

Diplomanden: Wolfgang Sach, Sven Hollenbach

1. Prüfer: Prof. Dr.-Ing. Franz-Josef Behr
2. Prüfer: Prof. Dr.-Ing. Eberhard Gülch

Integration und 3D-Visualisierung heterogener Geobasisdaten

Die Bedeutung virtueller 3D-Modelle wächst in vielen Bereichen des öffentlichen Lebens. Die Erfassung von Daten zur flächenhaften Erzeugung von 3D-Modellen wird von unterschiedlichen Branchen vorangetrieben. 3D-Stadtmodelle finden Anwendung in der Simulation von Schallausbreitung, der Stadtplanung oder den Umweltwissenschaften. Durch den hohen Wiedererkennungswert von Landmarken und die realitätsnahe Darstellung üben 3D-Modelle eine hohe Anziehungskraft auf den Betrachter aus.

Die Erstellung von 3D-Modellen kann auf verschiedene Arten erfolgen, erfordert jedoch immer die Kombination heterogener 2D/3D-Geodaten. Dabei bedingt ein hoher Automatisierungsgrad eine geringere Detailstufe des Modells und ein hoher Detaillierungsgrad weit reichende manuelle Bearbeitung. Ziel dieser Diplomarbeit ist die automatisierte Berechnung und Darstellung von 3D-Modellen aus Geobasisdaten. Im Mittelpunkt steht hierbei die Implementierung eines Prototyps, der das Einbinden verschiedener Datenquellen und die Visualisierung des erstellten Modells ermöglicht. Bei der Entwicklung der Software stehen die Nutzung einer Grafikkbibliothek sowie die Speicherung der Modelle in einer 3D-Geodatenbank im Vordergrund. Dabei kommen Visualisierungsprinzipien aus der Spieleprogrammierung zur Anwendung.

Für die Beschreibung von 3D-Modellen existieren verschiedene Datenformate. Dazu gehört neben dem *VRML*¹-Format auch das *XML*²-basierte Format *X3D (Extensible 3D)*, das es dem Anwender ermöglicht, Geometrien und deren Darstellungsattribute, wie z.B. Textur oder Färbung, in *XML*-artigen Baumstrukturen zu beschreiben. Die Visualisierung dieser Datenformate findet in Bildbetrachtern (*Viewer*) statt. Einige dieser Programme, wie der *Cortona VRML-Client*, sind frei verfügbar, wohingegen andere kommerziell vertrieben werden, beispielsweise *BS Contact VRML/X3D*. Sie nutzen meist Grafikkbibliotheken wie *OpenGL*³ oder *DirectX* zur Visualisierung der Daten. *OpenGL* ist im Gegensatz zu *DirectX* ein offener Standard und wird von Grafikkartenherstellern genutzt und gepflegt. Aus diesem Grund und wegen ihrer Übersichtlichkeit wird diese Bibliothek bei dieser Arbeit zur Visualisierung der 3D-Geometrien genutzt. Einen anderen Ansatz der 3D-Darstellung bieten die Planetenbrowser *Google Earth* und *NASA World Wind*. Sie besitzen Höhenmodelle zur Darstellung der Geländeoberfläche, die mit eigenem Kartenmaterial überlagert sind. *Google Earth* bietet über den Import von *KML*⁴-Dokumenten

¹ VRML: Virtual Reality Modelling Language

² XML: Extensible Markup Language

³ OpenGL: Open Graphics Library

⁴ KML: Keyhole Markup Language

dem Anwender die Möglichkeit, eigene 3D-Modelle zu importieren. In *NASA World Wind* ist ein ähnliches Verfahren zur Zeit nicht implementiert.

Zur Speicherung von 3D-Geodaten können Datenbanktabellen herangezogen werden. Auf diese kann mit Hilfe von Datenbankmanagementsystemen (DBMS) zugegriffen werden.

Das in dieser Arbeit benutzte DBMS ist *PostgreSQL* mit der Erweiterung *PostGIS*. Diese Kombination erlaubt die Speicherung von 2D/3D-Daten im *WKT*⁵- bzw. *EWKT*⁶-Format und ermöglicht räumliche Abfragen. Über eine eingebundene Bibliothek kann aus dem Quellcode eine Verbindung zur Datenbank hergestellt und mittels *SQL*⁷-Befehlen Tabellen erstellt oder Anfragen übermittelt werden.

Um 3D-Modelle berechnen zu können, ist die Verknüpfung mehrerer Datenquellen nötig. Diese umfassen Gebäudegrundrisse sowie Höheninformationen der Erdoberfläche. In dieser Arbeit werden Grundrisse aus den Baden-Württembergischen *BGRUND*-Daten sowie Höhendaten aus dem *DGM*⁸ und *DOM*⁹ im *ASCII*-Format (*XYZ*) genutzt. Zur Texturierung der Geländeoberfläche werden digitale Orthophotos eingebunden. Diese liegen meist im in der Photogrammetrie gängigen *TIF*¹⁰-Format vor. Zur Anzeige der berechneten Modelle ist ein Visualisierungsfenster in das Programm eingebettet. Hierfür wird die Grafikkbibliothek *OpenGL* verwendet. Diese bietet Funktionen zu wichtigen Techniken der 3D-Visualisierung. Dazu gehören *Texture-Mapping* zum Anbringen von Textur auf beliebige Flächen, *Z-Buffering* zur Berechnung der Darstellungsreihenfolge sowie Navigation und Objektselektion.

Um dem Benutzer diese Funktionen übersichtlich zu präsentieren, ist die Einbindung der *MFC* (*Microsoft Foundation Classes*) realisiert. Mit deren Hilfe ist es möglich, verschiedene Steuerelemente auf der Benutzeroberfläche des Programms zu platzieren und zu kontrollieren.

Die während der Diplomarbeit entwickelte Anwendung *EVA3D* (*Erweiterte Visualisierungs-Anwendung 3D*) ist eine dialogfeldbasierte *Windows*-Applikation. Über ein Menü können die verschiedenen Funktionen (Import, Berechne, Export) aufgerufen werden. Den Großteil der Anwendung nimmt das *OpenGL*-Fenster zur Darstellung der 3D-Objekte ein. Es stehen fünf Knöpfe zur Verfügung, mit denen die verschiedenen Navigationsbefehle für das Visualisierungsfenster aktiviert werden können.

Die Anwendung ermöglicht es, die Einstellungen zur Datenbank und den eingebundenen Daten in einer *XML*-Projektdatei zu speichern und zu laden. Das Parsen der Datei wird im Quellcode mit Hilfe der *MSXML*-Bibliothek ausgeführt. Je nach eingebundenen Daten werden im aktuellen Projekt verschiedene Berechnungsfunktionen auswählbar. Das geladene Geländemodell kann zu Dreiecken vermascht und in der Datenbank gespeichert werden. Sind Grundrisse und *DOM*-Daten geladen, kann eine Gebäudeberechnung statt finden. Hierbei werden im ersten Schritt die *DOM*-Punkte in einer Datenbanktabelle gespeichert. Nun werden mit Hilfe des berechneten Gebäudezentroids und einer räumlichen Abfragen die potentiellen Gebäudehöhenpunkte im Umkreis um den Gebäudezentroid geladen. Für jeden dieser Punkte wird in einem Testverfahren überprüft, ob er sich innerhalb des

⁵ WKT: Well-Known Textformat

⁶ EWKT: Extended Well-Known Textformat

⁷ SQL: Structured Query Language

⁸ DGM: Digitales Geländemodell

⁹ DOM: Digitales Oberflächenmodell

¹⁰ TIF: Tagged Image Format

Gebäudeumrings befindet. Mit den erhaltenen Punkten kann die Höhe des Gebäudes abgeleitet werden. Im letzten Schritt werden die Gebäude mit der Geländeoberfläche verschnitten und abgespeichert.

Um die berechneten Objekte darzustellen, wird das *OpenGL*-Fenster initialisiert. Hier werden Darstellungseigenschaften, wie nahe und ferne *Clipping*-Ebene und Projektion, festgelegt. Über einen Timer wird die 3D-Szene im Fenster in einem bestimmten Intervall neu geladen. Dies ermöglicht Bewegungen der Kamera in der erstellten 3D-Welt. Hierzu wird das System des *Double-Buffering* genutzt, um eine konstante Bildrate zu gewährleisten. Im Visualisierungsfenster ist eine räumlich veränderbare Lichtquelle platziert um unterschiedliche Beleuchtungen der dargestellten Objekte zu erzielen. Über einen weiteren Steuerknopf ist es möglich, einzelne Objekte zu selektieren und ihnen aus Pixelbildern erzeugte Textur zuzuweisen.

EVA3D bietet mit der Anbindung an die *GeoMedia*-Schnittstelle *DCDI*¹¹ die Möglichkeit zur Datenaufbereitung. Registriert man *EVA3D* als Anwendungsadapter in der Initialisierungsdatei des *DCDI*, ist es möglich, die Navigation beider Systeme zu synchronisieren und in *GeoMedia* digitalisierte Daten in *EVA3D* zu importieren. Hierzu überprüft *EVA3D* ständig eine Importdatei auf Veränderung und lädt diese gegebenenfalls neu.

Um beliebigen Anwendungen den Datenimport in *EVA3D* zu ermöglichen, ist eine eigene Schnittstelle verwirklicht worden. Durch eine in einer Schemadatei definierte *GML*¹²-Datei ist es möglich, Gebäudedaten in das System zu integrieren.

EVA3D erlaubt den Export in *Google Earth*. Hierzu werden die in *EVA3D* enthaltenen Koordinaten der Geometrien mit Hilfe der eingebundenen *PROJ.4*-Bibliothek in geographische *WGS84*-Koordinaten überführt und die 3D-Modelle in *KML*-Notation in die Exportdatei geschrieben. Das Beschreibungsformat *X3D* wird zum Export der 3D-Geometrien ebenfalls unterstützt. Hier werden die einzelnen Geometrien als *IndexedFaceSet* abgespeichert, denen Subtexturen oder Farbwerte zugewiesen werden können.

Durch den hohen Automatisierungsgrad der Modellierung und die Nutzung von 2D-Geobasisdaten ist die entwickelte Anwendung als Ergänzung zu gängigen 2D-Geoinformationssystemen zu sehen. Das 3D-Modell bietet eine neue Sichtweise auf die darzustellenden Daten und erlaubt Anwendungen in zahlreichen Gebieten. So kann das bei dieser Arbeit entwickelte 3D-Modell für Simulationen von Hochwasser oder Schallausbreitung direkt genutzt werden. Es kann aber auch als Grundgerüst für ein komplexeres 3D-Stadtmodell dienen, mit dem beispielsweise städtebauliche Planungen durchgeführt werden können. So liefert diese Arbeit ein mächtiges Werkzeug zur automatisierten Erstellung vielseitig nutzbarer 3D-Modelle.

Die übersichtliche interne Datenstruktur erlaubt die Erweiterung der Software in verschiedenen Gebieten. Denkbar sind die Einbindung weiterer Datenformate, eine Erkennung verschiedener Klein- und Dachformen sowie die Integration weiterer visueller Effekte.

¹¹ DCDI: Desktop Client Desktop Interface

¹² GML: Geography Markup Language



Darstellung eines Testprojektes in *EVA3D*. Zu sehen ist ein Geländemodell, das mit einer georeferenzierten topographischen Karte texturiert ist. Die Gebäudehöhen wurden aus DOM-Daten berechnet. Gut zu sehen ist die unterschiedliche Beleuchtung der Fassaden.



Vergleich zwischen Wirklichkeit und erreichter Darstellung in *EVA3D*.