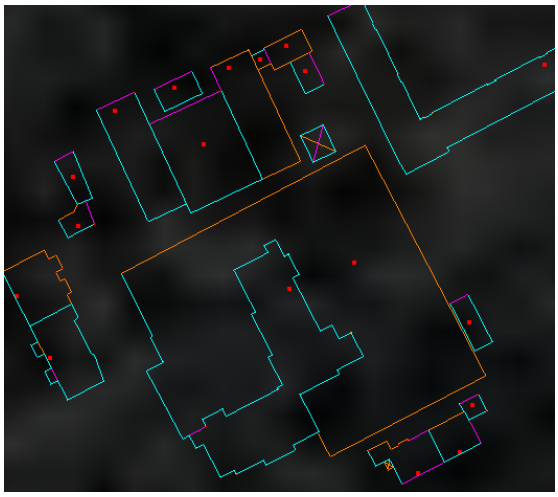

Abbildung 7.13: Probleme mit Gebäudegrundrissen: *cyan*: Linien, die direkt auf Flächendefinition folgen: aufgesammelt und verbunden; *orange*: Linien, die vorher definiert wurden; *rot*: Flächendefinitionspunkte; *violett*: Linien auf Hierarchie-Stufe 1; Bei zwei der gezeigten Flächen handelt es sich um Parkplätze.

dung befindet.
Neben den Gebäuden sind noch weitere Daten in den CAD-Plänen eingezeichnet. Diese Daten wurden vor allem für Visualisierungszwecke angelegt. Flussgrenzen z.B. sind unter Brücken nicht vorhanden. Beschriftungen sind nicht mit einem jeweiligen Objekt verknüpft sondern manuell an die passende Stelle positioniert worden. Straßen- und Bordsteinkanten sowie andere Straßenlinien sind einfache Begrenzungslinien und keine Flächen. So ist nicht bekannt, auf welcher Seite sich die Straße befindet.
Einige der eingezeichneten Symbole lassen sich jedoch besser nutzen. Leider wurden Masten nicht weiter differenziert, doch die eingezeichneten Bäume wurden benutzt und an ihrer Stelle mit Zufallsparametern versehene Bäume in das Modell aufgenommen.
Sehr interessant sind die eingezeichneten Straßenbahngleismittelachsen. Diese Linienzüge auf dem Straßenlayer zeichnen sich dadurch aus, dass sich in ihrer Nähe kennzeichnende Symbole befinden. Die Extraktion muss manuell unterstützt werden, da auch diese Linien z.B. unter Brücken aufhören. Auch Weichen und Kreuzungspunkte müssen evtl. manuell bearbeitet werden. Schließlich sollten Informationen über Fahrtrichtung, Straßenbahnliniennummern und Haltestellen angefügt werden. Die Braunschweiger Verkehrs-AG besitzt leider keinen georeferenzierten Liniennetzplan.
Daten für die Grundrisse der Gebäude im Bereich von Köln liegen im binären Shape-Format von ArcView vor. Für jeden Datensatz existiert eine Index-Datei (`*.shx`) und eine Shape-Datei (`*.shp`) mit den eigentlichen Polygonen. Das Einlesen der Daten mit Hilfe der *shapelib* ist unproblematisch.

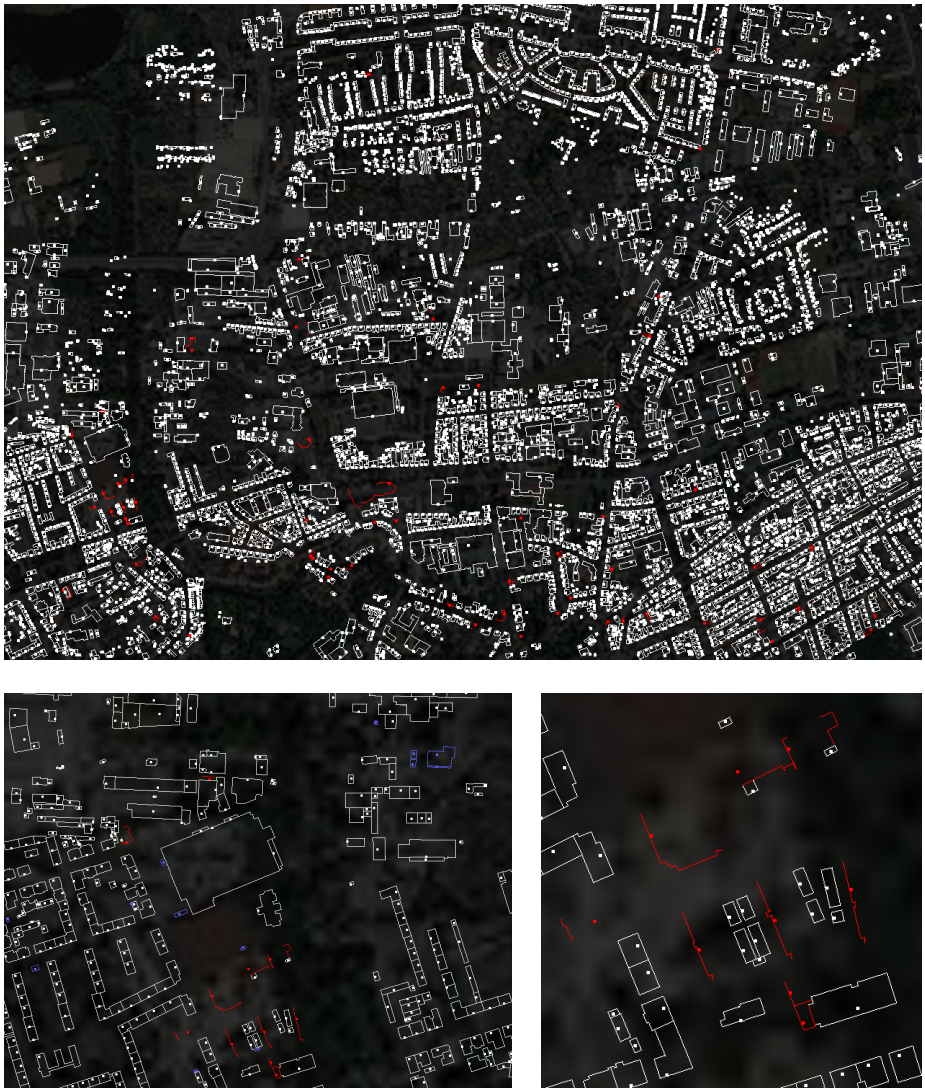

Abbildung 7.14: Probleme mit Gebäudegrundrissen: Übersicht und zwei Detailansichten: *weiß*: Fehlerlos erkannte Gebäude mit meist korrekten Grundrissen; *rot*: Gebäude ohne zusammenhängenden Grundriss; *blau*: Flächen ohne Schraffuren

7.3.4 Gebäude

Um Dachformen guter Qualität zu erhalten, muss eine aufwendige Extraktion programmiert werden, die aus Zeitgründen bei weitem nicht realisierbar war. So beschränkte ich mich auf Flachdächer mit einer durchschnittlichen Gebäudehöhe. Das vorerst benutzte Kriterium der maximalen Höhe innerhalb des Grundrisses führte teilweise zu größeren Fehlern, da über die Dächer ragende, große Bäume viel zu hohe Gebäude entstehen ließen (siehe Abb. 7.15).

Abbildung 7.15: Die Wahl der maximalen Höhe innerhalb des Grundrisses für die Gebäudehöhe des Blockmodells kann zu starken Artefakten führen wie bei diesem einstöckigen Gartenhäuschen neben hohen Bäumen, die in den Grundriss hineinragen.

Zur Texturierung der Dächer macht es nur Sinn, True Orthophotos zu benutzen, da andernfalls das Dach um mehrere Meter verschoben sein kann. Zweckmäßigerweise wird die gleiche Textur wie für die Erdoberfläche benutzt, doch Dächer auf Texturkachelgrenzen sind hierbei sehr problematisch. Eine Möglichkeit wäre die Benutzung sich überlappender Texturen, doch dafür muss eine feste Breite der Überlappung gewählt werden. Eine bessere Lösung ist daher, die Dachpolygone an den Texturkachelgrenzen zu unterteilen. Die Unterstützung komplexerer Gebäudeformen, die nicht dem Blockmodell entsprechen, ist momentan nur mit Hilfe einer OBJ-Datei pro Gebäude gegeben und nicht so gut für große Mengen dieser Objekte geeignet.

7.3.5 Manuell eingefügte Objekte

Für einzelne Projekte oder zur Demonstration ist es möglich, beliebige texturierte 3D-Objekte im Obj-Format in das Modell einzufügen. Diese Objekte können, wie auch die Kamera, mit Hilfe einer Nurbs-Kurve animiert werden.

7.3.6 Bodennutzungsdaten

Die Bodennutzungsdaten waren erst nach diversen Telefonaten und viel Geduld zu finden. Die Daten liegen im Vektorformat vor und wurden bereits aus einem anderen Format konvertiert. Wegen mehrfach übereinanderliegenden Polygonen, teilweise fehlenden Polygonen und der Tatsache, dass ich diese Daten sowieso aus Geschwindigkeitsgründen im Rasterformat besser gebrauchen kann, konvertierte ich die Daten manuell mit Hilfe von Screenshots in Kacheln mit

einer 2m-Auflösung (siehe Abb. 7.16). Genauer aufgelöste und differenziertere Bodennutzungsdaten wären für die Visualisierung natürlich wünschenswert, doch ist mit den so gewonnenen Daten schon einiges möglich.

Abbildung 7.16: Manuelle Korrektur fehlender Teilflächen

Die ursprüngliche Idee, Detailtexturen lokal abhängig von der Bodennutzungskarte zu benutzen, wurde aufgegeben, da dies beim Rendern der Erdoberfläche passieren muss. Die benutzte Terrain Engine ist aber dafür ungeeignet.

7.3.7 Texturierung der Erdoberfläche

Für einen einfachen Zugriff auf die Farbinformationen, wie er z.B. bei Konvertierungen gebraucht wird, reicht es nicht aus, die Daten für die Texturierung nur auf der Grafikhardware vorzuhalten. Zudem werden die Daten benötigt, falls sie zusammen mit anderen Rasterdaten kombiniert werden. Es ist also notwendig, Kacheln für die Luftfotos getrennt von den Kacheln für die Erdoberflächen-Texturen zu benutzen. Dadurch ist eine sehr flexible Texturierung möglich, doch ist diese Methode etwas langsamer als eine direkte Nutzung der Luftfotos als Textur. Für eine optimale Performance für den häufigen Fall der Texturierung mit Luftfotos ist die Benutzung von Bilddateien im bereits komprimierten Format der Grafikkarte interessant. So können die Daten direkt geladen werden, anstatt wie im Moment aus der JPG-Datei dekomprimiert, umkopiert und wieder anders komprimiert zu werden. Die Ausgangsdaten liegen als 5000x5000 Pixel große JPG-Dateien vor. Geringfügiges Downsampling und das Aufteilen in je 16 kleinere Kacheln halten die Ladezeiten im akzeptablen Rahmen. Für das Aufteilen wurden Batchdateien generiert, die ein Bildverarbeitungsprogramm mit den entsprechenden Kommandozeilenparametern aufrufen und die Koordinaten und Dateinamen passend berechnen.

7.3.8 LIDAR-Daten

Auch die Konvertierung der LIDAR-Daten war nicht besonders schwierig. Die 2x2 km^2 großen Kacheln im ASCII-Format mussten jedoch aus Geschwindigkeitsgründen in kleinere Kacheln zerlegt und im Rohdatenformat gespeichert

werden.

7.3.9 Datenstrukturen

Die umfangreiche Spezifikation des GML-Formats[Cona] vom Open Gis Consortium [Conb] habe ich nicht implementiert, da das Format recht komplex ist (das PDF-Dokument der Implementation Specification von GML 3.0 umfasst ca. 550 Seiten). Zudem war mir während der Implementierung noch nicht klar, ob dies das optimale Format sein würde. Auch ist die Formatkonvertierung des GIS nicht das Hauptziel der Diplomarbeit. Für langfristigen, professionellen Einsatz ist die Nutzung eines Standarddatenformats jedoch sinnvoll.
Das Exportieren der Stadtmodell-Geometrie in andere 3D-Formate macht meist nur für kleine Gebiete Sinn, da übliche Programme nicht mit den großen Datenmengen umgehen können. Das Blockmodell der Gebäude wurde für einige Quadratkilometer bereits als (`.genmod`)-Datei exportiert. Die Export-Funktion ist jedoch eher in einem experimentellen Stadium implementiert und noch nicht in das User-Interface integriert.

7.4 Visualisierung

Für das Culling wurden nur einfache Implementierungen benutzt. Als grobe Approximation eines Contribution Cullings wurden einfach abhängig von der Distanz zum *CoI* weit entfernte Objekte weggelassen. Ein Occlusion Culling ist nicht implementiert, das View Frustum Culling arbeitet mit Bounding Spheres. Dabei wurden nur zwei diskrete Hierarchiestufen zur Gruppierung der Objekte benutzt (pro Kachel und pro Objekt). Für das View Frustum Culling wird das Produkt aus Projektions- und Modelviewmatrix benutzt, so dass es auch für die Stereoprojektion und für die *DAVE* benutzt werden kann.
Für eine schönere Darstellung der Erdoberfläche wurde eine Detailtextur benutzt. Auch Höhenlinien wurden per Multitexturing ermöglicht. Dafür wurde eine eindimensionale Textur benutzt, deren Texturkoordinaten ebenfalls automatisch generiert werden.
Für eine bessere Orientierung kann sich der Benutzer eine Übersichtskarte einblenden lassen, in der auch das *CoI* eingezeichnet ist. Zudem sind die Himmelsrichtungen jeweils durch einen in der Luft schwebenden Text markiert. Auch können die Koordinaten und die Höhe des Punktes unter dem Mauszeiger angezeigt werden. Dies ist auch für Informationen über das Gebäude unter dem Mauszeiger möglich. Dazu gehören Straßenname, Hausnummer und Gebäudehöhe (siehe Abb. 7.17 und Abb. 7.18).
Für die Texturierung der Erdoberfläche können Luftfotos, Bodennutzungskarte und farbcodierte Höhendifferenzen zwischen DOM und DGM zusammen kombiniert werden. Zudem kann der Alphakanal der Textur an den Stellen modifiziert werden, wo Wasser in der Bodennutzungkarte eingetragen ist, um so eine reflektierende Wasseroberfläche zu simulieren. Dieses Kombinieren wird einmalig beim Erzeugen der Texturkachel vorgenommen. Im Gegensatz zur Benutzung von Multitexturing können dadurch beliebige Algorithmen zur Kombination der Farben benutzt werden und es gibt auch keine geschwindigkeitsbedingten Nachteile.

Abbildung 7.17: Eingeblendete Zusatzinformationen

Abbildung 7.18: Anwendungsbeispiel zur Distanzmessung: Mein Radweg zur Universität

Zuerst wurden zwei diskrete LoDs für die Lufttextur benutzt. Für Braunschweig handelt es sich um Auflösungen von gut 10cm und gut 10m pro Pixelkantenlänge, also den Faktor 100. Daher wurde später noch eine dritte Stufe (mit etwa 2m Auflösung) eingefügt.
Zur Auswahl der Datenvisualisierung stehen das DGM und das DOM im 1m-Raster, Bodendetails abhängig vom Bodennutzungstyp, die CAD-Daten und die extrahierten Stadtobjekte. Zu letzteren gehören Gebäude und Bäume sowie manuell eingefügte Objekte.
Die generierten Bodendetails sind nicht sehr überzeugend (siehe Abb. 7.19). Dies liegt u.a. an der recht generalisierten Bodennutzungskarte. Hier ist noch einiges an Arbeit notwendig, um gute Ergebnisse zu erzielen. Die Farbinformationen der Orthofotos sollten berücksichtigt werden. Dennoch wird die Idee des Ansatzes deutlich. Die Details werden hier für jeden Frame neu aus Zufallszahlen generiert. Um die zeitliche Kohärenz zu gewährleisten, wird der Zufallszahlengenerator für jeden Quadratmeter mit einer aus der Position berechneten Zahl initialisiert.

Abbildung 7.19: die automatisch generierten Bodendetails wirken unrealistisch

Um einfach zwischen verschiedenen Texturen wechseln zu können (z.B. für eine NPR-Darstellung, siehe Abb. 7.20), kann per Konfigurationsdatei das Verzeichnis für die Texturen angegeben werden.

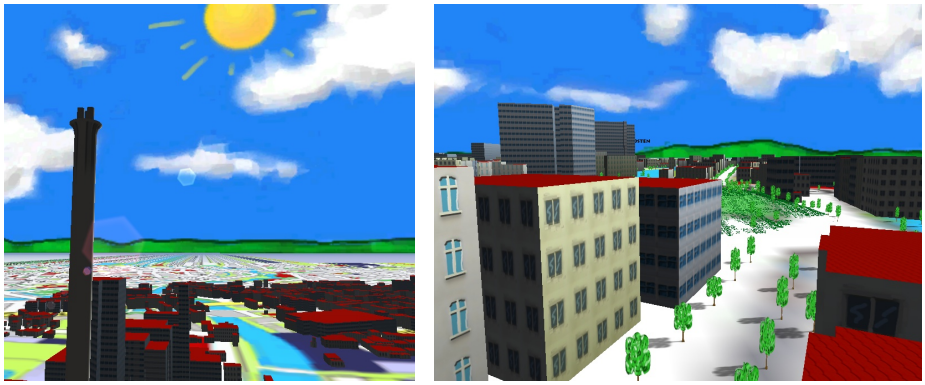

Abbildung 7.20: NPR-Texturen

Als Spezialeffekt wurden Lens Flares eingefügt (siehe Abb. 7.21).

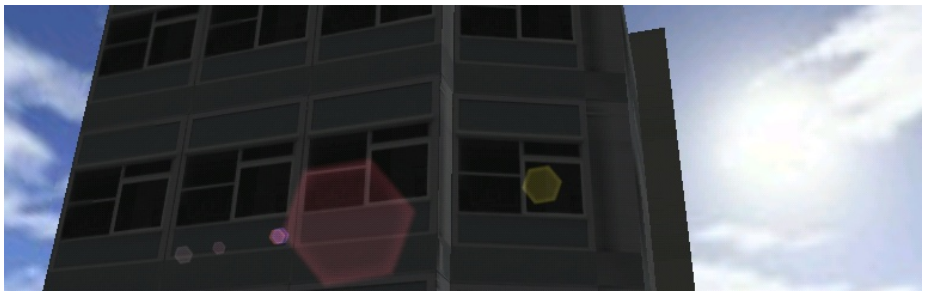

Abbildung 7.21: Lens Flares

7.5 Quelltext

Alle Kacheln sind von der Klasse `GeoDataArea` abgeleitet, die Daten und Funktionen für einen rechteckigen Bereich bereitstellt. Die Klasse `GeoDataRasterMap` ist davon abgeleitet und stellt zusätzliche Funktionen für die Verwaltung von Rasterdaten in diesem Bereich zur Verfügung.
Alle Kacheln werden mit Hilfe der Klasse `MapArray` verwaltet, also bei Bedarf geladen oder wieder freigegeben. Bei Konvertierungen werden Daten im angegebenen rechteckigen Bereich geladen, zur Visualisierung wird die minimale Entfernung zum *CoI* benutzt. Da es sich bei Kacheln um Rasterdaten handelt, ist die Klasse `MapArray` von der Klasse `GeoDataRasterMap` abgeleitet.
Die Klasse `CityModel` benutzt hauptsächlich diese Daten:

```
...
// Map Arrays: arrays with tiles
MapArray<TerrainTexture>* terrainTextureMapArray;
MapArray<TerrainTexture>* mediumTerrainTextureMapArray;
MapArray<AirPhoto>* airPhotoMapArray;
MapArray<GroundTypeMap>* groundTypeMapArray;

MapArray<TerrainEngineTile>* terrainEngineMapArray;
MapArray<HeightMap>* terrainHeightMapArray;
MapArray<HeightMap>* surfaceHeightMapArray;

MapArray<CadMap>* cadMapArray;
MapArray<CityObjects>* cityObjectsMapArray;


// single Maps
TerrainCoarseHeightMap* terrainCoarseHeightMap;
TerrainTexture* airPhotoCoarse;

...
```

Die Klasse `CityObjects` kann zur Zeit drei Arten von Objekten (Gebäude, Bäume, allgemeine Objekte) enthalten, die jeweils wieder als eigene Klassen implementiert wurden.

7.6 Externe Bibliotheken

Es wurden auch nicht selbst geschriebene Programmteile benutzt:

glut: für die Kapselung von OpenGL-Fenstern und Eingabegeräten

opengl-stereo: für den Stereo-Betrieb (z.B.: im Rückprojektionsraum)

frustum: für das korrekte view frustum culling

smartptr: für reference-counter basierte Pointer

ifsLibLight: für die Triangulierung von Polygonen

Mini: Terrain Engine

MeshViewer: als Framework für einen obj-Loader

cxImage: zum einheitlichen Lesen von png und jpg

shapelib: zum Lesen von shape-Dateien

tr: zum Erstellen von hochaufgelösten Bildern

7.7 Ergebnisse

Am Ende der Bearbeitungszeit wurden Daten der Stadt Braunschweig vom gesamten Gebiet zur Verfügung gestellt. Dabei handelt es sich um eine Fläche von gut 200 km^2. Da sich bei den Grundrissdaten wieder das Dateiformat geändert hatte, musste hier der Quelltext erneut angepasst werden. Die anderen Konvertierungen verliefen fast problemlos. Eine Geschwindigkeitsoptimierung für die Höhendaten konnte wegen der unregelmäßigen Datengrenzen der Daten nicht immer benutzt werden. Auch wurden drei kleinere Fehler gefunden, die im Testgebiet nicht aufgetreten waren.
Für die Konvertierung der Gebäude wurde nun die durchschnittliche Höhe für die Dachhöhe benutzt. Für eine etwas realistischere Darstellung der Dächer wurde nun auch die durchschnittliche Dachfarbe aus den Luftbildern extrahiert und bei den Gebäudedaten abgespeichert (siehe Abb. 7.22).
Entwicklungs-PCs waren durchschnittliche Modelle mit 350 MHz bzw. 900 MHz. Die Konvertierung erfolgte auf dem zuletzt genannten System und auch die Visualisierung läuft darauf bereits zufriedenstellend. Schnellere Systeme können aber auch voll ausgenutzt werden, da hier Daten schneller geladen werden können und der sichtbare Bereich vergrößert werden kann.
Einen Überblick über die Gesamtkonvertierungszeiten auf dem 900 MHz-System und Größenordnungen der wichtigsten automatisch konvertierten Daten gibt die folgende Tabelle.

	Originaldaten	Konv.-zeit	konvertierte Daten
Orthofotos	4.1 GB, 900 Dateien	1,5 d	3 GB, 14000 Dateien
Höhendaten	7.3 GB, 70 Dateien	6 d	1.3 GB, 22000 Dateien
Katasterdaten / Gebäude, Bäume	1.6 GB, 70 Dateien	3 h	70 MB, 340 Dateien

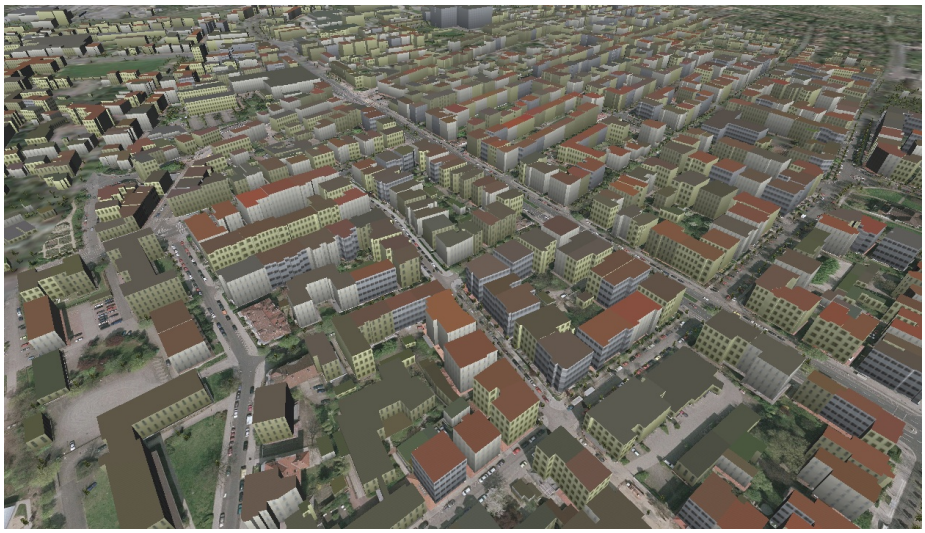

Abbildung 7.22: Ausschnitt des kompletten Datensatzes von Braunschweig.

Für Köln standen keine hochaufgelösten Höhendaten zur Verfügung, auch waren die vollständigen Informationen über Gebäudegrundrisse nur für einen kleinen Bereich von Leverkusen vorhanden. Daher können nur diese Grundrisse, eine korrekte Skybox und die Luftfotos auf den groben Höhendaten anzeigt werden. Dennoch wird daraus ersichtlich, dass der *CityModelViewer* für verschiedene Städte benutzt werden kann.

Kapitel 8

Ausblick

Es wurde gezeigt, dass es eine große Menge von Anwendungen für 3D-Stadtmodelle gibt. Um die sinnvolle Weiterentwicklung von Verfahren zu ermöglichen, sind Standardisierungen notwendig, damit diese einfach für viele unterschiedliche Städte benutzt werden können. Erst wenn Forscher sich nicht mehr um das ganze Modell kümmern müssen, kann eine richtige Spezialisierung für die einzelnen Teile schnelle Fortschritte machen. Meine Implementierung zeigt, dass schon heute mit relativ einfachen Mitteln die Daten für ein 3D-Stadtmodell genutzt werden können. Hauptsächlich die Finanzierung dieser Daten spricht gegen die Möglichkeit einer verbreiteten Nutzung. Anforderungen für den privaten Gebrauch werden sich erst in Zukunft zeigen.
Für die professionelle Verwaltung der Daten ist langfristig ein ganz anderes System erforderlich, als das von mir benutzte. So müssen die konvertierten Daten mit im GIS abgelegt und verarbeitet werden. Programme können dann mit allgemeinen Schnittstellen auf die Daten zugreifen und direkt oder für Exporte nutzen. Die Ergebnisse können teilweise wieder als neue Objekte in das Modell einfließen, wie z.B. bei Lärmimmissions-Simulationen. Die Stadt Braunschweig plant den Einsatz von ArcGIS, um das jetzige GIS von Sicad abzulösen. Dies wird im Vergleich zur derzeitigen Situation einiges vereinfachen. Abb. 8.1 zeigt den Vorschlag für ein solches zukünftiges System.
In dieser Arbeit wurde versucht, die bereits vorhandenen Datenquellen gut auszunutzen. Interessant für weitere Anwendungen ist die Untersuchung, welche weiteren Daten nützlich wären, um das Modell den nächsten Anforderungen entsprechend zu verbessern. Für den Fall Braunschweig wären meiner Ansicht nach schräg aufgenommene Luftbilder und die komplette Erfassung des Straßenverkehrsnetzes inklusive der Fahrbahngrenzen sehr hilfreich. Die Bewertung der Wichtigkeit ist jedoch sehr stark von den individuellen Interessensgruppen abhängig. Die zur Verfügung stehenden finanziellen Mittel werden dabei entscheidend sein.

8.1 Mögliche Verbesserungen

Die aktuelle Implementierung bietet noch viele Verbesserungsmöglichkeiten. So könnten die Daten schneller und somit für einen größeren Bereich gezeigt werden, realistischere Ansichten erzeugt und weitere Objekte angezeigt werden.

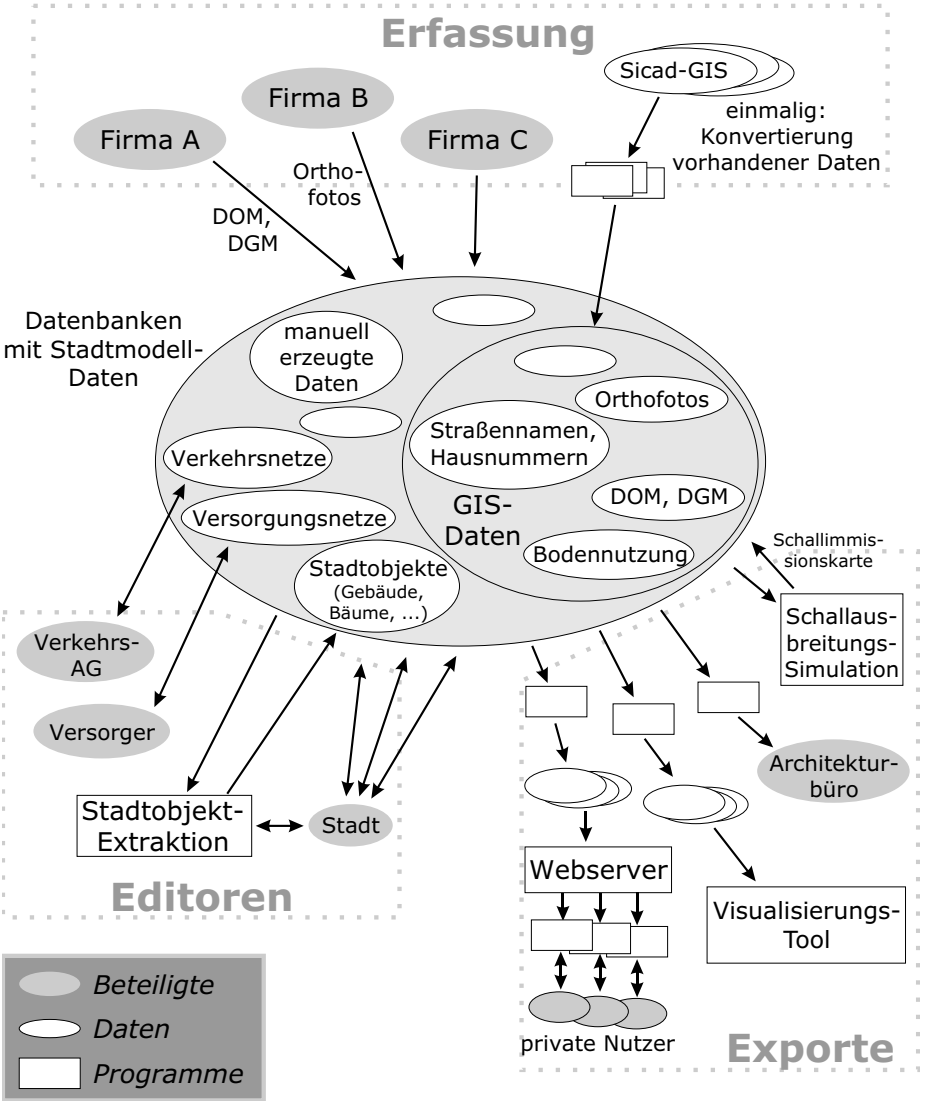

Abbildung 8.1: Vorschlag für ein zukünftiges System

Für konkrete Anwendungen sind weitere Verbesserungen gerade im Bereich des User-Interfaces nützlich.
Das Konvertierungs-Tool für die Stadtobjekte ist sehr eng mit dem *CityModelViewer* gekoppelt, doch handelt es sich um ein eigenes Programm, damit der Bereich bestimmt und die Dateinamen der Originaldaten automatisch generiert werden können. Eine bessere Lösung wäre die Bereitstellung eines User-Interfaces zur Angabe von Konvertierungsoptionen in Verbindung mit einem Programm, das die Dateinamen der Originaldateien automatisch in ein einheitliches Format umbenennen kann.
Für einen professionellen Einsatz ist eine bessere Softwaretechnik und -architektur notwendig, vor allem wenn mehrere Personen daran arbeiten. Die Klassenstruktur sollte im Hinblick auf eine bessere Übersichtlichkeit und klar definierte Schnittstellen überarbeitet werden.

8.1.1 Momentane Einschränkungen

Zur Zeit wird nur die Benutzung eines Koordinatensystems unterstützt. Alle benutzten Datenquellen müssen in diesem System und bei Meridianstreifensystemen auch im gleichen Streifen vorliegen. Die Kacheln müssen je Datensatz alle rechteckig sein und die gleiche Größe haben, was aber in der Praxis keine Einschränkung darstellt. Das Benutzen einer Skybox, in der etwa 20 km entfernte Hügel enthalten sind, stellt eine Einschränkung auf den Stadtbereich dar, da außerhalb die gleiche Skybox angezeigt wird und der Unterschied zum echten Erscheinungsbild sehr groß ist. Sollen größere Flächen bis hin zur kompletten Weltkugel angezeigt werden, reicht auch die Benutzung einer 2D-Projektion nicht mehr aus und alle Koordinaten müssen spätestens vor der Darstellung noch einmal transformiert werden. Wegen der vielen Ziffern der Koordinaten des Gauß-Krüger-Systems gibt es schon jetzt eine Transformation vor der Übergabe der Zahlen an OpenGL in ein lokales Koordinatensystem, um die stark sichtbaren Rundungsfehler zu vermeiden, die sonst durch die Benutzung von Float-Matrizen und -Berechnungen in der Grafikhardware auftreten würden. Abgesehen davon gibt es eher praktische Limitationen wie die Größe des verfügbaren Datenspeichers und die unterschiedlichen Datenformate der jeweiligen Landkreise.

8.1.2 Geschwindigkeitsoptimierungen

Der *CityModelViewer* läuft schon jetzt akzeptabel auf durchschnittlichen PCs. Profiling wurde noch kaum eingesetzt, da die Ursachen vieler Geschwindigkeitsprobleme offensichtlich waren. Es sollte jedoch benutzt werden, um weitere besonders langsame Programmteile zu identifizieren.
Im Bereich der Schnittstelle zu OpenGL kann noch einiges schneller werden. Die konsequente Nutzung von Vertex-Arrays bzw. Vertex Buffers, Display-Listen und Exensions würde die Framerate noch z.T. deutlich erhöhen.
Ein viel schnelleres Laden der Daten ist auch möglich. Datenformate könnten direkt für Visualisierung erzeugt werden, so dass die Daten direkt in den Speicher geladen werden können. Dazu muss eine weitere Trennung der Daten in eine allgemeine Repräsentation und für die Visualisierung erfolgen. Beispiele sind die Orthofotos oder Höhendaten, die als Originaldaten einmal in hoher Qualität für eine Verarbeitung zur Verfügung stehen, für die Visualisierung aber in ein schnell

zu ladendes, stärker komprimiertes Format umgewandelt werden, aus dem jeweils nur die Daten in der benötigten Auflösung gelesen werden brauchen. Noch nützlicher ist der Einsatz von eigenen Threads zum Laden der Daten, so dass die Ansicht nicht mehr beim Nachladen blockiert wird. Bei der Datenübertragung über das Internet ist dies noch viel wichtiger, da es sich hier um noch längere Ladezeiten handelt. Die Latenz kann in diesem Fall durch eine progressive Darstellung weiter minimiert werden, indem bei erst teilweise geladenen Daten die schon verfügbaren Informationen bereits angezeigt werden. Sofern noch Rechenzeit zur Verfügung steht, könnten detailliertere Daten im weiteren Umkreis dazugeladen werden, so dass sich die Qualität der Szene immer mehr steigert, sobald die Kamera sich nicht mehr schnell bewegt.
Es wurden für die Objekte nur wenige oder ein LoD programmiert. Durch bessere LOD-Techniken kann der darstellbare Radius für eine interaktive Geschwindigkeit noch erweitert werden. Ein jeweils optimales, adaptives, dynamisches LoD für jedes dargestellte Objekt bzw. Objektgruppen muss implementiert werden, um die komplette Szene ein Echtzeit darzustellen.
Szenengraph und Quadtrees sollten zur hierarchischen Gruppierung genutzt werden, z.B. um die Geschwindigkeit des Cullings zu verbessern.
Einige der Geschwindigkeitsoptimierungen machen den Code unleserlich und verkomplizieren dessen Benutzung und Portabilität. Eine saubere, allgemeine Lösung kann viel Zeit kosten, weshalb ich in vielen Fällen darauf verzichten musste.

8.1.3 GUI und Editoren

Es ist noch kein komfortables User-Interface integriert worden. Je nach Anwendung sind viele verschiedene nützliche Elemente und Funktionen vorstellbar. Es ist durchaus sinnvoll, das Programm auch zur Modifizierung bestimmter Daten zu benutzen, z.B. zum Einfügen von Verkehrsschildern in das Stadtmodell. Für alle benutzten Daten sollten Editoren zur Verfügung stehen; oft ist es dabei hilfreich, wenigstens gleichzeitig das Orthofoto einzublenden. Für selbstentwickelte Tools kann man evtl. ein Plugin-Konzept in Erwägung ziehen, bei dem GUI-Elemente und Stadtmodelldaten benutzt werden können.

8.1.4 Daten

Die Trennung der Daten einmal zur Visualisierung und andererseits für die Verwaltung wurde bereits angesprochen. Letztere sollten möglichst universell und portabel sein. Die Möglichkeit, alle nötigen Informationen in nur einer einzigen Datenbank zu verwalten, ist für die nächste Zeit nicht absehbar. Die Verknüpfung mehrerer Datenbanken über Objekt-IDs erscheint momentan am besten. Eine einheitliche Abfragemöglichkeit wie SQL ist wünschenswert.
Die Nutzung von OGC-Standards ist hier sehr zu empfehlen. Um die gleichen Daten mit vielen Benutzern verwalten zu können, ist z.B. der Einsatz eines CVS-ähnlichen Systems denkbar. Allerdings sollte die Vergabe und Einschränkung von Rechten der Benutzer für bestimmte Attribute möglich sein. So sollten nur wenige Nutzer das Recht haben, den Grundriss eines Gebäudes zu ändern. Jede Gruppe von Nutzern könnte eine lokale Kopien von den notwendigen Informationen haben und diese gelegentlich mit dem zentralen Datenbestand synchronisieren. Zur Bearbeitung einiger Daten empfiehlt sich der Einsatz einer

professionellen Datenbank. Andere Informationen könnten über ein GUI in der 3D-Ansicht eingegeben werden.

8.1.5 Projektvorschläge

Weiterführende studentische Arbeiten könnten z.B. in folgenden Bereichen stattfinden:

Einfügen von kleineren Objekten und Straßenmobiliar:
Laternen, Masten und Hochspannungsleitungen, Ampeln, Verkehrsschilder, Parkbänke, Pflanzen, Springbrunnen, Denkmäler, Kunstwerke

Einfügen von Verkehrsnetzen:
für Autos, Straßenbahnen, Eisenbahnen, Schiffe, Flugzeuge

Nutzung dieser Verkehrsnetze:
mit künstlicher Intelligenz zur Simulation von Verkehr

Echtzeitnutzung von aktuellen Daten:
freie Parkplätze im Parkhaus, Fahrpläne und Fahrzeugpositionen des ÖPNV, Lokalnachrichten

Geschwindigkeitsoptimierungen:
LoDs für alle Objekte, Multithreading und der Einbau einer anderen Terrain Engine, Nutzung von weiteren OpenGL Extensions

Gebäudedetails:
Extraktion von detaillierteren Dachformen und Fassadentexturierung mit realen Fotos

Netzwerkbasierter Datenzugriff:
für Internetanwendungen

Datenformate:
Nutzung von OGC-Standards, weiter Exporte z.B. für Maya

8.2 Zukünftige Anwendungsmöglichkeiten

Wozu braucht man ein 3D-Stadtmodell? Es ist sicherlich nicht unverzichtbar, kann aber sehr praktisch sein.
Im Idealfall könnte ein System professionell eingesetzt werden und gleichzeitig als Forschungsplattform dienen. So kann man die aktuellen Probleme lösen und die Forschung in die Praxis umsetzen.
Bisher existiert für die Stadt Braunschweig nur ein grobes Holzmodell für den Innenstadtbereich. Bei der Stadtplanung werden für Projektvorschläge einzelne Holzklötze durch detailliertere Gebäudemodelle ersetzt. Diese Art des Modells hat durchaus Vorteile. So können es mehrere Benutzer gleichzeitig von allen Seiten betrachten, darüber diskutieren und einfache Modifizierungen direkt vornehmen. Allerdings ist das Anfertigen der physischen Modelle aufwändig. Eine Ansicht aus der Fußgängerperspektive ist nicht möglich. Eine CAVE könnte viele Vorteile vereinen, hat aber den Nachteil der relativ hohen Kosten.

Heutige Modelle, die man z.B. im VRML-Format im Internet ansehen kann, sind eher eine technische Spielerei als ein nützliches Werkzeug.
Ernsthafte Anwender sind in naher Zukunft noch kleine Gruppen im professionellen Bereich, doch ein alltäglicher Einsatz für die Öffentlichkeit ist absehbar. Potenzielle Einsatzmöglichkeiten sind hier leicht zu erkennen. Bei jedem Griff zum Stadtplan oder den gelben Seiten könnte in Zukunft auch ein 3D-Stadtmodell benutzt werden. Für ortsbezogene Informationen ist eine Suchmaschine vorstellbar, die einfach zu bedienen ist und GIS-Funktionalität zur Verfügung stellt.
Für den Alltagseinsatz muss ein Stadtmodell schnell und unkompliziert verfügbar sein. Es sollte mit PCs, PDAs und auch Handys benutzt werden können, um z.B. den nächsten Bäcker zu finden, die Öffnungszeiten eines Kaufhauses anzuzeigen oder den besten Weg vom aktuellen Standort zum gewünschten Ziel anzuzeigen. Bei einem mobilen Einsatz ist die Bestimmung der eigenen Position mit GPS bzw. Galileo besonders interessant. Die Anzeige von Informationen mit Hilfe von Augmented Reality ist ein weiterer Schritt.
Die Erfassung der vielen notwendigen Daten stellt ein großes Problem dar, da viele detaillierte Informationen nur manuell eingegeben werden können. Eine Idee ist die freiwillige Eingabe der Daten jeweils durch die Eigentümer. Auf Wunsch können so bessere Fassadenfotos, Links, Telefonnummern und Adressen angeben werden. Evtl. sollte der User auswählen können, ob er nur die offiziellen Daten oder auch die zusätzlichen Annotationen der Besitzer sehen will. Für den Fall der Fassadenbilder wären automatische Plausibilitätsprüfung und Farbanpassung notwendig, um sicherzustellen, dass diese auch mit der Realität übereinstimmen und sich natlos in das Modell einfügen. Eine weitere Möglichkeit ist die Idee, dass ähnlich des *Wiki*-Prinzips jeder registrierte Nutzer prinzipiell das Recht hat, alle Zusatzinformationen zu ändern. Der Erfolg ist fraglich, zumindest sind Moderatoren notwendig, die die Änderungen überwachen.
Bei diesen schönen Visionen darf man aber wirtschaftliche Machbarkeit und die Kosten-Nutzen-Bilanz nicht aus den Augen verlieren, denn das Hauptproblem für den Einsatz von 3D-Stadtmodellen ist die Finanzierung der Daten. Anwender mit den notwendigen Mitteln werden also zuerst davon profitieren. Für eine breite Nutzung sollten die Daten aber kostenlos erhältlich sein. Erst ab einem bestimmten Prozentsatz von Benutzern werden sich viele Anwendungen lohnen und dadurch das Modell tragen können. Wenn diese Grenze überschritten ist, könnten z.B. durch Werbung im Modell weitere Funktionen oder Details finanziert werden.
Je größer die Gebiete sind, für die die Technik eingesetzt werden kann, desto mehr lohnen sich auch Software-Entwicklungen. Landschafts- und später Weltmodelle größtenteils in der Auflösung der heutigen Stadtmodelle könnten entstehen. Einige Gebiete könnten sogar noch genauer modelliert werden, so dass genauere Fassaden oder auch Innenräume eingebaut werden.

Kapitel 9

Anlagen

Auf den beiliegenden DVDs befinden sich Programme und Quelltexte, konvertierte Daten, Screenshots und ein Video, die Vortragsfolien sowie diese Diplomarbeit in elektronischer Form inklusive Hyperlinks.

Danksagungen

Ich danke Allen, die mich bei dieser Arbeit durch geduldige Beantwortung von Fragen, Bereitstellung von Quellcode und Ressourcen sowie Anregungen unterstützt haben.
Auch danke ich der Stadt Braunschweig, Abteilung Geoinformation, für die Bereitstellung der Daten des gesamten Stadtgebietes.

Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit selbstständig und nur unter Verwendung der aufgeführten Hilfsmittel von mir erstellt wurde.

Braunschweig, den 26. März 2004

Unterschrift

Literaturverzeichnis

[AG01] Bruce Gooch Amy Gooch. *Non-Photorealistic Rendering*. AK Peters, 2001.

[Bre00] Claus Brenner. Dreidimensionale gebäuderekonstruktion aus digitalen oberflächenmodellen und grundrissen. Technical Report Deutsche Geodätische Kommission, Reihe C, Nr. 530, München 2000, Institut für Photogrammetrie, Universität Stuttgart, http://www.ifp.uni-stuttgart.de/, 2000.

[Cona] Open Gis Consortium. Opengis geography markup language (gml) implementation specification, website, http://www.opengis.org/.

[Conb] Open Gis Consortium. Website, http://www.opengis.org/.

[DDSD03] Xavier Décoret, Frédo Durand, François Sillion, and Julie Dorsey. Billboard clouds for extreme model simplification. In *Proceedings of the ACM Siggraph*. ACM Press, 2003.

[Deb96] Paul E. Debevec. *Modeling and Rendering Architecture from Photographs*. PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley CA, 1996.

[DSSD99] Xavier Décoret, Gernot Schaufler, François Sillion, and Julie Dorsey. Multi-layered impostors for accelerated rendering. In *Computer Graphics Forum (Proc. of Eurographics '99)*, Sep 1999.

[Duca] Mark A. Duchaineau. http://www.eingana.com/.

[Ducb] Mark A. Duchaineau. Roam: Realtime optimally-adapting meshes, http://www.cognigraph.com/roam_homepage/.

[FZ03] C. Früh and A. Zakhor. Constructing 3d city models by merging ground-based and airborne views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages p. II–562 – 69, 2003.

[Ish00] Toru Ishida. Digital city kyoto: Social information infrastructure for everyday life, 2000.

[kSH] Helsingin kaupunki (Stadt Helsinki)). Virtual helsinki, http://www.virtualhelsinki.net/.

[Muh96] Karrima Muhammad. Numerical simulation of environmental fluid flow: Contaminant dispersion in a city, 1996.

[NH99] C. Brenner Norbert Haala. Extraction of buildings and trees in urban environments. Technical Report International Archives of Photogrammetry and Remote Sensing 54, S. 130-137, Institut für Photogrammetrie, Universität Stuttgart, http://www.ifp.uni-stuttgart.de/, 1999.

[Pol99] Marc Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, ESAT-PSI, K.U.Leuven, 1999.

[Sha00] Babak Ameri Shahrabi. Automatic recognition and 3d reconstruction of buildings through computer vision and digital photogrammetry. Technical Report Deutsche Geodätische Kommission, Reihe C, Nr. 526, München 2000, Institut für Photogrammetrie, Universität Stuttgart, http://www.ifp.uni-stuttgart.de/, 2000.

[Sys] GeoSim Systems. Geosim cities, http://www.geosimcities.com.

[Tri] Trident. Website, http://www.trident3d.net/.

[Wor] Binary Worlds. Descensor engine (procedural world generation), http://www.binaryworlds.com/products.html.

[WWSR03] Peter Wonka, Michael Wimmer, François Sillion, and William Ribarsky. Instant architecture. *ACM Transactions on Graphics*, 22(4):669–677, july 2003. Proceedings of SIGGRAPH 2003.