



Integration von 2D- und 3D-Geodaten am Beispiel der dynamischen Generierung virtueller Stadttouren

Arne SCHILLING und Alexander ZIPF

European Media Laboratory EML, Heidelberg, Germany

Arne.Schilling@eml.villa-bosch.de Alexander.Zipf@eml.villa-bosch.de

Zusammenfassung

Im Rahmen der dynamischen Erzeugung von 3D-Welten spielt insbesondere die Problematik der automatisierten Integration von 2D und 3D-Daten eine große Rolle. Auf einige der damit verbundenen Aspekte wird in diesem Artikel eingegangen und Mechanismen, sowie der realisierte Prototyp zur automatischen Generierung integrierter 3D-Datensätze vorgestellt. Dabei wird zwischen den unterschiedlichen Problemen und Ansätzen bei Gebäude- und Geländemodellen, sowie weiteren GIS-Daten unterschieden. Darauf aufbauend wird eine Strategie der Erzeugung von 3D-Besichtigungstouren basierend auf diesen integrierten Datensätzen dargestellt. So werden Komponenten vorgestellt, die eine virtuelle 3D-Tour durch Heidelberg mit aus 2D- und 3D-Datenquellen automatisch integrierten und generierten 3D-Geodaten ermöglicht. Dem Besucher wird dabei eine dreidimensionale und z.T. interaktive Visualisierung einer berechneten Tour vermittelt, indem der Benutzer durch das hybride 3D-Modell von Heidelberg geführt wird.

1 Einführung

Ein großes Hindernis für die 3D-Visualisierung von Landschaften oder Stadtmodellen stellt die in der Regel ungenügende Datengrundlage dar. Die Erstellung detailgetreuer Modelle ist aufwendig und muss meist manuell erfolgen. Verfahren für eine automatische Erfassung ganzer Stadtteile durch Laserscans oder Photogrammetrie befinden sich in der Entwicklung (BRENNER und HAALA 2000; EVANS und HUDSON-SMITH 2001). Solche Modelle werden z.B. für die Mobilfunkplanung eingesetzt. Nicht nur, aber gerade für großflächige Untersuchungen ist es daher wünschenswert, auf bestehende 2D-Daten, wie sie in GIS oft schon vorgehalten werden, zurückzugreifen und diese als eine der Datengrundlage für die 3D-Visualisierung zu nutzen. Es stehen also mittlerweile verschiedene Geodaten digital zur Verfügung: Karten, Geländemodelle, 3D-Gebäude (CAD). etc. Diese liegen jedoch in unterschiedlichen Formaten vor oder sind verschieden referenziert. Damit besteht verstärkt die Notwendigkeit die erstellten 3D-Modelle mit umfangreichen 2D-Vermessungsdaten zu integrieren. Eine erste Fragestellung ist daher, die unterschiedlichen Geodaten so zu verwalten und zu integrieren, dass ein gemeinsamer Zugriff für die eine 3D-Visualisierung möglich wird. Hierzu müssen zweidimensionale Geodaten interpretiert und in die dritte Dimension überführt werden. Die solchermaßen generierten „2,5D-Modelle“ werden mit echten 3D-Modellen über einen Verschneidungsmechanismus fusioniert. Insbesondere werden auch 3D-Ansichten von Gebieten realisiert, zu denen bisher nur zweidimensionale

Daten in einem OpenGIS-konformen Geodatenserver vorliegen. Wünschenswert ist hierbei ein möglichst großer Automatisierungsgrad. Insbesondere muss auf die unterschiedlichen Datenquellen je nach Einsatzzweck dynamisch und transparent zugegriffen werden können, um Virtual Reality Modelle zu erzeugen, die z.B. interaktive 3D-Teilansichten eines Stadtmodells darstellen. Die 2D-Geodateninfrastruktur sollte dabei möglichst unverändert bleiben, da in der Regel diverse Fachanwendungen auf diese zugreifen. Ein Lösungsansatz ist die Verwendung eines Virtual Reality-Servers (VR-Servers), der über Schnittstellen (Loader) zu verschiedenen 2D- und 3D- Geodatenquellen verfügt. Räumliche Anfragen werden für die jeweiligen Datenquellen umgeformt und an diese weitergeleitet. Die Antworten werden gesammelt und in Echtzeit zusammengeführt.

Eine zweite Fragestellung dreht sich um die Anwendung des VR-Servers für die dreidimensionale Tourenvisualisierung, die über das Internet zur Verfügung gestellt werden soll. Eine Tourenplanungskomponente berechnet Touren in Form einer Abfolge von Routensegmenten, Abbiegeregeln und Haltepunkten. Das Ergebnis wird bisher in Kartenform präsentiert. Die entwickelten Komponenten lassen den Benutzer in ein virtuelles, z.T. interaktives Modell eintauchen und führen ihn entlang der Tour durch das Modell. Somit kann bereits vor dem Aufenthalt in der Stadt ein Eindruck über die architektonische Charakteristik und die Atmosphäre gewonnen werden.

Die vorgestellte Arbeit wurde im Rahmen des Projektes Deep Map (MALAKA und ZIPF 2000) durchgeführt, das vom European Media Laboratory (EML) am Beispiel der Stadt Heidelberg entwickelt wird. Es handelt sich um ein Touristeninformationssystem mit unterschiedlichen Benutzungsschnittstellen (WWW, VR, mobil). Als Vorarbeiten wurden für Deep Map neben Tourenplanungskomponenten schon sowohl ein OpenGIS-konformer Geodatenserver für 2D-Daten (ZIPF und ARAS 2001), als auch eine (temporale) 3D-Datenbank realisiert (ZIPF und KRÜGER 2001a, b), die als Basis verwendet werden konnten. Das 3D-Datenmodell basiert dabei auf dem von FLICK (1998) veränderten Modell nach Molenaar (1990). Die Implementierungsbasis bildet ein 3DServer von COORS (1997, 2001), der für die spezifischen Anforderungen erweitert und verändert wurde.

2 Datengrundlagen und –integration

Die Integration von Geodaten zur 3D-Visualisierung kann als ein dreistufiger Prozeß betrachtet werden:

1. Konvertierung von 2D Daten zu 3D Daten
2. Umrechnung auf ein einheitliches SRS
3. Verschneidung (Identifikation sich entsprechender Objekte in den Datengrundlagen)

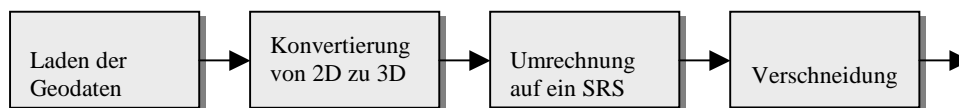


Abb. 1: Teilprozesse der Datenintegration

Wie erwähnt stellt die ungenügende Datenlage oft ein bedeutendes Hindernis für ein flächendeckendes 3D-Stadtmodell dar. In Heidelberg wurde der zentrale Altstadtbereich mittels einer Laserscan-Befliegung dreidimensional erfasst und liegt als ODF-, bzw. teilweise texturierte VRML-Datei vor. Das Modell wurde vom Institut für Photogrammetrie (IfP) der Universität Stuttgart erstellt. Dabei kam ein vom IfP entwickeltes Verfahren zum Einsatz, das eine weitgehend automatische Generierung von Stadtmodellen ermöglicht (BRENNER und HAALA 2000). Die zweidimensionalen Gebäudegrundrisse (ALK), sowie Strassen und Baublöcke etc. sind für das gesamte Stadtgebiet verfügbar. Um auch diese für die 3D Visualisierung zu nutzen, werden zu den zweidimensionalen Punkten, Linien und Polygonen Höhenwerte hinzugefügt. Die Höheninformationen stammen dabei entweder aus der Datenquelle selbst und werden als Attribute mitgeliefert oder werden dem DGM entnommen. Für die Gebäude Heidelbergs wurden bei einer Geländebegehung des Geographischen Institutes die Stockwerkszahlen erhoben und zur Verfügung gestellt (WINKLER 1999). Die Höhe der Gebäudeunterkante stammt aus dem DGM. Mit Hilfe dieser Informationen kann aus einem Polygon über eine Extrusion ein Blockmodell bzw. ein Prisma erzeugt werden, dessen Dimensionen in ausreichender Genauigkeit der Realität entsprechen.

2.1 Generierung von 3D aus 2D Geodaten

Ein typisches Beispiel für die Erstellung von 3D-Geodaten ist die Generierung von einfachen Blockmodellen aus Gebäudegrundrissen. Liegen die Daten entsprechend aufbereitet vor ist dies heute über Werkzeuge wie z.B. den ESRI ArcView 3D Analyst in wenigen Minuten per Hand zu realisieren. LANGE (1999) beschreibt bereits ein Verfahren, bei dem Gebäude und Bäume weitgehend automatisch aus 2D GIS-Daten erzeugt und in ein digitales Landschaftsmodell integriert werden. Unser Ansatz bedarf jedoch einer dynamischen Erzeugung aus heterogenen Datenquellen über eine API. So werden die 2D-ALK-Daten in einem Geodatenserver auf Basis von ESRI SDE (Spatial Database Engine) verwaltet und auf diese über eine in Java realisierte OpenGIS konforme CORBA-Schnittstelle zugegriffen. Um dies zu realisieren, müssen zunächst die Höheninformationen gewonnen werden. So wurden die für die Gebäude erhobenen Stockwerkszahlen in Meter umgerechnet. Die Höhe der Gebäudeunterkante über NN wird aus dem DGM abgeleitet, indem von allen Punkten des Grundriss-Polygons die minimale Höhe auf dem DGM berechnet werden. Beide Informationen stehen nach dem Laden in dem Geodatenserver auf Basis von SDE als Attribute zur Verfügung. Der VR-Server errechnet daraus ein Blockmodell, indem er den Grundriss entlang der z-Achse extrudiert und in die richtige Höhenlage verschiebt. Trotz der Einfachheit des Blockmodells liefert es bei kleinen Maßstäben schon gute Ergebnisse, da die Dimensionen ungefähr der Realität entsprechen.

Neben den Gebäudegrundrissen sind in einem GIS auch Geodaten vorhanden, die von Natur aus keine vertikale Ausdehnung besitzen und die bei der Konvertierung ihre eigentliche Dimensionalität behalten sollen (z.B. Waldflächen, Felder, Strassen, Eisenbahnlinien usw.). Deren Integration in ein 3D-Modell gestaltet sich schwieriger:

Linienzüge müssen an den Oberflächenverlauf des DGM angepasst werden, wobei es nicht ausreicht, jeden Punkt um einen z-Wert zu erweitern. Sondern es muss bei jeder Kreuzung eines Liniensegments mit einer Kante des DGM ein neuer Punkt eingefügt werden.

Flächen müssen in das DGM „hineingeschnitten“ werden, indem die Triangulation und die Topologie so angepasst werden, dass zwischen Dreiecken innerhalb und außerhalb der

Fläche unterschieden werden kann. Sie werden also zu Bestandteilen des DGM und sichtbar, wenn die entsprechenden Dreiecke farblich markiert werden.

Es lassen sich bei der Konvertierung also die folgenden Möglichkeiten unterscheiden:

- 1 a) die Objekte besitzen eine vertikale Ausdehnung, diese ist als Attribut gespeichert,
b) die Objekte besitzen keine vertikale Ausdehnung,
- 2 a) die Objekte werden als Ganzes auf eine bestimmte Höhe verschoben,
b) die Objekte werden an den Verlauf des DGM angepasst.

2.2 Umwandlung verschiedener Referenzsysteme

Werden 2D- und 3D-Geodaten aus verschiedenen Datenquellen verarbeitet, so ist es recht wahrscheinlich, dass sie unterschiedlich georeferenziert sind. Sollen die Geodaten gemeinsam verwendet werden ist eine Umrechnung auf ein einheitliches räumliches Bezugssystem notwendig. Da noch keine Implementation der entsprechenden OGC-Spezifikationen in Java vorlag, wurde auf das Java-Paket von SEDRIS zurückgegriffen (SEDRIS 2001). Da Gauß-Krüger nicht im Paket enthalten ist, wurden entsprechende Klassen für die Umrechnung zwischen Gauß-Krüger Koordinaten und geodätischen Koordinaten hinzugefügt. Im Folgenden muss zwischen der Integration der Objektdaten (Gebäude etc.) und der Integration des Geländemodells unterschieden werden:

2.3 Integration durch automatische Objektverschneidung

Durch den Verschneidungsprozess werden aus den Objekten verschiedener Datenquellen komplexe 3D-Features mit mehreren Ansichten (Views) erzeugt, die die Features in verschiedenen Detaillierungsgraden zeigen. Dabei können zwei Strategien verfolgt werden: die semiautomatische und der automatische Verschneidung. Im VR-Server wurde ein automatischer Verschneidungsalgorithmus realisiert: Der Einfachheit halber arbeitet er nur zweidimensional, betrachtet also nur die Grundrisse der Objekte in der x-y-Ebene. Die Höhe über NN spielt keine Rolle. Zwei Objekte, die verschiedene Detailstufen repräsentieren, werden einander zugeordnet, wenn entweder der Mittelpunkt des ersten Objektes im Grundriss des zweiten Objektes liegt oder der Mittelpunkt des zweiten im Grundriss des ersten Objektes liegt. Durch die dynamische Verschneidung und die Erzeugung komplexer Features wird die Automatisierung der gegenseitigen Zuordnung der unterschiedlichen Objekte in den verschiedenen Grundlagendatensätzen erreicht. Probleme ergeben sich hinsichtlich der Digitalisierung der Ursprungsdaten, da hierbei unterschiedliche Kriterien bei der Objektbegrenzung und Generalisierung angewandt wurden. Ein Objekt der einen Datenquelle ist oft nur schwer einem Objekt einer anderen zuzuordnen. Die folgenden Fälle treten auf:

- ein Objekt der einen Quelle ist bei einer anderen in mehrere Objekte aufgeteilt,
- ein Objekt der einen Quelle fehlt bei einer anderen Quelle,
- ein Objekt der einen Quelle nimmt eine viel größere Fläche ein als das entsprechende Objekt einer anderen Quelle,
- ein Objekt der einen Quelle ist gegenüber dem entsprechenden Objekt einer anderen Quelle verschoben oder verzerrt.

Einige der möglichen Konstellationen soll die folgende Abbildung veranschaulichen, die die Stadthalle und den Marstall in der Heidelberger Altstadt zeigt. Auf der linken Seite ist der

mit Hilfe der Geschosshöhe extrudierter Grundriss aus der digitalen Grundkarte, auf der rechten Seite das 3D Modell aus der ODF-Datei abgebildet.

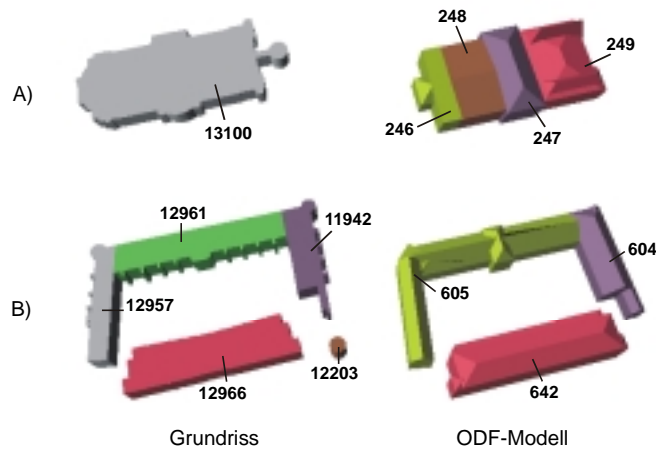


Abb. 2: Aufteilung von Gebäudekomplexen bei verschiedenen Datenquellen. Links digitale Grundkarte (ALK), rechts 3D-Modell aus ODF-Datei. A) Marstall, B) Stadthalle

Im Fall der Stadthalle wird das Gebäude im Grundriss durch ein einziges Objekt mit der ID 13100 repräsentiert, im ODF-Modell jedoch in 4 Teilobjekte mit den IDs 246, 247, 248 und 249 unterteilt. Aus der Verschneidung geht ein Feature hervor, das im View des ODF-Modells ein aus 4 Teilen zusammengesetztes FeatureObject enthält. Im Fall des Marstalls lassen sich die Paare 12966/642 und 11942/604 eindeutig einander zuweisen. Der Westflügel und das Mittelteil sind im Grundriss separiert, im ODF-Modell als ein Objekt digitalisiert. Der Turm an der Ost-Süd-Ecke hat kein Pendant im ODF-Modell. Daraus ergeben sich die beiden weiteren Features 12957+12961/605 und 12203/-.

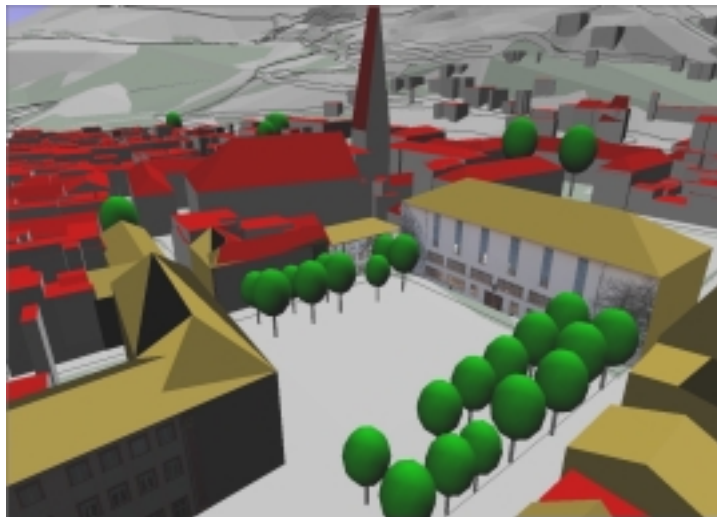


Abb. 3: Generiertes integriertes 3D-Modell. Vogelperspektive Universitätsplatz.

2.4 Integration von Geländemodellen

Bei der Integration des Geländemodells werden ebenfalls mehrere Fokusregionen verwendet. Das DGM besteht aus mehreren Layern, die verschiedene Detaillierungsgrade repräsentieren. Die Layer liegen dabei in den Datenquellen nicht als trianguliertes Dreiecksnetz vor, sondern als Menge von Knotenpunkten. Die Triangulation erfolgt erst, nachdem die geeigneten Punkte ausgewählt wurden. Vom Vermessungsamt der Stadt Heidelberg wurden die genauen Höhen und Positionen aller Kanaldeckel zur Verfügung gestellt. Zusammen mit der Digitalisierung der Höhenlinien aus der topographischen Karte 1 : 50000 konnte damit ein DGM erstellt werden, das sowohl das bebaute als auch unbebaute Gelände abdeckt. Aus dem genauesten DGM wurden weitere mit größerer Maschenweite abgeleitet. Alle Geländedaten sind in der SDE-Datenbank als 2D-Punkte gespeichert. Die zugehörigen z-Werte stehen als Attribut in einer Tabellenspalte zur Verfügung. Aus diesen Daten werden 3D-Punkte generiert. Der VR-Server hat somit Zugriff auf drei unterschiedliche genaue Datensätze mit 3D-Punktinformationen, aus denen dynamisch Geländemodell berechnet werden können.

Die Auswahl der jeweiligen Layer bzw. Datenquellen orientiert sich am Routenverlauf, so dass nur in der Nähe der Route das feinmaschige TIN verwendet wird und mit zunehmender Entfernung die Maschenweite des DGMs zunimmt. Als Triangulationsalgorithmus kommt eine Klasse (DelaunyClarkson) aus dem Paket „VisAD“ (VisAD 2001), einer Klassenbibliothek für die Visualisierung und Analyse numerischer Daten, zum Einsatz, das auch topologische Informationen berechnet. Als Ergebnis liefert die Klasse ein Multiresolution-Geländemodell, das an die möglichen Aufenthaltsorte des Betrachters angepasst ist. Die Abb. 5 zeigt das Resultat, bei dem die Route ungefähr in der Mitte verläuft (zu Fokuskarten vgl. Zipf 2002). Neben dieser Konvertierung wurde auch der Versuch unternommen, die Generierung von 3D-Modellen objektspezifisch zu verfeinern. Insbesondere für Gebäude besteht der Bedarf, durch Angabe weiterer Parameter das Erscheinungsbild der Modelle weiter zu verbessern.

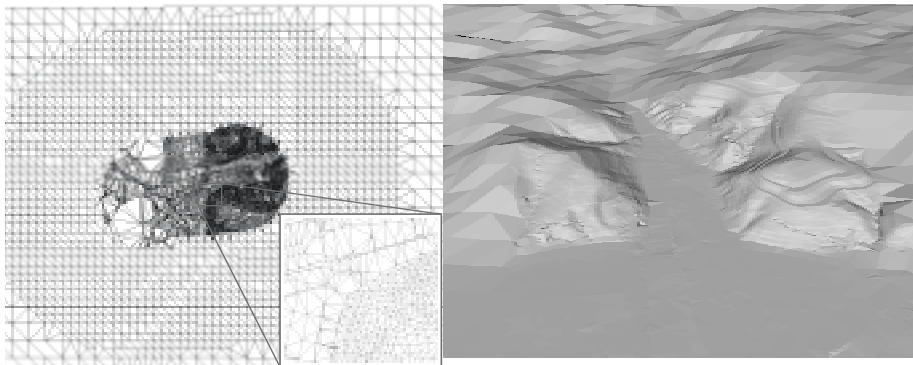


Abb. 4/5: An Tourverlauf dynamisch angepasstes Multiresolution-Geländemodell als TIN-Gitterlinien und in schattierter Darstellung

3 Einbindung von 3D-Touranimationen in Deep Map

Im Folgenden soll kurz die Architektur des Systems vorgestellt werden. Zur Realisierung von 3D-Tourvisualisierungen werden eine Reihe neuer Komponenten und Agenten benötigt. Der VR-Server-Agent nimmt Nachrichtenobjekte mit räumliche Anfragen entgegen und generiert eine geeignete 3D-Szene. Er ist mit dem eigentlichen VR-Server verbunden und leitet die Anfragen entsprechend weiter. Die Ausgabe der 3D-Szene erfolgt in VRML-Dateien. Die Visualisierungskomponente ist für die Visualisierung auf der Client-Seite und die Interaktion mit dem Benutzer zuständig. Die GUI der Visualisierungskomponente besteht aus einem Applet und einem 3D-Browser (VRML-Browser als Plug-In). Das Applet wird auf den Rechner des Client übertragen und präsentiert Felder für Benutzereingaben und Auswahlmöglichkeiten. Es kommuniziert über das HTTP-Protokoll mit dem Servlet. Die Daten werden an das Applet übertragen und über das VRML External Authoring Interface (EAI) in die Szene eingebunden. Die 3D-Darstellung und Navigation wird von dem VRML-Browser übernommen, dem die Szene vom Applet übergeben wird. Um eine schrittweise Übertragung der erzeugten VRML-Daten zu ermöglichen, wird eine Obergrenze für die in einer Datei enthaltenen Features festgesetzt. Besteht eine Szene aus mehr Features als diese Obergrenze, wird sie beim Export in mehrere Dateien aufgeteilt.

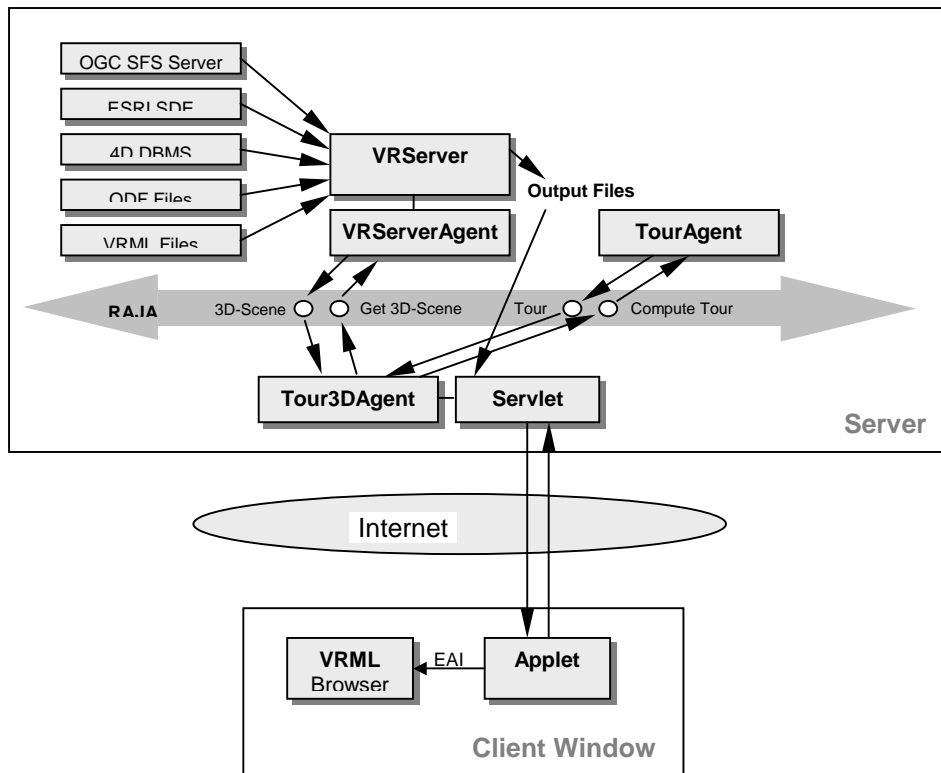


Abb. 6: Gesamtarchitektur mit 3D-Touren-Agent und VR-Server

3 Dynamische Generierung von an Touren angepasste 3D-Modellen

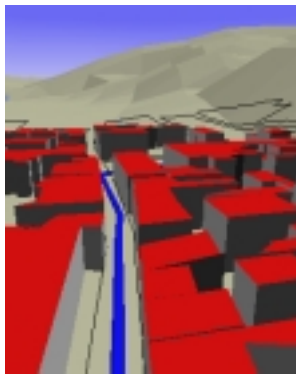
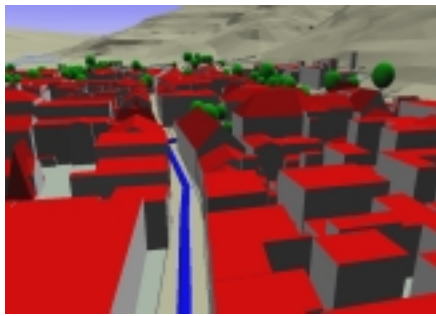
Räumliche Abfragen bestehen im Wesentlichen aus einer Abfrage-Geometrie und einer Liste der Informationsebenen. Die Geometrie beschreibt die möglichen Aufenthaltsorte des Benutzers und kann punkt-, linien- oder flächenförmig sein. Da der VR-Server hauptsächlich für die Routenvisualisierung eingesetzt werden soll, ist die Geometrie bisher mit dem Verlauf der Route (Liniensegmente) angegeben wird, identisch. Die räumliche Abfrage muss so übersetzt werden, dass in der resultierenden Szene die Objekte mindestens bis zu einem bestimmten Abstand sind. Für jede Ebene wird also die Auswahlregion durch Berechnung des Puffers, der die Route im Abstand von d umschließt, ermittelt.



Abb. 7: Puffer für unterschiedliche Detailstufen um eine berechnete Tour

Vom einem 3D-Touren-Modul werden tourenspezifische Simulationen erstellt. Sie sollen dem Benutzer einen mehr oder weniger realitätsnahen Einblick in den Verlauf der von ihm gewählten Tour vermitteln. Die Simulationen werden durch die Auswahl von ViewPoints gestartet. Die Abbildungen 8 und 9 zeigen die Ergebnisse einer Tourenvisualisierung bei unterschiedlichen Konfigurationen. Der Vergleich soll die Möglichkeit der Anpassung an unterschiedliche Übertragungsraten und Rechnerleistungen aufzeigen. Verändert wurden die Auswahl der Themen, die verwendeten Ebenen, die Größen der Fokusregionen und der Generalisierungsgrad. Das Konzept von Fokusregionen kann auch bei 2D-Karten zur Verbesserung der Lesbarkeit eingesetzt werden (Zipf und Richter 2002).

Die Simulationen werden durch Manipulation des Viewpoints realisiert. Mit Hilfe von LocationSensoren und OrientationSensoren kann für jeden Tourenabschnitt Position und Blickrichtung der virtuellen Kamera definiert werden. Über VRML-Routing werden diese Werte mit dem Transform-Knoten, der den ViewPoint enthält, verknüpft.



Thema	Layer	m _{max} [m]	Dreiecke / Features
DGM	TIN	5000	24045
	RSG 250 m	12000	
	RSG 500 m	50000	
Gebäude	ODF	500	803
	SDE (flatness 1.0)	600	
Baublöcke		1000	141
Straßenseiten		1000	156
Bäume		1000	162
Gewässer		10000	31
Gesamtgröße			2,28 MB

Thema	Layer	m _{max} [m]	Dreiecke / Features
DGM	RSG 125 m	3000	3069
	RSG 250 m	8000	
	RSG 500 m	50000	
Gebäude	SDE (flatness 3.0)	300	451
Straßenseiten		500	64
Gewässer		5000	11
Gesamtgröße			551 KB

Abb. 8/9: Modelle mit unterschiedlichen Generierungsparametern

4 Ausblick

War in früheren Ansätzen nur die Verwendung relativ statischer VRML-Dateien zur Routenvisualisierung möglich (vgl. ZIPF und MALAKA 1999), können nun diese neuen Ansätze der automatisierten Überführung von 2D und 3D Geodaten vorgestellt werden. Im Beispiel wurden in Java Komponenten entwickelt, die ein vollständiges Stadt- und Landschaftsmodell (Gelände- und Stadtmodell) aus diversen Datenquellen gemäß gegebenen Restriktionen wie resultierender Dateigröße angepaßt an einen errechneten Tourenverlauf erzeugen. Es wurde gezeigt, wie derartige Funktionen in die bestehende Agenten-Infrastruktur integriert werden kann. Das System verfügt über eine prototypische Web-Oberfläche. Da die Gebäude aus GIS-Daten generiert wurden sind die entsprechenden Schlüssel auch in der generierten VRML-Welt verfügbar. Somit konnte die Deep Map Datenbank mit Informationen zu historischen Sehenswürdigkeiten in das System integriert werden, so daß per Mausklick auf ein VRML-Gebäude in einem Fenster die entsprechenden Informationen dargestellt werden. Zukünftige Arbeiten müssen noch die erzielbare Performanz evaluieren und verbessern. Die automatisierte Erzeugung (noch) ansprechender Computeranimationen ist selbstverständlich eine längerfristig bestehende Aufgabenstellung.

5 Literatur

- Brenner, C. & N. Haala (2000): *Erfassung von 3D Stadtmodellen*. In: PFG - Photogrammetrie, Fernerkundung, Geoinformation, Heft 2/2000, S. 109-118.
- Coors, V. (1997): *Konzeption und Entwicklung einer Client/Server-Architektur für den Zugriff und die Visualisierung von geographischen 3D-Informationen via World Wide Web*. Diplomarbeit Technische Hochschule Darmstadt Fachbereich Informatik.
- Coors, V. (2001): *Feature-preserving Simplification in Web-based 3D-GIS*. In: Proceedings of International Symposium on Smart Graphics, New York, March 2001
- Evans, S. & A. Hudson-Smith (2001): *Information Rich 3D Computer Modeling of Urban Environments*. Centre for Advanced Spatial Analysis Working Paper Series, 35.
- Flick, S. (1998): *Konzeption eines adaptiven Frameworks für 3D-Geo-Informationssysteme*. PhD thesis. Fraunhofer IGD. Darmstadt.
- Lange, E. (1999): *Von der analogen zur GIS-gestützten 3D-Visualisierung bei der Planung von Landschaften*. In: Geo-Informationssysteme, 2, S. 29-37.
- Malaka, R. & A. Zipf (2000): *DEEP MAP - Challenging IT research in the framework of a tourist information system*. In: Fesenmaier, D.; S. Klein & D. Buhalis (eds.): *Information and Communication Technologies in Tourism 2000*. Proceedings ENTER.
- Molenaar, M. (1990): *A Formal Data Structure for Three Dimensional Vector Maps*. In: Proceedings of the 4th International Symposium on Spatial Data Handling. 830-843.
- Schilling, A. (2002): *Integration und Visualisierung von 2D- und 3D-Geodaten in einem verteilten GIS am Beispiel virtueller Stadttouren*. Diplomarbeit. Geographisches Institut Heidelberg.
- Source for Environmental Representation and Interchange (SEDRIS) (2001): www.sedris.org.
- Visualization for Algorithm Development (VisAD) (2001): <http://www.ssec.wisc.edu/~billh/visad.html>.
- Winkler (1999): *Verkehrsbedingte Luftverunreinigungen und Lärmbelastungen in Heidelberg*. Dissertation. Geographisches Institut. Universität Heidelberg. Ibidem..
- Zipf, A. & H. Aras (2001): *Realisierung verteilter Geodatenserver mit der OpenGIS SFS für CORBA*. In: GIS Geo-Informationssysteme, 03/2001. Heidelberg. pp 36-41.
- Zipf, A. & R. Malaka (1999): *Web-basierte Planung und animierte Visualisierung von 3D Besichtigungstouren im Rahmen des Touristeninformationssystems Deep Map*. In: Zagel, B. (Hrsg.): *GIS in Verkehr und Transport*. Huethig Verlag. Heidelberg.
- Zipf, A. and Krüger, S. (2001): *TGML - Extending GML by Temporal Constructs - A Proposal for a Spatiotemporal Framework in XML*. ACM-GIS 2001. The 9th ACM International Symposium on Advances in Geographic Information Systems. Atlanta.
- Zipf, A. & S. Krüger (2001): *Ein objektorientierter Framework für temporale 3D-Geodaten*. AGIT 2001, Symposium für Angewandte Geographische Informationsverarbeitung, 04.-06. Juli 2001, Salzburg. Austria.
- Zipf, A. (2002): *User-Adaptive Maps for Location-Based Services (LBS) for Tourism*. Proceedings of ENTER 2002. International Congress on Tourism and Communications Technologies in Tourism Innsbruck Austria. Springer Computer Science. Heidelberg.
- Zipf, A. und Richter, K.-F. (2002 submitted): *Using FocusMaps to Ease Map Reading. Developing Smart Applications for Mobile Devices*. KI - Künstliche Intelligenz. Sonderheft Spatial Cognition.