

Otto-von-Guericke Universität Magdeburg



Fakultät für Informatik  
Institut für Simulation und Graphik

# Diplomarbeit

## SEMI-AUTOMATISCHE VERFAHREN FÜR DIE FOTOREALISTISCHE TEXTURIERUNG VIRTUELLER STADTMODELLE

**Verfasser:** Birgit Meisert

**Betreuer:** Prof. Dr. Bernhard Preim  
(Universität Magdeburg)

Dr.-Ing. Rüdiger Mecke  
(Fraunhofer IFF Magdeburg)

**Abgabe:** 12.03.2008

**Meisert, Birgit:**

*Semi-automatische Verfahren für die fotorealistische Texturierung virtueller  
Stadtmodelle*

Diplomarbeit, Otto-von-Guericke-Universität  
Magdeburg, 2008.

**INHALTSVERZEICHNIS**

<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>1. Einführung</b>	<b>1</b>
1.1. Motivation	1
1.2. Zielstellung	1
1.3. Ergebnisse	2
1.4. Aufbau der Arbeit	2
<b>2. Grundlagen</b>	<b>5</b>
2.1. 3D-Stadtmodelle	5
2.2. Geografische Informationssysteme	8
2.3. Methoden zur Ermittlung von Modelldaten / Datenquellen	10
2.3.1. Messverfahren	10
2.3.2. Karten und Pläne	11
2.3.3. Fotografien	11
2.3.4. Andere Datenbanken	12
2.4. Objekte in 3D-Stadtmodellen	13
2.4.1. Gebäude und Bauwerke	13
2.4.2. Grund und Boden	14
2.4.3. Vegetation	15
2.4.4. Straßenbegleitende Objekte	16
2.5. Texturierung von Geometriemodellen	16
2.5.1. Texturen	16
2.5.2. Gewinnung von Texturen	20
2.5.3. Texturierung zur Visualisierung von Daten in 3D-Stadtmodellen	25
2.5.4. Texturierungsmethoden	27
2.5.5. Orientierung von geometrischen Objekten	29
<b>3. Stand der Technik</b>	<b>31</b>
3.1. Nutzungspotentiale von 3D-Stadtmodellen	31
3.2. Arbeitsablauf zur Erstellung eines 3D-Stadtmodells	32
3.3. Recherche in ausgewählten Kommunen	34
3.3.1. Eisleben	35
3.3.2. Wittenberg	35
3.3.3. Staßfurt	36
3.3.4. Zusammenfassung der Recherche	36

<b>3.4. Visualisierungsprogramme mit Geobezug</b>	<b>38</b>
3.4.1. Spezielle Visualisierungssysteme mit Geobezug	38
3.4.2. Bestehende GI-Systeme mit 3D-Visualisierungskomponenten	41
3.4.3. 3D-Konstruktionskomponenten als PlugIn	42
<b>3.5. Existierende Stadtmodelle</b>	<b>44</b>
3.5.1. Private bzw. unternehmerische Auftraggeber	44
3.5.2. Behördliche Auftraggeber	45
<b>3.6. Innovative fotobasierte Techniken</b>	<b>47</b>
<b>3.7. Zusammenfassung</b>	<b>49</b>
<b>4. Analyse und Entwurf</b>	<b>51</b>
<b>4.1. Anforderungsanalyse</b>	<b>51</b>
<b>4.2. Vegetation und straßenbegleitende Objekte</b>	<b>52</b>
<b>4.3. Gebäude und Bauwerke</b>	<b>53</b>
<b>4.4. Grund und Boden</b>	<b>55</b>
4.4.1. Ablaufanalyse zur Texturierung von Grund und Boden	55
4.4.2. Schematische Darstellung zur Generierung digitaler Straßenpolygone	57
4.4.3. Ausgangsgeometrie	59
4.4.4. Geometrieuntersuchung zur Optimierung der Texturierung	59
4.4.5. Manuelle Korrekturmöglichkeiten für die Texturierung	70
4.4.6. Krümmungen des Straßenverlaufs in der Realität	72
4.4.7. Weitere Möglichkeiten zur Texturierung von Krümmungen	73
4.4.8. Detaillierung der Straßengeometrie	74
<b>4.5. Dialoggestaltung</b>	<b>81</b>
4.5.1. Prinzipien Dialogentwurf	81
4.5.2. Entwurf	81
<b>4.6. Qualitätsanalyse von Texturen</b>	<b>84</b>
4.6.1. Betrachtung von Höntzsch	84
4.6.2. Erweiterung der Herangehensweise von Höntzsch	85
4.6.3. Kacheln von Texturen	86
<b>5. Prototypische Umsetzung</b>	<b>87</b>
<b>5.1. Entwicklungswerkzeuge</b>	<b>87</b>
<b>5.2. Benutzeroberfläche</b>	<b>87</b>
<b>5.3. Kantenverfolgung zur Krümmungserkennung</b>	<b>89</b>
<b>5.4. Texturanpassung mit Hilfe der Objektausrichtung</b>	<b>94</b>
<b>5.5. Bordsteingenerierung und Bordsteinintegration</b>	<b>96</b>
<b>5.6. Ergebnisbilder der Texturierung und Bordsteingenerierung</b>	<b>99</b>

**6. Zusammenfassung / Ergebnisse** \_\_\_\_\_ **103**

**7. Ausblick** \_\_\_\_\_ **105**

**Literaturverzeichnis** \_\_\_\_\_ **107**

**Anhang** \_\_\_\_\_ **112**



## Abkürzungsverzeichnis

1D	-	Eindimensional
2D	-	Zweidimensional
3D	-	Dreidimensional
AABB	-	Axis Aligned Bounding Box
ALB	-	Automatisiertes Liegenschaftsbuch
ALK	-	Automatisierte Liegenschaftskarte
CAD	-	Computer Aided Design
DGM	-	Digitales Geländemodell
DOM	-	Digitales Oberflächenmodell
GIS, GI-System	-	Geografisches Informationssystem
LOD	-	Level Of Detail
LVerGeo	-	Landesamt für Vermessung und Geoinformation Sachsen-Anhalt
OOBB	-	Object Orientated Bounding Box
VR	-	Virtual Reality



## **1. Einführung**

### **1.1. Motivation**

Interaktive 3D-Modelle und virtuelle Technologien werden in den Bereichen der Stadtplanung, des Stadt- und Standortmarketings sowie der Denkmalpflege und der historischen Stadtdokumentation immer stärker als innovative und hilfreiche Werkzeuge begriffen. Diese neue Akzeptanz wurde durch die Fortschritte der jüngeren Vergangenheit im Bereich der semi-automatischen Modellgenerierung sowie durch neue Möglichkeiten in der grafisch anspruchsvollen Darstellung großvolumiger 3D-Modelle ermöglicht.

Der fotorealistischen Darstellung urbaner Räume kommt in diesem Zusammenhang eine besondere Bedeutung zu. Die wirklichkeitsgetreue Darstellung im virtuellen Stadtmodell ist bei der Beurteilung der verschiedenen Fragestellungen der oben genannten Bereiche oftmals entscheidend für ein erfolgreiches Arbeitsergebnis. Bei der Generierung von virtuellen Stadtmodellen werden daher 3D-Objekte bzw. Bereiche von besonderer Bedeutung und hoher Priorität meist mit fotorealistischen Texturen versehen. Diese Art der Texturierung ist jedoch gegenwärtig meist noch mit hohem manuellem Aufwand verbunden.

Die fotorealistische Texturierung bzw. die hierbei verwendeten Verfahren haben starken Einfluss auf die Qualität des Gesamtmodells sowie auf den Zeitaufwand, der zu dessen Erstellung notwendig ist. Fortschrittliche Methoden der Texturierung können in diesem Zusammenhang die Qualität des 3D-Modells erhöhen und gleichsam den zur Modellgenerierung erforderlichen Zeitaufwand verringern.

### **1.2. Zielstellung**

In der vorliegenden Diplomarbeit werden Verfahren zur fotorealistischen Texturierung von digitalen 3D-Objekten untersucht und fortschrittliche Methoden einer semi-automatischen Texturierung entwickelt bzw. in den Prozess der 3D-Modellgenerierung integriert. Die neu zu entwickelnden Methoden sollen dabei die Qualität und die Akzeptanz des virtuellen Modells insgesamt erhöhen und so dazu beitragen, neue Potentiale in den eingangs aufgeführten Anwendungsgebieten zu eröffnen.

Zur Erreichung dieser Ziele ist es notwendig, die in den Kommunen zur Verfügung stehenden Systeme und Ausgangsdaten sowie die bestehenden Arbeitsprozesse zur Generierung von 3D-Stadtmodellen zu betrachten. Basierend auf dieser Grundlagenermittlung ist zu untersuchen, welche Methoden bei der Generierung der verschiedenen Modellkomponenten (Grund und Boden,

Gebäude, Vegetation usw.) eingesetzt und welche Verbesserungspotentiale hier sichtbar werden.

Im Rahmen der Arbeit soll die fotorealistische Texturierung des Grund und Bodens mit seiner Straßenstruktur, seinen Wegen und Bordsteinen detailliert untersucht und weiterentwickelt werden. Die zu entwickelnden Verfahren der fotorealistischen Texturierung sollen hierbei so weit wie möglich auf manuelle Eingaben verzichten. Die semi-automatischen Ansätze sollen zur Qualitätssteigerung in der grafischen Darstellung sowie zur Zeitreduktion bei der Erstellung der Straßenstrukturen und Freiflächen führen. Ziel ist es, diese Verfahren in ihrer Bedienung intuitiv zu gestalten, so dass die Akzeptanz der Nutzer in den eingangs beschriebenen Anwendungsgebieten gesichert ist.

### 1.3. Ergebnisse

In der vorliegenden Arbeit wurden mögliche Probleme und Vorgehensweisen diskutiert, um die Problematik des Themengebietes zu erfassen und darauf aufbauend neue Methoden der semi-automatischen Texturierung zu erarbeiten. Die resultierende Anwendung ist in der Lage, Gehwege und Bordsteine zu erstellen und diese mit passenden Texturen zu versehen. Zudem kann die Ausrichtung der Textur an der Orientierung des Polygons angepasst werden. Für diese Vorgänge sind einige Vorüberlegungen und Schritte notwendig, die in der vorliegenden Arbeit näher erläutert und als Algorithmus umgesetzt wurden.

### 1.4. Aufbau der Arbeit

**Kapitel 2** widmet sich der Grundlagenbetrachtung zum Thema 3D-Stadtmodelle. Es erfolgt die Klärung themenrelevanter Begriffe und die Vorstellung von Methoden zur Daten- und Texturgewinnung.

**Kapitel 3** beschäftigt sich mit dem aktuellen Stand der Technik. Es werden existierende Stadtmodelle und Programme zur Erstellung von Stadtmodellen vorgestellt und Ergebnisse aus Recherchen in Kommunen dargelegt.

**Kapitel 4** beinhaltet den im Rahmen der vorliegenden Arbeit entwickelten Konzeptentwurf zur automatischen Texturierung virtueller Stadtmodelle. Der Schwerpunkt des Konzepts liegt im Bereich der Grund- und Bodentexturierung.

**Kapitel 5** beschreibt die prototypische Umsetzung von Teilen des Konzepts. Einige Verfahren werden näher erläutert.

**Kapitel 6** fasst die Arbeit zusammen und zeigt die erreichten Ergebnisse der prototypischen Umsetzung.

**Kapitel 7** gibt einen Ausblick auf weitere Arbeiten im Themengebiet der Texturierung von 3D-Stadtmodellen. Insbesondere werden Ansätze zur Erweiterung und Verbesserung der prototypischen Umsetzung aufgezeigt.



## 2. Grundlagen

In diesem Kapitel werden themenrelevante Grundlagen vorgestellt und erläutert. Es erfolgt eine Einführung in Begriffe und Techniken zu den Themen digitale 3D-Stadtmodelle, Datengewinnung, Texturierung und Texturen.

### 2.1. 3D-Stadtmodelle

Stadtmodelle in der herkömmlichen Bedeutung sind in der Regel aus Gips, Kunststoffen oder Holz gefertigte Miniaturmodelle der realen Welt, bestehend aus nachfolgenden Darstellungskomponenten:

- Grund und Boden,
- Bauwerke, Gebäude, Denkmäler,
- sekundäre straßenbegleitende Objekte (Straßenbeleuchtung, Haltestellen, Verkehrsschilder, usw.) und
- Vegetation.



Abbildung 2.1: Hof in Eisleben - herkömmliches Stadtmodell aus Holz (Quelle:[IFF])

Herkömmliche Stadtmodelle sind meist stark abstrahiert und verzichten auf Texturierung und Farbgebung [Kop04] (s. Abb.2.1). Sie haben den Nachteil, dass sie teuer in der Herstellung und unflexibel gegenüber Veränderungen sind. Im Vergleich zu den herkömmlichen bieten die digitalen 3D-Stadtmodelle einige Vorteile.

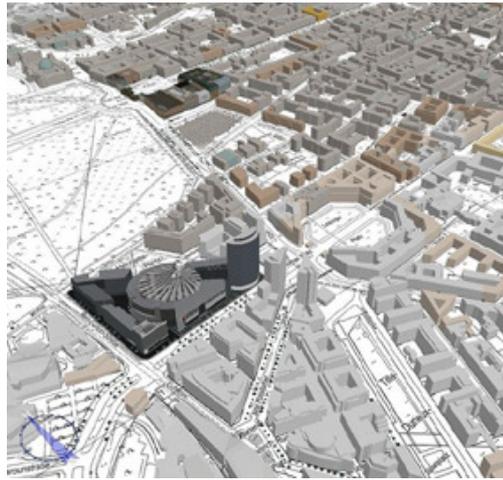


Abbildung 2.2: virtuelles Stadtmodell Berlin (Quelle: [Berlin])

Ein digitales 3D-Stadtmodell ist eine dreidimensionale digitale Abbildung einer Stadt bzw. eines –bereiches und enthält die nachfolgende zusätzliche Darstellungskomponente:

- Zusatzinformationen (z.B. hausspezifische Infos wie Baujahr und Nutzung).

Durch die Gesamtheit und Variabilität der Darstellungskomponenten sowie der zusätzlichen Möglichkeit der Sachdatenbindung, ist das digitale Stadtmodell wesentlich informativer als ein herkömmliches Stadtmodell, das nur aus den Baukörpern besteht.

Durch die Nutzung von Texturen können dem digitalen Stadtmodell je nach Anwendungszweck unterschiedliche „Layouts“ angepasst werden. Diese reichen von schematischen und abstrakten Darstellungen (s. Abb. 2.3) bis zur realistischen Texturierung, um das Original weitestgehend wirklichkeitsgetreu darzustellen.



Abbildung 2.3: Nichtphotorealistische Darstellung von Gebäuden (Quelle: [3DGLX])

### Level of Detail in 3D-Stadtmodellen

Das Level of Detail (LOD) dient im klassischen Sinn zur Steigerung der Rechengeschwindigkeit des Computers. Hierbei werden abhängig von der Entfernung der Kamera zu einem Objekt die Objektdetails dargestellt. Befindet man sich im Nahbereich, wird eine hohe Detailgenauigkeit geboten, entfernt man sich, wird diese verringert.

Im Bereich der Stadtmodelle bezeichnet LOD den Detaillierungsgrad in der Datenerfassung. Grobe Modelle enthalten einfache Blockmodelle von Gebäuden, detailreichere Modelle bestehen aus genauer modellierten Gebäuden mit korrekten Texturen sowie Vegetation und Straßenmöbeln. [Lan04]

Oft ist der Detailreichtum in den Kerngebieten größer als in den Randgebieten einer Stadt, da zu Beginn der Modellerstellung das Augenmerk oft auf diesem Gebiet liegt. Dadurch fällt das LOD meist vom Kerngebiet zum Randgebiet hin ab.

Für die Einteilung in LOD 1 bis LOD 4 gibt es verschiedene Varianten. Je nach Literatur stellen die LOD-Stufen unterschiedliche Detaillierungsgrade der Gebäude dar. Die weiteren Ausführungen beziehen sich auf das nachfolgende Modell.

#### **Detaillierungsgrade eines Gebäudes:** [Pet05] (s. Abb. 2.4)

LOD 0: Regionalmodell	DGM <sup>1</sup> (2,5D) mit Orthofoto und Flächennutzung
LOD 1: Klötzchenmodell	ohne Dachform und Textur
LOD 2: Detailliertes Modell	differenzierte Dachstrukturen, Textur optional, Vegetationsmerkmale
LOD 3: Stadt-/ Standortmodell	fein ausdifferenziertes Architekturmodell mit Textur, Vegetation und Straßenmöbel
LOD 4: Innenraummodell	begehbare Architekturmodelle

---

<sup>1</sup> DGM - Digitales Geländemodell

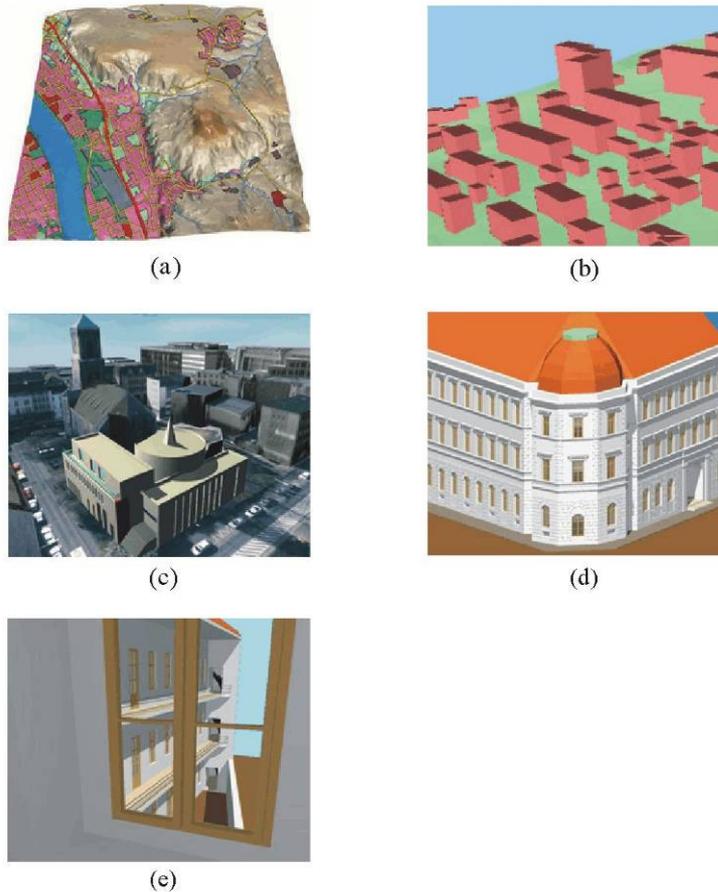


Abbildung 2.4: (a) LOD 0 - Regionalmodell,  
 (b) LOD 1 - Klötzchenmodell,  
 (c) LOD 2 - Detailliertes Modell,  
 (d) LOD 3 - Stadt- /Standortmodell,  
 (e) LOD 4 - Innenraummodell

## 2.2. Geografische Informationssysteme

Ein Geographisches Informationssystem (GIS oder GI-System) dient der Erfassung, Speicherung, Prüfung, Verwaltung, Manipulation, Integration, Analyse und Visualisierung von raumbezogenen Daten und ihrer Beziehungen. [GisN] Nach gängigem Verständnis besteht ein GIS aus einer räumlich adressierbaren Datenbank und geeigneter, darauf abgestimmter Anwendungssoftware.

GI-Systeme können in verschiedenen Ausprägungen vorliegen, wie beispielsweise die Landinformationssysteme (LIS), Kommunale Informationssysteme (KIS) und Bodeninformationssysteme (BIS) [GisW].

Neben diesen Grundformen existieren verschiedene Fachschalen, die das System nach Bedarf ergänzen. Unter einer Fachschale versteht man ein eigenständiges Modul, in der Regel auf Basis eines Hersteller-GIS.

Die Anwendungsbereiche für GIS sind sehr vielseitig. Einer der größten Anwender ist das Vermessungs- und Katasterwesen. Weitere Einsatzgebiete sind:

- Umwelt (UIS - Umwelt-Informationssysteme),
- Planungswesen (Stadtplanung, Raumordnung),
- Netz-Informationssysteme der Ver- und Entsorgung und Telekommunikation (NIS) und
- Fachinformationssysteme (FIS).

Die Datenbasis vieler GI-Systeme bildet das Liegenschaftskataster. Es setzt sich aus der Automatisierten Liegenschaftskarte (ALK) und dem Automatisierten Liegenschaftsbuch (ALB) zusammen.

Die Liegenschaftskarte im Land Sachsen-Anhalt weist folgende obligatorische Angaben nach [KM05]:

- Grenzen der politischen Einheiten (Landes-, Regierungsbezirks-, Landkreis-, Gemeindegrenzen), Katasteramtsbezirke (Gemarkungs-, Flur-, Flurstücksgrenzen), Nutzungsarten nach tatsächlicher Nutzung und nach gesetzlicher Klassifizierung (Bodenschätzungsgrenzen),
- Grenzpunkte,
- Flurstücksbezeichnung (Gemarkung, Flur, Flurstücksnummer),
- Lagebezeichnung (Feldlage, Gewinn, Hausnummer, Straßen-, Wege-, Gewässernamen),
- Gebäudebestand (Grundriss, keine Gebäudeartendifferenzierung),
- Nutzungsarten (tatsächliche Ordnungsstufe für Straßen und Gewässer),
- Ertragsfähigkeit (Klassenzeichen, Wertzahlen)

Das Liegenschaftsbuch im Land Sachsen-Anhalt enthält Katasterstammdaten zur Bezeichnung und Beschreibung der Liegenschaft mit den Eigentumsangaben und nachrichtliche Fremdanlagen. Der Datenbestand des Liegenschaftsbuches ist logisch strukturiert nach den Einheiten Flurstück, Eigentümer (Bestand), Buchungsstelle im Grundbuch (Grundstück), Gemarkung und Gemeinde. Diesen Einheiten ist unter einem Ordnungsmerkmal (Kennzeichen) jeweils ein fester, sachlich zusammenhängender Datenkatalog zugeordnet. Durch gemeinsame Verknüpfungsmerkmale werden die Datenkataloge miteinander verbunden. [KM05]

Neben den Informationen von ALK und ALB sind Sachdaten wichtig, um ein informatives 3D-Stadtmodell zu erstellen. Wesentlich für eine automatisierte Texturierung sind die festgeschriebenen Objektschlüssel der ALK, welche die Objektart kennzeichnen. Für Gebäude gibt es den allgemeinen Schlüssel 1001. Sind nähere Informationen über die Nutzung bekannt, können die Gebäude mit einem spezifischeren Schlüssel versehen werden. In dem Objektschlüsselkatalog des Landes Sachsen-Anhalt (OSKA-LSA) steht beispielsweise der Schlüssel 1112 für ein Rathaus und 1411 für ein Bürogebäude.

Auch für Flächennutzungsarten wird dieses System angewandt, z.B. erhält Ackerland den Schlüssel 2110 und Verkehrsflächen sind an dem Schlüssel 5000 zu erkennen. Innerhalb der einzelnen Gruppen (z.B. Verkehrsflächen) erfolgen weitere Unterteilungen (z.B. Straße, Weg).

### 2.3. Methoden zur Ermittlung von Modelldaten / Datenquellen

An dieser Stelle soll auf Quellen von Modelldaten eingegangen werden, um aufzuzeigen, welche Vielfalt existiert und welche Daten sich für die automatische Texturierung nutzen lassen.

#### 2.3.1. Messverfahren

Um ein Digitales Geländemodell (DGM) zu erhalten, kann u.a. die Photogrammetrie und das LIDAR-Verfahren eingesetzt werden.

„**Photogrammetrie** oder Bildmessung ist die Rekonstruktion eines Objektes nach Form und Lage aus fotografischen Aufnahmen. [FH68]“ Dies sagte Finsterwalder vor ca. 40 Jahren. Diese Methode bildet die Tiefenwahrnehmung des Menschen nach, in dem aus Stereobildpaaren mit Hilfe von Korrelationsverfahren ein Digitales Oberflächenmodell erstellt wird. Die aufgenommenen Bilder sind nach Entzerrung auch als Orthofotos<sup>2</sup> nutzbar. Probleme hat dieses Verfahren bei engen Straßen und hohen Gebäuden, da in diesen Fällen Teile der Erdoberfläche verdeckt werden können.

Die **LIDAR**-Technik ("*light detection and ranging*"), auch als *3D-laser-scanning* bekannt, ist eine Methode zur Entfernungs- und Geschwindigkeitsmessung sowie zur Fernmessung atmosphärischer Parameter [BV05].

Die Methode misst die Zeit zwischen dem Senden und Empfangen von Laser-Signalen bei bekannter „Flugbahn“. Aus diesen Daten wird der Abstand des Zielpunktes zum Laser und anschließend der exakte Standort des Punktes berechnet. Als Ergebnis erhält man eine Punktedatei mit dreidimensionalen Koordinaten von nahezu jeder Oberfläche. Mit den neuesten Systemen des *laser-scannings* lassen sich komplette Oberflächen, egal ob Gebäude oder Felswand, in wenigen Sekunden bei einer hohen Messgenauigkeit aufnehmen. [CR06]

Nachteilig sind hier die derzeitigen hohen Anschaffungskosten der Gerätschaften und die Abhängigkeit der Qualität des Ergebnisses von der Erfahrung des Auswerters.

Das Ergebnis der beiden vorgestellten Verfahren sind Punktwolken, die trianguliert werden müssen. Entsteht dabei eine einzelne große Fläche, kann diese nur mit einer einzelnen Textur belegt werden. Um eine differenzierte

---

<sup>2</sup> Orthofoto – siehe Seite 9 Orthofoto

automatische Texturierung zu ermöglichen, ist eine Einteilung in mehrere Flächen und eine Klassifizierung dieser notwendig.

### 2.3.2. Karten und Pläne

Karten und Pläne sind in vielen Kommunen bereits vorhanden und können ggf. zur Gewinnung von Stadtmodelldaten genutzt werden. Sie werden in vektorbasierte und bildbasierte Karten unterschieden.

Zu **vektorbasierten Karten** zählen Liegenschafts- bzw. Katasterkarten. Sie sind die maßstäbliche verkleinerte Darstellung aller Liegenschaften (Gebäude und Flurstücke) in 2D und flächendeckend verfügbar.

Ein weiteres Beispiel für eine vektorbasierte Karte ist die Bodennutzungskarte. Die benötigten Daten zur Aufstellung einer Bodennutzungskarte sind in den GIS der Kommunen meist vorhanden.

Aus diesen Karten können neben den geometrischen Daten, wie Gebäudegrundrisse und Flurstückspunkte, auch Sachdaten entnommen werden. Für die automatische Texturierung kann die Angabe zur „tatsächlichen Nutzung“ von Gebäuden und Grundstücken benutzt werden.

Ein Beispiel für eine **bildbasierte Karte** ist die Stadtkarte, oder auch der Stadtplan. Eine Stadtkarte eignet sich jedoch auf Grund von Problemen in der Generalisierung wenig für eine Nutzung im 3D-Stadtmodell. Da einige Straßen aus Visualisierungsgründen oft viel zu breit eingezeichnet werden, sind diese nicht maßstäblich (Abb. 2.5). Unter Umständen lassen sich einige Daten, wie der Beschriftungstext, für NPR-Visualisierungen nutzen.



Abbildung 2.5: Die Nutzung einer Stadtkarte in einem 3D-Stadtmodell kann zu Problemen in der Übereinstimmung von Flächengrößen führen (Quelle [Lan04])

### 2.3.3. Fotografien

In den meisten Kommunen sind Fotografien vorhanden. Es gibt Luftbilder, terrestrisch aufgenommene Fotografien oder Satellitenfotos. Der Umfang der terrestrisch aufgenommenen Fotografien in den einzelnen Kommunen kann sehr unterschiedlich sein.

**Luftbilder** werden i. d. R. von Flugzeugen aus aufgenommen, sie liegen als Orthofotos und Schrägbildaufnahmen vor. Kommunen können diese unter anderem vom Landesamt für Vermessung und Geoinformation (LVermGeo) beziehen.

Ein **Orthofoto** ist eine verzerrungsfreie und maßstabsgetreue fotografische Abbildung der Erdoberfläche. Der fotografische Inhalt besteht meist aus mehreren Luftbildern. Generell unterscheidet man zwischen „Pseudoorthofotos“ und „echten“ Orthofotos [Net05].

Bei den Pseudoorthofotos sind die Gebäudeverzerrungen nicht korrigiert, die bei der Aufnahme durch das vergleichsweise tief fliegende Flugzeug entstehen.

Echte Orthofotos lassen sich aus einem hoch aufgelösten DOM<sup>3</sup> und mehreren Luftbildaufnahmen generieren. Alle Gebäude sind dann lediglich in der Draufsicht enthalten.

Bilder von Stadtgebieten können ebenfalls aus **Satellitenfotos** gewonnen werden. Satellitenfotos existieren für die gesamte Erdoberfläche, jedoch haben verfügbare (für nicht-militärische Zwecke) Satellitenfotos eine geringere Auflösung als Luftbilder. Zudem können Verdeckungen durch Wolken auftreten.

**Schrägbildaufnahmen** (Pictometry) sind ein relativ neues Luftbildformat. Die Luftbilder sind aus der Schrägperspektive aufgenommen und zeigen jeden Standort im städtischen Umfeld aus den vier Himmelsrichtungen. Sind die Aufnahmen georeferenziert, d.h. mit den geografischen Daten verbunden, können die Gebäudetexturen automatisch extrahiert werden.

Die BLOM-Gruppe ist aktuell dabei, Schrägbild-Datenbanken von über 900 europäischen Städten aufzubauen [Blom].

**Terrestrisch aufgenommene Bilder** können manuell oder aus speziell ausgerüsteten Fahrzeugen aufgenommen werden. Während der Aufnahme aus Fahrzeugen wird die Position per GPS<sup>4</sup> bestimmt und mit den Aufnahmen abgespeichert. Diese Verfahren werden in der Regel eingesetzt, um Fassadenfotos zu erhalten.

Aus den verschiedenen Bildtypen lassen sich unter Nutzung spezieller Verfahren und Messtechniken Geometrien für 3D-Stadtmodelle gewinnen. Auch Texturen können bei der Verknüpfung mit den geographischen Daten automatisch extrahiert und den 3D-Geometrien zugewiesen werden. Sind die Bilder nicht georeferenziert, erfolgt eine manuelle Bearbeitung, um Texturen zu erhalten (s. Abschn. 2.5.).

#### 2.3.4. Andere Datenbanken

Der Inhalt städtischer Datenbanken kann sehr unterschiedlich sein. Es können z.B. Datenbanken zu Vegetationsbeständen, Straßennamen und Hausnummern sowie Verkehrsnetzen vorhanden sein. Diese lassen sich zur automatischen Platzierung

---

<sup>3</sup> DOM - Digitales Oberflächenmodell

<sup>4</sup> GPS - Global Positioning System

von Vegetationsobjekten und zur Bereitstellung von Sachdaten nutzen. Für die automatische Texturierung können Daten zum Alter einzelner Objekte Verwendung finden. Durch entsprechende Methoden in der Texturgenerierung kann der Textur ein älteres Erscheinungsbild verliehen werden.

## 2.4. Objekte in 3D-Stadtmodellen

Zu den Objekten einer Stadt zählen neben den Gebäuden und der Erdoberfläche, die Vegetation und die so genannten straßenbegleitenden Objekte. Je nach Anwendungszweck sind die einzelnen Objektgruppen in unterschiedlicher Ausprägung und Anzahl vorhanden. Im folgenden Abschnitt wird die Generierung dieser Geometrien für ein 3D-Stadtmodell betrachtet.

### 2.4.1. Gebäude und Bauwerke

In vielen Anwendungen sind **Gebäude** der wichtigste Bestandteil des 3D-Modells. Für kleine Einzellösungen werden die Gebäude oft mit hohem Aufwand manuell erstellt. Um ganze Städte auf diese Weise zu erstellen, ist der Aufwand zu groß, es besteht das Bestreben nach einer größtmöglichen Automatisierung.

Aus diesem Grund sind auf diesem Gebiet bereits einige semi-automatische und automatische Rekonstruktionsverfahren entstanden. Jedoch sind Dachgauben, Überhänge und Balkone schwer automatisch zu erfassen. An dieser Stelle muss oft manuell verfeinert werden.

Je nach LOD-Anforderung ist die Verwendung verschiedener Methoden möglich. Ein einfaches Blockmodell kann aus den Grundrissdaten der ALK erstellt werden. Je nach Art des Objektes (Typschlüsselnummer) und/oder der Anzahl der Stockwerke ist eine festgelegte Höhe zu verwenden.

Um Gebäude des LOD 2 zu erhalten, sind zu den Blockmodellen Dachgeometrien hinzuzufügen. Diese können aus Datenbanken der Stadtverwaltung entnommen werden, sofern sie erfasst wurden. Andernfalls können LIDAR-Daten oder photogrammetrische Informationen zur Rekonstruktion Verwendung finden.

Zur Rekonstruktion von Dachflächen existieren verschiedene Ansätze. Beim parametrischen Ansatz wird von wenigen generischen Dachformen ausgegangen und versucht, die optimalen Parameter für die Dachform zu schätzen.

LOD 3 Gebäude sind lt. Festlegung (S. 8) fein ausdifferenzierte Gebäudemodelle. Diese können z.B. durch manuelles Modellieren in CAD-Programmen entstehen.

Innenraummodelle (LOD 4) sind selten Bestandteil von 3D-Stadtmodellen. Auch sie können nur durch manuelle Rekonstruktion generiert werden.

Gebäude können mit Texturen aus verschiedenen Quellen texturiert werden. Dazu gehören Luft- und Schrägbilder, Fassadenfotos und Standardoberflächentexturen. Mehr zur Gewinnung und den Vor- und Nachteilen folgt im Abschnitt 2.5.

**Bauwerke**, wie Brücken, Denkmäler und Skulpturen zeichnen sich durch besondere und oft einmalige Eigenschaften aus. Dadurch ist eine automatische

Erfassung sehr schwer. Eine realistische Rekonstruktion ist nur auf dem manuellen Wege möglich. Andernfalls können Standardmodelle eingesetzt werden.

#### 2.4.2. Grund und Boden

Wie oben bereits erwähnt, liegt bei vielen Anwendungen der Focus auf den Gebäudemodellen. Eine ansprechende automatische Generierung von Straßenmodellen wurde daher bislang vernachlässigt. Die Texturierung erfolgt meist mit Luftbildern, was zu einem Missverhältnis zwischen dem optischen Eindruck der Gebäude und der Bodengeometrie führt (s. Abb. 2.6 b).

Die Höhendaten des DGMS können durch Messverfahren der LIDAR- und Photogrammetrie-Technik gewonnen werden. Das Ergebnis ist eine Punktwolke, die anschließend trianguliert werden muss. Die gewählte Textur wird auf die gesamte Fläche projiziert.

Eine Einteilung der Bodenfläche in Straßen, Grund- oder Flurstücke ermöglicht eine differenzierte Texturierung. Um diese Einteilung zu erhalten, können die 2D-Daten des Liegenschaftskatasters verwendet werden. Jedoch sind hier keine Höhendaten vorhanden. Die entstehenden Flächen liegen in einer Ebene.

Das Verbinden beider Datenquellen ermöglicht eine differenzierte Texturdarstellung mit korrekten Höhenangaben. Bei der Texturierung sollten Verzerrungen der Textur vermieden werden.

Je nach gewünschtem LOD kann auch die Straßentexturierung angepasst werden. Dies erfolgt durch das Zusammenfassen von einzelnen Flächen. In einem niedrigen LOD könnten beispielsweise Gehweg und Fahrbahn zu einer Fläche zusammengefasst und mit der gleichen Textur versehen werden.



(a)



(b)

Abbildung 2.6: Blick aus (a) der Vogelperspektive und (b) der Fußgängerperspektive auf das Stadtmodell Magdeburg [IFF]. In (b) sind die Nachteile der Verwendung eines Luftbildes als Bodentextur deutlich erkennbar.

### 2.4.3. Vegetation

Die folgenden Modellierungsarten wurden in [Kop02] vorgestellt und werden in Spiel- und Trickfilmumgebungen sowie teilweise bereits in 3D-Stadtmodellen eingesetzt. Zwischen den Verfahren kann im Hinblick auf die Komplexität der Geometrien folgende Reihenfolge aufgestellt werden: Das Kugelbaummodell ist das Modell mit der geringsten Komplexität, gefolgt von der Ebenentechnik. Das Modell mit der höchsten Komplexität sind die L-Systeme.

Bei **Kugelbaumobjekten** werden die Bäume in ihre Hauptbestandteile zerlegt, welche aus einfachen geometrischen Volumenkörpern nachgebildet werden. Im einfachsten Ansatz werden diese Körper mit schlichten Farben versehen. Um die Realitätswirkung zu erhöhen, ist eine Erweiterung um eine Textur möglich.

In der **Ebenentechnik** werden die Bäume als Tafelflächen dargestellt, worauf die Textur, das Bild einer Ansichtsseite des Baumes projiziert wird. Diese Technik ist für Objektansammlungen wie Wälder, aber nicht für Einzelobjekte geeignet. Die statische Ansicht der Objekte würde zu einer unnatürlichen Szene führen. Werden mehrere Tafelobjekte für die Krone genutzt, entstehen so genannte Kreuztafelobjekte, die mit geringem Aufwand eine relativ realitätsnahe Darstellung ermöglichen.

Das Ziel der Vegetationsmodellierung mit einem sog. **L-System** ist die Erzeugung einer möglichst realistischen Abbildung. Dies wird durch die Nachbildung des natürlichen Bildungs- und Wachstumsprozesses eines Objektes erreicht. Es werden Elementketten der einzelnen Elemente, wie Blätter und Äste, ausgehend von einem Startelement unter Anwendung einer Produktionsregel, generiert und dargestellt.

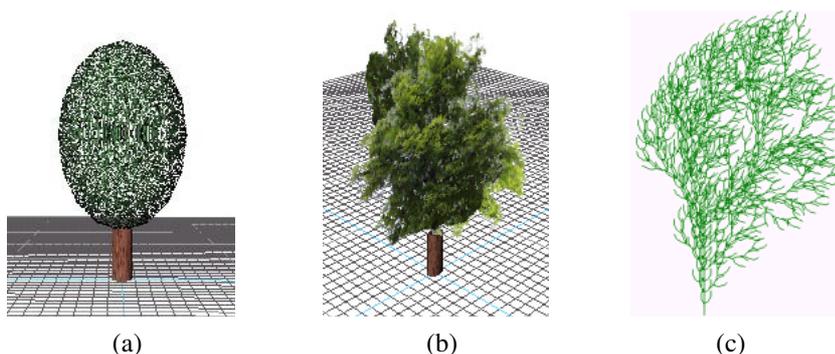


Abbildung 2.7: a) Texturiertes Kugelbaummodell  
 b) Ebenentechnik mit Textur einer Kastanie  
 c) Strauchartiges Modell eines L-Systems

Zur Texturierung von Vegetationsmodellen gibt es verschiedene Ansätze. Im einfachsten Fall werden einfache Farben verwendet, realistischer ist die Verwendung von Fototexturen. Das Integrieren transparenter Flächen in der Kronentextur, kann ein lichtdurchlässiges Erscheinungsbild erreichen. Um Varianzen und jahreszeitliche Veränderungen zu simulieren, kann Multitexturing eingesetzt werden.

Eigenschaften \ System	Kugelbaum	Ebenentechnik	L-Systeme
Optischer Eindruck	-	o	+
Erstellungsaufwand	+	+	-
Variabilität	o	o	+
Performance	+	+	-

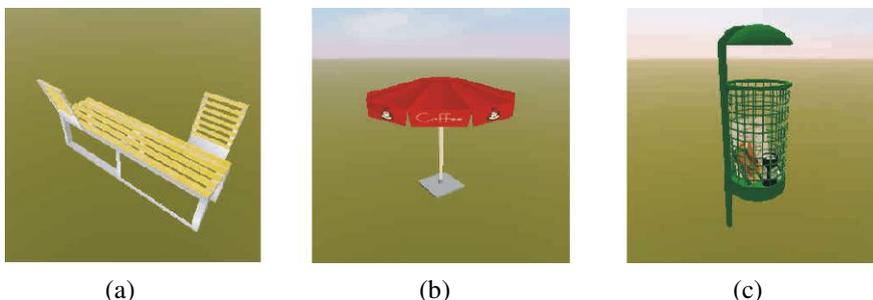
Tabelle 2.1: Überblick über die vorgestellten Verfahren zur Erstellung von Vegetationsobjekten

Legende: + gut  
o mittel  
- schlecht

#### 2.4.4. Straßenbegleitende Objekte

Straßenbegleitende Objekte bzw. Straßenmöbel, bezeichnen Objekte, wie Verkehrsschilder, Bänke, Abfalleimer, Zäune und vieles mehr. Sie dienen in 3D-Stadtmodellen, ähnlich den Vegetationsmodellen, der realistischeren Darstellung von virtuellen 3D-Szenen und erhöhen den Wiedererkennungswert.

In den meisten Fällen werden die 3D-Modelle der Straßenmöbel manuell hergestellt. Eine Fotografie eines realen Objektes dient dabei als Vorlage.



(a)

(b)

(c)

Abbildung 2.8: Beispiele für Straßenbegleitende Objekte: (a): Bank, (b) Sonnenschirm, (c) Abfallkorb (Quelle: [SSJ])

## 2.5. Texturierung von Geometriemodellen

### 2.5.1. Texturen

Der Begriff Textur wird in vielen Bereichen der Wissenschaft in verschiedenem Kontext verwendet. In dieser Arbeit wird die Textur als eine Oberflächenstruktur von Objekten verstanden. Diese Oberflächenstruktur kann visuell oder taktil wahrgenommen werden, hier wird nur die visuelle Wahrnehmung von Oberflächenstrukturen berücksichtigt.

Texturen werden in der Computergrafik eingesetzt, um virtuellen Objekten eine Struktur und damit ein realistisches Erscheinungsbild zu verleihen. Die Textur bezeichnet dabei ein n-dimensionales Bild, welches auf der Oberfläche des Objektes dargestellt wird.

In der Computergrafik erfolgt eine Unterscheidung zwischen diskreten und prozeduralen Texturen. Diskrete Texturen haben für jeden Texel<sup>5</sup> einen diskreten Wert in einer Tabelle gespeichert. Der abgespeicherte Wert kann jederzeit ausgelesen werden. Der Wert des Texels in prozeduralen Texturen berechnet sich dagegen durch die Auswertung einer mathematischen Funktion.

Je nach Möglichkeiten, Verfügbarkeit und Anwendungszweck kann die Anwendung von diskreten oder prozeduralen Texturen von Vorteil sein.

Ein Nachteil der diskreten gegenüber den prozeduralen Texturen ist die Auflösung. Bei den diskreten Texturen liegen diskrete Werte vor, so dass ab einem bestimmten Detailgrad die Diskretisierungsstufen wahrgenommen werden. Prozedurale Texturen dagegen haben keine feste Auflösung und werden folglich unabhängig von ihrer Skalierung detailliert dargestellt [Ebert et al.03].

Die Parametrisierbarkeit der prozeduralen Texturen ist ein weiterer Vorteil gegenüber den diskreten. Durch Variation der Funktionsparameter kann leicht eine große Menge an verwandten Texturen erstellt werden. Die diskreten Texturen sind hingegen auf die statischen Werte beschränkt.

Ein weiterer Unterschied ist der Speicherplatzbedarf der beiden Texturarten. Da bei prozeduralen Texturen lediglich die Funktion gespeichert werden muss, beträgt dieser wenige Kilobyte. Bei diskreten Texturen liegt dieser jedoch oft im Megabytebereich.

Eine schnelle und effiziente Auswertung des Texels ist jedoch nur bei den diskreten Texturen möglich. Die Texel dieser Texturen können einfach ausgelesen werden, während bei den prozeduralen Texturen für jedes Texel der Wert aus der mathematischen Funktion errechnet werden muss.

Darüber hinaus wirken die prozeduralen Texturen nur bei bestimmten Materialien, wie Holz, Metall oder Marmor realistisch. In den meisten Fällen ist der optische Eindruck der diskreten Texturen besser.

	Diskrete Texturen	Prozedurale Texturen
Texturauflösung	+	++
Parametrisierbarkeit	-	++
Speicherplatzbedarf	o / -	++
Effiziente Auswertung	+	-

Legende: ++ sehr gut  
+ gut  
o mittel  
- schlecht

Tabelle 2.2: Vergleich von diskreten und prozeduralen Texturen

<sup>5</sup> Texel (*texture element*)– Das kleinste Element eines Texturbildes.

Die unterschiedlichen Varianten von Texturen, wie in Abbildung 2.9 dargestellt, ihre Gewinnung bzw. Erzeugung und ihre Vor- und Nachteile bei der Nutzung in 3D-Stadtmodellen werden im Folgenden betrachtet.

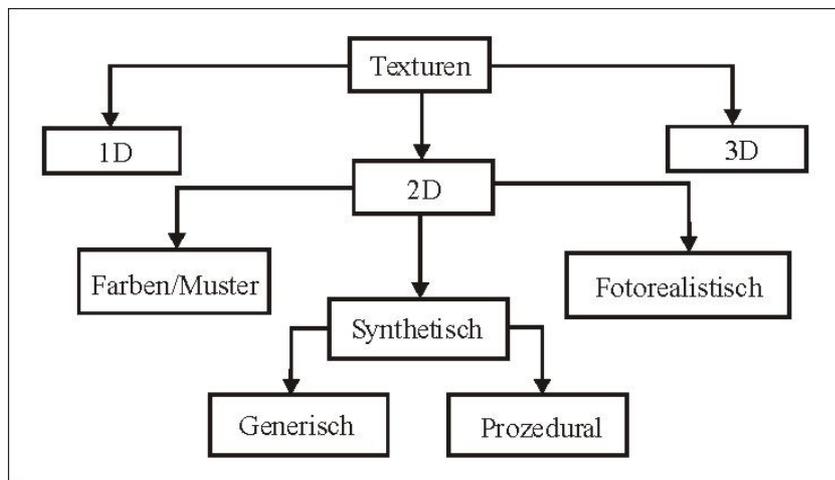


Abbildung 2.9: Texturunterteilung - Abgesehen von den fotorealistischen Texturen, lassen sich alle Texturarten prozedural oder diskret generieren.

Texturen lassen sich, wie in Abb. 2.9 zu sehen, zunächst nach ihrer Dimension einteilen [Orla]. Für die vorliegende Arbeit wird der Bereich der 2D-Texturen betrachtet. Der Vollständigkeit halber erfolgt eine kurze Ausführung zu 1D- und 3D-Texturen.

Eine **1D-Textur** ist ein  $1 \times n$ -Feld, sie wird in der Computergrafik häufig als LookUp-Tabelle<sup>6</sup> verwendet. Beispielsweise zur Darstellung von Höhenunterschieden oder zur Abschwächung der Intensität von Lichtquellen.

Im Bereich der **2D Texturen** können einfache Farben und Muster, synthetische Texturen sowie Fotografien Verwendung finden. Sie können mittels *texture mapping*, *reflection mapping* und *environment mapping* auf eine Oberfläche projiziert werden und erzeugen so eine glatte realistische Oberfläche. Soll die Oberfläche zusätzlich eine reliefartige Struktur erhalten, können mit einer *bump map* die Oberflächennormalen geändert werden. Genauere Informationen zu diesen Verfahren sind in fachbezogenen Quellen, z.B. [FVD05] nachzuschlagen.

**Farben und Muster** sind geeignet, wenn die Priorität nicht auf einer fotorealistischen Darstellung der Szene, sondern im Bereich der Datenanalyse und -simulation liegt. Genaueres hierzu im Abschnitt 2.5.3.

**Synthetische Texturen** erhöhen die realistische Wirkung gegenüber Farben erheblich, können die künstliche Erstellung jedoch nicht verbergen, vgl. Abbildungen 2.10 und 2.11. Diese Texturen können zum einen in Bildbearbeitungsprogrammen oder mit Hilfe der Textursynthese hergestellt werden und untergliedern sich entsprechend in generische und prozedurale Texturen. Unter Textursynthese wird hier das Berechnen von Texturen mittels mathematischen Funktionen verstanden.

<sup>6</sup> LookUp-Tabelle – In der Informatik ein auf einer Tabelle beruhendes Berechnungsverfahren.

Inzwischen gibt es große Bibliotheken, die ein weites Spektrum an verschiedenen Materialien und Objekten abdecken.

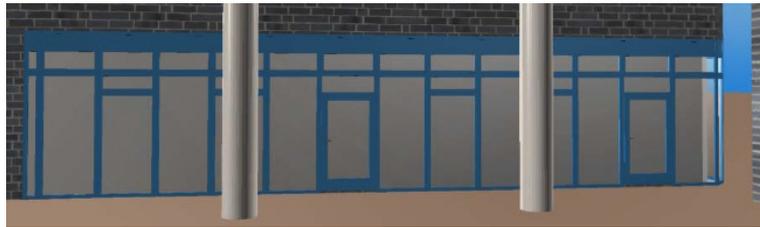


Abbildung 2.10: Fassadengestaltung mit synthetischen Texturen (Quelle: [Guc05])



Abbildung 2.11: Fassadengestaltung mit Fototexturen (Quelle: [Guc05])

Texturen für Gebäudeobjekte können Einzelobjekte oder Aggregate sein. Einzelobjekte sind große Einzeloberflächenelemente, z.B. Fenster oder Türen. Aggregate sind dagegen Fassaden- oder Dachstrukturen und Materialmuster, wie Backsteinfassaden, Dachziegel und Holzstrukturen.

Die Aggregate können weiter unterschieden werden in regelmäßige oder deterministische Texturen und statistische oder stochastische Texturen. Erstere zeigen ausgeprägte Strukturen, die sich mit einer geometrischen Regelmäßigkeit wiederholen (z.B. Wegpflasterung). Letztere Texturart zeigt keinerlei regelmäßige Struktur, sie erinnert vielmehr an Bildrauschen (z.B. eine Lederoberfläche).

**Fototexturen** erhöhen nicht nur die Realitätswirkung, sondern können Modellierungsaufwand ersparen, in dem diese z.B. für Gebäudefassaden genutzt werden. Sollen die Geometrien jedoch auch aus der Nähe und aus verschiedenen Winkeln betrachtet werden, bleibt eine Modellierung von einigen Details unumgänglich. Als Beispiel kann hier ein Balkon dienen. Dieser wird nicht modelliert, sondern mittels einer Textur auf einer ebenen Fläche abgebildet. So erhält der „texturierte Klotz“ im Vergleich zur Verwendung einer generischen Fassadentextur eine realistischere Darstellung. Von der Nähe bzw. der Seite aus betrachtet, wird das Fehlen der Balkongeometrie jedoch sichtbar. Dieses Beispiel zeigt, dass in bestimmten Fällen die Modellierung von Details nicht durch die Texturierung ersetzt werden kann.

**3D-Texturen** sind eine Erweiterung der 2D-Texturen. Sie können als „dünne Scheiben“ von Texturen aufgefasst werden, die genutzt werden, um eine dreidimensionale Bild-*map* zu erstellen. Angewandt werden diese beispielsweise im Bereich der Visualisierung von medizinischen Volumendaten.

### 2.5.2. Gewinnung von Texturen

Abhängig von der Nutzung der Textur, können zwei Varianten unterschieden werden. Die Objekttextur und die Oberflächentextur [Sch05].

Objekttexturen stellen komplette Teile eines Objektes (z.B. eine Tür oder eine Fassade eines Hauses) dar und werden gewöhnlich aus Fotografien (frontal oder Luftbild) gewonnen. Sie können auch durch *rendering* einer virtuellen Szene oder von Einzelobjekten entstehen, wie z.B. Vegetationstexturen aus L-Systemen (s. Abschn. 2.4.3). In dem Bereich der Stadtvisualisierung finden Objekttexturen hauptsächlich bei der Gestaltung von Gebäuden Verwendung.

Oberflächentexturen sind Ausschnitte von Materialoberflächen. Diese Texturen müssen kachelbar sein, d.h. „übergangslos in u und v Richtung periodisch aneinandergereiht“ (Schulze, 2005, [Sch05]) werden können. Oberflächentexturen werden vorwiegend für die Texturierung von Grund- und Bodengeometrien, aber auch bei der Gestaltung von Gebäuden eingesetzt.

In den folgenden Abschnitten soll zunächst die Gewinnung von Texturen aus Fotografien, als Beispiel für die diskreten Texturen, betrachtet werden. Anschließend folgen Verfahren zur Gewinnung von prozeduralen Texturen, als Beispiel der Textursynthese.

#### 2.5.2.1. Gewinnung von Texturen aus Fotografien

Um Fotografien als Quelle von Texturen nutzen zu können, müssen diese unter Beachtung folgender Bedingungen aufgenommen werden:

- Die zu fotografierende Oberfläche sollte einen möglichst gleichmäßigen Licht-, Kontrast- oder Sättigungsverlauf (z.B. keine Schatten) aufweisen.
- Es sollten möglichst gleichmäßige Strukturen vorhanden sein, partiell auffällige Elemente können zu übergeordneten Mustern führen (z.B. Astlöcher in Holzstrukturen).
- Der Fotograf sollte möglichst orthogonal zur Objektebene stehen.
- Der Blick auf das Objekt sollte frei sein, d.h. es sollten möglichst keine anderen Objekte vorhanden sein, durch die das eigentliche Objekt verdeckt würde (z.B. Bäume und Autos bei Gebäudefassaden).
- Der Objektanteil an der Aufnahme darf nicht die gesamte Fotografie einnehmen, um Randverzerrungen zu vermeiden.

Da diese Bedingungen in der Realität oft nicht eingehalten werden können, müssen die Fotografien in der Regel manuell nachbearbeitet werden. Zu den häufigsten Arbeitsschritten zählen das Freistellen bzw. Ausschneiden, die Retusche, die Skalierung, das Anpassen von Farbe und Kontrast sowie die Festlegung von transparenten Gebieten [Sch05]. Für diese Arbeitsschritte können Bildbearbeitungsprogramme genutzt werden, mit denen sich die genannten Arbeitsschritte recht einfach umsetzen lassen.

## Objektoberflächenaufnahmen

Bei dem Erstellen von Texturen, die wiederholt aneinander gelegt werden sollen, ist darauf zu achten, dass die Bilder an allen Seiten gut aneinander passen. Sonst können Bruchkanten und andere Artefakte entstehen, die beim Betrachten negativ auffallen.

Dazu ein Beispiel anhand einer Fotografie eines Straßenpflasters aus der Innenstadt der Lutherstadt Wittenberg. Das Ausgangsbild wurde zunächst entzerrt und anschließend auf einen Bereich zugeschnitten. Da es im Ausgangsbild einen Helligkeitsunterschied von rechts nach links gibt, wurde der Ausschnitt horizontal gespiegelt und aneinander gefügt (s. Abb. 2.12 und 2.13). Somit wird ein harter farblicher Übergang zwischen den Bildelementen vermieden.



Abbildung 2.12: Ungünstiger Ausschnitt (a), da die Bruchstücke unten links wiederholt auf der texturierten Fläche vorkommen und ein Texturstein entsteht, der in seinen Ausmaßen nicht zum Rest des Bildes passt (b).



Abbildung 2.13: Dieser Ausschnitt (a) ist besser geeignet, da die Texturelemente gut aneinander passen und keine auffälligen Elemente enthalten sind (b).

## Objektaufnahmen

Bei der Aufnahme von Objekt fotografien ist ebenfalls darauf zu achten, dass die gesamte Objektoberfläche gleichmäßig diffus ausgeleuchtet ist. Die sonst entstehenden Schlagschatten wirken in virtuellen Umgebungen unrealistisch, da sie nicht zwangsläufig, der aktuellen Lichtsituation entsprechen.

Neben der Retusche mittels „Stempelfunktion“ von Bildbearbeitungsprogrammen, gibt es die folgende Möglichkeit, bewegliche störende Elemente aus Bildern zu entfernen.

Für das Verfahren nach Stoske und Bertling [StBe] müssen zunächst viele Aufnahmen von der gewünschten Szene oder des Objektes gemacht werden. Dabei ist es wichtig, dass der Standort gleich bleibt. Anschließend wird für das neue Bild von jedem Pixel der Mittelwert über alle Bilder errechnet und dargestellt.

In dem Ergebnisbild der beschriebenen Methode sind die beweglichen Elemente nur teilweise entfernt. An Orten, an denen sich häufig bewegte Objekte aufhalten, bleibt immer ein entsprechender Anteil zurück. Somit sind Rückstände und Wischeffekte auch mit einer sehr großen Bildanzahl nicht zu vermeiden.

Ein besseres Ergebnis kann erzielt werden, in dem die Extreme vor der Mittelwertberechnung entfernt werden. Diese Methode entspricht einer statistischen Auswertung und erfordert eine wesentlich geringere Anzahl an Bildern. Allerdings funktioniert das Verfahren nur, wenn die unbewegten Flächen in mindestens zwei Bildern existieren.

Das gleiche Konzept verfolgt der „Tourist Remover“ [fLAB08] der futureLAB AG (s. Abb. 2.14).

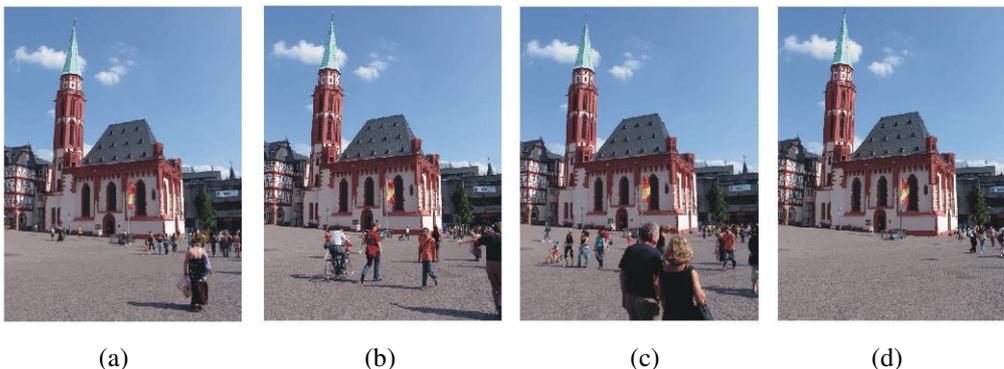


Abbildung 2.14: Auszug aus einer Bilderreihe einer Kirche - (a) bis (c) sind Ausgangsaufnahmen, (d) ist das Ergebnisbild (Quelle: [fLAB08])

Die Methode der statistischen Auswertung ist somit gut geeignet, um bewegliche störende Objekte aus Fotografien zu entfernen. Je nach Anzahl der Objekte muss unter Umständen eine größere Anzahl von Bildern aufgenommen werden. Jedoch können auch hiermit statische Objekte wie Bäume nicht entfernt werden. Bei diesen Objekten muss weiterhin auf Bildbearbeitungsprogramme zurückgegriffen werden.

**Luftbilder** werden aus Überfliegungen gewonnen. Zu ihnen zählen neben den „einfachen“ Luftbildern, Orthofotos und Schrägbilder (vgl. Abschn. 2.3.3.) Die Zuordnung zu den entsprechenden Geometrien kann bei vorhandener geografischer Referenzierung der Bilder weitgehend automatisch erfolgen.

Aus Orthofotos können Dach- und Straßentexturen extrahiert werden.

Schrägbilder eignen sich, um Gebäudetexturen zu erhalten. Für eine Anwendung in 3D-Stadtmodellen, die vorwiegend aus der Fußgängerperspektive betrachtet werden, eignen sich Schrägbilder weniger. Es wird eine Verzerrung wahrgenommen, da die Bilder im Gegensatz zur Blickrichtung (von unten) aus der Luft aufgenommen wurden. Sollen ebenfalls Dach und Straßentexturen aus Schrägbildern gewonnen werden, kann es Probleme durch Verdeckungen und Verzerrungen geben.

Grundsätzlich wäre es sinnvoll das Verfahren zur Entfernung von beweglichen Objekten für Luftbilder zu verwenden, um beispielsweise störende Fahrzeuge aus den Bildern zu entfernen. Der Vorteil ist, dass bei der Nutzung der bearbeiteten

Bilder als Straßentextur keine „platt gedrückten“ Fahrzeuge auf den Straßen zu sehen sind. Es ist jedoch fraglich, ob der Aufwand und die Kosten den Nutzen rechtfertigen. Da mehrere Aufnahmen aufgenommen werden müssen, sind viele Überfliegungen notwendig. Anschließend müssen in den einzelnen Fotos übereinstimmende Bereiche gesucht werden, die vom selben Blickpunkt aufgenommen wurden. Erst dann kann der Algorithmus auf die Bilder angewandt werden.

### 2.5.2.2. Gewinnung von Texturen mittels Textursynthese

Die Textursynthese ist die automatische Erzeugung von Texturen, welche Oberflächeneigenschaften bzw. Oberflächenstrukturen aufweisen und möglichst nicht von echten Texturen zu unterscheiden sind.

Prozedurale Verfahren generieren die Textur durch mathematische Berechnungen. Der Benutzer wählt eine Reihe von numerischen Parametern, die an mathematische Funktionen oder Algorithmen übergeben werden.

Nachfolgend ein Beispiel für eine solche Funktion, die ein Streifenmuster liefert (s. Abb. 2.15):

```
if (sin( $\pi$ *p.x/w)>0) then return c0;           (2.1)
else return c1;
[Shi03]
```

Die Konstanten  $c_0$ ,  $c_1$  sind die Farben Schwarz und Weiß,  $p.x$  der X-Wert des aktuellen Punktes und  $w$  ein Parameter zur Skalierung der Streifenbreite.



Abbildung 2.15: Streifenmuster, das durch die Funktion (2.1) in Abhängigkeit von  $w$  erzeugt wird

Ein weiteres Beispiel für prozedurale Texturen sind die Vegetationstexturen, welche aus den in Abschnitt 2.4.3. vorgestellten L-Systemen generiert werden können.

### Softwarebeispiele zur Erzeugung prozeduraler Texturen:

**Werkzeug3 TE** von „theprodukt“ [<http://www.werkkzeug.com>] ist ein Tool zur einfachen Erzeugung von prozeduralen Texturen. Designern soll damit ermöglicht werden, natürlich wirkende Oberflächen ohne Kenntnis der korrekten mathematischen Formeln einfach zusammen zustellen.

**MaPZone 2.6** ist ein *procedural texture editor* [<http://www.mapzoneeditor.com>] der Firma Allegorithmic. Er basiert auf einem so genannten *compositing graph*, welcher den wichtigsten Teil dieses Werkzeugs bildet. In ihm können Filter und FX-Maps<sup>7</sup> definiert und so miteinander kombiniert werden, dass die gewünschte Textur entsteht.

### Vergleich der Texturierung mit Fotografien und synthetischen Materialien

Ein spezieller Vorteil der prozeduralen Textursynthese ist, dass nicht die eigentliche Textur, sondern lediglich das Verfahren gespeichert werden muss. Dies spart in der Regel sehr viel Speicherplatz. Zudem können auf diese Weise schnell große Texturen erzeugt werden. Dies ist ein Vorteil für Anwendungen, in denen schnelle Reaktionszeiten gefordert sind. Zu diesen Anwendungen zählen neben den 3D-Computerspielen auch die virtuellen 3D-Stadtmodelle.

Der Erstellungsaufwand von prozeduralen Texturen kann je nach Methode und Zweck sehr hoch werden, mit modernen Tools aber auch gering bleiben. Der Bearbeitungsaufwand von Fotografien variiert ebenfalls.

Die Texturauflösung hängt bei Fotografien von den Fähigkeiten des Fotografen und der Auflösung der Kamera ab. Texturen aus der synthetischen Herstellung sind auflösungsunabhängig.

Der optische Eindruck von prozeduralen Texturen wird stetig besser. Die Ergebnisse tasten sich immer weiter an die realistischen Vorlagen heran. Bei der Texturierung von Gebäudefassaden zur realistischen Darstellung können sie die Fotografien unter gleichen Qualitätsvoraussetzungen jedoch nicht übertreffen. Letztlich muss der Nutzer entscheiden, welche Textur im gegebenen Kontext verwendet werden soll.

	Fotografie	Synthetische Herstellung
Speicherplatzbedarf	o / -	+
Erstellungsaufwand	+ / -	+ / -
Texturauflösung	+	++
Optischer Gesamteindruck	+	o

Legende:    +    gut  
                   o    mittel  
                   -    schlecht

Tabelle 2.3: Vergleich von fotografischen und synthetischen Texturen

<sup>7</sup> FX-Map - Basis der Prozeduralen Texturen in MaPZone

### 2.5.3. Texturierung zur Visualisierung von Daten in 3D-Stadtmodellen

Im Bereich der Stadtmodelle kann die Farbgebung bzw. Texturierung von Objekten neben der realitätsgetreuen Darstellung ebenfalls zur Visualisierung oder Analyse (*data-mining*-Verfahren) von Daten genutzt werden. Im folgenden Abschnitt wird das Thema der Visualisierung vom Standpunkt der Stadtvisualisierung kurz angerissen. Die Grundlageninformationen entstammen dem Buch zur Visualisierung von Heidrun Schumann [Sch00]



Abbildung 2.16: *data mining* im LandXplorer am Beispiel von Kundenverhalten codiert durch Farben (Quelle: [3DGLX])

Die generellen Ziele in der Visualisierung bestehen in explorativen und bestätigenden Analysen, der Ergebnisdarstellung sowie der Präsentation als Kommunikationsgrundlage und zur Entscheidungsfindung. In Stadtmodellen ist beispielsweise eine explorative Analyse über den Leerstand von Gebäuden denkbar. Wird der Leerstand nacheinander für mehrere Zeiträume dargestellt, kann so die Veränderung aufgezeigt und ein Trend abgelesen werden.

Der Visualisierungsprozess beinhaltet vier Stationen, um aus den gegebenen Daten eine Visualisierung zu erhalten (s. Abb. 2.17).

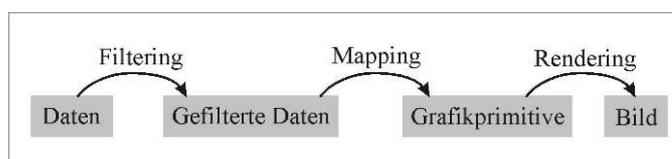


Abbildung 2.17: Visualisierungspipeline von den Daten bis zum Bild

Zunächst werden die zu visualisierenden Daten ausgewählt, analysiert und entsprechend der Aufgabenstellung bearbeitet. Die so gefilterten Daten werden auf Grafikprimitive gemappt. Beim *Mapping* erfolgt in der Regel die Auswahl, Platzierung, Skalierung und Parametrisierung von Grafikprimitiven, somit die Bestimmung der Geometrie sowie deren Parameter und Materialeigenschaften. Die Auswahl der *Mapping*-Funktion erfolgt entsprechend der Klassifizierung der Ausgangsdaten. Letztlich wird durch das *Rendering* mit entsprechenden Einstellungen das Bild der Visualisierung erzeugt. In 3D-Stadtmodellen erfolgt das *Rendering* der Szene in Echtzeit.

Im Nachfolgenden soll auf den Bereich des *Mapping* im Sinne der oben gezeigten Abbildung 2.17, bezogen auf digitale Stadtmodelle, näher eingegangen werden.

Für eine Analyse oder Präsentation in einem virtuellen 3D-Stadtmodell müssen die Daten aus den GI-Systemen eingepflegt werden. Die Auswahl und Vorverarbeitung, das *Filtering*, übernimmt in der Regel der Anwender. Das Visualisierungsverfahren zum *Mapping* der Daten sollte der Benutzer selbst wählen können. Idealerweise erfolgt vom System ein Vorschlag. Diesen kann der Anwender annehmen oder die Eigenschaften des vorgeschlagenen *Mapping*-Verfahrens ändern.

Mögliche Daten, welche in einem Stadtmodell zu visualisieren sind:

- Leerstand von Gebäuden
- Nutzungsarten von Gebäuden, z.B. kulturelle Einrichtungen - Schulen
- Nutzungsarten von Flächen, z.B. Wohnraum - Gewerbeflächen - Grünflächen
- Eigentümer, z.B. gewerblich – öffentlich – privat
- Unterscheidung zwischen Neu-/Altbauten
- Unterscheidung zwischen renoviert und unrenoviert (Wohnungsgenossenschaften)
- Standorte von Objekten, wie z.B. Funkmasten

Der Leerstand und die Nutzungsart von Gebäuden oder Flächen sind als nominale Daten zu klassifizieren, wobei Informationen über die Baujahre von Gebäuden zu den ordinalen Daten zählen.

Ordinale Daten definieren sich darüber, dass es eine Ordnung zwischen den Daten gibt, diese Ordnung aber nicht quantitativ ist. Eine Zuordnung von Zahlen ist eine Interpretation. Ordinale Daten sollen einen qualitativen Vergleich ermöglichen, welcher gut über Positionen, Helligkeit und Sättigung realisiert werden kann. So können ältere Baujahre beispielsweise über einen ungesättigten hellen Farbton und jüngere Baujahre über den gleichen Farbton veranschaulicht werden, der gesättigter und dunkler ist. Ordinale Daten stellen außerdem Anforderung an die Farbwahl. Empfundene Farbwerte ( $E(c)$ ) müssen die Ordnung der Daten widerspiegeln. So muss bei der Abbildung der Datenwerte  $x_1 < x_2 < \dots < x_n$  auf die Farbwerte  $c_1, c_2, \dots, c_n$  gelten:  $E(c_1) < E(c_2) < \dots < E(c_n)$ .

Nominale Daten zeichnen sich durch das Nichtexistieren einer Rangfolge unter den Daten aus. Die Darstellung nominaler Daten will eine Kategorisierung der Daten ermöglichen. Dazu können Präsentationsvariablen wie Position, Farbton und Textur genutzt werden. Bei dem Gebrauch von Farben zur Visualisierung nominaler Daten ist neben einem möglichst große Abstand zwischen den Farben auch auf eine Mindestgröße der Farbdarstellung (Unterscheidung von Farben: 16', Farbidentifikation: 20') zu achten.



Abbildung 2.18: Kennzeichnung von Neubauten über eine rote Einfärbung (Quelle: [SMB])

Die Abbildung auf Farbe kann nach Schumann bei folgenden Einsatzmöglichkeiten Verwendung finden:

- der Unterscheidung von Regionen oder Elementen
- der Betonung von Regionen oder Elementen
- dem Vergleich ordinaler und nominaler Daten

Grundsätzlich sind bei der Verwendung von Farben in der Visualisierung unter anderem folgende Grundprinzipien zu beachten:

- Minimierung der Anzahl der Farben
- Minimierung der Anzahl der stark gesättigten Farben
- Nutzung von Komplementärfarben für die Klassifikation
- Auffällige Farbveränderung für kritische Bereiche

Probleme können bei Nichtbeachtung von so genannten „vorbelegten“ Farben und Farbanomalien auftreten.

Die Abbildung von Daten auf eine Textur ist eine weitere Möglichkeit, die primär für nominale Daten Verwendung findet. Hierbei werden unterschiedliche Schraffuren eingesetzt. Für ordinale Daten nutzt man Variationen der Grundschräffur, wie z.B. die Änderung der Dichte der Schraffurlinien. Es ist jedoch darauf zu achten, dass Texturen einer größeren Ausdehnung als Farben bedürfen, um gut unterscheidbar zu sein.

#### 2.5.4. Texturierungsmethoden

Das Projizieren der Textur auf das 3D-Objekt erfolgt durch das *texture mapping*. Oberflächeneigenschaften, wie Spiegelung und Unebenheiten können durch *environment mapping* und *bump mapping* simuliert werden. Da die letzteren beiden Verfahren nicht in der Stadtvisualisierung eingesetzt werden, wird an der Stelle auf entsprechende Fachliteratur, z.B. [FVD05], verwiesen.

Der Begriff *texture mapping* (dt. Musterabbildung) bezeichnet die Zuordnung eines zweidimensionalen Bildes (= Textur) zu der Oberfläche eines dreidimensionalen Objektes. Durch Texturen können computergenerierte Objekte bei gleich bleibender Polygonzahl realistischer und detailgetreuer erscheinen.

Die *texture mapping*-Vorschrift besteht nach Englert [Eng92] aus folgenden Teilen:

- dem verwendeten *mapping*-Verfahren
- der Orientierung der Texturmatrix auf der Objektoberfläche und
- dem Skalierungsfaktor der Texturmatrix bei der Abbildung auf das Objekt.

Das *mapping*-Verfahren gibt an, mit welcher Methode das zweidimensionale Bild auf das Objekt aufgebracht wird. Die Orientierung und Skalierung sind dabei meist Parameter dieser Methode. Die Methode gibt an, welchem Punkt  $(x,y,z)$  auf der Oberfläche des Objektes welcher Wert der Textur zugewiesen wird, wobei  $(u,v)$  die Texturpunkte (Texel) im Texturraum sind.

$$(u,v) = F(x,y,z) \quad (2.2)$$



Abbildung 2.19: Grafik zum Prozess der Texturabbildung nach [FVD05]

Bei einfachen 3D-Objekten kann die Projektion der Textur planar erfolgen, komplexe 3D-Objekte werden nach dem so genannten *two-stage-mapping* [BS86] texturiert. In der Stadtvisualisierung werden hauptsächlich ebene Flächen (z.B. Gebäudewände, Straßen) mit Texturen versehen, deshalb erfolgt die Projektion meist planar.

Das *Multitexturing* (dt. Mehrfachtexturierung) kann genutzt werden, um Texturen zu überblenden und zu addieren. *Multitexturing* bezeichnet das Aufbringen von zwei oder mehr Texturen auf ein Primitiv.

Seit einigen Jahren ist dieses Verfahren ein fester Bestandteil der Spielebranche, es wird hier vorwiegend für *light*- und *detailmaps* genutzt. Bei der *lightmap* werden Schattenwurf und Lichteinfall auf ein Polygon vorberechnet und anschließend in einer separaten Textur gespeichert. Die *detailmap* enthält ein Muster, das über die Basistextur gelegt wird und so kleine Details simulieren soll. Im Bereich der Stadtmodelle kann die Mehrfachtexturierung ebenfalls Anwendung finden (vgl. Abschn. 2.4.3 der vorliegenden Arbeit).

### 2.5.5. Orientierung von geometrischen Objekten

Die *bounding box* gehört zu den sog. *bounding volumes*. Diese Volumen sind einfache geschlossene geometrische Körper, welche eine oder mehrere meist komplexe geometrische Formen umhüllen. Sie werden eingesetzt, um die Effizienz geometrischer Operationen, z.B. Kollisionstests, zu beschleunigen bzw. zu vereinfachen. Typische Formen im zweidimensionalen Raum sind Kreise und Rechtecke.

Die Box, üblicherweise *bounding box* genannt, kann in sog. *axis-aligned bounding box* (AABB) und *object-orientated bounding box* (OOBB) unterschieden werden. Eine AABB ist entlang der Achsen des Koordinatensystems ausgerichtet, während die OOBB sich der Form des Objektes optimal anpasst.

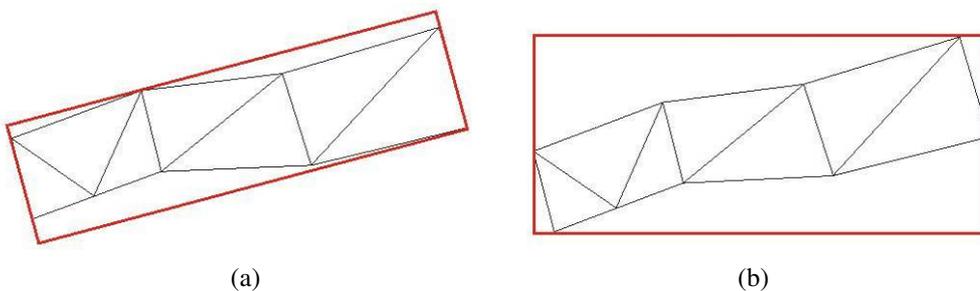
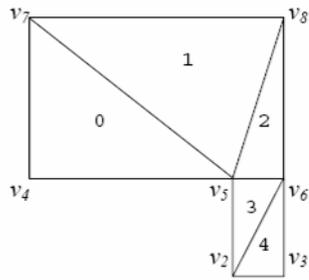


Abbildung 2.20: Objekt mit (a) objektorientierter *bounding box* (rot) und (b) achsenorientierter *bounding box* (rot)

Neben den Kollisionstests ergibt sich ein weiterer Verwendungszweck der *bounding box* in der automatischen Texturierung von Straßen. Eine *bounding box* enthält die Ausrichtung des Objektes, im Folgenden Objektausrichtung genannt, welche zur korrekten Ausrichtung der Textur genutzt werden kann. Die AABB gibt bei schief in der Ebene liegenden Objekten lediglich eine ungefähre Orientierung des Objektes. Eine präzise Orientierung liefert hingegen die OOBB (vgl. Abb.2.20).

Eine andere Variante ist die Nutzung der **Skelettierung** eines Objektes. Die Skelettierung von Geometrieobjekten kann durch verschiedene Methoden erreicht werden (s. [Bru07]). Das Ergebnis ist eine Skelettlinie, die entlang der Mitte des Objektes verläuft (s. Abb. 2.21).



(a) Original mesh

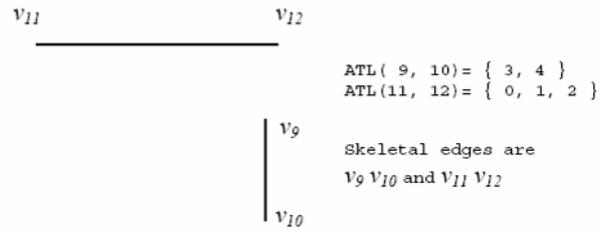
(e) After contracting edge  $(v_8, v_9)$  to vertex  $v_{12}$ 

Abbildung 2.21: Skelettierung eines Polygons (Quelle: [XTTZ01])

In (a) ist das Originalpolygon und in (b) die entstandenen Skelettkanten dargestellt.

### 3. Stand der Technik

Dieses Kapitel gibt zunächst einen Überblick über Nutzungspotentiale und Arbeitsabläufe zur Erstellung digitaler Stadtmodelle. Es folgt eine Recherche zu vorhandenen Daten in ausgewählten Kommunen. Anschließend werden einige gängige Visualisierungsprogramme mit Geobezug sowie existierende 3D-Stadtmodelle vorgestellt. Im Anschluss daran werden innovative fotobasierte Techniken vorgestellt. Abschließend erfolgt eine Zusammenfassung der betrachteten Themen.

#### 3.1. Nutzungspotentiale von 3D-Stadtmodellen

Die Realisierung von 3D-Stadtmodellen verursacht Kosten (s. Abschn. 3.2. der vorliegenden Arbeit), die sich nur amortisieren, wenn sich das Modell für mehrere Anwendungszwecke nutzen lässt und so einen Mehrwert schafft.

Anwendungsgebiete und somit Nutzergruppen für digitale 3D-Stadtmodelle liegen beispielsweise in der Stadtplanung, der Tourismuswirtschaft, dem Standortmarketing, der (Stadt-) Verwaltung sowie der Spiele- und Medienwelt (vgl. [ABH03]).

Mit den Mitteln der Virtual Reality (VR) lassen sich städtebauliche Planungsmaßnahmen realistisch darstellen, ohne dass beim Betrachter berufsspezifische Kenntnisse vorausgesetzt werden müssen. Die Entwurfsdiskussion im virtuellen Raum eröffnet neue Wege in der Entscheidungsfindung im Vergleich zur zweidimensionalen bzw. analogen Darstellung.

Unterstützend für diese Form der Arbeit mit digitalen Stadtmodellen sind Methoden wie Messfunktionen, das Anzeigen von Sachdaten, sowie die Nutzung interaktiver Darstellungsfunktionen.

In der Tourismuswirtschaft wird durch 3D-Stadtmodelle eine realistische Darstellung der touristischen Angebote ermöglicht und die Nutzung der Breitenwirkung von Internet-Technologien eröffnet. Dies hat einen Marketingeffekt für die Stadt. Es können zudem gewerbliche und historische Informationen eingebunden werden. Auch die Suche eines bestimmten Ortes mit anschließendem „Flug“ zum Zielort (wie man es bereits aus diversen Suchmaschinen kennt) ist denkbar. Weiterhin können Rundflüge erstellt und zu Demonstrations- und Werbezwecken abgespielt werden.

Eine aufgabenbezogene digitale Darstellung kommt ebenfalls dem Standortmarketing zu Gute. Das Standortmarketing legt besonderen Wert darauf, potentielle Investoren ohne Ortskenntnis von der Qualität eines Standortes zu überzeugen. Ein digitales 3D-Stadtmodell ist in der Lage, die entscheidenden Informationen in kurzer Zeit und unabhängig von der Position des Investors klar

zu vermitteln. In diesem Fall besteht die Notwendigkeit, hochwertige und effektive Präsentationsmittel (z.B. ein interaktives 3D-Modell, mit Zusatzinformationen angereichert) zu nutzen. Hierbei liegt das Augenmerk weniger auf einer realistischen Darstellung, als auf einer guten und flexiblen Visualisierung der angepriesenen Standorteigenschaften.

Auch für die Stadtverwaltung können 3D-Stadtmodelle nützlich sein. So können virtuelle Stadtmodelle in der Stadtdokumentation eingesetzt werden, indem sie nicht mehr vorhandene Stadtansichten verschiedener Epochen sichtbar machen.

Weiterhin ist es möglich 3D-Stadtmodelle zu Analysen (z.B. Sichtbarkeitsanalysen) und physikalischen Simulationen einzusetzen, beispielsweise in der Funknetzplanung, Überflutungssimulationen (s. Abb. 3.1b), Simulation von Schallausbreitung oder Luftbewegungen (s. Abb. 3.1a).

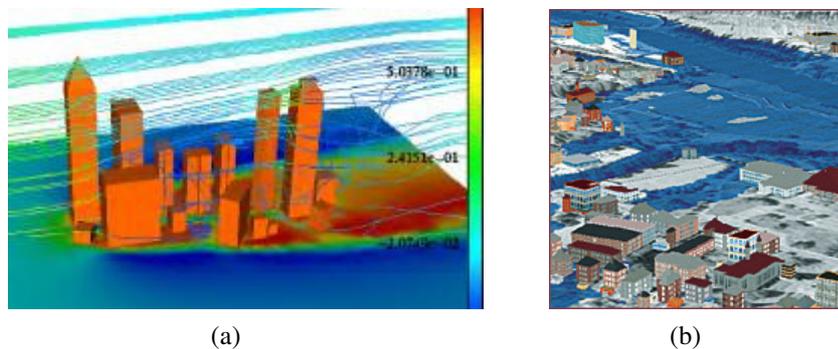


Abbildung 3.1: (a) Luftdruck und Partikelpfade einer Luftströmungssimulation (Quelle: [Lan04])  
(b) Überflutungssimulation (Quelle: [CEDIM])

Auch im Bereich der Medien, Werbung und Spiele werden heute 3D-Modelle eingesetzt. Hier können nach Belieben neue fantasievolle Umgebungen erschaffen, oder bekannte Orte nachgebaut werden, z.B. um große Zerstörungen darzustellen.

### 3.2. Arbeitsablauf zur Erstellung eines 3D-Stadtmodells

Die Aufteilung der nachfolgenden Arbeitsschritte, ihre Reihenfolge sowie die angegebenen Aufwandsabschätzungen gehen aus Erfahrungswerten hervor, die in Modellierungsprojekten des Fraunhofer-Instituts Magdeburg gewonnen wurden. Die Spanne der Prozentzahlen ergibt sich aus unterschiedlichen Projektvoraussetzungen und -zielen.

#### 1. Daten- und Grundlagenermittlung (10 - 25%)

Die Daten- und Grundlagenermittlung beinhaltet das Sammeln und Auswerten aller erforderlichen Ausgangsdaten zur Generierung digitaler 3D-Objekte (z.B. Grundrisse, Fassadenansichten, Straßenpläne, Texturen) des Stadtmodells. Diese Daten werden von den Kommunen bereitgestellt. Gelegentlich sind nachträgliche Messungen und Datenerfassungen erforderlich, um einen vollständigen Datenbestand zu erhalten.

## **2. Generierung von 3D-Modellen (50 - 80%)**

Aus den gesammelten zweidimensionalen Daten müssen mit Hilfe von Konstruktions- und Modellierungswerkzeugen (z.B. CAD-Programme, automatisierte Tools) virtuelle Modelle generiert und texturiert werden. Anschließend werden diese in der Modellwelt zusammengeführt.

## **3. Zusatzinformationen (5 - 10%)**

Die Anreicherung mit zusätzlichen Informationen entsprechend dem Anwendungszweck ist in einem weiteren Schritt zu vollziehen. Diese Informationen können unter anderem Sachdaten zu Gebäuden, Eigentümerverzeichnisse oder touristische Merkmale sein.

## **4. Interaktive Funktionen (5 - 20%)**

Die Bereitstellung interaktiver Funktionen und Systemfeatures sowie die Implementierung einer benutzerfreundlichen Bedienführung folgen als finaler Schritt. Der genaue Aufwand ist von dem Anwendungszweck und dem Umfang der vorhandenen Daten abhängig.

Wie aus der Darstellung in Abb. 3.2a zu entnehmen ist, beansprucht die Generierung der 3D-Modelle (Punkt 2) den größten Aufwand. Dazu gehören die Modellierung und die Texturierung. Für die automatische Modellierung und Texturierung von Gebäuden gibt es bereits gute Ansätze, um den Aufwand zu verringern. Die Beschleunigung der Modellierung und ansprechende Texturierung von Straßen und Gehwegen wurde hingegen bislang selten betrachtet.

Werden Techniken zur automatischen Modellierung von Gebäuden mit einer semi-automatischen Texturierung von Bodengeometrien kombiniert, ändert sich der Arbeitsaufwand wie folgt:

### **1. Daten- und Grundlagenermittlung (5 - 20%)**

Die Ausgangsdaten können direkt aus dem Geoinformationssystem ausgelesen und über verschiedene Verfahren zur Datengewinnung, z.B. aus Fotografien ergänzt werden.

### **2. Generierung von 3D-Modellen (30 - 60%)**

Die manuelle 3D-Modellierung wird auf einzelne komplexe Objekte reduziert, womit der Zeitaufwand erheblich verringert wird. Das Zusammenfügen der 3D-Objekte entfällt, da diese direkt in der Modellwelt an der korrekten Position erschaffen werden. Auf diese Weise wird zusätzlich das manuelle Platzieren der selbigen am korrekten Ort überflüssig. Das Modell ist somit konsistent zu den bestehenden 2D-Daten der Kommunen.

Die Texturierung kann ebenfalls bereits durch Objektreferenzierung und geografische Referenzierung der Textur erfolgt sein. Sie muss nur noch in besonderen Bereichen, z.B. komplexe Geometrien, wie Krümmungen im Straßenverlauf, manuell nachgebessert werden.

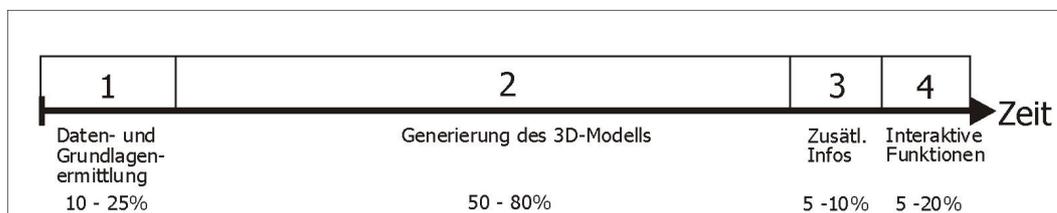
### 3. Zusatzinformationen (5 - 10%)

Zusätzliche Informationen können, soweit vorhanden und vom System unterstützt, direkt beim Modellierungsprozess eingebunden werden.

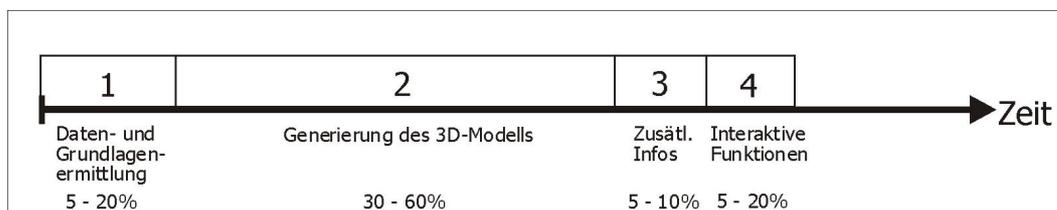
### 4. Interaktive Funktionen (5 - 20%)

Die Bereitstellung der interaktiven Funktionen und Systemfeatures ist in den meisten Fällen weiterhin auf dem herkömmlichen Weg zu absolvieren.

Durch automatische und semi-automatische Funktionen kann der Aufwand und damit die Kosten zur Erstellung eines virtuellen 3D-Stadtmodells verringert werden (s. Abb. 3.2 b). Insbesondere der größte Teil des Arbeitsaufwandes, die Generierung der 3D-Modelle, kann um ca. 20% verringert werden. Methoden, die zur Reduktion des Aufwandes eingesetzt werden können, werden in Abschnitt 4 der vorliegenden Arbeit behandelt.



(a)



(b)

Abbildung 3.2: Zeitaufwand (a) ohne und (b) mit Unterstützung durch fortschrittliche Methoden der Modellgenerierung

### 3.3. Recherche in ausgewählten Kommunen

Um digitale 3D-Stadtvisualisierungen erstellen zu können, werden entsprechende Informationen benötigt. Die grundlegenden Daten (Gebäudegrundrisse, Flurstücke, Hausnummern) können von jeder Kommune bei der liegenschaftskatasterführenden Behörde eingeholt werden. Diese Daten sind in der Automatisierten Liegenschaftskarte (ALK) und in dem Automatisierten Liegenschaftsbuch (ALB), welche zusammen das Liegenschaftskataster des LVerGeo bilden, enthalten. Auch Luftbilder können vom LVerGeo bezogen und zur Visualisierung benutzt werden. Zusätzliche Daten können eigenständig von der Kommune gesammelt und eingebracht werden, um die Plandarstellung informativer und authentischer zu gestalten.

Nachfolgend werden Rechercheergebnisse zum Umfang vorhandener stadtmodellrelevanter Daten in ausgewählten Kommunen Sachsen-Anhalts dargelegt.

### 3.3.1. Eisleben

In der Lutherstadt Eisleben sind standardmäßig Daten der ALK und ALB vorhanden. Die Straßenverläufe wurden im Auftrag der Stadt vermessen. Neben diesen Daten ist eine umfangreiche Sammlung von Sachdaten, z.B. im Bereich der Gebäude, Grünflächen, Bäume und Verkehrsschilder, vorhanden. Diese Daten wurden von der Stadt Eisleben erfasst bzw. von Behörden, wie dem Einwohnermeldeamt eingeholt. Zur Verwaltung der unterschiedlichen Daten werden mehrere GI-Systeme verwendet (vgl. Abb.3.3). Fassadenfotos oder andere Fototexturen liegen nicht vor.

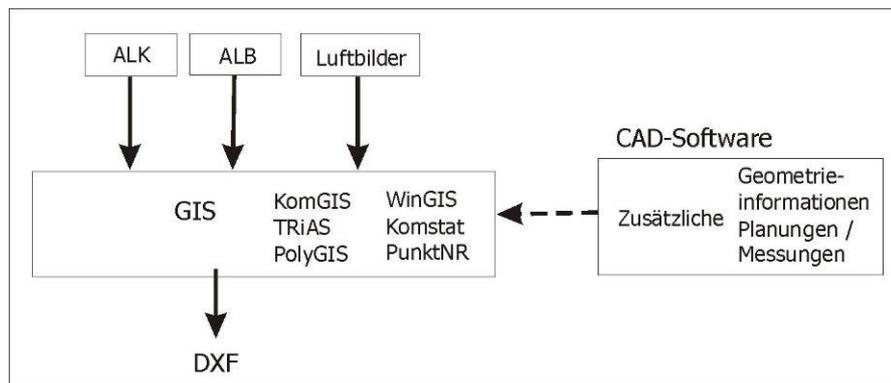


Abbildung 3.3: Zusammenfassende Systemdarstellung der Stadt Eisleben

### 3.3.2. Wittenberg

Die Stadt Wittenberg hat ebenfalls die ALK- und ALB-Daten des LVermGeo zur Verfügung. Die Gebäudedaten bezieht die Stadt aus den ALK-Daten. Die Daten der Straßenverläufe stammen aus Messungen der Stadtwerke und des Tiefbaus. Daher sind diese Daten nur in dem Umfang aufgenommen worden, wie es für die Planung der Stadtwerke oder des Tiefbaus notwendig war. Dementsprechend wurden hauptsächlich die großen Straßenzüge, weniger die kleinen Seitenstraßen und Gassen vermessen. In Wittenberg ist demzufolge keine komplette Stadtgrundlage vorhanden. Die bestehenden Lücken sollen in den nächsten Jahren nach und nach geschlossen werden.

Sachdaten zu Gebäuden und anderen Objekten der Stadt (bspw. Bäume) sind nur spärlich vorhanden. Im Gegensatz zum restlichen Stadtgebiet sind von der Wittenberger Innenstadt zahlreiche Fassadenfotos vorhanden.

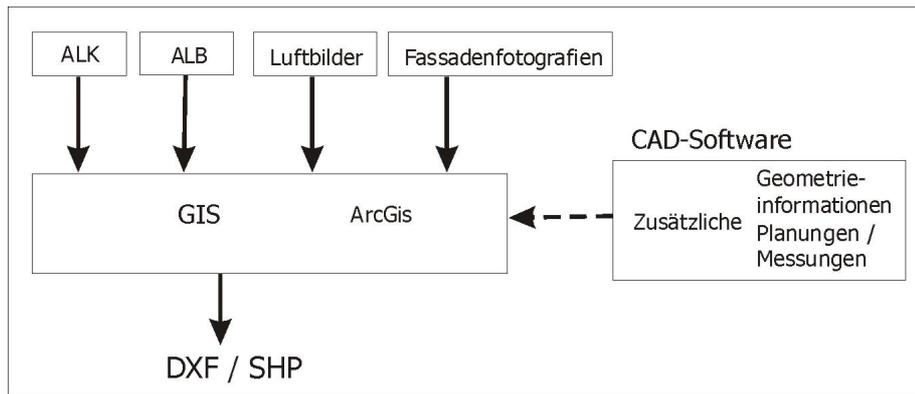


Abbildung 3.4: Zusammenfassende Systemdarstellung der Stadt Wittenberg

### 3.3.3. Staßfurt

Auch in Staßfurt sind ALK- und ALB-Daten vorhanden. Zusätzliche Sachdaten, wie Gebäude- und Bevölkerungsdaten, wurden kaum erfasst.

Luftbilder sind in der Kommune vorhanden und werden vom LVerGeo bereitgestellt, ebenso Schrägbilder ohne Georeferenzierung. Fassadenfotografien von Gebäuden sind in Staßfurt aktuell nicht vorhanden.

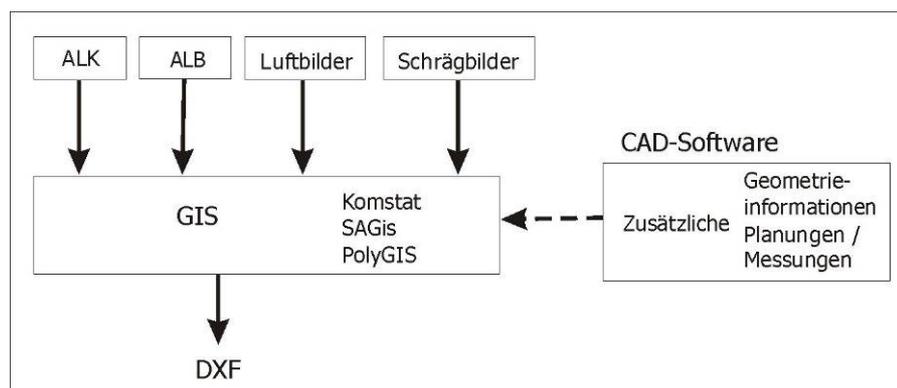


Abbildung 3.5: Zusammenfassende Systemdarstellung der Stadt Staßfurt

### 3.3.4. Zusammenfassung der Recherche

Innerhalb des Bundeslandes Sachsen-Anhalt arbeiten nicht alle Kommunen mit den gleichen GI-Systemen (vgl. Abb. 3.6). Lediglich innerhalb einer Kommune wird Wert darauf gelegt, dass die verschiedenen „Mitwirkenden“ (z. B. verschiedene Behörden, Vermessungsingenieure und Stadtwerke) mit den gleichen Werkzeugen arbeiten, um einheitliche und austauschbare Datensätze zu erhalten.

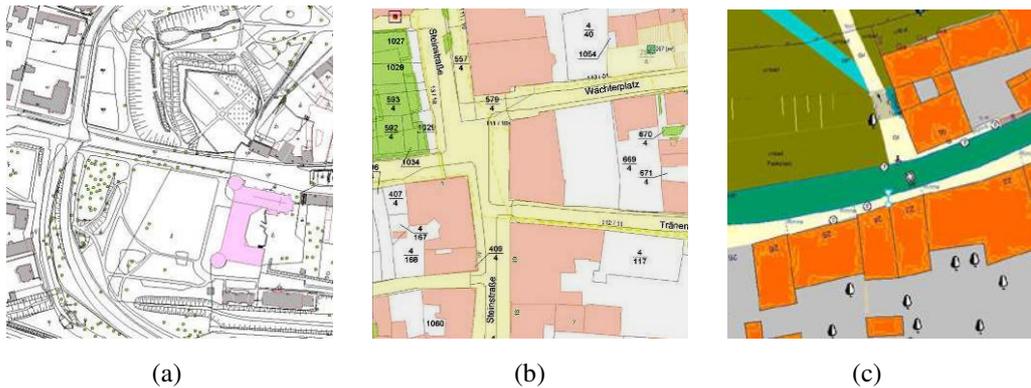


Abbildung 3.6: Ansicht von Stadtdaten der besuchten Kommunen in unterschiedlichen GI-Systemen

- (a) ArcGis9, verwendet in Wittenberg (Quelle: [KDV])
- (b) SAGis Web-Editor, verwendet in Staßfurt (Quelle: [CWSM])
- (c) TRiAS, verwendet in Eisleben (Quelle: Verwaltungsgemeinschaft Lutherstadt Eisleben)

Des Weiteren ist das Volumen bzw. die „Qualität“ der Daten, besonders der Sachdaten, in den verschiedenen Kommunen recht unterschiedlich. Ursächlich dafür sind die anfallenden Kosten, die bei einer solchen Erfassung entstehen. Deshalb wird die Erfassung von Daten oft mit anderen Projekten verbunden und es entsteht erst nach und nach ein vollständiges digitales Stadtbild.

Auch ist nicht sicher gestellt, dass die Straßengeometrien als Polygonzüge vorhanden sind. Oft liegen sie als einzelne Linien vor, die in ihren Attributen eine Referenz zum Objekt enthalten. Dieses Format ist für eine automatische Bearbeitung und Texturierung zur Erstellung digitaler 3D-Objekte unvorteilhaft, da ein direkter Zugriff auf die Geometriedaten eines Objektes nicht möglich ist. Dadurch erfordern Daten dieses Formats vorab eine extra Bearbeitung um geschlossene Polygonzüge zu erstellen. Diese Bearbeitung beinhaltet unter anderem das Triangulieren der Geometriepunkte.

Aus den GIS-Daten können nach genannter Vorverarbeitung automatisch einfache Geometriemodelle erzeugt werden. Für die meisten Gebäude der ALK entsteht so zunächst ein Blockmodell, das anschließend mit einem Dach versehen wird. Welche Dachform dabei zum Einsatz kommt, kann durch verschiedene Verfahren ermittelt werden. Hilfreich ist, wenn entsprechende Daten wie die Dachform (z.B. Satteldach, Pultdach, Walmdach oder Mansardendach) sowie dazugehörige Messdaten in den GI-Systemen gespeichert sind.

Die Geometriedaten des Straßennetzes sind teilweise lückenhaft und wenig detailliert.

Texturfotografien von Fassaden, Gehwegen oder Straßen sind in den Städten ebenfalls in unterschiedlichen Mengen vorhanden. Meist liegen ausschließlich Luftbilder vor. Fassadenfotografien existieren oft nur aus dem Bereich des Stadtzentrums.

Das Fazit der Recherche ist, dass keine effektive und ansprechende Straßentexturierung auf Grundlage der in den Städten vorhandenen Daten möglich ist. Es müssen Methoden gefunden werden, wie aus wenig detaillierten Daten eine ansprechende Darstellung der Straßengeometrien zu erreichen ist.

### 3.4. Visualisierungsprogramme mit Geobezug

In den folgenden Abschnitten werden Systeme vorgestellt, die sich mit dem Thema der dreidimensionalen Stadtvisualisierung beschäftigen, um einen Einblick in die Vielfalt der bereits existierenden Lösungen zu geben.

Die vorgestellten Systeme werden eingeteilt in

- I. Spezielle Visualisierungssysteme mit Geobezug
- II. Bestehende GI-Systeme mit 3D-Visualisierungskomponenten und
- III. Konstruktionskomponenten als PlugIn.

Spezielle Visualisierungssysteme wurden für spezifische Themen und Aufgaben der 3D-Geovisualisierung entwickelt. Bestehende GI-Systeme wurden in den vergangenen Jahren um 3D-Visualisierungskomponenten erweitert. Der Leistungsumfang dieser Komponenten ist im Allgemeinen nicht so umfangreich, wie in den speziellen Visualisierungssystemen. Konstruktionskomponenten als PlugIn erweitern herstellerfremde Software um Modellierungs- und Texturierungsmethoden.

Die nachfolgenden Informationen und Bilder zu den einzelnen Systemen stammen von den Internetpräsenzen und Produktinformationen der herstellenden Firmen.

#### 3.4.1. Spezielle Visualisierungssysteme mit Geobezug

Der **LandXplorer** der Firma 3D Geo [3DGeo] stellt eine Produktfamilie dar, die Softwarelösungen für interaktive 3D-Anwendungen, basierend auf 2D- und 3D-Geoinformationen, bietet. Dazu gehören u.a. die Module LandXplorer Studio, LandXplorer 3D GeoServer und LandXplorer Geotainment.

Die 3D Geo Desktop-Produkte (z.B. LandXplorer Studio) übernehmen unter anderem systematisch und automatisiert den Aufbau und die Fortführung von virtuellen 3D-Stadtmodellen, 3D-Landschaftsmodellen und virtuellen 3D-Umgebungen.

Mit den Modulen SmartBuildings und SmartTerrains werden Funktionen geboten, um Gebäude und Gelände mit individuellen Texturen auszustatten.

Die SmartBuildings (s. Abb. 3.7) sind stockwerkbasierte Gebäudemodelle zur Darstellung komplexer Gebäudegeometrien. Sie entstehen durch das „Übereinandersetzen“ von Stockwerken, deren Grundrisse interaktiv festgelegt werden. Weiterhin können Fassadentexturen eingesetzt und Dach- und Fensterformen zugewiesen werden.



Abbildung 3.7: LandXplorer - SmartBuilding

Der SmartTerrain-Editor, ermöglicht das Erzeugen von bodennahen Geometrien wie Straßen, Gehwegen, Mauern, Treppen oder Gras- und Wasserflächen aus 2D-Plänen. Die 3D-Geometrien werden durch Klassifizierung der 2D-Geometrien generiert. Mittels ihrer Attribute lässt sich die Darstellung individuell anpassen.

**Tridicon™** der Firma GTA Geoinformations GmbH [GTA] bietet eine Komplettlösung zur Erstellung und Nutzung von digitalen 3D-Stadtmodellen. Zur Modellerstellung, Pflege und Fortführung stehen u.a. tridicon™ 3D, tridicon™ CAPTURE und tridicon™ TEXTURE bereit.

Tridicon™ CAPTURE dient der Gewinnung terrestrischer Digitalfotos, welche in tridicon™ 3D zur Verfeinerung von Gebäudekörpern bzw. in tridicon™ TEXTURE zur fotorealistischen Texturierung von Gebäudefassaden Verwendung finden. Das Modul tridicon™ TEXTURE ermöglicht eine halbautomatische Texturierung, während tridicon™ 3D zum Editieren der Geländemodelle, Gebäudeobjekte und anderen topografischen Objekten dient.

Neu ist das Modul tridicon™ CityDiscoverer, welches 2007 auf der Messe InterGeo erstmals vorgestellt wurde.



Abbildung 3.8: tridicon™ CityDiscoverer – Darstellung der Veste Coburg

Der tridicon™ CityDiscoverer soll es ermöglichen, beliebig große 3D-Stadt- und Landschaftsmodelle zu visualisieren, zu modifizieren und zu analysieren. Es können u.a. Geländemodelle, Orthofotos, Gebäudemodelle, Vegetation, Karten aus dem GIS sowie Sachdaten zu einem Gesamtmodell zusammen gefügt werden. Einfache Gebäude lassen sich direkt in der Software konstruieren. Anschließend

können Planungsvarianten, Sichtlinien und Abschattungen angezeigt sowie Ansichtspunkte und Flugrouten definiert werden. Des Weiteren können Partikelsysteme (z.B. Wasser und Feuer) integriert und thematische Karten erstellt werden.

**VIS-All 3D** der Firma Software-Service John [SSJ] ist eine für Visualisierung, Planung und Simulation geeignete Software, deren Schwerpunkt in der Simulation und Animation der Elemente und Szenarien liegt. Die Software verfügt über eine offene COM-Schnittstelle und kann somit mit verschiedenen CAD- und GI-Systemen verbunden werden.

Die 3D-Szene kann auf der Erdkugel frei positioniert werden. Der Sonnenstand berechnet sich nach Lage der Szene und in Abhängigkeit von gewählter Tageszeit und Datum. Es können verschiedene Simulationen durchgeführt werden, beispielsweise des Sonnenstandes sowie Hochwasser- und Flugschneisen-simulationen. Weiterhin sind zeitabhängige Manipulationen von GIS-Elementen, wie drehende Windräder, über Skripte möglich. Weitere Funktionen sind Messungen von Strecken direkt im 3D-Raum und das Ein- bzw. Ausblenden von Einzelobjekten oder Objektgruppen.

Die Gebäudeobjekte sind automatisch erstellte Blockmodelle, die später mit eigenen Werkzeugen texturiert werden können. Die Texturierung erfolgt manuell in einem eigenen Texturfenster.

Zur Verfeinerung des Modells können 3D-Objekte aus Bereichen wie Vegetation, Straßenmöbel, Fahrzeuge und Menschen aus einer Bibliothek hinzu geladen werden.



Abbildung 3.9: Vis-All3D - Ausschnitt des Projekts Dortmund (Quelle: [SSJ])

Das Softwaresystem **IVS\_VDT** ist eine Eigenentwicklung des Fraunhofer-Instituts für Fabrikbetrieb und Fabrikautomatisierung (IFF).

Das System wurde konzipiert, um virtuelle Szenarien für Lernanwendungen, Veranschaulichungen und Training gestalten und darstellen zu können. Auf Basis von 3D-CAD-Daten und unter Verwendung umfangreicher Modellierungsfunktionen können beispielsweise Wartungs- und Montageprozesse realer Maschinen trainiert werden.

Der universelle Charakter des Visualisierungssystems macht das System auch für andere Themenbereiche nutzbar, wie beispielsweise die Stadtplanung [Höp05]. Es wurden unter anderem Stadtmodelle von Magdeburg und Eisleben erstellt. Die Generierung der 3D-Modelle erfolgt überwiegend manuell. Zur Texturierung werden wie in Abbildung 3.10 zu sehen, Luftbilder verwendet.



Abbildung 3.10: Blick auf die Innenstadt im Stadtmodell Magdeburg (Quelle: [IFF])

### 3.4.2. Bestehende GI-Systeme mit 3D-Visualisierungskomponenten

ESRI (Environmental Systems Research Institut) bietet mit **ArcGIS** [ArcGIS] eine Produktfamilie, welche aus gut aufeinander abgestimmten GIS-Produkten besteht.

Die Erweiterung ArcGIS 3D Analyst ist hier für die Erstellung von 3D-Stadtmodellen von Bedeutung. Diese Erweiterung enthält die Module ArcScene und ArcGlobe. ArcScene erlaubt dem Anwender direkt aus den GIS-Daten dreidimensionale Ansichten zu erstellen, zu analysieren und zu modifizieren. Weitere Funktionen der Software sind Oberflächenberechnung durch Interpolation, Oberflächenanalyse, interaktive Abfrage von 3D-Elementen, Volumenberechnungen und Sichtbarkeitsanalysen. Für die Visualisierung von extrem großen 3D-Datenmengen in globalem oder lokalem Kontext kommt das Modul ArcGlobe zum Einsatz. Auch „nicht 3D Daten“ werden mit Hilfe von ArcGlobe in räumliche Ansichten integriert (Abb. 3.11).

Ferner enthält die Software über 500 vorgefertigte 3D-Modelle, die in die Szene integriert werden können.

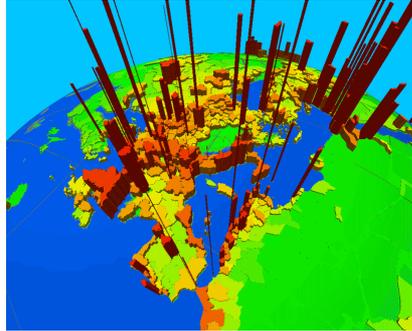


Abbildung 3.11: ArcGIS 3D Analyst - Visualisierung von „nicht 3D-Daten“ (Quelle [3DA])

### 3.4.3. 3D-Konstruktionskomponenten als PlugIn

**SketchUp Pro 5** [SkUP], ein Produkt von Google, ist ein Werkzeug zum Erstellen, Anzeigen und Ändern von dreidimensionalen Ideen. Es dient vorrangig zum Zeichnen und Skizzieren von Gebäuden und Umgebungen im Cartoon-Stil.

Verwendung findet dieses Tool beispielsweise in der Architektur, Innenarchitektur, Landschaftsarchitektur, Film und Bühne, Spieleentwicklung und im GIS-Bereich. Um den Einsatz im GIS-Bereich zu gewährleisten, existiert ein PlugIn für die GIS-Produkte der Firma ESRI (ArcGIS® plug-in), welches die Daten aus dem GIS für SketchUp exportiert. Die exportierten Dateien können anschließend in SketchUp geöffnet und bearbeitet werden. So können die automatisch erstellten Blockmodelle verfeinert und texturiert sowie Bäume und andere Objekte hinzugefügt werden. [SA07]

SketchUp ist geeignet, um einfache Gebäudedarstellungen und deren Texturierung manuell zu erstellen. Als Erweiterung ist angedacht, Attributinformationen einfließen zu lassen, beispielsweise die Nutzung von Gebäudeteilen (s. Abb. 3.12).

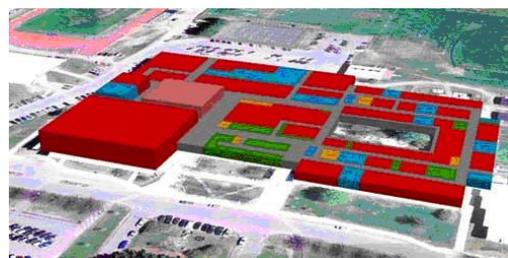


Abbildung 3.12: SketchUp - farbliche Visualisierung der unterschiedlichen Nutzung von Gebäudebereichen (Quelle: [SkUP])

**Nverse Photo** der Firma Precision Lightworks [Nverse] ist ein PlugIn für die Konstruktionsprogramme Autodesk VIZ®, 3ds Max™ und Carbon Graphics Geo. Es ermöglicht das Erstellen von 3D-Grafiken mit der gewohnten CAD-Software. Grundlage sind hier keine GIS-Daten sondern zwei Schrägbilder, aus denen die Geometriedaten mittels 2D-Interaktionen ermittelt werden.

Um die Daten aus den Schrägbildern zu erhalten, werden diese in das Programm geladen und anschließend die Kameraposition ermittelt, von der die Bilder aufgenommen wurden. Dazu werden in einem der beiden Bilder drei oder vier einfache Geometrien nachgezeichnet. Diese Objekte werden in das zweite Bild kopiert und dort angepasst. Ist die Kameraposition gefunden, werden anschließend weitere Objekte eingezeichnet, in das zweite Bild kopiert und dort automatisch ausgerichtet und skaliert. Die zugehörigen *texturemaps* werden automatisch erzeugt.

Bodenflächen sind ebenfalls mit Nverse Photo generierbar. Um die Fläche zu erzeugen, werden drei bis vier korrespondierende Punkte in beide Bilder gezeichnet. Die Bodentextur entsteht durch ein *resampling* eines der beiden Ausgangsbilder.

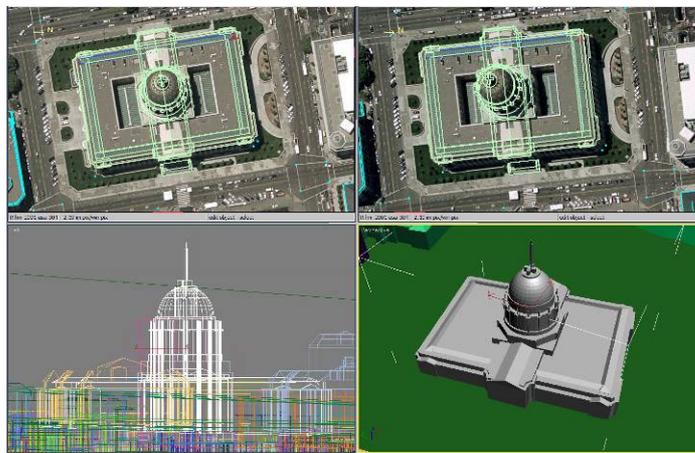


Abbildung 3.13: Nverse Photo in als PlugIn in 3ds Max™

### Zusammenfassung zu Visualisierungsprogramme mit Geobezug

Die hier vorgestellten Programme arbeiten mit unterschiedlichen Ausgangsdaten. Die meisten nutzen Geometriedaten der GI-Systeme, andere wie Nverse Photo beziehen ihre Daten aus Luft- bzw. Schrägbildern. Gemein ist allen, dass der Schwerpunkt auf der Generierung und Gestaltung von Gebäuden liegt. Die Bodengeometrien werden in der Regel mit Luftbildern versehen. Mit Ausnahme der Erweiterung SmartTerrainEditor des Werkzeugs LandXplorer Professional, berücksichtigt keines der vorgestellten Programme die Generierung von Bodengeometrien. Die separate Behandlung von Bordsteingometrien wird in keinem der betrachteten Programme unterstützt.

### 3.5. Existierende Stadtmodelle

Dieser Abschnitt beschreibt einige ausgewählte bereits existierende virtuelle Stadtmodelle. Dabei wird zwischen Stadtmodellen unterschieden, die von der zugehörigen Stadt oder von anderen Auftraggebern erstellt wurden.

Auch hier stammen die Informationen und das Bildmaterial von den Internetpräsenzen der jeweiligen Firmen bzw. Auftraggebern.

#### 3.5.1. Private bzw. unternehmerische Auftraggeber

**Google Earth**, ein Produkt von Google [GEarth], verbindet die herkömmliche Internetsuche von Google mit Satellitenbildern, Landkarten, Geländedaten und 3D-Gebäuden. Die 3D-Gebäude entstehen nicht bei Google selbst, sondern werden durch verschiedene Firmen, wie 3D Geo und Privatpersonen eingebunden. Dadurch werden sowohl ganze Stadtbereiche als auch einzelne Gebäude integriert. In vielen Städten werden zunächst touristisch attraktive Gebäude eingebunden, nach und nach kommen weitere hinzu. Demzufolge sind 3D-Objekte zum einen nicht flächendeckend, zum anderen vorrangig in großen Städten und an touristisch interessanten Orten zu finden. Viele dieser Gebäude sind zurzeit nur texturierte Klötze. Es sind jedoch auch Gebäude im LOD 3 und LOD 4 vorhanden, wie einige Sehenswürdigkeiten in Berlin.

Der Grund und Boden wird mit Luftbildern texturiert, welche für das deutsche Gebiet zu 90% von der Firma GeoContent [GeoC] stammen. Für Google Earth sind gute Texturen sehr wichtig, sie sind die Basis für dessen Erfolg.

Zusätzlich können die Gebäude in Google Earth mit Informationen und URL-Adressen verbunden werden.

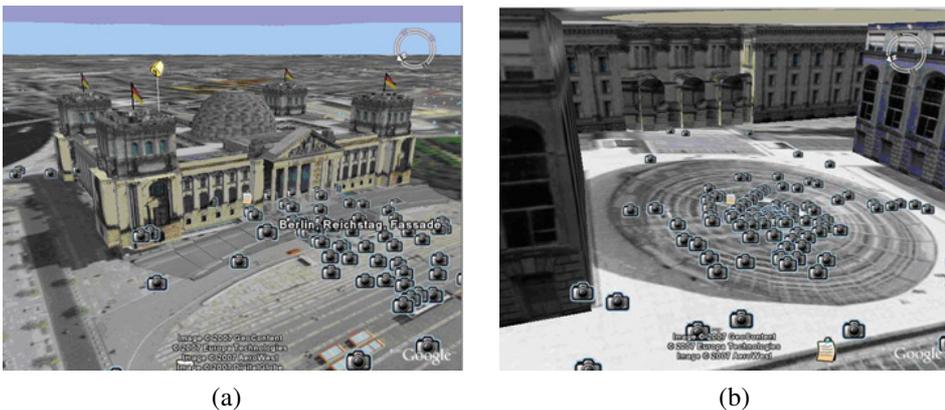


Abbildung 3.14: Google Earth - Berliner Reichstag von außen (a) und einer Innenansicht (b)

Die **3D RealityMaps** sind ein Produkt der Firma Remote Sensing Solutions GmbH [RSS]. Sie erstellen und verkaufen 3D-Stadtpläne, mit denen sich im Stadtgebiet navigieren lässt. Es können Rundflüge gestartet, Adressen angefliegen, sowie das U- und S-Bahnnetz eingesehen werden. Zurzeit sind Pläne von München, Berlin und Potsdam verfügbar.

Die Modelle entstanden aus Gelände-, Gebäude- und Straßendaten, die durch das Vermessungsamt bereitgestellt wurden. Dazu kamen Bilder einer hochauflösenden Stereo Kamera, mit welcher der darzustellende Bereich aufgenommen wurde. Auch in diesen Modellen sind gute Texturen wichtig, um den Wiedererkennungswert zu erhöhen.



Abbildung 3.15: 3D RealityMaps - Ansicht des Modells von München  
(Quelle: [München])

### 3.5.2. Behördliche Auftraggeber

Die Senatsverwaltung für Stadtentwicklung der Stadt **Berlin** [Berlin] bietet auf ihrer Internetpräsenz einige Beispielbilder ihres 3D-Innenstadtmodells.

Ziel des Projekts ist die Entwicklung und Umsetzung von Methoden zur nachhaltigen Nutzung eines 3D-Stadtmodells. Sowohl das entwickelte Autorensystem als auch das Präsentationssystem basieren auf der Grundlage des LandXplorer-Systems der Firma 3D Geo GmbH. Die Datenbasis des Berliner Stadtmodells besteht aus 2D-Geodaten und bereits vorhandenen 3D-Stadtmodellldaten der Berliner Verwaltung. Zusätzlich wurden auf einem 250 km<sup>2</sup> großen Innenstadtgebiet 3D-Geodaten mit photogrammetrischen und Laser-Scanning-Methoden ermittelt. In den Daten sind alle Detailstufen der Gebäudemodelle vorhanden, d.h. von LOD 0 bis LOD 4. Langfristig soll für das gesamte Stadtgebiet ein amtliches Stadtmodell erzeugt werden.

Anwendung findet das 3D-Stadtmodell in der Stadtplanung und der Wirtschaftsförderung. Entscheidungsprozesse in der Politik und bei Investoren können begleitet werden, indem vorhandene Standorte und Vorhaben virtuell zusammengefasst und mit Fachinformation bearbeitet und präsentiert werden. Teile des 3D-Stadtmodells, rund 10%, sind ebenfalls in Google Earth zu sehen.



Abbildung 3.16: 3D-Modell von Berlin - Potsdamer Platz (Quelle: [Berlin])

In den Beispielbildern und –videos sind ausschließlich Luftbilder als Bodentextur zu sehen, bzw. zusätzliche grafische Darstellungen, wie den ehemaligen Mauerverlauf.

Die Stadt **Coburg** [SMC] wurde von der GTA Geoinformatik GmbH flächendeckend modelliert. Das gesamte Stadtgebiet liegt im LOD 1 vor, nur der Stadtkern und die Veste Coburg sind mit Dachformen und Texturen versehen (s. Abb. 3.17). Zur Texturierung der Bodengeometrie wurden Luftbilder verwendet.

Als Grundlage für das Stadtmodell wurden Daten der digitalen Liegenschaftskarte, dem Baumkataster sowie das DGM und Luftbilder genutzt.

Verwendung findet das 3D-Stadtmodell bereits als Entscheidungshilfe in der Stadtplanung sowie als Basis für Umwelt- und Lärmanalysen. Zur Nutzung im städtischen GIS sind die 3D-Gebäudeobjekte durch den Gebäudeschlüssel eindeutig mit der digitalen Liegenschaftskarte verknüpft.



Abbildung 3.17: Stadtmodell Coburg im Blockmodell, nur der Stadtkern und die Veste Coburg (im Hintergrund) sind detailliert modelliert und texturiert

### Zusammenfassung zu den existierenden Stadtmodellen

In den vorgestellten Stadtmodellen werden die in den vorherigen Abschnitten herausgearbeiteten Defizite zwischen der Gebäude- und der Bodenmodellierung sichtbar. Es werden hauptsächlich Luftbilder als Bodentextur verwendet, die aus der Fußgängerperspektive stark von der Qualität der Gebäudetexturen abweichen.

### 3.6. Innovative fotobasierte Techniken

Die nachfolgend vorgestellten Techniken sind innovative Techniken, die auf der Ver- und Bearbeitung von Fotografien basieren. Das Ergebnis dieser Methoden sind 3D-Objekte, jedoch keine digitalen Stadtmodelle.

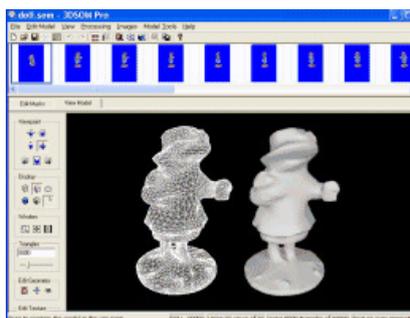
**3DSOM** (3D Software Object Modeller) [3DSOM] ist eine Software der Firma Creative Dimension Software Ltd zur automatischen Modellierung realer Objekte aus Fotografien. Dabei kann eine große Reihe an Objekten erzeugt werden, auch solche mit Löchern und „schwierigen“ organischen Formen.



(a)



(b)



(c)



(d)

Abbildung 3.18: 3DSOM

- a) Aufbauanordnung für Aufnahme der Fotos
- b) Freistellen der Objekte im Bild
- c) Erzeugung der 3D-Geometrie aus den freigestellten Objekten
- d) Texturierte Geometrie

Die Bilder der Objekte können mit einer handelsüblichen Kamera aufgenommen werden. Dabei steht das Objekt im Zentrum einer bedruckten Matte, vor einem einfarbigen Hintergrund, um das Freistellen des Objektes später zu erleichtern (s. Abb. 3.18 a). Unter diesen Voraussetzungen wird eine Reihe von Fotos aus verschiedenen Blickwinkeln aufgenommen.

Anschließend wird die Form des Objektes in den Fotografien freigestellt. Für Bilder, die nicht unter optimalen Beleuchtungsbedingungen aufgenommen wurden, stellt 3DSOM eine Reihe von manuellen und halbautomatischen Werkzeugen bereit (s. Abb. 3.18 b).

Im Anschluss wird aus den freigestellten Formen das *3D-mesh* erzeugt (s. Abb. 3.18 c). Die Geometrie kann manuell bearbeitet werden.

Das Aufbringen der Texturen erfolgt durch das Überblenden der einzelnen Input-Bilder. Mittels so genannter „Copy and Paste“-Funktionen, wie sie aus Bildbearbeitungsprogrammen bekannt sind, kann die Textur manuell bearbeitet werden (s. Abb. 3.18 d).

Nach Fertigstellung können die Objekte in verschiedene Formate exportiert werden, um sie in Web-Seiten, anderen Autorenprogrammen und Anwendungen integrieren zu können.

**VideoTrace** [VTrace] ist ein Modellierungsprogramm zur Generierung von 3D-Objekten aus Videos. Um das Geometriemodell zu erstellen, wird die Objektform in einem oder mehreren Frames eines Videos nachgezeichnet. Das Programm berechnet aus den Zeichnungen automatisch das Geometriemodell. Die entstandenen Objekte können anschließend in virtuelle Umgebungen integriert werden.

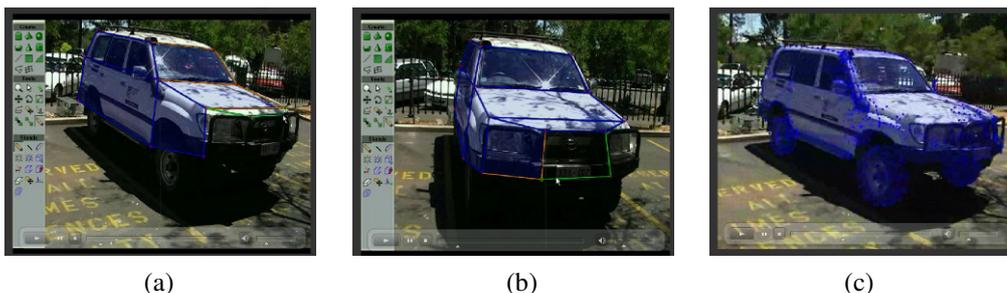


Abbildung 3.19: VideoTrace

- (a) Beginn des Nachzeichnens der Objektkonturen in einem Frame,
- (b) Erweitern der Zeichnung in einem anderen Frame,
- (c) fertiges Geometriegerüst

Die beiden vorgestellten Verfahren sind innovative flexible Möglichkeiten zur Generierung von Geometriemodellen. Auf einen möglichen Einsatz in digitalen 3D-Stadtmodellen wird in Abschnitt 4.3 eingegangen.

### 3.7. Zusammenfassung

Die Anzahl der Städte, die digitale 3D-Stadtmodelle einsetzen, ist in den letzten Jahren ständig angestiegen. Anlass dazu waren nicht zuletzt die Fortschritte in der Erstellung der Modelle, aber auch die größere Bedeutung und Nutzbarkeit für die Städte und deren Partner, wie oben geschildert.

Digitale 3D-Stadtmodelle werden häufig in ostdeutschen Städten verwendet, die nach 1990 ihr Stadtbild ändern und attraktiver gestalten wollen. Planerische und touristische Anwendungen stehen daher im Vordergrund. Für touristische Anwendungen sind detaillierte realistische 3D-Modelle von Vorteil. Der oft geringe Umfang vorhandener detaillierter Geometrie- und Sachdaten erschwert dies für kleine Kommunen. Es werden Methoden benötigt, die aus den wenigen Daten eine ansprechende Visualisierung erstellen.

In diesem Abschnitt wurden u. a. Datenbestände in Kommunen, existierende Stadtmodelle und Modellierungssoftware zur Erstellung dieser betrachtet. Zusammenfassend betrachtet ist der Funktionsumfang der beschriebenen Modellierungssoftware, das Anwendungsgebiet und die Motivation zur Erstellung der erwähnten Stadtmodelle häufig recht unterschiedlich. Es entstehen dreidimensionale Modelle unterschiedlicher Größe und Detaillierung. Die Software hat je nach Spezifikation unterschiedliche Methoden zur Erstellung der Geometrie.

Allen gemein ist das Ziel, eine dem Zweck entsprechende, möglichst genaue Modellierung und Visualisierung umzusetzen. Ob ein spezielles Visualisierungstool, eine GIS-Erweiterung oder PlugIn-Komponenten zur Erstellung verwendet werden, hängt von den Visualisierungszielen, Anwendungsbereichen und Ausgangsdaten sowie den Gewohnheiten und Kenntnissen der Nutzer ab.

In den Beschreibungen der betrachteten Stadtmodelle und Modellierungssoftware sind selten konkrete Angaben zu den verwendeten Texturierungsmethoden von Grund und Boden enthalten. Auf vielen Beispielen sind Luftbildtexturen zu sehen, während Gebäude einen hohen Detailgrad aufweisen. Dementsprechend kann darauf geschlossen werden, dass bei einem Großteil der Modellerstellungsprogramme innerhalb der Texturierung der Schwerpunkt auf den Gebäuden, nicht aber auf den Bodengeometrien liegt.

Um eine Diskrepanz zwischen den Qualitäten der Gebäude- und der Bodentextur zu vermeiden (s. Abb. 2.6 b), ist oft manuelles Texturieren und vorheriges Bearbeiten der Bodengeometrie nötig. Dies ist sehr zeitaufwendig und die Ergebnisse sind oft nicht maßstabsgetreu. Lediglich bei der Verwendung von Orthofotos kann davon ausgegangen werden, dass das Verhältnis zwischen Textur und Geometrie korrekt ist. Aber auch diese Bilder haben heute noch nicht die gute Auflösung von Texturfotografien. Die beste momentan zu erreichende Auflösung von Luftbildern beträgt 20cm/Pixel (lt. GeoContent), wobei diese nicht überall vorliegt und daher im Allgemeinen von schlechteren Auflösungen ausgegangen werden muss.

Daher ist es notwendig, Verfahren zur semi-automatischen fotorealistischen Texturierung der Bodengeometrie zu entwickeln. Dieser Aufgabe widmet sich das folgende Kapitel.

## 4. Analyse und Entwurf

In den nachfolgenden Abschnitten dieses Kapitels wird auf die semi-automatische Texturierung von Geometriemodellen, im Speziellen von Grund- und Bodengeometrien eingegangen. Zunächst erfolgen eine Anforderungs- und Ablaufanalyse zur Texturierung in 3D-Stadtmodellen. Im Anschluss wird auf Varianten zur Erzeugung und Texturierung der Objekte in 3D-Stadtmodellen eingegangen. Besonderes Augenmerk liegt auf der Texturierung der Straßengeometrien. Schließlich folgt das Konzept zur Gestaltung eines Benutzerdialogs.

### 4.1. Anforderungsanalyse

Im Idealfall ist es mit einer Texturierungssoftware möglich, im Virtuellen Raum ein originalgetreues Abbild der vorhandenen und geplanten Realität nachzubilden. Wie dieses Ziel zu erreichen ist, sollen die kommenden Betrachtungen zeigen.

Die zu entwickelnden Methoden sollen eine Vereinfachung des Erstellungsprozesses ermöglichen. Dabei ist ein Kompromiss zwischen den Anforderungen der Zielgruppe, der gewünschten Funktionalität und einer optimalen Systemreaktion zu finden.



Abbildung 4.1: Eine Auswahl an unterschiedlichen Pflastervarianten aus der Altstadt der Lutherstadt Wittenberg

**Zielgruppe** der Modellierungs- und Texturierungssoftware sind im Kontext der vorliegenden Arbeit Anwender, die digitale Stadtmodelle erstellen wollen, ohne diese Aufgabe an externe Experten abzugeben. Es kann davon ausgegangen werden, dass dies vorwiegend auf Beschäftigte aus den Bereichen Stadtplanung und Städtebau sowie Stadt- und Standortmarketing zutrifft.

Die Stadtplanung und der Städtebau beschäftigen sich mit der vorausschauenden Ordnung der baulichen und sonstigen Nutzung von Grund und Boden.

Die Stadtplanung lenkt dabei die räumliche Entwicklung und Nutzung der städtischen Flächen, der Städtebau ist für die Umsetzung der Planung und die bauliche Gestaltung des urbanen Raumes zuständig.

Die Anwender aus den genannten Bereichen verfügen über Kenntnisse in ihren Fachgebieten, haben jedoch nicht zwingend Erfahrung mit Modellierungs- und Texturierungswerkzeugen. Somit sind die Nutzer gemäß der Charakterisierung nach Shneiderman [Shn02] zu den „Anfängern bzw. erstmaligen Benutzern“ zu zählen. Anwender dieser Gruppe haben zu Beginn der Arbeit mit neuen Programmen Bedenken. Diese gilt es durch eine Begrenzung des Vokabulars auf eine kleine Anzahl von vertrauten und konsequent benutzten Konzeptbegriffen abzubauen. Auch eine geringe Anzahl von benötigten Handlungen zur Erfüllung von einfachen Aufgaben kann Vertrauen aufbauen, Ängste reduzieren und zu einer positiven Verstärkung führen. Aus diesem Grund sollte die Interaktion und Kommunikation einfach und verständlich gehalten werden.

Die **Funktionalität** der Software soll den Texturierungsvorgang umfassen, der soweit als möglich automatisch geschehen soll. Zur manuellen Anpassung müssen dem Anwender Werkzeuge zur Verfügung gestellt werden. Diese sollen effizient, und sogleich leicht bedienbar sein.

Eine **schnelle Systemreaktion**, im Sinne von zeitnahe Reagieren auf die Eingaben des Anwenders im Texturierungsprozess, ist erforderlich für die Akzeptanz des Systems durch den Anwender. Das bedeutet, dass die vom Anwender eingegebenen Einstellungen zeitnah visualisiert werden müssen. Gleichzeitig ist darauf zu achten, dass durch die verwendeten Algorithmen und Texturen keine Verzögerung bei der Präsentation des Modells auftreten. Der Gesamteindruck der Visualisierung steht u. a. im Stadt- und Standortmarketing im Vordergrund, deshalb sollte eine ausreichende Bildwiederholungsrate gesichert sein. Selbst bei den beeindruckendsten Effekten wird die Qualität der Präsentation durch Unterbrechungen der Animation sehr gemindert.

Das **Ergebnis** sollte eine möglichst realistische Texturierung sein, die einfach, schnell und mit wenig Aufwand zu erstellen ist.

## 4.2. Vegetation und straßenbegleitende Objekte

**Vegetationsobjekte** werden in der Regel aus Modelldatenbanken importiert und manuell positioniert. Es gibt Ansätze zur Erkennung von Bäumen in Luftbildern mit Hilfe der Mustererkennung. Sind die Luftbilder georeferenziert, kann hiermit eine automatische Platzierung im Stadtmodell erreicht werden. Es ist auch denkbar, dass Positionsdaten aus Datensammlungen des LVerGeo genutzt werden kann, z.B. das Baumkataster. Damit wären neben der Position zusätzlich Informationen über Art und Alter verfügbar.

Für Stadtmodelle kleiner detaillierter Bereiche ist die Verwendung von gesammelten Daten des Baumkatasters sinnvoll. Mit Daten zu Alter, Art und Größe der Vegetationsobjekte kann eine genauere Nachbildung der 3D-Szene geschaffen werden.

Die Geometrie der **straßenbegleitenden Objekte** stammt ebenfalls meist aus Modelldatenbanken, die Positionierung erfolgt oft manuell. Die Position der Objekte kann ebenfalls aus Datensammlungen des Katasters bezogen werden, wenn diese, ähnlich eines Baumkatasters, von der Stadt geführt werden. Damit sind Informationen über Art und Position verfügbar, die zur automatischen Platzierung genutzt werden können. Des Weiteren ist es denkbar, Straßenumöbel wie Verkehrsschilder aus Luft-, speziell Schrägbildern zu filtern. Als Unterstützung können charakteristische Schattenwürfe benutzt werden.

Jedoch ist die heute zu erreichende Auflösung dieser Fotografien nicht ausreichend hoch, um eine erfolgreiche Mustererkennung und damit Identifizierung der einzelnen Straßenschilder zu gewährleisten. Hinzu kommt, dass Luftbilder, die zur automatischen Auswertung und Texturierung aufgenommen werden, idealerweise keine Schatten enthalten sollten, um Diskontinuitäten im Farbverlauf zu vermeiden.

### 4.3. Gebäude und Bauwerke

Wie bereits in Abschnitt 2.4.1 der vorliegenden Arbeit beschrieben, existieren verschiedene Verfahren zur Rekonstruktion von Gebäuden. Je nach Ausgangsdaten und verfügbaren Verfahren können so unterschiedlich detaillierte und genaue 3D-Modelle entstehen.

Neben den dort erwähnten Verfahren können ebenfalls die Techniken 3DSOM und VideoTrace (s. Abschn. 3.6. der vorliegenden Arbeit) zur Gewinnung von Geometrieinformationen eingesetzt werden, insbesondere für Denkmäler, Brücken, Kunstobjekte und andere Wahrzeichen. Diese sind schwer durch automatische Verfahren zu erfassen und bedürfen daher einer manuellen Rekonstruktion.

Eine Bearbeitung der Geometrie ist immer dann nötig, wenn Gebäude detaillierter dargestellt werden sollen, als es die Ausgangsdaten erlauben. Gerade in kleinen Städten ist der Umfang der Einmessungen nicht sehr umfassend. Es kann sich dabei aber auch um automatisch schwer zu erfassende Details handeln, wie beispielsweise der zurückgesetzte Eingangsbereich eines Gebäudes.

Durch die Bearbeitung der Geometrie innerhalb der 3D-Szene können solche Details leicht hinzugefügt werden. Denkbar sind Methoden nach dem Sketch-Up-Prinzip (vgl. Abschn. 3.4.3), oder das schlichte Einfügen und Verbinden zusätzlicher Geometriepunkte. Die Positionierung der neuen Punkte kann durch bereits aufgebrauchte Fassadentexturen erleichtert werden.

### **Ablaufanalyse zur Texturierung von Gebäudemodellen**

Das gebräuchliche Verfahren zur (detaillierten) Texturierung von Gebäuden beginnt mit dem Importieren der modellierten Geometrie in ein CAD-Tool. Wurde das Gebäude von Hand modelliert, kann es in der Regel mit dem gleichen Modellierungstool texturiert werden. Ist das Objekt automatisch entstanden, so muss es meist aus dem jeweiligen Programm exportiert und in ein CAD-Programm geladen werden. Dort werden die entsprechenden Polygone oder Dreiecke selektiert, denen dann eine Textur zugewiesen wird. Erfahrungsgemäß folgen weitere Bearbeitungen, wie etwa das Zuweisen und Einstellen einer UVW-Map<sup>8</sup>. Im Anschluss wird das Objekt erneut exportiert und umgewandelt, um es in die 3D-Szene zu laden. Dies wird für alle zu texturierenden Objekte wiederholt. Soll bei einem Objekt die Textur, zum Beispiel auf Grund einer Umgestaltung verändert werden, muss es erneut in das CAD-Tool exportiert, texturiert und anschließend wieder in das Visualisierungstool importiert werden. Das erforderliche Umwandeln in die entsprechenden Dateiformate (z.B. DXF in 3DS oder WRL) verursacht einen hohen Arbeitsaufwand und ist mit Fehlerquellen, z.B. in Form von Datenverlusten, verbunden.

Ein integriertes Verfahren kann das Umwandeln zwischen den Arbeitsschritten vermeiden und diese Fehlerquelle eliminieren.

Dazu werden die Geometrien automatisch aus den zur Verfügung stehenden Daten generiert, texturiert und in die virtuelle Umgebung überführt. Wird davon ausgegangen, dass die Daten direkt aus einem GIS exportiert wurden, besteht die Möglichkeit, dass den Geometrien bereits Texturen zugeordnet sind. Ist dies nicht der Fall, kann den Objekten aufgrund des ALK-Objektschlüssels (vgl. Abschn. 2.2) eine generische Textur zugewiesen werden. Ein Gebäude mit dem Schlüssel 1112 - Rathaus bekäme beispielsweise eine Sandsteintextur. Die Auswahl der Textur kann weitere Objekteigenschaften, wie Baujahr und Nutzung des Gebäudes, berücksichtigen. Wird das Baujahr mit beachtet, kann für modernere Gebäude vom Typ 1112 beispielsweise eine Klinkertextur verwendet werden. Falls erforderlich, ist es anschließend möglich, die verwendete Textur direkt im virtuellen Raum anzupassen.

Ein weiterer Vorteil der integrierten Methode ist die Erhaltung der Veränderbarkeit von Geometrien. Das Bearbeiten der Geometrien in der 3D-Szene ermöglicht es, die Ansicht von Gebäuden schnell und unkompliziert zu variieren. Dieser Vorteil ist besonders nützlich beim Aktualisieren von Umgestaltungen oder beim Vergleich von Varianten im Planungsstadium, um die ideale Variante zu bestimmen.

Da der Schwerpunkt der vorliegenden Diplomarbeit nicht auf der Betrachtung von Gebäuden und Bauwerken liegt und auf diesem Gebiet bereits gute Verfahren existieren, wird nicht weiter darauf eingegangen.

---

<sup>8</sup> UVW-Mapping - Technik, um 2D-Bilder auf eine 3D-Oberfläche anzuwenden

## 4.4. Grund und Boden

Die Gestaltung von Fahrbahnen und Gehwegen reicht in der Realität von schlichten Asphaltstraßen bis hin zu reich detaillierten Pflasterungen. Ein kleiner Überblick wurde bereits in Abbildung 4.1 gegeben. Die Ausgangsdaten, die zur Erstellung eines 3D-Stadtmodells vorliegen, spiegeln diese Vielfalt in der Regel nicht wieder.

### 4.4.1. Ablaufanalyse zur Texturierung von Grund und Boden

Im Bereich der Texturierung von Bodengeometrien sind aktuell hauptsächlich zwei Extreme anzutreffen. Die erste Variante besteht darin, ein Luftbild zu nutzen und dieses auf den Boden zu projizieren. Diese Variante hat den Vorteil, dass der Aufwand gering ist und das „Ziel nach einem Arbeitsschritt erreicht wurde“. Ein großer Nachteil besteht darin, dass der Unterschied zwischen den gut ausmodellierten Gebäuden mit ansprechender Textur und dem schlecht aufgelösten, unscharfen Luftbild am Boden in den meisten Fällen erheblich ist. Zudem enthalten Luftbilder immer Objekte, wie Fahrzeuge und Bäume, die dann „platt gedrückt“ am Boden zu sehen sind. Dies fällt einem Betrachter, der auf Augenhöhe eines Fußgängers durch das Modell „gleitet“, unangenehm auf (vgl. Abb. 2.6).

Das andere Extrem, Variante 2, ist die fein ausmodellerte und anschließend texturierte Bodengeometrie, die in aufwendiger und „zeitraubender“ Handarbeit erstellt wird. Hier werden einzelne Elemente, wie Fahrbahn, Bordsteine und Entwässerungsrinnen, Parkflächen sowie Geh- und Radwege als einzelne Objekte modelliert. Diese können anschließend unterschiedliche und individuelle Texturen erhalten. Optimalerweise können Fototexturen genutzt werden, die im Zusammenspiel mit Ortskenntnissen und der daran angepassten Modellierung ein realitätsgetreues Bild der Szene wiedergeben. Auch ohne fotorealistische Texturen wirkt bei dieser Methode das Zusammenspiel der Boden- und Gebäudegestaltung wesentlich harmonischer für den Betrachter, als bei der Verwendung von Luftbildtexturen für die Bodengeometrie.

Eine dritte Variante, die den Mittelweg zwischen den beiden beschriebenen Texturierungsmethoden von Grund- und Bodengeometrien einschlägt, benutzt das Texturieren von Straßen vorwiegend mit generischen Texturen. Dabei wird meist das Vorhandensein von Bordsteinen, Gehwegen und Abflüssen vernachlässigt, da diese oft nicht in den originalen Geometriedaten zu finden sind.

Um eine automatische Zuordnung von Texturen zu den Geometrien zu erhalten, können erneut die Objektschlüssel der ALK oder eine Flächennutzungskarte herangezogen werden.

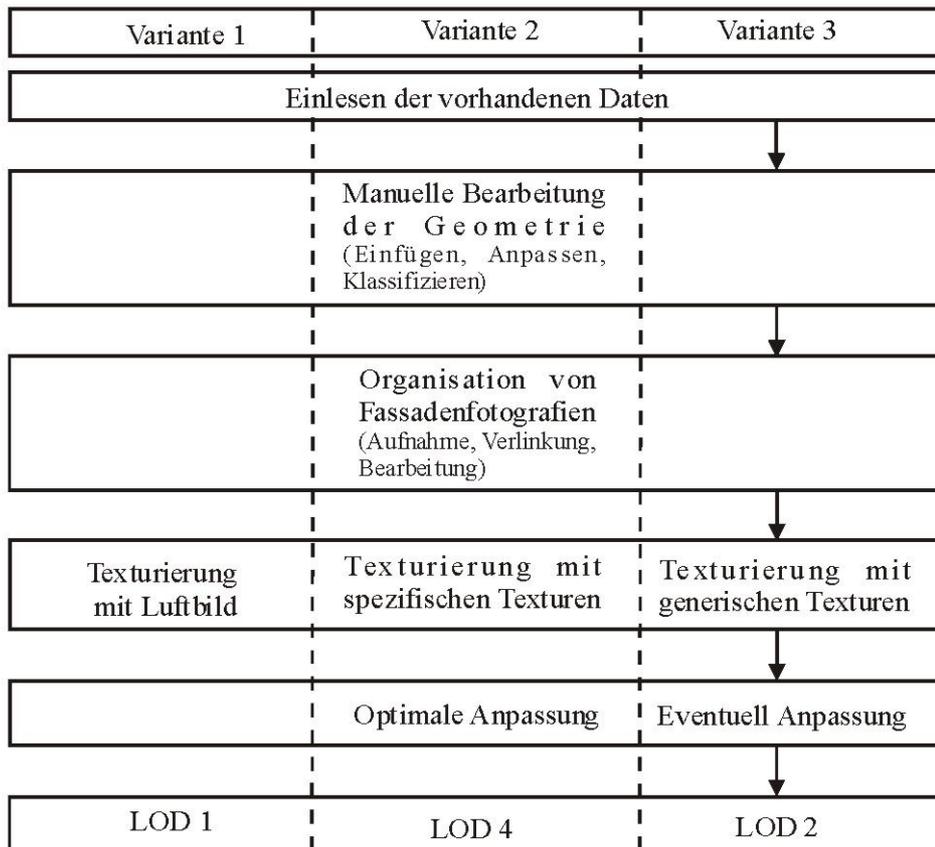


Abbildung 4.2: Schematische Darstellung der unterschiedlichen Vorgehensweisen in der Straßentexturierung

Nachfolgend soll sich auf eine Kombination der vorgestellten Varianten bezogen werden. Es wird davon ausgegangen, dass vollständige, aber nicht detailreiche Daten vorliegen. D.h. die Geometrie ist lückenlos vorhanden, aber nicht detailliert unterteilt und klassifiziert. Somit sind Klassifikationen im Rahmen der ALK möglich (z.B. Wiese, Ackerland und Straße), jedoch keine Unterteilung in Fahrbahn und Gehweg. Eine schematische Darstellung der Eigenschaften eines Polygonzuges ist in Abbildung 4.3 dargestellt. Zur Texturierung der Bodengeometrien sollen Texturkacheln aus Fotografien genutzt werden.

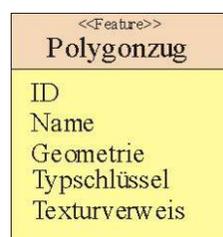


Abbildung 4.3: Eigenschaften eines Polygonzuges mit Berücksichtigung des Typschlüssels

#### 4.4.2. Schematische Darstellung zur Generierung digitaler Straßenpolygone

Im nachfolgenden Abschnitt wird eine Übersicht zum konzipierten Ablaufschema der automatischen Texturierung von Straßenpolygonen gegeben.

Um ein texturiertes Straßenpolygonnetz zu erhalten, müssen zunächst die Daten aus einem GI-System importiert werden. Je nach Format der Daten sind bereits geschlossene Polygonzüge vorhanden. Andernfalls werden diese in einem nachfolgenden Schritt erstellt. Dies ist die Datengrundlage, wie sie von der Kommune zu erhalten ist und umfasst den oberen umrahmten Bereich der Abbildung 4.4.

Der zweite eingerahmte Bereich umfasst die Schritte, welche zur Erstellung einer angepassten Texturierung notwendig sind. Die vorliegenden Polygonzüge müssen zunächst trianguliert werden. Ein Großteil der Triangulation kann automatisch erfolgen. Eine Nachkontrolle des Ergebnisses ist jedoch sinnvoll. Der Triangulierung der Geometriedaten sollte vor der Texturierung eine hohe Aufmerksamkeit geschenkt werden, da die Texturierung auf den Ergebnissen der Triangulierung aufbaut. Eine hohe Qualität der Vernetzung wirkt sich positiv auf die automatischen Ergebnisse aus und vermindert somit den Nachbearbeitungsaufwand des Anwenders.

Anschließend kann die semi-automatische Texturierung einsetzen. Zunächst erfolgt das Zuweisen der zu verwendenden Textur. Dies geschieht manuell durch den Anwender oder automatisch durch Auslesen von Eigenschaften der GIS-Daten. Im nächsten Schritt wird die Anpassung der Textur an die Orientierung des Polygons berechnet. Auf Grundlage dieser Werte erfolgt die Texturierung. Sind die erhaltenen Ergebnisse zufrieden stellend, endet der Prozess an dieser Stelle. Bei mangelhaften Ergebnissen kann der Anwender eine manuelle Nacharbeitung durchführen. Dazu zählen die Änderung von Orientierungs- und Mapping-Einstellungen sowie das Verändern der Triangulierung. Änderungen sind durch das Einfügen und Löschen von Punkten sowie durch eine erneute Triangulation von ausgewählten Punkten möglich. Nachteilig ist, dass durch das Löschen und Einfügen die Konsistenz der Daten nicht mehr gewährleistet ist.

Im Anschluss an die Anpassungen durch den Anwender erfolgt ein Sprung zu einem früheren Verarbeitungsschritt. Es erfolgt ein erneuter Vorschlag für die Texturorientierung. Der Anwender hat die Möglichkeit so lange Änderungen vorzunehmen, bis das gewünschte Ergebnis erreicht wurde.

Auf den grau hinterlegten Bereich in der Schema-Darstellung Abb. 4.4, die semi-automatische Texturierung, sollen sich die Betrachtungen der nachfolgenden Abschnitte beziehen.

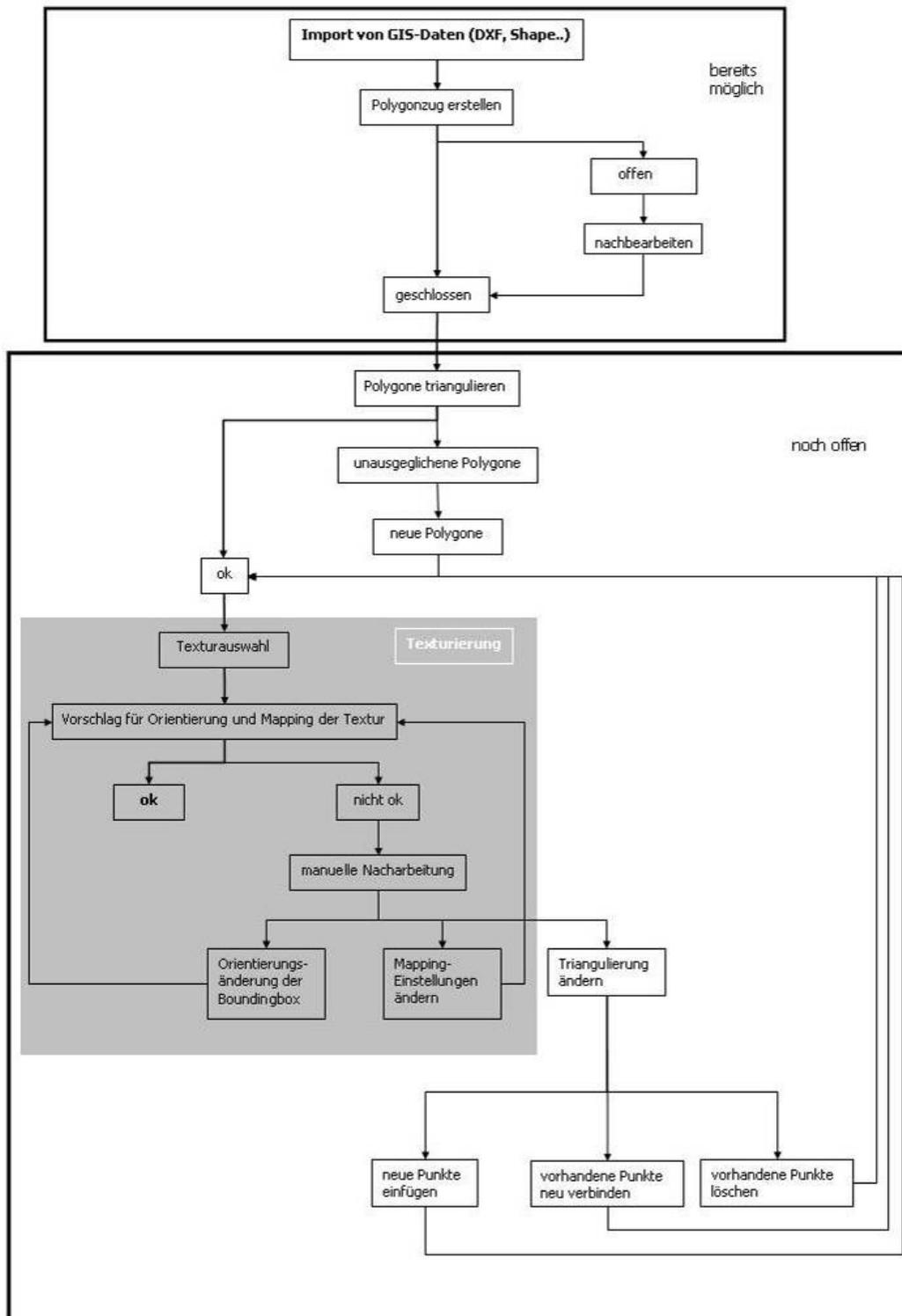


Abbildung 4.4: Ablaufschema zum Prozess der Erstellung texturierter Straßenpolygone

#### 4.4.3. Ausgangsgeometrie

Die Ausgangsdaten der Bodengeometrie können, wie in Abschnitt 2.4.2. beschrieben, aus verschiedenen Quellen stammen. Für die nachfolgenden Betrachtungen wird von Katasterdaten und zusätzlichen Einmessungen der jeweiligen Städte im Bereich der Straßengeometrie ausgegangen. Kommunen, wie die in Abschnitt 3.3. vorgestellten Städte, stehen in der Regel keine Höhendaten aus LIDAR- oder photogrammetrischen Messungen zur Verfügung.

Für die kommenden Betrachtungen werden an dieser Stelle folgende Annahmen getroffen:

- Die Geometrien sind geschlossen und gut trianguliert.
- Von den Flächen ist die Nutzung bekannt (Schlüssel aus ALK oder Flächennutzungskarte).
- Die Texturen sind maßstäblich und in einer vordefinierten Größe (z.B. 1m x 1m).

Um die gegebene Geometrie mit Texturen zu versehen, können verschiedene Vorgehensweisen gewählt werden. Die einfachste beinhaltet das „einfache *mapping*“ der Textur auf das Objekt. Erweitert werden kann diese durch das Berücksichtigen der objektorientierten *bounding box* und der Differenzierung zwischen geraden und gekrümmten Strecken. Eine zusätzliche Erweiterung zur Verbesserung des optischen Eindrucks, ist die Detaillierung der Straßengeometrie, z.B. durch das Ergänzen eines Bordsteins.

Die vier aufgeführten Verfahren sollen im nächsten Abschnitt eingehender betrachtet werden.

#### 4.4.4. Geometrieuntersuchung zur Optimierung der Texturierung

Die oben genannten Vorgehensweisen sollen in den kommenden Abschnitten in folgender Reihenfolge vorgestellt werden:

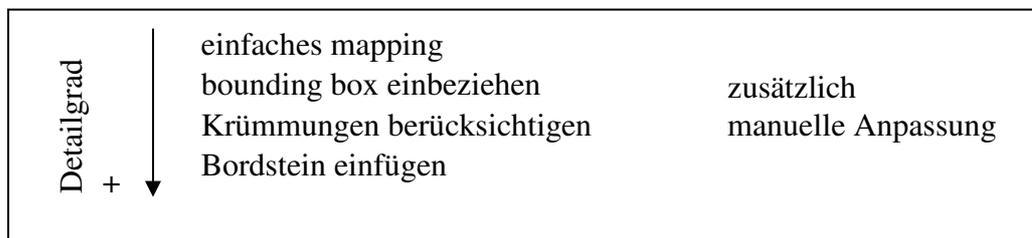


Abbildung 4.5: Schema zur schrittweisen Erweiterung und Detaillierung der Texturanpassung und Geometriedifferenzierung

Mit dem wachsenden Detailgrad der Texturierung steigt die Komplexität der Betrachtung.

#### 4.4.4.1. Einfaches mapping

Beim „einfachen *mapping*“ wird die ausgewählte Textur gekachelt auf die Geometrie projiziert, ohne auf die Orientierung der Textur relativ zum Objekt zu achten. Es entsteht der Eindruck einer aus bemustertem Papier ausgeschnittenen Schablone (vgl. Abb. 4.6)



Abbildung 4.6: Einfaches *mapping* einer Textur auf ein gekrümmtes Straßenpolygon

Manuelle Werkzeuge zum Drehen und Verschieben der Textur können Verbesserung verschaffen, jedoch kann der Aufwand bei Bearbeitung aller Flächen recht groß werden.

#### 4.4.4.2. Texturanpassung mit Hilfe der Objektausrichtung

Eine einfache Möglichkeit, eine automatische Rotation der Textur in Richtung der Objektausrichtung zu erhalten, ist das Berücksichtigen der Orientierung im *texture mapping*-Prozess. Die Objektausrichtung lässt sich auf Grundlage der Berechnung der *object-orientated bounding box* (OOBB) und der Skelettierung eines Objektes (s. Abschn. 2.5.5.) ermitteln.

Zur Ermittlung der **Skelettkanten** kann der Algorithmus von Xuetao et al. [XTTZ01] benutzt werden.

Skelettierungsalgorithmus:

- Finden der kürzesten Kante im Polygon
- Löschen der Punkte dieser Kante
- Einfügen eines neuen Punktes im Mittelpunkt der gelöschten Kante
- Verbinden des neuen Punktes mit den Kanten der gelöschten Punkte
- Streichen der gelöschten Dreiecke aus der Dreiecksliste
- Abbruchkriterium: Dreiecksliste ist leer
- Alle übrigen Kanten sind Skelettkanten

Die Skelettkanten sollen anschließend zur Bestimmung der Texturorientierung für die Dreiecke des Straßenpolygons dienen, welche sie schneiden. Der Vektor vom Start- zum Endpunkt kann eine Orientierung des Gesamtobjektes liefern.

Um die Orientierung einer **OBB** zu berechnen, existieren verschiedene Verfahren. Ein Verfahren ermittelt diese mithilfe einer Kovarianzmatrix [Eri05]. Der Grundgedanke ist, dass der Eigenvektor mit dem größten Eigenwert jener Achse entspricht, auf der die Punkte die größte Varianz aufweisen. Dieser Vektor kann als Orientierung für die *bounding box* und somit der Textur verwendet werden.

Das Berechnen einer *bounding box* mittels der Kovarianz funktioniert im Allgemeinen gut für gleichmäßig verteilte Punkte im Objektraum. Auf einzelne Straßenpolygone kann dies zutreffen. Werden jedoch mehrere Polygone selektiert, ist eine uniforme Verteilung nicht garantiert. Der Einfluss der innen liegenden Punkte kann die Orientierung verfälschen. Aus diesem Grund sollten nach Ericson [Eri05] *bounding box*-Berechnungen auf Grundlage des Gewichtens von Punkten vermieden werden.

Die zum Zweck der Orientierungsfindung von der Verfasserin entwickelte Methode (nachfolgend Methode 2) verfolgt den Ansatz, die beiden „Start- und Endpunkte“ (in der Abb. 4.7 dargestellt durch die grün eingefärbten Punkte) des Polygons zu finden und daraus die Objektausrichtung zu berechnen. Dazu wird zunächst untersucht, ob die Ausdehnung der Punkte in x- oder y-Richtung größer ist. Entlang der größeren Ausdehnung werden die Punktepaare mit den kleinsten und größten Werten bestimmt. Die Punkte der Paare werden gemittelt und der Vektor zwischen den Mittelpunkten berechnet. Dieser entspricht der Orientierung des Objektes.

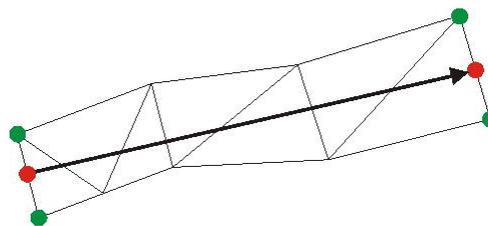


Abbildung 4.7: Ermittlung der Ausrichtung eines Polygons anhand der „Start- und Endpunkte“

Algorithmus der Methode 2:

- Erstellung einer Liste mit den äußeren Kanten des Polygons
- Finden der am weitesten entfernten Punktepaare
- Berechnung der Orientierung anhand der gefundenen Punktepaare

Für die Entscheidung zwischen den beiden vorgestellten Methoden zur Berechnung der Orientierung, wurden mit einigen Polygonen Tests durchgeführt. Eines der Testpolygone ist im Anhang abgebildet.

Das Resultat der Skelettierung war nicht immer zufrieden stellend, da abhängig von der Reihenfolge der gelöschten Punkte (bei gleich langen Polygonkanten), unterschiedliche Skelettkanten entstanden. Zudem sind nicht alle enthaltenen Orientierungen erfasst wurden. Um eine angepasste Texturierung zu ermöglichen, sind diese jedoch notwendig.

Ein Nachteil der von der Verfasserin entwickelte Methode (Methode 2) ist, dass das Verfahren nur für „schlanke“ Geometrien geeignet ist. Ist die Breite der Straße größer als die Punkteabstände an den Seiten der Geometrie, können falsche Punktpaare ermittelt werden und somit wird ein inkorrekt orientierter Vektor berechnet.

Straßenpolygone sind im Allgemeinen solche „schlanken“ Polygone, sodass diese Einschränkung geringer wiegt, als die nicht vollständigen Orientierungen der Skelette. Somit sprechen die Ergebnisse gegen eine Verwendung der Skelettierung.



Abbildung 4.8: Textur eines gekrümmten Straßensegments unter Berücksichtigung der Objektausrichtung nach Methode 2

Mit der Berücksichtigung der Objektausrichtung kann die Texturanpassung für einige Objekte verbessert werden. Bei Straßengeometrien, die gerade Strecken und Krümmungen enthalten, wird durch diese Methode das virtuelle Erscheinungsbild nicht wesentlich verbessert. In diesen Fällen ist eine zusätzliche Unterscheidung in gerade und gekrümmte Abschnitte notwendig. Die Lösung dieses Problems behandelt der nachfolgende Abschnitt.

#### 4.4.4.3. Automatische Erkennung von Krümmungen

Wenn im Straßenverlauf Kurven und Krümmungen enthalten sind, führt sowohl das „einfache *mapping*“ als auch eine Texturierung unter der Berücksichtigung der OBB zu unschönen Effekten. Die Textur folgt beharrlich der allgemeinen Objektausrichtung, aber nicht den Krümmungen innerhalb der Strecke (vgl. Abb. 4.6 und Abb. 4.8).

Um dies zu umgehen, kann ein Straßenzug in mehrere Abschnitte unterteilt und für diese erneut die Ausrichtung gefunden werden. Die Unterteilung kann über geeignete Interaktionsmöglichkeiten vom Nutzer ausgeführt werden, oder automatisch im Visualisierungsprogramm. Bei der automatischen Unterteilung untersucht das Texturierungsprogramm den Verlauf des Straßenzuges und unterteilt ihn automatisch, abhängig von der Stärke der Krümmung, in unterschiedlich große Abschnitte.

Erweiterter Algorithmus zur orientierten Texturierung (Variante 1):

- Erstellung einer Liste mit den äußeren Kanten des Polygons.
- Finden von Winkeländerungen zwischen aufeinander folgenden Kanten.
- Bei Winkeländerung über einen Schwellenwert hinaus, erfolgt die Sektionsbildung.
- Finden der am weitesten entfernten Punktepaare eines Segments.
- Berechnung der Orientierung anhand der am weitesten entfernten Punkte.

Die Untersuchung auf Krümmungen erfolgt im entwickelten Algorithmus ausschließlich auf den Randpunkten der Geometrie.

*Definition:* Randpunkte

Die Liste der Randpunkte, nachfolgend auch Border genannt, enthält alle äußeren Punkte der Geometrie und besitzt die Eigenschaft „Orientierung“. Diese gibt an, in welcher Richtung die Punkte abgelaufen werden, im Uhrzeigersinn (UZS) oder gegen den Uhrzeigersinn (GUZS). Ein Punkt der Border hat genau zwei Kanten, je eine zum Vorgänger und Nachfolger. Somit fallen innere Punktverbindungen, die aus der Triangulation stammen, weg.

Die Randpunkte werden nacheinander auf Winkeländerungen überprüft. Überschreitet der berechnete Winkel einen festgelegten Schwellenwert, werden die bis zu diesem Punkt analysierten Punkte und ihre zugehörigen Dreiecke als ein zusammengehöriger Abschnitt (auch Sektion) erfasst und ausgewählt.

Bei der Überprüfung der Geometrie sind verschiedene Besonderheiten und Varianten zu beachten.

1. Die Geometrie kann Ausbuchtungen enthalten (vgl. Abb. 4.9).  
Diese sollten die Richtung der Textur nicht beeinträchtigen. Um dies zu erreichen, muss der „Winkeländerungstest“ auf mehrere aufeinander folgende Randpunkte ausgeweitet werden. So kann vermieden werden, dass die Ausbuchtung fälschlicherweise als Krümmung erkannt wird.
2. Welcher Geometriepunkt ist der Startpunkt und in welcher Richtung bzw. Reihenfolge wird die Geometrie abgearbeitet?  
Es ist nicht voraussagbar, an welchem Punkt der Geometrie der Algorithmus startet. Der Startpunkt kann sowohl von der Speicherung der Punkte, als auch von der vorausgegangenen Verarbeitung abhängen. In ungünstigen Fällen werden dadurch fälschlicherweise Krümmungen erkannt, die keine korrekte Texturierung erlauben (s. Abb. 4.9).
3. Die Straßengeometrie kann aus parallelen Polygonen bestehen  
Diese Variante ist in Straßengeometrien selten zu erwarten, sollte jedoch berücksichtigt werden. Um die daraus entstehenden Schwierigkeiten zu umgehen, ist es sinnvoll, bei der Abtastung nur auf den Randpunkten des Polygons zu arbeiten. Außerdem müssen bei der Bildung der Sektion auch die Dreiecke berücksichtigt werden, die nicht direkt mit den „abgelaufenen“ Punkten verbunden sind (s. Abb. 4.17)

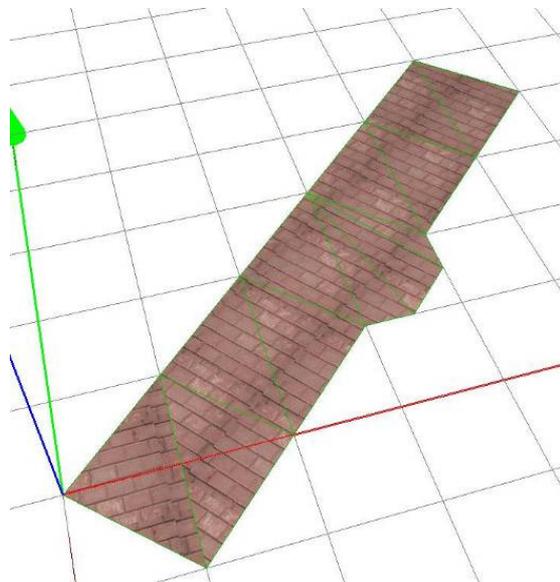


Abbildung 4.9: Geometrie mit so genannter Ausbuchtung - die Texturierungsrichtung wird trotz Krümmungserkennungsalgorithmus an dieser Stelle nicht verfälscht, jedoch ist der Startpunkt im Koordinatenursprung ungünstig, da diese Geometrie im Gegenuhrzeigersinn abgelaufen wird und somit das erste Dreieck als einzelne Sektion erkannt wird

Um ungünstige Startpunkte (vgl. Abb. 4.9) zu vermeiden, sollte eine „Vorkontrolle“ der Punkte erfolgen. Dies kann über das gleiche Verfahren zur Bestimmung der Ausrichtung geschehen, welches zum Finden der Start- und Endpunkte durchgeführt wird. Da diese Variante nur für schmale Objekte geeignet ist, sollte der nachfolgende Winkeltest in Betracht gezogen werden.

Unter der Annahme, dass die Geometrien ausschließlich nahezu rechtwinklige Abschlusskanten haben, kann ein Test auf rechte Winkel erfolgen. Dazu muss bei zwei benachbarten Punkten einer kurzen Kante ein Winkel von ca.  $90^\circ$  zwischen den jeweils angrenzenden Kanten existieren. Je nach Umlaufrichtung müssen beide positiv oder negativ sein und es wird der linke bzw. obere oder rechte bzw. untere als Startpunkt gewählt.

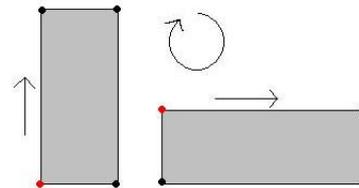


Abbildung 4.10  
Optimale Startpunkte  
(rot) bei Umlaufrichtung  
im Uhrzeigersinn

#### 4.4.4.4. Kombination von Objektausrichtung und Sektion

Sind die einzelnen Sektionen erkannt, werden sie entsprechend ihrer Ausrichtung texturiert. Es kann ein Ergebnis vergleichbar der Abbildung 4.10 a entstehen. Gegenüber der Abbildung 4.6 ist eine deutliche Verbesserung zu erkennen, jedoch entstehen harte Bruchkanten. Modellierer lösen dieses Problem häufig, indem das Kurvenstück in viele kleine Teilbereiche unterteilt und jeder Bereich einzeln texturiert wird. Diese Verfahrensweise beinhaltet das Einfügen neuer Punkte und hat einen hohen Arbeitsaufwand zur Folge. Wird dieses Vorgehen auf die automatische Erkennung übertragen, entsteht auch hier ein großer Rechenaufwand. Zudem wird die Ausgangsgeometrie durch das Einfügen neuer Punkte verändert.

Bei dem oben (S. 63 oben) beschriebenen Verfahren zur Sektionsbestimmung, wird bei jeder erneuten Krümmung eine neue Sektion bestimmt (Variante 1). Folgen mehrere Krümmungselemente aufeinander, wird jedes Segment einzeln texturiert.

Die Segmentgröße ergibt sich aus dem Aufbau des Polygons, abhängig von der Krümmung und Anzahl der Polygonpunkte. Um im automatischen Prozess die Größe der Segmente zu beeinflussen, kann der Schwellenwert des Winkels variiert werden (Variante 2).

Eine andere Möglichkeit (Variante 3) ist das Zusammenfassen aller aufeinander folgenden Segmente zu einem einzelnen. Eine neue Sektion beginnt erst, wenn wieder ein gerades Teilstück anschließt. Die Ausrichtung wird auf der Basis aller zugehörigen Punkten ermittelt und die Texturierung erfolgt über alle zusammen gefassten Sektionsteile.

Ein bildlicher Vergleich der Varianten 1 und 3 ist in Abbildung 4.10 zu sehen.

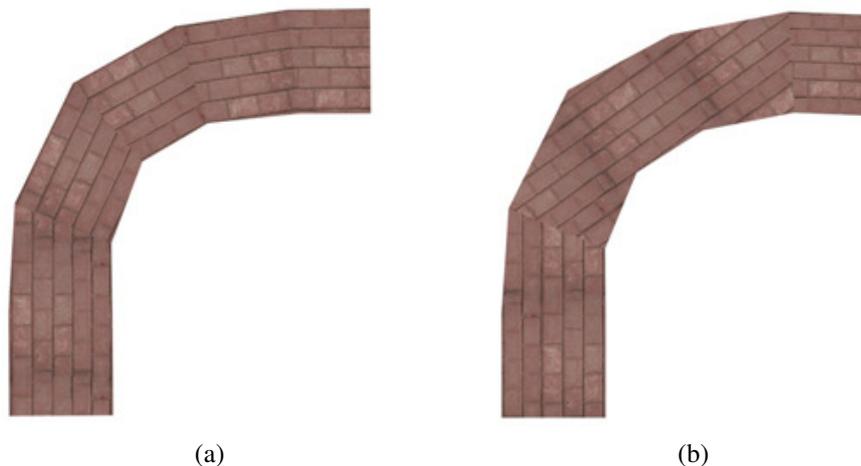


Abbildung 4.7: Vergleich Variante 1 (a) und Variante 3 (b), der mittlere Teil in (b) ist zu einer Ausrichtung zusammengefasst wurden.

Vorteil der Variante 3 ist, dass bei stark untergliederten Geometrien weniger Orientierungsberechnungen notwendig sind. Dies kann jedoch gleichzeitig ein großer Nachteil sein. Das zusammengefasste Segment kann bei vielen aufeinander folgenden Krümmungssegmenten sehr groß werden. Dadurch entspricht die ermittelte Ausrichtung eher einer Annäherung. In ungünstigen Fällen können Krümmungen in unterschiedlichen Richtungen enthalten sein. Um dies auszuschließen kann in die Berechnung der Winkeländerung eine Kontrolle eingebaut werden. Diese Kontrolle testet, ob sich der Winkel um mehr als einen festgelegten Wert (z.B.  $45^\circ$ ) ändert. Tritt dieser Fall ein, sollte eine neue Sektion beginnen.

Bei Verwendung der Variante 1 können viele sehr kleine Segmentflächen entstehen. Dies bedeutet einen erhöhten Rechenaufwand, führt aber in der Regel zu guten Ergebnissen.

### **Problem:**

Ist ein Polygonzug durch eine Krümmungserkennung in mehrere Sektionen unterteilt, kann sich die berechnete Orientierung einzelner Sektionen sehr stark von der Orientierung des Gesamtobjektes unterscheiden (s. Abb. 4.11). Dies tritt auf, wenn die Sektionen sehr kurzen Teilstrecken des Polygons entsprechen. Um hier eine fehlerhafte Texturierung zu vermeiden, sollten zusätzliche Parameter berücksichtigt werden.

Zunächst scheint die Berücksichtigung der Gesamtorientierung oder der zuletzt bearbeiteten Sektion (s. Abb. 4.12 a) zu helfen. Doch bieten diese Varianten nur in wenigen Fällen eine gute Lösung.

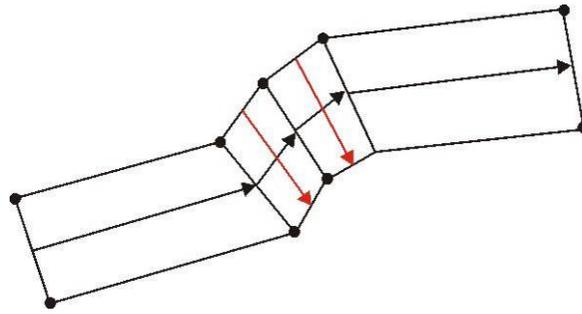


Abbildung 4.8: Das Polygon wurde in mehrere Abschnitte unterteilt. Die schwarzen Pfeile zeigen die korrekte Texturausrichtung. Die roten Pfeile zeigen die berechnete Texturausrichtung von schmalen Segmenten, wenn ausschließlich die Lage der Sektionspunkte betrachtet wird.

Eine andere Möglichkeit ist, die Berücksichtigung der Segmentkante, die zur Krümmungsbestimmung genutzt wurde. Die Textur wird parallel zu ihr ausgerichtet. Hier kommt es jedoch bei nicht parallelen Polygonkanten ebenfalls zu Abweichungen. Für eine Großzahl der Fälle liefert diese Variante jedoch die besseren Ergebnisse.

Beide zuletzt genannten Varianten können für verschiedene Fälle vor- oder nachteilig sein.

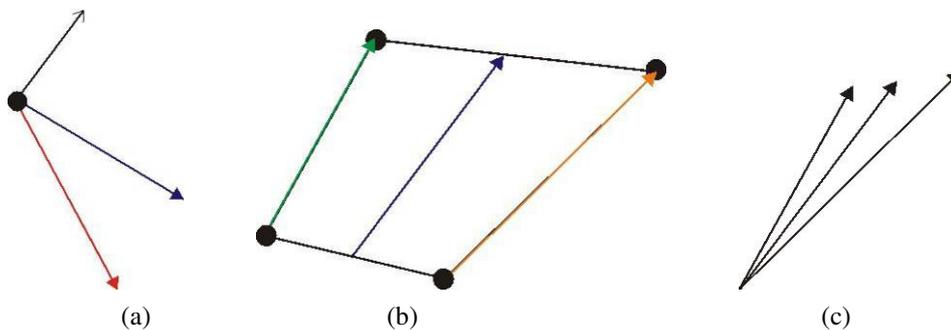


Abbildung 4.9: Varianten

- (a) Der rote und schwarze Vektor entspricht den Orientierungsvektoren eines kurzen Segments aus Abb. 4.11. Der blaue Vektor ist die Summe aus diesen. Die Verwendung des blauen Vektors als Texturorientierung führt zu keinem guten Ergebnis.
- (b) Drei mögliche Texturausrichtungen, die sich aus dem berechneten Vektor (blau) und den beiden Kanten (grün und gelb) ergeben.
- (c) Vergleich der drei Vektoren aus (b).

Das automatische Verfahren der orientierten Texturierung mit Berücksichtigung der OOB und Erkennung von Krümmungen hat den Vorteil, dass dem Nutzer weniger Aufwand gegenüber dem „einfachen *mapping*“ und dem anschließenden manuellen Korrigieren entsteht.

Es ist dennoch sinnvoll die Funktion des manuellen Ändern von Orientierungen für Polygone eines Straßenzuges dem Nutzer anzubieten, um nachträgliche Korrekturen durchführen zu können.

#### 4.4.4.5. automatische Erkennung von Kreuzungen

Kreuzungen sind ein Schnittpunkt von zwei oder mehr Straßen, bei denen unter Umständen verschiedene Texturen aufeinander treffen. Es gibt verschiedene Arten von Kreuzungen, mit den Grundtypen X, T und Kreisverkehr [Ian04].

Bei einer Kreuzung vom Typ X begegnen sich vier Straßen, wobei diese in verschiedenen Winkeln aufeinander treffen können. So ist beispielsweise auch die Form eines „+“ möglich, wie in Abbildung 4.13 a zu sehen.

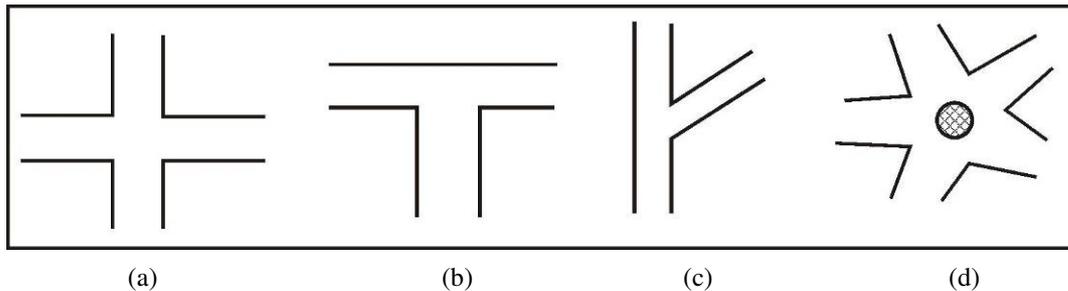


Abbildung 4.10: Kreuzungsvarianten – a) X-Typ, b) T-Typ, c) Y-Typ, d) Kreisverkehr

Eine Kreuzung vom Typ T besteht aus drei sich treffenden Straßen. Die Anlehnung an den Buchstaben T ist frei gewählt. Es existieren auch Erscheinungsformen, die einem „Y“ oder „V“ entsprechen.

In einem Kreisverkehr treffen sich meist mindestens drei Straßen. Die Besonderheit besteht in der Kreisfahrbahn um die Mittelinsel.

Sind die an einer Kreuzung beteiligten Straßen in unterschiedlichen Polygonen enthalten, entstehen keine zusätzlichen Komplikationen für die Texturierung. Die Polygone werden einzeln als „normale“ Straßenzüge behandelt und texturiert. So entsteht meist eine durchgängige Pflasterrichtung, von der die anderen Straßen mit den eigenen Ausrichtungen abzweigen (vgl. Abb. 4.14).

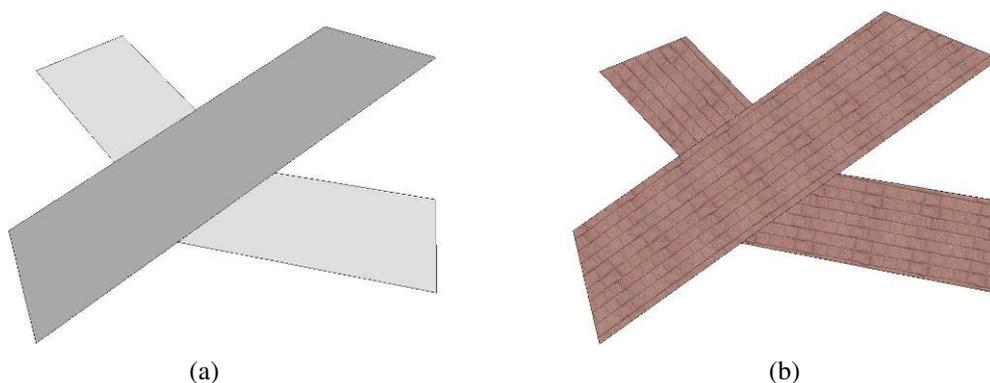


Abbildung 4.11: (a) Eine Kreuzung bestehend aus drei Polygonen,  
(b) Es werden alle Polygone einzeln entsprechend ihrer Ausrichtung texturiert

Kreuzungen sind jedoch ein sehr kompliziertes Gebiet in der automatischen Erkennung und Segmentierung, wenn die Kreuzung in einem einzelnen Polygon enthalten ist. Die Wahrscheinlichkeit für solche Polygone ist jedoch gering.

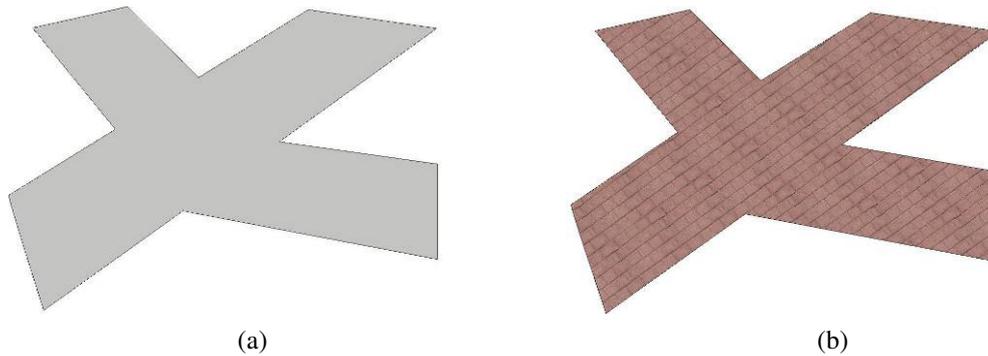


Abbildung 4.12: (a) Eine Kreuzung bestehend aus einem einzelnen Polygon,  
 (b) Die Kreuzung erhält über die gesamte Fläche nur eine Ausrichtung der  
 Textur

Auch hier ist die Unkenntnis über den gewählten Startpunkt nachteilig. Bei gleichmäßiger Punkteverteilung in den Kreuzungspolygonen und Kenntnis des Startpunktes, kann das Polygon gleichzeitig in beide Richtungen abgelaufen werden. Wird in einer der beiden Richtungen eine Abzweigung gefunden, kann geprüft werden, ob in der anderen Richtung ebenfalls eine Abzweigung vorhanden ist. Wenn ja, kommen die Kreuzungstypen X oder T in Frage. Hier zeigt sich ein weiteres Problem: Wie sollen die einzelnen Kreuzungstypen unterschieden werden? Darauf aufbauend stellt sich die Frage, welche der Polygone durchgehend und welche als abzweigende Straße texturiert werden (vgl. Abb. 4.14). Welche Polygone müssen zu welchen Gruppen zusammengefasst werden?

Diese Fragen sind automatisch sehr schwer zu beantworten. Es ist vorteilhafter, diese Regionen separat und manuell zu texturieren. Dafür sind drei Varianten möglich:

1. Die Kreuzungspolygone werden vom Anwender ausgewählt und manuell mit bereitgestellten Werkzeugen texturiert.
2. Der Anwender wählt eine Kreuzungsschablone aus, die der Form der Kreuzung im Straßenpolygon entspricht. Die Schablone wird durch Parameter an die Kreuzung angepasst. Das gerenderte Bild der Schablone funktioniert als Textur, die jedoch zusätzlichen Speicherplatz bedarf. Je nach Komplexität der Kreuzung, kann diese Variante von Vorteil sein.
3. Es wird eine eigene Textur für die Kreuzungspolygone berechnet. Dies kann mit Hilfe von prozeduralen Texturen erfolgen. Das Ergebnis ist eine passgenaue Textur, die wenig zusätzlichen Speicherplatz bedarf. Der Aufwand zur Erstellung richtet sich nach der Komplexität des Polygons und der gewünschten Textur.

#### 4.4.5. Manuelle Korrekturmöglichkeiten für die Texturierung

Manuelle Korrektur- bzw. Manipulationswerkzeuge können zur Verbesserung der automatisch erstellten Texturierung genutzt werden. Bei besonderen Formen können immer ungünstige Texturierungen entstehen. Dazu zählt das in Abb. 4.15 und 4.16 abgebildete Beispielpolygon. Das Ziel ist eine gleichmäßige Texturierung dieses Polygons mit einer senkrechten Orientierung.

##### Beispiel 1

Ausgehend vom rot markierten Startpunkt und einer Umlaufrichtung im Uhrzeigersinn wird direkt beim zweiten Punkt eine Krümmung erkannt. Der blau eingefärbte Teil würde als erste Sektion erkannt werden. Anschließend erfolgen wiederum zwei Krümmungserkennungen mit entsprechender Sektionsbildung. Die Abschnitte mit den Farben grün und grau erhalten dadurch eine eigene Texturorientierung. Zuletzt entsteht die gelbe Sektion. An den eingezeichneten Texturorientierungen, welche durch die roten Pfeile symbolisiert sind, ist erkennbar, dass keine einheitliche Textur entsteht.

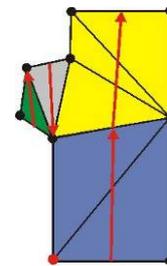


Abbildung 4.13:  
Beispiel 1

##### Beispiel 2

Erfolgt die Abtastung des Polygons im Gegenuhrzeigersinn, beginnend mit dem rechten unteren Punkt, zeigt sich ein anderes Ergebnis. Bis zur ersten Winkeländerung am oberen rechten Punkt werden alle angrenzenden Dreiecke zu einer Sektion (blau) zusammengefasst. Der Bereich der Ausbuchtung wird anschließend als eine separate Sektion zusammengefasst. An diesem Beispiel kann gut das Problem der parallelen Polygone (Problem Nr. 3 aus Abschn. 4.4.4.3) verdeutlicht werden. Durch das Nichtvorhandensein von Verbindungen der Ausbuchtungspunkte mit den rechten Polygonpunkten, werden die Dreiecke der Ausbuchtung bei der Sektionsbildung nicht berücksichtigt.

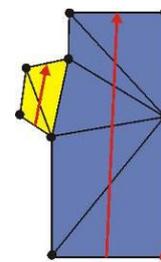


Abbildung  
4.13:  
Beispiel 2

Das Ergebnis in Beispiel 2 ist zufrieden stellend, für Fälle wie in Beispiel 1 ist eine **Änderung der Geometrie** sinnvoll. Je nach „Belieben“ des Anwenders können Triangulierungen geändert oder sogar Punkte verschoben, gelöscht und eingefügt werden.

Im ersten Beispiel kann ein zusätzlicher Punkt mittig in die obere Ausbuchtungskante eingefügt werden. Die Ausbuchtung wird als solche erkannt und alle Dreiecke werden zu einem Segment zusammengefasst und einheitlich texturiert.

Um mit derartigen Änderungen gezielt zu einem besseren Ergebnis zu gelangen, sind Kenntnisse des Algorithmus und einer ausgeglichenen Triangulierung von Polygonen notwendig. Diese können nicht bei allen potentiellen Anwendern vorausgesetzt werden. Daher sollte diese Funktion nur durch Freischalten erreichbar sein, sodass der Anwender selbst entscheiden kann, ob diese Funktion zur Verfügung steht.

Soll die Geometrie nicht verändert werden, bleibt das Korrigieren der Textur auf den einzelnen Polygonen. Die Textur kann dazu (nach Selektion der Polygone) gedreht und verschoben werden.

Diese Änderungen können als **Eingabe eines Wertes** in einem Dialog erfolgen. Neben den Eingabefeldern sollten zusätzlich Arrow-Keys zur Änderung der Werte zur Verfügung stehen.

Eine andere Möglichkeit ist die **Interaktion in der 3D-Szene**. Das Rotieren und Verschieben von Texturen lässt sich durch Hilfsobjekte (angelehnt an CAD-Programme) leicht darstellen. Da innerhalb der vorliegenden Arbeit ausschließlich Polygone in einer Ebene betrachtet werden, genügt eine zweidimensionale Darstellung der Hilfsobjekte. Der Umgang mit ihnen ist leicht erlernbar und den potentiellen Anwendern mitunter bereits aus GI-Systemen bekannt.

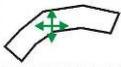
Modus	Button	Hilfsobjekte
Angeben der Orientierung		
Verschieben		
Drehen		
Navigation		

Abbildung 4.14: Übersicht der Modi zum manuellen Anpassen der Textur, den entsprechenden Buttons zum Aktivieren und Deaktivieren der Modi. In der letzten Spalte sind die Hilfsobjekte (grün) dargestellt, welche die Anwender unterstützen sollen.

Um das Rotieren und Verschieben zu ermöglichen, muss vom Anwender der entsprechende Modus aktiviert werden. Dies kann über Buttons im Menü geschehen. Um dem Anwender ein wiederholtes Öffnen des Menüs zu ersparen, können die Buttons (s. Abb. 4.17) auf einem nicht modularen Dialog platziert werden. Ein nicht modularer Dialog muss nicht geschlossen werden, um eine Aktion auszuführen.

Die Orientierungsangabe der Textur lässt sich ebenfalls durch Einzeichnen eines Richtungspfeils mit der Maus angeben. Dabei wird nach Aktivierung des entsprechenden Modus die Mausbewegung aufgezeichnet. Die Richtung kann aus den Start- und Endpunkten abgelesen werden. Soll eine genauere Verfolgung der Maus erfolgen, können Zwischenpunkte zur Approximation einer Funktion genutzt werden. Die Textur kann so an den Punkten des Pfades ausgerichtet werden. Dies ist im Wesentlichen für sehr „schwierige“ Polygone von Vorteil.

Um eine Akzeptanz des Tools bei einer Vielzahl von Anwendern zu erreichen, sollten verschiedene Möglichkeiten zur Durchführung von Änderungen angeboten werden. Jeder Anwender kann eigene Vorlieben haben, die oft durch bereits verwendete Systeme hervorgerufen werden.

Die Änderung der Parameter mit einer Variante eines Werkzeugs sollte die anderen Varianten beeinflussen. Das heißt, wird die Textur mit Hilfe des Hilfsobjektes gedreht, muss sich der Wert im zugehörigen Eingabefeld anpassen. Wie diese Funktionen in ein User Interface integriert werden können, wird in Abschnitt 4.5. beschrieben.

#### 4.4.6. Krümmungen des Straßenverlaufs in der Realität

Die Betrachtung einer Stadtkarte offenbart recht eindeutig, dass Krümmungen im Straßenverlauf in einer verhältnismäßig großen Zahl vorhanden sind (s. Abb. 4.18). Die so entstehende Vielfalt der Straßenführung in der Realität führt in der Texturierung von 3D-Modellen zu Schwierigkeiten und erhöhtem Aufwand.

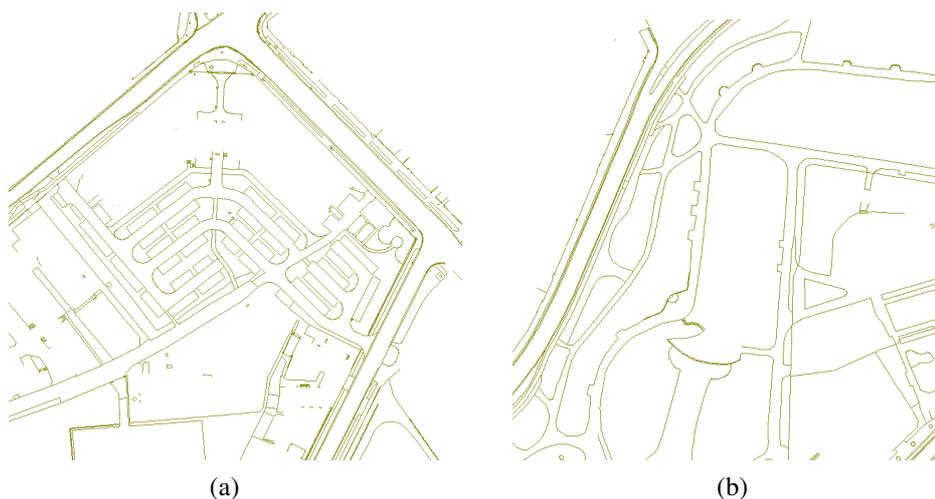


Abbildung 4.15: Kleine Ausschnitte des Straßennetzes der Lutherstadt Wittenberg, dargestellt im ArcView GIS 3.2. In beiden Bildern sind relativ viele Strecken mit Krümmungen zu sehen. In (b) wird die Bedeutung der Erkennung von Ausbuchtungen klar.

Bei einem aufmerksamen Spaziergang durch die Stadt, bei dem der Pflasterung der Wege besondere Beachtung geschenkt wird, fällt auf, dass bei Krümmungen im Straßenverlauf Kompromisse eingegangen werden. Auf den realen Gehwegen kommt es häufig zu den so genannten harten Bruchkanten. Ein Weg mit einer Krümmung wird in mehrere Abschnitte unterteilt, in denen sich die Pflasterrichtung immer ein paar Grad von der vorherigen unterscheidet (s. Abb. 4.19). Die Länge der Unterteilungen ist abhängig von der Stärke der Krümmung.



Abbildung 4.16: Gepflasterter Fußweg mit mehrfacher Änderung der Pflasterrichtung an Bruchkanten (rot: alte Richtung, blau: neue Richtung)

#### 4.4.7. Weitere Möglichkeiten zur Texturierung von Krümmungen

Das Problem der Texturierung von Krümmungen sind die entstehenden Verzerrungen, wenn ein Texturbild gekachelt „um die Kurve herum“ gelegt werden soll. Da der innere Kurvenradius kürzer ist als der äußere, kommt es zu Streckungen oder Stauchungen der Textur, um keine Lücken entstehen zu lassen. Erfahrungen und Gespräche mit Anwendern der Visualisierungsplattform des IFF haben zeigt, dass bis zu einem gewissen Grad der Krümmung und damit der Verzerrung, dies von vielen Betrachtern akzeptiert wird. Ist die Verzerrung jedoch zu stark, erweckt sie einen unansehnlichen Eindruck. Je nach Art der dargestellten Oberfläche oder Materialien, kann ein unterschiedlicher Grad der Verzerrung akzeptiert werden. Verzerrungen in regelmäßigen Strukturen (z.B. Pflaster) werden stärker wahrgenommen als in stochastischen (z.B. Erde).

Um eine Verzerrung zu vermeiden und gleichzeitig ohne abrupte Änderungen der Orientierung auszukommen, sind neben der betrachteten Projektion der Texturen unter der Berücksichtigung der Objektausrichtung weitere Techniken nötig.

Zum einen können **prozedurale Texturen** verwendet werden. Die benötigten Parameter zur Anpassung der Funktion an die Krümmung können aus den Sektionspunkten gewonnen werden.

Es ist ebenfalls möglich, mit einzelnen Elementen einer Textur, beispielsweise eines einzelnen Pflastersteins, zu arbeiten. Eine Verzerrung der Textur kann vermieden werden, indem je nach Krümmung entsprechend viele Steine hinzugefügt werden. Um die Position der **Texturbausteine** zu berechnen, muss der Grad der Krümmung und Ausdehnung der Sektion bekannt sein. Diese Parameter lassen sich ebenfalls aus den Sektionspunkten berechnen.

Für diese Variante ist es vorteilhaft, wenn die Geometrien als geometrische Kurven vorliegen. Als Standard-Ausgangsdaten ist dies jedoch nicht zu erwarten.

Diese Art der Texturierung würde der realen Verfahrensweise zum Pflastern der Straße nahe kommen.

Ist eine **maßstäbliche Fotografie** der gekrümmten Straße vorhanden, kann diese ebenfalls nach entsprechender Bearbeitung (vgl. Abschn. 2.5.2.1.) als Textur verwendet werden (s. Abb. 4.20). In vielen Fällen kann eine solche Textur nur an einer Stelle eingesetzt werden. Auf alle Krümmungen im Stadtgebiet bezogen, ergibt sich im Vergleich zu den oben genannten Möglichkeiten ein ungleich höherer Aufwand und Speicherplatzbedarf.



Abbildung 4.17: Textur eines gebogenen Bordsteins

#### 4.4.8. Detaillierung der Straßengeometrie

Das Einfügen zusätzlicher Geometrien kann zur Erhöhung des LOD der Straßengeometrien eingesetzt werden. Zu diesen Geometrien zählen Bordsteine, Gehwege, Radwege, aber auch die Unterteilung der Fahrbahn in einzelne Fahrstreifen. In den Geometriedaten ist eine solche Differenzierung der Straßengeometrien meist nicht vorhanden. Nachfolgend soll das automatische Einfügen dieser Geometrien am Beispiel des Bordsteins betrachtet werden.

Unter einem **Bordstein** wird eine erhöhte Kante aus Gestein verstanden. Sie werden im Straßenbau eingesetzt, um Verkehrsflächen zu begrenzen.

#### Voraussetzungen

Bordsteine sind in vielen 3D-Stadtmodellen nicht integriert, da die entsprechenden Geometriedaten sehr selten vorhanden sind. Doch können sie die realistische Wirkung eines digitalen 3D-Modells erheblich steigern. Besonders dann, wenn bei der Betrachtung oder Präsentation der 3D-Modelle zahlreiche Standorte in Fußgängerhöhe gewählt werden. Der visuelle Eindruck ändert sich stark, wenn statt der ebenen Straße, die Fahrbahn abgesenkt und von Bordsteinen umrandet ist.

In Abhängigkeit von der vorhandenen Geometrie muss die Bordsteingeometrie innerhalb oder außerhalb der Straßengeometrie eingefügt werden. Es müssen ebenfalls Kenntnisse über die Zusammensetzung der vorliegenden Straßengeometrien vorhanden sein. So kann die Straßengeometrie zum einen die gesamte Straße (Fahrbahn und Gehweg) oder nur die Fahrbahn enthalten.

## Vorüberlegungen

Um das korrekte Einfügen der Bordsteingeometrie zu gewährleisten, sollte dieser Vorgang nicht in den automatischen Prozess eingegliedert, sondern dem Nutzer als Funktion angeboten werden. Das heißt, diese Funktion kann nach Eingabe einiger Parameter vom Nutzer gestartet werden, die Erzeugung des Bordsteins erfolgt anschließend automatisch.

Folgende Angaben sind sinnvoll:

- Was ist vermessen? Fahrbahn - Fahrbahn und Gehweg - Gehweg
- Wo soll die Geometrie entstehen? innerhalb - außerhalb
- Wie breit soll der Bordstein werden?
- Welche Höhe soll der Bordstein bekommen?

Um diese Angaben an das System übermitteln zu können, ist die Eingabe der Werte in ein User Interface notwendig. Mögliche Eingabefelder sind:

- Radio-Buttons zur Auswahl von:
  - Position des Bordsteins relativ zur Straßengeometrie mit der Auswahl „innen“ oder „außen“
- Eingabefelder zur Festlegung von:
  - Breite des Bordsteins
  - Höhe des Bordsteins
  - Breite der Textur
  - Höhe der Textur
- Auswahlmöglichkeit der Textur über einen Datei-Öffnen-Button

Je nach ALK-Typschlüssel des Polygons kann die Bordstein-Erstellung vom System aktiviert oder deaktiviert werden. Flächen vom Typ „Ackerland“ benötigen in der Regel keine Umrandung dieser Art. In solchen Fällen müssen die Eingabefelder deaktiviert werden.

Die Gestaltung eines solchen Dialogs wird in Abschnitt 4.5. genauer betrachtet.

Für das Einfügen der Bordsteingeometrien ergeben sich zwei verschiedene Varianten:

1. Es wird nur die Straße abgesenkt und die entstehenden Flächen mit einer Textur versehen (s. Abb. 4.23 a). Nachteilig ist, dass die Bordsteintextur nur an der Innenseite des Bordsteins zu sehen ist, nicht aber oben.
2. Zusätzlich zum Absenken der Straße wird eine neue Bordsteingeometrie eingefügt, welche die Straßenzüge umschließt und eine realistischere Texturierung ermöglicht (s. Abb. 4.21 b).

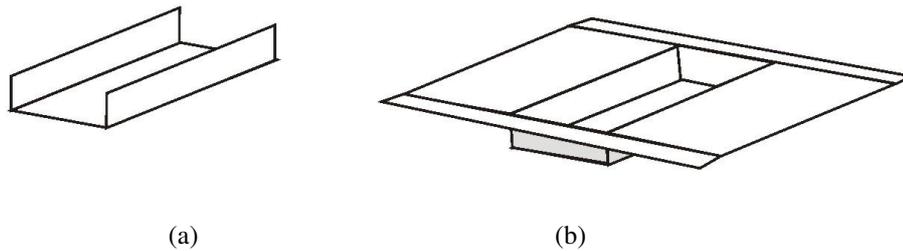


Abbildung 4.18: Abgesenkter Straßenzug mit Bordstein der Variante 1 (a) und Variante 2 (b)

Welche Abschnitte einer Straße (Fahrbahn, Gehweg) die Geometrie beinhaltet, hängt von den jeweiligen Vorgaben bei der Vermessung ab. Für die Zusammensetzung der Straßengeometrien ergeben sich folgende Möglichkeiten:

1. Die vermessene Straßengeometrie entspricht ausschließlich der Fahrbahn.
2. Fahrbahn und Gehweg bilden die vermessene Straßengeometrie.
3. Die vermessene Straßengeometrie stellt ausschließlich den Gehweg dar.
4. Die Geometrie entstand automatisch aus den Zwischenräumen von Grundstücken oder Gebäuden.

Folgende Vorgehensweisen ergeben sich aus den oben genannten Möglichkeiten:

- zu 1. Bordstein und Gehweg werden außerhalb eingefügt.  
Hierfür sollte unter normalen Umständen ausreichend Platz zwischen der Fahrbahn und den Gebäudegrundrissen vorhanden sein.
- zu 2. Gehweg und Bordstein werden innerhalb eingefügt  
Es sollte auf ein gutes Größenverhältnis zwischen der übrig bleibenden Fläche der Fahrbahn und dem eingefügten Gehweg inkl. Bordstein geachtet werden.
- zu 3. Der Bordstein wird innerhalb der Gehweggeometrie auf der Fahrbahnseite eingefügt.  
Auch hier sollte auf ein gutes Verhältnis zwischen Gehweg- und Bordsteinbreite geachtet werden.
- zu 4. Inhaltlich entspricht dies der Möglichkeit Nr. 2, deshalb werden hier ebenfalls die Bordsteine und Gehwege an der Innenseite der Straßengeometrien angebracht.

Vorteilhaft ist es, wenn nicht mehrere dieser beschriebenen Varianten in den Daten aus dem GI-System vertreten sind. Andernfalls ist eine automatische oder manuelle Kennzeichnung der Geometrien erforderlich.

## Berechnung

Ist bekannt wo die Geometrie, relativ zur Ausgangsgeometrie generiert werden soll, kann die Berechnung erfolgen. Dazu wird auch hier die Randgeometrie (bzw. Border) berechnet. Damit ist bekannt, ob die Randgeometrie mit (UZS) oder entgegen (GUZS) dem Uhrzeigersinn abgelaufen wird.

Anschließend kann die Position der neuen Punkte ermittelt werden. In die Berechnung fließt die Winkelhalbierende am aktuellen Punkt, die vom Anwender angegebene Breite des Bordsteins und die Position des aktuellen Punktes ein. Um die gesamte Straßengeometrie entsteht ein Bordstein.

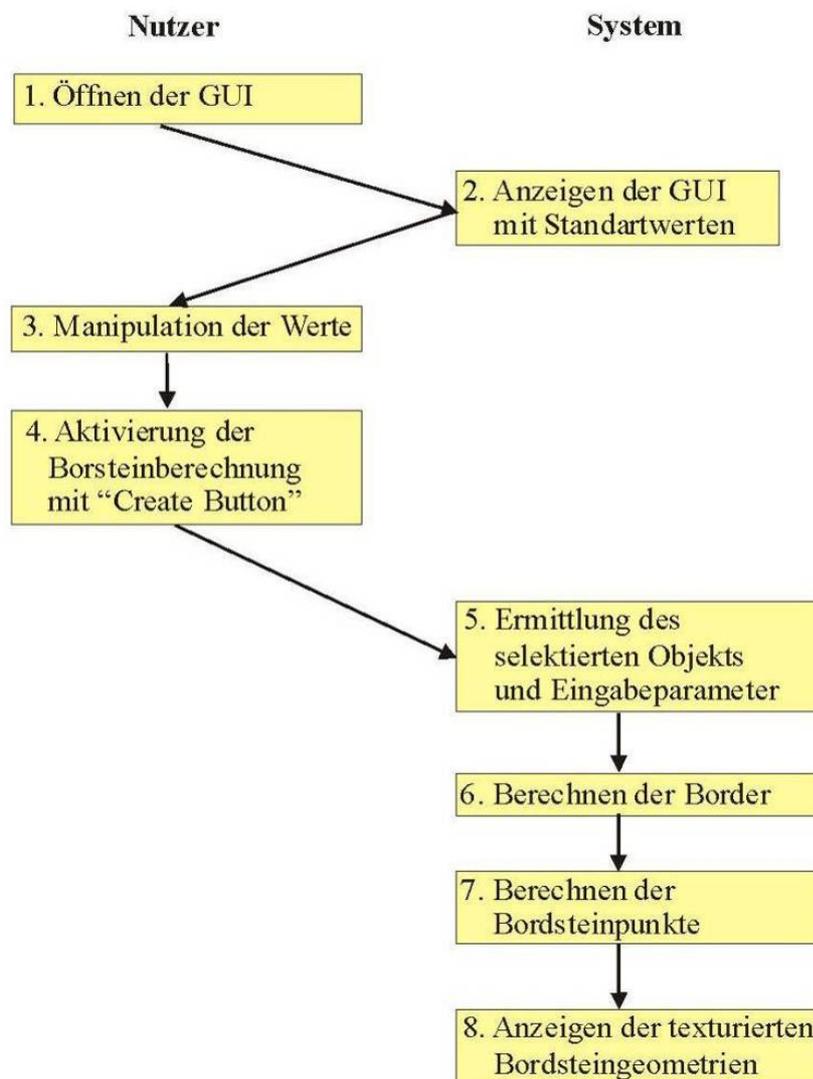


Abbildung 4.19: Zusammenfassende Darstellung des Ablaufs der Bordsteingenerierung

## Probleme

Wird ein zu breiter Gehweg außerhalb der Fahrbahngeometrie generiert, d.h. der reale Gehweg ist an einer Stelle schmaler, kann es dazu führen, dass sich Gehweggeometrie und Gebäudegeometrien überlappen. Da die Gebäudegeometrie aufgrund der Höhe die Gehweggeometrie überdeckt, verschwindet der Gehweg unter dem Gebäude. Auch bei angrenzenden Bodengeometrien kann es zu Überschneidungen kommen. Um zu verhindern, dass neu generierte Flächen die vorhandenen Originalflächen überdecken, sollte die Festlegung getroffen werden, dass die originalen Flächen vorrangig gezeichnet werden.

Eine weitere Schwierigkeit stellen auch hier Abzweigungen und Kreuzungen dar. An diesen Stellen, sprich an Treffpunkten von Straßen, sind real keine Bordsteine vorhanden und es kann zu Überlappungen kommen (von abgesenkten Bordsteinen wird abgesehen).

Soll dies im automatischen Prozess berücksichtigt werden, muss für jeden Punkt kontrolliert werden, ob dieser zusätzlich von einer anderen Straßengeometrie genutzt wird, bzw. seine Koordinaten mit einem Punkt eines anderen Polygons übereinstimmen.

Um den Aufwand zu verkleinern ist es von Vorteil, wenn in der Geometrie bereits entsprechende Informationen, ähnlich den nachfolgend aufgezählten, gespeichert sind.

- Jede Straßengeometrie hat Informationen gespeichert, welche anderen Straßengeometrien an sie grenzen, sprich mit welchen anderen Polygonen Punkte geteilt werden.
- Im Idealfall hat jeder Punkt die Informationen, zu welchen Geometrien er gehört bzw. ob es andere Punkte mit den gleichen Koordinaten gibt.

Mit diesen Informationen können die Bordsteine an den Stellen, an denen sich Polygone treffen, automatisch ausgelassen werden. Die Bereitstellung dieser Informationen erfordert eine Vorverarbeitung der Daten.

Sind diese Informationen nicht vorhanden, können die Kanten nachträglich manuell ausgewählt und gelöscht werden. Dafür muss die Voraussetzung erfüllt sein, dass mehr als ein Punkt an diesen Ecken existieren, da sonst unter Umständen zu große oder zu kleine Flächen der Bordsteingeometrie gelöscht werden (vgl. Abb. 4.23).

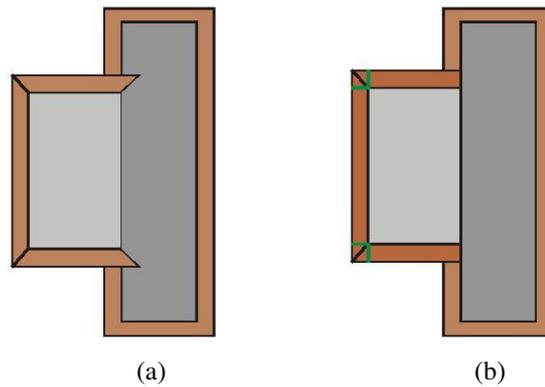


Abbildung 4.20: Fehlerhafte Ergebnisse nach manuellem Löschen der Bordsteinkanten in (a). Im dargestellten Fall ist zu viel des Bordsteins entfernt worden. In (b) sind die zusätzlichen Kanten (grün) enthalten, die ein angepasstes Entfernen der Kanten ermöglichen.

Es müssen zusätzliche Geometriepunkte generiert werden. Dies geschieht, wenn zwischen den Borderkanten des betrachteten Punktes ein Winkel von  $90^\circ$  liegt. Da in vielen Fällen von geraden Abschlusskanten der Polygone ausgegangen werden kann, sind dies oft Bereiche, in denen andere Polygone anschließen. Rechtwinklige Ecken von Polygonen können konvex oder konkav sein. Über das Vorzeichen der Winkel lässt sich die konvexe oder konkave Ausrichtung bestimmen. Dabei ist die Umlaufrichtung der Polygone entscheidend, da an derselben Ecke einmal positive und einmal negative Winkel berechnet werden (vgl. Abb. 4.24).

Konkave Ecken werden wie die sonstigen Krümmungen behandelt, an konvexen  $90^\circ$ -Ecken erfolgt die Berechnung der zusätzlichen Punkte.

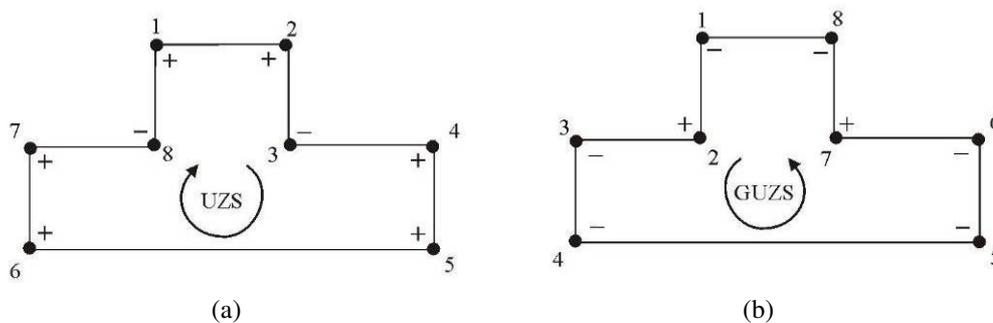


Abbildung 4.21: Unterschied zwischen den Vorzeichen der Winkel entsprechend der Umlaufrichtung. Im Uhrzeigersinn (a) erhalten die konvexen Winkel positive Vorzeichen, im Gegenuhrzeigersinn (b) sind die Vorzeichen negativ.

Die Gestaltungsmöglichkeiten von Straßenpolygonen können, wie zu Beginn des Abschnitts erwähnt ebenso innerhalb der Fahrbahn erweitert werden. Es kann eine Unterteilung in einzelne Fahrstreifen erfolgen. Je nach Grad der gewünschten Detaillierung können unter anderem Geometrien für mehrere Fahrstreifen, Fahrbahnmarkierungen, Mittelstreifen und Randsteinreihen entstehen (s. Abb. 4.25). Jede Geometrie kann anschließend eine eigene Textur erhalten. Die Berechnung von Mittelstreifen erfolgt analog zur Berechnung von Bordsteinen. Alle anderen genannten Differenzierungen lassen sich ohne eine Absenkung oder Erhöhung erstellen. Bei der Bestimmung der Breite muss auf ein gutes Verhältnis

zur Breite der Straßengeometrie geachtet werden, um keine Überdimensionierung einzelner Elemente zu erhalten.

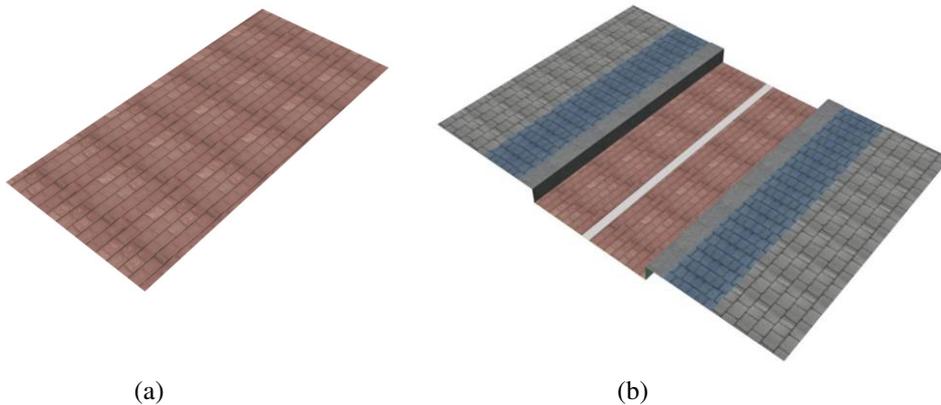


Abbildung 4.22: (a) Ausgangsgeometrie, (b) Detaillierte Geometrie

Ohne eine Unterteilung der Geometrie ist eine detaillierte Texturierung der Fahrbahn nur schwer möglich. Im Normalfall werden die maßstäblichen Texturen gekachelt. Um Strukturen wie eine Fahrbahntrennung zu integrieren, muss diese auf der Textur vorhanden sein. Durch die Kachelung, führt dies jedoch zu unerwünschten Effekten. Eine Möglichkeit, ist die Verwendung einer Textur, welche die gesamte Fahrbahnbreite abdeckt. Die fotografische Aufnahme einer solchen Textur ist unter der Beachtung der Anforderungen zur Aufnahme von Texturfotografien (s. Abschn. 2.5.2.) erschwert. Als Alternative kann das Multitexturing eingesetzt werden, um Fahrbahnmarkierungen darzustellen.

Unbedingt zu beachten ist, dass bei der Erstellung dieser Erweiterungen der Ausgangsdatensatz, z.B. durch das Einfügen von nicht vermessenen Punkten, verändert wird. Diese Veränderung sollte dem Nutzer im Voraus mitgeteilt werden. Gleichzeitig sollte der Vorschlag zur Speicherung der veränderten Daten in einer neuen Datei unterbreitet werden.

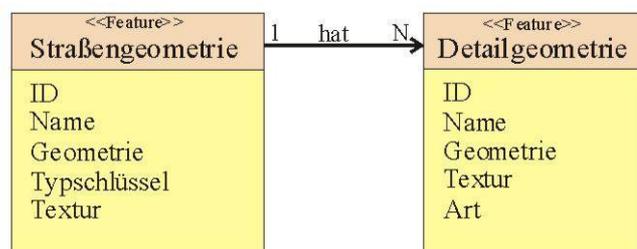


Abbildung 4.23: UML-Darstellung der Beziehung zwischen gegebenen Straßengeometrien und neu erstellten Detailgeometrien

## 4.5. Dialoggestaltung

In diesem Abschnitt wird der Entwurf eines Dialogs zu Steuerung der Texturierung und des Einfügens einer Detailgeometrie erläutert. Zu Beginn werden allgemeine Prinzipien [Pre99] der Dialoggestaltung erläutert, welche hilfreich sind, um die Bedienung eines Benutzerdialogs intuitiv zu gestalten.

### 4.5.1. Prinzipien Dialogentwurf

Der Entwurf eines Dialoges erfordert nach Preim [Pre99] die Berücksichtigung von wichtigen Prinzipien.

- Die visuelle Gestaltung von Dialogen sollte hinsichtlich der Dimensionierung und Positionierung von Dialogelementen, wie Buttons, Eingabefelder, Listen und Radiobuttons, konsistent sein.
- Die Reihenfolge von Aktionen sollte sich an den Arbeitsabläufen der Benutzer orientieren.
- In Eingabefeldern und bei Auswahlbuttons sollten Standardwerte voreingestellt und angezeigt werden, um die Benutzung des Dialogs effizient zu gestalten.
- Es sollte möglich sein, bestimmte Aufgaben komplett innerhalb eines Dialoges zu bewältigen. Sind sehr viele Eingaben notwendig, können Dialoge mit Eingaben zu Teilgebieten geöffnet werden.

Die Prinzipien können in visuelle und dynamische Aspekte des Dialogentwurfs untergliedert werden. Die visuellen Aspekte betreffen unter anderem die Anordnung, Größe, farbliche Gestaltung und Beschriftung, somit die räumliche Gliederung eines Dialogs.

Zu den dynamischen Aspekten zählt die Reihenfolge von Interaktionen, notwendige Restriktionen und welche Möglichkeiten für den Nutzer in welchen Zuständen möglich oder nicht erreichbar sind.

Ausführliche Informationen zur Gestaltung und Umsetzung von Benutzerschnittstellen sind unter anderem in [Pre99] und [Shn02] zu finden.

### 4.5.2. Entwurf

Die Benutzerschnittstelle soll den Anwender bei der Texturierung von Straßenpolygonen und Generierung von Detailgeometrien unterstützen. In diesem Konzept wird ausschließlich die Erstellung eines Bordsteins berücksichtigt. Weitere Detailgeometrien können in ähnlicher Weise integriert werden. Die Notwendigkeit der nachfolgend beschriebenen Optionen wurde bereits in den Abschnitten 4.4.5. und 4.4.8. dargelegt.

Der zu entwickelnde Benutzerdialog ist ein nicht-modaler Dialog, d.h. der Dialog braucht nicht geschlossen zu werden, damit Aktionen ausgeführt werden können.

Es muss die Möglichkeit bestehen, nacheinander Polygone zu bearbeiten, ohne ein unterbrechendes Schließen und Öffnen des Dialogs zu erzwingen.

Inhaltlich soll sich der Dialog in drei Bereiche untergliedern. Der erste Bereich enthält Einstellungen für die Texturierung der ausgewählten Geometrieobjekte. Die Einstellungen und Auswahlmöglichkeiten zur Erstellung der Detailgeometrien sind im zweiten Bereich zu finden. Da beide Bereiche Texturen enthalten, ist eine deutliche Trennung zur Vermeidung von Missverständnissen erforderlich. Im dritten Bereich soll eine Vorschau der Textur auf Grundlage der eingestellten Parameter möglich sein.

Im ersten Bereich des Dialogs befinden sich die Eingabefelder und Auswahlelemente zur Texturierung des ausgewählten Polygons. Der Nutzer soll die Textur auswählen und eine eventuelle Verschiebung und Drehung angeben können. Die Drehung bezieht sich ausschließlich auf die Textur und hat an dieser Stelle nichts mit dem Winkel der *bounding box* gemein. Weiterhin muss es eine Auswahl geben, mit welcher der Nutzer entscheiden kann, ob die Textur einfach gekachelt, oder wie in Abschnitt 4.4.4.4. beschrieben, in Richtung der aktuellen Objektausrichtung gemappt werden soll. Entscheidet sich der Anwender für die zweite Variante, muss ein zusätzliches Feld aktiviert werden, in dem die Gradzahl des Orientierungswinkels geändert werden kann. Eine Vorschau der Textur ist ebenfalls sinnvoll. Die Texturierung kann durch Betätigen eines Aktionsbuttons ausgelöst werden.

In allen Eingabefeldern, mit Ausnahme der Texturpfade, sollen Standardwerte enthalten sein, sodass ohne Änderung der Werte texturiert werden kann.

Im zweiten Bereich des Dialogs hat der Nutzer die Möglichkeit, Einstellungen zu der Bordsteingeometrie vorzunehmen. Es müssen die Auswahl der Textur und die Angaben zu Breite, Höhe und Winkel vorhanden sein. Ein Textur-Vorschaufenster ist sinnvoll. Weiterhin müssen Angaben zur Ausrichtung der Bordsteingeometrie getroffen werden. Dazu zählen Radiobuttons zur Auswahl der Lage des Bordsteins (innerhalb oder außerhalb der Geometrie). und Elemente zur Angabe der Höhe und Breite des Bordsteins. Das Auslösen der Bordsteinberechnung erfolgt ebenfalls mit Hilfe eines Aktionsbuttons.

Auch hier sind alle Auswahl- und Eingabemöglichkeiten mit Standardwerten belegt.

Soll die Generierung weiterer Detailgeometrien angeboten werden, vergrößert sich der Dialog rapide. Um diesen dennoch übersichtlich zu gestalten, müssen die zusätzlichen Informationen gut strukturiert werden. Eine Möglichkeit ist das Nutzen von spezialisierten Dialogen. Diese lassen sich aus dem Hauptdialog öffnen und enthalten die speziellen Eingabemasken. Eine zweite Möglichkeit ist das Verwenden von ausklappbaren Menüeinträgen. Als dritte Variante ist die Verwendung der gleichen Eingabefelder für alle Detailgeometrien zu nennen. Dann erfolgt die Auswahl der Art der Geometrie in einer so genannten Listbox. Mit dieser Variante kann lediglich die Generierung einer Detailgeometrie pro Berechnung erfolgen. Die beiden zuvor genannten Varianten erlauben die Eingabe von Parametern mehrerer Geometrien.

Die Preview, oder auch „Vorschaufenster“, dient der Darstellung des ausgewählten Polygons. In dem Fenster sollen Buttons enthalten sein, mit denen sich verschiedene Modi starten lassen. Zur Verdeutlichung der Funktion müssen allen Buttons spezielle und eindeutige Symbole zugeordnet werden.

Modus 1 kann das Festlegen der Texturausrichtung mittels eines Pfeils ermöglichen, das Buttonsymbol ist entsprechend ein „Pfeil“. Das Verschieben der Textur auf dem Polygon wird durch Modus 2, das Drehen durch Modus 3 erlaubt. Die Funktionen könnten durch ein „Kreuz“ und einen „runden Pfeil“ symbolisiert werden. Modus 4, versinnbildlicht durch eine „Hand“, führt wieder in die normale Navigation zurück.

Die hier vorgeschlagenen Symbole werden in dieser Funktion bereits in anderen Systemen gebraucht, sodass von einer schnellen Akzeptanz ausgegangen werden kann.

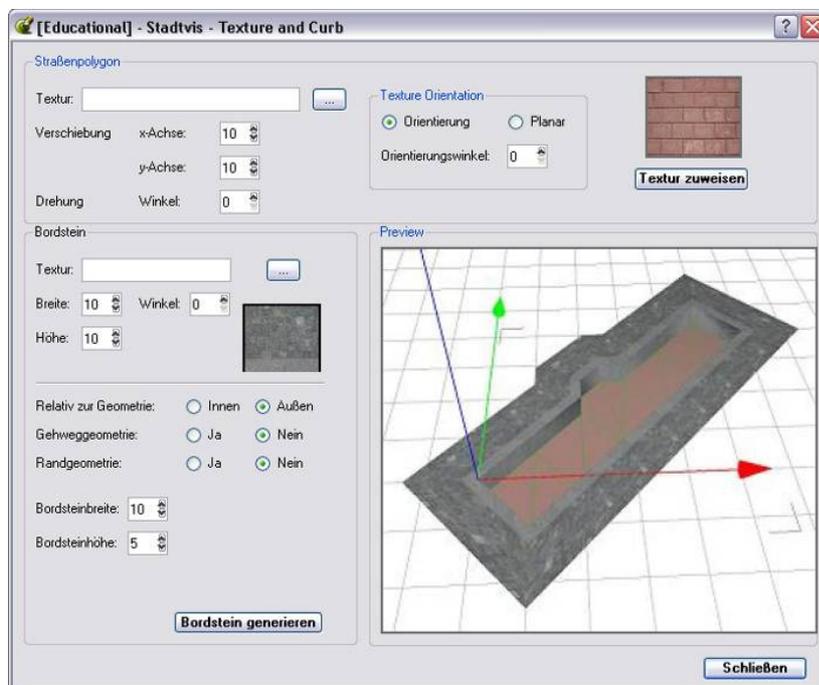


Abbildung 4.24: Vorläufiger Entwurf des Dialogs zur Steuerung der Texturierung und zum Anfügen eines Bordsteins mit Hilfe des Qt-Designers

„Wichtige“ Aktionen, wie das Texturieren, sollten über *shortcuts* ausgeführt werden können. Dadurch sollen „Experten durch regelmäßige Anwendung“ (vgl. Benutzerprofile in [Shn02]) schnellere Arbeitsprozesse ermöglicht werden. Die vom Anwender zuletzt eingestellten Angaben müssen hierfür als Standardeinstellungen übernommen werden.

## 4.6. Qualitätsanalyse von Texturen

In der 3D-Stadtvisualisierung ist ein Tool zur Qualitätsanalyse von verwendeten Texturen hilfreich. Dabei kann die Qualität hinsichtlich der Bildschärfe im 3D-Stadtmodell automatisch analysiert werden. „Einsatzgebiet“ eines solchen Tools sind die Fassadentexturen von Gebäuden. Da Grund- und Bodengeometrien in der Regel gekachelt werden, ist eine Qualitätsminderung des optischen Eindrucks zwischen Ursprungsbild und Ansicht im 3D-Modell hinsichtlich der Schärfe nicht zu erwarten. Hier kann eine Analyse auf glatte Übergänge an den Bildrändern der Kacheln erfolgen, um zu beurteilen, ob die Textur gekachelt werden soll.

Im nächsten Abschnitt wird eine Methode zur Bewertung der Bildschärfe beschrieben, die innerhalb der Diplomarbeit von Bernd Höntzsch entwickelt und umgesetzt wurde. Im Anschluss erfolgt die Erläuterung einer Erweiterung für eine Anwendung im Bereich der 3D-Stadtmodelle.

In Abschnitt 4.6.3 wird eine Möglichkeit vorgestellt, die Eignung von Texturen für die Kachelung zu beurteilen.

### 4.6.1. Betrachtung von Höntzsch

Die Diplomarbeit von Höntzsch [Hön05] beschäftigt sich mit der „Schärfbestimmung in digitalen Fotos“. Die Schärfekontrolle wurde in dieser Untersuchung für zwei Anwendungsgebiete entwickelt, das Testen der maximalen Bildschärfe a) von Kameraobjektiven und b) beim Einsatz mit einem so genannten Wolfenbüttler Buchspiegel<sup>9</sup>. Ziel der in der Arbeit entwickelten Methode ist die Beurteilung und das Vergleichen der Schärfe eines planaren Objektes in mehreren vorliegenden Bildern, um das schärfste Bild zu identifizieren.

Die dort entwickelte Methode filtert das vorliegende Bild mit einem Kantenfilter. An Stellen mit schwachen Kanten, Kontrasten und Helligkeitsverläufen entstanden entsprechend der Definition von Kantenfiltern [RW97] dunkle Bildwerte. Helle Bildwerte ergeben sich an Positionen starker Kanten. Der maximale Helligkeitswert in dem erhaltenen Kantenbild gibt lt. Höntzsch Auskunft über die vorhandene Bildschärfe. Um Fehlbewertungen durch Rauschen zu vermeiden, muss der maximale Helligkeitswert mindestens zehnmal vorkommen, bevor er als Schärfewert übernommen wird.

Der Vergleich aller Kantenbilder der vorliegenden Fotografien dient der Bestimmung des schärfsten Bildes. Um ein optimales Kantenfilter für diese Anwendung zu finden, wurden verschiedene Kantenfilter-Operatoren verglichen.

---

<sup>9</sup> Wolfenbüttler Buchspiegel - Ein schonendes Verfahren zur Reproduktion von alten Büchern und Drucken.

Im Ergebnis wurde der Laplace-Operator im Bereich der genannten Anwendungen als am geeignetsten bewertet, da dieser die Qualitätsunterschiede der Testbilder gut unterscheiden konnte.

Das von Höntzsch vorgestellte Verfahren zur „Schärfebestimmung in digitalen Fotos“ soll in dem Folgenden Abschnitt erweitert werden, um eine Nutzung im Bereich der digitalen Stadtmodelle zu ermöglichen.

#### **4.6.2. Erweiterung der Herangehensweise von Höntzsch**

Bei der nachfolgend dargestellten, von der Verfasserin der Arbeit erweiterten Verfahrensweise zur Bestimmung der Bildqualität digitaler Fassadenfotografien, handelt es sich um ein Begutachten der Ausgangsbilder, die zur Texturierung genutzt werden sollen.

Qualitätsminderungen der Bildschärfe können auftreten, wenn kleine Texturbilder auf große Gebäudeflächen projiziert und somit größer skaliert werden. Die Auflösung des Ausgangsbildes ist in solchen Fällen nicht immer ausreichend, damit nach der Vergrößerung weiterhin scharfe Texturbilder vorhanden sind.

Diese Bilder werden durch das Verfahren von Höntzsch auf Schärfe untersucht und anschließend bewertet. Die Bewertung der Schärfe sollte in den Sachdaten der Gebäude vermerkt werden. Weiterhin ist eine Festlegung notwendig, bis zu welchem Wert ein Bild skaliert werden darf. In der Druckbranche liegt die Größenbeschränkung bei ca. 120% des Ausgangsbildes.

Bei der Verknüpfung des Bildes mit einer Geometrie im 3D-Stadtmodell, muss berechnet werden, um welchen Skalierungsfaktor das Bild vergrößert wird. Ist der Skalierungsfaktor größer 1.2, kann ein Hinweis durch das System erfolgen. Der Anwender kann so automatisch auf die Verwendung unterschiedlicher Texturqualitäten hingewiesen werden.

Zur Visualisierung des oben genannten Hinweises eignet sich eine Farbcodierung. Die Farben grün, gelb und rot bieten sich als Grobunterteilung für gut, mittel und schlecht an. Die Eignung stützt sich auf die bereits vorhandene Assoziation dieser Farben mit der genannten Einteilung. Um Zwischenstufen darzustellen, können Farben aus den Übergängen der drei Hauptfarben genutzt werden.

Im oben beschriebenen Beispiel der Farbcodierung wird die Bewertung der Texturen aller Gebäude angezeigt. Eine andere Variante ist das Anzeigen von Gegensätzen. Weisen benachbarte Gebäudeflächen unterschiedliche Texturqualitäten auf, wird der Anwender darauf aufmerksam gemacht. Diese Visualisierungsmethode zielt auf das Anzeigen von Widersprüchen, während die zuerst genannte eine Übersicht aller Qualitätsbewertungen erlaubt.

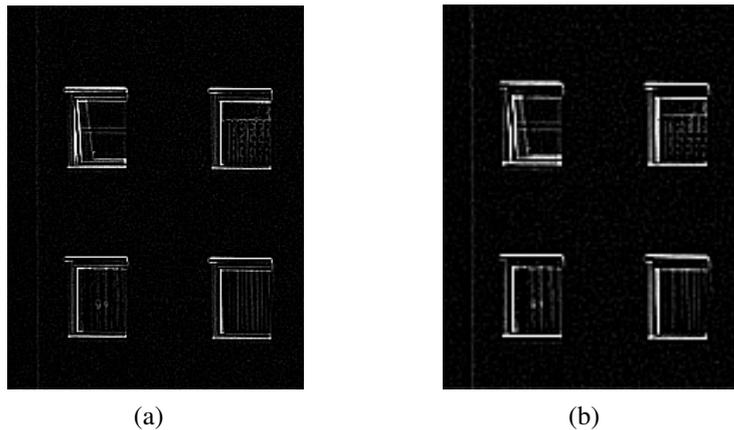


Abbildung 4.25: Ergebnisbilder des Laplace-Kantenfilters auf unterschiedlich großen Bildern, (a) ist das Ergebnis eines großen und (b) eines kleinen Quellbildes

### 4.6.3. Kacheln von Texturen

Gekachelte Texturen sollten eine möglichst gleich bleibende Helligkeit und Sättigung aufweisen. Intensitätsverläufe, hellere oder dunklere Regionen sollten vermieden werden. Ist dies nicht der Fall, kann es leicht zu übergeordneten Mustern kommen, die bei der Betrachtung negativ auffallen (vgl. Abschn. 2.5.2). Dahin gehend kann das Bild mittels einer Histogramm-Kontrolle analysiert werden. Sind die Farbwerte des Bildes über den gesamten Wertebereich  $[0, 255]$  verteilt, liegen hohe Farbschwankungen vor. Sind die Farbwerte auf einen kleinen Wertebereich beschränkt, sind die Intensitäten ähnlich. Sollen die Bilder nicht nur bewertet, sondern automatisch korrigiert werden, kann eine entsprechende Änderung des Histogramms vorgenommen werden.

## 5. Prototypische Umsetzung

In diesem Kapitel werden die prototypischen Umsetzungen, basierend auf der konzeptionellen Betrachtung beschrieben und erläutert. Die Schwerpunkte liegen in der Detaillierung der Polygone durch Generierung eines Bordsteins und der angepassten Texturierung von Straßenpolygonen durch die Beachtung von Krümmungen.

### 5.1. Entwicklungswerkzeuge

Zur Implementierung des Prototyps wurden neben den C++ Standardbibliotheken, die innerhalb der vorhandenen Visualisierungsplattform bestehenden Methoden genutzt und neue Funktionen entwickelt, die nachfolgend beschrieben und teilweise erläutert werden. Der eingefügte Quellcode orientiert sich an der C++-Syntax. Für die Programmierung der Benutzeroberfläche wurde die Bibliothek wxWidgets und der Dialog-Editor DialogBlocks verwendet.

Die Implementierung erfolgte mit Hilfe der integrierten Entwicklungsumgebung Visual Studio 2005 unter dem Betriebssystem Windows XP.

Die Wahl der Programmiersprachen ergab sich aus dem Anspruch der optimalen Integration in das Gesamtsystem, das ebenfalls auf diesen Sprachen aufgebaut ist.

### 5.2. Benutzeroberfläche

Der Dialog gliedert sich entsprechend der Konzeption in Abschnitt 4.5. in die zwei Zonen „Streetpolygon“ und „Curb“, die Vorschauoption „Preview“ wurde nicht umgesetzt. Der Bereich zur Einstellung der Parameter des Streetpolygons wurde in der oberen Hälfte untergebracht, da diese Funktion auf alle Polygone zutrifft. Die Detaillierungseinstellungen sind nicht für jedes Polygon erforderlich und daher im unteren Dialogteil angeordnet. In der Abbildung 5.1 ist die umgesetzte Benutzerschnittstelle dargestellt.

Die Sprache des Dialogs ist entgegengesetzt zum Entwurf in Englisch gehalten, um den Dialog optimal in die bestehende Anwendung zu integrieren.

Zur Texturauswahl, im Bereich „Streetpolygon“, steht ein Dialog-Öffnen-Button zur Verfügung. Die Pfad- und Dateiangaben sind im nebenstehenden Eingabefeld sichtbar. Darunter angeordnet sind die Eingabefelder zur Verschiebung der Textur in x- und y-Richtung sowie dem Rotationswinkel der Textur.

Weiter unten angeordnet sind die Einstellmöglichkeiten der Texturierungsform (Planar – Orientierung) und die Angabe des Orientierungswinkels. Die Eingabe des Winkels kann nur getätigt werden, wenn die Auswahl „Orientierung“ aktiv ist.

Andernfalls ist dieses Eingabefeld „ausgegraut“<sup>10</sup>. Seitlich dazu befindet sich ein Vorschauenfenster der gewählten Textur. Als unterstes Element des Bereichs folgt der Aktionsbutton „Assign Texture“, durch den der Texturierungsvorgang ausgelöst wird.

Im Bereich „Curb“ (dt. Bordstein) erfolgt die Auswahl der Bordsteintextur auf die gleiche Weise wie die Auswahl der Straßenpolygontextur. Unterhalb des Auswahlbuttons sind Einstellmöglichkeiten zur Breite, Höhe der Textur sowie des Texturwinkels. Auch hier erfolgt eine Vorschau der gewählten Textur in einem kleinen Fenster. Die darunter folgenden Angaben geben die Breite und Höhe des Bordsteins an. Anschließend erfolgt die Angabe, ob der Bordstein innerhalb oder außerhalb der Straßengeometrie positioniert werden soll. Als letztes Element folgt der Aktionsbutton „Create Border“, der die Generierung der Detailgeometrie startet.

In allen Auswahl- und Eingabefeldern, mit Ausnahme der Texturpfade, sind Standardwerte enthalten, sodass die Aktionen ohne Änderung der Werte ausgeführt werden können.

Um den Dialog zu schließen sind zwei Möglichkeiten vorhanden. Zum einen über den Standard-Schließen-Button oben rechts im Fenster, zum anderen über den Button „Quit Dialog“ unten rechts.

Die Aktionsbuttons „Assign Texture“, „Create Curb“ und „Quit Dialog“ sind durch eine stärkere Schriftbreite von den anderen Dialogelementen hervorgehoben, um die Bedeutung der Elemente zu betonen.

Das „Ausgrauen“ oder Deaktivieren einzelner Funktionen und Auswahlmöglichkeiten dient der Übersichtlichkeit des Dialogs. Es soll für den Anwender leicht erkennbar sein, welche Aktionen aktuell ausgeführt werden können. Zu den ausgraubaren Elementen zählt die Orientierungsangabe im Bereich des Streetpolygon, welche durch die Wahl der Texturierung mit Objektausrichtung aktiviert wird.

---

<sup>10</sup> Ausgrauen - In der Gestaltung von Benutzeroberflächen weit verbreitet und dient dem Kennzeichnen von inaktiven oder ungültigen Dialogelementen (z.B. Buttons, Menüeinträge).

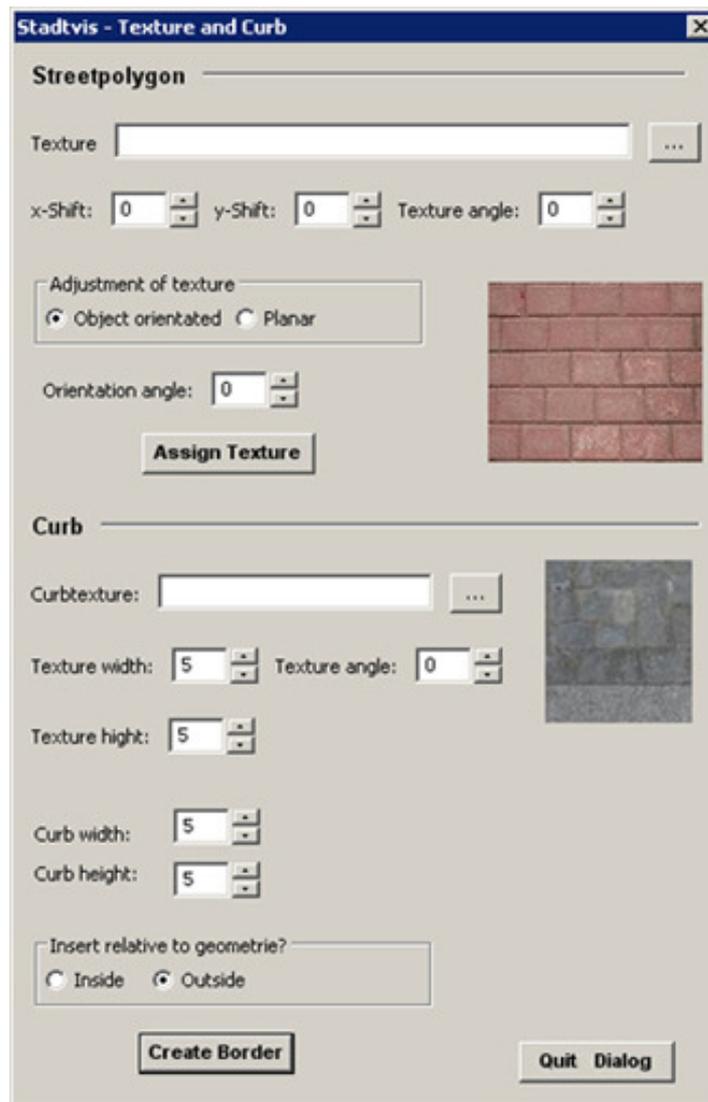


Abbildung 5.1: Layout des vorläufigen Dialogs zu Steuerung der Texturierung und Anfügen eines Bordsteins

### 5.3. Kantenverfolgung zur Krümmungserkennung

Der Algorithmus zur Erkennung von Krümmungen im Straßenverlauf wurde entsprechend der Konzeption umgesetzt. Durch betätigen des Buttons „Assign Texture“ oder des Shortcut „A“ wird die Texturierung ausgelöst und damit die Methode `CalcUVMapping` aufgerufen.

In `CalcUVMapping` wird zunächst die Umrandung, also die Border, der selektierten Polygone berechnet. Die Punkte der Border werden in ein Array geschrieben, welches in `CalculateSections` genutzt wird, um nacheinander die Sektionen zu berechnen. Von der Punktemenge der jeweiligen Region wird anschließend die AABB berechnet. Im Anschluss erfolgt in `doIt` die Berechnung der Sektionsorientierung.

```
//Ausschnitt der Funktion CalcUvMapping
while (mSelectedFacesRB[0].mPoints.size() != 0)
{
    // gibt eine Liste der noch zu bearbeiteten Punkte und
    // der zu einer Sektion gehörenden Faces zurück
    mSelectedFacesRB=rb.CalculateSections(mSelectedFacesRB
    , geo, faceList, turn, vector);
    turn++;

    std::vector<SelectedFaces> tmp_selection =
        mSelectedFaces;

    mSelectedFaces.clear();
    mSelectedFaces.push_back(SelectedFaces(geo));
    for (uint j=0;j<faceList.size();j++)
    {
        int t = faceList[j];
        mSelectedFaces[0].mFaceIndices.push_back(t);
    }
    // Berechnung der Bounding box der faceList-Punkte
    CalcBBoxForSelectedFaces();
    mSelectedFaces = tmp_selection;

    // Berechnung der Orientierung und Texturkoordinaten
    mTexturePlane.doIt(*mSelectedFaces[i].mGeometry,
    faceList, mSelectedFacesBBox, mCurrentRotation,
    withwithout, vector);
}
```

**CalculateSections** arbeitet auf der Border der selektierten Punkte. Ab dem zweiten Randpunkt wird der Winkel zwischen den ein- bzw. ausgehenden Borderkanten berechnet. Ist dieser kleiner  $10^\circ$ , werden die angrenzenden Flächen in eine Liste aufgenommen. Liegt der Winkel über dem festgelegten Schwellenwert von  $10^\circ$ , wird die aktuelle Sektion abgeschlossen. Der folgende Borderpunkt ist der Startpunkt der nächsten Sektion.

**CalculateSections** liefert zwei Listen zurück. Die erste, **mSelectedFacesRB**, ist eine Punkteliste und enthält die Punkte der Border ohne die bereits zu einer Sektion gezählten Punkte. Die zweite Liste, **faceList**, enthält die Indizes der Flächen, die zu der gefundenen Sektion gehören.

In **doIt** werden die so genannten Start- und Endpunkte der ermittelten Sektion gesucht, die Mittelpunkte berechnet und der aus den Mittelpunkten entstandene Vektor **angleObj** als Orientierung gespeichert. Der Winkel des Vektors **angleObj** zur x-Achse gibt den Rotationswinkel der Textur an.

### Berücksichtigung von Ausbuchtungen:

Polygonpunkte, die zu einer Ausbuchtung gehören, müssen bei der Bestimmung der Orientierung ignoriert und somit bei der Sektionssuche übersprungen werden. Um dies zu erreichen, müssen die Punkte der Ausbuchtung als solche erkannt werden. Dazu wird nach einer entsprechend großen Winkeländerung eine Anzahl der nachfolgenden Punkte kontrolliert (vgl. Abb. 5.2).

Die Kontrolle bezieht sich erneut auf den Winkel zwischen zwei Kanten. Die erste Kante K1 verbindet den direkten Vorgänger mit dem aktuellen Punkt, die zweite Kante K2 ergibt sich aus dem aktuellen Punkt und einem Nachfolger. In der Abbildung 5.2. ist zur Verdeutlichung der aktuelle Punkt rot, der Vorgänger grau und die Nachfolger orange, blau und grün markiert.

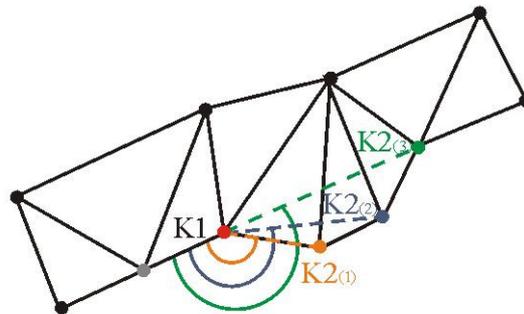


Abbildung 5.2: Darstellung zur Erkennung von Ausbuchtungen im Straßenverlauf

Wird zwischen der Kante K1 und einer Kante K2 keine Winkeländerung über den Schwellenwert hinaus, festgestellt, kann davon ausgegangen werden, dass es sich um eine Ausbuchtung handelt. In der Abbildung 5.2 trifft diese Aussage auf den grün markierten Nachfolger zu.

Die zu einer Ausbuchtung als zugehörig ermittelten Punkte werden aus der Liste der Border gelöscht und die zugehörigen Flächen in die `faceList` der aktuellen Sektion eingetragen. Anschließend wird die Sektionsbestimmung am letzten Punkt der Ausbuchtung fortgesetzt.

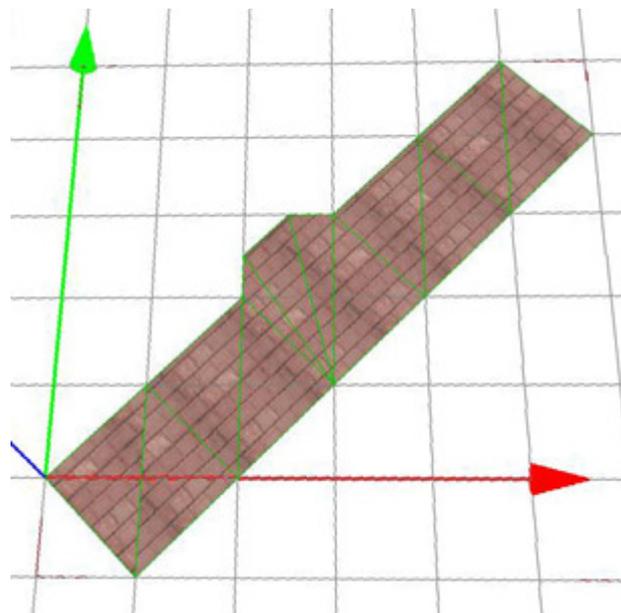


Abbildung 5.3: Ergebnisbild der Berücksichtigung von Ausbuchtungen bei der Orientierungsermittlung

```

// Quellecode zur Überprüfung, ob eine Krümmung oder eine
// Ausbuchtung vorliegt
if (angle > 10) // eine vorläufige Krümmung erkannt
{
    Point3f p_n2;
    x = 1; // Zähler der gefundenen Ausbuchtungspunkte
    curve = true; // Krümmung oder Ausbuchtung
    // Kontrolle der nachfolgenden Borderpunkte
    for (int a=2; a<5; a++)
    {
        // p_n2 ist Nachfolger
        if (s+a <= edges.size()) p_n2 = edges[s+a];
        else break;
        Vector2f v2(Point2f(p.x, p.y), Point2f(p_n2.x,
            p_n2.y));
        float angle_2 = acos(v1 * v2 / (v1.Length() *
            v2.Length()));
        angle_2 = angle_2*180.0f/3.14f;
        if (angle_2 < 10) {
            curve = false;
            break;
        }
        else
        {
            curve = true;
            x++; // Zähler der Ausbuchtungspunkte erhöhen
        }
    }
    // Überspringen der Ausbuchtungspunkte
    if (curve == false) s+=x;

    // Zurücksetzen des Zählers
    if (curve == true) x=0;
}

```

### Problem: Verwaltung der Geometriepunkte

Bei der Umsetzung wurde zunächst von gleichmäßigen Polygonen und optimalen Startpunkten ausgegangen. Unter diesen Voraussetzungen funktioniert der Algorithmus sehr gut. Die Verwaltung der Sektionspunkte und derer, die noch untersucht werden müssen, verläuft wie folgt:

- Die Border wird kopiert, es entsteht die Liste `remain_edges`, in der die noch zu verarbeitenden Punkte enthalten sind.
- Entsprechend der Anzahl der Flächen im gebildeten Segment werden Punkte am Ende der `remain_egdes` gelöscht
- Die abgelaufenen Punkte werden ebenfalls nicht in die neue `remain_egdes` übernommen
- Dadurch werden „vorn“ und „hinten“ die gleiche Anzahl von Punkten gelöscht.

In der Realität ist dieser Aufbau der Polygone nur selten vorhanden. Oft ist nicht die gleiche Anzahl von Punkten entlang der rechten und linken Seite zu finden. Dadurch kann es mit dem oben beschriebenen Algorithmus zu Fehlern kommen. Es können einerseits zu viele andererseits zu wenige Punkte gelöscht werden. Demzufolge ist eine korrekte Texturierung nicht gewährleistet. Dadurch kann ein Problem wie in Abb. 5.4. dargestellt, entstehen.

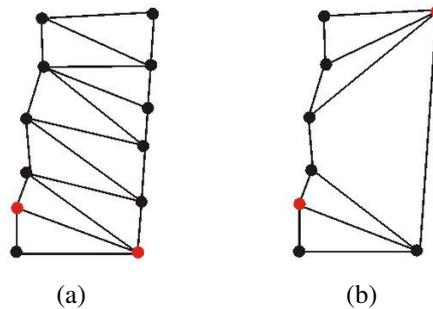


Abbildung 5.4: (a) ideales Polygon (b) reales Polygon

(b) Beim Bearbeiten des linken roten Punktes wird bereits der obere rote Punkt in `remain_edges` gelöscht, es entsteht eine fehlerhafte Texturierung

Zur Vermeidung dieses Problems, wird getestet, ob der zu löschende Punkt zu der Fläche des aktuellen Punktes gehört. Um nicht bei jedem Test alle Punkte und Flächen mehrfach durchlaufen zu müssen, wird zu Beginn die Liste `RoadPointList` erstellt. Diese enthält alle Punkte der Geometrie und die dazugehörigen Flächen.

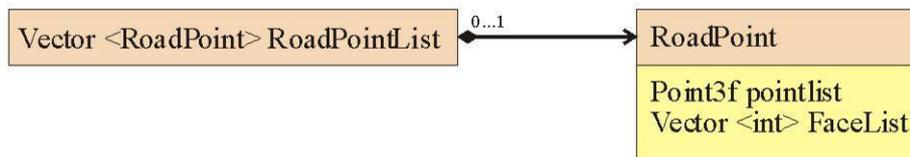


Abbildung 5.5: UML-Darstellung der `RoadPointList`

Deklaration der `RoadPointList` im Programm:

```
typedef std::vector<RoadPoint> RoadPointList;
struct RoadPoint
{
    Point3f point;
    std::vector<int> PointFaceList;
};
```

## 5.4. Texturanpassung mit Hilfe der Objektausrichtung

Die Berechnung der allgemeinen Orientierung des Objektes ist eine konzeptionelle Vorstufe zur differenzierten Texturierung. Auf dieser Ebene können Ergebnisse wie in Abbildung 5.6 b erreicht werden. In der prototypischen Umsetzung ist diese Art der Texturausrichtung bereits durch die Erweiterung zur automatischen Ermittlung der Orientierung innerhalb einzelner Sektionen abgelöst.

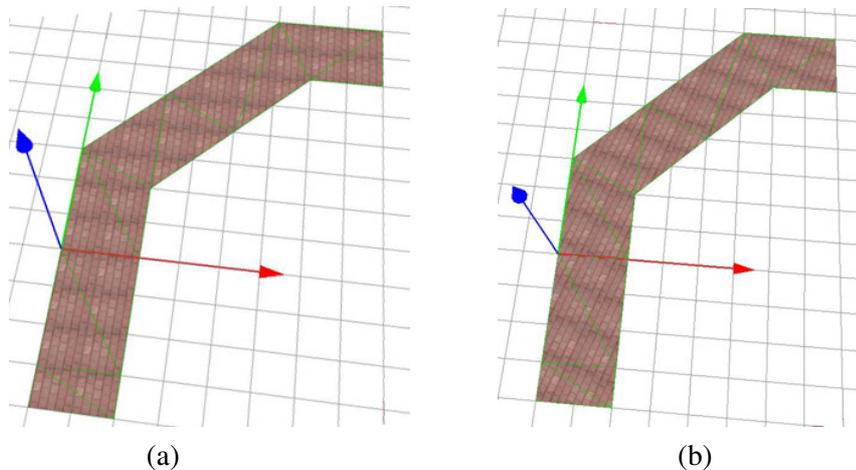


Abbildung 5.6: Texturiertes Polygon ohne (a) und mit (b) Berücksichtigung der allgemeinen Objektausrichtung

### Kombination von Sektionen und Orientierung

Die Ausrichtung der Textur entsprechend der Sektionsorientierung liefert im Vergleich zu Abb. 5.6 b ein besseres Ergebnis (s. Abb. 5.7).

Um falsche Texturausrichtungen bei zu kurzen Sektionen zu vermeiden, erfolgt ein Abgleich der berechneten Orientierung und der ersten Kante der Border dieser Sektion (vgl. Abb. 4.11). Ist der Winkel zwischen den Vektoren größer als  $10^\circ$ , wird die Richtung der Kante als Orientierung verwendet.

```
// Winkel zwischen x-Achse und Vektor der Kante
float vec_angle = acos((horizont * vector) /
                      (horizont.Length() * vector.Length()));
if (fabs(vec_angle - angleObj) >= 0.17453) //10° Unterschied
{
    //Übernahme der Orientierung der Kante als
    Sektionsorientierung
    angleObj = vec_angle;
}
```

**Ergebnisse der angepassten Texturierung:**

Die nachfolgend genannten „echten“ Straßenpolygone stammen aus einem Datensatz der Lutherstadt Wittenberg.

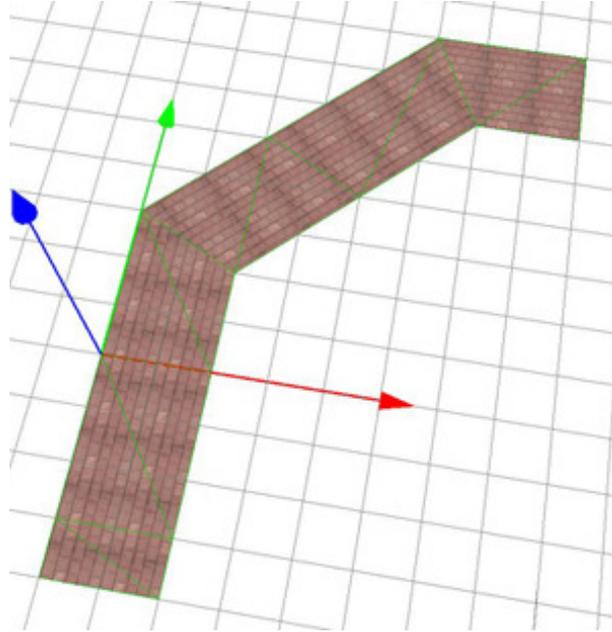


Abbildung 5.7: Die Textur wird dem Verlauf der Geometrie angepasst, wenn die Orientierung der einzelnen Sektionen berücksichtigt wird.

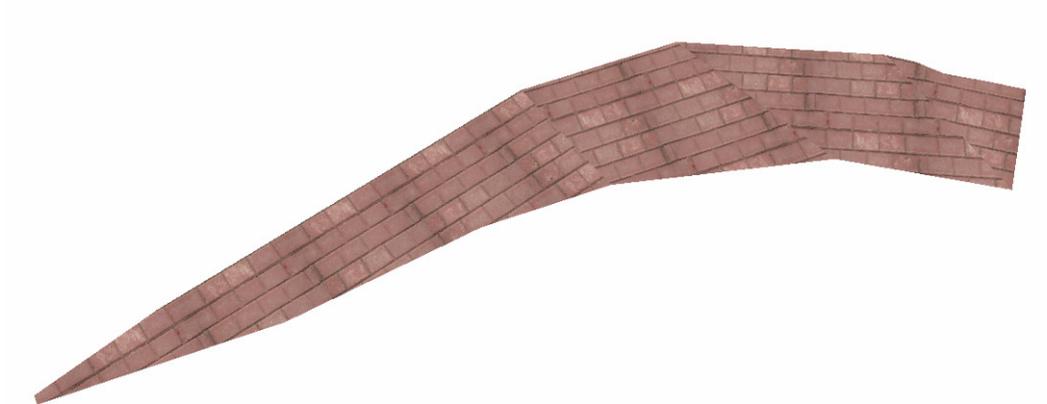


Abbildung 5.8: „echtes“ Straßenpolygon mit Krümmung

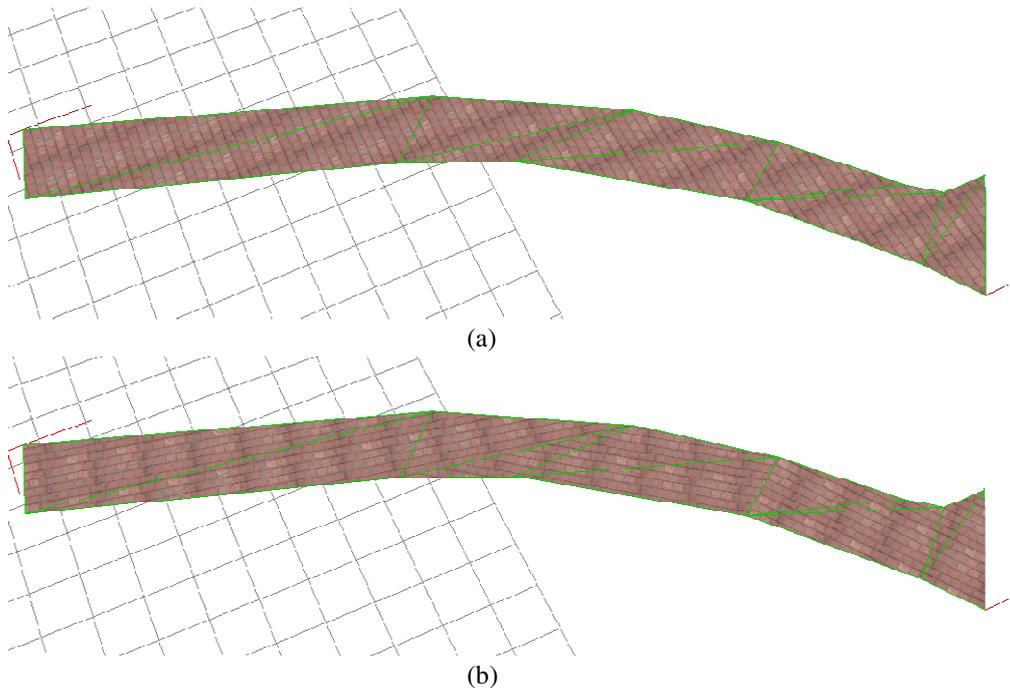


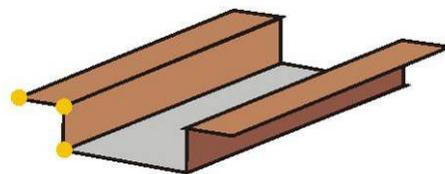
Abbildung 5.9: Vergleich der Methoden ohne (a) und mit (b) Krümmungserkennung

### 5.5. Bordsteingenerierung und Bordsteinintegration

Standardmäßig ist in dem Benutzerdialog das Einfügen der Bordsteingeometrie außerhalb der Straßengeometrie eingestellt. Somit wird angenommen, dass das Polygon lediglich die Fahrbahn enthält.

Vor der Berechnung der Bordsteinpunkte, erfolgt die Bildung der so genannten Border. Die Border-Eigenschaft „Orientierung“ ist wichtig zur Berechnung der Bordsteinpunkte. Sie ist ausschlaggebend, ob die Punkte in Umlaufrichtung rechts bzw. links der Borderkanten eingefügt werden.

An jedem Punkt der Border müssen mindestens drei neue Bordsteinpunkte eingefügt werden: der abgesenkte Straßenpunkt, der Bordsteinpunkt oberhalb des Straßenpunkts sowie der Punkt, der die Breite der Geometrie festlegt (im Bild rechts dargestellt durch die gelb markierten Punkte).



Das Straßenpolygon wird abgesenkt, weil die neuen Geometriepunkte auf der Höhe des ursprünglichen Straßenpolygons eingefügt werden. Andernfalls würde der Bordstein über andere Flächen, wie Grünflächen, hinausragen.

Das Berechnen und Erstellen der Bordsteingeometrie geschieht in `CreateBorderGeometry`. Zunächst erfolgt die Berechnung des neuen Geometriepunktes unter Berücksichtigung verschiedener Parameter. In die Berechnung fließt neben der oben genannten Umlauforientierung, die vom Nutzer angegebene Breite des Bordsteins, der Winkel zwischen den angrenzenden Kanten sowie die Auswahl über eine innere oder äußere Integration ein. Ist der Winkel zwischen den Kanten gleich null, kann der neue Punkt standardmäßig in einem  $90^\circ$  Winkel zu den Kanten generiert werden. An bestimmten rechtwinkligen Kanten erfolgt die Berechnung von zwei zusätzlichen Geometriepunkten (vgl. Abb. 4.22 und Abb. 4.23).

```
//Beispiel für das Berechnen eines neuen Geometriepunktes
v7.x = (v2.x*cos_a + v2.y*(-sin_a)); // Drehen
v7.y = (v2.x*sin_a + v2.y*cos_a);
v7.x = v7.x/v2.Length()*X;          // Länge zuweisen
v7.y = v7.y/v2.Length()*X;
v7.x = point.x+v7.x;                // Position relativ zum
                                   // ursprünglichen Punkt
                                   // „point“

v7.y=point.y+v7.y;
```

Die Werte für `cos_a` und `sin_a` ergeben sich aus den Winkeln der Kanten und werden in einem früheren Schritt vorberechnet.

Anschließend erfolgt die Erstellung einer neuen Geometrie und das Zuweisen der Geometrie- und Texturkoordinaten. Das Ergebnis einer solchen Berechnung ist in Abbildung 5.10 und 5.11 dargestellt. Die Textur muss vor Erstellung ausgewählt werden, ist jedoch nachträglich über den Benutzerdialog änderbar.

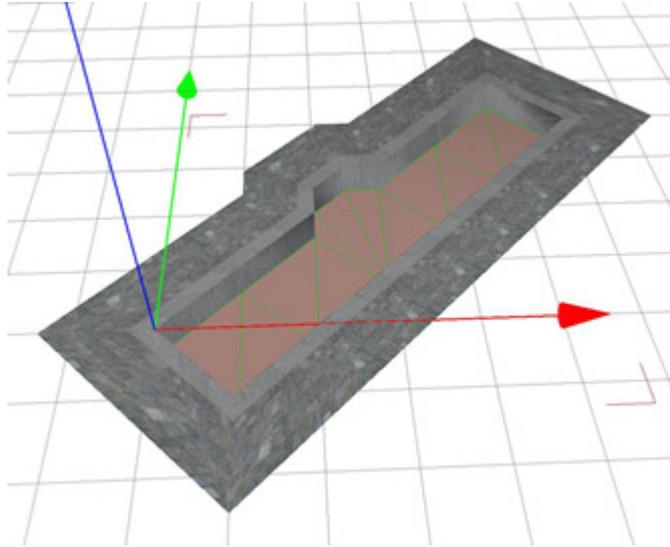
**Ergebnisse der Bordsteingenerierung:**

Abbildung 5.10: Testpolygon mit Ausbuchtung

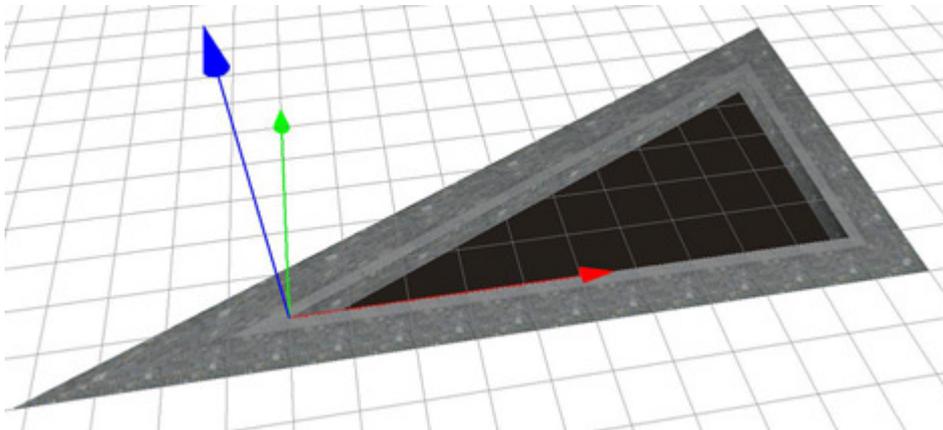


Abbildung 5.11: Testdreieck

## 5.6. Ergebnisbilder der Texturierung und Bordsteingenerierung

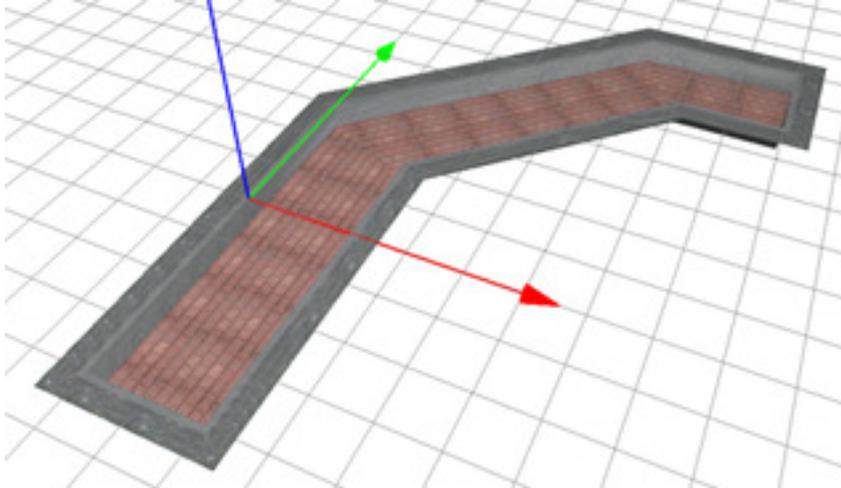


Abbildung 5.12: Testpolygon mit Bordsteingemetrie und automatisch angepasster Textur.

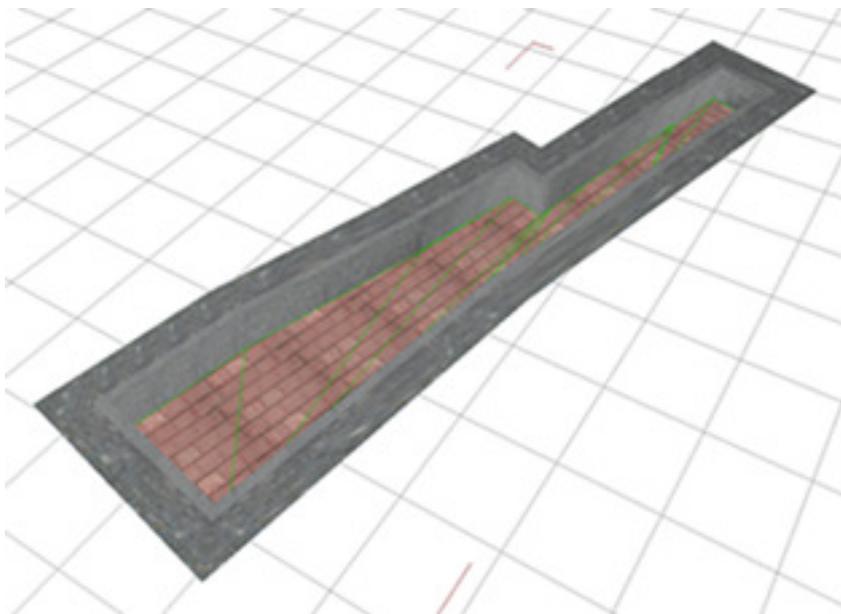


Abbildung 5.13: „Echtes“ Straßenpolygon mit Textur und Bordstein.

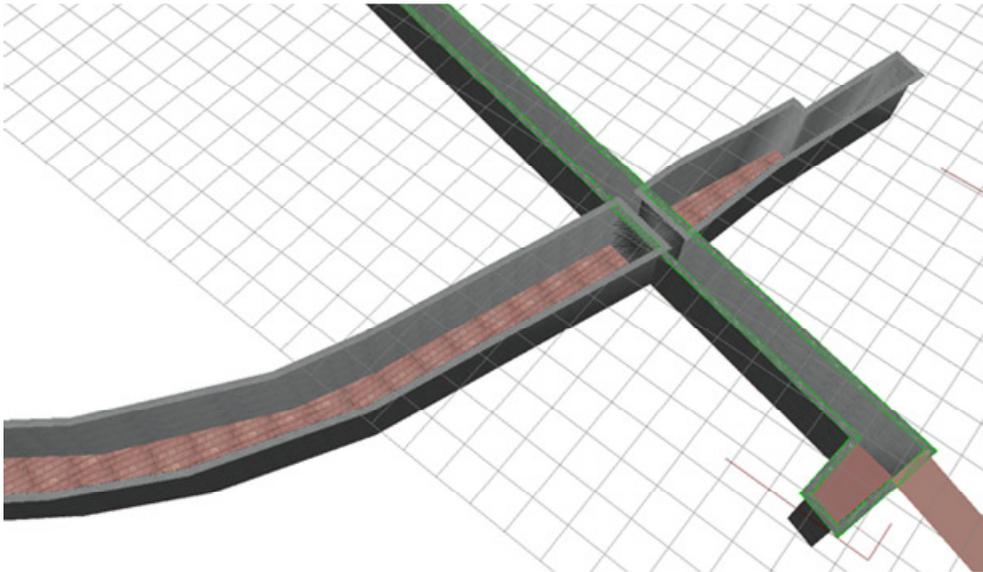


Abbildung 5.14: Drei „echte“ Straßenpolygone mit überhöhter Bordsteingeometrie.

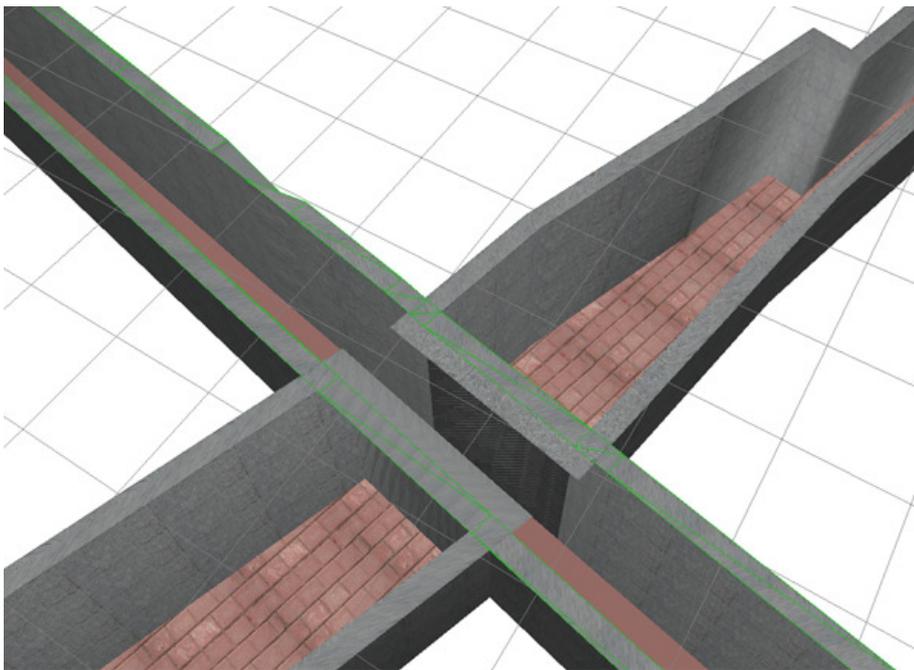


Abbildung 5.15: Nahansicht der Schnittstelle der drei Polygone mit überhöhter Bordsteingeometrie aus Abbildung 5.14

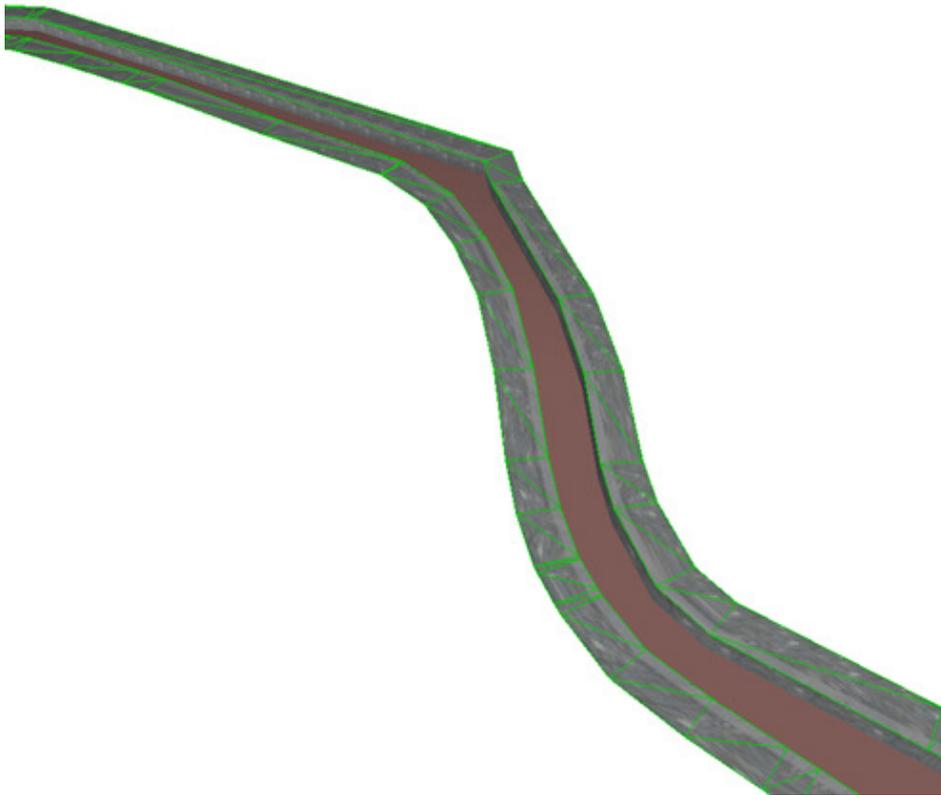


Abbildung 5.16: „Echtes“ Straßenpolygon der Lutherstadt Wittenberg, versehen mit einem Bordstein.

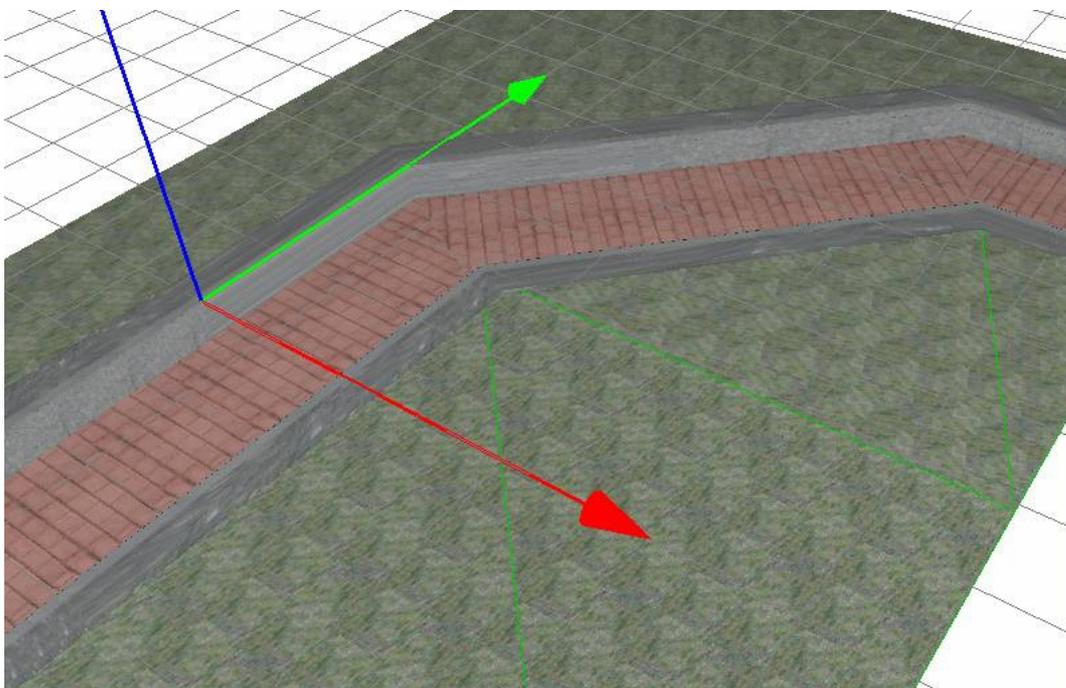


Abbildung 5.17: Texturiertes Testpolygon in Verbindung mit symbolisierten Grünflächen.

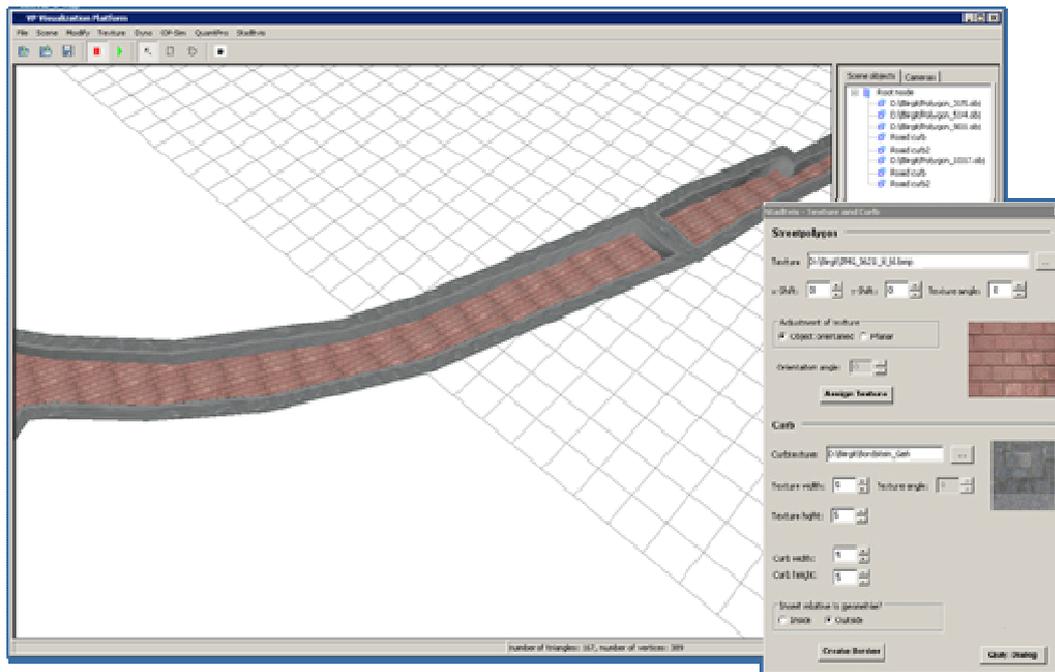


Abbildung 5.18: Zwei „echte“ Straßenpolygone, versehen mit Textur und Bordstein, dargestellt in der Visualisierungsplattform.

## 6. Zusammenfassung / Ergebnisse

Ziel der Arbeit war es, Methoden zur semi-automatischen Texturierung von Grund- und Bodengeometrien zu untersuchen und zu entwickeln. Im Speziellen sollte die Texturierung von Straßengeometrien betrachtet werden.

Dazu wurden zunächst die zur Verfügung stehenden Ausgangsdaten untersucht. Es zeigte sich, dass diese im Allgemeinen nicht stark detailliert und differenziert in den ausgewählten Kommunen vorliegen.

Bei der Recherche zu existierenden Methoden innerhalb der gestellten Spezialisierung des Themas konnten keine speziellen Algorithmen und Verfahren gefunden werden.

Unter Berücksichtigung dieser Voraussetzungen wurden verschiedene Methoden zur semi-automatischen Texturierung untersucht und Lösungen für Probleme aufgezeigt, die bei der Erstellung einer angepassten Texturierung auftreten. Das entwickelte Verfahren wurde in die bestehende Visualisierungsplattform integriert. Die Ausrichtung der Textur an geraden Straßenpolygonen ergab keine großen Probleme. Schwierigkeiten entstanden bei Polygonen mit enthaltener Krümmung. Die Schwierigkeiten waren für die ausgewogenen Testpolygone lösbar. Auch bei einigen „echten“ Straßenpolygonen konnten gute Ergebnisse erzielt werden. Eine stabile Texturierung von „echten“ Straßenpolygonen wurde auf Grund der sehr großen Variabilität der Anordnung, z.B. durch ungleichmäßige Verteilung, der Polygonpunkte noch nicht umgesetzt.

Die besten Ergebnisse des Algorithmus zur Krümmungserkennung lassen sich erzielen, wenn:

- die Punkte in der Border so sortiert sind, dass der untere linke bzw. der obere rechte Punkt der erste und eine Speicherung im Uhrzeigersinn gegeben ist.
- die Geometriepunkte gleichmäßig verteilt sind.
- eine gute Triangulierung der Polygone vorliegt.

Defizite in der entwickelten Verfahrensweise sind bei der zuverlässigen Erkennung von Unregelmäßigkeiten, im Sinne von Kreuzungen, Krümmungen und Ausbuchtungen vorhanden. Ist eine dieser Unregelmäßigkeiten nur an einer Polygonseite zu finden und die Analyse erfolgt nicht an derselben, werden diese nicht gefunden.

Der zweite Schwerpunkt der vorliegenden Arbeit liegt in der Erstellung eines Bordsteins, als Beispiel zur automatischen Detaillierung von Straßengeometrien. Der dafür entwickelte Algorithmus funktioniert für die Testpolygone sehr gut. Auch für eine Reihe von „echten“ Straßenpolygonen lieferte das Verfahren gute Ergebnisse. Ursächlich für die inkorrekten Ergebnisse ist meist eine fehlerhafte Klassifizierung in der Vorverarbeitung.

Ein Ziel der Arbeit ist, die Akzeptanz und Qualität der virtuellen Modelle insgesamt zu erhöhen. Dieses Ziel wird durch die konzipierten und teilweise umgesetzten Methoden erreicht.

Eine weitere Zielstellung fordert den weitestgehenden Verzicht auf manuelle Eingaben in den entwickelten Verfahren der fotorealistischen Texturierung. Dies wird erreicht, in dem Standardwerte vorgegeben sind. Der Anwender muss ausschließlich die Textur auswählen. Auch in der Bordsteinerstellung ist neben der Texturauswahl keine weitere Angabe zu tätigen, wenn mit den Standardwerten gearbeitet werden kann.

	Manuelle Texturierung 3D-Studio (Variante 2)	Luftbild (Variante 1)	Planares Mapping (Variante 3)	Konzipierter, Implementierter Ansatz
Grafische Qualität insgesamt	++	o	o	+
Bedienbarkeit	o	+	+	+
Anwendbarkeit für Fachleute Stadtplanung	-	+	+	+
Zeitaufwand	-	+	o	o
Flexibilität	+	-	o	-

Tabelle 6.1: Vergleichende Übersicht zu den betrachteten Texturierungsmethoden im Abschnitt 4.4.1

Legende: ++ sehr gut  
+ gut  
o mittel  
- schlecht

Zusammenfassend kann festgestellt werden, dass eine semi-automatische Detaillierung und Texturierung von optimalen Straßengeometrien mit dem entwickelten Algorithmus zu erstellen ist. Um die Verfahren zuverlässig auf „echte“ Straßenpolygone anzuwenden, sind Erweiterungen erforderlich. Auf diese wird im Kapitel 7 eingegangen.

## 7. Ausblick

Die entwickelten Verfahren sind, wie oben beschrieben, an optimalen Polygonen zuverlässig einzusetzen. Um die Algorithmen für die unterschiedlichsten und meist „schwierigen“ Strukturen der realen Straßenpolygone zu erweitern, sind folgende Änderungen notwendig.

Als zukünftige Entwicklungen in der Krümmungserkennung sind sinnvoll:

- Die Verwaltung die Geometriepunkte anpassen und optimieren.
- Suche nach optimalen Startpunkten.
- Analysieren der Geometrie an zwei Seiten.  
Dies erlaubt eine Erkennung von Kreuzungen und garantiert das Finden von anderen Unregelmäßigkeiten, wie beispielsweise Ausbuchtungen.
- Berücksichtigen der Anforderungen bei der Sektionsbestimmung von Polygonzügen mit innen liegenden Punkten.
- Das Einstellen des Schwellenwertes zur Erkennung von Krümmungen und Bestimmung der Sektionsgröße kann dem Anwender angeboten werden.  
Diese Erweiterung ermöglicht das individuellere Gestalten der Texturierung.
- Verknüpfung der Texturauswahl mit Sachdatenbanken.  
Stehen solche Datenbanken zur Verfügung, kann die manuelle Texturauswahl entfallen. Der Grad der Automatisierung des Texturierungsprozesses steigt mit dieser Erweiterung.

Als Erweiterung des Algorithmus zur Erstellung eines Bordsteins sind nachfolgende Funktionen sinnvoll:

- Der Ausbau der Flexibilität durch Umsetzen der Bordsteingenerierung im inneren der Straßengeometrie. Dies ist analog zum bereits bestehenden Algorithmus möglich.
- Bereitstellen der Möglichkeit zum Generieren weiterer Detailgeometrien zur zusätzlichen Differenzierung. Auch dies ist auf Grundlage des entwickelten Verfahrens leicht umsetzbar.
- Festlegung der gewünschten Standardparameter im Benutzerdialog durch den Anwender.  
Eine solche Funktion kann die Akzeptanz des Systems weiter erhöhen, da individuelle Einstellungen des Anwenders für weitere Arbeitsabläufe gespeichert werden können.

Im Zuge der Erweiterung ist es notwendig die Benutzeroberfläche anzupassen. Um den Dialog nicht zu überladen, sollte, wie bereits in Abschnitt 4.5 erwähnt,

eine übersichtliche Gestaltung gefunden werden. Auch sollte untersucht werden, welche der vorgeschlagenen Funktionen häufig oder eher selten genutzt werden. Entsprechend den Ergebnissen, ist der Aufbau der Benutzeroberfläche zu gestalten.

Zur Entwicklung einer automatischen Qualitätskontrolle der verwendeten Texturen können Verfahren wie in Abschnitt 4.6 umgesetzt werden. Eine solche Analyse kann die Verwendung von Fassadentexturen äquivalenter Qualitäten unterstützen und auf den Gebrauch von nicht geeigneten Kacheltexturen hinweisen.

Zur Texturierung von unebenen Bodengeometrien könnte das Verfahren der „Hard Constrains“ von Eckstein et al. [ESG01] eingesetzt werden. Bei der planaren Projektion würden auf unebenen Geländemodellen Verzerrungen auftreten. Diese können mit der Methode nach Eckstein et al. vermieden werden.

## Literaturverzeichnis

Alle Zugriffe auf die hier gelisteten Internetquellen waren bei Abgabe der Diplomarbeit möglich.

- [3DA] ESRI, ArcGIS 3D Analyst  
URL: [http://www.esri.com/software/arcgis/extensions/3danalyst/graphics/3d\\_zoom-global-pop.gif](http://www.esri.com/software/arcgis/extensions/3danalyst/graphics/3d_zoom-global-pop.gif)
- [3DGeo] 3D Geo  
URL: <http://www.3dgeo.de>
- [3DGLX] 3D Geo GmbH: Schlüsseltechnologie 3D für effektive und sichere Geoinformationsprodukte
- [3DSOM] Creative Dimension Software Ltd, 3DSOM  
URL: <http://www.3dsom.com/>
- [ABH03] Albert, J., Bachmann, M., Hellmeier, A.: Zielgruppen und Anwendungen für Digitale Stadtmodelle und Digitale Geländemodelle, Erhebung im Rahmen der Arbeitsgruppe „anwendungen und Zielgruppen“ der SIG3D im Rahmen der Initiative GDI-NRW, 2003
- [ArcGIS] ESRI: ArcGIS  
URL: <http://esri-germany.de/products/arcgis/about/desktop.html>
- [Berlin] Stadtmodell Berlin  
URL: <http://www.3d-stadtmodell-berlin.de>
- [Blom] Blom: Datenbanken der Pictometry Schrägaufnahmen  
URL: <http://www.blomasa.com/germany/de/produkte--services/produkte--services/datenbanken/pictometry-schr%C3%A4gaufnahmen>
- [Bru07] Brunner, David: Skelettierung von 3D-Objekten aus kubisch-raumzentrierten Gittern und deren Anwendung in der Segmentierung, Manipulierung und Klassifikation, Dissertation, Chemnitz, 2007
- [BS86] Bier, E.A., Sloan, K. R. *Two-Part Texture Mapping*, IEEE Computer Graphics and Application, 1986

- [BV05] Bähr, Vögtle: Digitale Bildverarbeitung - Anwendung in Photogrammetrie, Fernerkundung und GIS, Wichmann, 4. Auflage, 2005
- [CEDIM] Center for Disaster Management and Risk Reduction Technology  
URL: <http://www.cedim.de/902.php>
- [CR06] Collins, Brian D., Kayen, Robert: Applicability of Terrestrial LIDAR Scanning for Scientific Studies in Grand Canyon National Park, Arizona, 2006
- [CWSM] CWSM GmbH  
URL: <http://www.cwsm.de>
- [Ebert et al.03] Ebert, David S.; Musgrave, F. K. ; Peachey, Darwyn ; Perlin, Ken ; Worley, Steven: Texturing and Modeling: A Procedural Approach. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2003
- [Eng92] Englert, Gabriele: Visuelle Texturen: Modellierung, Generierung und Anwendung, Technische Hochschule Darmstadt, Fachbereich Informatik, Dissertation, Darmstadt, 1992
- [Eri05] Ericson, Christer.: Real Time Collision Detection - Series in Interactive 3D Technology, In: The Morgan Kaufman Series in Interactive 3D Technology, Elsevier, 2005
- [ESG01] Eckstein, Ilya, Surazhsky, Vitaly, Gotsman, Craig: Texture Mapping with Hard Constrains, Eurographics, Computer Science Department, Technion - Israel Institute of Technology, 2001
- [FH68] Finsterwalder, R., Hofmann, W.: Photogrammetrie, Berlin, 3. Auflage, 1968
- [fLAB08] futureLAB: TouristRemover  
URL: <http://www.snapmania.com/info/en/trm/>
- [FVD05] Foley, J.D., Van Dam, A. V.: Fundamentals of Interactive Computer Graphics, Addison-Wesley, 2. Auflage, 2005
- [GEarth] Google, Google Eart  
URL: <http://earth.google.com/>
- [GeoC] GeoContent  
<http://www.geocontent.de/>

- [GisN]           Gis-News  
URL: [http://www.gis-news.de/gis/gis\\_definit.htm](http://www.gis-news.de/gis/gis_definit.htm)
- [GisW]           GISWIKI  
URL: [http://www.giswiki.org/wiki/Geographic\\_information\\_system](http://www.giswiki.org/wiki/Geographic_information_system)
- [GTA]            GTA Geoinformatik GmbH  
URL: <http://www.gta-geo.com>
- [Guc05]          Guckel, Sabrina: Untersuchung zur fotorealistischen Visualisierung der DSGK-3D Daten durch Texture Mapping, HafenCity Universität, Department Geomatik, Diplomarbeit, Hamburg, 2005
- [Hön05]          Höntzsch, Bernd: Schärfestimmung in digitalen Fotos, Diplomarbeit (FH), Köln, 2005
- [Höp05]          Höpfner, A.: Interaktive Funktionalitäten bei der virtuellen Visualisierung urbaner Objekte und Strukturen, OvG-Universität Magdeburg, Fakultät für Informatik, Master's Thesis, Magdeburg, 2005
- [Ian04]          Stefano Ianigro: Ein Modell zur Simulation eines innerstädtischen Verkehrsablaufes und zur Steuerung von Lichtsignalanlagen mittels Petri-Netzen unter Berücksichtigung der Grünen Welle, S.12, Universität der Bundeswehr Hamburg, Fachbereich Wirtschafts- und Organisationswissenschaft, Dissertation, Hamburg, 1994
- [IFF]            Fraunhofer-Institut IFF Magdeburg, Abteilung VD
- [KDV]            Kommunale Datenverarbeitungsgesellschaft mbH Lutherstadt Wittenberg
- [KM]             Kummer, Möllering: Vermessungs- und Geoinformationsrecht Sachsen-Anhalt, 3. Auflage, Kommunal- und Schul-Verlag GmbH & Co. KG Wiesbaden, 2005
- [Kop02]          Koppers, Lothar: Generierung von Objekten für 3D-Stadtmodelle, Universität der Bundeswehr München, Fakultät für Bauingenieur- und Vermessungswesen, Dissertation, München, 2002
- [Kop04]          Koppers, Lothar: 3D Stadtmodelle - Techniken, Trends, Visionen, In: Mitteilungen 2/2004, Dessau, 2004

- [Lan04] Lancelle, M.: Automatische Generierung und Visualisierung von 3D-Stadtmodellen, Technische Universität Braunschweig, Institut für Computergrafik, Diplomarbeit, Braunschweig, 2004
- [München] 3D Reality Map: München 3D  
URL: [http://www.reality-maps.de/de/shop/d\\_shop\\_3dstadtplan\\_m.html](http://www.reality-maps.de/de/shop/d_shop_3dstadtplan_m.html)
- [Net05] Neteler, Markus: Orthofoto-Herstellung aus Luftbildern. Vom Luftbild zum Orthofoto, 2005  
URL: [http://gdf-hannover.com/lit\\_html/grasshandbuch\\_v12/node133.html](http://gdf-hannover.com/lit_html/grasshandbuch_v12/node133.html)
- [Nverse] Precision Lightworks, Nverse Photo  
URL: <http://www.precisionlightworks.com/Products.html>
- [Orla] Orlamünder: Vorlesungsskript Computergrafik, Technische Universität Dresden, Fakultät für Informatik
- [Pet05] Petzold, B.: 3D-Stadtmodelle - Eine Orientierungshilfe des Städtetages NRW, 2005
- [Pre99] Preim, Bernhard: Entwicklung interaktiver Systeme - Grundlagen, Fallbeispiele und innovative Anwendungsfelder, Springer-Verlag, 1999
- [RSS] Remote Sensing Solutions GmbH, 3D Reality Maps  
URL: [http://www.realitymaps.de/de/produkte/d\\_produkte\\_3dstadtplaene.html](http://www.realitymaps.de/de/produkte/d_produkte_3dstadtplaene.html)
- [RW97] Ritter, Gerhard X., Wilson, Joseph N.: *Handbook of Computer Vision Algorithms in Image Algebra*, S.79, CRC Press 1997
- [SA07] SketchUp: SketchUp and ArcGIS how they work together  
URL: [http://download.sketchup.com/downloads/markets/ESRI\\_Workflow.pdf](http://download.sketchup.com/downloads/markets/ESRI_Workflow.pdf)
- [Sch05] Schulze, Malte: Von computergrafischen zu haptischen Texturen. Virtual Reality für den Entwicklungsbereich Design/Styling in der Automobilindustrie, Universität Hannover, Fakultät für Elektrotechnik und Informatik, Dissertation, Hannover, 2005
- [Shn02] Shneiderman, Ben: User Interface Design, mitp-Verlag, 3.Auflage, 2002
- [SkUP] Google: SketchUp  
URL: <http://de.sketchup.com/?id=910>

- [Sch00] Schumann, Heidrun, Müller, Wolfgang: Visualisierung : Grundlagen und allgemeine Methoden, Springer-Verlag, 2000
- [SMB] 3D-Stadtmodell Berlin  
URL: <http://www.3d-stadtmodell-berlin.de/imperia/md/images/3d/innenstdt.jpg>
- [SMC] Stadtmodell Coburg  
URL: [http://www.graphisoft-nordbayern.de/stadtmodell-virtuelles-coburg-fertiggestellt\\_tipp\\_363.html](http://www.graphisoft-nordbayern.de/stadtmodell-virtuelles-coburg-fertiggestellt_tipp_363.html)
- [SSJ] Software Service John: VisAll 3D  
URL: <http://www.john-software.de>
- [StBe] Stoske, S., Bertling: Experiment: Entfernung von Bewegung, Velbert  
URL: <http://www.stoske.de/digicam/Experiment/nomov.html>
- [VTrace] Hengel, A. v.d., Dick, A., Thormählen, T., Ward, B., Torr, P. H. S.: Video Trace: Rapid interactive scene modelling from Video, ACM Transaction on Graphics, Vol. 26. No. 3, 2007
- [XTTZ01] Xuetao, L., Tong Wing, W., Tiow Seng, T., Zhiyong, H.: Decomposing Polygon Meshes for Interactive Applications, National University of Singapore, 2001

## Anhang

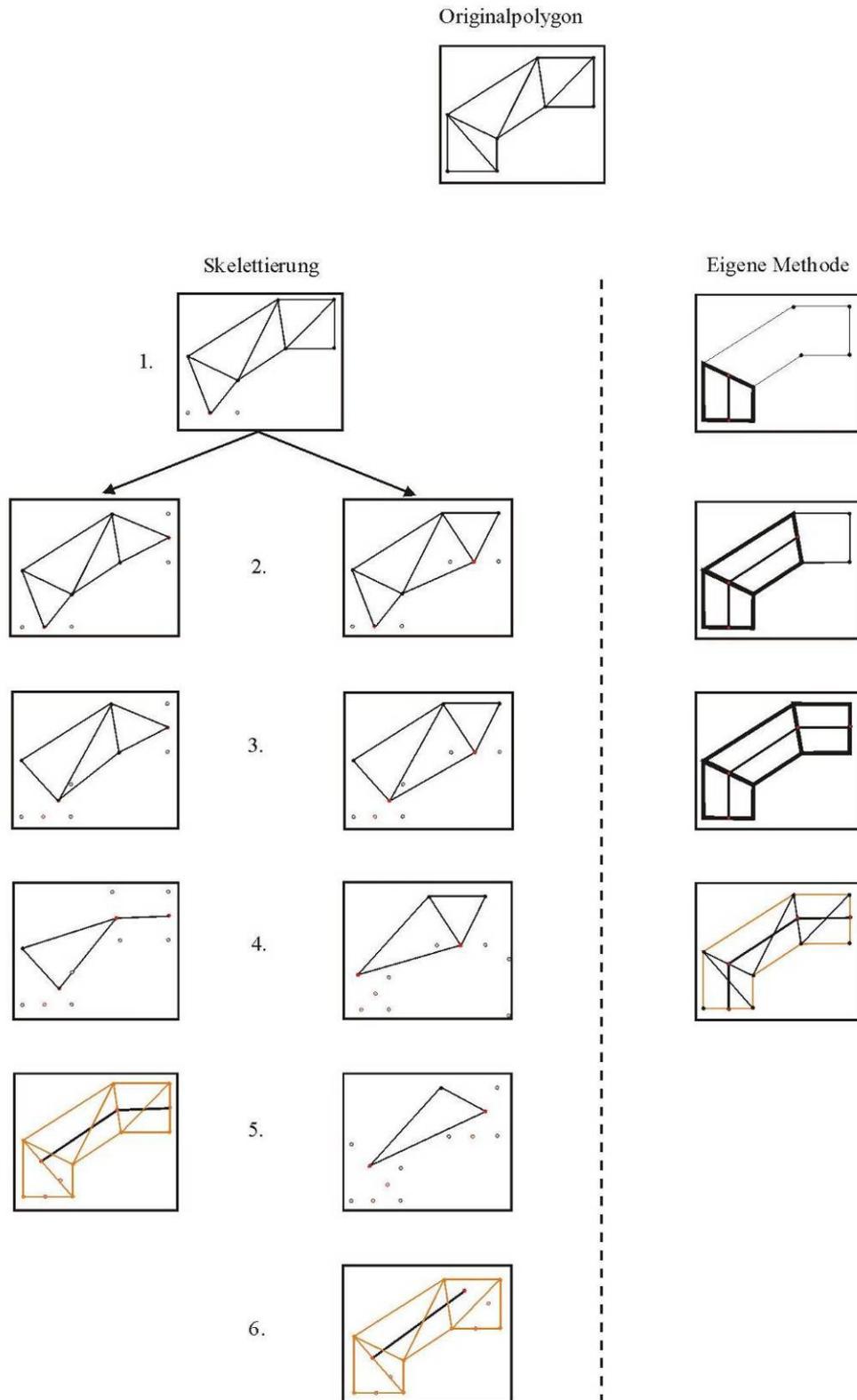


Abbildung A.0.1: Vergleich der Skelettierung nach Xuetao et al.[XTTZ01] und eigener Variante zur Bestimmung der Objektausrichtung



## **Selbständigkeitserklärung**

Hiermit versichere ich, Birgit Meisert, die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt zu haben. Alle Stellen, die ich wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften übernommen habe, sind als solche gekennzeichnet.

Magdeburg, 12.03.2008

---

Birgit Meisert