

## Dynamische Generierung von VR-Stadtmodellen aus 2D- und 3D-Geodaten für Tourenanimationen

### Abstract

The automated integration of 2D and 3D Geodata is important for the dynamic generation of 3D-city models. We present several aspects of this topic and introduce a prototype for the automated generation of integrated sets of 3D-Geodata. It is necessary to distinguish between methods for integrating building-versus terrain models. Based on the integrated city model we introduce components that dynamically build 3D tour animations for the internet (VRML) that are optimized for the particular tour. The user gets an interactive 3D visualization of a planned tour through the hybrid 3D model of Heidelberg.

### Zusammenfassung

Für die dynamische Erzeugung von 3D-Stadtmodellen spielt die Frage der automatisierten Integration von 2D und 3D-Daten eine bedeutende Rolle. In diesem Beitrag werden Mechanismen sowie der realisierte Prototyp zur automatischen Generierung integrierter 3D-Datensätze vorgestellt. Hierbei muß zwischen unterschiedlichen Ansätzen bei Gebäude- und Geländemodellen unterschieden werden. Basierend auf den integrierten Daten wird eine Strategie zur Erzeugung von Internet-tauglichen 3D-Animationen (VRML) von Besichtigungstouren vorgestellt. Dem Benutzer wird dabei eine dreidimensionale und interaktive Visualisierung einer berechneten Tour vermittelt, indem der Benutzer durch das hybride 3D-Modell von Heidelberg geführt wird.

## 1 Einleitung und Problemstellung

Eines der großen Hindernisse für die Erstellung von 3D-Stadtmodellen für unterschiedliche Virtual Reality (VR) – Anwendungen (vgl. z.B. Kraak *et al.* 1999) stellt die oftmals unzureichende Datengrundlage dar. Die Erstellung detailgetreuer Modelle ist aufwendig und erfolgt in vielen Fällen manuell. Verfahren für eine automatische Erfassung ganzer Stadtteile durch Laserscans oder photogrammetrische Methoden befinden sich in der Entwicklung (BRENNER und HAALA 2000,2001; EVANS und HUDSON-SMITH 2001). Gerade für großflächige Modelle ist es sinnvoll auf bestehende 2D-GeoDaten zurückzugreifen um diese auch zur 3D-Visualisierung zu nutzen. Heute stehen heterogene und unterschiedlich erhobene und referenzierte Geodaten digital zur Verfügung. Damit ist es wünschenswert neue 3D-Modelle mit den existenten 2D-Vermessungsdaten zu integrieren. Also lautet eine erste Problemstellung, die unterschiedlichen Geodaten so zu verwalten und zu integrieren, dass ein gemeinsamer Zugriff für die eine 3D-Visualisierung möglich wird. Hierzu müssen 2D-Geodaten in die dritte Dimension überführt werden. Die so generierten 3D-Modelle werden mit weiteren, originär in 3D erstellten, Modellen über einen Algorithmus fusioniert. Dadurch werden insbesondere auch 3D Ansichten von Gebieten realisierbar, für die bisher nur 2D-Daten flächendeckend vorliegen. Wünschenswert ist dabei ein möglichst hoher Automatisierungsgrad. Um (Teilansichten von) Stadtmodellen dynamisch generieren zu können, muß auf die verschiedenen Datenquellen transparent zugegriffen werden können. Die Geodateninfrastruktur sollte dabei möglichst unverändert bleiben, da auf diese meist schon Fachanwendungen zugreifen. Ein Lösungsansatz ist die Verwendung eines Virtual Reality Servers (VR-Servers), der über Schnittstellen (Loader) zu unterschiedlichen 2D- und 3D- Datenquellen verfügt. Räumliche Anfragen werden für die jeweiligen Datenquellen umgeformt und an diese weitergeleitet. Das 3D-Datenmodell orientiert sich dabei auf dem von FLICK (1996,1998) veränderten Modell nach Molenaar (1990). Die Architektur ist an einen von Coors (1997, 2001) entwickelten 3D Server angelehnt. Coors (2002) stellt eigene Weiterentwicklungen des Geometriemodells vor. Eine zweite Fragestellung betrifft die Nutzung des VR-Servers für ein konkretes Anwendungsgebiet – in unserem Fall die 3D-Tourenvisualisierungen über das Internet. Die entwickelten Komponenten führen den Benutzer entlang einer von einer Tourenplanungskomponente berechneten Tour durch ein interaktives virtuelles Modell der Stadt Heidelberg.

## 2 Integration von Geodaten

In Heidelberg wurde der zentrale Altstadtbereich mittels einer Laserscan-Befliegung dreidimensional erfaßt. Hieraus wurde vom Institut für Photogrammetrie (IfP) der Universität Stuttgart ein 3D Modell der Dachlandschaft erstellt. Dieses liegt als ODF- bzw. teilweise texturierte VRML-Datei vor. Dabei kam ein vom IfP entwickeltes Verfahren zum Einsatz, das eine weitgehend automatische Generierung von Stadtmodellen ermöglicht (Brenner, Haala & Fritsch 2001, Brenner & Haala 2001,

Ameri, B. & Fritsch, D. (2000). Die zweidimensionalen Gebäudegrundrisse (ALK, Amtliches Liegenschaftskataster), sowie Strassen und Baublöcke etc. sind für das gesamte Stadtgebiet verfügbar und wurden vom Vermessungsamt der Stadt Heidelberg freundlicherweise zur Verfügung gestellt. Um auch diese für die 3D Visualisierung zu nutzen, werden zu den zweidimensionalen Punkten, Linien und Polygonen Höhenwerte hinzugefügt. Das digitale Geländemodell wurde aus Daten des Geographischen Institutes Heidelberg und Daten des Vermessungsamtes fusioniert, so daß das bebaute als auch unbebaute Gelände abdeckt ist. Für die Gebäude wurden in einer Arbeit des Geographischen Institutes Stockwerkszahlen erhoben (Winkler 1999). Die Höhe der Gebäudeunterkante wird aus dem DGM gewonnen. Mittels dieser Informationen kann aus einem Polygon per Extrusion ein Blockmodell erzeugt werden. Zusätzlich wurde ein von der FhG für Graphische Datenverarbeitung (Fhg IGD), Darmstadt für das EML erstellte VRML-Modell des Heidelberger Schlosses integriert.

## 2.1 Erzeugung von 3D-Geobjekten aus 2D-Geodaten

Ein typisches Beispiel für die Erstellung von 3D-Geodaten ist die Generierung von einfachen Blockmodellen aus Gebäudegrundrissen. Liegen die Daten aufbereitet vor ist dies über Werkzeuge wie den ESRI ArcView 3D Analyst schnell per Hand zu realisieren. LANGE (1999) beschreibt 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 mittels einer Programmierschnittstelle (API). Daher werden die 2D-ALK-Daten in einem Geodatenserver auf Basis von ESRI SDE verwaltet und auf diese über eine in Java realisierte OpenGIS konforme CORBA-Schnittstelle zugegriffen (Zipf & Aras 2001). Der VR-Server errechnet durch Extrusion gemäß der Gebäudehöhen und Anpassung auf das Geländemodell ein 3D-Gebäudemodell. Trotz der Einfachheit des Blockmodells liefert es bei kleinen Maßstäben schon gute Ergebnisse.

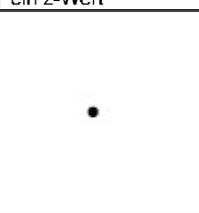
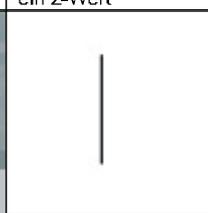
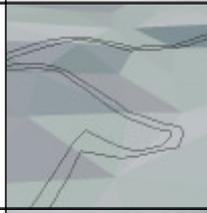
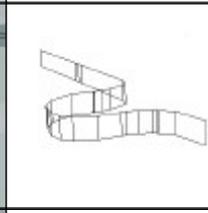
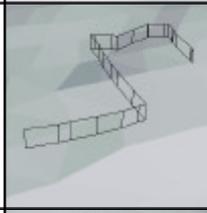
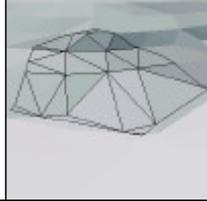
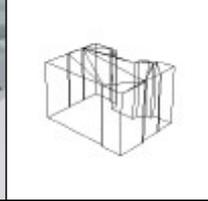
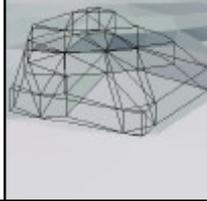
	ohne vertikale Ausdehnung		mit vertikaler Ausdehnung	
	ein z-Wert	Anpassung an DGM	ein z-Wert	Anpassung an DGM
Punkt				
Linie				
Fläche				

Fig 1: Möglichkeiten der Konvertierung von 2D nach 3D Geometrien.

Weiter sind auch Geodaten vorhanden, die keine vertikale Ausdehnung besitzen oder benötigen (z.B. Landnutzung (Wald, Felder), Verkehrslinien usw.). Diese müssen wie folgt behandelt werden, damit sie parallel zum Oberflächenverlauf des DGMs verlaufen:

*Linienzüge* müssen an den Oberflächenverlauf des DGM angepasst werden. Es reicht nicht jeden Punkt um einen z-Wert zu erweitern. Stattdessen muss bei jeder Kreuzung eines Liniensegments mit einer Kante des DGM ein neuer Punkt berechnet und eingefügt werden.

*Flächen* müssen in Facetten aufgeteilt werden, die parallel zu den Dreiecken des Geländes angeordnet sind. Alternativ werden sie in das DGM „hineingeschnitten“, indem die Triangulation und

die Topologie so angepasst werden, so dass zwischen Dreiecken innerhalb und außerhalb der Fläche unterschieden werden kann. Sie werden also zu Bestandteilen des DGM und durch farbliche Markierung sichtbar. Es lassen sich bei der Konvertierung also die folgenden Möglichkeiten unterscheiden (Fig. 1):

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.

Der VR-Server kann sowohl Linienzüge als auch Flächen genau dem Gelände anpassen. Somit können sämtliche konventionellen Karteninhalte für die Visualisierung herangezogen werden. Im Folgenden wird die unterschiedliche Integration von Objektdaten (Gebäude etc.) und des Geländemodells dargestellt:

### 2.3 Automatische Objektintegration

Durch einen Verschneidungsprozess werden aus den Objekten verschiedener Datenquellen komplexe 3D-Features mit mehreren Ansichten (Views) erzeugt, die die Features in verschiedenen Detaillierungsgraden zeigen. Bei näherer Betrachtung der Ursprungsdaten wird offensichtlich, dass sich die Zuordnung zwischen den Objekten oft nicht durch einfache 1:1 Beziehungen beschreiben lässt. Die Probleme ergeben sich aus der ursprünglichen Erhebung der Daten, da hierbei natürlicherweise unterschiedliche Kriterien für Objektbegrenzung und Generalisierung angewandt werden. Somit ist ein Objekt aus Datenquelle A oft nur schwer einem Objekt einer anderen zuzuordnen. Folgende Fälle können auftreten:

- 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

Da die Datenquellen möglichst unverändert angesprochen werden sollte, wurde ein automatischer Verschneidungsalgorithmus realisiert, der die Zuordnung der jeweiligen Objekte automatisch anhand der räumlichen Beziehungen der Objekte bewerkstelligt: Der Einfachheit halber betrachtet er nur die 2D Grundrisse der Objekte in der x-y-Ebene. Zwei Objekte, die verschiedene Detailstufen repräsentieren, werden einander zugeordnet, wenn entweder der Mittelpunkt des ersten Objektes im Grundriss des zweiten Objektes oder der Mittelpunkt des zweiten im Grundriss des ersten Objektes liegt. Hierdurch und die Erzeugung komplexer Features wird die Automatisierung der gegenseitigen Zuordnung der unterschiedlichen Objekte in den verschiedenen Grundlagendatensätzen erreicht. Durch Verschneidung erhalten die 3D Features mehrere unterschiedlich detaillierte Versionen, die bei der Darstellung entfernungsabhängig ein- und ausgeblendet (Level of Detail, LoD) werden können. Fig. 2 veranschaulicht einige mögliche Konstellationen. Links ist der extrudierte Grundriss aus der digitalen Grundkarte, rechts das 3D-Modell (ODF) abgebildet.

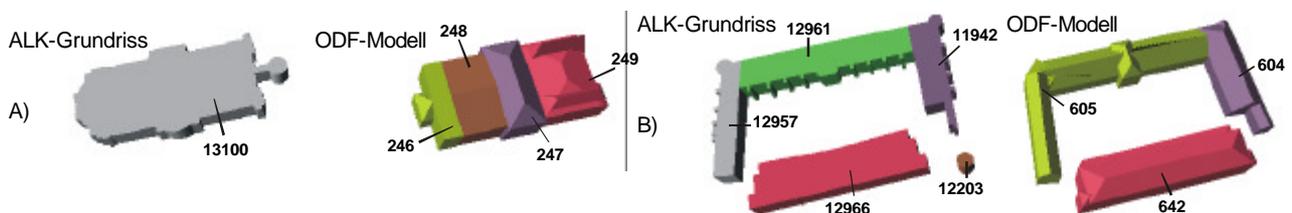


Fig 2: Aufteilung von Gebäudekomplexen bei verschiedenen Datenquellen. A) Stadthalle, B) Marstall. Links digitale Grundkarte (ALK), rechts 3D-Modell aus ODF-Datei.

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 FeatureObjekt enthält. Im Fall des Marstalls lassen sich die Paare 12966/642

und 11942/604 eindeutig zuweisen. Der Westflügel und das Mittelteil sind im Grundriss getrennt, im ODF-Modell als ein Objekt digitalisiert. Der südöstliche Turm hat kein Pendant im ODF-Modell. Damit ergeben sich die beiden weiteren Features 12957+12961/605 und 12203/-.

Zur Übertragung an den Client werden die Ergebnisse aus dem VR-Server nach VRML exportiert (Fig. 3). Dabei werden die unterschiedlichen „Views“ eines Feature in denen sich die unterschiedlichen Repräsentationen eines Geobjekts aus den unterschiedlichen Datenquellen befinden als „Blätter“ in verschiedene Level of Detail (LoD) - Stufen der VRML-Datei geschrieben (vgl. Coors & Flick 1998). Entfernungsabhängig schalten VRML-Browser zwischen diesen um. Die Entfernungs-Schwellenwerte werden aus View-spezifischen Maßstabsbereichen berechnet.

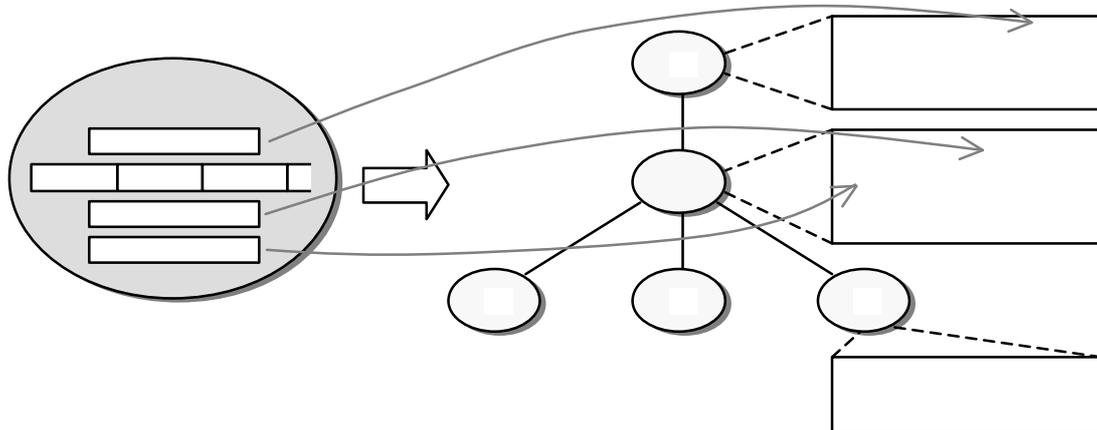


Fig 3: Export der Features nach VRML (mit LoD).

#### 2.4 Dynamisch generierte Multiresolution-Geländemodelle

Bei der Integration des Geländemodells kam ein Continuous-LoD Konzept zur Anwendung, das variable Maschenweiten in Abhängigkeit von der Entfernung vom Betrachter erlaubt. Das DGM besteht aus mehreren Ebenen (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. Aus dem genauesten DGM wurden weitere mit größerer Maschenweite abgeleitet. Die Geländedaten sind in SDE als 2D-Punkte plus Z-Wert gespeichert. 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 Datenquellen orientiert sich am Routenverlauf. Durch Berechnung von Fokusregionen wird die räumliche Auswahl vorgenommen. Somit wird nur in der Nähe der Route ein feinmaschiges TIN verwendet und in größerer Entfernung zunehmend gröbere RSGs (Regular Square Grids). Als Triangulationsalgorithmus kommt eine Klasse (DelaunyClarkson) aus dem Paket „VisAD“ (VisAD 2001) zum Einsatz, das auch topologische Informationen berechnet. Das Ergebnis ist ein an die möglichen Aufenthaltsorte des Betrachters angepaßtes Multiresolution-Geländemodell. Fig. 4 zeigt ein Ergebnis, bei dem die Route ungefähr in der Mitte verläuft.

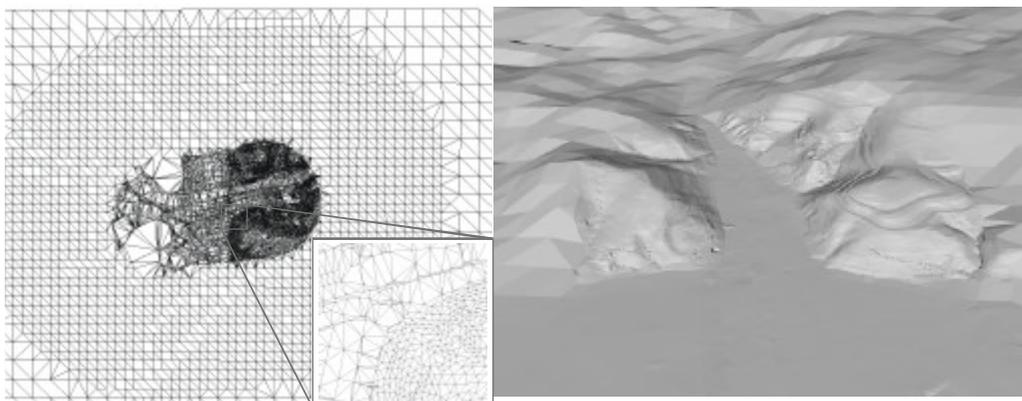


Fig. 4: An Tourverlauf angepaßtes Multiresolution-Geländemodell (Darstellung als TIN und schattiert).

Bei der Generierung des Geländemodells können zudem an den Stellen an denen im Resultat Gebäude stehen sollen, für diese Plattformen in das DGM geschnitten werden (Fig. 5)

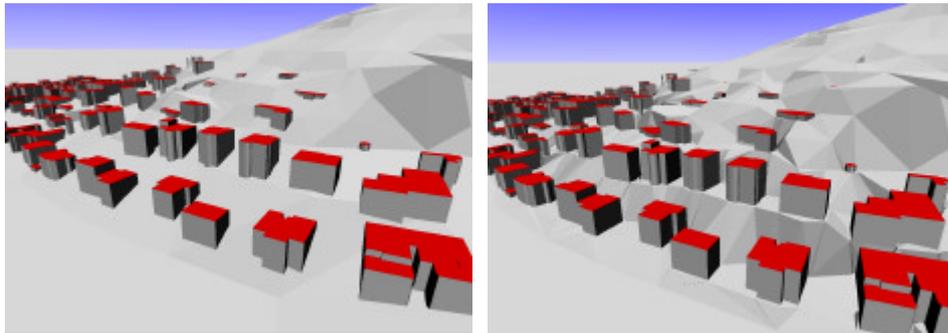


Fig 5: Einbettung von Gebäuden in das ursprüngliche und in ein um Plattformen erweitertes DGM

### 3 Dynamische Erzeugung von an Tourverläufen angepasste 3D-Modelle

Räumliche Abfragen bestehen aus einer Abfrage-Geometrie und einer Liste der Layer. Die Geometrie beschreibt die möglichen Aufenthaltsorte des Benutzers und kann punkt-, linien- oder flächenförmig sein. Da der VR-Server für die Routenvisualisierung eingesetzt werden soll, entspricht die Geometrie bisher dem Verlauf der Route (Liniensegmente). Die räumliche Abfrage muß so übersetzt werden, dass in der resultierenden Szene die Objekte mindestens bis zu einem bestimmten Abstand sind. Für jede Feature-Ebene (z.B. Buildings) wird also die Auswahlregion durch Berechnung des Puffers, der die Route im Abstand in einem bestimmten Abstand umschließt, ermittelt (Fig. 6).

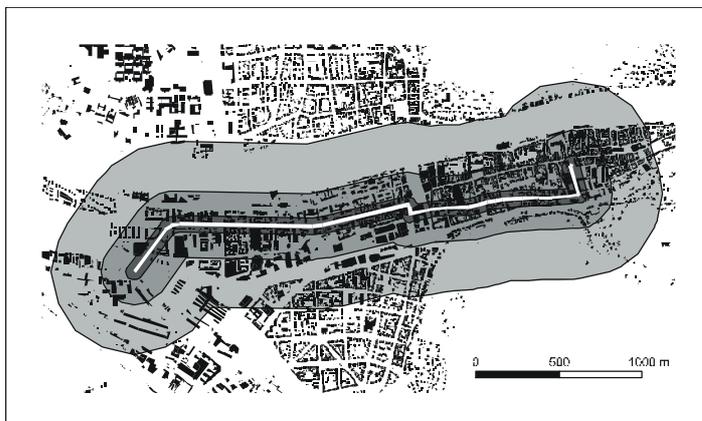


Fig. 6: Puffer für unterschiedliche Detailstufen um Tourenverlauf

Das 3D-Touren-Modul erstellt tourenspezifische Animationen mit der Tour im Fokus. Das Konzept von Fokusregionen kann auch bei 2D-Karten zur Verbesserung der Lesbarkeit eingesetzt werden (Zipf und Richter 2002). Fig. 7 zeigt dann ein aus den diversen Datenquellen dynamisch für den blau eingezeichneten Tourenverlauf generiertes 3D Stadtmodell. Die Bäume liegen als Punktinformation im GIS vor. Ihre Darstellung besitzt ebenfalls Detailstufen.

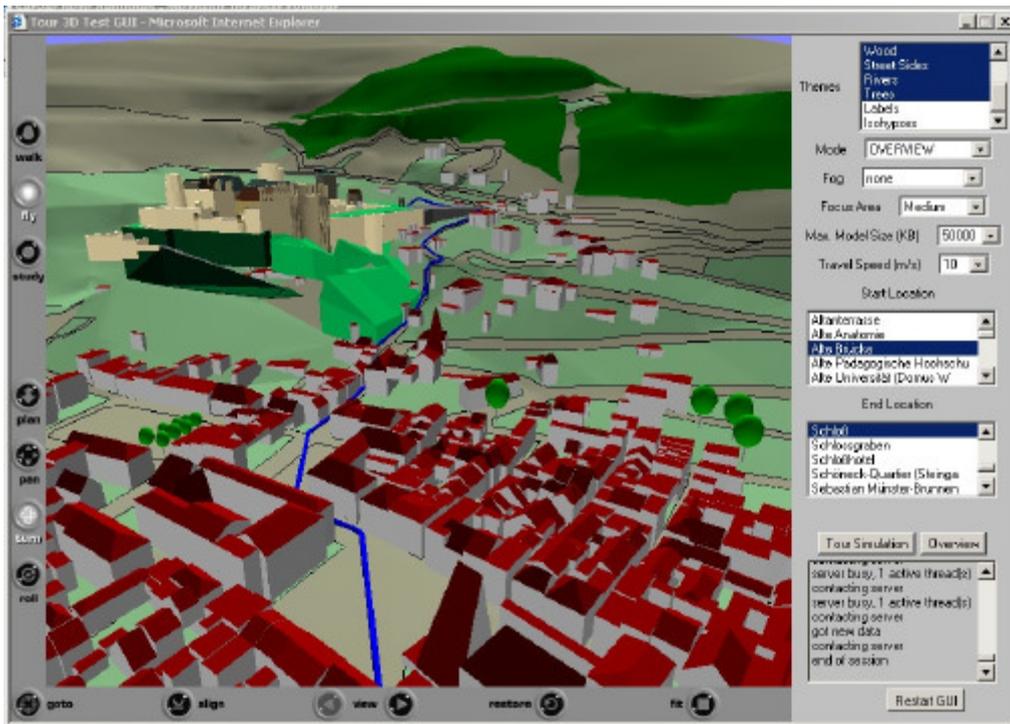
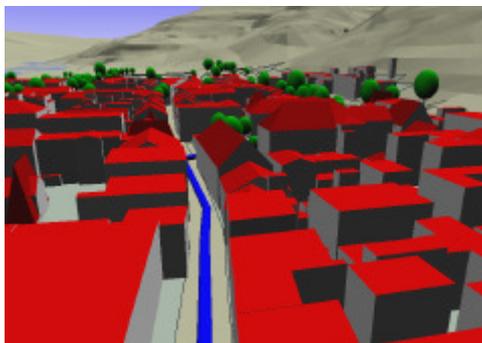
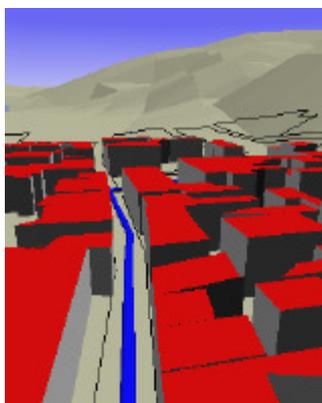


Fig. 7: Aus mehreren Datenquellen automatisch generiertes 3D-Modell von Heidelberg

Fig. 8 zeigt Ergebnisse einer Tourenvisualisierung mit verschiedenen Parametern. Dies zeigt die Möglichkeit der Anpassung an unterschiedliche Übertragungsraten oder Rechnerleistungen. Variiert wurde Auswahl der Themen, die verwendeten Ebenen, die Größen der Fokusregionen (u.a. für Gebäude und DGM) und der Generalisierungsgrad (dynamische Geometrievereinfachung). Die Animationen werden durch Anpassung des „ViewPoints“ im VRML-Browser realisiert.



Thema	Layer	m <sub>max</sub>	Dreiecke / Feature
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 <sub>max</sub>	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

Fig. 8: Modelle mit unterschiedlichen Generierungsparametern (Vergleich)

### 3 Kommunikation und Systemarchitektur

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 Nutzungsoberfläche (GUI) der Visualisierungskomponente besteht aus einem Applet und einem 3D-Browser (VRML-Browser als Plug-In). Das Applet realisiert Auswahlmöglichkeiten für Benutzereingaben. Es kommuniziert über das HTTP-Protokoll mit einem Servlet. Die resultierenden VRML-Daten werden übertragen und über das VRML External Authoring Interface (EAI) in die Szene eingebunden (Fig 9). Die 3D-Darstellung und Navigation übernimmt der VRML-Browser. Um eine schrittweise Übertragung der erzeugten VRML-Daten zu realisieren, wird eine Obergrenze für die in einer Datei enthaltenen Features definiert.

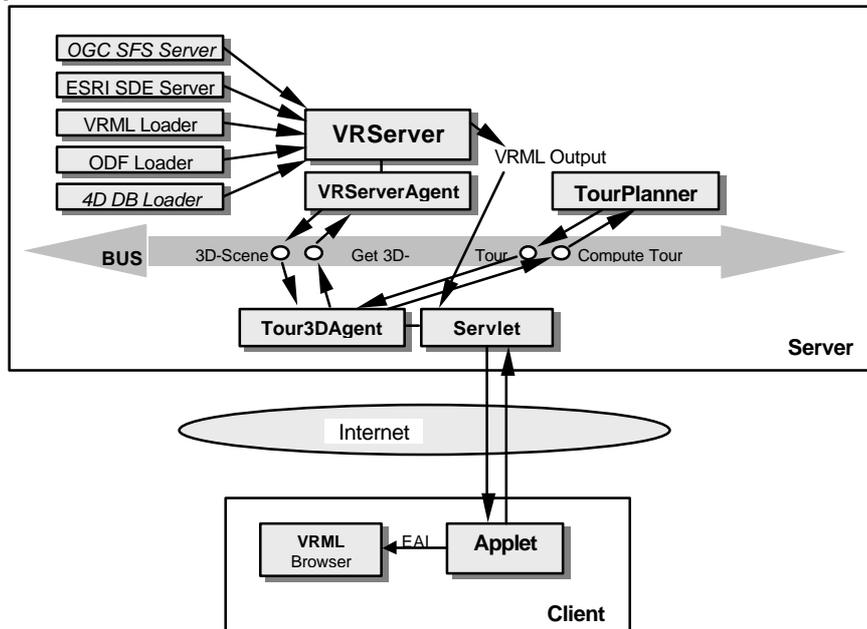


Fig. 9: Gesamtarchitektur mit 3D-Touren-Modul und VR-Server (die kursiv gedruckten Loader sind noch nicht fertiggestellt)

#### 4 Ausblick

War in früheren Ansätzen nur die Verwendung relativ statischer VRML-Dateien zur Routenvisualisierung möglich (z.B. 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. 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 (Weinmann *et al.* 1999), so daß per Mausklick auf ein VRML-Gebäude in einem Fenster die entsprechenden Informationen dargestellt werden. Um solche Dienste auch für portable Systeme wie z.B. Location Based Services (Zipf 2002) verfügbar zu machen, muß die Datenmenge drastisch reduziert werden. So muss evaluiert werden, welche Informationen in welcher Genauigkeit dem Nutzer präsentiert werden können. Z.B. arbeiten Coors (2002) oder Krüger (1998, 2000) an Methoden zur 3D-Visualisierung auf mobilen Geräten. In Zukunft werden vermutlich immer mehr Stadtgebiete mittels neuer Methoden dreidimensional erfasst werden. Dennoch wird auch in Zukunft ein Großteil der Geodaten in zweidimensionalen Geodatenbanken gespeichert sein, so dass deren Integration weiterhin sinnvoll erscheint. Das Konzept der Generierung virtueller Tourenanimationen wird besonders attraktiv, wenn man diese Techniken mit historischen GIS-Daten verknüpft werden, um Touristen zukünftig „virtuelle Zeitreisen“ anbieten zu können, einem der Fernziele des Projektes „Deep Map“ (Malaka und Zipf 2000). Hierzu existieren mit einem temporalen 3D-Geodatenmodell (ZIPF und KRÜGER 2001a, b) weitere Vorarbeiten, so daß eine Realisierung in greifbare Nähe rückt.

## 5 Acknowledgements

Diese Arbeit entstand im Rahmen des von der Klaus-Tschira Stiftung (KTS) geförderten Projektes Deep Map am EML, sowie des BMBF-Projektes SMARTKOM (01 IL 905) und des EU-Projektes CRUMPET (IST-1999-20147). Besonderen Dank gilt dem Vermessungsamt der Stadt Heidelberg, dem Geographischen Institut der Universität Heidelberg, dem Institut für Photogrammetrie der Universität Stuttgart und dem Fraunhofer Institut für Graphische Datenverarbeitung, Darmstadt.

## 6 Literatur

- Ameri, B. and Fritsch, D. (2000): Automatic 3D building reconstruction using plane roof structures. ASPRS Congress 2000, Washington, DC.
- Brenner, C, Haala, N & Fritsch, D. (2001): Towards fully automated 3D city model generation. In Automatic Extraction of Man-Made Objects from Aerial and Space Images III. 2001.
- Brenner, C. & Haala, N. (2001): Automated Reconstruction of 3D City Models. In Abdelguerfi, M. (ed), 3D Synthetic Environment Reconstruction. Kluwer Academic Publishers, pp. 75-101.
- 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. & Flick, S. (1998): Integrating Levels of Detail in a Web-based 3D-GIS, 6th ACM Symposium on Geographic Information Systems (ACM GIS 98), Washington D.C., USA, 1998
- Coors, V. (2002): Resource-adaptive interactive 3D-maps. 2nd International Symposium on Smart Graphics, New York, USA, June 2002.
- Coors, V. (2002 in print): 3D-GIS In networking environments. CEUS Journal. Special issue "3D Cadastre".
- 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. (1996): An object-oriented framework for the realisation of 3D Geographic Information Systems, Proceedings of 2th joint European conference and exhibition on Geographical Information, Barcelona, Spain, pp 187-196.
- Flick, S. (1998): Konzeption eines adaptiven Frameworks für 3D-Geo-Informationssysteme. Dissertation. Fraunhofer IGD. Darmstadt.
- Kraak, M. J.; Smets, G. & Sidjanin, P. 1999: Virtual reality, the new 3d interface for geographical information systems. In: Camara, A. and Raper, J. (eds.): Spatial Multimedia and Virtual Reality. Research Monograph, London, Taylor & Francis, 130-136.
- Krüger, A. (2000): Automatische Abstraktion in 3D-Graphiken. Dissertationen zur Künstlichen Intelligenz, Bd. 232, Berlin: Akad. Verlags-Ges. AKA, 2000.
- Krüger, A. (1998): Automatic graphical abstraction in intent-based 3D-illustrations. In: Proceedings of AVI'98 in l'Aquila, Italy, 1998.
- Lange, E. (1999): Von der analogen zur GIS-gestützten 3D-Visualisierung bei der Planung von Landschaften. In: Geo-Informationen-Systeme (GIS), Zeitschrift für raumbezogene Information und Entscheidungen, 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.
- VisAD (Visualization for Algorithm Development)(2001):<http://www.ssec.wisc.edu/~billh/visad.html>.
- Weinmann, R., Häußler, J. Zipf, A und Malaka, R. (2000): Die Besucher Heidelbergs informieren - die multimediale historische Deep Map Datenbank. In: HGG-Journal. 2000/1 Heidelberg.
- Winkler, R. (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: Geo-Informationen-Systeme (GIS), Zeitschrift für raumbezogene Information und Entscheidungen. 03/2001. Heidelberg. pp 36-41.

- Zipf, A. and Krüger, S. (2001a): 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 (2001b): Zipf, A. und Krüger, S. (2001): Flexible Verwaltung temporaler 3D-Geodaten. In: GIS - Geo-Informationen-Systeme. Zeitschrift für raumbezogene Information und Entscheidungen. 12/2001. pp 20-27.
- Zipf, A. & Malaka, R. (1999): Web-basierte Planung und animierte Visualisierung von 3D Besichtigungstouren im Rahmen des Touristeninformationssystems Deep Map. In: Zigel, B. (Hrsg.): GIS in Verkehr und Transport. Wichmann Verlag. Heidelberg.
- Zipf, A. und Richter, K.-F. (2002): Using FocusMaps to Ease Map Reading. Developing Smart Applications for Mobile Devices. KI - Künstliche Intelligenz. Sonderheft Spatial Cognition.
- 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. (2001): Interoperable GIS-Infrastruktur für Location-Based Services (LBS) - M-Commerce und GIS im Spannungsfeld zwischen Standardisierung und Forschung. In: Geo-Informationen-Systeme (GIS). Zeitschrift für raumbezogene Information und Entscheidungen. 09/2001. 37-43.