

# Prozedural unterstützte Generierung von Gebäudemodellen für interaktive Anwendungen

## Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers  
im Studiengang Computervisualistik

vorgelegt von  
Robert Schrader

Erstgutachter: Prof. Dr.-Ing. Stefan Müller  
(Institut für Computervisualistik, AG Computergraphik)

Zweitgutachter: Dr. Frank Detje  
IABG mbH, VG63 - Human Factors

Koblenz, im April 2007



## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	<input type="checkbox"/>	<input type="checkbox"/>
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	<input type="checkbox"/>	<input type="checkbox"/>

.....  
(Ort, Datum)

.....  
(Unterschrift)

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Generierte Stadt- und Gebäudemodelle in 3D</b>	<b>5</b>
2.1	Photogrammetrische Ansätze . . . . .	5
2.2	Prozedurale Ansätze . . . . .	7
2.3	Ausgewählte Verfahren und informationstechnische Umsetzungen	9
2.4	Zwischenfazit . . . . .	12
<b>3</b>	<b>Allgemeine Stadtgeographie</b>	<b>13</b>
3.1	Stadtbeginn und allgemeine Merkmale einer Stadt . . . . .	15
3.2	Stadtstruktur und -entwicklung . . . . .	15
3.3	Stadtgliederung . . . . .	18
3.4	Morphogenetische und städtebauliche Elemente . . . . .	23
3.5	Zwischenfazit . . . . .	26
<b>4</b>	<b>Stadtgrundriss</b>	<b>27</b>
4.1	Prozedurale Anforderungen und Ansätze für die Flächenunterteilung	29
4.1.1	Vorhandene Techniken . . . . .	30
4.1.2	Zwischenfazit . . . . .	34
4.2	Spatial Landmark Graph (SLG) . . . . .	35
4.2.1	Knoten . . . . .	36
4.2.2	Kanten . . . . .	37
4.2.3	Flächen und Pfade . . . . .	39
4.3	SLG: Flächenunterteilung . . . . .	40
4.3.1	Initialisierung . . . . .	41
4.3.2	Begrenzungs-Knoten platzieren . . . . .	41
4.3.3	Innere Knoten und Kanten bilden . . . . .	43
4.3.4	Außen liegende Flächen (Pfadsuche) . . . . .	44
4.3.5	Außen liegende Flächen (Pfadauswahl) . . . . .	46
4.3.6	Zwischenräume . . . . .	47
4.3.7	Überflüssige Knoten & Kanten entfernen . . . . .	50
4.3.8	Innen liegende Flächen . . . . .	51
4.3.9	Lage der Kanten bestimmen . . . . .	51
4.4	SLG: Parameter . . . . .	52
4.4.1	Anzahl der Begrenzungs-Knoten . . . . .	53
4.4.2	Winkel und Abstände . . . . .	54
4.4.3	Zwischenräume . . . . .	55
4.5	SLG: Hierarchiebildung . . . . .	56
4.6	SLG: Zusätzliche Manipulationen . . . . .	56
4.6.1	Kantenlisten ordnen . . . . .	56
4.6.2	Flächen und Kantenlisten schrumpfen . . . . .	57
4.6.3	Kanten splitten . . . . .	59
4.7	SLG: Mathematik . . . . .	59
4.8	Zwischenfazit . . . . .	61

<b>5</b>	<b>Stadtaufriss</b>	<b>63</b>
5.1	Prozedurale Anforderungen und Ansätze für 3D-Formen . . . . .	64
5.1.1	Vorhandene Techniken . . . . .	65
5.1.2	Zwischenfazit . . . . .	67
5.2	Grundriss-Erweiterung . . . . .	68
5.2.1	Einfügen von Stockwerken . . . . .	69
5.2.2	Platzierung von Eingängen . . . . .	69
5.2.3	Platzierung von Fenstern . . . . .	71
5.2.4	Platzierung von Treppenhäusern . . . . .	72
5.3	Aufriss-Generierung (3D-Modelle) . . . . .	72
5.3.1	Basis-Elemente und Indexed Geometry . . . . .	73
5.3.2	Texturierung und Punkt-Normalen . . . . .	78
5.3.3	Triangulierung . . . . .	82
5.3.4	Generierung der 3D-Modelle und Level-of-Detail . . . . .	83
5.4	Zwischenfazit . . . . .	87
<b>6</b>	<b>Stadtobjekte</b>	<b>89</b>
6.1	Stadt . . . . .	91
6.1.1	Parameter und vordefinierte Werte einer Stadt . . . . .	92
6.1.2	Allgemeine Stadtgeographische Parameter von Unterobjekten	93
6.2	Stadtteil . . . . .	95
6.2.1	Parameter und vordefinierte Werte eines Stadtteils . . . . .	95
6.2.2	Allgemeine stadtgeographische Parameter von Unterobjekten	96
6.3	Hausblock . . . . .	97
6.3.1	Parameter und vordefinierte Werte eines Hausblocks . . . . .	97
6.3.2	Allgemeine stadtgeographische Parameter von Unterobjekten	99
6.4	Grundstück . . . . .	100
6.4.1	Parameter und vordefinierte Werte eines Grundstücks . . . . .	100
6.4.2	Allgemeine stadtgeographische Parameter von Unterobjekten	104
6.5	Gebäude . . . . .	104
6.5.1	Parameter und vordefinierte Werte eines Gebäudes . . . . .	104
6.5.2	Allgemeine stadtgeographische Parameter von Unterobjekten	111
6.6	Stockwerk . . . . .	111
6.6.1	Parameter und vordefinierte Werte eines Stockwerks . . . . .	112
6.7	Wohnung . . . . .	114
6.7.1	Parameter und vordefinierte Werte einer Wohnung . . . . .	115
<b>7</b>	<b>Praktische Umsetzung: CITYLIFE</b>	<b>117</b>
7.1	Konzept, Aufbau und Funktionalitäten . . . . .	117
7.2	Benutzeroberfläche und Bedienung . . . . .	121
7.3	Verwendete 3D-Grafik-Engine . . . . .	124
7.4	Ergebnisbilder und Leistungsverhalten . . . . .	125
<b>8</b>	<b>Gesamtfazit</b>	<b>131</b>
8.1	Zusammenfassung . . . . .	131
8.2	Bewertung . . . . .	132
8.3	Ausblick . . . . .	134

## Abbildungsverzeichnis

1	Computergenerierte Nachbildung von Manhattan im Jahr 1933 für den Kinofilm <i>King Kong</i> (Quelle: Online unter <a href="http://www.cityofsound.com/blog/2005/12/">http://www.cityofsound.com/blog/2005/12/</a> , letzter Zugriff 10.04.07). . . . .	1
2	Szenen aus dem Computerspiel <i>Paradise City</i> (Quelle: Online unter <a href="http://www.paradisecity-thegame.com/">http://www.paradisecity-thegame.com/</a> , letzter Zugriff 10.04.07). . . . .	2
3	Stadtmodell aus digitalem Höhenmodell und Luftbildern (Quelle: [BrHa02]). . . . .	6
4	Prozedural generierte Bürogebäude (Quelle: [PaMu03]). . . . .	8
5	Ergebnisbild des photogrammetrischen Verfahrens nach BRENNER (Quelle: [ClBr00]). . . . .	9
6	Ergebnisbilder des prozeduralen Verfahrens nach MÜLLER (Quelle: [PaMu03]). . . . .	10
7	Ergebnisbilder des prozeduralen Verfahrens nach WONKA et al. (Quelle: [WoWiSi01]). . . . .	11
8	Ergebnisbilder des prozeduralen Verfahrens nach MÜLLER und WONKA et al. (Quelle: [MuWoHaUl05]). . . . .	11
9	Modell der Stadtregion nach BOUSTEDT (Quelle: [HeHe06]). . . . .	16
10	Das Ringmodell nach BURGESS (Quelle: [HeHe06]). . . . .	17
11	Das Sektorenmodell nach HOYT und das Mehrkerne-Modell nach HARRIS und ULLMAN (Quelle: [HeHe06]). . . . .	17
12	Stadtstrukturmodelle nach ALBERS (Quelle: [HeHe06]). . . . .	18
13	Grundformen des Dachs nach HEINEBERG (Quelle: [HeHe06]). . . . .	20
14	FNP der Stadt Darmstadt mit farblicher Unterscheidung der Nutzungstypen (Quelle: Online unter <a href="http://www.darmstadt.de/wirtschaft/stadtplanung/gesamstadt/">http://www.darmstadt.de/wirtschaft/stadtplanung/gesamstadt/</a> , letzter Zugriff 11.04.07). . . . .	22
15	Vertikale Flächennutzungsverteilung nach CARTER (Quelle: [HeHe06]). . . . .	23
16	Morphogenetische Ebenen (Quelle: Gerhard Curdes, <i>Stadtstruktur und Stadtgestaltung</i> . Stuttgart 1993. 2. Auflage 1997). . . . .	24
17	Auschnitte von Liegenschaftskarten (Quellen: [HaBe05] und Online unter <a href="http://staff.science.uva.nl/~leo/singel%2077/verkoop/">http://staff.science.uva.nl/~leo/singel%2077/verkoop/</a> , letzter Zugriff 11.04.07). . . . .	29
18	Gebäude-Grundrisse (Quellen: Online unter <a href="http://www.immquadrat.de/text/immobilienangebote/harsdoerfferplatz.html">http://www.immquadrat.de/text/immobilienangebote/harsdoerfferplatz.html</a> [©immquadrat Projekt GmbH] und <a href="http://de.wikipedia.org/wiki/Grundriss">http://de.wikipedia.org/wiki/Grundriss</a> , letzter Zugriff 11.04.07). . . . .	30
19	Entwicklung der Koch-Inseln mit einem L-System (Quelle: [PaMu03]). . . . .	31
20	BSP- und Slicing Tree (Quelle: [HHCGDCG01]). . . . .	32
21	Bin Packing und Floor Planing (Quellen: [YoYaWo01] und [ChFuLiRo02]). . . . .	33
22	Voronoi-Diagramm (Quelle: [ToKe06]). . . . .	34
23	Die vereinfachte Flächenunterteilung des SLG: a) Begrenzungsknoten und Strahlen, b) Innere Struktur, c) Generierte Teilflächen. . . . .	36
24	Eine Kante im SLG. . . . .	37

25	Kanten-Orientierungsproblematik im SLG. . . . .	38
26	Kanten-Lage im geographischen Zusammenhang am Beispiel einer Wohnung. . . . .	39
27	Verteilung von vier Begrenzungs-Knoten. . . . .	42
28	Verschieben von Begrenzungs-Knoten an konkaven Ecken. . . . .	43
29	Konstruktion der inneren Knoten und Kanten mit Hilfe von Schnittpunkten. . . . .	43
30	Gefundener Pfad auf dem Umriss und zwei mögliche Innepfade bei der Pfadsuche. . . . .	44
31	Zwischenräume für Straßen, Hausflur und Wohnungsflur. . . . .	47
32	Teilfläche reduziert auf vier Seiten. . . . .	47
33	Vorbereitung der Gebäudeplatzierung sowie geschlossene, halb-offene und offene Bauweise. . . . .	49
34	Grundrisse der Stufen Hausblock bis Wohnung mit nicht trivialen Grundflächen, erstellt mit CITYLIFE (siehe Kapitel 7). . . . .	62
35	CSG-Konstruktionsprozess (Quelle: [RaWa01]). . . . .	65
36	Delaunay-Triangulierung mit eingezeichneter Umkreisbedingung (Quelle: [AnVi01]). . . . .	66
37	Erstellung einer Fassade mit der Split-Grammar-Technik (Quelle: [MuVeUIVG01]). . . . .	67
38	Straight-Skeleton eines Polygons und die daraus entstehende 3D-Form (Quelle: [AiAu01]). . . . .	67
39	Geometrie eines Wand-Elements. . . . .	74
40	Geometrie eines Fenster-Elements. . . . .	75
41	Geometrie eines Eingang-Elements. . . . .	76
42	Geometrie eines Treppenhaus-Elements. . . . .	77
43	Geometrie eines Satteldach- und Zeldach-Elements. . . . .	77
44	Texturemap eines Gebäudes. . . . .	81
45	Beleuchtetes Polygon (mit Gouraud-Shading). . . . .	82
46	Triangulierung eines konkaven Polygons. . . . .	82
47	Platzierung der Innenwände. . . . .	86
48	Einfaches Gebäudemodell, erstellt mit CITYLIFE (siehe Kapitel 7). . . . .	88
49	Beispielhafte Wachstumsfunktion für die Anzahl von Räumen pro Stockwerksfläche. . . . .	107
50	UML-Klassendiagramm von CITYLIFE. . . . .	119
51	Die Benutzeroberfläche von CITYLIFE. . . . .	122
52	Konstruktionsprozess eines Hausblocks: a) Umriss-Eingabe, b) Grundriss-Generierung, c) Aufriss-Generierung, d) Interaktive Betrachtung der Ergebnisse. . . . .	125
53	Ein Gebäudemodell in vier LOD-Stufen. . . . .	126
54	Verschiedene Gebäudetypen. . . . .	126
55	Innere Konstruktion zweier Gebäude (bzgl. der Raumaufteilung und Größe von Innenräumen in Stockwerken). . . . .	127
56	Grund- und Aufriss einer Stadt mit ca. 1500 Gebäuden. . . . .	127
57	Verschiedene Betrachtungspositionen innerhalb eines großen Stadtmodells. . . . .	128
58	Generierter Grund- und Aufriss auf der Grundlage eines Luftbilds von Berlin aus Google Earth. . . . .	129

# 1 Einleitung

Die Erstellung und Visualisierung von dreidimensionalen Objekten ist der wichtigste und zugleich umfassendste Teilbereich der 3D-Computergrafik und -Modellierung. Eine besondere Herausforderung ist dabei die realistisch wirkende, digitale Repräsentation einer Stadt, die durch ihre Vielzahl an verschiedenen Strukturen und Formen, entstanden durch ökonomische, historische und kulturelle Entwicklungsprozesse, ein hoch komplexes System beschreibt. Um solch ein kompliziertes Gebilde zu erfassen und nachzubilden, sind in den meisten Fällen ein reales Vorbild, ein ausgeklügeltes Entwurfs-Modell und auch gestalterisches Geschick vonnöten.

Basierend auf der rasanten Entwicklung von Computer Hard- und Software werden digitale Stadt- und Gebäudemodelle bereits heute in vielen kommerziellen 3D-Anwendungen, wie Computerspielen, Serious Games, Navigationssystemen, Planungs- und Trainingssoftware als virtuelle, interaktive Umgebungen eingesetzt. Des Weiteren finden diese 3D-Modelle in der Forschung und Stadtplanung als hilfreiche Erweiterungen von 2D-Ansichten und in der Filmindustrie als komplette Stadtkulissen Verwendung (siehe Abb. 1 und 2).



Abbildung 1: Computergenerierte Nachbildung von Manhattan im Jahr 1933 für den Kinofilm *King Kong* (Quelle: Online unter <http://www.cityofsound.com/blog/2005/12/>, letzter Zugriff 10.04.07).

Die Herstellung bzw. Modellierung von visuell ansprechenden Stadt- und Gebäudemodellen ist hierbei immer ein sehr zeit- und kostenintensiver Vorgang, falls der Entwurfs- und Konstruktionsprozess manuell ausgeführt wird. Der Zeitfaktor könnte dabei zwar durch mehr Personal verringert werden, die Kosten würden allerdings noch weiter steigen. In aktuellen Projekten des Entertainmentbereichs ist es keine Seltenheit mehr, dass die Erstellung von virtuellen (Stadt-) Umgebungen Kosten von bis zu zwei Mannjahren verschlingt.

Aufgrund dessen gibt es seit einigen Jahren verstärkte Bemühungen, diese komplexen 3D-Modelle weitgehend automatisch zu generieren, um die Entwicklung der oben aufgeführten Anwendungen bzw. Produktionen schneller und kostengünstiger umzusetzen.



Abbildung 2: Szenen aus dem Computerspiel *Paradise City* (Quelle: Online unter <http://www.paradisecity-thegame.com/>, letzter Zugriff 10.04.07).

Diese Arbeit behandelt die automatische Erstellung bzw. Generierung von Stadt- und Gebäudemodellen und definiert als Grundlage die folgenden Problemstellungen und Anforderungen. Diese wurden so gewählt, dass eine Abgrenzung zu bereits vorhandenen automatisierten Verfahren möglich ist und um eine Gesamtlösung für einen so noch nicht behandelten aber häufig vorkommenden Anwendungsfall zu liefern:

- Die generierten Modelle werden in einer *3D-Echtzeit-Anwendung* als urbane Umgebung verwendet. Dabei ist insbesondere ihre effiziente Darstellung von Relevanz, um die eigentlichen Simulationen hinsichtlich der Computerleistung nicht negativ zu beeinflussen.  
Z.B. sind die Visualisierungen von Serious Games, die häufig in Stadtkulissen oder auch innerhalb von einzelnen Gebäuden ihre Trainings- und Ausbildungsinhalte vermitteln, von dieser Thematik betroffen.
- Die Stadt- und Gebäudemodelle sollen in kürzester Zeit mit keinen oder nur wenigen Benutzereingaben generiert werden, um vor allem die zügige Entwicklung von Prototypen und Demonstratoren gewinnbringend zu unterstützen.
- Die Gestaltung der Modelle muss nicht zwangsläufig nach realen Vorlagen erfolgen und sollte daher auch auf spezielle simulierte Situationen angepasst werden können. Zusätzlich ist es hilfreich, Karten, Übersichtspläne oder Skizzen von Gebäudekomplexen und Straßenzügen als Grundlage für den Konstruktionsprozess zu verwenden.
- Die Modelle werden mit Innenräumen generiert, um Benutzern das interaktive Navigieren und Handeln innerhalb von Gebäuden zu ermöglichen.

Aus diesen Problemstellungen und Anforderungen wurden die Ziele dieser Arbeit wie folgt abgeleitet:

- Allgemein soll ein Verfahren konzipiert, entwickelt und in einem System umgesetzt werden, mit welchem insbesondere interaktiv darstellbare Gebäudemodelle generiert werden können.
- Die Komplexität soll dabei von einzelnen Gebäuden bis hin zu ganzen Stadtgebilden reichen und dementsprechend skalierbar sein.
- Der gesamte Entstehungsprozess von Modellen soll durch den Benutzer auf allen Stufen durch geeignete Parameter gesteuert und kontrolliert werden. Dabei kann einfaches Kartenmaterial als Entwurfsvorlage verwendet werden.
- Für möglichst realistisch wirkende Ergebnisse soll eine wissenschaftlich begründetes Gestaltungskonzept in das System integriert werden. Dieses wird aus den Gesetzmäßigkeiten der *allgemeinen Stadtgeographie* abgeleitet.
- Die resultierenden Modelle können durch ihr Aussehen und ihre Form in allgemein gültige Gebäudetypen (Wohnen, Büro, Industrie usw.) unterschieden werden.
- Alle Gebäude werden mit Innenräumen versehen und gleichzeitig für die Echtzeit-Darstellung optimiert.

In Kapitel 2 dieser Arbeit werden die grundlegenden Ansätze zur Generierung von Stadt- und Gebäudemodellen beschrieben und bewertet, sowie für den Anknüpfungspunkt der restlichen Arbeit einer dieser Ansätze ausgewählt — der prozedurale Ansatz. Zusätzlich werden an dieser Stelle noch einige Verfahren zur Generierung von Stadtmodellen vorgestellt, um das allgemeine Potenzial dieser Ansätze zu veranschaulichen.

In Kapitel 3 werden die hier relevanten Forschungsbereiche der allgemeinen Stadtgeographie und des (deutschen) Städtebaus in angemessenem Umfang behandelt und als spätere Konstruktionsgrundlage der Modelle vereinfacht und angepasst. Alle somit gewonnenen Erkenntnisse lassen sich sinnvoll in das hier entwickelte Vorgehensmodell integrieren, wodurch der realistische Eindruck aller entstehenden 3D-Modelle entscheidend verbessert wird.

Kapitel 4 und 5 beschreiben die entwickelten Verfahren zur Grundriss- und Aufriss-Generierung<sup>1</sup> von Stadtmodellen und sind damit der wichtigste und zugleich umfangreichste Teil dieser Arbeit. Bemerkenswert ist hier, dass beide Verfahren zusammen eine Gesamtlösung für den horizontalen und vertikalen Aufbau von Städten und Gebäuden darstellen, mit der die gestellten Anforderungen vollständig erfüllt werden.

In Kapitel 6 werden die verschiedenen hierarchisch angeordneten Bestandteile eines Stadtgebiets vorgestellt und ihre Parameterauswahl und -berechnung behandelt. Dabei kommen insbesondere die in Kapitel 3 definierten Gesetzmäßigkeiten der allgemeinen Stadtgeographie und des (deutschen) Städtebaus zum Einsatz. Alle hier vorhandenen Stadtobjekte verwenden die Verfahren aus Kapitel 4 und

---

<sup>1</sup>Stadtgrundriss und -aufriss beschreiben die horizontale und vertikale Gestalt einer Stadt.

5 und bilden, zusammen mit ihren Parametern, das komplexe System eines städtischen Gebiets nach, womit die oben aufgeführten Problemstellungen in Lösungsansätze überführt werden.

In Kapitel 7 dieser Arbeit wird die praktische Umsetzung bzw. das Programm CITYLIFE konzeptionell beschrieben, dessen Benutzung beispielhaft erklärt und einige Ergebnisbilder präsentiert. Mit CITYLIFE wird u.a. gezeigt, dass sich das hier entwickelte Gesamtverfahren zur Generierung von Stadt- und Gebäudemodellen, neben einer prinzipiell möglichen softwaretechnischen Umsetzung, in einem benutzerfreundlichen, flexiblen und performanten Programm realisieren lässt.

Im letzten Teil (Kapitel 8) werden abschließend die Ergebnisse zusammengefasst, bewertet, sowie Vorschläge für zukünftige Arbeiten aufgeführt.

Zusammenfassend kann festgestellt werden, dass das hier entwickelte prozedural unterstützte *Gesamtsystem* eine sinnvolle Zusammenführung von stadtgeographischen Grundlagen und informationstechnischen Methoden darstellt, die in ihrer praktischen Umsetzung — CITYLIFE — eine benutzerfreundliche und effiziente Anwendung bilden, mit der realistische Stadt- und Gebäudemodelle für 3D-Echtzeit-Anwendungen generiert werden können.

## 2 Generierte Stadt- und Gebäudemodelle in 3D

Die Erstellung von 3D-Modellen kann i.A. als aufwändiger und zeit-intensiver Prozess bezeichnet werden. Dabei kommt normalerweise spezielle Software, sogenannte *3D-Modellierungsprogramme*, zum Einsatz, die aufgrund ihrer Komplexität nur von erfahrenen bzw. ausgebildeten Benutzern professionell und effektiv bedient werden können. Trotz fortschrittlicher Entwicklungen in diesem Bereich muss insbesondere die Form (Geometrie) von 3D-Modellen immer noch weitgehend manuell durch den Benutzer (*Modellierer*) eingegeben werden. Insbesondere stellen realistische Stadtmodelle mit ihrer großen Anzahl an verschiedenen Gebäuden eine beachtliche Herausforderung für den Modellierer dar.

Neben dieser manuellen Modellierung von digitalen Städten bzw. Gebäuden gibt es bereits eine Reihe von Ansätzen, diese komplexen 3D-Modelle mit Hilfe von Algorithmen und/ oder raumbezogenen Daten nahezu automatisch zu generieren.

Dafür werden in den folgenden Unterkapiteln zunächst die grundlegenden vorhandenen Ansätze zur Generierung von 3D-Stadtmodellen —

- **photogrammetrisch** und
- **prozedural**

— beschrieben und anschließend ausgewählte Techniken und informationstechnische Umsetzungen vorgestellt.

Warum der *prozedurale Ansatz* für diese Arbeit gewählt wurde, wird in Kapitel 2.4 begründet.

### 2.1 Photogrammetrische Ansätze

Die photogrammetrischen Ansätze zur Generierung von Stadtmodellen basieren zunächst immer auf der Auswertung von geologischen, raumbezogenen Messdaten (*Geoinformationen* bzw. *Geodaten*), die im entsprechenden Stadtgebiet gesammelt wurden. Liegen genügend und geeignete Informationen vor, können aus dieser Datenbasis im eigentlichen Verfahren 3D-Modelle abgeleitet werden. Dieser Ansatz versucht dabei *dreidimensionale Abbilder* aus real existierenden Tatsachen zu erstellen (vgl. Abb. 3).

Ein photogrammetrischer Ansatz kann vereinfacht in folgende Einzelschritte unterteilt werden:

#### 1. Datensammlung

Die Sammlung von verwendbaren Geodaten ist immer zeit- und vor allem kosten-intensiv. Meistens werden die Daten von speziellen Kameras und Sensoren in Flugzeugen/Helikopter, Fahrzeugen oder auch Satelliten gesammelt. Zusätzlich müssen diese Aufnahmen unter bestimmten Bedingungen (Blickwinkel, Auflösung usw.) erstellt werden, um überhaupt von einem photogrammetrischen Verfahren verwendet werden zu können [MaLa01]. Zur Vollständigkeit sollen hier einige weit verbreitete Datenquellen angegeben werden:

- *Luftbilder* werden meistens für den allgemeinen Aufbau des städtischen Gebiets verwendet. Bebaute Flächen können durch ihre



Abbildung 3: Stadtmodell aus digitalem Höhenmodell und Luftbildern (Quelle: [BrHa02]).

Dachstrukturen dabei von Straßen unterschieden werden und somit die Lage und Größe von Gebäuden bestimmt werden.

- *LIDAR (light detection and ranging)* bezeichnet die Abtastung von Gelände mit Hilfe von Laserstrahlen, wodurch z.B. Höheninformationen der Gebäude gewonnen werden können (auch als Digitale Oberflächenmodelle (*DOM, DSM oder DEM*) bezeichnet [ClBr00])<sup>2</sup>.
- *Bodennahe 3D-Laserscans und Photographien (terrestrische Bilder)*<sup>2</sup> können bereits aus (langsam) bewegten Fahrzeugen aufgenommen und für den Aufbau und das Aussehen von Gebäudefassaden verwendet werden.

## 2. Datenaufbereitung

Für die weitere Verarbeitung vorhandener Daten müssen diese, z.B. aufgrund von zu geringen Auflösungen, Verzerrungen oder allgemein fehlender Informationen, aufbereitet werden. Im günstigen Fall liegen nach diesem Schritt räumliche Geodaten als Vektoren oder dreidimensionale Punktwolken vor.

## 3. Modell-Generierung

Für die Generierung von Gebäudemodellen werden z.B. Dachstrukturen erstellt, in die Höhe verschoben und unter ihnen die Fassaden der Gebäude als einfache Blöcke eingefügt [VeDev01]. Sind 3D-Punktwolken vorhanden, können aus diesen Gebäude-Oberflächen abgeleitet und direkt visualisiert werden.

Eine detaillierte Beschreibung von photogrammetrischen Ansätzen erfolgt in [ClBr00], [LaFe01], [MaLa01] und [Hoeng01].

<sup>2</sup>3D-Scans produzieren i.A. enorme Datenmengen, die erst durch aktuelle Hardware gespeichert und effizient behandelt werden können.

### Vorteile des photogrammetrischen Ansatzes

- Durch diesen Ansatz entstandene Stadtmodelle repräsentieren (bei gegebener Qualität) ansprechende Abbilder realer Stadtgebiete.
- Die Gestaltung von Form und Oberflächen der Gebäude wird im allgemeinen Fall vollständig von Geodaten und Fotos festgelegt. D.h. für die Umsetzung, dass nur die korrekte Interpretation der vorhandenen Informationen beachtet werden muss und keine weiteren Verfahren für das realistische Aussehen der Modelle entwickelt werden müssen.

### Nachteile des photogrammetrischen Ansatzes

- Die größte Schwierigkeiten bei photogrammetrischen Ansätzen ist die Gewinnung von Geodaten des behandelten Geländes. Selbst für eine geringe bis mittlere Qualität der 3D-Modelle müssen diese Informationen aus verschiedenen Datenquellen, z.B. LIDAR-Scans und Luftbilder, zusammengesetzt werden. Durch den enormen finanziellen Aufwand, der bei der Gewinnung dieser Gebiets-Aufnahmen entsteht<sup>3</sup>, muss im Normalfall auf vorhandene (staatliche) Datenbestände zurückgegriffen werden. Allerdings sind Geoinformationen von Städten (in Deutschland) entweder nicht vorhanden oder teilweise nur "direkt an den Stellen und Ämtern verfügbar, wo sie auch bearbeitet werden. In vielen Stadtverwaltungen gibt es keinen zentralen Datenbestand" [MaLa01].
- Neben der schwierigen Beschaffung geeigneter Geodaten ist auch deren Qualität und einheitliche Form ein großes Problem:
  - Es existieren keine allgemeinen Dateiformate für die Daten von Gelände-Scans oder anderen Geodaten, wodurch das Verwenden verschiedener Quellen und das Austauschen von Datenbeständen erschwert wird.
  - Die vorhandenen Verfahren zur Datenaufbereitung von Bildern sind aufgrund von geringen Auflösungen und Bildfehlern meistens nicht robust, so dass zusätzliche *manuelle* Bearbeitungsschritte notwendig sind.
  - Speziell die Luftbilder sind je nach Flughöhe und verwendeter Kamera unterschiedlich verzerrt, bzw. werden Gebäude an den Bildrändern nicht mehr senkrecht von oben erfasst, was die spätere Auswertung zusätzlich belastet.
- Die hier gesichteten photogrammetrischen Verfahren sind vollständig abhängig von Geoinformationen, d.h. ohne Datenbasis können keine Modelle generiert werden.

## 2.2 Prozedurale Ansätze

Die prozeduralen Ansätze verwenden zur Struktur- und Formgebung der Stadtmodelle im Gegensatz zu den photogrammetrischen Ansätzen keine Geodaten

---

<sup>3</sup>Für diese Arbeiten müssen normalerweise spezialisierte Unternehmen beauftragt werden, die mit geeigneter Sensortechnik ausgerüstet sind und ggf. Zugang zu Luftfahrzeugen besitzen.

und visualisiert diese, sondern erstellen 3D-Modelle allein auf der Basis von computergenerierten Daten und Algorithmen. Wie realistisch bzw. realitätsnah die entstehenden Stadtmodelle dabei sind, wird somit durch die hierfür entwickelten mathematische Beschreibungen und Berechnungsvorschriften bestimmt.

Durch die Abhängigkeit von formalen bzw. mathematischen Beschreibungen der Realität werden mit dem prozeduralen Ansatz, anstatt direkter Abbilder, (u.U. ansprechende) Annäherungen zu realen Städten bzw. Gebäuden produziert (vgl. Abb. 4).

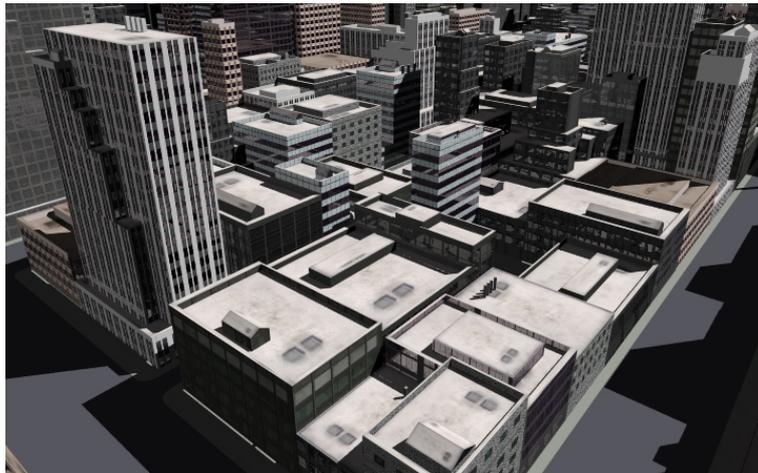


Abbildung 4: Prozedural generierte Bürogebäude (Quelle: [PaMu03]).

Bei einem prozeduralen Verfahren steht also die mathematische Beschreibung, Simulation und Abbildung der Realität bzw. von realen Gegebenheiten (Form, Struktur, Wachstum/Entwicklung, Reaktion usw.) im Mittelpunkt. Z.B. konnten bereits durch fraktale Konzepte oder Lindenmayer Systeme (siehe Kapitel 4.1.1) die Formen von Pflanzen und im letzteren Fall auch von Gebäuden und Straßennetzen realistisch nachgebildet werden [PaMu03], [MuWoHaUl05].

#### **Vorteile des prozeduralen Ansatzes**

- Die problematische Sammlung und Kombination von Geoinformationen entfällt bei diesem Ansatz.
- Durch geeignete Formalismen bzw. Algorithmen sind trotz fehlender realer Bezüge realistisch wirkende Ergebnisse möglich.

#### **Nachteile des prozeduralen Ansatzes**

- Realistische *Abbilder* von Städten können *allein* durch prozedurale Konzepte nicht erstellt werden<sup>4</sup>. Dies gilt insbesondere für komplexe, historisch gewachsene Straßennetzstrukturen (z.B. von Berlin, München, Hamburg usw.).

---

<sup>4</sup>Nach Sichtung der vorhandenen Quellen und Wissensstand des Autors.

- Allgemein gültige Formalismen für urbane Strukturen und Formen sind aufgrund ihrer Vielfältigkeit und Unregelmäßigkeit schwer zu entwerfen.
- Höherer Realismus der Ergebnisse resultiert meistens in komplexeren mathematischen Beschreibungen und Berechnungsschritten.

### 2.3 Ausgewählte Verfahren und informationstechnische Umsetzungen

Im Folgenden werden ausgewählte photogrammetrische und prozedurale Verfahren und ihre jeweiligen informationstechnischen Umsetzungen zur Generierung von Stadtmodellen vorgestellt:

#### C. BRENNER, 2000 (photogrammetrisch)

In seiner Arbeit entwickelt BRENNER ein System zur "dreidimensionalen Gebäuderekonstruktion", das vorhandene Daten von DOM (digitalen Oberflächenmodellen) und Kartenmaterial (2D-GIS-Daten) verarbeiten kann. Die DOM-Daten werden dabei durch "flugzeuggetragene Laserscanning-Systeme" und die 2D-GIS-Daten aus der ALK<sup>5</sup> oder der digitalisierten Deutschen Grundkarte gewonnen. Die Texturen der entstehenden Modelle werden aus Luft- und terrestrisch gewonnenen Bildern extrahiert [CIBr00].

Durch die Kombination von mehreren Datenquellen werden in diesem Verfahren insbesondere die Probleme der Datensammlung und -aufbereitung in photogrammetrischen Verfahren aufgezeigt. Zusätzlich werden für diese Problematik aber Lösungswege entworfen und umgesetzt.

Mit diesem Verfahren können aus den beschriebenen Datenquellen qualitativ ansprechende 3D-Stadtmodelle automatisch generiert und dargestellt werden (vgl. Abb. 5).



Abbildung 5: Ergebnisbild des photogrammetrischen Verfahrens nach BRENNER (Quelle: [CIBr00]).

---

<sup>5</sup>Automatisierte Liegenschaftskarte.

**P. MÜLLER, 2000 (prozedural)**

In der Arbeit von MÜLLER (2000) werden durch erweiterte Lindenmayer Systeme<sup>6</sup> realistisch wirkende Straßennetze und Gebäudemodelle generiert. Zusätzlich ist das Verfahren beliebig skalierbar, wodurch auch sehr große Stadtgebiete erstellt werden können [PaMu03].

Die entstehenden Straßennetze wurden insbesondere an die quadratische Straßennetz-Topologie von Manhattan (New York) angelehnt, wobei durch das entwickelte Verfahren auch weitere Strukturmuster generiert werden können (vgl. Abb. 6).

Die Gebäudemodelle werden bei diesem Verfahren innerhalb der Hausblock-Parzellen des erstellten Straßennetzes platziert und können die Formen von (Büro-)Hochhäusern repräsentieren. Hierbei ist zu beachten, dass die Unterteilung der von Straßen eingeschlossenen Hausblöcke durch einen einfachen Algorithmus (*BSP*, siehe Kapitel 4.1.1) durchgeführt wird und nur für möglichst rechteckige nicht konkave Flächen realistische Ergebnisse liefert.

Die Darstellung der Stadtmodelle wird durch das Modellierungsprogramm *Autodesk® Maya®* übernommen, wobei statische Bilder, aber auch vordefinierte Animationsfilme (Kameraflüge durch und über die Stadt) erstellt wurden (vgl. Abb. 6).

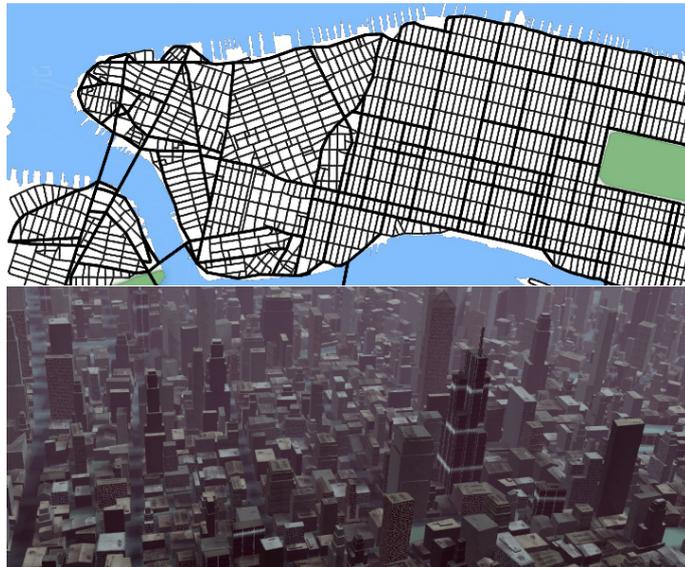


Abbildung 6: Ergebnisbilder des prozeduralen Verfahrens nach MÜLLER (Quelle: [PaMu03]).

**P. WONKA et al., (prozedural)**

WONKA et al. beschreiben in ihrer Arbeit ein Verfahren (Split Grammar<sup>7</sup>) zur

<sup>6</sup>Lindenmayer Systeme (L-Systeme) sind Zeichketten-basierte Ersetzungsgrammatiken (siehe Kapitel 4.1.1 und [PaMu03]).

<sup>7</sup>Die Split Grammar ist eine Ersetzungsgrammatik ähnlich den L-Systemen (siehe Kapi-

prozeduralen Generierung von Fassaden-Strukturen.

Die dabei entstehenden 3D-Gebäude-Geometrien sind detailliert und können durch entsprechende Parameter (Regelsätze) flexibel erweitert bzw. modifiziert werden. Dabei können verschiedene Gebäudetypen (z.B. Büro- und Familienhäuser) mit der Split Grammar umgesetzt werden [WoWiSi01].

Mit diesem Verfahren können Gebäude-Fassaden in hoher Qualität erzeugt und auch in Echtzeit<sup>8</sup> dargestellt werden (vgl. Abb. 7).



Abbildung 7: Ergebnisbilder des prozeduralen Verfahrens nach WONKA et al. (Quelle: [WoWiSi01]).

#### **P. MÜLLER, P. WONKA et al., 2006 (prozedural)**

Die oben beschriebenen prozeduralen Ansätze wurden 2006 von MÜLLER, WONKA et al. in einer gemeinsamen Arbeit zusammengeführt, um die jeweiligen Vorzüge der oben beschriebenen Verfahren zu kombinieren. Es werden folglich große Stadtmodelle mit einer möglichst rechteckigen Straßennetz-Struktur und einer Vielzahl von unterschiedlichen, hoch-detaillierten Gebäuden erstellt [MuWoHaU105].



Abbildung 8: Ergebnisbilder des prozeduralen Verfahrens nach MÜLLER und WONKA et al. (Quelle: [MuWoHaU105]).

tel 5.1.1)

<sup>8</sup>In geringer Anzahl oder durch Einsatz von Level-Of-Detail-Stufen.

Die so entstehenden 3D-Modelle konnten zum Zeitpunkt dieser Arbeit allerdings noch nicht in Echtzeit dargestellt werden. Die erstellten statischen Bilder stellen jedoch beeindruckende Ergebnisse dar (vgl. Abb. 8).

## 2.4 Zwischenfazit

Aufgrund der allgemein gültigen Probleme von photogrammetrischen Verfahren bei der Datensammlung und -aufbereitung

- keine flächendeckende Verfügbarkeit von Geodaten,
- eigene Erstellung oder Beschaffung dieser Daten ist zu kosten-intensiv und
- es existieren keine einheitlichen Dateiformate für Geoinformationen

bietet sich prinzipiell eine prozedurale Vorgehensweise für die behandelte Thematik an. Die eigentliche Entscheidung für einen prozeduralen Ansatz wurde aber erst durch die nachfolgenden Beweggründe bzgl. der hier definierten Ziele und Problemstellungen gefällt:

- Es sollen, insbesondere für Serious Games oder ähnliche Echtzeit-Simulationen, auch nicht real existierende städtische Gebiete bzw. Gebäude generiert werden, um z.B. Benutzer auch in Szenarien agieren zu lassen, die auf bestimmte Trainings- oder Ausbildungs-Situationen abgestimmt wurden.
- Die Abhängigkeit von Geodaten bzw. großen Datenmengen, wie sie bei photogrammetrischen Verfahren i.A. der Fall ist, soll aus folgenden Gründen vermieden werden:
  - Stadt- oder Gebäudemodelle sollen schnell (ggf. sogar in Echtzeit) und ohne langwierige Preprocessing-Verfahren<sup>9</sup> generiert werden.
  - Die Gestaltung und Vielfalt der Modelle soll nicht oder nur teilweise eingeschränkt werden.
  - Es soll möglich sein, das entwickelte System ohne zusätzliche Datenbankbindung oder Datenträger weiterzugeben.
- Die generierten Gebäude sollen begehbar sein. Allerdings sind flächendeckende Geodaten für Innenräume in keiner Weise vorhanden bzw. können nicht in einer verwendbaren Form bereitgestellt werden, wodurch hier nur ein prozeduraler Ansatz dieser Anforderung gerecht werden kann.

Aus diesen Gründen wird in dieser Arbeit ein prozedurales Verfahren auf der Basis von Algorithmen entwickelt, das in den nachfolgenden Kapiteln eingehend beschrieben wird.

---

<sup>9</sup>Verfahren zu Vorverarbeitung (von Daten).

### 3 Allgemeine Stadtgeographie

Das im Rahmen dieser Arbeit entwickelte Verfahren besteht ausschließlich aus prozeduralen Konzepten (siehe Kapitel 2.4). Dadurch werden keine Geoinformationen als Datenbasis verwendet und anschließend visualisiert, sondern die Stadtmodelle allein durch Algorithmen geformt und gestaltet. Diese mathematischen Berechnungsschritte und insbesondere ihre Steuerung sollen mit dem hier beschriebenen wissenschaftlichen Bereich der allgemeinen Stadtgeographie verknüpft werden, um möglichst realistische Stadt- und Gebäudemodelle zu erzeugen.

“Die Stadtgeographie ist eine der traditionsreichsten und wichtigsten Teildisziplinen der Anthropogeographie. Ihr Ziel ist die raumbezogene Erforschung städtischer Strukturen, Funktionen, Prozesse und Probleme“ [HeHe06]. Mit der damit verbundenen Analyse von existierenden städtischen Gebieten sowie den daraus entstehenden Modellen und Ansätzen, bietet die (allgemeine) Stadtgeographie eine geeignete theoretische Ausgangsbasis für die Erstellung von virtuellen Stadt- und Gebäudemodellen.

Die Hauptforschungsrichtungen der allgemeinen Stadtgeographie lassen sich in folgende Bereiche unterscheiden<sup>10</sup>:

- **Morphogenetische Stadtgeographie (Stadt morphologie)**

Die Stadtmorphologie “hat die Analyse der Grundriss- und Aufrissgestaltung der Städte sowie die Genese der Formelemente zum Forschungsgegenstand (morphogenetisch = gestaltbildend).”

Des weiteren zählen Stadterneuerung, Stadterhaltung und Stadtimagepflege zu den Forschungsgebieten der Stadtmorphologie.

- **Funktionale Stadtgeographie**

I.A. untersucht die funktionale Stadtgeographie “sog. funktionale Raumeinheiten innerhalb der Städte (z.B. City, Wohnviertel, Industrie-, und Gewerbegebiet) und deren raumzeitliche Veränderung”.

Insbesondere für die Stadt- und Kommunalplanung können somit *funktionale Stadtkartierungen* (im Folgenden als *Flächennutzungspläne* bezeichnet) erstellt werden.

- **Zentralitätsforschung**

Die Zentralitätsforschung basiert hauptsächlich auf der *Theorie der Zentralen Orte* von CHRISTALLER (1933), die “die funktionalen Stadt-Land-Beziehungen, d.h. vor allem die zentralörtlichen Verflechtungen (oder Zentralität), in den Vordergrund stadtgeographischer Analyse” stellt.

- **Städtesystemforschung**

Diese Forschungsrichtung wurde im Wesentlichen aus der Zentralitätsforschung entwickelt. Bei der Städtesystemforschung sind “Bestandsaufnahmen regionaler Städtesysteme, d.h. vor allem die Beziehungen zwischen den Städten [...] vor allem auch die Entstehung regionaler oder nationaler Städtesysteme” von zentraler Bedeutung.

Dabei wird heute der Forschungsschwerpunkt insbesondere auf den Ausbau von *Straßennetzen* verlagert.

---

<sup>10</sup>Alle folgenden Zitate wurden aus [HeHe06] entnommen.

- **Kulturgenetische Stadtgeographie**

Die kulturgenetische Stadtgeographie untersucht "kulturraumspezifische Unterschiede u.a. der Urbanisierungsprozesse (Verstädterung) oder etwa der inneren Gliederung der Städte."

Mit dieser Betrachtungsweise konnten bereits "Modellvorstellungen sog. kulturgenetischer oder kulturraumspezifischer Stadttypen" entwickelt werden, die für "vielfältige interkulturelle Vergleiche der differenzierten Stadtentwicklungs- und Verstädterungsprozesse" verwendet werden können.

- **Sozialgeographische Stadtforschung**

Die heutige sozialgeographische Stadtforschung beschäftigt sich "mit allen üblichen Grundfunktionen innerhalb von städtisch geprägten Räumen, d.h. mit dem 'Arbeiten' (z.B. Berufspendelverkehr), 'Sichversorgen' (z.B. schichtenspezifisches Einkaufsverhalten), 'Sichbilden' (z.B. sozialgruppenspezifische Beteiligung an bestimmten Schularten) sowie 'Sicherholen' (z.B. Freizeitverhalten im Wohnungsumfeld), bezogen auf die Aktivitäten von sozialen Gruppen [...] in städtischen Siedlungen."

Auf Grundlage dieses Forschungsgebiets können u.a. Planungen, Statistiken oder Entwicklungen von Städten unter sozial-räumlicher Sichtweise erstellt und analysiert werden.

- **Verhaltens- und handlungsorientierte Stadtgeographie**

"Die verhaltensorientierte [...] Stadtgeographie beschäftigt sich insbesondere mit der Wahrnehmung und Bewertung städtischer Strukturen und Standorte [...] sowie mit den Zusammenhängen zwischen Raumwahrnehmung/-bewertung [...] und raumrelevanten Verhalten spezieller Gruppen".

Bei den entstandenen Arbeiten wird der Mensch (als Individuum) mit seinem Handeln und seiner Wahrnehmung im städtischen Zusammenhang als Mittelpunkt betrachtet, wodurch ein *handlungstheoretischer Ansatz* für Stadtplanung und -analyse verwendet werden kann.

- **Angewandte Stadtgeographie**

Die angewandte Stadtgeographie ist eine "stärker planungs- oder praxisbezogene Arbeitsrichtung" zu deren Aufgabenfeld u.a. vorbereitende Untersuchungen für die Stadterneuerung sowie Analysen zur Verkehrsberuhigung in Wohngebieten zählen.

Mit Hilfe dieser Hauptforschungsrichtungen der Stadtgeographie werden in den folgenden Unterkapiteln Ansätze und Methoden zur *Strukturierung* und *Gliederung* von Städten beschrieben, die mit dem hier entwickelten prozeduralen Ansätzen vereinfacht, aber durchaus sinnvoll verknüpft werden können. Für die Stadtgliederung sollen zusätzlich die *deutsche Baunutzungs- und Planzeichnungsverordnung* einbezogen werden, um insbesondere die Verwendung konkreter und umsetzbarer *Maßzahlen* zu ermöglichen [BauNVO93], [PlanzV90].

Die Behandlung von weiteren stadtgeographischen Techniken und Ansätzen, wie z.B. Perzeptions-, Industrialisierungs-, oder Globalisierungsprozesse, würden im Rahmen dieser Arbeit zu weit führen.

In den nachfolgenden Unterkapiteln wird zunächst der Stadtbegriff behandelt und die allgemeinen Merkmale einer Stadt im stadtgeographischen Zusammenhang definiert.

Des weiteren werden Eigenschaften und theoretische Modelle von Stadtstruktur, -entwicklung und -gliederung mit Hilfe der allgemeinen Stadtgeographie und des (deutschen) Städtebaus beschrieben.

### 3.1 Stadtbegriff und allgemeine Merkmale einer Stadt

Der Stadtbegriff lässt sich nach HEINEBERG nicht eindeutig definieren, so dass eine Stadt je nach wissenschaftlicher Betrachtungsweise<sup>11</sup> unterschiedlich definiert werden kann [HeHe06]. Generell kann eine Stadt aber als *Agglomerationsraum* bzw. ein "verstädtertes Gebiet mit einer gewissen Kernbildung, einer bestimmten Flächenausdehnung und einer größeren Mindestbevölkerungszahl" bezeichnet werden [HeHe06].

Für diese Arbeit soll der Stadtbegriff daher vereinfacht durch folgende Merkmale definiert werden:

- Eine Stadt ist eine *größere Siedlung* mit einer bestimmten Einwohnerzahl. Dabei kann eine Stadt in **Stadtgrößenklassen**, bzw. nach deutscher Statistik in **statistische Stadttypen**, gemäß Tabelle 1 eingeordnet werden.

Stadttyp	Einwohnerzahl
Landstadt	2.000 – 5.000
Kleinstadt	> 5.000 – 20.000
Mittelstadt	> 20.000 – 100.000
Großstadt	> 100.000

Tabelle 1: Stadtgrößenklassen nach deutscher Statistik.

- Es gibt eine deutliche *innere Gliederung* in **Wohn-, Gewerbe-, Industrie- und Hauptgeschäftsviertel** (Innenstadt/City).
- Bebauungs- und Bevölkerungsdichte sind im Stadtzentrum hoch und nehmen in Richtung Stadtrand ab, wodurch mindestens drei (meistens konzentrische) *Stadtregionen bzgl. ihrer Zentralität* unterschieden werden können und aus dem Modell der Stadtregion von BOUSTEDT (1970) abgeleitet werden (vgl. Abb. 9):
  - *Stadtkern* (Kern- oder Innenstadt, City-Bereich, räumlicher Mittelpunkt der Stadt)
  - *Stadtmantel* (Kerngebiet bis verstädterte Zone)
  - *Stadtrand* (suburbane Zone, Vororte)

### 3.2 Stadtstruktur und -entwicklung

Die Theorien und Modelle der Stadtstruktur und -entwicklung beschäftigen sich insbesondere mit dem übergeordneten, räumlichen Aufbau einer Stadt bzgl. ihrer

<sup>11</sup>Je nach Ausprägung von räumlicher Ausdehnung, Bebauungs- und Bevölkerungsdichten, Wachstum und Entwicklungen, Gebäudehöhen, historischem Hintergrund, Verwaltungseinheiten usw.

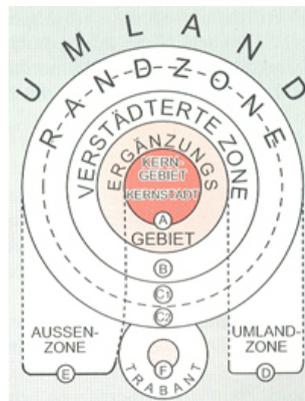


Abbildung 9: Modell der Stadtregion nach BOUSTEDT (Quelle: [HeHe06]).

(historischen) Entwicklung. Im Laufe der Zeit sind dazu eine Vielzahl von Arbeiten entstanden, die Stadtstruktur und -entwicklung aus unterschiedlichen Sichtweisen begründen [HeHe06], [ErRa02].

Für diese Arbeit bietet es sich (auch aus zeitlichen Gründen) an, nur diejenigen Modelle zu betrachten,

- die ein städtisches Gebiet ggf. vereinfacht, aber dafür allgemeingültig strukturieren (z.B. für Lage von Wohn-, Gewerbe- oder Industriegebieten) und/oder
- Angaben über den Aufbau des Verkehrs- bzw. Straßennetzes liefern (Anordnungsmuster von Stadtteilen und Hausblöcken, z.B. Schachbrettmuster).

Aufgrund dessen werden im Folgenden die weit verbreiteten Strukturmodelle der *Chicagoer Schule* sowie die Struktur- und Leitbilder des *Städtebaus* vorgestellt.

- **Sozialökonomische Strukturmodelle** (der Chicagoer Schule)

Durch die Arbeiten von BURGESS (1925/1929), HOYT (1939) und HARRIS & ULLMAN (1945) sind drei theoretische Modelle zur Beschreibung der Stadtstruktur und -entwicklung entstanden, die bis heute eine breite Anerkennung genießen. Alle drei Modelle werden dem komplexen Bereich der Stadtgeographie zwar nicht vollständig gerecht, allerdings können durch sie Städte in *sozialräumliche* und *funktionale Strukturen* eingeteilt werden:

- **Ringmodell** nach BURGESS (1925/1929)  
In diesem Modell wird eine Stadt u.a. aufgrund der "Hypothese konzentrischen Wachstums" in mehrere Ringzonen um das Stadtzentrum eingeteilt [HeHe06] (vgl. Abb. 10).
- **Sektorenmodell** nach HOYT (1939)  
Das Stadtstruktur des Sektorenmodells basiert auf den Annahmen, dass sich homogene soziale Räume entlang von Hauptverkehrswegen in Abhängigkeit der Mietpreis-Entwicklungen bilden bzw. verschieben (vgl. Abb. 11).
- **Mehrkerne-Modell** nach HARRIS und ULLMAN (1945)  
Das Mehrkerne-Modell berücksichtigt die zentralörtlichen Funktionen

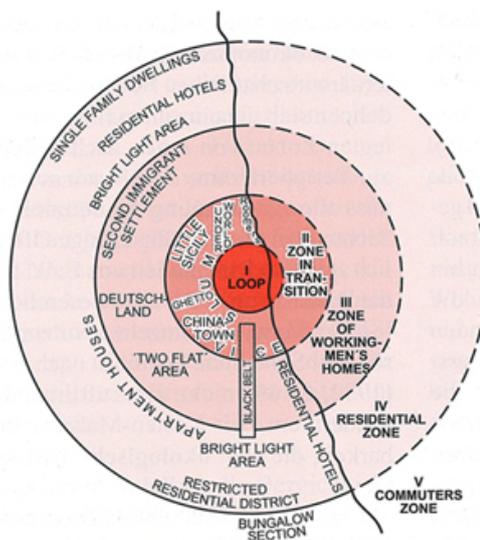


Abbildung 10: Das Ringmodell nach BURGESS (Quelle: [HeHe06]).

von Städten. Dabei wird davon ausgegangen, dass mit wachsender Größe einer Stadt auch "die Zahl und Spezialisierungen ihrer sog. Kerne (Stadtmitte, [...] Shopping-Center, [...] kleine Industriezentren)" [HeHe06] ansteigt (vgl. Abb. 11).

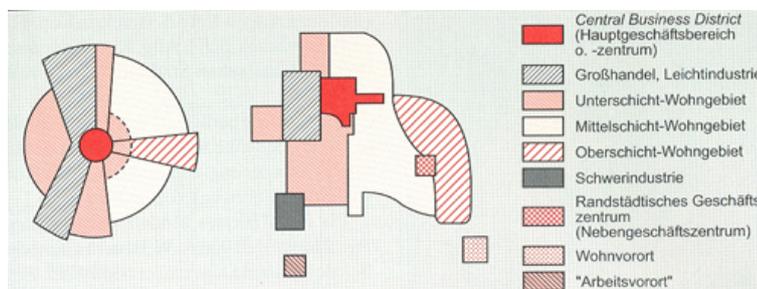


Abbildung 11: Das Sektorenmodell nach HOYT und das Mehrkerne-Modell nach HARRIS und ULLMAN (Quelle: [HeHe06]).

- **Strukturmodelle des Städtebaus**

Aus dem vielschichtigen Bereich der Stadtplanung bzw. des Städtebaus sollen hier die Arbeiten von ALBERS (1974) erwähnt werden, mit denen u.a. eine schematische Topologie von Stadtstrukturmodellen entwickelt wurde<sup>12</sup> (vgl. Abb. 12). Mit Hilfe dieses Ansatzes werden die folgenden allgemeinen Straßennetzstrukturen unterschieden:

- *Orthogonales Grundrissmuster: Rechteck*

<sup>12</sup>In dem von ALBERS entwickelten Strukturmodellen werden z.B. Raster-, Stern-, Kamm- oder Satelliten-Topologien definiert. Eine genauere Beschreibung befindet sich in [HeHe06].

- Orthogonales Grundrissmuster: Quadrat
- Orthogonales Grundrissmuster: Rippen
- Radiales Grundrissmuster
- Ring / Konzentrisches Grundrissmuster
- Sackgassengrundriss
- Diagonales Grundrissmuster
- Radialkonzentrisches / Ringradiales Grundrissmuster

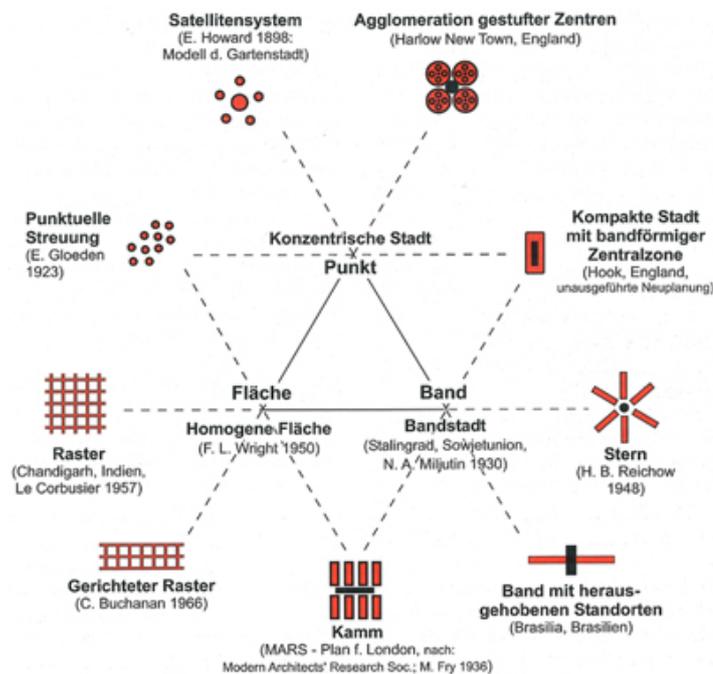


Abbildung 12: Stadtstrukturmodelle nach ALBERS (Quelle: [HeHe06]).

Mit den beiden Ansätzen der Sozialökonomie und des Städtebaus können bereits einfache aber realistisch wirkende Stadtstrukturen entworfen werden. Eine zugehörige prozedurale Umsetzung wäre, aufgrund der hier erfolgten Vereinfachungen, ebenfalls möglich<sup>13</sup>.

Aus zeitlichen Gründen konnten die Modelle der Stadtstruktur und -entwicklung in dieser Arbeit nicht verwendet bzw. umgesetzt werden, wodurch die spätere (räumliche) Gebieteinteilung von Städten und Stadtteilen nur nach einem vereinfachten Ringmodell (Burgess) erfolgt.

### 3.3 Stadtgliederung

Neben seiner übergeordneten Struktur kann für ein städtisches Gebiet zusätzlich seine *innere Gliederung* und *räumliche Gestaltung* beschrieben werden. Dafür sollen hier hinsichtlich der prozeduralen Umsetzbarkeit, die *morphogenetische* und

<sup>13</sup>Realistisch wirkende, orthogonale und konzentrische Straßennetze konnten bereits in der Arbeit von MÜLLER (2000) prozedural erzeugt werden [PaMu03].

die *funktionale* Stadtgliederung aufgeführt werden. Mit beiden Ansätzen können Grundriss- und Aufrissformen von Städten sowie deren räumliche Nutzungen unterschieden und definiert werden.

Die weiteren Ansätze der sozial-, aktions- und wahrnehmungsräumlichen Stadtgliederung werden aus Zeitgründen nicht behandelt [HeHe06].

- **Morphogenetische Stadtgliederung**

Die morphogenetische Stadtgliederung beschäftigt sich maßgeblich mit der historisch bedingten Gestaltung von Stadtgrundrissen und -aufrissen und den damit verbundenen räumlichen Auswirkungen auf die Gliederung einer Stadt [HeHe06].

In der Stadtmorphologie werden bei der Analyse von **Stadtgrundrissen** zunächst die folgenden hierarchischen Ebenen unterschieden:

1. *Stadt* (Region)
2. *Stadtteil*
3. *Block* (Hausblock)
4. *Parzelle* (Flurstück/Grundstück)

Mit diesen Hierarchie-Elementen kann die Stadtgestalt als zusammenhängende Topologie beschrieben werden und daraus zusätzlich Straßenzüge abgeleitet werden. Die Elemente einer Stufe können hinsichtlich ihrer räumlicher Nähe in Bezug gesetzt werden und beeinflussen generell die Elemente ihrer unteren Stufen. Dabei ist die Form der oberen Ebenen bzgl. Veränderungen besonders persistent, d.h. dass sich im Laufe der Zeit die Struktur des übergeordneten, historischen Straßennetzes einer Stadt nicht oder nur wenig verändert. Dagegen kann ein Grundstück relativ zügig in den unterschiedlichsten Formen bebaut werden und/ oder mit anderen Grundstücken zusammengelegt werden [HeHe06], [ErRa02].

Bei der **Aufrissgestaltung**<sup>14</sup> werden bei der morphogenetischen Stadtgliederung die folgenden *Aufrisselemente* für den Aufbau von Gebäuden definiert [HeHe06]:

- **Geschossanzahl und Gebäudehöhe**

Mit diesen beiden Werten kann die vertikale Ausdehnung eines Gebäudes bestimmt werden. Dabei werden sämtliche Keller- und Obergeschosse berücksichtigt.

- **Fassadenausprägung und Gebäudekonstruktion**

“Die [...] Fassadenausprägungen lassen sich grob in Fachwerkfassaden und verschiedene Arten von Massivfassaden, wie Putz, Hausteine, Sichtmauerwerke etc. gliedern”.

Des Weiteren kann definiert werden, dass “das innere konstruktive Gefüge sowohl die Gliederung der Fassade an sich — Verhältnis

---

<sup>14</sup>Der Aufriss beschreibt die vertikale Gestalt und Form von Gebäuden.

Öffnung zur Wandfläche — als auch ihre plastische Ausformung [bedingt]“ (BMBau<sup>15</sup>, 1983). D.h., dass Fassadenform und innere Konstruktion eines Gebäudes häufig in enger Beziehung zueinander stehen.

#### – Dachform

Das Gebäudedach ist ein weiteres wichtiges Aufrisselement und kann in folgende, historisch entstandenen Formen unterschieden werden (vgl. Abb. 13):

- \* *Satteldach*
- \* *Walmdach*
- \* *Walmkrüppeldach*
- \* *Mansardendach*
- \* *Pulldach*
- \* *Pulldach 'versetzt'*
- \* *Sheddach*
- \* *Flachdach*
- \* *Zeltdach*

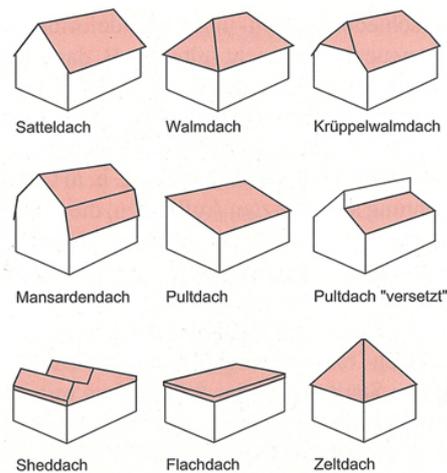


Abbildung 13: Grundformen des Dachs nach HEINEBERG (Quelle: [HeHe06]).

Da die meisten dieser Dachformen nicht symmetrisch sind, kann zusätzlich ihre Ausrichtung (zur Straßenseite) angegeben werden:

- \* *Giebelständig* (der Giebel des Hauses ist zur Straße orientiert.)
- \* *Traufständig* (das Haus steht mit seiner Traufe, meistens Breitseite, zur Straße.)

#### • Funktionale Stadtgliederung

Die Gliederung von Stadtflächen und -gebieten anhand ihrer Raumfunktionen bzw. "bestimmter räumlicher Funktionsvergesellschaftungen (z.B. City, Gewerbegebiet)" [HeHe06], wird in der Stadtgeographie als

<sup>15</sup>Bundesministerium für Raumordnung Bauwesen und Städtebau.

funktionale Stadtgliederung bezeichnet<sup>16</sup>. Welche Funktionstypen dabei unterschieden werden, hängt meistens von den Eigenschaften des Gebiets und den vorhandenen Problemstellungen ab. Eine allgemeine Definition von amtlich verwendeten **Flächennutzungstypen** wird im ersten Abschnitt der BauNVO<sup>17</sup> [BauNVO93] aufgeführt (die Beschreibungen werden im Wesentlichen übernommen):

#### **Allgemeine Arten der baulichen Nutzung (Bauflächen)**

- *Wohnbauflächen*
- *Gemischte Bauflächen*
- *Gewerbliche Bauflächen*
- *Sonderbauflächen*

#### **Besondere Arten der baulichen Nutzung (Baugebiete)**

- *Kleinsiedlungsgebiete*  
(Wohngebäude, Nutzgärten, landwirtschaftliche Betriebe, zur Versorgung des Gebiets dienende Geschäfte und Gastronomie, kleinere Handwerksbetriebe.)
- *Reine Wohngebiete*
- *Allgemeine Wohngebiete*  
(Wohngebäude, zur Versorgung des Gebiets dienende Geschäfte und Gastronomie, kleinere Handwerksbetriebe, Anlagen für kirchliche, kulturelle, soziale, gesundheitliche und sportliche Zwecke.)
- *Besondere Wohngebiete*  
(Wohngebäude, Läden, Hotels, Gastronomie, sonstige Gewerbebetriebe, Geschäfts- und Bürogebäude, Anlagen für kirchliche, kulturelle, soziale, gesundheitliche und sportliche Zwecke.)
- *Dorfgebiete*  
(Dorfgebiete dienen der Unterbringung land- und forstwirtschaftlicher Betriebe, dem Wohnen und der Unterbringung von nicht wesentlich störenden Gewerbebetrieben sowie der Versorgung der Bewohner des Gebietes dienenden Handwerksbetrieben.)
- *Mischgebiete*  
(Dienen vorwiegend dem Wohnen und der Unterbringung von nicht störenden Gewerbebetrieben.)
- *Kerngebiete*  
(Dienen vorwiegend der Unterbringung von Handelsbetrieben sowie der zentralen Einrichtungen der Wirtschaft, Verwaltung und Kultur.)
- *Gewerbegebiete*  
(Gewerbegebiete dienen vorwiegend der Unterbringung von nicht erheblich belästigenden Gewerbebetrieben.)

---

<sup>16</sup>Wird die Flächennutzung eines Gebiets mit den sozialen Schichten der Wohnbevölkerung verknüpft, können daraus Analysen der sozialräumlichen Stadtgliederungen abgeleitet werden.

<sup>17</sup>Verordnung über die bauliche Nutzung der Grundstücke (Baunutzungsverordnung - BauNVO).

- *Industriegebiete*  
(Industriegebiete dienen ausschließlich der Unterbringung von Gewerbebetrieben, die häufig in anderen Baugebieten unzulässig sind.)
- *Sondergebiete*  
(Als Sondergebiete, die der Erholung dienen, kommen insbesondere Wochenendhausgebiete, Ferienhausgebiete, Campingplatzgebiete in Betracht.)

Eine funktionale Stadtgliederung kann durch spezielle Karten, den sog. **Flächennutzungsplänen (FNP)**, dargestellt werden (vgl. Abb. 14). Wie diese Pläne erstellt werden, wird (in Deutschland) durch die PlanzV<sup>18</sup> geregelt.

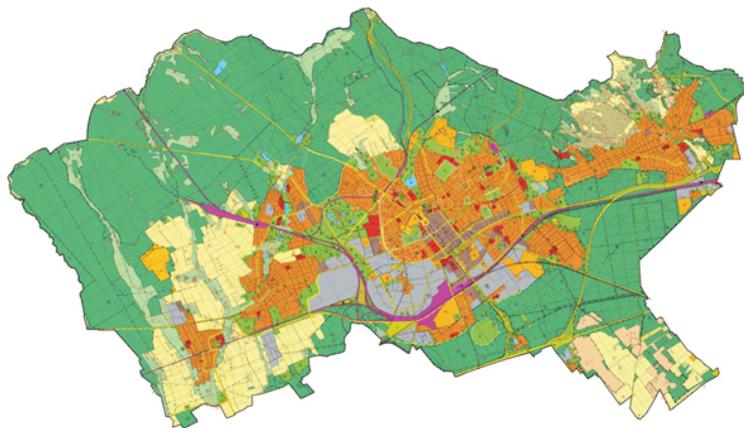


Abbildung 14: FNP der Stadt Darmstadt mit farblicher Unterscheidung der Nutzungstypen (Quelle: Online unter <http://www.darmstadt.de/wirtschaft/stadtplanung/gesamtstadt/>, letzter Zugriff 11.04.07).

Einen weiteren interessanten Ansatz zur Flächennutzungsverteilung liefert die Arbeit von CARTER (1972), in der **“die vertikale Differenzierung bzw. Abfolge der Nutzungen”** [HeHe06] innerhalb von Gebäuden bzgl. ihrer Lage im Stadtgebiet berücksichtigt wird. In diesem Modell wird für jedes Stockwerk ein Flächennutzungstyp angegeben, der die funktionale Nutzung der zugehörigen Räumlichkeiten bestimmt (vgl. Abb. 15).

Gebäude können, neben ihrer baulichen Nutzung, mit Hilfe der BauNVO durch weitere Eigenschaften charakterisiert werden:

- **Grundflächenzahl (GRZ) und Geschossflächenzahl (GFZ)**  
GRZ und GFZ sind zwei Maßzahlen, die bei bebauten Gebieten nicht überschritten werden dürfen (vgl. Tab. 2). Dabei gibt die GRZ an, wie viel Quadratmeter Grundfläche ( $area_{Gebäude}$ ) je Quadratmeter Grundstücksfläche ( $area_{Grundstück}$ ) zulässig sind (z.B. darf in Kerngebieten die gesamte und in Wohngebieten nur knapp die Hälfte der Grundstücksfläche bebaut werden). Die GFZ legt fest,

<sup>18</sup>Verordnung über die Ausarbeitung der Bauleitpläne und die Darstellung des Planinhalts (Planzeichenverordnung - PlanzV) von 1990.

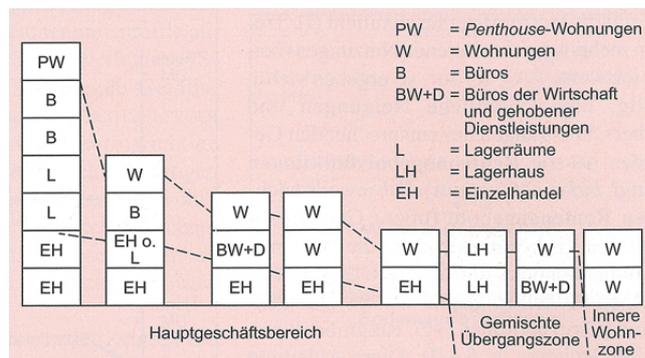


Abbildung 15: Vertikale Flächennutzungsverteilung nach CARTER (Quelle: [HeHe06]).

Baugebiet	GRZ	GFZ
Kleinsiedlungsgebieten	0.2	0.4
reine & allg. Wohngebiete	0.4	1.2
besonderen Wohngebiete	0.6	1.6
Dorfgebiete	0.6	1.2
Kerngebiete	1.0	3.0
Gewerbe- und Industriegebiete	0.8	2.4
sonstigen Sondergebieten	0.8	2.4

Tabelle 2: GRZ und GFZ in deutschen Bebauungsgebieten.

wie viel Quadratmeter Geschossfläche ( $area_{Stockwerk_i}$ ) je Quadratmeter Grundstücksfläche zulässig sind (z.B. könnte auf einem  $200m^2$  großen Grundstück mit GFZ 1.0 ein zweistöckiges Gebäude mit  $100m^2$  Grundfläche errichtet werden).

$$\begin{aligned}
 GRZ &= \frac{area_{Gebäude}}{area_{Grundstück}} \\
 GFZ &= \frac{\sum area_{Stockwerk_i}}{area_{Grundstück}} \quad (1)
 \end{aligned}$$

#### – Art der Bauweise

In der BauNVO werden die *offene* und die *geschlossene* Bauweise wie folgt unterschieden: "In der offenen Bauweise werden die Gebäude mit seitlichem Grenzabstand [...] errichtet. [...] In der geschlossenen Bauweise werden die Gebäude ohne seitlichen Grenzabstand errichtet" [BauNVO93].

Hier soll zusätzlich die *halboffene* Bauweise verwendet werden, bei der die Gebäude nur mit einer Seite an der Seitengrenze platziert werden.

### 3.4 Morphogenetische und städtebauliche Elemente

In diesem Kapitel werden alle für diese Arbeit relevanten, morphogenetischen und städtebaulichen Elemente vorgestellt und deren Eigenschaften beschrieben. Hierfür wird die morphogenetische Grundriss-Hierarchie (vgl. Abb. 16) um die

Elemente Gebäude, Stockwerk und Wohnung, sowie weitere städtebauliche Elemente (Straßen, Flure usw.) erweitert:

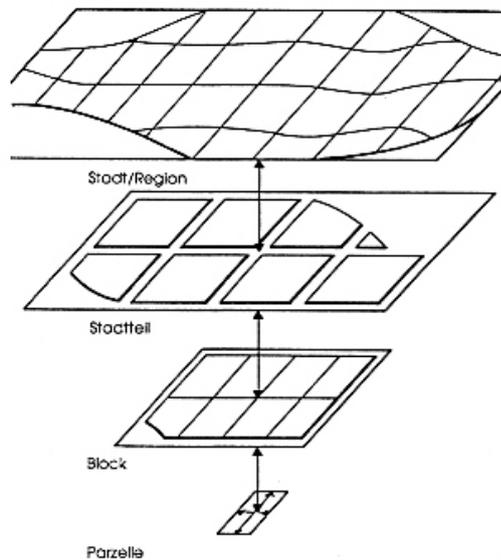


Abbildung 16: Morphogenetische Ebenen (Quelle: Gerhard Curdes, *Stadtstruktur und Stadtgestaltung*. Stuttgart 1993. 2. Auflage 1997).

### 1. Stadt und Straßen

Eine Stadt beschreibt einen Agglomerationsraum und kann in Stadtteile und -viertel unterteilt bzw. strukturiert werden. Für eine vereinfachte Strukturierung einer städtischen Region können z.B. die Modelle der *Chicagoer Schule* verwendet werden (siehe Kapitel 3.2).

Die Straßen der morphogenetischen Stadt-Ebene beschreiben das übergeordnete Straßennetz mit generell breiteren bzw. mehrspurigen Fahrbahnen (z.B. Stadtautobahnen).

### 2. Stadtteil und Straßen

Die Stadtteile einer Stadt sind entweder aus ehemals eigenständigen Siedlungen bzw. Dörfern im Zuge des Stadtwachstums entstanden bzw. 'einverleibt' oder gänzlich neu errichtet worden. Im ersteren Fall kann ein Stadtteil aus abstrakter Sicht auch als Kleinstadt mit eigener zentralörtlicher Funktion angesehen werden.

Der Grundriss der morphogenetischen Stadtteil-Ebene definiert die Hausblöcke, welche durch das hier vorhandene (untergeordnete) Straßennetz voneinander getrennt werden.

### 3. Hausblock

Der Hausblock (auch Block oder Straßenblock genannt) ist die größte stadtgeographische Einheit, die nicht von Straßen durchzogen ist. Die meisten Stadtpläne verwenden daher den Hausblock als kleinste Einheit, um insbesondere die Straßennetz-Struktur der abgebildeten Stadt darzustellen.

Durch das *Liegenschaftskataster* werden Hausblöcke in einzelne, benachbarte Parzellen bzw. *Flurstücke* unterteilt. Aus einem oder mehreren die-

ser Flurstücke wird dann jeweils ein (bebaubares) Grundstück gebildet [BauBG60].

#### 4. Grundstück und Gehweg

Ein Grundstück kann als Eigentum definiert werden und ist ein abgegrenzter Bereich der Erdoberfläche. Im Falle der ordnungsgemäßen Bebauung wird es als *Baugrundstück* bezeichnet [BauBG60]. Im stadtgeographischen Zusammenhang wird ein Grundstück durch mindestens ein Flurstück beschrieben und ist eine Teilfläche eines Hausblocks<sup>19</sup>.

An den Straßenseiten eines Grundstücks soll hier vereinfacht das Vorhandensein durchgängiger Gehwege angenommen werden. An den Grundstücksgrenzen können zusätzlich Umrandungen, z.B. in Form von Zäunen, errichtet werden.

#### 5. Gebäude, Stockwerk und Hausflur

Auf einem Baugrundstück können i.A. eine Reihe von baulichen Anlagen errichtet werden. Vereinfacht wird hier davon ausgegangen, dass solch ein Grundstück mit maximal einer Gebäudeform bebaut ist. Ein Gebäude wird als "selbständig benutzbare, überdachte bauliche Anlage, die von Menschen betreten werden können und geeignet oder bestimmt sind, dem Schutz von Menschen, Tieren oder Sachen zu dienen" (BauO NRW, §2) definiert und besteht i.A. aus einer Anzahl von Stockwerken (Vollgeschossen).

In allen Teilen eines städtischen Gebiets gibt es dabei Gebäude, die verschiedene Stockwerke hinsichtlich ihrer funktionalen Nutzung besitzen. Insbesondere im Erdgeschoss eines Gebäudes befinden sich oft Geschäfts- oder Gastronomieflächen, die von Büro- und/ oder Wohnräumen in den oberen Stockwerken abgelöst werden. Ein theoretisches Modell für diese vertikale Nutzungsverteilung liefert die Arbeit von CARTER (1972) [HeHe06]. Dabei sollte beachtet werden, dass diese Nutzungsverteilung auch häufig die Stockwerkshöhen beeinflusst. Z.B. hat ein im Erdgeschoss liegendes Geschäft eine größere Deckenhöhe als die Wohnbereiche in den oberen Stockwerken. Dazu zählen auch Eingangshallen von Bürohochhäusern oder Produktions- und Verwaltungsgeschosse in Industriegebäuden.

Hat ein Gebäude mehr als ein Stockwerk und/ oder mehrere Wohnungen, werden seine Räumlichkeiten im Normalfall über einen länglichen Raum (Gang), den Hausflur, erreicht. Im Erdgeschoss ist die Haustür direkt mit dem Hausflur verbunden.

#### 6. Wohnung und Wohnungsflur

In der Bauordnung des Landes Sachsen-Anhalt (BauO LSA) wird eine Wohnung wie folgt definiert: "Jede Wohnung muss von anderen Wohnungen und fremden Räumen baulich abgeschlossen sein und einen eigenen, abschließbaren Zugang unmittelbar vom Freien, von einem Treppenraum, einem Flur oder einem anderen Vorraum haben."

Im Normalfall besitzt eine Wohnung einen Gang (hier als *Wohnungsflur* bezeichnet), über den die Zimmer erreicht werden können und dieser mit der Wohnungstür verbunden ist. Im Gegensatz zum Hausflur müssen nicht alle

---

<sup>19</sup>Da der Grundstücksbegriff nicht eindeutig definiert ist, kann z.B. auch eine zur Stadt gehörende Wasserfläche als Grundstück bezeichnet werden. Diese Sonderformen sollen hier allerdings vernachlässigt werden.

Zimmer über den Wohnungsflur betretbar sein, da auch Verbindungstüren zwischen Zimmern ihre Erreichbarkeit sicherstellen können. In den meisten Fällen können dabei die Flurflächen innerhalb einer Wohnung minimiert werden, um größere und/ oder mehrere Räume (bei der Planung) einzufügen. Teilweise kann dabei auch ein Abschnitt eines Zimmers als Wohnungsflur verwendet werden.

### 3.5 Zwischenfazit

Mit den wissenschaftlichen Grundlagen der Stadtgeographie sowie den in Deutschland geltenden Bau- und Planungsverordnungen wurden hier konkrete und prozedural fassbare Eigenschaften für den komplexen sowie diffusen Stadtbe-griff erarbeitet. Insbesondere durch die morphogenetische und funktionale Stadt-gliederung werden geeignete Konzepte für die Platzierung und den Aufbau von Gebäuden beschrieben:

- Der Stadtgrundriss beschreibt allgemein die horizontale Form.
- Der Stadtaufriss beschreibt allgemein die vertikale Form.
- Es existieren verschiedene Dachformen und -ausrichtungen.
- Die innere und äußere Gebäudekonstruktion stehen in enger Beziehung.
- GRZ und GFZ legen das Maß und die Höhe der Bebauung eines städtischen Teilgebiets fest. Hierbei kann ein deutliches Kern-Rand-Gefälle festgestellt werden.
- Es existieren verschiedene Arten, wie ein Gebäude auf einem Grundstück platziert werden kann (Art der Bauweise).
- Es gibt eine Reihe von Flächennutzungstypen in horizontaler sowie auch in vertikaler Ausprägung.
- Der Stadtraum kann in hierarchisch angeordnete Elemente gegliedert werden, die sich gegenseitig beeinflussen.

Aufgrund dieser Erkenntnisse wird die Generierung von Stadtmodellen in die beiden Teilprobleme *Stadtgrundriss* und *Stadtaufriss* aufgeteilt, die in den Kapiteln 4 und 5 behandelt werden. Die abgeleiteten morphogenetischen und städtebaulichen Elemente, Stadt bis Wohnung, werden dabei vollständig berücksichtigt und umgesetzt<sup>20</sup>.

---

<sup>20</sup>Einige Eigenschaften der einzelnen Elemente müssen dabei aus zeitlichen Gründen vereinfacht werden.

## 4 Stadtgrundriss

Der Stadtgrundriss definiert die horizontale Ausdehnung, Struktur und Gliederung eines städtischen Gebiets und wird in dieser Arbeit, neben dem Stadtaufriss, als erstes Gestaltungselement für die Generierung von Stadt- und Gebäudemodellen verwendet. Folglich dient der Stadtgrundriss bzw. seine Erstellung als Einstiegspunkt für den hier entwickelten Gesamtprozess zur Generierung von urbanem Gelände, wobei die dabei entstehenden Ergebnisse später vom Stadtaufriss übernommen und weiterverarbeitet werden (siehe Kapitel 5).

Die Generierung eines Grundrisses kann im informationstechnischen Zusammenhang durch ein Verfahren beschrieben werden, das eine gegebene Fläche nach bestimmten Regeln unterteilt und folglich eine Anzahl von Teilflächen erstellt. Der Stadtgrundriss soll in dieser Arbeit mit Hilfe solch einer *Flächenunterteilung* prozedural generiert und seine Flächen durch eine *Datenstruktur* repräsentiert werden.

Um die Gesamtstruktur eines vollständigen Stadtgrundrisses zu vereinfachen, wird diese in hierarchisch angeordnete Ebenen zerlegt, die aus den einzelnen morphogenetischen und städtebaulichen Elementen in Kapitel 3.4 abgeleitet werden. Dabei entsteht für jede Teilfläche einer Stufe jeweils ein Grundriss der nächst unteren Ebene. Z.B. werden durch einen dreigeteilten Stadtteil genau drei Hausblöcke generiert, die dann wiederum in Grundstücke unterteilt werden können.

Die Hierarchie-Ebenen werden wie folgt definiert und ihnen dabei jeweils ein Grundriss zugeordnet:

1. Auf *Stadt-Ebene* wird eine Fläche so unterteilt, dass ihre Teilflächen *Stadtteile* (und Straßen) beschreiben.
2. Auf *Stadtteil-Ebene* wird eine Fläche so unterteilt, dass ihre Teilflächen *Hausblöcke* (und Straßen) beschreiben.
3. Auf *Hausblock-Ebene* wird eine Fläche so unterteilt, dass ihre Teilflächen *Grundstücke* beschreiben.
4. Auf *Grundstück-Ebene* wird eine Fläche so unterteilt, dass ihre Teilflächen *Gebäude* (und Gehwege und ggf. Freiflächen wie z.B. Gärten) beschreiben.
5. Auf *Gebäude/Stockwerk-Ebene* wird eine Fläche so unterteilt, dass ihre Teilflächen *Wohnungen* (und Hausflure) beschreiben.
6. Auf *Wohnungs-Ebene* wird eine Fläche so unterteilt, dass ihre Teilflächen *Räume* (und Wohnungsflure) beschreiben.

Jeder dieser einzelnen Grundrisse hat im stadtgeographischen Zusammenhang seine individuellen besonderen Merkmale. Falls alle diese Besonderheiten realistisch umgesetzt werden sollen, ist die vollständige Flächenunterteilung eines städtischen Gebiets bis hin zu den Räumen einer Wohnung ein schwer fassbares und komplexes Gebilde, dass im Folgenden mit Hilfe von geeignete Vereinfachungen durch *ein einzelnes* Verfahren abgebildet wird.

Zunächst bietet es sich an, die horizontale Komplexität eines städtischen Gebiets zu verringern, indem die einzelnen Grundrisse vereinfacht nach der Lage und Anordnung ihrer Teilflächen gruppiert werden:

- *Stadt und Stadtteil:*  
Ein Stadtteil soll hier vereinfacht als 'verkleinerte Stadt' betrachtet werden, wodurch dessen Aufbau mit dem einer Stadt verglichen wird (siehe Kapitel 3.4).  
Die Teilflächen einer Stadt bzw. eines Stadtteils liegen am Außenrand und im Inneren der Gesamtfläche. Z.B. ist der Stadtkern einer Stadt meistens von weiteren Stadtteilen (dem Stadtmantel) umgeben, welche wiederum von den Stadtteilen des Stadtrands umschlossen werden.  
In diesem Verfahren werden schwerpunktmäßig die Flächen der Stufen *Hausblock, Grundstück, Gebäude, Stockwerk und Wohnung* realistisch unterteilt und aus zeitlichen Gründen die Grundrisse dieser beiden Ebenen (*Stadt* und *Stadtteil*) nur vereinfacht generiert<sup>21</sup>. Durch die Möglichkeit, diese Grundrisse manuell zu definieren, kann diese Vereinfachung aber ausgeglichen werden.
- *Hausblock, Gebäude/Stockwerk und Wohnung:*  
Die Grundrisse dieser Ebenen können darauf reduziert werden, dass ihre Teilflächen immer am Außenrand der Gesamtfläche anliegen und dadurch vollständig innen liegende Teilflächen nicht vorhanden oder für diese Hierarchiestufen nicht relevant sind:
  - Bei einem Hausblock liegen alle Teilflächen bzw. Grundstücke an einer Straßenseite und damit am Außenrand.
  - Auf einem Stockwerk liegen alle Wohnungen an einer Außenwand des Gebäudes (jede Wohnung hat mindestens eine Wand mit Fenstern). Innen liegende Flächen beschreiben den Hausflur und ggf. das Treppenhaus.
  - Die Räume einer Wohnung liegen meistens an ihren Außenseiten<sup>22</sup>. Innen liegende Flächen beschreiben den Wohnungsflur und ggf. das Treppenhaus.

Die dafür notwendige Grundriss-Unterteilung soll in dieser Arbeit schwerpunktmäßig behandelt werden, da sie für das *eigentliche* Aussehen der (beim Stadtaufriß) entstehenden Gebäude-Modelle entscheidend ist.

- *Grundstück:*  
Der Grundriss eines Grundstücks wird hier gesondert betrachtet, da auf die zugehörige Fläche nur das Gebäude, sowie der Gehweg platziert werden und dafür keine Flächenunterteilung im hier verwendeten Zusammenhang durchgeführt werden muss.  
Allerdings beeinflusst die Grundstücks-Ebene durch das ihr zugeschriebene Platzieren der Gebäude maßgeblich deren Grundrissformen und soll daher in das hier entwickelte Verfahren so realistisch wie möglich integriert werden.

Damit ergeben sich zunächst die folgenden übergeordneten Schwerpunkte an ein Grundriss-Verfahren:

<sup>21</sup>Eine realistisch wirkende Grundriss-Unterteilung könnte z.B. durch die Umsetzung der Strukturmodelle der Chicagoer Schule erreicht werden (siehe Kapitel 3.2).

<sup>22</sup>Dabei kann ein Raum auch vollständig im Gebäude-Inneren, d.h. ohne Fenster, liegen und z.B. die Funktion eines Badezimmers einnehmen.

1. Verwendbare Grundrisse von Stadt und Stadtteilen
2. Realistische Grundrisse von Hausblöcken
3. Realistische Grundrisse (und Platzierung) von Gebäuden<sup>23</sup>.

Im folgenden Unterkapitel werden die Anforderungen an ein Flächenunterteilungs-Verfahren definiert, das hauptsächlich die Flächen der Ebenen Hausblock bis Wohnung realistisch unterteilen kann. Zusätzlich werden vorhandene Ansätze für die Flächennunterteilung vorgestellt und bewertet, wobei sich ergeben wird, dass keines dieser Verfahren im hier benötigten Zusammenhang verwendet werden kann. Daher soll in den weiteren Abschnitten ein im Rahmen dieser Arbeit neu entwickeltes Grundriss-Verfahren beschrieben werden, das im Folgenden als *Spatial Landmark Graph (SLG)* bezeichnet wird.

Das Besondere am SLG-Verfahren ist, dass es die Anforderungen aus Kapitel 4.1 vollständig erfüllt und für den späteren Stadtaufriss realistisch aufgeteilte Flächen als Datenbasis liefert. Dazu können fast beliebige polygonale Formen verarbeitet werden<sup>24</sup>, wodurch die Form der Eingangsflächen bzw. Umrisse, die z.B. auf Grundlage einer Stadtkarte durch den Benutzer definiert wurden, nur minimal eingeschränkt wird.

#### 4.1 Prozedurale Anforderungen und Ansätze für die Flächenunterteilung

Hier werden alle Anforderungen für einen Flächenunterteilungs-Algorithmus definiert, der die Flächen der oben beschriebenen Ebenen Hausblock bis Wohnung realistisch und für Stadt- und Stadtteil-Ebene vereinfacht unterteilen kann. Dafür



Abbildung 17: Auschnitte von Liegenschaftskarten (Quellen: [HaBe05] und Online unter <http://staff.science.uva.nl/~leo/singel%2077/verkoop/>, letzter Zugriff 11.04.07).

können die Bedingungen für Hausblöcke aus der Betrachtung von Liegenschaftskarten<sup>25</sup> und für Gebäude aus Grundrissplänen hergeleitet werden (vgl. Abb. 17

<sup>23</sup>Ein Gebäude-Grundriss sollte Informationen über Hausflur, Wohnungen sowie deren Wohnungsflure und Zimmer enthalten.

<sup>24</sup>Die vom SLG verarbeiteten Flächen dürfen sich nicht selbst überschneiden.

<sup>25</sup>Die Liegenschaftskarte eines städtischen Gebiets beinhaltet die Abgrenzungen aller Grundstücke und die Lage der darauf stehenden Gebäude. Werden die Grundstücke um ihre funktionale Nutzung erweitert, kann die Liegenschaftskarte als Flächennutzungsplan verwendet werden.

und 18). Zusätzlich werden noch Anforderungen für die Umsetzung und Steuerung des Verfahrens aufgestellt.

- Auf Stadt-, Stadtteil- und Hausblock-Ebene sind die Grundrissflächen meistens durch (ggf. auch konkave) Vielecke beschrieben. Diese nicht trivialen Flächen müssen geeignet unterteilt werden können.
- Alle entstehenden Teilflächen haben ab Hausblock-Ebene möglichst viele rechte Winkel und sind am jeweiligen Außenrand angeordnet. Speziell für Grundstücksflächen gilt, dass Seiten, die von der Straße nach innen führen zur Vorderseite senkrecht stehen.
- Das Einfügen von Straßen, Gehwegen und Fluren sollte mit möglichst geringem Aufwand ermöglicht werden.
- Für die spätere Verwendung der Ergebnisse im Aufriss-Verfahren werden sinnvoll strukturierte Flächendaten benötigt (z.B. in Form eines Graphen).
- Für eine gute Steuerbarkeit soll die Unterteilung auf jeder Ebene durch entsprechende Parameter kontrollierbar sein.
- Für die praktische Umsetzung soll das Verfahren effizient implementierbar und seine Komplexität an den Zeitrahmen dieser Arbeit angepasst sein.



Abbildung 18: Gebäude-Grundrisse (Quellen: Online unter <http://www.immquadrat.de/text/immobilien-angebote/harsdoerfferplatz.html> [© immquadrat Projekt GmbH] und <http://de.wikipedia.org/wiki/Grundriss>, letzter Zugriff 11.04.07).

#### 4.1.1 Vorhandene Techniken

Im Folgenden werden vorhandene Verfahren und Algorithmen vorgestellt, um eine gegebene Fläche zu unterteilen.

##### Lindenmayer-Systeme

“Ein Lindenmayer-System (L-System) ist ein regelbasiertes, paralleles Ersetzungssystem, das komplexe Endprodukte prozedural generieren kann. Mit Hilfe einer Menge von Ersetzungsregeln (auch Produktionen genannt) ersetzt man sukzessive Teile eines einfachen Initialobjektes und definiert auf diese Weise komplexe Objekte“ [PaMu03].

L-Systeme bezeichnen damit einen mathematischen Formalismus, der den *Ersetzungssystemen* bzw. *-grammatiken* zugeordnet werden kann. Diese Systeme liefern Zeichenketten, die durch weitere Verfahren, sogenannte *Interpreter*, ausgewertet werden können.

In der Computergrafik wird dieses Verfahren z.B. für das Generieren von Fraktalen (bekannte Beispiele sind die Kochkurve und -Inseln oder die Mandelbrot-Menge bzw. das Apfelmännchen)[FrGra01], Modellierung von Pflanzen [PrHaHaMe94], aber auch als grafische Ausgabetechnik (*Turtlegrafik*) der Programmiersprache LOGO verwendet (vgl. Abb. 19).

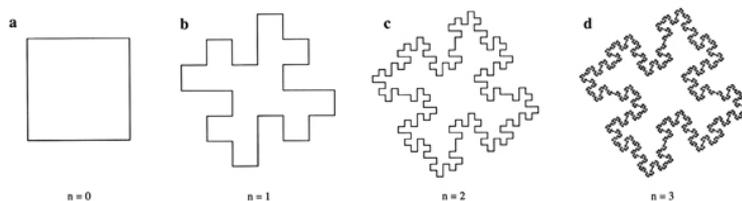


Abbildung 19: Entwicklung der Koch-Inseln mit einem L-System (Quelle: [PaMu03]).

Im Zusammenhang mit der prozeduralen Generierung von Stadtmodellen konnten mit Hilfe von L-Systemen bereits komplexe und realistisch aussehende Straßennetze erstellt und mit Bindung an eine Graphenstruktur die Grundrissflächen von Hausblöcken generiert werden [PaMu01], [PaMu02], [PaMu03].

#### • Vorteile

- Ersetzungsgrammatiken haben einen einfachen und gut strukturierten Aufbau.
- Mit simplen Regeln können bereits komplexe Strukturen generiert werden.
- L-Systeme wurden im informationstechnischem Zusammenhang entwickelt und sind daher einfach zu implementieren.

#### • Nachteile

- Für die Verwendung im stadtgeographischen Zusammenhang, bzw. für die Generierung von realistischen Straßennetzen, sind aufwändige Grammatik-Erweiterungen und *für jeden Straßennetztyp*<sup>26</sup> die Definition von komplexen Regelsätzen notwendig [PaMu01].
- Für die Generierung von Grundriss- bzw. Flächendaten muss eine zusätzliche Graphenstruktur entwickelt werden, da L-Systeme nur Zeichenketten produzieren.

L-Systeme stellen einen interessanten Ansatz zur automatischen Generierung von komplexen Strukturen dar, die für stadtgeographische Grundriss-Elemente durchaus realistisch aussehende Ergebnisse liefern. Aufgrund der zu langen Entwicklungszeit, bedingt durch die zahlreichen Anpassungen und Erweiterungen, sowie die nicht triviale Handhabung von komplexen Regelsätzen, wurde dieser

<sup>26</sup>Straßennetze werden in eine Vielzahl von Topologien, wie z.B. orthogonale, radiale oder konzentrische Grundrissmuster, unterschieden (siehe Kapitel 3).

Algorithmus hier jedoch nicht verwendet.

### Binary Space Partitioning und Slicing Trees

Das Binary Space Partitioning-Verfahren (BSP bzw. BSP-Tree) ist ein rekursiver Algorithmus, der einen Raum<sup>27</sup> pro Iteration in zwei Teilräume teilt und diese in einer Baumstruktur repräsentiert. Dieses Verfahren wird in vielen Bereichen der Computergrafik, wie z.B. beim Raytracing, verwendet. Wird das BSP-Verfahren im zweidimensionalen Raum verwendet, können mit dieser Methode Flächen bzw. Grundrisse unterteilt werden. Dabei wird die Fläche z.B. abwechselnd in linke/rechte und obere/untere Hälfte unterteilt.

Slicing Trees erweitern das BSP-Verfahren insoweit, dass eine Fläche pro Iteration nicht notwendigerweise vollständig horizontal oder vertikal geteilt werden muss, sondern auch "guillotinierte und freie Packmuster" [JaPu05] entstehen können (vgl. Abb. 20). Umgesetzt wurde das Slicing Tree-Verfahren u.a. von HONG et al., wobei die rechteckigen Teilflächen durch ein Mosaik-Muster repräsentiert werden [HHCGDCG01].

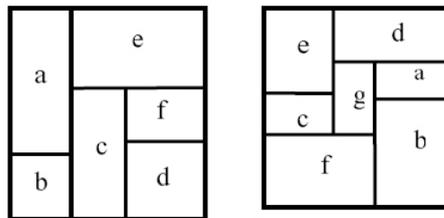


Abbildung 20: BSP- und Slicing Tree (Quelle: [HHCGDCG01]).

Im Zusammenhang mit der prozeduralen Generierung von Stadtmodellen wird das BSP-Verfahren bei der Unterteilung von Hausblockflächen in Grundstücke verwendet und liefert für möglichst rechteckige Flächen gute Ergebnisse [PaMu01], [PaMu02], [PaMu03]. Eine weitere Verwendung findet dieses Verfahren in der Arbeit von NOEL, in der ein Gebäude-Grundriss mit rechteckigen Räumen erstellt wird [JoNo03].

#### • Vorteile

- Der allgemeine BSP-Algorithmus ist einfach aufgebaut.
- Das allgemeine BSP-Verfahren ist effizient und einfach zu implementieren.
- Wird der Raum durch zwei orthogonale Achsen geteilt, entstehen viele rechte Winkel.
- Durch die Verwendung von Slicing Trees können die Ergebnisse des BSP-Verfahren (bei steigender Komplexität) verbessert werden [YoChSh01].
- Das Verfahren ist für das räumliche Suchen gut geeignet.

#### • Nachteile

<sup>27</sup>Dieser Raum ist durch multidimensionale Daten beschrieben.

- Für vieleckige und konkave Flächen liefert dieser Algorithmus hinsichtlich der Anforderungen aus Kapitel 4.1 keine verwendbaren Ergebnisse.

### Bin Packing und Floor Planing

Bin Packing- und Floor Planing-Verfahren beschreiben vereinfacht, wie viele Objekte mit bestimmter Ausdehnung maximal in ein anderes Objekt eingefügt werden können (auch *Platzierungs-Verfahren* genannt [PeMa06]).

Für dieses Einfügen von Unterobjekten kann der betrachtete Objektraum, vergleichbar mit dem BSP-Verfahren, in Teilräume unterteilt werden. Diese Technik kann auch auf mehrdimensionale Daten übertragen werden und findet im 2D insbesondere beim Prozessor-Layout Verwendung, wobei eine maximale Anzahl von rechteckigen Chip-Bausteinen auf einer minimal großen Fläche untergebracht wird [KuKe01], [ChKuHu00].

Diese Platzierungs-Verfahren werden z.B. in den Arbeiten von CHEN et al. (2002) und LIAO et al. (2003) so erweitert, dass neben rechteckigen auch konkave und sogar kreisförmige Flächenobjekte eingefügt werden können [ChFuLiRo02], [LiLuYe03] (vgl. Abb. 21).

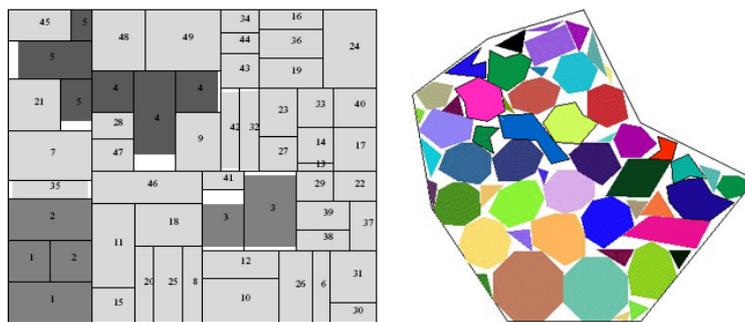


Abbildung 21: Bin Packing und Floor Planing (Quellen: [YoYaWo01] und [ChFuLiRo02]).

In dieser Arbeit wurde versucht, die Platzierungs-Problematik auf eine Grundriss-Unterteilung zu übertragen, indem eine maximale Anzahl von Teilflächen mit bestimmter Größe so in eine andere Fläche eingefügt wird, dass entstehende Zwischenräume minimiert werden. Z.B. sollen möglichst viele  $100m^2$  große, rechtwinklige Grundstücke in eine  $10000m^2$  große Hausblock-Fläche eingefügt werden.

#### • Vorteile

- Die Ausmaße und Form der (nicht unbedingt rechteckigen) Teilflächen kann exakt vorbestimmt werden.
- Es entstehen minimal große Freiräume.

#### • Nachteile

- Für das Einfügen von nicht rechteckigen Teilflächen sind bereits aufwändige Anpassungen des Platzierungs-Verfahrens notwendig.
- Die zusätzliche Verwendung von nicht rechteckigen 'Container'-Flächen ist ein hoch komplexes Verfahren und konnte bisher nur in einzelnen Fällen umgesetzt werden [ChFuLiRo02].

Durch den beschränkten Zeitrahmen dieser Arbeit und aufgrund der Komplexität von Platzierungs-Verfahren mit nicht rechteckigen Ausgangs- und Teilflächen, wurde eine Verwendung dieses Verfahrens ausgeschlossen.

### Voronoi-Diagramme

Ein Voronoi-Diagramm teilt einen zweidimensionalen Raum vollständig in wabenähnliche Teilflächen. Dabei wird eine Punktmenge als Datenbasis verwendet, deren Punkte jeweils im Inneren genau eines der entstehenden Polygone liegen. Für die Eck-Punkte der Polygone gilt, dass sie zu allen diesen Punkten, die in angrenzenden Polygonen liegen, in einem minimalen Abstand entfernt liegen [AuKl01] (vgl. Abb. 22). Werden die Eingangs-Punkte mit ihren nächstgelegenen verbunden, entsteht eine Delaunay-Triangulierung (siehe Kapitel 5.1.1).

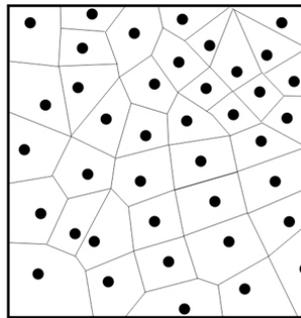


Abbildung 22: Voronoi-Diagramm (Quelle: [ToKe06]).

In der Arbeit von KELLY (2006) wurde das Voronoi-Diagramm bereits zur Generierung von Stadtgrundrissen (auf Hausblock-Ebene) verwendet [ToKe06].

- **Vorteile**

- Die Gesamtfläche wird vollständig bzw. ohne Freiräume unterteilt.
- Die Teilflächen können bei entsprechender Punktverteilung mit gleichförmigen Waben- bzw. Zellstrukturen verglichen werden und sind somit z.B. für die Gebietseinteilungen von Städten und Stadtteilen geeignet.

- **Nachteile**

- Die Datenbasis besteht aus Punkten, deren Verteilung nur schwer mit dem resultierenden Voronoi-Diagramm in Verbindung gebracht werden kann. Zusätzlich ist eine gezielte Steuerung nur bedingt möglich, da das Verschieben eines Punktes mindestens die Form *aller angrenzenden* Polygone ändert.
- Eingangsdaten in Form von Flächen können nicht verarbeitet werden.
- Es entstehen wenig rechteckige Flächen.

### 4.1.2 Zwischenfazit

Alle vorgestellten Verfahren erfüllen i.A. nur wenige der Anforderungen aus Kapitel 4.1. Allen ist gemeinsam, dass vieleckige und ggf. konkave Flächen nicht oder

nur ungeeignet verarbeitet werden können und Unterteilungen, die einer Liegenschaftskarte gleichen, nicht möglich sind. Die aufwändige Anpassung eines dieser Verfahren an die hier gestellten Anforderungen hätte, neben der zunächst notwendigen allgemeinen Umsetzung, den Zeitrahmen dieser Arbeit gesprengt. Aus diesen Gründen wurde hier ein eigenes Unterteilungs-Verfahren entwickelt (der *Spatial Landmark Graph*, siehe nachfolgenden Unterkapitel), das die Liste der Anforderungen vollständig erfüllt und für eine Grundriss-Generierung aller oben aufgeführten Ebenen (Stadt bis Wohnung) verwendet und umgesetzt werden kann.

## 4.2 Spatial Landmark Graph (SLG)

Der *Spatial Landmark Graph* (SLG) ist ein im Rahmen dieser Arbeit entstandenes Verfahren, das zur Grundriss-Generierung im stadtgeographischen Zusammenhang verwendet werden kann. Der SLG besteht dabei aus einer Datenstruktur zur Speicherung von Flächendaten und einer Vielzahl von Funktionalitäten, die insbesondere für die Unterteilung benutzt werden.

Für die Speicherung, Unterteilung und Manipulation von polygonalen Flächendaten wird dabei ein gerichteter Graph mit den folgenden Bestandteilen verwendet:

- *Knoten*
- *Kanten*
- *Flächen bzw. Teilflächen und Pfade*

Ein gerichteter Graph enthält Kanten, die einen definierten Start- und End-Knoten besitzen und somit eine Richtung haben. Durch diese Eigenschaft können geometrische Berechnungen, insbesondere der Trigonometrie und Vektorrechnung, schnell und effizient ausgeführt werden.

Aus den Anforderungen in Kapitel 4.1 wird folgendes (vereinfachte) Verfahren für eine Flächenunterteilung abgeleitet (vgl. Abb. 23):

1. Die Außenkanten der gegebenen Fläche werden so unterteilt, dass eine bestimmte Anzahl von ungefähr gleich großen Abschnitten entsteht. Jeder Abschnitt ist durch eine linke und rechte Grenze festgelegt (im Folgenden als *Begrenzungs-Knoten* bezeichnet) und bildet die Vorderseite einer späteren Teilfläche.
2. An beiden Begrenzungs-Knoten jedes Abschnitts, wird ein Strahl im rechten Winkel in das Innere der Gesamtfläche verschossen<sup>28</sup> (vgl. Abb. 23 a).
3. Für jeden Strahl werden die Schnittpunkte mit den anderen Strahlen im Inneren der Gesamtfläche berechnet. Anschließend wird aus den Verbindungen der Schnittpunkte eine innere vernetzte Struktur aufgebaut (vgl. Abb. 23 b).

---

<sup>28</sup>Diese beiden Strahlen definieren linke und rechte Seite einer Teilfläche.

4. Für jeden entstandenen Abschnitt aus 1 wird mit einer Menge von inneren Kanten eine Teilfläche gebildet. Welche Kanten dabei ausgewählt werden, wird durch eine Pfadsuche und -auswahl bestimmt (siehe Kapitel 4.3.4 und 4.3.5).
5. Die ggf. entstandenen Freiräume im Inneren der Gesamtfläche werden zu weiteren Teilflächen umgewandelt.
6. Alle Teilflächen werden in weiteren Bearbeitungsschritten so angepasst, dass möglichst viele parallele Seiten bzw. rechte Winkel entstehen (vgl. Abb. 23 c).

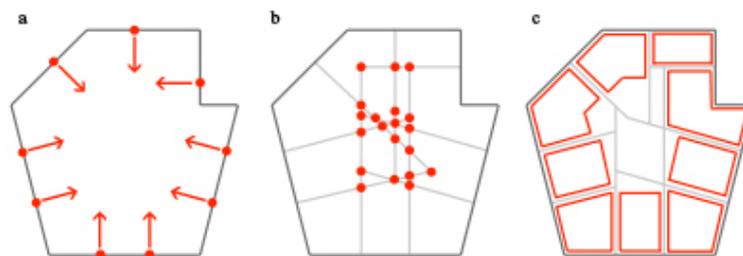


Abbildung 23: Die vereinfachte Flächenunterteilung des SLG: a) Begrenzungs-Knoten und Strahlen, b) Innere Struktur, c) Generierte Teilflächen.

Das vollständige Verfahren für die Unterteilung, sowie die Bestandteile des SLG werden in den folgenden Unterkapiteln beschrieben.

#### 4.2.1 Knoten

Ein Knoten besitzt im SLG zunächst immer einen zwei-dimensionalen Positionsvektor ( $\vec{p}_{node}$ ) in der Ebene. Zusätzlich wird dieses Objekt u.a. mit einer Typisierung nach seiner Lage im Graphen erweitert, um die Knoten während eines Unterteilungsschrittes zu lokalisieren<sup>29</sup>.

##### Lage im Graphen

- *Eck-Knoten:*  
Ein Eck-Knoten definiert eine Ecke des Graphen-Umrisses. Die Menge aller Eck-Knoten bestimmt durch ihre Positionsvektoren und die Reihenfolge, in der die Knoten eingefügt wurden, die Gesamtfläche und damit die initiale Datenbasis des SLG.
- *Innen-Knoten:*  
Innen-Knoten liegen innerhalb der Graphenfläche.
- *Begrenzungs-Knoten:*  
Begrenzungs-Knoten werden speziell für die SLG-Unterteilung eingeführt

<sup>29</sup>Der Knoten kann z.B. auf dem Umriss oder im Inneren der Gesamtfläche liegen und/oder einen Begrenzungs-Knoten repräsentieren.

und liegen auf dem Umriss des Graphen zwischen Eck- und/ oder anderen Begrenzungs-Knoten. Sie repräsentieren im stadtgeographischen Zusammenhang z.B. die an der Straße liegenden (linken und rechten) Grenzpunkte der Grundstücke.

Für die spätere Unterteilung der Gesamtfläche wird für diese Knoten zusätzlich ein Richtungsvektor bestimmt, der rechtwinkelig ins Innere des Graphen zeigt und wie ein Strahl behandelt werden kann.

#### 4.2.2 Kanten

Eine Kante besteht aus zwei Knoten und repräsentiert deren Verbindung<sup>30</sup> innerhalb der Graphen-Struktur. Im SLG ist eine Kante gerichtet und besitzt somit einen definierten Start- und End-Knoten ( $nodeStart_{edge}$ ,  $nodeEnd_{edge}$ , vgl. Abb. 24). Durch diese Eigenschaft kann mit Hilfe der Positionsvektoren von Start- und End-Knoten die Geradengleichung der Kante direkt bestimmt werden, wobei insbesondere der Richtungsvektor ( $\vec{v}_{edge}$ ) für viele Berechnungen benötigt wird:

$$\begin{aligned}\vec{v}_{edge} &= \vec{p}_{nodeEnd_{edge}} - \vec{p}_{nodeStart_{edge}} \\ f_{edge}(x) &= \vec{v}_{edge} * x + \vec{p}_{nodeStart_{edge}}\end{aligned}\quad (2)$$

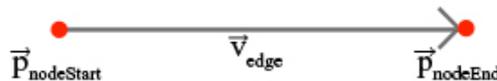


Abbildung 24: Eine Kante im SLG.

Für viele Berechnungsvorschriften, wie z.B. der Pfadsuche, ist es notwendig, den Graphen entlang einer bestimmter Kanten-Richtung zu durchlaufen. Die dabei behandelten Kanten sind allerdings nur selten (bzgl. ihrer Richtung) geordnet, d.h. dass der End-Knoten einer Kante den Start- oder End-Knoten einer Nachfolger-Kante repräsentiert (vgl. Abb. 25).

Um dieses Problem zu lösen, müssen für jede behandelte Kante zunächst *immer* die Mengen ihrer Vorgänger- und/ oder Nachfolger-Kanten bekannt sein. Ist diese Bedingung erfüllt, kann mit folgendem Algorithmus die Graphenstruktur in oder gegen die Richtung einer Start-Kante durchlaufen werden (hier werden beispielhaft nur die Außen-Kanten einer SLG-Struktur behandelt):

```

1 TraverseBorderEdges( listEdges )
2 {
3     /* start configuration */
4     edge = listEdges.GetFirstBorderEdge();
5     node = edgeStart.nodeEnd;
6
7     for( i = 0; i < listEdges.GetSize(); i++ )
8     {
9         /* search next edge */
10        for( j = 0; j < listEdges.GetSize(); j++ )
11        {
12            if(
13                listEdges[j].type == BORDER &&

```

<sup>30</sup>Eine Kante besitzt nur Referenzen zu ihren Knoten. Wird eine Kante gelöscht bleiben die Knoten erhalten. Damit können Knoten von mehreren Kanten gleichzeitig benutzt werden.

```

14     edge != listEdges[j] && listEdges[j].HasNode( node )
15     )
16     {
17         edgeNext = listEdges[j];
18         nodeNext = listEdges[j].GetNodeOpposite( node );
19         break;
20     }
21 }
22
23 ...
24
25 /* assign for next iteration */
26 edge = edgeNext;
27 node = nodeNext;
28 }
29 }

```

Listing 1: Durchgehen einer Kantenliste im Pseudocode.

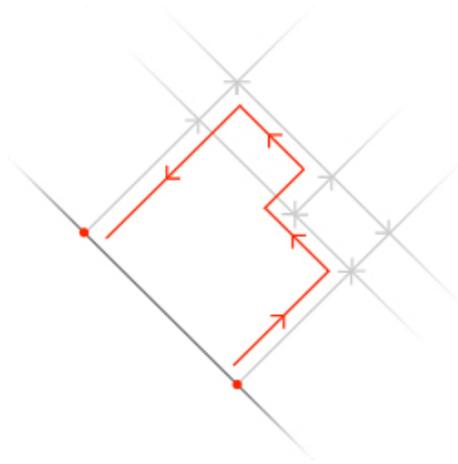


Abbildung 25: Kanten-Orientierungsproblematik im SLG.

Im SLG wird eine Kante u.a. mit einer Typisierung nach ihrer Lage im Graphen und geographischen Zusammenhang erweitert. Diese beiden Eigenschaften sind im Laufe der Unterteilung und für die Aufriss-Generierung wichtig, um die einzelnen Kanten innerhalb der Graphenstruktur sowie in den zugehörigen stadtgeographischen Ebenen exakt zu lokalisieren (vgl. Abb. 26).

### Lage im Graphen

- *Außen:*  
Außen-Kanten bilden den Umriss des Graphen und können bis auf Innen-Knoten alle anderen Knotentypen besitzen.
- *Innen:*  
Innen-Kanten liegen innerhalb des Graphen und besitzen mindestens einen Innen-Knoten.

### Lage im geographischen Zusammenhang

Diese Eigenschaft wird u.a. für die spätere Generierung des Aufrisses verwendet.

- *Außen liegend zur Straße*
- *Außen liegend zu einem Zwischenraum*

- Außen liegend zu einer Nachbarfläche
- Außen liegendes Fenster zur Straße
- Außen liegendes Fenster zu einem Zwischenraum
- Außen liegender Haupteingang
- Außen liegender Eingang

Folgende Werte werden nur innerhalb von Gebäuden verwendet:

- Innen liegend zu einem Hausflur
- Innen liegend zu einer Nachbarwohnung
- Innen liegend zu einem Wohnungsflur
- Innen liegend zu einem Nachbarraum
- Innen liegender Haupteingang
- Innen liegender Eingang
- Innen liegender Treppeneingang

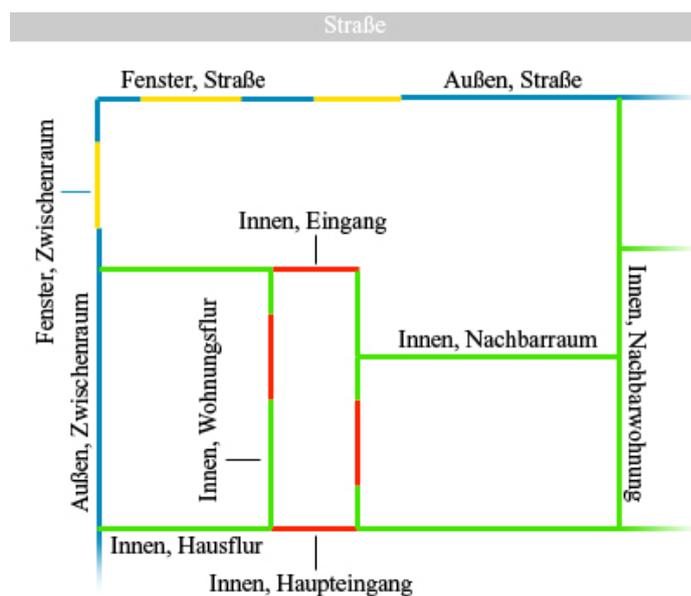


Abbildung 26: Kanten-Lage im geographischen Zusammenhang am Beispiel einer Wohnung.

### 4.2.3 Flächen und Pfade

Flächen und Pfade sind im Normalfall keine Bestandteile einer allgemeinen Graphenstruktur. Für den SLG ist es aber sinnvoll, die Fläche inklusive einer Menge von Pfaden als Element hinzuzufügen, um die Unterteilungen des Graphen direkt zu repräsentieren. Damit werden Zugriffe auf und Berechnungen mit

diesen Flächen in der praktischen Umsetzung modularisiert, vereinfacht sowie beschleunigt.

Ein **Pfad** besteht aus einer Menge von Kanten und Knoten, die einen geschlossenen Polygonzug beschreiben. Für seine Kantenliste muss also gelten:

$$nodeStart_{edgeFirst} \equiv nodeEnd_{edgeLast} \quad (3)$$

Eine **Fläche** bzw. **Teilfläche** besitzt eine Liste zum Speichern von Pfaden, welche bei der Pfadsuche mit diesen gefüllt wird. Bei der Pfadauswahl wird genau ein Pfad aus dieser Liste ausgewählt und dessen Kanten und Knoten übernommen. Folglich repräsentiert eine Fläche erst nach der Pfadauswahl eine eindeutige Teilfläche des Graphen.

Im SLG wird ein Pfad u.a. mit einer Typisierung nach seiner Lage im Graphen erweitert, um seine repräsentierte Fläche bei der späteren Pfadsuche und -auswahl innerhalb der Struktur zu lokalisieren.

#### Lage im Graphen

- *Außenrand:*  
Der Pfad beinhaltet mindestens eine Außen-Kante.
- *Außenrand mit Ecke:*  
Der Pfad beinhaltet mindestens zwei Außen-Kanten und einen Eck-Knoten.
- *Außenrand mit konkaver Ecke:*  
Der Pfad beinhaltet mindestens zwei Außen-Kanten, einen Eck-Knoten und hat eine konkave Außenecke (d.h. zwei Außen-Kanten umschließen einen überstumpfen Winkel<sup>31</sup>).
- *Innen:*  
Der Pfad liegt im Inneren des Graphen und beinhaltet keine Außen-Kanten.

Eine Fläche wird hier u.a. ebenfalls mit einer Typisierung nach seiner Lage im Graphen erweitert, um in späteren Berechnungen die repräsentierte Fläche zu lokalisieren. Dieser Wert wird dabei von der Lage des ausgewählten bzw. zugewiesenen Pfades abgeleitet.

#### Lage im Graphen

- *Rand:*  
Fläche liegt außen.
- *Innen:*  
Fläche liegt innerhalb des Graphen.

### 4.3 SLG: Flächenunterteilung

Nachdem der Umriss bzw. die Fläche des SLG durch mindestens drei Eck-Knoten definiert wurde, kann mit der eigentlichen Flächenunterteilung begonnen werden. Dieser Vorgang wird in die folgenden nacheinander ausgeführten Einzelschritte zerlegt:

---

<sup>31</sup>Der Winkel ist größer als 180°

1. **Initialisierung** des SLG bzw. seiner Struktur.
2. **Begrenzungs-Knoten platzieren.**
3. **Außen liegende Flächen bilden (Pfadsuche und Pfadauswahl).**
4. **Zwischenräume** für Straßen, Flure und Gehwege bilden, sowie die Gebäude auf ihren Grundstücksflächen platzieren.
5. **Überflüssige Knoten und Kanten entfernen.**
6. **Innen liegende Flächen** bilden.
7. **Lage der Kanten bestimmen.**

Alle aufgeführten Schritte werden in den folgenden Unterkapiteln ausführlicher behandelt.

#### 4.3.1 Initialisierung

Durch die Initialisierung wird der Graph, der bis jetzt nur aus einer Menge von Eckpunkten besteht, für die anschließende Unterteilung vorbereitet.

In einem ersten Schritt werden alle vorhandenen Eck-Knoten, unter Berücksichtigung der Reihenfolge, in der sie eingefügt wurden, durch Außen-Kanten verbunden. Jede Kante wird dabei wie folgt definiert:

$$nodeStart_{edge_i} = nodeCorner_i \quad , \quad nodeEnd_{edge_i} = nodeCorner_{i+1} \quad (4)$$

Im zweiten Schritt der Initialisierung wird für die neu erstellten Kanten ihre Lage im geographischen Zusammenhang bestimmt. Dieser Wert wird entweder von der höheren Hierarchieebene übernommen (falls diese vorhanden ist) oder je nach stadtheographischer Lage neu definiert (vergleichbar mit den Zuweisungen in Kapitel 4.3.9).

#### 4.3.2 Begrenzungs-Knoten platzieren

Nach der Initialisierung der Graphenfläche, können nun die Begrenzungs-Knoten auf dem Umriss der Gesamtfläche eingefügt werden. Diese Knoten definieren die Grenzen der generierten Teilflächen auf den Außen-Kanten des SLG. Z.B. legen jeweils zwei Begrenzungs-Knoten eines Hausblocks die linke und rechte Grundstücksgrenze eines Grundstücks an der Straßenseite fest.

Die Anzahl dieser Knoten bestimmt die maximale Anzahl der zu generierenden außen liegenden Teilflächen und bestimmt damit maßgeblich die Form der Unterteilung.

Mit folgendem Algorithmus werden diese Knoten möglichst gleichmäßig auf den Außen-Kanten des Graphen verteilt:

1. In einer Schleife über die Anzahl der Begrenzungs-Knoten wird pro Iteration jeweils diejenige Außenkante durch einen Begrenzungs-Knoten geteilt, deren Länge maximal ist. Hierfür wird die Kante durch zwei gleich lange Kanten ersetzt, die den eingefügten Begrenzungs-Knoten als gemeinsamen Knoten besitzen<sup>32</sup>.

---

<sup>32</sup>Die beiden neuen Kanten sind dabei halb so lang wie die ursprüngliche Kante.

Um die Begrenzungs-Knoten möglichst gleichmäßig zu verteilen, werden hier die Kanten erst im zweiten Schritt tatsächlich geteilt und hier pro Iteration nur ein Zähler für die jeweilige Kante erhöht. Diese Strategie liefert vor allem für diejenigen Kanten bessere Ergebnisse, auf die eine gerade Anzahl von Begrenzungs-Knoten platziert werden soll. Z.B. wird eine Kante auf der zwei Begrenzungs-Knoten eingefügt werden, sich in drei neue Kanten teilen, die jeweils ein Drittel der ursprünglichen Länge besitzen. Wird die Kante hier direkt geteilt, entstehen beim ersten Teilen zwei Kanten mit der halben ursprünglichen Länge, wobei eine von beiden wiederum in zwei Kanten geteilt wird, deren Länge dann nur noch ein Viertel der ursprünglichen Länge beträgt (vgl. Abb. 27).

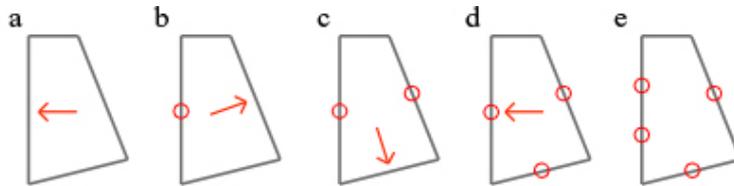


Abbildung 27: Verteilung von vier Begrenzungs-Knoten.

Die Länge einer Außenkante ergibt sich aus folgender Formel:

$$length_{edge} = \frac{length_{edge}}{num_{landmarkNodes_{edge}}} \quad (5)$$

Dieser Schritt wird so lange wiederholt, bis alle Begrenzungs-Knoten auf den Außen-Kanten verteilt wurden.

2. Nun können alle Außenkanten, auf denen Begrenzungs-Knoten platziert werden sollen, geteilt und durch neue Kanten ersetzt werden. Dafür wird zunächst die Position der jeweiligen Begrenzungs-Knoten durch eine einfache Skalierung des Kantenvektors berechnet und gespeichert.

$$\vec{p}_{node_i} = \vec{p}_{nodeStart_{edge}} + \frac{\vec{v}_{edge}}{(num_{landmarkNodes_{edge}} + 1) * i} \quad (6)$$

Nachdem alle Positionen der Begrenzungs-Knoten berechnet wurden, werden die zugehörigen Außen-Kanten an diesen Positionen geteilt bzw. gesplittet (siehe Kapitel 4.6.3). Folglich besitzen die dabei entstehenden Kanten die neuen Begrenzungs-Knoten als gemeinsame Knoten.

Um die Verteilung für konkave Graphenflächen zu optimieren, werden die nächstgelegenen Begrenzungs-Knoten einer überstumpfen Ecke in ihre Richtung verschoben (vgl. Abb. 28). Damit wird die Vorderseite einer hier entstehenden Teilfläche kürzer und somit ihr normalerweise größerer Flächeninhalt kleiner. Die neue Position dieser Begrenzungs-Knoten wird durch folgende Formel berechnet (mit der minimalen Strecklänge  $length_{min}$  gemäß Kapitel 4.4.2):

$$\vec{p}_{landmark} = \vec{p}_{nodeCorner} + \vec{v}_{edge} * \frac{length_{min}}{length_{edge}} \quad (7)$$

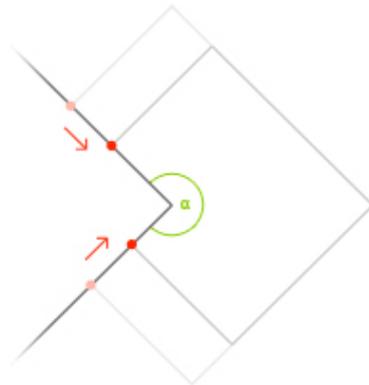


Abbildung 28: Verschieben von Begrenzungsknoten an konkaven Ecken.

Wurde in diesem Schritt nur ein oder kein Begrenzungsknoten platziert, ist die gesamte Unterteilung beendet. Es wird dabei nur eine Teilfläche angelegt, die aus den Außen-Kanten des SLG erstellt wird. Dieses Verhalten kann explizit für die Stadtelemente Grundstück und Gebäude eingesetzt werden, da ein Grundstück durch das Einfügen von Zwischräumen (siehe Kapitel 4.3.6) und die Gebäudegrundfläche erst durch die Stockwerke in Teilflächen zerlegt wird.

### 4.3.3 Innere Knoten und Kanten bilden

Nachdem die Begrenzungsknoten auf den Außen-Kanten platziert wurden, liegen die Grenzen der späteren Teilflächen auf dem Umriss des Graphen fest. Die innere Struktur zum Bilden dieser Teilflächen wird hier durch folgendes Vorgehen aufgebaut (vgl. Abb. 29):

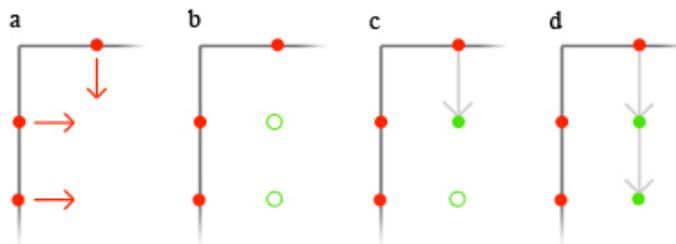


Abbildung 29: Konstruktion der inneren Knoten und Kanten mit Hilfe von Schnittpunkten.

1. An jedem Begrenzungsknoten wird ein Strahl im rechten Winkel in das Innere der Graphenfläche verschossen.
2. Alle Schnittpunkte mit anderen Strahlen, die innerhalb der Graphenfläche liegen, werden gespeichert. Für jeden Begrenzungsknoten wird dabei eine Liste mit Schnittpunkten aufgebaut, die durch seinen Strahl entstehen. (Wie die Positionen dieser Schnittpunkte berechnet werden, und ob sie innerhalb der SLG-Gesamtfläche liegen, wird in Kapitel 4.7 behandelt.)



```

1 SearchPath( path , edge , nodeEnd )
2 {
3   path.InsertEdge( edge );
4
5   if( nodeEnd == nodeLandmarkFirst )
6     area.InsertPath( path.Copy() );
7
8   else
9   {
10    for( i = 0; i < nodeEnd.GetInsideOutsideEdges().GetSize(); i++ )
11    {
12      edgeNext = nodeEnd.GetInsideOutsideEdges()[i];
13
14      if( edgeNext != edge && !path.IsIn( edgeNext ) )
15        SearchPath( path , edgeNext , edgeNext.nodeEnd );
16    }
17  }
18 }

```

Listing 2: Vereinfachte Pfadsuche im Pseudocode.

- (a) (Zeile 1-3) Aufgerufen wird die Rekursion immer mit dem aktuell betrachteten Knoten und der letzten betrachteten Kante; der Knoten ist dabei im Normalfall der End-Knoten dieser Kante. Da sich im Pfad-element schon alle benötigten Außen-Kanten und -Knoten befinden, kann die erste Rekursion mit der vom Begrenzungs-Knoten ausgehenden Innen-Kante und ihrem End-Knoten aufgerufen werden.
- (b) (Zeile 5-6) Falls der übergebene Knoten der erste Begrenzungs-Knoten ist, wurde ein gültiger Pfad gefunden. Der Pfad wird nun kopiert und in der Pfadliste des Flächenelements gespeichert. Die aktuelle Rekursionsstufe ist damit beendet.
- (c) (Zeile 8-17) Ist der übergebene Knoten nicht der erste Begrenzungs-Knoten wird für jede seiner ein- und ausgehenden Innen-Kanten, die nicht die übergebene Kante ist und noch nicht behandelt wurde, die Rekursion erneut aufgerufen.

Durch diesen Algorithmus werden jedoch auch Pfade erfasst, die auf Grund zu großer Strecken oder einer zu komplexen Form für spätere Teilflächen nicht geeignet sind. Daher ist es sinnvoll, einige Tests der späteren Pfadauswahl, die bereits mit nicht geschlossenen Pfaden durchgeführt werden können, direkt in den Algorithmus der Pfadsuche zu integrieren. Dadurch wird zusätzlich der notwendigerweise hohe Rechenaufwand des rekursiven Algorithmus entscheidend verringert.

Bevor die Rekursion ein weiteres Mal aufgerufen wird, werden für jede Kante bzw. in dem bisher gefundenen Pfad die folgenden Eigenschaften überprüft:

- Kanten mit sehr spitzen und sehr stumpfen Winkeln werden übersprungen. Für einen Rekursionsaufruf muss gelten:

$$[1^\circ, 10^\circ] < \angle_{180}(\vec{v}_{edge}, \vec{v}_{edgeNext}) < [170^\circ, 179^\circ] \quad (9)$$

- Der nächste Knoten darf nicht links vom linken und nicht rechts vom rechten Begrenzungs-Knoten liegen (vereinfacht auf die x-Koordinaten der Punkte mit  $nodeLeft = nodeLandmarkLast$ ,  $nodeRight = nodeLandmarkFirst$ ):

$$pos.x_{nodeLeft} < pos.x_{nodeNext} < pos.x_{nodeRight} \quad (10)$$

(D.h. implizit, dass der Winkel zwischen der ersten und zweiten, sowie der vorletzten und letzten Innen-Kante nicht überstumpf sein darf.)

- Eine vorgegebene Anzahl von überstumpfen Winkeln darf für den gesamten inneren Teil des Pfades nicht überschritten werden. Für eine weitere Rekursion muss daher gelten:

$$\sum_{i=0}^n (\angle_{360}(\vec{v}_{edge_i}, \vec{v}_{edge_{Next_i}}) > 180^\circ) \leq \text{limit}_{reflexAngles} \quad (11)$$

- Die längste gerade Strecke des Pfades darf eine vorgegebene Länge nicht überschreiten.  
Diese Größe kann schon während der Rekursion mit einem nicht geschlossenen Pfad geprüft werden, da dessen längste gerade Strecke nicht kleiner werden kann. Dagegen kann die minimale Länge erst im abgeschlossenen Pfad korrekt bestimmt werden, weil diese sich noch vergrößern kann.

Die Pfadsuche wird beendet, wenn alle Begrenzungs-Knoten behandelt wurden.

#### 4.3.5 Außen liegende Flächen (Pfadauswahl)

Wurden alle gültigen Pfade bzw. möglichen Ausmaße der Teilflächen bestimmt, kann nun mit der Pfadauswahl fortgefahren werden.

Je nach Komplexität der inneren Graphenstruktur können pro Teilfläche sehr viele Pfade gefunden worden sein (z.B. über 100). Aus dieser u.U. sehr großen Menge muss nun jeweils genau ein Pfad pro Teilfläche ausgewählt werden, um deren Fläche eindeutig zu definieren.

Dabei wird die Pfadliste jeder Teilfläche durchgegangen und derjenige Pfad ausgewählt, der die folgenden Kriterien am ehesten erfüllt:

1. Der Pfad darf bereits zugewiesene Pfade anderer Teilflächen nicht (über)schneiden, wodurch überlappende Teilflächen vermieden werden. *Dieses Kriterium muss erfüllt sein.*
2. Die kürzeste gerade Strecke im Pfad darf einen bestimmten Wert nicht unterschreiten. (Z.B. dürfen die Räume einer Wohnung keine Seiten unter  $2m$  besitzen.) *Dieses Kriterium muss erfüllt sein.*
3. Der Pfad hat vier rechte Winkel.
4. Der Pfad hat eine Fläche von:

$$\text{area}_{path} = \frac{\text{area}_{graph}}{\text{num}_{areas}_{graph}} \quad (12)$$

Kann einer Teilfläche kein Pfad zugewiesen werden, d.h. alle Pfade erfüllen Kriterium 1. oder 2. nicht, wird die Teilfläche gelöscht.

Nach der Pfadauswahl ist im Normalfall<sup>33</sup> eine Teilfläche durch eine Menge von Außen- und Innen-Kanten sowie deren Start- und End-Knoten eindeutig definiert. Genau zwei dieser Knoten sind Begrenzungs-Knoten und legen die linke und rechte Grenze bzw. die Vorderseite der Teilfläche auf dem SLG-Umriss fest.

<sup>33</sup>Die Teilfläche wurde nicht gelöscht.

### 4.3.6 Zwischenräume

Sobald die außen liegenden Teilflächen erstellt wurden, müssen für einige Hierarchieebenen zusätzlich Räume zwischen allen oder nur bestimmten Teilflächen erstellt und ggf. weiter behandelt werden. Dieser Schritt ist für das Einfügen folgender Grundriss-Elemente notwendig:

- Straßen
- Haus- und Wohnungsflure
- Gehwege
- Platzierung von Gebäuden

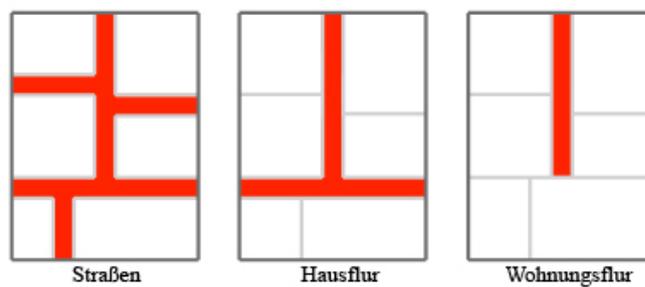


Abbildung 31: Zwischenräume für Straßen, Hausflur und Wohnungsflur.

Um diese Zwischenräume in die bisherige Unterteilung einzufügen, werden die vier Seiten<sup>34</sup> (vgl. Abb. 32) einer Teilfläche bestimmt und anschließend verkleinert (siehe Kapitel 4.6.2, vgl. Abb. 31). Welche Bedingungen und Voraussetzungen für diese Räume in einer Hierarchieebene gelten, wird in Kapitel 3.4 und 6 beschrieben.

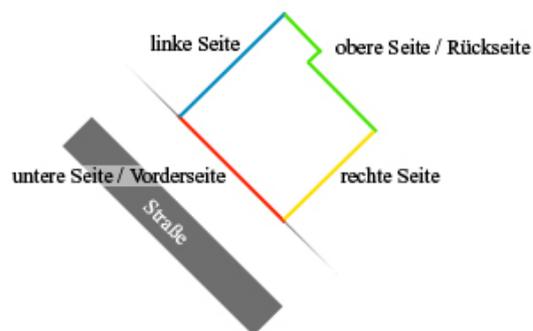


Abbildung 32: Teilfläche reduziert auf vier Seiten.

#### **Straßen**

Straßen entstehen in bzw. zwischen der gesamten inneren Struktur, wodurch alle Teilflächen des SLG behandelt werden.

<sup>34</sup>Für das Einfügen der Zwischenräume wird eine Teilfläche zunächst immer auf vier Seiten (eine untere/vordere, linke, rechte und obere/hintere Seite) reduziert, um die nachfolgenden Berechnungen zu vereinfachen.

- Für eine bestimmte Straßenbreite (*distance*), werden die Teilflächen-Seiten um  $\frac{distance}{2}$  verkleinert (geschrumpft).
- Damit keine 'halben' Straßen generiert werden, werden nur die Seiten bzw. Kanten der Teilflächen geschrumpft, die nicht auf dem Umriss des Graphen liegen. D.h. es werden hier nur linke, rechte und obere Seiten behandelt.

### Hausflur

Als Ausgangspunkt für die Generierung von Hausfluren wird ein vorgegebener Begrenzungs-Knoten ausgewählt. An diesem soll später der Hauseingang eingefügt werden.

Da der Flur die Verbindung zum Hauseingang darstellen soll, werden dafür zunächst die zwei Teilflächen geschrumpft, zwischen denen der Eingang liegen soll. Dabei wird jeweils nur diejenige Teilflächen-Seite um einen bestimmten Betrag verschoben (links oder rechts), die im Inneren des Graphen liegt und den Begrenzungs-Knoten für den Eingang enthält. Damit sind diese beiden Teilflächen (Wohnungen) bereits durch den Flur erreichbar.

Für die restlichen Teilflächen (Wohnungen) wird der nachfolgende Algorithmus verwendet. Dieser wird solange für jede Teilfläche aufgerufen, bis sie mindestens eine freie (nicht von anderen Teilflächen benutzte) Kante hat, die für das Einfügen eines Wohnungseingangs breit genug ist:

```

1  while( area.maxFreeDistance > widthEntrance )
2      area.CreateCorridor();
3
4  CreateCorridor()
5  {
6      if(
7          maxFreeEdges == freeEdgesTop &&
8          minLengthEdges == lengthEdgesTop
9      )
10     Shrink( edgesTop, widthCorridor );
11
12     else if(
13         maxFreeEdges == freeEdgesLeft &&
14         minLengthEdges == lengthEdgesLeft
15     )
16         Shrink( edgesLeft, widthCorridor );
17
18     else
19         Shrink( edgesRight, widthCorridor );
20 }

```

Listing 3: Vereinfachte Generierung eines Hausflurs im Pseudocode.

1. (Zeile 1-3) Die Funktion *CreateCorridor()* wird solange aufgerufen, bis eine freie Kante mit entsprechender Länge gefunden wurde.
2. (Zeile 6-20) Es wird diejenige Seite geschrumpft, welche die meisten freie Kanten besitzt und gleichzeitig am kürzesten ist. Dies resultiert in einem minimal großen neuen Raum, der an den bereits vorhandenen Zwischenraum angefügt wird.

### Wohnungsflur

Wie bei einem Hausflur werden für den Zwischenraum des Wohnungsflurs zuerst die beiden Teilflächen geschrumpft, die den Begrenzungs-Knoten für den Eingang beinhalten.

Da im Unterschied zum Hausflur nicht jeder Raum einer Wohnung über den Flur erreichbar sein muss, werden nur die oberen Seiten der restlichen Teilflächen geschrumpft. Durch dieses Vorgehen wird erreicht, dass die Flur-Fläche minimal

wird und möglichst wenige Seiten des Flurs an einer Außenwand liegen. (Im Normalfall besitzen dabei erst 5-Zimmerwohnungen Räume mit definierten bzw. vorhandenen oberen Seiten.)

### Gehweg

Ein Gehweg wird immer an der Straßenseite eines Grundstücks eingefügt. Dafür wird die untere, bzw. vordere Seite eines Grundstücks um einen bestimmten Betrag ins Innere des Graphen verschoben.

Dieser Schritt teilt die Grundstücksfläche also in einen Gehweg und eine Restfläche, auf der später ein Gebäude platziert werden kann. Anders als bei den vorigen Zwischenräumen ist es hier sinnvoll, den entstehenden Raum (Gehweg) direkt einer neuen Teilfläche zuzuordnen. Diese wird anschließend in weitere viereckige Flächen geteilt, um die spätere Aufriss-Generierung zu erleichtern.

### Gebäudeplatzierung

Auf die Teilfläche zur Bebauung des Grundstücks wird im Normalfall ein Gebäude platziert. Die Platzierung, bzw. Bauweise, wird in die drei nachfolgenden Arten unterschieden (siehe Kapitel 3.3):

- *Geschlossen*
- *Halboffen*
- *Offen*

Dabei wird die Bebauungsfläche des Grundstücks verkleinert und als spätere Grundlage der Gebäudegrundfläche, bzw. für die Ausmaße des Gebäude-SLG verwendet.

Generell ist es zunächst sinnvoll, diese Grundstücks-Teilfläche möglichst rechtwinklig zu gestalten. Dies kann erreicht werden, indem die obere Seite (Rückseite) der Teilfläche parallel zur Vorderseite gelegt wird. Da die linke und rechte Seite der Teilfläche bereits senkrecht zur Vorderseite liegen, kann mit dieser Behandlung der oberen Seite die Anzahl der rechten Winkel maximiert und für Grundstücke an Ecken die Form der Bebauungsfläche zugleich realistisch angepasst werden (vgl. Abb. 33).

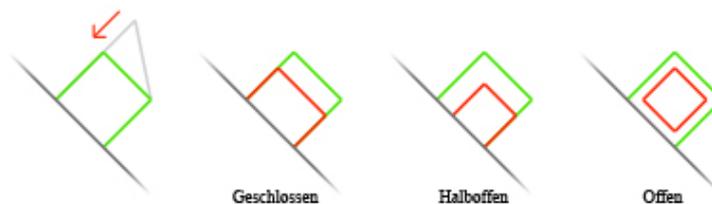


Abbildung 33: Vorbereitung der Gebäudeplatzierung sowie geschlossene, halboffene und offene Bauweise.

Dafür wird die Teilfläche wie folgt verändert:

- Um die Grenzen des Grundstücks hinsichtlich seiner Tiefe nicht zu überschreiten, wird zuerst die Länge der linken und rechten Seite bestimmt und die Kürzere ausgewählt. Diese Länge definiert den maximal gültigen Abstand *distance* zwischen Vorder- und Rückseite.

- Die Kanten der Teilflächen-Vorderseite werden nun kopiert, um *distance* ins Innere des Graphen verschoben und mit den Kanten der oberen Seite getauscht. Für Teilflächen, die an einer Ecke liegen, muss die so entstehende Rückseite noch skaliert werden, da sie im Normalfall nicht die Länge der Vorderseite besitzt.<sup>35</sup> Der Faktor *factor*, um den die Kanten der oberen Seite skaliert werden müssen, berechnet sich aus dem Abstand von Start und End-Knoten der Vorderseite und der Entfernung der Punkte auf linker und rechter Seite in einem Abstand von *distance*:

$$factor = \frac{|\vec{p}_{nodeEnd_{bottom}} - \vec{p}_{nodeStart_{bottom}}|}{|\vec{p}_{nodeDistance_{right}} - \vec{p}_{nodeDistance_{left}}|} \quad (13)$$

Im zweiten Schritt wird die Bebauungsfläche des Grundstücks nun für die eigentliche Platzierung bzw. Art der Bauweise angepasst:

- **Geschlossen**  
Für eine geschlossene Platzierung von Gebäuden, wird jeweils nur die obere Seite behandelt. Diese wird um einen bestimmten Wert zur unteren Seite verschoben. Dabei sollte das vorher definierte Verhältnis der Seitenlängen nicht unterschritten werden (siehe Kapitel 6.4.1).
- **Halboffen**  
Für die halboffene Bauweise wird wie bei der geschlossenen Bauweise die obere Seite und anschließend entweder die linke oder rechte Seite um einen bestimmten Betrag ins Innere der Grundstücksfläche verschoben.
- **Offen**  
Für die offene Bauweise werden im Normalfall alle Seiten zum Flächenzentrum des Grundstücks verschoben<sup>36</sup>.

Nachdem die Zwischenräume eingefügt bzw. generiert wurden, ist die Form aller Teilflächen und Freiräume in einer Hierarchieebene festgelegt und die eigentliche Flächenunterteilung des SLG beendet.

#### 4.3.7 Überflüssige Knoten & Kanten entfernen

Während der bisherigen Unterteilungsschritte, insbesondere bei der Generierung von Zwischenräumen, können eine Reihe von überflüssigen Kanten und Knoten entstanden sein. Diese werden nun aus dem SLG entfernt, um seine Struktur und den Speicherplatz zu optimieren.

Hierbei werden zuerst alle Innen-Kanten und -Knoten gelöscht, die zu keiner Teilfläche (bzw. ihrem zugewiesenen Pfad) gehören. Nun befinden sich nur noch Kanten und Knoten im Graphen, die entweder zu einer Teilfläche gehören oder auf dem Umriss liegen.

Für alle übrigen Kanten wird geprüft, ob sie mit ihrer nachfolgenden Kante verbunden, bzw. zu einer Kante vereint, werden können. Dafür müssen folgende Kriterien erfüllt werden:

<sup>35</sup>Die Vorderseite eines Grundstücks ist an (konvexen) Ecken länger und bei konkaven Ecken kürzer als die obere Seite.

<sup>36</sup>In bestimmten Fällen, z.B. einer zu kleinen Fläche, werden nicht alle Seiten geschrumpft und damit eine halboffene oder geschlossene Bauweise umgesetzt.

- Der Winkel zwischen den Kanten beträgt  $0^\circ$ .

$$\angle_{180}(\vec{v}_{edge}, \vec{v}_{edgeNext}) = 0^\circ \quad (14)$$

- Beide Kanten gehören zu der gleichen Teilfläche oder liegen auf dem SLG-Umriss und gehören zu keiner Teilfläche.

Können zwei aufeinander folgende Kanten verbunden werden, werden sie und ihr gemeinsamer Knoten gelöscht und eine neue Kante mit den übrig gebliebenen beiden Knoten erstellt.

$$nodeStart_{edgeNew} = nodeStart_{edge} \quad , \quad nodeEnd_{edgeNew} = nodeEnd_{edgeNext} \quad (15)$$

#### 4.3.8 Innen liegende Flächen

Den innen liegenden Flächen werden hier alle Zwischen- bzw. Freiräume zugewiesen, die bei der Erstellung der außen liegenden Teilflächen, sowie der Generierung von Zwischenräumen keiner Teilfläche zugewiesen wurden. Diese Flächen werden jeweils aus allen zusammenhängenden Kanten (und ihren Knoten) gebildet, die eine der folgenden Bedingungen erfüllen:

- Die Kanten liegen außen und gehören zu keiner außen liegenden Teilfläche.
- Die Kanten liegen innen und gehören zu genau einer außen liegenden Teilfläche.

Aus Zeitgründen war es nicht möglich, diese Freiräume weiter zu behandeln, um ggf. weitere Teilflächen im Inneren des Graphen unter stadtgeographischen Gesetzmäßigkeiten zu generieren. Dies ist allerdings nur für die Grundriss-Generierung auf Stadt- und Stadtteil-Ebene interessant, da im hier verwendeten Zusammenhang ausschließlich diese beiden (morphogenetischen) Elemente Unterobjekte enthalten können, die nicht am Außenrand ihrer Gesamtfläche liegen (z.B. das Stadtzentrum als innen liegender von anderen Stadtteilen umgebender Stadtteil). Für die restlichen Hierarchieebenen kann dieser Aspekt vernachlässigt werden, ohne die Ergebnisse hinsichtlich der Anforderungen aus Kapitel 4.1 negativ zu beeinflussen.

#### 4.3.9 Lage der Kanten bestimmen

Nachdem die innen liegenden Flächen generiert wurden, ist die gesamte Fläche des SLG seinen außen und innen liegenden Teilflächen zugeteilt worden. Für alle Kanten werden nun ihre Lagen im geographischen Zusammenhang bestimmt, um sie z.B. bei der Aufriss-Generierung (im städtischen Gebiet) zu lokalisieren. Die Zuweisung dieser Werte erfolgt in Abhängigkeit der Kanten-Lage im SLG und dessen stadtgeographischer Zugehörigkeit:

- **Stadt**  
Zwischen allen Teilflächen (späteren Stadtteilen) wurde eine Straße eingefügt und somit liegt jede Kante *an einer Straße*.
- **Stadtteil**  
Alle Kanten liegen ebenfalls *an einer Straße*, da zwischen allen Teilflächen Straßen eingefügt wurden.

- **Hausblock**  
 Alle Außen-Kanten liegen *an einer Straße*, weil ein Hausblock immer von Straßen umgeben ist.  
 Wird eine Innen-Kante von zwei Teilflächen benutzt, liegt sie damit auf der Grenze zwischen zwei Grundstücken und damit *zu einer Nachbarfläche*.  
 Ansonsten liegt eine Innen-Kante an der Grenze *zu einem Zwischenraum*.
- **Grundstück**  
 Alle Außen-Kanten, die noch nicht der Straßen- oder Rückseite zugeordnet wurden, liegen auf der Grenze zwischen zwei Grundstücken und somit *an einer Nachbarfläche*.  
 Alle Innen-Kanten liegen auf der Grenze *zu einem Zwischenraum*.
- **Gebäude und Stockwerke**  
 Eine Außen-Kante liegt auf der Grenze *zu einem Nachbargebäude*, falls das zugehörige Grundstück an dieser Stelle an eine Nachbarfläche grenzt. Ist die Außen-Kante zum Gehweg ausgerichtet, liegt sie an der Vorderseite des Gebäudes und damit *an der Straße*. Befindet sich die Kante weder an einer Nachbarfläche noch an der Straße, muss sie an der Grenze *zu einem Zwischenraum* liegen.  
 Wird eine Innen-Kante von zwei Teilflächen benutzt, liegt sie zwischen zwei Wohnungen und somit *an einer Nachbarfläche*. Ansonsten liegt ihre Rückseite an einem Hausflur bzw. *zu einem Zwischenraum*.
- **Wohnung**  
 Liegt eine Außen-Kante an der Außenseite des Gebäudes, übernimmt sie die Lage der Gebäude-Kante (*zur Straße, zu einem Nachbargebäude oder zu einem Zwischenraum*). Befindet sich eine Außen-Kante mit ihrer Rückseite *an einem Zwischenraum*, grenzt sie an einen Hausflur. Ist dies nicht der Fall muss sie *an einer Nachbarwohnung* liegen.  
 Wird eine Innen-Kante von zwei Teilflächen benutzt, liegt sie zwischen zwei Räumen und somit *an einer Nachbarfläche*. Ansonsten liegt ihre Rückseite *an einem Wohnungsflur*.

#### 4.4 SLG: Parameter

Die gesamte Struktur bzw. Unterteilung des SLG kann durch folgende Parameter beeinflusst werden:

- **Anzahl Begrenzungs-Knoten**  
 Dieser Wert gibt die Anzahl der Begrenzungs-Knoten bzw. entstehenden Teilflächen an. Wie dieser Wert automatisch berechnet werden kann, wird in Kapitel 4.4.1 beschrieben und hier zunächst die dafür erforderlichen Parameter definiert:
  - **Typ der Berechnungsvorschrift** [ *Wachstumsfunktion* | *nach Umfang* ]
  - **Durchschnittlicher Flächeninhalt einer Teilfläche**
  - **Seitenverhältnis der Teilflächen**
  - **Skalierungsfaktor und Exponent** (für die Wachstumsfunktion)
  - **Minimale Anzahl von Begrenzungs-Knoten**

- **Positionsabweichung der Begrenzungs-Knoten**

Die Positionierung der Begrenzungs-Knoten kann zusätzlich durch einen Abweichungswert beeinflusst werden, indem diese Knoten um einen Zufallswert entlang der zugehörigen Kante verschoben werden (siehe Kapitel 4.3.2). Der Abweichungswert ( $\delta$ ) definiert dabei die minimale bzw. maximale Entfernung zur ursprünglichen Position (mit  $\{x = rand(a, b) | x \in \mathbb{R} \wedge a \leq x \leq b\}$ ):

$$\vec{p}_{node} = \vec{p}_{node} + f_{edge}(rand(-\delta, \delta)) \quad (16)$$

- **Länge der kürzesten/längsten Kante**

Diese Werte begrenzen die Seitenlängen einer Teilfläche und können durch das Verfahren in Kapitel 4.4.2 berechnet werden.

- **Anzahl der überstumpfen Winkel**

Die (maximale) Anzahl von überstumpfen Winkeln muss für jeden Pfadtyp definiert werden:

- Anzahl der überstumpfen Winkel in Pfaden am Außenrand
- Anzahl der überstumpfen Winkel in Pfaden am Außenrand mit Ecke
- Anzahl der überstumpfen Winkel in Pfaden am Außenrand mit konkaver Ecke

- **Typ, Start-Knoten und Breite der Zwischenräume**

Durch diese drei Parameter werden die Zwischenräume des Grundrisses gesteuert (siehe Kapitel 4.4.3).

Mit diesen Variablen ist die Flächenunterteilung kontrollierbar und ihre Wert-Zuweisung somit die Grundlage für die Generierung von realistischen Grundrissen. Welche dieser Parameter in Beziehung gesetzt und prozedural bestimmt werden können, wird in den folgenden Unterkapiteln beschrieben.

#### 4.4.1 Anzahl der Begrenzungs-Knoten

Die Anzahl der Begrenzungs-Knoten und somit die Anzahl der im Normalfall entstehenden Teilflächen kann in dieser Arbeit durch eine von zwei Berechnungsvorschriften automatisch bestimmt werden:

1. **Basierend auf dem Umfang**

Die erste Möglichkeit, die Anzahl der Begrenzungs-Knoten zu bestimmen, geht von gleichmäßig verteilten und gleichgroßen Teilflächen am Außenrand des Graphen aus. Dabei soll die resultierende Anzahl proportional mit der Größe des Umfangs (und damit der Fläche) des Graphen ansteigen.

Es wird angenommen, dass alle Teilflächen einen bestimmten Flächeninhalt ( $area_{average}$ ) haben, quadratisch sind und nebeneinander am Außenrand des Graphen liegen bzw. angeordnet sind. Dadurch ergibt sich die Länge einer Teilflächen-Vorderseite mit folgender Formel:

$$length_{side} = \sqrt{area_{average}} \quad (17)$$

Die Anzahl der Begrenzungs-Knoten wird nun bestimmt, indem der Umfang des Graphen durch diesen Längen-Wert geteilt wird, d.h. es wird

berechnet, wie viele Vorderseiten auf dem Außenrand platziert werden können:

$$numLandmarks = \frac{boundary_{graph}}{length_{side}} \quad (18)$$

Schließlich wird noch die Anzahl der Ecken vom Ergebnis subtrahiert, da bei Teilflächen, die an einer Ecke liegen, jeweils zwei Seiten auf dem Außenrand des Graphen liegen:

$$numLandmarks = numLandmarks - numCorners_{graph} \quad (19)$$

Dieses Verfahren wird benutzt, falls ungefähr gleich große Teilflächen mit einem bestimmten Flächeninhalt generiert werden. Z.B. kann diese Berechnung für die Anzahl von Straßenblöcken eines Stadtteils verwendet werden, der unabhängig von seiner Größe immer in Blöcke von  $\approx 1ha$  unterteilt werden soll.

## 2. Wachstumsfunktion

Die zweite Möglichkeit, die Anzahl der Begrenzungs-Knoten zu bestimmen, basiert auf einer radizierenden Wachstumsfunktion. Dabei nimmt die geschätzte Anzahl, bei stark ansteigender Graphenfläche, nur langsam zu und konvergiert letztendlich gegen einem bestimmten Wert.

Solch ein Verhalten lässt sich z.B. durch eine Wurzelfunktion über der (evtl. skalierten) Graphenfläche mit variablen Wurzelexponenten modellieren:

$$numLandmarks = \exp \sqrt{\frac{area_{graph}}{scale}} \quad (20)$$

Dabei kann die Berechnung mit Hilfe eines Skalierungsfaktors (*scale*) und eines Exponenten (*exp*) beeinflusst werden. Die Größe des Exponenten gibt dabei an, wie schnell die Funktion gegen ihr Maximum konvergiert. Mit Hilfe des Skalierungsfaktors kann der Abstand zum konvergierenden Wert, bzw. die Größe einer Schrittweite, verändert werden.

Dieses Verfahren wird benutzt, falls die Anzahl der Teilflächen zum Einen begrenzt ist, und zum Anderen nicht proportional mit der Gesamtfläche ansteigen soll. Z.B. können die Berechnungsparameter für die Zimmeranzahl einer Wohnung so angegeben werden, dass sie ab einer Fläche von  $50m^2$  drei, ab einer Fläche von  $80m^2$  vier und erst ab einer Fläche von  $200m^2$  maximal fünf Zimmer besitzt.

### 4.4.2 Winkel und Abstände

Für den SLG ist es sinnvoll, eine minimale und maximale Streckenlänge zu bestimmen, um die Seitenlängen bzw. Ausmaße der entstehenden Teilflächen einzugrenzen. Um die Form der Teilflächen möglichst einfach zu halten, wird zusätzlich die maximale Anzahl der überstumpfen Winkel angegeben. Diese Werte werden insbesondere in den Kapiteln 4.3.4 und 4.3.5 für die Suche und Auswahl der Teilflächen-Pfade benötigt.

Durch die **minimale und maximale Streckenlänge** wird die untere und obere Grenze für die Längen der Teilflächen-Seiten definiert. Diese Grenzen sind notwendig, um besonders kurze oder lange Seiten zu vermeiden. Z.B. sollten die Ausmaße der Räume einer  $40m^2$ -Wohnung nicht  $1m$  unter- oder  $10m$  überschreiten. Beide Werte können mit dem Flächeninhalt und dem Verhältnis der Seiten einer Fläche berechnet werden. Dabei berechnet sich die minimale Länge, bei einem Verhältnis  $ratio = \frac{length_{sideShort}}{length_{sideLong}}$ ,  $length_{sideShort} < length_{sideLong}$  durch

$$length_{min} = \sqrt{areaAverage * ratio} \quad (21)$$

und die maximale Länge durch

$$length_{max} = \frac{\sqrt{areaAverage}}{ratio} \quad (22)$$

Für den SLG wird hierfür angenommen, dass alle Teilflächen im Durchschnitt gleichgroß und rechteckig sind und die Seitenlängen in einem bestimmten Verhältnis zueinander stehen. Z.B. sollte ein rechteckiges Grundstück maximal doppelt so lang wie breit sein, d.h. die Längen von Vorderseite und linker/rechter Seite dürfen das Verhältnis von 1:2 nicht überschreiten.

Die **maximale Anzahl der überstumpfen Winkel** wird für alle Pfadtypen (siehe Kapitel 4.2.3) festgelegt, um die entstehenden Teilflächen zusätzlich zu beeinflussen. Z.B. sollen Pfade, die an einer konkaven Außen-Ecke liegen, keine weiteren überstumpfen Winkel enthalten, um ihre Form nicht noch komplexer zu gestalten.

#### 4.4.3 Zwischenräume

Die Generierung von Zwischenräumen in Kapitel 4.3.6 kann durch die folgenden Parameter gesteuert werden:

- **Typ und Start-Knoten**

Der Zwischenraum-Typ kann entweder als

- *Straße*,
- *Hausflur* oder
- *Wohnungsflur*

definiert werden. Für einen Haus- oder Wohnungsflur ist es zusätzlich möglich, die Position des Haupteingangs<sup>37</sup> durch die Angabe der Lage eines Begrenzungs-Knotens festzulegen. Z.B. ist der Zwischenraum eines Stockwerks ein Hausflur, der über eine Haustür *zur Straße* betreten werden kann.

- **Breite**

Dieser Wert definiert die Breite des entstehenden Zwischenraums.

---

<sup>37</sup>Der Haupteingang ist auf Stockwerks-Ebene die Haustür und auf Wohnungs-Ebene die Wohnungstür.

## 4.5 SLG: Hierarchiebildung

Für die Abbildung der hierarchischen Ebenen und deren Beziehungen gemäß Kapitel 3.4 und 4 werden folgende Annahmen abgeleitet:

- Jede Ebene besitzt einen Grundriss, indem alle Teilflächen die Elemente auf der nächst unteren Ebene repräsentieren. Die Teilflächen einer Stadt sind Stadtteile, deren Teilflächen sind Hausblöcke usw.  
Dabei ist es aber nicht zwingend erforderlich, dass nur Elemente der jeweiligen unteren Stufe repräsentiert werden. Eine Ebene kann auch Elemente der übernächsten bzw. einer anderen unteren Ebene beinhalten, solange dies im stadtgeographischen Zusammenhang sinnvoll ist.
- Eine Teilfläche definiert durch ihren Umriss die Form und Fläche eines Elements der nächst unteren Ebene.
- Die Bildung der Hierarchie sollte iterativ erfolgen und die Unterteilungsschritte auf allen Ebenen (durch Parameter) steuerbar sein.

Durch seine generischen und parametrisierten Eigenschaften kann der SLG ohne Änderungen für die Bildung dieser Hierarchie verwendet werden.

Eine neue Hierarchiestufe kann dabei durch zwei einfache Schritte erstellt werden:

1. Für jede Teilfläche wird ein Unterelement (mit eigenem SLG) erzeugt.
2. Die Positionen aller Knoten auf dem Umriss der Teilfläche werden direkt in den SLG des Unterelements übernommen und dort als Eckpunkte eingefügt.  
Für diesen Graphen kann nun seine Flächenunterteilung durchgeführt werden.

Anhand dieser beiden Schritte kann die gesamte Hierarchie durch eine einfache rekursive Funktion vollkommen automatisch unterteilt werden, vorausgesetzt es sind für jede Ebene entsprechende Parameter vorhanden. Jede Ebene kann hier aber auch einzeln betrachtet werden, um die Flächenunterteilung gezielt zu steuern.

## 4.6 SLG: Zusätzliche Manipulationen

Für das Erstellen von Zwischenräumen, z.B. für Straßen und Flure, sowie weiteren Besonderheiten wie die Generierung von Gehwegen und die Platzierung von Gebäuden, Fenstern und Türen müssen die Teilflächen des Graphen verkleinert, seine Kanten nach bestimmten Regeln geteilt oder verschoben werden. Für viele Berechnungsschritte ist es auch notwendig, eine Kantenliste vor dem gerichteten Durchlaufen so zu ordnen, dass eine bestimmte Such-Richtung gewährleistet werden kann.

Für diese Manipulationen wurden entsprechende Funktionalitäten in den SLG integriert in den folgenden Unterkapiteln beschrieben.

### 4.6.1 Kantenlisten ordnen

Für fast alle Berechnungen, die auf aneinander liegenden Kanten basieren, müssen bei Grundriss- und Aufriss-Generierung vielfach die Kanten von Graphen und

ihren Teilflächen in einer bestimmten Richtung durchlaufen werden (siehe Kapitel 4.2.2).

Für einen geschlossenen Polygonzug, der durch eine Kantenliste<sup>38</sup> definiert wird, kann dieses Durchlaufen somit links oder rechts herum bzw. **mit oder gegen den Uhrzeigersinn** erfolgen. Dabei wird die Richtung durch die Orientierung der ersten Kante und ihrem End-Knoten vorgegeben.

Für die hier verwendeten Verfahren sollen die betrachteten Kantenlisten immer **gegen der Uhrzeigersinn** durchlaufen werden, um die zugehörigen Algorithmen einheitlicher zu gestalten. Dies erleichtert auch das spätere Erstellen von 3D-Polygondaten, welche für die Grafik-API<sup>39</sup> bzgl. ihrer Sichtbarkeit immer gegen den Uhrzeigersinn definiert werden müssen.

Um die Ausrichtung der ersten Kante bzw. der gesamten Kantenliste gegen den Uhrzeigersinn zu orientieren, wird folgendes Verfahren verwendet:

1. Es wird mit einer Kante  $edge$  und deren End-Knoten  $nodeEnd_{edge}$  gestartet.
2. Es wird eine weitere Kante  $edgeNext$  gesucht, die ebenfalls  $nodeEnd_{edge}$  besitzt.
3. Ist der gemeinsame Knoten  $nodeEnd_{edge}$  nicht der Start-Knoten von  $edgeNext$ , wird diese invertiert<sup>40</sup>. Damit hat  $edgeNext$  die gleiche Orientierung wie  $edge$  und es wird die nächste Kante gesucht.

Nach diesem Schritt haben alle behandelten Kanten die gleiche Ausrichtung, wobei noch nicht bekannt ist, ob die Kantenliste mit oder gegen den Uhrzeigersinn orientiert ist. Dafür wird in einem zweiten Schritt die Art der Orientierung bestimmt und ggf. geändert:

1. Da im Normalfall durch die Kantenliste ein geschlossener Polygonzug beschrieben wird, kann dessen Flächeninhalt gemäß Kapitel 4.7 berechnet werden. Das Besondere bei dieser Flächenberechnung ist, dass der Ergebniswert positiv ist, wenn die Punkte des Polygons gegen den Uhrzeigersinn orientiert sind und negativ für den umgekehrten Fall.
2. Ist der Flächeninhalt ein negativer Wert, müssen folglich alle Kanten invertiert werden, um ihre Orientierung umzukehren.

#### 4.6.2 Flächen und Kantenlisten schrumpfen

Das Verschieben von Teilflächen-Kanten wird insbesondere für das Einfügen von Straßen, Fluren und Gehwegen, sowie das Platzieren von Gebäuden auf Grundstücken benötigt.

Prinzipiell können Flächen gleichmäßig verkleinert (geschrumpft) werden, indem alle Außenkanten um den gleichen Abstand nach innen verschoben werden. D.h. implizit, dass die Eckpunkte um einen bestimmten Betrag in die Richtung der Winkelhalbierenden versetzt werden müssen. Umgekehrt werden Flächen vergrößert (expandiert), wenn nach außen verschoben wird. Nicht gleichmäßiges

---

<sup>38</sup>Eindeutig definierte Menge von Kanten.

<sup>39</sup>Die Grafik-API ist die Schnittstelle zwischen Betriebssystem bzw. Software und Grafik-Hardware. Die beiden bekanntesten APIs sind OpenGL und DirectX (Microsoft)

<sup>40</sup>Eine Kante wird durch das Vertauschen ihres Start- und End-Knotens invertiert.

Schrumpfen oder Expandieren von Flächen bedeutet, dass die Außen-Kanten um verschiedene Beträge verschoben werden.

Um Flächen oder Kantenlisten zu schrumpfen, werden zunächst die Außenkanten der behandelten Teilfläche auf maximal vier Seiten reduziert. Dabei werden eine untere, linke, rechte und evtl. obere Seite mit Hilfe der Lage im geographischen Zusammenhang der Kanten definiert. Diese Reduzierung bzw. Gruppierung der Außen-Kanten ist notwendig, um insbesondere die untere Seite (Vorderseite) einer Teilfläche im stadtgeographischen Zusammenhang zu lokalisieren. Die Vorderseite einer Teilfläche ist für die Hierarchie-Ebenen Stadt, Stadtteil, Hausblock, Grundstück und Gebäude/Stockwerk zu einer Straße und für das Element Wohnung an der Außenwand eines Gebäudes ausgerichtet.

Die vier Seiten einer Teilfläche werden wie folgt definiert:

1. Die **untere** bzw. **vordere** Seite beinhaltet alle Kanten, die zu einer Straße bzw. für Wohnungen außen liegen. Wo eine Kante im SLG liegt, wird durch ihre Eigenschaft *Lage im geographischen Zusammenhang* festgelegt (siehe Kapitel 4.2.2).

Darüber hinaus beginnt die gefundene, gegen den Uhrzeigersinn orientierte Kantenliste immer an der linken und endet an der rechten Teilflächengrenze. Diese Grenzen sind durch die beiden enthaltenen Begrenzungs-Knoten definiert.

Für die Vorderseite gilt:

$$\begin{aligned} nodeStart_{edgesBottom} &== nodeLandmark_{right} \wedge \\ nodeEnd_{edgesBottom} &== nodeLandmark_{left} \end{aligned} \quad (23)$$

2. Die **linke Seite** *endet* am linken Begrenzungs-Knoten und beinhaltet alle Kanten der Teilfläche, die im Inneren des Graphen liegen und auf gerader Strecke gegen den Uhrzeigersinn von diesem Knoten erreicht werden.
3. Die **rechte Seite** wird wie die linke Seite definiert, nur dass diese am rechten Begrenzungs-Knoten *beginnt*.
4. In die **obere Seite** einer Teilfläche werden alle Kanten eingefügt, die nicht auf der unteren, linken oder rechten Seite liegen. Folglich bleibt diese Seite undefiniert, falls alle Teilflächen-Kanten bereits auf die anderen drei Seiten verteilt wurden.

Die Kanten einer Teilflächen-Seite können generell durch das Verändern der Positionen ihrer jeweiligen Start- und End-Knoten verschoben werden. Dabei müssen für eine Seite immer Vorgänger- und nachfolgende Seite definiert sein, um korrekte Ergebnisse an Start- und End-Knoten zu erhalten. Im Folgenden wird beschrieben, wie die Kantenliste einer Seite um einen Wert *distance* in die Richtung des Teilflächen-Inneren verschoben wird:

- Der Start-Knoten der ersten Kante (*edgeStart*) wird entlang der letzten Kante seiner Vorgängerseite (*edgePrev*) verschoben. Dabei berechnet sich die Entfernung (*distance<sub>translate</sub>*) zu seiner neuen Position mit folgender Formel:

$$distance_{translate} = \frac{distance}{\sin(\angle(\vec{v}_{edgePrev}, \vec{v}_{edgeStart}))} \quad (24)$$

- Der End-Knoten wird entlang der ersten Kante seiner nachfolgenden Seite verschoben. Die Entfernung wird wie beim Start-Knoten mit den entsprechenden Parametern berechnet.
- Alle restlichen Knoten der betrachteten Kantenliste werden entlang der Winkelhalbierenden ihrer zugehörigen Kanten ( $edgePrev$  und  $edgeNext$ ) verschoben. Dabei wird die jeweilige Distanz wie folgt berechnet:

$$distance_{translate} = \|\vec{v}_{edgeNext} - \vec{v}_{edgePrev}\| * distance \quad (25)$$

### 4.6.3 Kanten splitten

Allgemein wird eine Kante durch zwei neue ersetzt, falls ein einzufügender Knoten auf ihrer Geraden liegt. Dies ist bei der Platzierung von Begrenzungs-Knoten auf dem SLG-Umriss bzw. dessen Außen-Kanten immer der Fall.

Um einen Knoten auf einer Kante einzufügen, werden folgende Schritte ausgeführt:

1. Ist die Position des neuen Knotens nicht als Vektor sondern als Skalar (*scalar*) angegeben<sup>41</sup>, kann dieser Positionsvektor ( $\vec{p}_{node}$ ) mit der Kanten-Gleichung berechnet werden:

$$\vec{p}_{node} = f_{edge}(scalar) \quad (26)$$

2. Die zu teilende Kante wird gelöscht und zwei neue Kanten ( $edge_0$  und  $edge_1$ ), sowie der neue Knoten in die SLG-Struktur eingefügt. Dabei bekommt  $edge_0$  als Start-Knoten den Start-Knoten der gelöschten Kante und als End-Knoten den neuen Knoten zugewiesen.  $edge_1$  bekommt folglich den neuen Knoten als Start-Knoten und den End-Knoten der gelöschten Kante als End-Knoten zugewiesen<sup>42</sup>.

Für das spätere Einfügen von Fenstern und Türen (siehe Kapitel 5.2) werden Kanten mehrfach und nach bestimmten Regeln gesplittet. Soll z.B. eine  $1m$  lange Kante (für ein späteres Fenster) mittig in eine andere  $4m$  lange eingefügt werden, wird wie folgt vorgegangen:

1. Die  $4m$  lange Kante wird (durch das Einfügen eines Knotens) in eine  $1.5m$  und eine  $2.5m$  lange Kante geteilt.
2. Danach wird die  $2.5m$  lange Kante durch eine  $1m$  und eine  $1.5m$  lange Kante ersetzt.
3. Die  $1m$  lange, mittig platzierte Kante kann nun für das Fenster verwendet werden.

## 4.7 SLG: Mathematik

In diesem Kapitel sollen die wichtigsten mathematischen Grundlagen für die Funktionalitäten des SLG beschrieben werden:

<sup>41</sup>Das Skalar beträgt z.B. 0, 5 für eine mittige Platzierung.

<sup>42</sup>Wird eine Kante gelöscht, wird nur die Verbindung zweier Knoten gelöscht (siehe Kapitel 4.2.2).

### Flächeninhalt eines Polygons

Der Flächeninhalt eines Polygons kann mit Hilfe der Positionsvektoren seiner Punkte  $\vec{p}_0$  bis  $\vec{p}_{n-1}$  berechnet werden (dabei gilt  $\vec{p}_n = \vec{p}_0$ ):

$$area = \left| \frac{1}{2} \sum_{i=0}^{n-1} (\vec{p}_i \cdot x * \vec{p}_{i+1} \cdot y - \vec{p}_{i+1} \cdot x * \vec{p}_i \cdot y) \right| \quad (27)$$

Zu beachten ist, dass die Punkte in der Reihenfolge angegeben werden, in welcher sie das Polygon definieren, also mit oder gegen den Uhrzeigersinn.

### Mittelpunkt bzw. Schwerpunkt eines Polygons:

Der Mittelpunkt eines Polygons kann ebenfalls mit Hilfe der Positionsvektoren seiner Punkte  $\vec{p}_0$  bis  $\vec{p}_{n-1}$  berechnet werden (dabei gilt  $\vec{p}_n = \vec{p}_0$ ):

$$center = \frac{\sum_{i=0}^{n-1} \vec{p}_i}{n} \quad (28)$$

### Winkel zwischen zwei Richtungsvektoren

Der Winkel zwischen zwei Richtungsvektoren  $\vec{v}_0$  und  $\vec{v}_1$ , z.B. von zwei aufeinander folgenden Kanten, kann in dieser Arbeit je nach Verwendungszweck auf zwei Arten berechnet werden:

1. Die erste Methode liefert einen Winkel zwischen  $0^\circ$  und  $180^\circ$ . Die Berechnung beruht auf den Eigenschaften des *Skalarprodukts* und lautet:

$$\angle_{180}(\vec{v}_0, \vec{v}_1) = \arccos \left( \frac{\vec{v}_0 \circ \vec{v}_1}{|\vec{v}_0 \vec{v}_1|} \right) * \frac{180}{\pi} \quad (29)$$

Diese Berechnungsvorschrift kann insbesondere dann sinnvoll eingesetzt werden, wenn zwei Richtungsvektoren auf gleiche Richtung oder Orthogonalität überprüft werden und deren Orientierung unerheblich ist (z.B. bei der Pfadauswahl).

2. Die zweite Methode liefert einen Winkel zwischen  $0^\circ$  und  $360^\circ$  und basiert auf der trigonometrischen Funktion  $\tan$ , die gegenüber  $\sin$  und  $\cos$  alle vier Quadranten des Einheitskreises abdeckt.

$$\angle_{360}(\vec{v}_0, \vec{v}_1) = (\arctan(\vec{v}_1 \cdot y, \vec{v}_1 \cdot x) - \arctan(\vec{v}_0 \cdot y, \vec{v}_0 \cdot x)) * \frac{180}{\pi} \quad (30)$$

Diese Formel berücksichtigt folglich die Orientierung der Kanten und kann innerhalb eines Polygons für die Bestimmung von überstumpfen Winkeln verwendet werden.

### Schnittpunkte von Geraden

Der Schnittpunkt von zwei Geraden  $g_0$  und  $g_1$  kann durch das Gleichsetzen ihrer Gleichungen bestimmt werden. Liegen beide Geraden in der Ebene, kann diese Berechnung mit Hilfe ihrer *Determinanten* vereinfacht werden.

Dabei können die Skalare des Schnittpunkts auf den beiden Geraden durch folgende Formeln berechnet werden:

$$scalar_{g_0} = \frac{\det_{g_0}}{\det_{g_0 g_1}}, \quad scalar_{g_1} = \frac{\det_{g_1}}{\det_{g_0 g_1}} \quad (31)$$

Der Positionsvektor des Schnittpunkts  $\vec{p}$  wird nun durch das Einsetzen eines Skalars in die zugehörige Geradengleichung bestimmt:

$$\vec{p} = g_0(\text{scalar}_{g_0}) \vee \vec{p} = g_1(\text{scalar}_{g_1}) \quad (32)$$

Für eine effiziente Implementierung ist es sinnvoll, nur einen Skalarwert zu berechnen.

### Punkt-in-Polygon-Test

Um zu überprüfen, ob ein Punkt innerhalb eines Polygons liegt, gibt es eine Vielzahl von Algorithmen und Strategien, die meistens auf Winkel- und Schnittpunkt-Berechnungen basieren und deren Ergebniswerte je nach Komplexität eines Polygons stark variieren können.

Für diese Arbeit wurde ein Schnittpunkt-Verfahren ausgewählt und so angepasst, dass beliebig konvexe und konkave Polygonzüge verarbeitet und alle dabei auftretenden Spezialfälle behandelt werden. Das Verfahren wird wie folgt definiert:

1. Es wird von der Position des Testpunktes ein Strahl in eine zufällig bestimmte Richtung verschossen.
2. Dieser Strahl wird mit allen Seiten bzw. Kanten des Polygons geschnitten und für gültige Schnittpunkte ein Zähler erhöht<sup>43</sup>. Ein Schnittpunkt ist dabei gültig, wenn für seine Skalare auf der Kante  $\text{scalar}_{edge}$  und dem Strahl  $\text{scalar}_{ray}$  gilt:

$$0.0 < \text{scalar}_{edge} < 1.0 \wedge \text{scalar}_{ray} \geq 0.0 \quad (33)$$

Falls das Skalar auf der Kante  $edge$  0.0 oder 1.0 beträgt, liegt der Schnittpunkt auf einer Ecke des Polygons und der Algorithmus muss für einen korrekten Test mit einem anderen Strahl neu gestartet werden.

3. Nachdem alle Kanten des Polygons behandelt wurden, liegt der Testpunkt *innerhalb* des Polygons, falls die Anzahl der gültigen Schnittpunkte *ungerade* ist. Ist die Anzahl *gerade*, liegt der Testpunkt dementsprechend *außerhalb*.

### Punkt-auf-Kante-Test

Der hier verwendete Algorithmus für einen *Punkt-in-Polygon-Test* kann als Basis für einen Punkt-auf-Kante-Test benutzt werden. Dabei muss der oben beschriebene zweite Schritt wie folgt geändert werden:

- Für einen gültigen Schnittpunkt muss gelten:

$$0 < \text{scalar}_{edge} < 1 \wedge \text{scalar}_{ray} == 0 \quad (34)$$

Ist diese Bedingung erfüllt, liegt der Punkt auf einer Kante und der Algorithmus wird sofort beendet.

## 4.8 Zwischenfazit

Im Rahmen dieser Arbeit ist eine Graphenstruktur (SLG) entwickelt worden, die Flächen so unterteilen kann, dass Grundrisse für die morphogenetischen und städtebaulichen Elemente aus Kapitel 3.4 erstellt werden können (vgl. Abb. 34).

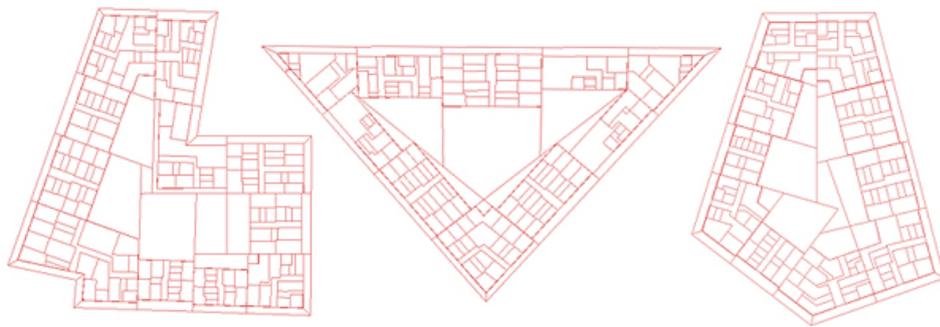


Abbildung 34: Grundrisse der Stufen Hausblock bis Wohnung mit nicht trivialen Grundflächen, erstellt mit CITYLIFE (siehe Kapitel 7).

Dabei wurden die in Kapitel 4.1 gestellten Anforderungen berücksichtigt und entsprechend umgesetzt.

Die Anzahl und Art der Parameter des SLG wurden so gewählt, dass die Unterteilung möglichst intuitiv gesteuert und durch zusätzliche Berechnungsverfahren vereinfacht werden kann. Inwieweit die Parameter des SLG abstrahiert und durch stadtgeographische Angaben berechnet werden, wird in Kapitel 6 für jede einzelne Hierarchie-Ebene beschrieben.

Die Datenstruktur wurde zusätzlich so angelegt, dass eine hierarchische Grundriss-Unterteilung einfach umgesetzt und fast vollständig automatisiert werden kann.

Für die Repräsentation von Flächendaten werden beim SLG ausschließlich Vektoren (Positionen und Richtungen von Knoten und Kanten) und Polygonzüge (Flächen) verwendet. Da diese geometrischen Elemente ebenfalls bei der Aufriss-Generierung benötigt werden, kann die SLG-Struktur ohne Anpassungen bzw. Vorverarbeitung übernommen werden. Die Teilflächen bzw. Kanten des SLG dienen beim Stadtaufriß-Verfahren also als Basisdaten. Das Besondere hierbei ist, dass für viele Algorithmen des Aufriss-Verfahrens die Funktionalitäten des SLG wiederverwendet und somit dessen Umsetzung entscheidend vereinfacht wird.

---

<sup>43</sup>Der Schnittpunkt wird wie im oberen Abschnitt *Schnittpunkte von Geraden* berechnet.

## 5 Stadtaufriss

Der Stadtaufriss ist neben dem Stadtgrundriss das zweite stadtgeographische Gestaltungselement, welches das Aussehen einer Stadt definiert. Er beschreibt die vertikale Ausdehnung bzw. Form von Städten, die insbesondere durch Fassaden und Dächer von Gebäuden beeinflusst wird. Daraus folgt, dass der Schwerpunkt eines prozeduralen Aufriss-Verfahrens auf der Generierung von Gebäuden liegt, indem aus zweidimensionalen Daten dreidimensionale Formen abgeleitet werden.

Als Grundlage bzw. Datenbasis für den Stadtaufriss dienen im Normalfall horizontale Flächendaten, die mit Hilfe von Grundrissen repräsentiert werden können. Sinnvoller Weise bietet es sich in dieser Arbeit an, die Flächendaten des SLG (siehe Kapitel 4.2) zu übernehmen, die u.a. aus Positions- und Richtungsvektoren bestehen und damit ohne weitere Bearbeitungsschritte direkt verwendet werden können.

Im hier entwickelten Aufrissverfahren werden bei der Generierung von 3D-Modellen insbesondere die Teilprobleme

- Innenräume (Innenwände),
- Fassaden (Außenwände),
- Dächer, aber auch
- Gehwege und
- Grundstücksflächen

behandelt und die entstehenden Geometrien für die 3D-Darstellung in Echtzeit optimiert. Daraus ergeben sich neben der realistischen Formgebung, Gestaltung und Parametrisierung von Gebäudemodellen die computergrafischen Schwerpunkte

- *Indexed Meshs*<sup>44</sup>,
- *Triangulierung*<sup>45</sup>,
- *Texturierung*<sup>46</sup> und
- *Level-of-Detail-Stufen*<sup>47</sup>.

---

<sup>44</sup>Ein Mesh beschreibt in der Computergrafik eine Menge von Punktkoordinaten mit verschiedenen Eigenschaften, wobei jeweils drei dieser Punkte ein Dreieck beschreiben. (Nur Dreiecke können in der 3D-Grafik angezeigt werden und sind somit das Basiselement aller dreidimensionalen Formen.) Werden diese Dreiecke nicht jeweils durch drei neu erstellte Punkte definiert, sondern Referenzen (Indizes) von Punkten verwendet, spricht man von Indexed Meshs bzw. Indexed Geometry. Punkte von Indexed Meshs können für die Darstellung von Dreiecken wiederverwendet und somit ihre Anzahl minimiert werden.

<sup>45</sup>Die Triangulierung bezeichnet in der Computergrafik ein komplexes Verfahren, bei dem aus einer Punktmenge ein Netz aus Dreiecken generiert wird, so dass eine geschlossene Oberfläche entsteht und angezeigt werden kann.

<sup>46</sup>Die Texturierung bezeichnet ein Verfahren der Computergrafik, das dazu dient, die Oberflächen von 3D-Modellen mit zweidimensionalen Bildern (Texturen) auszustatten.

<sup>47</sup>Level-of-Detail (LOD) beschreibt in der Computergrafik die Detailstufe eines Meshs in einem 3D-Modell (siehe Kapitel 5.3.4).

Durch die Verwendung der SLG-Grundrisse mit Informationen über den inneren Aufbau von Gebäuden, kann die innere und äußere Struktur dieser Stadtelemente in Beziehung gesetzt und somit eine realistische Gestaltung der entstehenden Geometrien erreicht werden. D.h., dass die Lage der Gebäude-Innenräume die allgemeine Platzierung von Fenstern und Türen in den Innen- und Außenwänden festlegt. Die dafür erforderlichen Erweiterungen der vorhandenen Grundrisse werden in Kapitel 5.2 behandelt.

Diese Informationen sind bei allen vorgestellten prozeduralen und photogrammetrischen Verfahren zur Generierung von Stadt- und Gebäudemodellen (siehe Kapitel 2.3) nicht vorhanden, bzw. werden dort nicht behandelt.

Folglich muss eine Vielzahl von Einzelverfahren entwickelt, in Beziehung gesetzt und so parametrisiert werden, dass eine sinnvolle, realistische und vor allem umsetzbare Gesamtlösung für die Aufriss-Generierung entstehen kann. In den folgenden Unterkapiteln werden zunächst die dafür notwendigen Anforderungen definiert, sowie einzelne vorhandene Teillösungen beschrieben und bewertet. Anschließend wird das im Rahmen dieser Arbeit entwickelte prozedurale Gesamtverfahren zur Grundriss-Erweiterung und Aufriss-Generierung beschrieben.

## 5.1 Prozedurale Anforderungen und Ansätze für 3D-Formen

In diesem Kapitel sollen alle Anforderungen für ein prozedurales Verfahren zur Generierung von dreidimensionalen Aufrissformen definiert und anschließend vorhandene Techniken vorgestellt und bewertet werden.

Die Anforderungen und Funktionalitäten werden wie folgt definiert:

- Die Eingangsdaten sollen als zweidimensionale Flächendaten vorliegen und direkt verarbeitet werden können.
- Die entstehenden Formen sollen im stadtgeographischen Zusammenhang realistisch aussehen und durch geeignete Parameter gesteuert werden.
- Es sollen insbesondere Wände, Fenster, Türen, Decken, Böden und Dächer von Gebäuden generiert werden.
- Für die korrekte Beleuchtung und Texturierung der entstehenden Modelle müssen Punktnormalen<sup>48</sup> *automatisch* und Texturkoordinaten nach vordefinierten Regeln an allen Punkten berechnet werden.
- Für die optimierte Darstellung in Echtzeit sollen die Modelle durch eine minimale Anzahl von Vertices<sup>49</sup> und Polygonen definiert werden. Folglich muss die Unterstützung von Indexed Meshs ebenfalls vorhanden sein.
- Das Verfahren soll die *automatische* Generierung von mehreren Level-of-Detail-Stufen unterstützen, um die Verwendung der Modelle in Echtzeit-Anwendungen *entscheidend* zu optimieren.

---

<sup>48</sup>Die Normale eines Punktes (Punktnormale) ist ein normierter Richtungsvektor, der so ausgerichtet ist, dass z.B. die Beleuchtung von Dreiecken, welche den zugehörigen Punkt enthalten, korrekt berechnet werden kann (siehe Kapitel 5.3.2).

<sup>49</sup>Ein Vertex ist ein Punkt mit Positionsvektor und weiteren Eigenschaften, wie z.B. Index, Normalenvektor und Texturkoordinate.

### 5.1.1 Vorhandene Techniken

Eine einheitliche Gesamtlösung für das Generieren von Gebäude-Geometrien, wie sie in dieser Arbeit benötigt wird, konnte bei der Literaturrecherche nicht gefunden werden. Für die prozedurale Generierung von Fassaden und Dächern sind aber Einzelverfahren vorhanden und sollen im Folgenden beschrieben werden (*Split Grammar* und *Straight-Skeleton*). Davor wird zur Vollständigkeit ein Verfahren für die allgemeine Generierung von dreidimensionalen Formen (*Constructive Solid Geometry*), sowie für die Triangulierung von Punktmengen vorgestellt (*Delaunay-Triangulierung*).

Für alle Verfahren lagen zum Zeitpunkt dieser Arbeit keine frei verfügbaren bzw. brauchbaren Implementierungen vor. Eine eigene Umsetzung dieser Einzelverfahren mit einer zusätzlichen Anpassung und den damit verbundenen Erweiterungen, die für die hier gestellten Anforderungen notwendig sind, hätte den Zeitrahmen dieser Arbeit gesprengt. Daher werden im Rahmen dieser Arbeit für alle Teilprobleme eigene angepasste Verfahren entwickelt und in einer Gesamtlösung zusammengeführt.

Die im Nachfolgenden vorgestellten Techniken beschreiben aber dennoch interessante Ansätze, die in den Entwicklungsprozess der eigenen Teilverfahren mit eingeflossen sind.

#### Constructive Solid Geometry (CSG)

Die Constructive Solid Geometry (CSG) bzw. Konstruktive Festkörpergeometrie ist in der Computergrafik ein Verfahren zum Modellieren von Körpern. Mit Hilfe dieser Technik können komplexe dreidimensionale Oberflächen erzeugt werden, indem primitive Körper (z.B. Würfel, Zylinder, Pyramiden oder Kugeln) miteinander verknüpft werden. Diese Verknüpfungen basieren auf booleschen Operationen und werden für die CSG in Vereinigung, Differenz und Schnitt unterteilt [RaWa01], [TrSy03] (vgl. Abb. 35).



Abbildung 35: CSG-Konstruktionsprozess (Quelle: [RaWa01]).

Prinzipiell ist die CSG für die prozedurale Generierung von Aufriss-Geometrien gut geeignet, da z.B. eine Wand mit Fenstern durch die wiederholte Anwendung des Differenz-Operators mit nur zwei Würfelkörpern erstellt werden kann.

Inwieweit die benötigte Berechnung von Punkt-Normalen und Texturkoordinaten bei der CSG überhaupt vorgesehen ist bzw. effizient eingebaut werden kann, wird aus den vorhandenen Quellen nicht ersichtlich.

Das in dieser Arbeit entwickelte Verfahren für die Generierung von 3D-Geometrie verwendet ebenfalls einfache Basis-Körper, verzichtet allerdings auf die booleschen Operationen, um insbesondere die prozedural Berechnung von Texturkoordinaten zu ermöglichen (siehe Kapitel 5.3.1).

#### Delaunay-Triangulierung

Mit dem Verfahren der Delaunay-Triangulierung kann im Bereich der Computer-

grafik eine Menge von (zweidimensionalen) Punkten in ein darstellbares Dreiecks-Mesh umgewandelt werden. Dabei erfüllen alle entstehenden Dreiecke die *Umkreisbedingung*<sup>50</sup> und bilden somit eine möglichst gleichmäßig unterteilte Oberfläche [RoSc01], [AnVi01] (vgl. Abb. 36). Dieses Verfahren wird normalerweise im zweidimensionalen Raum durchgeführt, kann aber auch für dreidimensionale Positionsvektoren erweitert und umgesetzt werden [ReKl01].

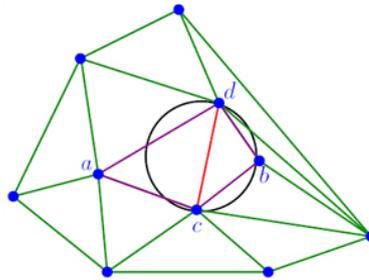


Abbildung 36: Delaunay-Triangulierung mit eingezeichneter Umkreisbedingung (Quelle: [AnVi01]).

Damit wäre die Delaunay-Triangulierung eine geeignete Technik, um z.B. Grundstücksflächen, Stockwerksböden und -decken zu unterteilen. Allerdings wird, wie bei der CSG, die Behandlung von Punkt-Normalen und Texturkoordinaten in diesem Verfahren nicht ersichtlich.

Da im hier entwickelten Verfahren die Triangulierung von Polygonen nicht vernachlässigt werden kann, soll in Kapitel 5.3.3 ein eigener Algorithmus für diese Problematik entwickelt werden.

### Split Grammar für Fassaden (L-Systeme)

Eine Split Grammar ist ein Verfahren zur Generierung von komplexen Geometrieformen, bei dem eine einfache geometrische Form (z.B. Würfel) auf Basis von Regeln durch weitere Formen ersetzt wird [WoWiSi01].<sup>51</sup> Die einzige Bedingung dabei ist, dass die neuen Formen die Ausmaße der zu ersetzenden Form nicht überschreiten dürfen. Ein Ersetzungsschritt könnte z.B. ein großes Wandstück, bestehend aus einem Würfel, so durch drei Kleinere ersetzen, dass eine Öffnung für eine Tür entsteht (vgl. Abb. 37).

Die Regeln einer Split Grammar können in textueller Form definiert werden und steuern einen zusätzlichen Algorithmus zur Geometrie-Erzeugung. Wird die Technik der Split Grammar iterativ bzw. rekursiv angewendet, können bereits mit einem simplen und kleinen Regelwerk detailreiche Geometrien erzeugt werden. In der Arbeit von MÜLLER (2001) werden mit dieser Technik (dort als *Shape Grammar* bezeichnet) bereits vollständige Außenformen von Gebäuden erstellt. Allerdings können Innenräume "durch die Split Grammar nicht modelliert werden" [PaMu01].

<sup>50</sup>Die Umkreisbedingung setzt voraus, dass der Umkreis eines Dreiecks keine weiteren Punkte der vorgegebenen Punktmenge enthalten darf. Dadurch werden die Dreiecke des Meshs mit maximalen Innenwinkeln generiert und sind somit möglichst gleichmäßig geformt.

<sup>51</sup>Das Split Grammar-Verfahren kann wie die L-Systeme den Ersetzungsgrammatiken zugeordnet werden (siehe Kapitel 4.1.1).

Da die Wände eines Gebäudes i.A. ausschließlich mit Würfeln beschrieben werden können, wäre dieser Ansatz insbesondere für die Generierung von Fassaden geeignet<sup>52</sup>. Dabei können allerdings nur sehr einfache Regeln mit einer geringen Rekursionstiefe benutzt werden, um die Anzahl der entstehenden Polygone niedrig zu halten und somit die Darstellung in Echtzeit zu gewährleisten.

Das Prinzip der Split Grammar soll in den hier entwickelten Verfahren für die Generierung von dreidimensionalen Aufriss-Elementen (Kapitel 5.3), sowie der Erstellung von Level-of-Detail-Stufen (Kapitel 5.3.4) und der Platzierung von Fenstern und Türen (Kapitel 5.2) ansatzweise verwendet werden.



Abbildung 37: Erstellung einer Fassade mit der Split-Grammar-Technik (Quelle: [MuVeU1VG01]).

### Straight-Skeleton-Algorithmus für Dächer

Der Straight-Skeleton-Algorithmus berechnet das *Skelett* (Straight-Skeleton) eines Polygons. Dieses Skelett ist durch einen Linienzug definiert, auf dem jeder Punkt von seinen nächstgelegenen Polygonseiten in einer maximale Distanz entfernt liegt. Das Straight-Skelett kann mit Hilfe der Winkelhalbierenden eines Polygons bestimmt werden [FeCa01], [FeOb01] (vgl. Abb. 38).

Dieser Algorithmus ist mit einer Erweiterung gut für die Generierung von Walm-dachformen geeignet und findet bereits in einigen prozeduralen Aufriss-Verfahren Verwendung [ToKe06], [HaSe04].

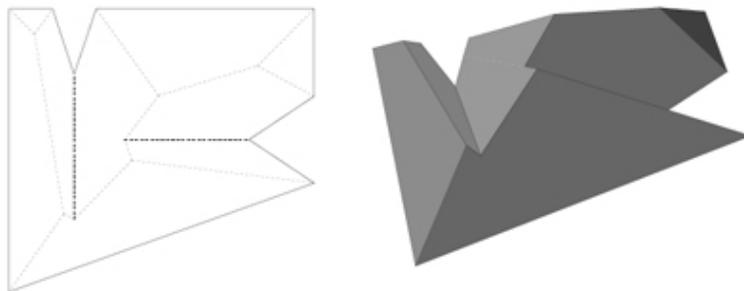


Abbildung 38: Straight-Skeleton eines Polygons und die daraus entstehende 3D-Form (Quelle: [AiAu01]).

### 5.1.2 Zwischenfazit

Alle vorgestellten Verfahren wären für einige der hier definierten Teilprobleme der Aufriss-Generierung zwar verwendbar, allerdings erzeugen alle diese Techniken keine für Echtzeit-Anwendungen direkt brauchbaren Ergebnisse und sind ohne

<sup>52</sup>In [MuWoHaU105] und [SiHa05] wurden bereits mit einer erweiterten Split Grammar und komplexen Regelsätzen realistische Fassaden-Formen generiert.

eine langwierige Anpassung in dieser Arbeit nicht brauchbar. Insbesondere die Thematik von Texturkoordinaten und Punktnormalen wird in allen Verfahren nicht beschrieben oder gänzlich vernachlässigt. Die vorgestellten Verfahren liefern dennoch einige interessante Ansätze und informationstechnische Grundlagen, die für das hier entwickelte Aufriss-Verfahren aufgegriffen werden. Dazu zählt u.a. die Ersetzungstheorie der Split Grammar.

## 5.2 Grundriss-Erweiterung

Für das Aufriss-Verfahren werden die Grundriss-Daten aus den Strukturen des SLG (siehe Kapitel 4.2) übernommen. Dabei können die Knoten, Kanten und Flächen direkt und ohne Anpassungen für die hier notwendigen Geometrie-Berechnungen verwendet werden, da sie als Positions- und Richtungs-Vektoren vorliegen.

Die Datenstruktur des SLG repräsentiert damit die Schnittstelle zwischen Stadtgrundriss und -aufriss bzw. stellt die Datenbasis für den Einstiegspunkt der Aufriss-Generierung bereit.

Diese vorhandenen Grundrissdaten können nun durch einen entsprechenden Algorithmus direkt in die Höhe extrudiert<sup>53</sup> werden, um z.B. die 3D-Geometrien der Außenwände eines Gebäudes zu bilden. (Das Gebäudedach wird durch einen separaten Algorithmus danach generiert.) Die dabei fehlenden Fenster und Türen könnten im Anschluss durch entsprechende Texturen an den Fassaden angebracht werden. Dafür werden meistens Bilder von Fassaden auf die Flächen der Wände gelegt, um den visuellen Eindruck einer detaillierten Außenwand mit Fenstern und Türen zu erzeugen. Mit Hilfe einer Split Grammar oder einem vergleichbaren Verfahren könnten die Fenster und Türen allerdings auch durch komplexe Geometrien repräsentiert werden.

Ein Verfahren zur Erzeugung von (einfachen) Gebäude-Modellen bietet sich immer dann an, wenn keine Informationen über die Innenräume von Gebäuden vorliegen bzw. behandelt werden und die realistische Platzierung von Fenstern und Türen bzgl. der Innenkonstruktion vernachlässigt werden kann. Für alle vorgestellten prozeduralen und photogrammetrischen Verfahren (siehe Kapitel 2.3) wird der Aufriss in vergleichbarer Form generiert und der innere Aufbau von Gebäuden mit seinen Auswirkungen auf die Gestaltung von Fassaden (bzw. die Auswirkungen von Fassaden auf Innenräume) nicht behandelt.

In dieser Arbeit können mit Hilfe der SLG-Grundrisse Fenster und Türen in Abhängigkeit des inneren Aufbaus eines Gebäudes/Stockwerks an den Innen- und Außenwänden platziert werden. Dabei wird diese Platzierung noch *vor* der Generierung von 3D-Formen direkt in den jeweiligen Grundriss durch den SLG eingefügt. Diese Vorgehensweise bietet sich an, da der SLG bereits über alle notwendigen Funktionalitäten verfügt und damit keine weiteren Verfahren und Implementierungen benötigt werden. Zusätzlich kann das eigentliche Aufriss-Verfahren zur Generierung von 3D-Modellen gegen diese Art der

---

<sup>53</sup>*Extrude* bzw. Extrudieren bezeichnet in der Computergrafik i.A. ein Verfahren, bei dem eine zweidimensionale Fläche (Polygon) in einen (dreidimensionalen) Würfel umgewandelt wird. (Entsprechend wird aus einer Kante eine Fläche gebildet.)

Grundriss-Erweiterung sinnvoll abgegrenzt werden.

In den folgenden Unterkapiteln werden zunächst Grundrisse für die einzelnen Stockwerke eines Gebäudes erstellt und anschließend die Platzierung von Fenstern und Türen, aber auch der Treppenhäuser, für die Stadtobjekte Gebäude/Stockwerk und Wohnung beschrieben. Für letzteren Punkt werden die Kanten des jeweiligen Grundrisses vielfach nach bestimmten Parametern durch den SLG gesplittet (siehe Kapitel 4.6.3).

Nach diesen Berechnungsschritten ist der Grundriss so erweitert worden, dass direkt mit der Generierung von 3D-Formen begonnen werden kann.

### 5.2.1 Einfügen von Stockwerken

Der SLG erzeugt zunächst nur Grundrisse in der Ebene, d.h. für alle Gebäude liegen nach der Unterteilung ihrer Flächen nur die Grundrisse der Erdgeschosse vor. Für ein mehrstöckiges Gebäude werden allerdings Grundrisse für *jedes* Stockwerk benötigt, damit die Platzierung von Fenstern und Türen einzeln und individuell durchgeführt werden kann. Dies ist sinnvoll, um insbesondere die Stockwerke von Mischgebäuden<sup>54</sup> je nach ihrer Flächennutzung zu gestalten.

Die Obergeschosse von mehrstöckigen Gebäuden werden wie folgt generiert:

1. Für jedes Obergeschoss wird ein Stockwerk-Objekt inklusive Graph angelegt und in einer Liste des zugehörigen Gebäude-Objekts gespeichert. Ein späterer Zugriff wird über die Geschosnummer ermöglicht.
2. Die gesamte SLG-Struktur des Ergeschosses wird kopiert und in die jeweiligen Graphen der oberen Stockwerke eingefügt. Die Aufteilung der Stockwerke in Flure und Wohnungen bleibt damit über alle Stockwerke erhalten.

Die Grundrisse von Kellergeschossen können mit diesem Verfahren ebenfalls generiert werden, allerdings beeinflussen sie nicht die Aufriss-Gestalt einer Stadt und werden in dieser Arbeit nicht behandelt.

### 5.2.2 Platzierung von Eingängen

Eingänge bzw. Türen werden vor den Fenstern in die Gebäude-Grundrisse eingefügt, da spätere Wandstücke in bestimmten Fällen<sup>55</sup> beide Elemente enthalten können und dabei den Eingängen ein höherer Stellenwert zugeschrieben wird. Ohne Eingang wäre z.B. das Betreten eines Geschäfts von der Vorderseite nicht direkt möglich.

In dieser Arbeit werden folgende Typen und Eigenschaften von Eingängen bzw. Türen in einem Gebäude definiert:

---

<sup>54</sup>Die Stockwerksflächen von Mischgebäuden werden im stadtgeographischen Zusammenhang unterschiedlich genutzt. Z.B. kann das Erdgeschoss eines Gebäudes Geschäfte enthalten und Obergeschosse entweder als Büro- oder Wohnfläche genutzt werden.

<sup>55</sup>Z.B. enthalten die Außenwände vieler Geschäfte an der Straßenseite sowohl Schau- fenster als auch eigene Eingänge. Das zugehörige Gebäude besitzt dabei einen zusätzlichen Haupteingang.

- **Haustür**  
Die Haustür liegt an einer Außenwand im Erdgeschoss des Gebäudes und grenzt im Normalfall an einen Hausflur. Pro Gebäude ist nur eine Haustür vorhanden und diese liegt mittig im zugehörigen Außen-Wandstück des Hausflurs. Im SLG-Grundriss wird eine Haustür folglich in einer Innen(Zwischenraum)-Fläche eines Gebäudes/Stockwerks eingefügt.
- **Wohnungstür**  
Die Wohnungstür liegt auf der Grenze von Haus- und Wohnungsflur. Pro Wohnung ist nur eine Wohnungstür vorhanden und diese liegt mittig im zugehörigen Wandstück des Wohnungsflurs. Im SLG-Grundriss wird eine Wohnungstür damit in einer Innen(Zwischenraum)-Fläche einer Wohnung und des zugehörigen Stockwerks eingefügt.
- **Zimmertür**  
Die Zimmertür liegt entweder zwischen dem Wohnungsflur und einem Raum oder ist die Verbindungstür zwischen zwei Räumen. Die Zimmertüren liegen mittig in der zugehörigen Zimmerwand. Im SLG-Grundriss wird eine Zimmertür dabei in eine Innen- und Außen-Fläche oder in zwei Außen-Flächen einer Wohnung eingefügt.
- **Eingangstür für Einzelhandel- und Gastronomie-Räume**  
Die Eingangstür für Einzelhandel- und Gastronomie-Räume liegt z.B. an der Vorder- bzw. Straßenseite des Erdgeschosses. Diese Türen können mittig, aber auch links oder rechts in der zugehörigen Gebäudewand ausgerichtet sein. Im SLG-Grundriss wird dafür eine Tür in eine Außen-Fläche einer Wohnung eingefügt.

Alle Türen haben eine bestimmte Breite, die bei der Erweiterung des Grundrisses die wichtigste Eigenschaft darstellt.

Für das Einfügen von Türen wird dabei wie folgt vorgegangen:

1. Je nachdem wo die Tür platziert werden soll, werden diejenigen Teilflächen-Kanten in der SLG-Struktur gesucht, die eine bestimmte *Lage im geographischen Zusammenhang* besitzen. Z.B. werden für eine Wohnungstür nur Kanten behandelt, die *an einem Hausflur* liegen und zu einer innen liegenden Teilfläche der Wohnung gehören (dem Wohnungsflur).
2. Wurden mehrere Kanten gefunden, wird die längste ausgewählt.
3. Die ausgewählte Kante wird nun, wie in Kapitel 4.6.3 beschrieben wurde, zweimal gesplittet, so dass aus einer Kante drei kleinere neue Kanten entstehen. Dieser Schritt wird durch entsprechende Parameter (Türbreite und minimale Abstände zu Nachbarwänden) kontrolliert. Im Prinzip handelt es sich hierbei um eine Ersetzungsgrammatik auf Kantenebene.
4. Die mittlere dieser entstehenden Kanten wird für die Tür markiert, indem ihre *Lage im geographischen Zusammenhang* angepasst wird. Z.B. wird einer Wohnungstür der Wert *innen liegender Haupteingang* zugewiesen. Bei der späteren Aufriss-Generierung kann allein durch diese Eigenschaft eine Grundriss-Kante exakt lokalisiert und der zugehörige Geometrietyp erzeugt werden.

Durch die wiederholte Anwendung dieser vier Schritte werden alle Türen, die im gesamten Gebäude vorhanden sind, in die Grundrisse der jeweiligen Stadtelemente (Gebäude/Stockwerk und Wohnung) platziert, indem Kanten in den SLG eingefügt und für die spätere Aufriss-Generierung markiert werden.

### 5.2.3 Platzierung von Fenstern

Die Fenster eines Gebäudes werden nach den Türen im Gebäude-Grundriss bzw. -SLG platziert und sind insbesondere in den Obergeschossen das zentrale Gestaltungselement für Gebäude-Fassaden.

Alle Fenster werden an den Außenwänden der einzelnen Räumlichkeiten eingefügt. Damit werden sie ausschließlich an den *außen liegenden* Kanten der Grundrisse von Wohnungen oder Stockwerken platziert.

Wie die Türen, haben alle Fenster u.a. eine bestimmte Breite. Um die Verwendung von großen zusammenhängenden Schaufensterreihen bis hin zu einzelnen kleinen Fenstern zu ermöglichen, wird die Platzierung von Fenstern in zwei Arten unterschieden:

#### 1. Mehr- oder einfach getrennte Platzierung

Die Fenster werden hier einzeln und mit regelmäßigen Abständen untereinander in einer Außenwand platziert. Eine Wand kann dabei nur ein oder mehrere Fenster haben. Z.B. können in einen Raum mit einer  $5m$  langen Außenwand zwei  $1m$  breite Fenster mit einem Abstand von wiederum  $1m$  untereinander und zu den Zimmergrenzen eingefügt werden.

#### 2. Mehrfach zusammenhängende Platzierung

Bei dieser Platzierung werden die Fenster direkt nebeneinander, also ohne Zwischenwände (der Abstand untereinander beträgt  $0m$ ), mit einem bestimmten Abstand zu den Nachbarwänden, eingefügt. Z.B. können in die  $8m$  lange Außenwand eines Geschäfts drei  $2m$  breite Schaufenster mit einem seitlichen Abstand von jeweils  $1m$  eingefügt werden.

Für das Einfügen von Fenstern wird dabei wie folgt vorgegangen:

1. Alle *außen liegenden* Kanten einer Wohnung, in die Fenster eingefügt werden sollen und die nicht an ein Nachbargebäude grenzen, werden gesucht.
2. Alle gefundenen Kanten werden nun, wie in Kapitel 4.6.3 beschrieben wurde, mindestens zweimal gesplittet (für mindestens ein Fenster). Sollen mehrere Fenster eingefügt werden, entstehen bei den ersten zwei Splits eine Kante mit Länge des Abstands zur Nachbarwand, eine Kante mit Länge der Fensterbreite und eine Kante mit der restlichen Länge, welche für das Einfügen des zweiten Fensters in die Wand wiederum zweimal gesplittet werden kann usw.
3. Die Kanten, welche mit der Länge der Fensterbreite erstellt wurden, werden nun markiert, indem ihre *Lage im geographischen Zusammenhang* angepasst wird. Dabei wird ihnen je nach Lage der Außenwand entweder der Wert *außen liegendes Fenster zur Straße* oder *außen liegendes Fenster zu einem Zwischenraum* zugewiesen.

Nach Abschluss dieses Schrittes ist der Grundriss um Fenster und Türen erweitert worden und enthält damit, bis auf die Lage des Treppenhauses, alle benötigten Informationen für die Aufriss-Generierung.

#### 5.2.4 Platzierung von Treppenhäusern

Als letzten Schritt der Grundriss-Vorbereitung für den Aufriss, wird das Treppenhaus bei mindestens zweistöckigen Gebäuden eingefügt. Dabei wird ein bestimmter Raum (Teilfläche einer Wohnung) des Grundrisses ausgewählt und für die spätere Geometrie-Generierung abgespeichert.

Die Kriterien für einen Raum, der als Treppenhaus verwendet werden kann, sind wie folgt definiert:

- Der Raum ist viereckig. (Dies erleichtert das Einfügen von Geometrien für Treppen.)
- Der Raum hat eine maximale Fläche bei minimalem Umfang. D.h. die vier Seiten der Fläche sind ungefähr gleich lang und der Flächeninhalt ist gleichzeitig minimal. (In einem Gebäude kann dabei z.B. eine Teilfläche mit  $\approx 5m^2$  ausgewählt werden.)

Nachdem eine Wohnungs-Teilfläche ausgewählt und abgespeichert wurde, wird zusätzlich ein Eingang zum Hausflur (*innen liegender Treppeneingang*) in eine ihrer Außen-Kanten eingefügt.

Nun ist der Gebäude-Grundriss vollständig mit allen für den Aufriss notwendigen Informationen erweitert worden und das eigentliche Aufriss-Verfahren zur Modell-Generierung wird gestartet.

### 5.3 Aufriss-Generierung (3D-Modelle)

Nachdem die vorhandenen Grundrisse des SLG vorbereitet wurden, können nun die einzelnen 3D-Modelle<sup>56</sup> für jedes Gebäude aber auch die der Hausblöcke, welche aus Gehwegen und Grundstücksflächen bestehen, generiert werden.

Dafür werden die Modelle aus einzelnen Basis-Elementen zusammengesetzt, wobei für jeden dieser Bausteine Texturkoordinaten und Punkt-Normalen, entsprechend ihrer Lage im Modell, berechnet werden. Alle verwendeten Basis-Elemente werden dabei als indizierte (wiederverwendbare) und triangulierte Punkte in das Mesh des zugehörigen Modells eingefügt. Durch die Verwendung von Indizes, um die Dreiecke des Meshs zu definieren, wird der Speicherplatz verringert und die späteren Zugriffe der Grafik-API beschleunigt.

Die Kanten des zugrunde liegenden SLG-Grundrisses bestimmen in diesem Verfahren die Form und horizontalen Ausmaße dieser Basis-Elemente. Welche Kanten verwendet und wie die 3D-Modelle letztendlich zusammengesetzt werden, wird in Kapitel 5.3.4 beschrieben.

---

<sup>56</sup>Ein 3D-Modell ist i.A. ein Container-Element für 3D-Geometrien und ggf. Animationen. Für jede Level-of-Detail-Stufe enthält ein 3D-Modell mindestens ein Mesh mit Punkt- und Dreiecksdaten. (Mehrere Meshs in einer Level-of-Detail-Stufe werden im Normalfall in einer Baum-Struktur hierarchisch angeordnet, um spätere Transformationen zu erleichtern.)

Des Weiteren können sämtliche 3D-Modelle mit beliebig vielen Detailstufen (*Level-of-Detail*-Stufen) generiert werden, um ihre Darstellung in Echtzeit-Anwendungen zu optimieren (siehe Kapitel 5.3.4).

Nach diesem prozeduralen Verfahren liegen schließlich 3D-Modelle vor, deren Aufbau und Struktur für eine Echtzeit-Darstellung optimal ist und bei gleichwertiger Gestaltung nicht verbessert werden kann. Durch die Verwendung der SLG-Grundrisse basiert die Gestaltung aller Modelle auf stadtgeographischen Gesetzmäßigkeiten und ist somit möglichst realistisch.

### 5.3.1 Basis-Elemente und Indexed Geometry

Für den Aufbau von vollständigen Gebäude- und Hausblock-Modellen werden in diesem Kapitel einfache Basis-Elemente definiert, die wiederverwendbar sind und durch eine Reihe von Parametern gesteuert werden können. Sie werden später auf den Kanten der SLG-Grundrisse platziert und ihre allgemeine Form durch die *Lage im geographischen Zusammenhang* und ihre Ausmaße maßgeblich durch den Richtungsvektor dieser Kanten bestimmt. Folgende Formen werden durch Basis-Elemente repräsentiert:

- Wand (innen und außen)
- Fenster (innen und außen)
- Eingänge (innen und außen)
- Boden und Decke
- Treppenhaus
- Dach
- Gehweg
- Hausblock- und Grundstücksfläche

Für das Einfügen von Basis-Elementen in das Mesh eines zugehörigen 3D-Modells wird wie folgt vorgegangen:

1. Es wird eine Menge von Punkten definiert, welche die Form des jeweiligen Basis-Elements festlegen.
2. Das Mesh wird für jeden dieser Punkte nach einem bereits vorhandenen Punkt mit gleicher Position durchsucht. Wenn ein Punkt gefunden wurde, wird sein Index zurückgegeben. Bei erfolgloser Suche wird der Punkt neu erstellt, in das Mesh eingefügt und ebenfalls sein Index zurückgegeben. Der Index eines neuen Punktes ergibt sich dabei aus  $numVertices_{mesh} - 1$ , d.h. der erste eingefügte Punkt hat Index 0, der zweite 1 usw.
3. Nun sind für alle im Basis-Element benötigten Punkte die Indizes vorhanden und die Dreiecke können definiert werden. Dabei wird jedes Dreieck des Meshs durch drei Indizes (z.B. 0, 4, 99) *gegen den Uhrzeigersinn* definiert (Triangulierung). Für nicht triviale Triangulierungen wird das Verfahren aus Kapitel 5.3.3 verwendet.

- Über die vorliegenden Indizes kann auf die Punkte des Meshs zugegriffen werden, um ihre Normalen und Texturkoordinaten in Abhängigkeit ihrer Lage im Mesh zu bestimmen (siehe Kapitel 5.3.2).

Wichtig ist hierbei, dass alle Basis-Elemente des 3D-Modells innerhalb eines einzelnen Mesh-Objekts definiert werden. Dies entlastet die Grafik-Hardware bei der Echtzeit-Darstellung, da für jedes Mesh die Hardware-Speicher für Punkt- und Dreiecksdaten pro Frame<sup>57</sup> ausgetauscht bzw. beschrieben werden müssen.

Folgende Basis-Elemente werden für die Generierung von Gebäude- und Hausblock-Modellen verwendet:

- **Wand (innen und außen):**

Eine Wand kann durch eine einfache vertikale viereckige Fläche (Polygon<sup>58</sup>) mit bestimmter Breite (entspricht der Länge der zugehörigen Grundriss-Kante) und Höhe (Stockwerkshöhe) repräsentiert werden. Dabei muss zwischen Innen- und Außenwand-Element unterschieden werden, um eine entsprechende Texturierung zu ermöglichen. Zusätzlich wird das Polygon der Innenwand um einen bestimmten Abstand (Wanddicke) ins Innere des Gebäudes geschoben (vgl. Abb. 39).

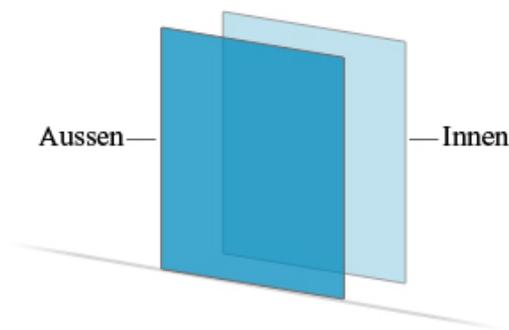


Abbildung 39: Geometrie eines Wand-Elements.

Mit folgendem Pseudocode soll die Generierung eines Außenwand-Elements an einer Grundriss-Kante veranschaulicht werden.

```
1 CreateWallOutside( edge )
2 {
3   /* calculate positions */
4   position0 = edge.nodeStart.position;
5   position1 = edge.nodeEnd.position;
6   position2 = position1 + heightWall;
7   position3 = position0 + heightWall;
8
9   /* search and get vertices */
10  vertex0 = mesh.GetVertex( position0 );
11  vertex1 = mesh.GetVertex( position1 );
12  vertex2 = mesh.GetVertex( position2 );
13  vertex3 = mesh.GetVertex( position3 );
```

<sup>57</sup>Ein Frame bezeichnet beim 3D-Rendering den einmaligen Aufbau des Bildschirm. Die Anzahl der Frames pro Sekunde (fps) kann als Leistungsindikator für 3D-Anwendungen genutzt werden. Die fps entspricht der Bildwiederholungsrate (Hz) bei Displays.

<sup>58</sup>Ein Polygon beschreibt i.A. eine ebene Fläche mit mindestens drei Punkten. In dieser Arbeit soll ein Polygon immer durch eine viereckige Fläche repräsentiert werden.

```

14
15  /* create trivial triangulation */
16  mesh.CreateTriangle( vertex0.index, vertex1.index, vertex2.index );
17  mesh.CreateTriangle( vertex0.index, vertex2.index, vertex3.index );
18  }

```

Listing 4: Generierung eines Wand-Elements im Pseudocode.

- (Zeile 4-7) Die vier Eckpunkte des Polygons werden berechnet. Dabei werden die Positionen gegen den Uhrzeigersinn, bzw. in folgender Reihenfolge, bestimmt:
  1. unten links
  2. unten rechts
  3. oben rechts
  4. oben links
- (Zeile 10-13) Die neu berechneten Positionen werden im Mesh-Objekt gesucht und je nach Suchergebnis ein vorhandenes oder neues Punkt-Objekt zurückgegeben.
- (Zeile 16-17) Ein Polygon besteht aus zwei Dreiecken und da die Reihenfolge der Punkte bekannt ist, kann die Triangulierung dieser Fläche direkt bestimmt werden.

Diese Funktion wird für alle übrigen Basis-Elemente in vergleichbarer Form umgesetzt.

- **Fenster (innen und außen)**

Ein Fenster wird in dieser Arbeit maßgeblich durch drei Polygone mit einer bestimmten Höhe repräsentiert. (Die Summe der Höhen entspricht der Stockwerkshöhe.) Das erste Polygon liegt zwischen Boden und Fenstersims und definiert die untere Fensterwand, das zweite definiert das eigentliche Fenster und das dritte reicht von der Oberkante des Fensters bis zur Decke und definiert die obere Fensterwand. Zusätzlich besteht die Möglichkeit, ein Vordach mit einem doppelseitigen Polygon oder einen Fensterrahmen mit Hilfe von vier weiteren Polygonen einzufügen. Ist dieser Rahmen vorhanden, sollte das Polygon des Fensters sinnvollerweise um einen bestimmten Betrag ( $\frac{Wandbreite}{2}$ ) in die Wand versetzt werden (vgl. Abb. 40).

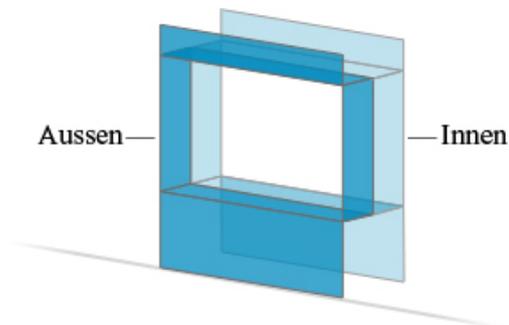


Abbildung 40: Geometrie eines Fenster-Elements.

Wie bei einer Innenwand werden alle Fensterwände eines Innen-Fensters um den Betrag der Wandbreite nach innen verschoben. Reicht das Fenster bis zum Boden und/ oder bis zur Decke, werden die entsprechenden Fensterwände nicht generiert.

- **Eingang (innen und außen)**

Ein Eingang kann maßgeblich durch zwei Polygone mit einer bestimmten Höhe repräsentiert werden. (Die Summe der Höhen entspricht der Stockwerkshöhe.) Das erste Polygon reicht vom Boden bis zur Oberkante der Tür und definiert die eigentliche Tür, das zweite liegt zwischen Tür-Oberkante und Decke und bestimmt die obere Türwand. Wie bei einem Fenster-Element besteht auch hier die Möglichkeit ein Vordach in Form eines flachen Würfels (bestehend aus fünf Polygonen) oder einen Rahmen einzufügen. Ist ein Türrahmen vorhanden, sollte das Polygon der Tür ebenfalls um einen bestimmten Betrag ( $\frac{\text{Wandbreite}}{2}$ ) in die Wand verschoben werden (vgl. Abb. 41).

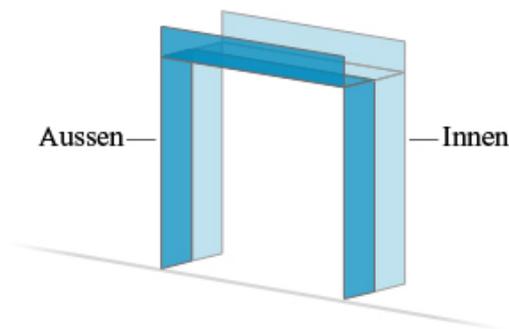


Abbildung 41: Geometrie eines Eingang-Elements.

Wie bei einer Innenwand werden alle Wandstücke einer Innen-Tür um den Betrag der Wandbreite nach innen verschoben.

- **Boden und Decke**

Die Böden und Decken von Räumen<sup>59</sup> oder Fluren werden durch die zugehörigen Teilflächen des SLG-Grundrisses definiert. Die Fläche der Decke entspricht der Bodenfläche und wird um den Betrag der Stockwerkshöhe angehoben.

Die polygonalen Boden- oder Deckenflächen können allerdings mehr als vier Ecken besitzen, woraus folgt, dass ihre Triangulierung nicht mehr trivial ist. Dieses Problem wird hier mit Hilfe des Verfahrens nach Kapitel 5.3.3 gelöst und in allen Fällen eine korrekte Triangulierung bzw. geschlossene Oberfläche erzeugt.

- **Treppenhaus**

Ein als Treppenhaus markierter Raum in einem Stockwerk beinhaltet im Normalfall einen Aufgang in die nächste Etage. Aus zeitlichen Gründen

<sup>59</sup>Wird durch den Raum ein Treppenhaus beschrieben, besitzt dieser nur im untersten Geschoss einen Boden und im obersten eine Decke.

wird hier nur eine sehr einfache Form der Podesttreppe ohne Treppenstufen eingefügt. Dafür werden maximal fünf Polygone gemäß Abb. 42 generiert.

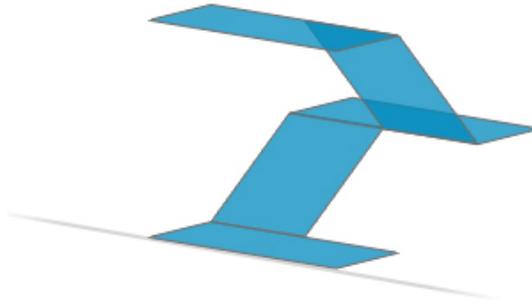


Abbildung 42: Geometrie eines Treppenhaus-Elements.

- **Dach**

Ein Dach kann eine Vielzahl von Formen haben und wird zusätzlich durch den Gebäude-Grundriss und -Typ beeinflusst (siehe Kapitel 6.5.1). In dieser Arbeit werden zwar nur Flach-, Zelt- und Satteldach behandelt, allerdings wird durch eine geeignete Parametrisierung (Dachhöhe und verschiedene Übersprünge) die Individualität und Varianz dieser Gebäudedächer gewährleistet (vgl. Abb. 43).

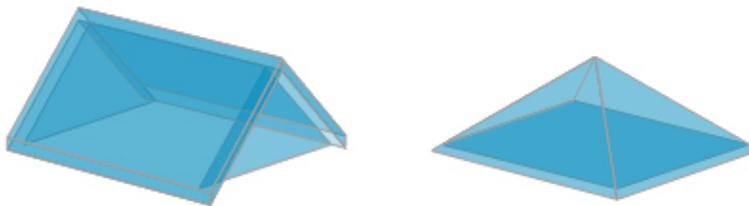


Abbildung 43: Geometrie eines Satteldach- und Zeltdach-Elements.

Das **Flachdach** ist in diesem Verfahren die einfachste geometrische Dachform. Dabei werden die Ausmaße der Gebäudefläche übernommen und das daraus entstehende Flachdach auf Gebäudehöhe angehoben. Als zusätzliches Detail kann eine kleine Mauer an den Außenseiten des Dachs erstellt werden. Für die Generierung dieses Basis-Elements können die Berechnungsschritte von Böden/Decken und Wänden wiederverwendet werden.

Das **Zeltdach** besteht aus Dreiecksflächen, die von den einzelnen Außenkanten ausgehend am Mittelpunkt<sup>60</sup> der Gebäudefläche in einer bestimmten Höhe zusammentreffen. Dabei wird aus der Gebäudefläche wieder die Grundfläche des Dachs erstellt und auf Gebäudehöhe angehoben. Es bietet sich hier an, diese Grundfläche um einen bestimmten Betrag zu vergrößern (expandieren), um einen gewissen Überhang zu den Gebäude-Fassaden herzustellen.

<sup>60</sup>Um unlogische Dachkonstruktionen zu vermeiden, wird ein Zeltdach nur verwendet, wenn der Mittelpunkt bzw. Schwerpunkt der Gebäudefläche innerhalb des Gebäudes liegt.

Das **Satteldach** kann in seiner einfachsten Form durch einen Keil mit einer bestimmten Höhe repräsentiert werden, dessen vier Eckpunkte wiederum aus der Gebäudefläche übernommen und auf Gebäudehöhe angehoben wurden. Dabei wird die obere Keil-Kante als Giebel und die dazu parallelen seitlichen Kanten als Traufen bezeichnet. Für einen höheren Detailgrad werden die Dachschrägen durch zwei weitere flache Würfel erweitert und diese um bestimmte Beträge an Trauf- und Giebel-Seiten expandiert, um Dachüberhänge entstehen zu lassen. Folglich wird ein Satteldach nur dann generiert, wenn die Gebäudefläche durch vier Eckpunkte beschrieben werden kann.

- **Gehweg**

Ein Gehweg-Element an einer Straßenseite des Hausblocks kann durch jeweils zwei Polygone aus den zugehörigen Teilflächen eines Grundstück-Grundrisses abgeleitet werden. Das erste Polygon repräsentiert den eigentlichen Gehweg und wird aus einer Teilfläche des Grundstücks generiert. Anschließend wird es auf die Höhe des Gehwegs angehoben und mit einem zweiten Polygon die Vorderseite des Gehwegs (Bordstein) zur Straße repräsentiert.

- **Hausblock- und Grundstücksfläche**

Die Gesamtfläche eines Hausblocks wird durch die innen liegenden Teilflächen seines SLG-Grundrisses und den Teilflächen seiner Grundstücke (ohne Gehwegsflächen) definiert. Die Triangulierung dieser Flächen wird wie bei den Böden und Decken mit dem Verfahren in Kapitel 5.3.3 erzeugt.

Hier soll angemerkt werden, dass sich für Gebäude und Grundstücksflächen natürlich noch weitere gestalterisch interessante Basis-Elemente, wie z.B. Schornsteine, Erker, Laternen, Zäune oder Balkone definieren lassen, diese aus zeitlichen Gründen allerdings nicht mehr behandelt werden konnten. Ein nachträgliches Einfügen dieser Aufriss-Elemente ist aber ohne großen Aufwand möglich.

### 5.3.2 Texturierung und Punkt-Normalen

Die Texturierung ist, nach der geometrischen Formgebung, das wichtigste Gestaltungselement eines 3D-Modells und wird in diesem Kapitel vor der Berechnung von Punkt-Normalen, die für eine korrekte und qualitativ hochwertige Beleuchtung notwendig sind, behandelt.

#### **Texturierung**

Durch das Texturieren können die Oberflächen des 3D-Modells mit Materialien (z.B. Farben, detaillierte Bildern, Strukturen aber auch Transparenzen) versehen werden, ohne die Geometrie weiter verfeinern zu müssen. Enthält ein Material ein Bild, werden dafür i.d.R. separat erstellte (zweidimensionale) Bilddateien verwendet. Diese Bilder bzw. einzelne Bildausschnitte werden als *Texturen* bezeichnet.

Für die Platzierung einer Textur auf einem Oberflächen-Element müssen für alle seine Punkte *Texturkoordinaten* in Form von zweidimensionalen Vektoren angegeben werden, um den Bildausschnitt auf jedem zugehörigen Polygon festzulegen<sup>61</sup>.

---

<sup>61</sup>Es können auch dreidimensionale oder ganze Listen von Texturkoordinaten pro Punkt angegeben werden, wobei diese Spezialfälle hier nicht behandelt werden.

Die obere linke Ecke eines Bildes hat dabei die Texturkoordinate  $(0, 0)$  und die untere rechte Ecke  $(1, 1)$ .

Z.B. werden die Texturkoordinaten der Eckpunkte eines viereckigen, vertikal ausgerichteten Polygons, auf das der gesamte Inhalt einer Bilddatei gelegt werden soll, wie folgt definiert:

- unterer linker Punkt:  $(0, 1)$
- unterer rechter Punkt:  $(1, 1)$
- oberer rechter Punkt:  $(1, 0)$
- oberer linker Punkt:  $(0, 0)$

Texturkoordinaten können auch größere bzw. kleinere Komponenten als 0 und 1 besitzen, wodurch die Textur bei der Darstellung in den meisten Visualisierungen neben- oder untereinander wiederholt wird — dieser Vorgang wird als *Texture-Tiling* oder *Textur-Kachelung* bezeichnet. Für ein Polygon mit den Texturkoordinaten  $(0, 2), (2, 2), (2, 0), (0, 0)$  wird das Motiv der Bilddatei also viermal (zweimal in X- und zweimal in Y-Richtung) aufgelegt.

Textur und Zuweisung der Texturkoordinaten auf den Oberflächen können zusammen als *Texturemapping* bezeichnet werden.

*Beispiel:*

Die sechs Seiten eines dreidimensionalen Würfels sollen mit sechs verschiedenen 'Zahlen'-Texturen versehen werden. Die einfachste Vorgehensweise dafür wäre, sechs einzelne Bilddateien zu erstellen und jedem Würfel-Polygon (mit den oben verwendeten Texturkoordinaten  $(0, 1), (1, 1), (1, 0), (0, 0)$ ) jeweils eine als Material zuzuweisen. Dies ist i.A. die gängigste und einfachste Möglichkeit, um ein 3D-Modell zu texturieren. Wird dieses Objekt allerdings in Echtzeit dargestellt, müssen pro Frame für jedes Polygon das zugehörige Material bzw. seine Textur in der Grafik-Hardware ausgetauscht werden. Bei dem hier verwendeten Würfel müsste dieser Wechsel bereits sechsmal durchgeführt werden.

Dieses Austauschen von Materialien (*Materialchanges*) benötigt zusätzliche Berechnungs- und Zugriffszeiten und sollte für 3D-Modelle bzw. die komplette 3D-Szene in einer Echtzeit-Anwendung so selten wie möglich verwendet werden. Zusammenfassend kann eine Optimierung für die Echtzeit-Darstellung, neben der Verwendung von Level-Of-Detail-Stufen (siehe Kapitel 5.3.4) dadurch erreicht werden, dass alle 3D-Modelle jeweils aus nur *einem* Mesh bestehen (siehe Kapitel 5.3.1) und mit nur *einem* Material texturiert werden.

Aufgrund dessen kann die Texturierung des oben beschriebenen Würfels wie folgt geändert werden:

- Es wird nur eine Bilddatei erstellt, die alle sechs Zahlenmotive enthält.
- Die Texturkoordinaten der einzelnen Polygon-Punkte werden so definiert, dass jeweils nur ein bestimmter Teilausschnitt des Gesamtbildes auf das Polygon gelegt wird.  
Sind z.B. die sechs Zahlenmotive nebeneinander in die Bilddatei eingefügt worden, werden für das Polygon mit der *zweiten* Zahl die Texturkoordinaten wie folgt gesetzt:

- unterer linker Punkt:  $(\frac{1}{6}, 1)$

- unterer rechter Punkt:  $(\frac{2}{6}, 1)$
- oberer rechter Punkt:  $(\frac{2}{6}, 0)$
- oberer linker Punkt:  $(\frac{1}{6}, 0)$

Aus den oben genannten Gründen wird in dem hier entwickelten Verfahren nur eine Textur(datei) pro Gebäude- bzw. Hausblock-Modell als Material verwendet. Daraus folgt, dass insbesondere für die 3D-Modelle von Gebäuden eine Textur aus vielen Einzelmotiven bestehen muss, um auf allen zugehörigen Basis-Elementen entsprechende Bildausschnitte anzubringen. Eine Textur, die aus mehreren Teilmotiven besteht, wird als *Texturemap* bezeichnet.

Für das hier verwendete Texturemapping werden dafür zunächst alle Bereiche der verschiedenen Teilmotive innerhalb der Texturemap definiert. Anschließend können die Texturkoordinaten von Polygonen der Basis-Elemente durch folgende Mapping-Arten generiert werden:

- **Einfaches Mapping**

Beim einfachen Mapping wird das komplette Teilmotiv auf das Polygon gemappt. Dabei wird es folglich durch die Ausmaße des Polygons verzerrt. Wurde z.B. das Motiv einer Haustür in dem Bereich  $(0.2, 0.2)$  und  $(0.4, 0.4)$  definiert, werden dem Polygon der Tür die vier Texturkoordinaten  $(0.2, 0.4)$ ,  $(0.4, 0.4)$ ,  $(0.4, 0.2)$  und  $(0.2, 0.2)$  zugewiesen

- **Gekacheltes Mapping**

Beim gekachelten Mapping können die Teilmotive *teilweise* oder *mehrfach* auf dem Polygon angebracht werden. Insbesondere bei Wandtexturen bietet es sich an, diese ohne horizontale Verzerrung und fortlaufend auf die einzelnen Außenwänden eines Stockwerks zu mappen. Dafür muss zusätzlich die *reale* Länge der Teiltextrur angegeben werden, um die X-Komponenten der Texturkoordinaten wie folgt zu bestimmen ( $p0 - p3$  sind dabei linke untere, rechte untere, rechte obere und linke obere Ecke des Polygons):

$$\begin{aligned} x_{p0,p3_{polygon}} &= x_{p1,p2_{polygonPrev}} \\ x_{p1,p2_{polygon}} &= x_{p0,p3_{polygon}} + \frac{length_{polygon}}{length_{texture}} \end{aligned} \quad (35)$$

Z.B. wird dadurch eine  $5m$  lange Wandtextur  $1\frac{1}{2}$ -mal auf ein  $7.5m$  langes Wand-Element gemappt. Zusätzlich kann an einem direkt folgenden Wand-Element die Textur ohne Unterbrechung fortgeführt werden.

Die hier verwendete Texturemap für ein Gebäude wird in Abb. 44 veranschaulicht<sup>62</sup>. Allein mit dieser relativ kleinen und einfachen Texturemap werden alle Oberflächen eines Gebäudes vollständig in angemessener Qualität und Vielfalt texturiert (siehe Kapitel 6.5.1 für eine weitere Beschreibung).

### **Punkt-Normalen**

Eine Punkt-Normale ist ein Richtungsvektor, der von der Grafik-API für die Beleuchtung, aber auch für dynamische Materialeigenschaften, von Polygonen verwendet wird und somit an allen Mesh-Punkten hinsichtlich einer besseren visuellen Qualität vorhanden sein soll.

<sup>62</sup>Die Texturemap für einen Hausblock besteht hierbei aus nur zwei rechteckigen Motiven.

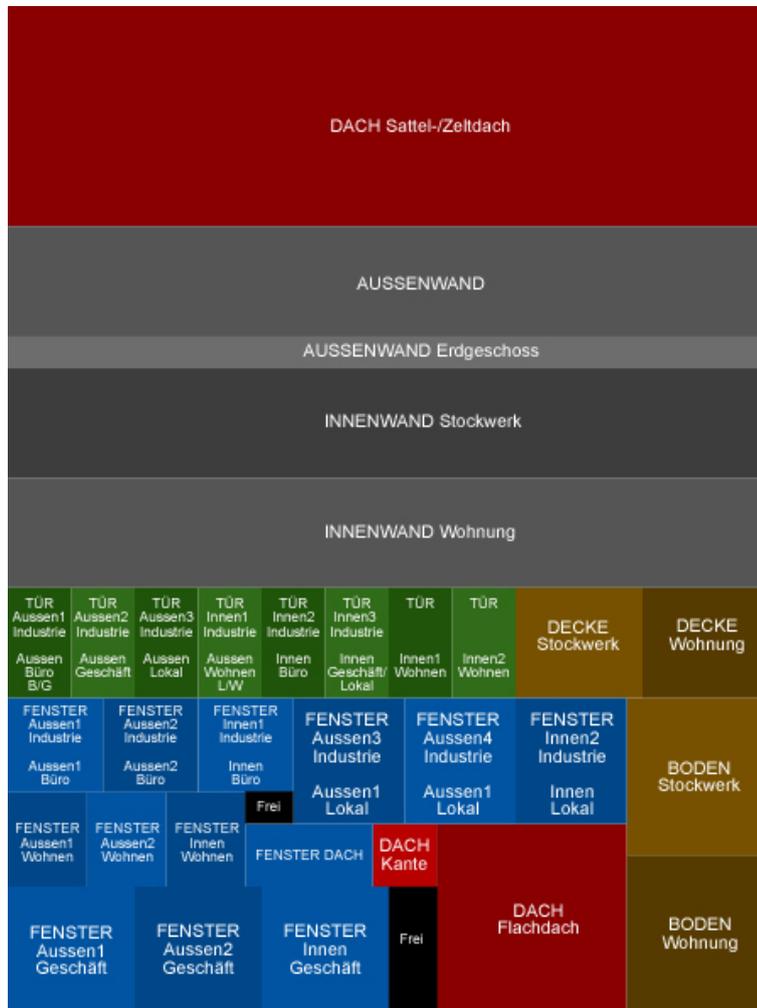


Abbildung 44: Texturemap eines Gebäudes.

Eine Wand soll dabei z.B. an den Stellen heller schattiert werden, auf die mehr Licht fällt und umgekehrt. Die einfachste Art, diese Helligkeitsabstufungen zu ermöglichen, ist die Punkt-Normalen durch senkrecht zur Polygonfläche stehende normierte Richtungsvektoren zu repräsentieren (vgl. Abb. 45).

Sollen die Helligkeitsübergänge an angrenzenden nicht parallelen Polygonen dabei ineinander verlaufen bzw. interpoliert werden, muss an jedem Punkt die Ausrichtung der anliegenden Polygone bekannt sein. Diese Informationen können hier durch die zugehörigen SLG-Kanten bestimmt und damit die vier Punkt-Normalen eines Polygons wie folgt generiert werden:

$$\begin{aligned}
 \vec{n}_{p0,p3_{polygon}} &= \|\vec{v}_{edge} - \vec{v}_{edgePrev}\| \\
 \vec{n}_{p1,p2_{polygon}} &= \|\vec{v}_{edgeNext} - \vec{v}_{edge}\|
 \end{aligned}
 \tag{36}$$

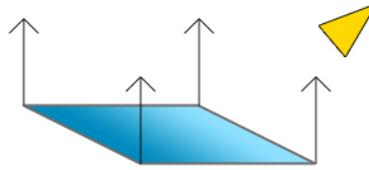


Abbildung 45: Beleuchtetes Polygon (mit Gouraud-Shading).

Durch die Verwendung der SLG-Grundrisse für alle Basis-Elemente, und dem damit verbundenen Zugriff auf sämtliche Nachbarschaftsinformationen der verwendeten Kanten, wird die Umsetzung der hier beschriebenen Algorithmen zur Texturierung und Berechnung von Punkt-Normalen entscheidend vereinfacht.

### 5.3.3 Triangulierung

Durch die Triangulierung wird aus den Punkten eines Meshs seine darstellbare Oberfläche in Form von Dreiecken gebildet. Dabei müssen für jedes Dreieck jeweils drei Punkte bzw. deren Indizes angegeben werden, um eine *einseitige* Fläche zu definieren. Die Richtung der Dreiecks-Normalen, also von welcher Seite die Fläche sichtbar ist, wird dabei durch die Reihenfolge der Indizes bestimmt.

Wird z.B. ein in der Ebene liegendes Dreieck mit Indizes *gegen den Uhrzeigersinn* definiert, zeigt seine Normale in Richtung der positiven Y-Achse und die Fläche kann von oben betrachtet werden. (Entsprechend kann ein *mit dem Uhrzeigersinn* orientiertes Dreieck nur von unten gesehen werden.)

Die Datenbasis einer Triangulierung ist immer eine beliebige Menge von Punkten. Für das hier entwickelte Verfahren wird angenommen, dass diese Punkte die Ecken eines Polygons definieren — es liegen also keine Punkte im Inneren der Fläche. Durch die Verwendung der SLG-Grundrisse können diese Ausgangs-Daten um eine zugehörige Liste mit Kanten zwischen den Punkten bzw. Knoten erweitert werden, falls damit eine vorhandene (Teil-)Fläche im Graphen beschrieben wird (dies ist bei der Triangulierung der entsprechenden Basis-Elemente immer der Fall).

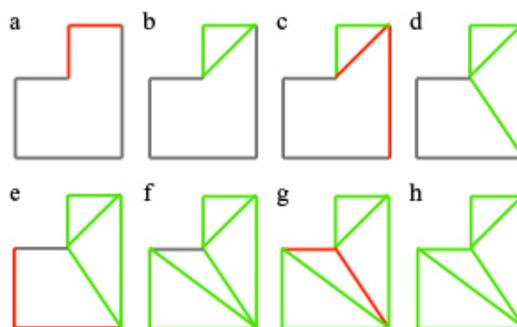


Abbildung 46: Triangulierung eines konkaven Polygons.

Mit diesen Voraussetzungen wird eine Fläche wie folgt trianguliert (vgl. Abb. 46):

1. Die Kantenliste wird *gegen den Uhrzeigersinn* geordnet, falls die entstehende Oberfläche von oben sichtbar sein soll und *mit dem Uhrzeigersinn* im umgekehrten Fall (z.B. bei der Fläche eines Decken-Elements). Die Liste wird dabei gemäß Kapitel 4.6.1 geordnet.
2. Ein beliebiges Element der Kantenliste wird ausgewählt (*edge*) und die nächste Kante bestimmt (*edgeNext*, vgl. Abb. 46 a). Enthält die Liste weniger als drei Elemente (siehe Schritt 4), ist die Triangulierung beendet (vgl. Abb. 46 h).
3. Ist der Winkel zwischen beiden Kanten überstumpf, wird der Algorithmus mit einer anderen Kante bei Schritt 2 erneut gestartet.  
Ist der Winkel nicht überstumpf, wird aus den drei Punkten der beiden Kanten ein Dreieck gebildet und in das Mesh eingetragen (vgl. Abb. 46 b). Wichtig ist hier, dass in diesem neuen Dreieck keine weiteren Punkte des Polygons liegen dürfen, um Überschneidungen zu vermeiden. Da die Kanten bereits gegen den Uhrzeigersinn geordnet sind, wird das Dreieck in folgender Reihenfolge definiert:

$$\triangle(\vec{p}_{nodeStart_{edge}}, \vec{p}_{nodeEnd_{edge}}, \vec{p}_{nodeEnd_{edgeNext}}) \quad (37)$$

4. *edge* und *edgeNext* werden aus der Kantenliste entfernt und die durch das Dreieck neu entstehende Verbindung zwischen *nodeStart<sub>edge</sub>* und *nodeEnd<sub>edgeNext</sub>* als Kante in die Liste eingefügt.  
Nun wird der Algorithmus bei Schritt 2 erneut gestartet.

Der hier entwickelte Triangulierung-Algorithmus ist einfach, sehr robust und liefert die geforderten Ergebnisse mit geringer Berechnungszeit. Die Umsetzung dieses Teilverfahrens wird wiederum durch die Verwendung des SLG und dessen Funktionalitäten entscheidend vereinfacht.

### 5.3.4 Generierung der 3D-Modelle und Level-of-Detail

Nachdem die Verfahren für Geometrie, Texturierung, korrekte Beleuchtung (Punkt-Normalen) und Triangulierung für alle Basis-Elemente eines Gebäudes und Hausblocks entwickelt wurden, können nun die Meshs der eigentlichen 3D-Modelle aus diesen Bausteinen zusammengesetzt werden.

Wie bereits in Kapitel 5 erwähnt wurde, besteht ein 3D-Modell zunächst aus einer beliebigen Anzahl<sup>63</sup> von LOD-Stufen ( $\geq 1$ ). Jede LOD-Stufe besitzt immer einen Distanz-Betrag und mindestens ein Mesh. Der Distanz-Betrag bestimmt dabei die Entfernung (zwischen Betrachter und 3D-Modell), ab der die zugehörige(n) Mesh(s) angezeigt werden<sup>64</sup>. Dabei ist es sinnvoll, detailreiche und damit aufwändig darzustellende Meshs nur in der Nähe des Betrachters einzublenden

<sup>63</sup>Die maximale Anzahl von Level-of-Detail-Stufen wird durch den verfügbaren Hardware-Speicher begrenzt. Mehr als fünf oder sechs LOD-Stufen werden i.A. nicht verwendet, da bei einer höheren Anzahl die Abstufungen der Detailtiefe vom Betrachter nicht mehr wahrgenommen werden und die weitere Entlastung der Computer-Hardware insbesondere durch den zusätzlichen Speicherbedarf der LOD-Meshs, nur noch minimal ist.

<sup>64</sup>Es werden immer nur die Meshs einer einzelnen LOD-Stufe angezeigt und alle restlichen ausgeblendet.

(z.B. bis 5m) und für einfache Meshs größere Entfernungen (z.B. 100m) anzugeben, um die Echtzeitdarstellung vieler Gebäude bei nahezu gleich bleibendem visuellen Eindruck entscheidend zu verbessern.

Für das hier entwickelte Verfahren zur prozeduralen Generierung von 3D-Modellen können für jedes Modell beliebig viele LOD-Meshs erstellt werden.

Im Folgenden wird beschrieben, welche LOD-Parameter für die Meshs der Gebäude- und Hausblock-Modelle definiert sind und wie diese Meshs aus den jeweiligen Basis-Elementen zusammengesetzt werden. Nach Beendigung dieses Teilverfahrens ist die Aufriss-Generierung abgeschlossen und die dabei entstehenden 3D-Modelle können durch die Grafik-API dargestellt werden.

### Gebäude-Modell

Für ein Gebäude-Modell müssen für jede LOD-Stufe die folgenden **LOD-Parameter** definiert werden:

- Stufennummer<sup>65</sup>
- Entfernung
- Textur<sup>66</sup>
- Transparenz [ *ja* | *nein* ]
- Innenräume [ *vorhanden* | *nicht vorhanden* ]
  - Fensterrahmen innen [ *vorhanden* | *nicht vorhanden* ]
  - Türrahmen innen [ *vorhanden* | *nicht vorhanden* ]
- Außenwände (Fassaden) [ *für jedes Stockwerk* | *für ganze Gebäudeseite* ]
  - Fenster außen [ *vorhanden* | *nicht vorhanden* ]
  - Eingänge außen [ *vorhanden* | *nicht vorhanden* ]
  - Fensterrahmen außen [ *vorhanden* | *nicht vorhanden* ]
  - Türrahmen außen [ *vorhanden* | *nicht vorhanden* ]
- Dach [ *einfach* | *komplex* ]

Die **Konstruktion eines Meshs** des Gebäude-Modells lässt sich sinnvoll in folgende Teilverfahren zerlegen:

- Außenwänden mit Türen und Fenstern (pro Stockwerk)
- Innenwände mit Türen und Fenstern (pro Stockwerk)
- Böden und Decken (pro Stockwerk)
- Treppenhaus (pro Stockwerk)
- Dach (für Obergeschoss)

---

<sup>65</sup>I.A. wird der LOD-Stufe mit der kleinsten Entfernung (= 0m) bzw. den höchsten Details die Nummer 0 zugewiesen. Folglich hat die entfernteste Stufe die höchste Nummer.

<sup>66</sup>Die Textur eines Modells kann wie das Mesh, je nach Detailtiefe ausgetauscht werden. Z.B. könnten weiter entfernte Meshs eine kleinere Texturdatei verwenden, um die Hardware-Ressourcen zusätzlich zu entlasten.

Alle fünf Einzelschritte werden für jede LOD-Stufe mit entsprechenden Parametern iterativ ausgeführt. Dabei wird jeweils ein Mesh generiert und dem jeweiligen Gebäude-Modell zugewiesen.

- **Außenwänden mit Türen und Fenstern (pro Stockwerk)**

Für die Außenwände bzw. Fassaden werden zunächst alle Außen-Kanten des Gebäude-Grundrisses bzw. seiner Teilflächen benötigt. Für die Erfassung aller Außenwände mit sämtlichen Fenstern und Türen, müssen die außen liegenden Kanten in den Teilflächen der Stockwerke und Wohnungen durchsucht werden.

Nachdem alle Kanten gefunden wurden, ist die Außenwand durch einen geschlossenen Kantenzug definiert, der hier *mit dem Uhrzeigersinn* geordnet wird<sup>67</sup>, damit die später entstehenden Polygone *nach außen* sichtbar sind. Die Kantenliste wird danach durchgegangen und jede Kante anhand ihrer *Lage im geographischen Zusammenhang* durch ein Basis-Element ersetzt und in des Gebäude-Mesh der aktuellen LOD-Stufe eingefügt.

Ein **Wand-Element** wird für folgende Werte generiert:

- *Außen liegend zur Straße*
- *Außen liegend zu einem Zwischenraum*
- *Außen liegend zu einer Nachbarfläche*

Liegen mindestens zwei dieser Kanten direkt nebeneinander und beinhalten keine Ecke, werden sie zu einem Wandstück bzw. einer Fläche zusammengefasst. Diese Vereinfachung reduziert zusätzlich die Anzahl der entstehenden Polygone.

Bei der höchsten LOD-Stufe, also bei dem Mesh mit den wenigsten Details, ist es oft der Fall, dass die Fassaden des Gebäudes keine Fenster und Türen enthalten sollen und damit die Außenwände einer Gebäudeseite durch nur ein Polygon repräsentiert werden können. Dabei werden nur für das Erdgeschoss die Wände mit Gebäudehöhe erstellt und das Verfahren für die Außenwände anschließend beendet. Bei dem so generierten Mesh kann damit die Anzahl der Polygone auf ein Minimum beschränkt werden.

Ein **Fenster-Element** wird für folgende Werte generiert:

- *Außen liegendes Fenster zur Straße*
- *Außen liegendes Fenster zu einem Zwischenraum*

Durch die LOD-Parameter wird dabei bestimmt, ob Fensterglas und/ oder -rahmen erstellt werden.

Ein **Eingangs-Element** wird für folgende Werte generiert:

- *Außen liegender Haupteingang*

---

<sup>67</sup>Bei dem Ordnen dieser Kantenliste muss beachtet werden, dass die Übergangs-Kanten von Stockwerks- und Wohnungs-Teilflächen durch die Positionen ihrer Knoten verglichen werden müssen, da diese Teilflächen zu unterschiedlichen SLG-Strukturen gehören.

- *Außen liegender Eingang*

Durch die LOD-Parameter wird dabei bestimmt, ob eine Tür mit oder ohne Rahmen erstellt wird.

- **Innenwände mit Türen und Fenstern (pro Stockwerk)**

Die Innenwände eines Stockwerks werden aus den Hausflur-Kanten (innen liegende Teilflächen des Stockwerk-Grundrisses) und den Kanten der Flure und Räume aller Wohnungs-Grundrisse generiert.

Alle dabei verwendeten Kantenlisten werden *gegen den Uhrzeigersinn* geordnet, damit die entstehenden Polygone *von innen* sichtbar sind. Dabei bestimmt wieder die *Lage im geographischen Zusammenhang* der Kanten, welche Basis-Elemente erstellt werden (Wand, Fenster oder Tür).

Alle dabei entstehende Wandflächen werden um den Betrag der Wandbreite, falls sie die Rückseite einer Außenwand repräsentieren und sonst um den Wert  $\frac{\text{Wandbreite}}{2}$  ins Innere der jeweiligen Teilfläche versetzt (vgl. Abb. 47).

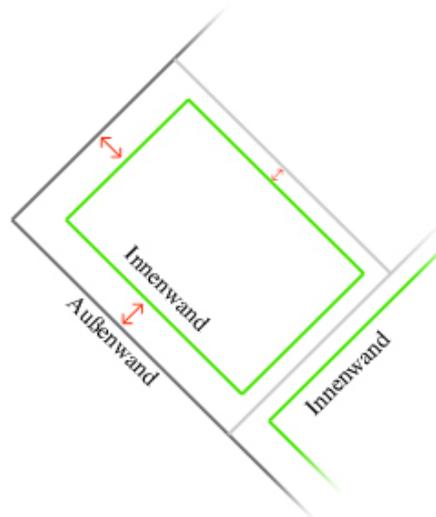


Abbildung 47: Platzierung der Innenwände.

Ein **Wand-Element** wird für folgende Werte generiert:

- *Außen liegend zur Straße*
- *Außen liegend zu einem Zwischenraum*
- *Außen liegend zu einer Nachbarfläche*
- *Innen liegend zu einem Hausflur*
- *Innen liegend zu einer Nachbarwohnung*
- *Innen liegend zu einem Wohnungsflur*
- *Innen liegend zu einem Nachbarraum*

Wie bei den Außenwänden können nebeneinander liegende Wand-Elemente zusammengefasst werden.

Ein **Fenster-Element** wird für folgende Werte generiert:

- *Außen liegendes Fenster zur Straße*
- *Außen liegendes Fenster zu einem Zwischenraum*

Dabei bestimmt ein LOD-Parameter, ob ein Fensterrahmen berechnet wird.

Ein **Eingangs-Element** wird für folgende Werte generiert:

- *Innen liegender Haupteingang*
- *Innen liegender Eingang*
- *Innen liegender Treppeneingang*

Dabei bestimmt ein LOD-Parameter, ob ein Türrahmen eingefügt wird.

- **Böden und Decken (pro Stockwerk)**

Die Böden und Decken eines Stockwerks werden aus den Eckpunkten der Teilfläche des Hausflurs und den Gesamtflächen der Wohnungen generiert. Falls eine Wohnung ein Treppenhaus beinhaltet, das keinen Boden bzw. keine Decke besitzt, werden statt der Gesamtfläche alle Teilflächen der restlichen Räume verwendet.

- **Treppenhaus (pro Stockwerk)**

Ein Treppenhaus-Element wird bei mehr als zwei Stockwerken jeweils in den vorher definierten Raum (Teilfläche) einer Wohnung eingesetzt.

- **Dach (für Obergeschoss)**

Das Dach-Element wird nur einmal pro Gebäude, in der Iteration mit dem obersten Stockwerk generiert. Hier werden die *gegen den Uhrzeigersinn* orientierten Eckpunkte mit ihren Verbindungskanten übergeben.

### **Hausblock-Modell**

Da die Basis-Elemente für einen Hausblock hier nur auf eine Art generiert werden können, wird dementsprechend eine LOD-Stufe (*Stufennummer = 0, Entfernung = 0, ...*) mit einem Mesh pro Hausblock-Modell erstellt.

Die dabei generierten Basis-Elemente Gehweg und Grundstücksfläche ergeben sich dabei direkt aus den jeweiligen SLG-Grundrissen.

- **Gehweg**

Für die Oberflächen der Gehwege werden die Eckpunkte und Kanten der entsprechenden Teilflächen der Grundstücke verwendet.

- **Grundstücksfläche**

Für die geschlossene Oberfläche des Hausblocks werden die einzelnen Bebauungs-Teilflächen der Grundstücke, sowie alle innen liegenden Teilflächen des Hausblock-Grundrisses verwendet.

## **5.4 Zwischenfazit**

Mit dem hier entwickelten Verfahren können auf Basis der SLG-Grundrisse alle 3D-Modelle des Stadtaufnisses erstellt werden. Dabei werden die folgenden, hier benötigten Modellierungsprozesse durch prozedurale Teilverfahren behandelt:

- Platzierung der Mesh-Vertices (Geometrie)

- Indizierte Triangulierung (Bildung von Oberflächen bzw. Dreiecken)
- Texturierung mit Texturemaps (Oberflächengestaltung bzw. Materialien)
- Punkt-Normalen (Beleuchtung)
- Level-of-Detail-Stufen

Eine weitere Bearbeitung ist somit nicht mehr notwendig und die Modelle können direkt durch die Grafik-API dargestellt werden (vgl. Abb. 48).

Zusätzlich sind die entstehenden Modelle für die Echtzeit-Darstellung optimiert und können durch eine Vielzahl von Parametern bzgl. Formgebung und Detailtiefe gesteuert werden.

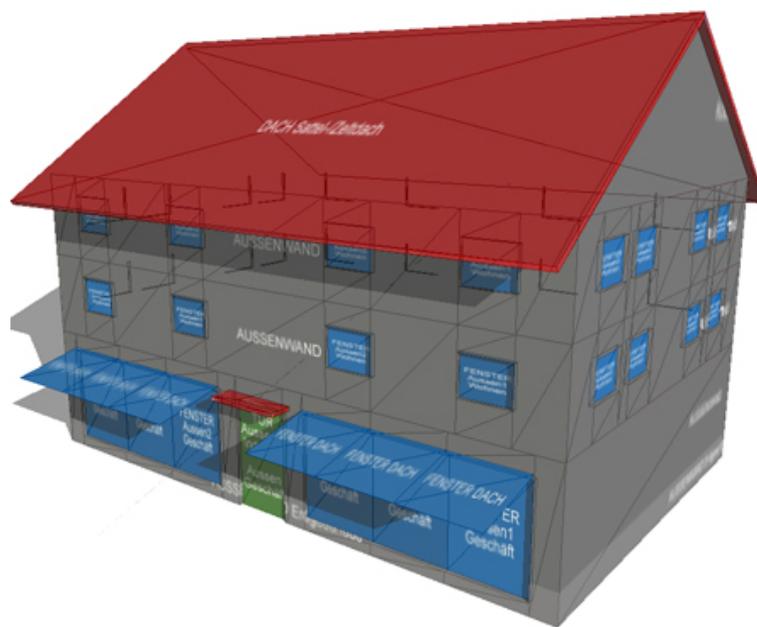


Abbildung 48: Einfaches Gebäudemodell, erstellt mit CITYLIFE (siehe Kapitel 7).

Die Auswahl der Aufriss-Parameter wird im folgenden Teil dieser Arbeit (Kapitel 6) behandelt und so an stadtgeographische und städtebaulichen Gesetzmäßigkeiten angepasst, dass realistisch wirkende 3D-Modelle generiert werden.

## 6 Stadtojekte

In dieser Arbeit wird der Agglomerationsraum einer Stadt, angelehnt an die morphogenetischen und städtebaulichen Elemente einer Stadt gemäß Kapitel 3.4, durch die folgende Objekthierarchie repräsentiert:

1. **Stadt**
2. **Stadtteil**
3. **Hausblock**
4. **Grundstück**
5. **Gebäude**
6. **Stockwerk**
7. **Wohnung**<sup>68</sup>

Diese Hierarchie ist u.a. so angelehnt, dass eine Stufe die horizontalen, räumlichen Ausmaße der nächst Höheren nicht überschreitet und somit immer als eine ihrer Teilfläche definiert werden kann. Jeder Stufe wird daher ein SLG für ihre Grundriss-Generierung zugewiesen (siehe Kapitel 4.2) und für ihre vertikale Ausdehnung bzw. die Erstellung der 3D-Modelle das Aufriss-Verfahren aus Kapitel 5 verwendet (ab der Stufe Hausblock).

Des weiteren beschreibt jede hier verwendete Stufe hinsichtlich ihrer Parametrisierung eine eigenständige stadtgeographische Einheit, die jeweils in einem der nachfolgenden Unterkapiteln sinnvoll definiert wird und von den anderen Stufen abgegrenzt werden kann.

Das gesamte Stadtgebiet kann also mit Hilfe dieser Objekt-Hierarchie vollständig strukturiert und gegliedert werden. Dabei enthält ein Objekt im Normalfall immer nur die Objekte der nächst unteren Stufe. Bei den Stadtojekten Stadt, Stadtteil, Hausblock, und Gebäude sind aber auch Hierarchiesprünge bzw. verschiedene Kombinationen möglich. Z.B. kann eine Stadt aus Hausblöcken oder ein Stadtteil aus einer Menge von Gebäuden definiert werden. Im stadtgeographischen Zusammenhang ist dies zwar weniger sinnvoll, bietet aber bei der späteren Umsetzung und Verwendung des Verfahrens zusätzliche Gestaltungsmöglichkeiten.

Jedes Stadtojekt ist neben seiner horizontalen und ggf. vertikalen Gestalt durch eine Vielzahl von Parametern definiert, die so differenziert angelegt sind, das eine *unbegrenzte* Anzahl von verschiedenen dreidimensionalen Stadt- und Gebäudemodellen erzeugt werden kann. D.h., dass sowohl Grundriss- aber auch Aufriss-Generierung nahezu vollständig durch die nachfolgenden Parameter gesteuert werden.

Alle Parameter eines Stadtojekts werden in die folgenden Kategorien eingeteilt:

- **Grundriss-Parameter**

Diese Parameter steuern den SLG eines Objekts und somit die Flächenunterteilung. Die Menge und Struktur der Grundriss-Parameter ist

---

<sup>68</sup>Die **Räume** einer Wohnung ergeben sich durch die Teilflächen ihres SLG-Grundrisses und werden aus zeitlichen Gründen nicht als eigenständige Objekte definiert.

für alle Objekte gleich und entspricht den SLG-Parametern (siehe Kapitel 4.4).

- **Objektspezifische Parameter**

Für einige Objekte werden individuelle Parameter definiert, die ihre Besonderheiten behandeln und die Grundriss- und Aufriss-Verfahren zusätzlich beeinflussen. Die Erweiterung des SLG-Grundrisses und somit die Grundlage der Aufriss-Generierung wird dabei hauptsächlich von den spezifischen Parametern der Objekte Gebäude und Stockwerk gesteuert.

Wie diese Parameter definiert sind und bestimmt werden können, wird in den folgenden Unterkapiteln der entsprechenden Objekte beschrieben.

- **Allgemeine stadtgeographische Parameter**

Diese Parameter werden für jedes Objekt definiert und beschreiben die allgemeinen stadtgeographischen und städtebaulichen Gegebenheiten der zugehörigen Hierarchiestufe. Das Besondere an diesen Variablen ist, dass sie eine Abstraktionskategorie für die Grundriss- und objektspezifischen Parameter darstellen. Folglich können aus diesen Parametern alle anderen Werte hergeleitet und somit die Steuerung des Gesamtverfahrens sinnvoll vereinfacht werden.

Die Parameter werden aus den hier beschriebenen stadtgeographischen und städtebaulichen Gesetzmäßigkeiten gemäß Kapitel 3.2 und 3.3 abgeleitet und dabei so angepasst, dass sie für die Gestaltung von Grund- und Aufriss verwendet werden können:

- **Flächennutzungstypen**

Nach der funktionalen Stadtgeographie können die einzelnen Teile eines städtischen Gebiets nach ihren vorherrschenden Flächennutzungen strukturiert und gegliedert werden (siehe Kapitel 3.3). Die große Menge der dafür vorgesehenen Flächennutzungstypen wird hier so vereinfacht und reduziert, dass nur diejenigen erhalten bleiben, die hinsichtlich Grundriss- und Aufriss-Gestaltung klar voneinander abgegrenzt und mit dem hier entwickelten Verfahren umgesetzt werden können. Damit ergeben sich folgende Flächennutzungstypen:

- \* *Industrie*
- \* *Wohnen*
- \* *Büro*
- \* *Einzelhandel*
- \* *Gastronomie*
- \* *Freiflächen (Grünflächen)*

- **Gebietslage [Stadtkern | Stadtmantel | Stadtrand]<sup>69</sup>**

Die Gebietslage bestimmt die übergeordnete Lage der Objekte im Stadtgebiet und kann als zusammenfassender Wert der *Bevölkerungs- und Bebauungsdichte* angesehen werden (siehe Kapitel 3). In Verbindung mit den Flächennutzungstypen kann dieser Parameter für die Bestimmung von Gebäudehöhen, Art der Bauweise und die

---

<sup>69</sup>In diesem Verfahren liegt der Stadtkern im Zentrum und der Stadtrand am Außenrand eines Stadtgebiets. Der Stadtmantel liegt zwischen Stadtkern und Stadtrand.

Flächengröße der Unterobjekte verwendet werden. Z.B. sind die Wohn- und Bürogebäude im *Stadtkern* im Durchschnitt höher und in ihrer Grundfläche größer als am *Stadtrand*. Ein Industriegebiet hat dagegen in jeder Lage immer eine ähnliche Gestalt.

– **Kulturraum** [*Europäisch*]

Der Kulturraum eines städtischen Gebiets beeinflusst maßgeblich die Form von Grund- und Aufriss aller Stadtobjekte. Aus zeitlichen Gründen wird in diesem Verfahren nur die allgemeine europäische Stadt umgesetzt. Dieser Parameter hat hier folglich immer den Wert *Europäisch* und wird hier hauptsächlich für weiterführende Arbeiten eingeführt.

Die *allgemeinen stadtgeographischen Parameter* sind für die hier eingeführte Objekt-Hierarchie die wichtigste Schnittstelle zwischen den verschiedenen Stufen. Dabei wurde die Anzahl der Parameter selbst und auch ihrer Werte möglichst gering gehalten, um so eine intuitive und einfache Steuerung des notwendigerweise komplexen Gesamtverfahrens zu ermöglichen.

In den folgenden Unterkapiteln werden die einzelnen Stadtobjekte mit ihren gegenseitigen Beziehungen, sowie den jeweiligen Parametersätzen und deren Berechnungsvorschriften beschrieben.

Dabei ist zu beachten, dass die Parameter eines Objekts bei einer prozeduralen Hierarchieerstellung von ihren Oberobjekten beeinflusst werden. Zusätzlich wird auf jeder Stufe eine Menge von vordefinierten Parameterwerten benötigt, um den automatischen Ablauf der Grundriss- und Aufriss-Verfahren ohne weitere Eingaben durchführen zu können.

Alle im Folgenden zugewiesenen Parameterwerte, für die keine konkreten Vorgaben in den hier vorhandenen Quellen existieren<sup>70</sup>, wurden so ausgewählt und in einem sukzessiven Prozess angepasst, dass realistisch wirkende Ergebnisse entstehen. Alle Wertzuweisungen wurden dabei in Anlehnung an die stadtgeographischen und städtebaulichen Gesetzmäßigkeiten aus Kapitel 3 durchgeführt.

## 6.1 Stadt

Das Objekt *Stadt* wird in dieser Arbeit als höchste bzw. erste Hierarchiestufe verwendet. Die Ausmaße der Gesamtfläche sowie sämtliche nicht prozedural zu berechnende Parameter einer Stadt müssen somit immer manuell vorgegeben werden.

Bei der vereinfachten Grundriss-Unterteilung einer Stadt entstehen an ihrem Außenrand Objekte vom Typ *Stadtteil*, deren Ausmaße durch die jeweiligen SLG-Teilflächen bestimmt werden. Zusätzlich werden Straßen zwischen den Stadtteilen eingefügt, welche hier die Hauptverkehrswege eines Stadtgebiets darstellen<sup>71</sup>.

---

<sup>70</sup>Konkrete Werte existieren z.B. für die Stadtgrößenklassen in Form von Einwohnerzahlen oder für GFZ und GRZ (siehe Kapitel3).

<sup>71</sup>Die Hauptverkehrswege werden in diesem Verfahren aufgrund der vereinfachten Grundriss-Generierung von Städten immer zwischen die Stadtteile gelegt. Die Generierung

Im den folgenden Unterkapiteln werden alle Parameter dieses Objekts definiert und eine Menge von vordefinierten Werten zugewiesen. Zusätzlich wird die Berechnung der *stadtgeographischen Parameter* der Unterobjekte beschrieben.

### 6.1.1 Parameter und vordefinierte Werte einer Stadt

Für ein *Stadt*-Objekt werden die folgenden Parameter und vordefinierten Werte verwendet.

#### Objektspezifische Parameter

Für eine Stadt werden die folgenden spezifischen Parameter nach Kapitel 3.1 abgeleitet:

- **Stadtgröße** [*Großstadt* | *Mittelstadt* | *Kleinstadt*]  
Die Größe einer Stadt beeinflusst in diesem Verfahren maßgeblich die maximale Gebäudehöhe bzw. Geschossanzahl.
- **Örtliches Zentrum**  
Das örtliche Zentrum einer Stadt, bzw. der Bezugs- und Mittelpunkt des zentral gelegenen Stadtkerns, wird in diesem Verfahren auf den Mittelpunkt des durch die Stadtfläche beschriebenen Polygons vereinfacht. Dies ist i.A. ein geeigneter Wert und kann über den SLG berechnet werden (siehe Kapitel 4.7). Dazu besteht die Möglichkeit, diesen Positionsvektor auch manuell anzugeben, um zusätzliche Varianten bei der Gestaltung zu ermöglichen. (Das örtliche Zentrum einer Stadt wird ab Hausblock-Ebene für die prozedurale Berechnung der jeweiligen stadtgeographischen Parameter verwendet.)

Die Werte dieser Parameter werden wie in Tabelle 3 beispielhaft zugewiesen.

Parametername	vordefinierter Wert
Stadtgröße	Mittelstadt
Kulturraum	Europäisch
Örtliches Zentrum	wird berechnet

Tabelle 3: Objektspezifische Parameter eines Stadt-Objekts.

#### Grundriss-Parameter

Die Teilflächen einer Stadt sollen gleichmäßig und mit einer bestimmten Fläche erstellt werden, wobei ihre Form im Inneren der Fläche auch konkav sein kann. Dazu gibt es Zwischenräume für Straßen, die als Hauptstraßen eine relativ große Breite besitzen. Alle weiteren Grundriss-Parameter werden beispielhaft zugewiesen. (vgl. Tab. 4).

#### Allgemeine stadtgeographische Parameter

Das *Stadt*-Objekt soll hier eine beispielhafte Flächennutzung, allerdings ohne er-  
von Straßennetzen nach anderen stadtgeographischen Gegebenheiten wird zunächst somit vernachlässigt.

Parametername	vordefinierter Wert
Anzahl der Begrenzungs-Knoten	wird berechnet
Typ der Berechnungsvorschrift für die Anzahl der Begrenzungs-Knoten	nach Umfang
Durchschnittlicher Flächeninhalt einer Teilfläche	25000m <sup>2</sup>
Seitenverhältnis der Teilflächen	1 : 2
Skalierungsfaktor und Exponent für Wachstumsfunktion	0; 0
Minimale Anzahl von Begrenzungs-Knoten	1
Positions-Abweichung der Begrenzungs-Knoten	0.1
Typ der Zwischenräume	Straßen
Breite der Zwischenräume	20m
Start-Knoten für Zwischenräume	Straßenseite
Kürzeste und längste Kante	wird berechnet
Maximale Anzahl von überstumpfen Winkeln für Pfade am Außenrand, Außenrand mit Ecke und Außenrand mit konkaver Ecke	1; 1; 1

Tabelle 4: Grundriss-Parameter eines Stadt-Objekts.

zwungene Freiflächen, besitzen. Die Gebietslage bleibt bei einer Stadt immer undefiniert (vgl. Tab. 5).

Parametername	vordefinierter Wert
Flächennutzung Industrie	15%
Flächennutzung Wohnen	40%
Flächennutzung Büro	25%
Flächennutzung Einzelhandel	15%
Flächennutzung Gastronomie	5%
Flächennutzung Freiflächen (Grünflächen)	0%
Gebietslage	Undefiniert
Kulturraum	Europäisch

Tabelle 5: Allgemeine stadtgeographische Parameter eines Stadt-Objekts.

### 6.1.2 Allgemeine Stadtgeographische Parameter von Unterobjekten

Alle Unterobjekte einer Stadt, die aus ihren SLG-Teilflächen generiert wurden, werden bei einer automatischen Hierarchiebildung sinnvollerweise durch dieses Objekt beeinflusst. Dafür bietet es sich an, nur die Werte der stadtgeographischen Parameter eines Unterobjekts zu bestimmen und an dieses zu übergeben. Diese Werte können dann im Unterobjekt für die Herleitung der restlichen Parameter verwendet werden und stellen somit eine geeignete, vor allem einfache Schnittstelle dar.

Die automatische Bestimmung der allgemeinen stadtgeographischen Parameter eines Stadtteils wird negativ durch die vereinfachte Grundriss-Erstellung einer Stadt beeinflusst. Dadurch werden hier ggf. Werte zugewiesen, die im stadtgeographischen Zusammenhang weniger sinnvoll sind. Diese Problematik lässt

sich allerdings durch eine manuelle Stadtteil-Definition bzw. Werte-Anpassung beheben.

Die allgemeinen stadtgeographischen Parameter jedes einzelnen Unterobjekts werden wie folgt bestimmt.

- **Gebietslage**

Die Gebietslage der Stadtteile bzw. aller anderen Unterobjekte kann in Abhängigkeit zu ihrer örtlichen Lage im Stadtgebiet gesetzt werden. Dafür wird die Entfernung des Mittelpunkts des Unterobjekts zum *örtlichen Zentrum* der Stadt bestimmt ( $d_i$ ). Um diese Werte in Beziehung zu setzen, wird zusätzlich die maximale Distanz der Stadt-Eckpunkte zum Zentrum berechnet. Diese Distanz ( $d_{max}$ ) dient fortan als Referenzwert, der die größte Entfernung zum *örtlichen Zentrum* innerhalb des Stadtgebiets beschreibt. Die Gebietslage kann nun durch folgende Formel bestimmt werden:

$$locality_{i_{child}} = \begin{cases} Stadtrand & d_i > d_{max} * \frac{2}{3} \\ Stadtkern & d_i < d_{max} * \frac{1}{3} \\ Stadtmantel & sonst \end{cases} \quad (38)$$

- **Flächennutzungstypen**

Die Verteilung der einzelnen Flächennutzungstypen eines Unterobjekts ( $FN_{i_{child}}$ ) wird in Abhängigkeit der Gebietslage und den Flächennutzungstypen der Stadt ( $FN_{i_{parent}}$ ) berechnet. Dabei werden Wohn- und Industrieflächen näher zum Stadtrand, sowie Büro- und Einzelhandelsflächen näher zum Stadtkern gesetzt.

Für ein Unterobjekt am *Stadtrand* werden die Flächennutzungstypen wie folgt verteilt (mit  $\{x = rand(a, b) | x \in \mathbb{R} \wedge a < x < b\}$ ):

$$\begin{aligned} FN_{Industrie_{child}} &= rand(0.25, 1.25) * FN_{Industrie_{parent}} \\ FN_{Wohnen_{child}} &= rand(0.5, 1.5) * FN_{Wohnen_{parent}} \\ FN_{Büro_{child}} &= rand(0, 0.5) * FN_{Büro_{parent}} \\ FN_{Einzelhandel_{child}} &= rand(0, 0.75) * FN_{Einzelhandel_{parent}} \\ FN_{Gastronomie_{child}} &= rand(0, 0.75) * FN_{Gastronomie_{parent}} \\ FN_{Freifläche_{child}} &= rand(0, 1) * FN_{Freifläche_{parent}} \end{aligned} \quad (39)$$

Für ein Unterobjekt im *Stadtmantel* werden die Flächennutzungstypen einheitlich durch eine Normalverteilung bestimmt (mit  $\{x = rand(a, b) | x \in \mathbb{R} \wedge a < x < b\}$ ):

$$FN_{i_{child}} = rand(0, 1) * FN_{i_{parent}} \quad (40)$$

Für ein Unterobjekt im *Stadtkern* werden die Flächennutzungstypen wie folgt verteilt (mit  $\{x = rand(a, b) | x \in \mathbb{R} \wedge a < x < b\}$ ):

$$\begin{aligned} FN_{Industrie_{child}} &= rand(0, 0.5) * FN_{Industrie_{parent}} \\ FN_{Wohnen_{child}} &= rand(0, 0.75) * FN_{Wohnen_{parent}} \\ FN_{Büro_{child}} &= rand(0.5, 1.5) * FN_{Büro_{parent}} \\ FN_{Einzelhandel_{child}} &= rand(0.25, 1.25) * FN_{Einzelhandel_{parent}} \\ FN_{Gastronomie_{child}} &= rand(0, 1) * FN_{Gastronomie_{parent}} \\ FN_{Freifläche_{child}} &= rand(0, 1) * FN_{Freifläche_{parent}} \end{aligned} \quad (41)$$

Diese Werte werden im Nachhinein immer nur im Zusammenhang mit der Wertesumme  $\sum_{i=0}^{n-1} FN_i$  interpretiert, daher lässt sich mit diesen Formeln eine sinnvolle Verteilung durchführen. Der absolute prozentuale Wert eines Flächennutzungstyps wird dabei wie folgt berechnet:

$$percent_{FN_i} = \frac{FN_i}{\sum_{i=0}^{n-1} FN_i} * 100 \quad (42)$$

## 6.2 Stadtteil

Das Objekt *Stadtteil* definiert nach der Stadt das nächst kleinere städtische Gebiet bzw. morphogenetische Gliederungs-Element. Ein Stadtteil wird hier in eine Menge von Hausblöcken unterteilt, die von Straßen umgeben sind.

Wie bei einem *Stadt*-Objekt kann durch den SLG die Grundriss-Generierung nur vereinfacht vorgenommen werden, wodurch die Hausblöcke an den Außenseiten einer Stadtteilfläche entstehen.

### 6.2.1 Parameter und vordefinierte Werte eines Stadtteils

Für ein *Stadtteil*-Objekt werden die folgenden Parameter und vordefinierten Werte verwendet.

#### Grundriss-Parameter

Die Teilflächen eines Stadtteils sollen wie bei einer Stadt gleichmäßig und mit einer bestimmten Fläche erstellt werden, wobei ihre Form konkav sein kann, falls sie eine Ecke des Stadtteil-Umrisses beinhaltet. Dazu gibt es Zwischenräume für Straßen, die eine geringere Breite als die Hauptstraßen besitzen (vgl. Tab. 6).

Parametername	vordefinierter Wert
Anzahl der Begrenzungs-Knoten	<i>wird berechnet</i>
Typ der Berechnungsvorschrift für die Anzahl der Begrenzungs-Knoten	nach Umfang
Durchschnittlicher Flächeninhalt einer Teilfläche	5000m <sup>2</sup>
Seitenverhältnis der Teilflächen	1 : 2
Skalierungsfaktor und Exponent für Wachstumsfunktion	0; 0
Minimale Anzahl von Begrenzungs-Knoten	1
Positions-Abweichung der Begrenzungs-Knoten	0.1
Typ der Zwischenräume	Straßen
Breite der Zwischenräume	10m
Start-Knoten für Zwischenräume	Straßenseite
Kürzeste und längste Kante	<i>wird berechnet</i>
Maximale Anzahl von überstumpfen Winkeln für Pfade am Außenrand, Außenrand mit Ecke und Außenrand mit konkaver Ecke	0; 1; 0

Tabelle 6: Grundriss-Parameter eines Stadtteil-Objekts.

### Allgemeine stadtgeographische Parameter

Diese Parameter werden normalerweise durch das Oberobjekt *Stadt* gesetzt. Ist dies nicht der Fall, also der Stadtteil gehört zu keiner Stadt und wurde anderweitig erstellt, müssen die Parameterwerte vordefiniert werden. In Tabelle 7 wird eine beispielhafte Wertzuweisung wiedergegeben.

Parametername	vordefinierter Wert
Flächennutzung Industrie	0%
Flächennutzung Wohnen	55%
Flächennutzung Büro	20%
Flächennutzung Einzelhandel	20%
Flächennutzung Gastronomie	5%
Flächennutzung Freiflächen (Grünflächen)	0%
Gebietslage	Stadtkern
Kulturraum	Europäisch

Tabelle 7: Allgemeine stadtgeographische Parameter eines Stadtteil-Objekts.

### 6.2.2 Allgemeine stadtgeographische Parameter von Unterobjekten

Die Unterobjekte bzw. Hausblöcke des Stadtteils sollen durch seine stadtgeographischen Parameter und ggf. dem vorhandenen *Stadt*-Objekt beeinflusst werden. Dafür werden die allgemeinen stadtgeographischen Parameter jedes einzelnen *Hausblock*-Objekts wie folgt bestimmt:

- **Gebietslage**

Die Gebietslage der Hausblöcke wird wie bei den Unterobjekten einer Stadt in Abhängigkeit zur örtlichen Lage im Stadtgebiet gesetzt (siehe Kapitel 6.1.2).

- **Flächennutzungstypen**

Die Flächennutzungstypen der Hausblöcke werden ebenfalls in einem ersten Berechnungsschritt (wie bei den Unterobjekten einer Stadt) in Abhängigkeit zur Gebietslage und der Flächennutzung des Stadtteils berechnet (siehe Kapitel 6.1.2).

Danach werden einzelne Werte je nach vorherrschender Flächennutzung auf Null gesetzt, um Hausblöcke mit einer möglichst sinnvollen funktionalen Nutzung zu generieren. Z.B sollten in Industriegebieten keine Wohngebäude stehen.

Dafür wird zunächst die vorherrschende Flächennutzung  $FN_{general}$  (*Wohn-, Gewerbe-, Industrie* oder *Mischgebiet*) aus den Flächennutzungstypen bestimmt:

$$\begin{aligned}
 FN_{total} &= \sum_{i=0}^{n-1} FN_i \\
 FN_{general} &= \begin{cases} \text{Industriegebiet} & \frac{FN_{Industrie}}{FN_{total}} \geq \frac{1}{2} FN_{total} \\ \text{Gewerbegebiet} & \frac{FN_{Büro} \wedge FN_{Einzelhandel}}{FN_{total}} \geq \frac{1}{2} FN_{total} \\ \text{Wohngebiet} & \frac{FN_{Wohnen}}{FN_{total}} \geq \frac{1}{2} FN_{total} \\ \text{Mischgebiet} & \text{sonst} \end{cases} \quad (43)
 \end{aligned}$$

Hierbei muss der Anteil des jeweiligen Flächennutzungstyps größer oder gleich 50% von der gesamten Flächennutzung betragen, um die vorherrschende Flächennutzung festlegen zu können. Die einzelnen Flächennutzungstypen werden nun wie folgt zugewiesen (mit *(Industrie, Büro, Wohnen, Einzelhandel, Gastronomie, Unbebaut)* als 6-Tupel von Faktoren):

$$FN_{child} = FN_{child} * \begin{cases} (1, 0, 0, 0, 0, 1) & FN_{general} = \text{Industriegebiet} \\ (1, 1, 0, 1, 1, 1) & FN_{general} = \text{Industrieg.} + \text{Gewerbegeb.} \\ (0, 1, 0, 1, 1, 1) & FN_{general} = \text{Gewerbegebiet} \\ (0, 0, 1, 1, 1, 1) & FN_{general} = \text{Wohngebiet} \\ (0, 1, 1, 1, 1, 1) & \text{sonst} \end{cases} \quad (44)$$

### 6.3 Hausblock

Ein Hausblock repräsentiert im stadtgeographischen Zusammenhang die kleinste Einheit, die von Straßen umgeben ist. Seine Fläche wird bei der Grundriss-Erstellung durch den zugehörigen SLG in Grundstücke unterteilt und besitzt keine Zwischenräume, bzw. ist nicht von Straßen durchzogen.

Das hier entwickelte Verfahren ist schwerpunktmäßig auf Hausblock- und alle unteren Stadtobjekte angepasst, wodurch in diesen Hierarchiestufen möglichst realistische Grund- und Aufrisse in Abhängigkeit zu den stadtgeographischen Parametern generiert werden können.

#### 6.3.1 Parameter und vordefinierte Werte eines Hausblocks

Für ein *Hausblock*-Objekt werden die folgenden Parameter und vordefinierten Werte verwendet.

##### Objektspezifische Parameter

Diese Parameter werden für einen Hausblock wie folgt definiert:

- **Höhe**  
Dieser aufrisspezifische Wert bezeichnet den vertikalen Abstand zwischen Straße und Bordsteinkante.
- **Durchschnittliche Gebäudefläche**  
Die durchschnittliche Gebäudefläche wird auf dieser Ebene angegeben, um möglichst gleichgroße Gebäudegrundflächen innerhalb eines Hausblocks zu erhalten. Dieser Wert wird durch die vorherrschende Flächennutzung und Gebietslage des Hausblocks beeinflusst und später bei der Gebäudeplatzierung von den einzelnen *Grundstücks*-Objekten verwendet.
- **LOD-Stufen**  
Die LOD-Liste für dieses Objekt enthält Einträge aus folgender Parameterstruktur (siehe Kapitel 5.3.4):
  - Stufennummer
  - Entfernung

- (Textur<sup>72</sup>)
- Transparenz

Für dieses Objekt enthält die Liste nur einen Eintrag<sup>73</sup>, da ein Hausblock-Modell hier nur in einer Detailstufe generiert werden kann.

- **Textur**

Dieser Parameter enthält den Dateinamen der verwendeten Texturemap. Die Textur (256 × 512 Pixel) muss dabei zwei übereinander liegende quadratische Bildmotive für Gehweg- und Hausblock-Fläche enthalten.

Die Werte dieser Parameter werden wie in Tabelle 8 beispielhaft zugewiesen.

Parametername	vordefinierter Wert
Höhe	25cm
Durchschnittliche Gebäudefläche	wird berechnet
LOD-Stufen	(0, 0, nein)
Textur	block.1.tga

Tabelle 8: Objektspezifische Parameter eines Hausblock-Objekts.

### Grundriss-Parameter

Die Teilflächen werden, wie bei den bisherigen Objekten, gleichmäßig und mit einer bestimmten Fläche erstellt, wobei ihre Form im Inneren der Hausblock-Fläche nicht konkav sein darf. Die durchschnittliche Größe von Grundstücken (und auch Gebäuden) wird später durch die stadtgeographischen Parameter berechnet. Zwischenräume werden hier nicht benötigt und bleiben undefiniert. Alle weiteren Grundriss-Parameter werden beispielhaft zugewiesen (vgl. Tab. 9).

Parametername	vordefinierter Wert
Anzahl der Begrenzungs-Knoten	wird berechnet
Typ der Berechnungsvorschrift für die Anzahl der Begrenzungs-Knoten	nach Umfang
Durchschnittlicher Flächeninhalt einer Teilfläche	wird berechnet
Seitenverhältnis der Teilflächen	1 : 2
Skalierungsfaktor und Exponent für Wachstumsfunktion	0; 0
Minimale Anzahl von Begrenzungs-Knoten	1
Positions-Abweichung der Begrenzungs-Knoten	0.05
Typ der Zwischenräume	Undefiniert
Breite der Zwischenräume	0m
Start-Knoten für Zwischenräume	Undefiniert
Kürzeste und längste Kante	wird berechnet
Maximale Anzahl von überstumpfen Winkeln für Pfade am Außenrand, Außenrand mit Ecke und Außenrand mit konkaver Ecke	0; 0; 0

Tabelle 9: Grundriss-Parameter eines Hausblock-Objekts.

<sup>72</sup>Die Textur wird aus den objektspezifischen-Parametern übernommen

<sup>73</sup>In der zugehörigen Tabelle als 3-Tupel: (Stufennummer, Entfernung, Transparenz).

### Allgemeine stadtgeographische Parameter

Diese Parameter werden normalerweise durch das Oberobjekt *Stadtteil* gesetzt. Ist dies nicht der Fall, also der Hausblock wurde anderweitig erstellt, müssen die Parameterwerte vordefiniert werden. Tabelle 10 veranschaulicht eine beispielhafte Zuweisung.

Parametername	vordefinierter Wert
Flächennutzung Industrie	0%
Flächennutzung Wohnen	75%
Flächennutzung Büro	0%
Flächennutzung Einzelhandel	20%
Flächennutzung Gastronomie	5%
Flächennutzung Freiflächen (Grünflächen)	0%
Gebietslage	Stadtkern
Kulturraum	Europäisch

Tabelle 10: Allgemeine stadtgeographische Parameter eines Hausblock-Objekts.

Mit Hilfe dieser vordefinierten bzw. durch einen Stadtteil vorgegebenen Parameter können nun die durchschnittlichen Flächengrößen von Grundstücken und Gebäuden berechnet werden.

Die *durchschnittliche* Grundstücks-Fläche (*durchschnittlicher Flächeninhalt einer Teilfläche*) wird in Abhängigkeit zur Gebietslage und der vorherrschenden Flächennutzung bestimmt (vorherrschende Flächennutzung und Gebietslage werden zur besseren Übersichtlichkeit als Paar dargestellt ( $FN_{general, locality}$ )).

$$area_{Grundstück} = \begin{cases} 150m^2 & (Wohngebiet \vee Mischgebiet, Stadtkern) \\ 175m^2 & (Wohngebiet \vee Mischgebiet, Stadtmantel) \\ 200m^2 & (Wohngebiet \vee Mischgebiet, Stadtrand) \\ 250m^2 & (Gewerbegebiet, *) \\ 300m^2 & (Industriegebiet, *) \end{cases} \quad (45)$$

Die *durchschnittliche Gebäudefläche* wird wie folgt bestimmt:

$$area_{Gebäude} = \begin{cases} 125m^2 & (Wohngebiet \vee Mischgebiet, Stadtkern) \\ 100m^2 & (Wohngebiet \vee Mischgebiet, Stadtmantel) \\ 75m^2 & (Wohngebiet \vee Mischgebiet, Stadtrand) \\ 200m^2 & (Gewerbegebiet, *) \\ 250m^2 & (Industriegebiet, *) \end{cases} \quad (46)$$

Aus beiden Werten kann später die Anzahl der Gebäude-Stockwerke mit Hilfe der Geschossflächenzahl (GFZ) abgeleitet werden (siehe Kapitel 3.3 und 6.5.1).

### 6.3.2 Allgemeine stadtgeographische Parameter von Unterobjekten

Wie bei allen vorigen Objekten, sollen auch hier alle Unterobjekte durch die Eigenschaften ihres Oberobjekts beeinflusst werden. Die Werte der stadtgeographischen Parameter eines *Grundstücks*-Objekts ergeben sich dabei wie folgt:

- **Gebietslage**

Die Gebietslage eines Grundstücks wird für ein einheitlicheres Stadtbild vom Hausblock übernommen und nicht wie bei den oberen Hierarchiestufen in Abhängigkeit zur örtlichen Lage im Stadtgebiet berechnet.

- **Flächennutzungstypen**

Die einzelnen Flächennutzungstypen jedes Grundstücks werden zunächst durch eine Normalverteilung berechnet, da sie die gleiche Gebietslage wie der zugehörige Hausblock besitzen (mit  $\{x = rand(a, b) | x \in \mathbb{R} \wedge a < x < b\}$ ):

$$FN_{child} = rand(0, 1) * FN_{iparent} \quad (47)$$

Danach werden einzelne Werte je nach vorherrschender Flächennutzung auf Null gesetzt, um später nur Gebäude zu generieren, die entweder nur einen oder mehrere sinnvoll kombinierte Flächennutzungstypen besitzen. Z.B sollen damit Industrie-Gebäude mit zusätzlichen Wohnbereichen vermieden werden. Hierfür wird die vorherrschende Flächennutzung ( $FN_{general}$ , siehe Kapitel 6.2.2) eines Hausblocks bestimmt und die Flächennutzungstypen wie folgt angepasst (mit *(Industrie, Büro, Wohnen, Einzelhandel, Gastronomie, Unbebaut)* als 6-Tupel von Faktoren):

$$FN_{child} = FN_{child} * \begin{cases} (0, 0, 0, 0, 0, 1) & FN_{general} = Unbebaut \\ (1, 0, 0, 0, 0, 0) & FN_{general} = Industriegebiet \\ (0, 1, 0, 1, 1, 0) & FN_{general} = Gewerbegebiet \\ (0, 0, 1, 1, 1, 0) & FN_{general} = Wohngebiet \\ (0, 1, 1, 1, 1, 0) & sonst \end{cases} \quad (48)$$

## 6.4 Grundstück

Ein Grundstück ist hier immer ein Unterobjekt eines Hausblocks und kann nur in diesem Zusammenhang erstellt werden. Der SLG eines Grundstücks wird bei der Grundriss-Erstellung nicht unterteilt und die Flächen für Gehwege, eigentliche Grundstücksfläche und Gebäudefläche durch zusätzliche Flächen-Manipulationen generiert (siehe Kapitel 4.6). Folglich hat jedes Grundstück maximal ein Unterobjekt vom Typ *Gebäude*, das durch weitere Berechnungen realistisch auf der Gesamtfläche platziert werden kann. In den folgenden Unterkapiteln werden die dafür notwendigen Berechnungsschritte und auch alle anderen Parameter-Zuweisungen beschrieben.

### 6.4.1 Parameter und vordefinierte Werte eines Grundstücks

Für ein *Grundstück*-Objekt werden keine Grundriss-Parameter benötigt, da die Flächenaufteilung in Gehweg und bebaute Fläche, sowie die Platzierung des Gebäudes allein durch das Einfügen und Verschieben von Kanten durchgeführt wird.

Damit ergeben sich folgende Parametersätze:

#### Objektspezifische Parameter

Für dieses Objekt werden die nachfolgenden spezifischen Parameter definiert, welche hauptsächlich für die Platzierung von Gebäuden verwendet werden:

- **Art der Bauweise** [ *Geschlossen* | *Halboffen* | *Offen* ]  
Die Art der Bauweise bestimmt maßgeblich, welche Seiten des später entstehenden Gebäudes an den Grundstücksgrenzen anliegen (siehe Kapitel 3.3). Dieser Wert wird später anhand der allgemeinen stadtgeographischen Parameter berechnet und legt zusätzlich die Größe der Gebäudefläche fest.
- **Ausrichtung des Gebäudes** [ *Vorderseite* | *Mitte* | *Rückseite* ]  
Dieser Parameter wird zusätzlich zur *Art der Bauweise* für die Gebäudeplatzierung definiert und bestimmt die Gebäudelage relativ zur Grundstückstiefe.  
Die *Ausrichtung des Gebäudes* ( $alignment_{Gebäude}$ ) wird in Abhängigkeit zur *Art der Bauweise* ( $c$ ) bestimmt (mit  $\{x = rand(a, b) | x \in \mathbb{N} \wedge a \leq x \leq b\}$ ):

$$alignment_{Gebäude} = \begin{cases} \text{Vorderseite} & c = \text{Geschlossen} \\ \text{Vorderseite} & c = \text{Halboffen} \\ \text{Vorderseite} & c = \text{Offen} \wedge rand(0, 2) = 0 \\ \text{Mitte} & c = \text{Offen} \wedge rand(0, 2) = 1 \\ \text{Rückseite} & c = \text{Offen} \wedge rand(0, 2) = 2 \end{cases} \quad (49)$$

- **Dachausrichtung** [ *Giebelständig* | *Traufständig* ]  
Die Ausrichtung des Dachgiebels bzw. der Dachtraufe gibt an, welche Dachseite zur Straße ausgerichtet ist (siehe Kapitel 3.3). Zusätzlich kann mit diesem Parameter berechnet werden, wie die Länge der vorderen/hinteren und linken/rechten Gebäudeseiten im Verhältnis stehen.  
Die *Dachausrichtung* ( $alignment_{Dach}$ ) wird hier mit Hilfe der *Art der Bauweise* ( $c$ ) bestimmt (mit  $\{x = rand(a, b) | x \in \mathbb{N} \wedge a \leq x \leq b\}$ ):

$$alignment_{Dach} = \begin{cases} \text{Giebelständig} & c = \text{Geschlossen} \\ \text{Giebelständig} & c = \text{Halboffen} \\ \text{Giebelständig} & c = \text{Offen} \wedge rand(0, 1) = 0 \\ \text{Traufständig} & c = \text{Offen} \wedge rand(0, 1) = 1 \end{cases} \quad (50)$$

- **Verhältnis der Gebäudeseiten**  
Das Verhältnis der Gebäudeseiten kann direkt nach Zuweisung der Dachausrichtung bestimmt werden. Dafür wird gemäß Kapitel 3.3 angenommen, dass die Dachtraufe i.A. an der längeren Gebäudeseite (Breitseite) liegt und sich damit das Verhältnis ( $ratio_{Gebäude}$ ) wie folgt berechnen lässt (mit  $\{x = rand(a, b) | x \in \mathbb{R} \wedge a \leq x \leq b\}$ ):

$$ratio_{Gebäude} = \begin{cases} rand(\frac{2}{3}, 1) & alignment_{Dach} = \text{Traufständig} \\ rand(1, \frac{5}{3}) & alignment_{Dach} = \text{Giebelständig} \end{cases} \quad (51)$$

- **Breite des Gehwegs**

Die Werte dieser Parameter werden wie in Tabelle 11 beispielhaft zugewiesen.

Wurden Bauweise und Ausrichtung des Gebäudes bestimmt, müssen anschließend weitere Werte für die Platzierung der Gebäude-Teilfläche berechnet werden. Für Größen- und Position-Anpassung des späteren Gebäudes wird dabei für jede Teilflächen-Seite die Distanz bestimmt, um die sie ins Innere der

Parametername	vordefinierter Wert
Art der Bauweise	wird berechnet
Ausrichtung des Gebäudes	wird berechnet
Dachausrichtung	wird berechnet
Verhältnis der Gebäudeseiten	wird berechnet
Breite des Gehwegs	2.5 – 3m

Tabelle 11: Objektspezifische Parameter eines Grundstück-Objekts.

Grundstücksfläche verschoben wird.

Für diese Platzierung wird die Rückseite der Gebäude-Teilfläche zunächst parallel zu der Vorderseite gelegt. Liegt das Grundstück nicht an einer Hausblock-Ecke, entsteht somit immer eine rechteckige Fläche. Beinhaltet das Grundstück eine Hausblock-Ecke, entsteht dabei eine Fläche mit mindestens vier rechten Winkeln und einer maximalen Anzahl von parallelen Kanten (siehe Kapitel 4.3.6). Durch diese vorbereitende Manipulation entstehen bereits Gebäude-Grundflächen die für ein Aufriss-Verfahren verwendet werden können. In dieser Arbeit wird die Gebäude-Platzierung allerdings noch weiter differenziert und hinsichtlich des stadtgeographischen Zusammenhangs angepasst.

Die vier Distanzen für untere (vordere), linke, rechte und obere (hintere) Seite werden in Abhängigkeit zur Bauweise ( $c$ ), Ausrichtung ( $alignment$ ) und dem Seitenverhältnis ( $ratio$ ) des Gebäudes, sowie der durchschnittlichen Gebäudefläche des zugehörigen Hausblocks ( $area_{average}$ ) berechnet:

1. Es werden die Kantenlisten für die untere, linke und rechte Seite der Gebäude-Teilfläche bestimmt ( $edges_{bottom}$ ,  $edges_{left}$  und  $edges_{right}$ , siehe Kapitel 4.6.2).

2. Nun werden folgende Längen berechnet:

- Die Gesamtlänge von unterer, linker und rechter Seite  $l_{edges_{bottom}}$ ,  $l_{edges_{left}}$  und  $l_{edges_{right}}$
- Die durchschnittliche Länge der Gebäude-Vorderseite ( $l_{bottom_{average}}$ ) ergibt sich aus dem Minimum von optimaler und effektiver Länge der Gebäude-Vorderseite ( $l_{bottom_{optimal}}$ ,  $l_{bottom_{effective}}$ ):

$$\begin{aligned}
l_{bottom_{optimal}} &= \frac{\sqrt{area_{average}}}{ratio} \\
l_{bottom_{effective}} &= \frac{l_{edges_{bottom}}}{num_{edges_{bottom}}} \\
l_{bottom_{average}} &= \min(l_{bottom_{optimal}}, l_{bottom_{effective}}) \quad (52)
\end{aligned}$$

- Die Distanz ( $d_{width}$ ), um welche linke und/ oder rechte Seite insgesamt geschrumpft werden sollen (mit  $\{x = rand(a, b) | x \in \mathbb{R} \wedge a \leq x \leq b\}$ ):

$$d_{width} = rand(l_{bottom_{effective}}, l_{edges_{bottom}} * ratio) \quad (53)$$

$d_{width}$  wird hierbei so bestimmt, dass das Seitenverhältnis  $ratio$  eingehalten werden kann. (Die Distanz ( $d_{depth}$ ) für untere und obere Seite wird erst später in Abhängigkeit der Bauweise bestimmt.)

3. Nun kann die Distanz  $d_{depth}$ , sowie die einzelnen Beträge zum Schrumpfen der vier Gebäude-Seiten ( $d_{bottom}$ ,  $d_{left}$ ,  $d_{right}$  und  $d_{top}$ ) in Abhängigkeit zu den objektspezifischen Parametern berechnet werden (mit  $d_{depth} = \min(l_{edges_{left}}, l_{edges_{right}})$ ):

**Geschlossene Bauweise**

$$\begin{aligned}
 d_{depth} &= d_{depth} - l_{bottom_{average}} * ratio \\
 (d_{top}, d_{bottom}) &= \begin{cases} (d_{depth}, 0) & alignment = Vorderseite \\ (\frac{d_{depth}}{2}, \frac{d_{depth}}{2}) & alignment = Mitte \\ (0, d_{depth}) & alignment = Rückseite \end{cases} \\
 (d_{left}, d_{right}) &= (0, 0)
 \end{aligned} \tag{54}$$

**Halboffene Bauweise**

$$\begin{aligned}
 d_{depth} &= d_{depth} - (l_{bottom_{effective}} - d_{width}) * ratio \\
 (d_{top}, d_{bottom}) &= \begin{cases} (d_{depth}, 0) & alignment = Vorderseite \\ (\frac{d_{depth}}{2}, \frac{d_{depth}}{2}) & alignment = Mitte \\ (0, d_{depth}) & alignment = Rückseite \end{cases} \\
 (d_{left}, d_{right}) &= (0, 0)
 \end{aligned} \tag{55}$$

**Offene Bauweise (mit  $\{x = rand(a, b) | x \in \mathbb{R} \wedge a \leq x \leq b\}$ )**

$$\begin{aligned}
 d_{depth} &= d_{depth} - (l_{bottom_{effective}} - d_{width}) * ratio \\
 x &= \begin{cases} rand(0.1, 0.3) & alignment = Vorderseite \\ rand(0.4, 0.6) & alignment = Mitte \\ rand(0.7, 0.8) & alignment = Rückseite \end{cases} \\
 d_{bottom} &= d_{depth} * x \\
 d_{top} &= d_{depth} * (1 - x) \\
 (d_{left}, d_{right}) &= (\frac{d_{width}}{2}, \frac{d_{width}}{2})
 \end{aligned} \tag{56}$$

Die Seiten der Gebäude-Teilfläche werden abschließend mit dem SLG-Verfahren nach Kapitel 4.6.2 um die berechneten Werte verschoben.

**Allgemeine stadtgeographische Parameter:**

Diese Parameter werden immer durch das Oberobjekt *Hausblock* gesetzt und müssen daher nicht vordefiniert werden (vgl. Tab. 14).

Mit den vorhandenen Parameterwerten kann nun die *Art der Bauweise* bestimmt werden. Dafür wird die vorherrschende Flächennutzung des Grundstücks wie in Kapitel 6.2.2 berechnet und mit Hilfe der Gebietslage die Bauweise (*c*) zugewiesen (vorherrschende Flächennutzung und Gebietslage werden zur besseren Übersichtlichkeit als Paar dargestellt ( $FN_{general}, locality$ )):

$$c = \begin{cases} \textit{Geschlossen} & (*, \textit{Stadtkern}) \\ \textit{Halboffen} & (\textit{Wohngebiet} \vee \textit{Mischgebiet}, \textit{Stadtmantel}) \\ \textit{Offen} & (\textit{Industriegebiet}, \textit{Stadtmantel}) \\ \textit{Offen} & (\textit{Gewerbegebiet}, \textit{Stadtmantel}) \\ \textit{Offen} & (*, \textit{Stadtrand}) \end{cases} \tag{57}$$

## 6.4.2 Allgemeine stadtgeographische Parameter von Unterobjekten

Für jedes Grundstück wird maximal nur ein *Gebäude*-Objekt generiert. Dieses Unterobjekt bekommt die allgemeinen stadtgeographischen Parameter des Grundstücks zugewiesen, die bereits auf Hausblock-Ebene berechnet wurden (vgl. Tab. 10).

Die Zuweisung der *Gebäude*-Parameter ist damit beendet.

## 6.5 Gebäude

Ein Gebäude wird entweder auf einem Grundstück automatisch platziert oder manuell definiert. Dieses Objekt besitzt mindestens ein Stockwerk-Unterobjekt und ist insbesondere für die Aufriss-Erstellung ein wichtiges Element, wobei für die Generierung der zugehörigen 3D-Modellen die beiden unteren Hierarchiestufen (Stockwerk und Wohnung) vollständig erstellt werden müssen.

Der SLG eines Gebäudes wird bei der Grundriss-Generierung nicht unterteilt und damit nur eine Teilfläche mit den Ausmaßen seiner Grundfläche erstellt, dessen Eckpunkte für jedes einzelne Stockwerk übernommen werden. Der eigentliche Gebäude-Grundriss wird hier erst in den unteren Hierarchiestufen durch die zugehörigen SLGs der Stockwerke und Wohnungen erstellt. Des Weiteren werden die Unterobjekte eines Gebäudes in vertikaler statt in horizontaler Ausdehnung definiert (Erd- bis Dachgeschoss).

Das Besondere ist hier, dass sämtliche Stockwerke *einzel*n und *unabhängig voneinander* bei der Parameterzuweisung behandelt werden können. Damit werden durch das Aufriss-Verfahren Gebäude generiert, die auf allen oder einigen Stockwerken *verschiedene Flächennutzungen* haben (natürlich in Abhängigkeit der stadtgeographischen Eigenschaften des zugehörigen Gebäude-Objekts). Die Stockwerk-Parameter sind dabei so angelegt, das für jede Etage u.a die Form von Fenstern und Türen verändert werden kann.

Z.B. könnte ein fünfstöckiges Gebäude im Erdgeschoss Geschäfte mit *großen Schaufensterreihen*, im 2. und 3. Stockwerk Büroräume mit *mittelgroßen Doppelfenstern* und in den oberen Stockwerken Wohnungen mit *kleineren, einzelnen Fenstern* enthalten.

Welche Parameter für ein Gebäude und seine Stockwerke definiert werden, wird in den folgenden Unterkapiteln behandelt.

### 6.5.1 Parameter und vordefinierte Werte eines Gebäudes

Für ein *Gebäude*-Objekt wird zwar keine Flächen-Unterteilung durchgeführt, aber die Grundriss-Parameter für alle Stockwerke an dieser Stelle einheitlich definiert. Dieses Verfahren resultiert folglich in identischen Grundrissen auf jedem Stockwerk (mit Wohnungen) und wird für das Einfügen eines Treppenhauses, aber auch für regelmäßige Fassaden benötigt (siehe Kapitel 5.2)<sup>74</sup>.

Die folgenden Parameter und vordefinierten Werte werden verwendet:

---

<sup>74</sup>Für verschiedene Arten von Stockwerks-Innenräumen in einem Gebäude wird später jeweils ein zusätzlicher Parameter definiert.

## Objektspezifische Parameter

Für ein Gebäude werden folgende Parameter definiert:

- **Höhe**  
Die Gebäudehöhe wird später durch die Stockwerks-Anzahl und -Höhen bestimmt.
- **Anzahl der Stockwerke**  
Die Anzahl der Stockwerke eines Gebäudes wird hier mit Hilfe der stadtgeographischen Parameter bestimmt und ihre maximale Anzahl durch die Stadtgröße beeinflusst.  
Zusätzlich wird für die spätere Zuweisung der Stockwerk-Parameter die *Anzahl der Stockwerke pro Flächennutzungstyp* benötigt und ebenfalls durch die stadtgeographischen Parameter berechnet.
- **Wandbreite bzw. Mauerdicke**  
Dieser Parameter definiert die Dicke einer Außen- und Innenwand.
- **Flächengröße des Treppenhauses**  
Dieser Flächeninhalt wird für das Einfügen des Treppenhauses bei der Grundriss-Erweiterung benötigt (siehe Kapitel 5.2).
- **Dach-Parameter**  
Diese Parameter-Struktur definiert die Eigenschaften des Gebäudedachs wie folgt:
  - **Typ** [*Satteldach* | *Zeltdach* | *Flachdach*]
  - **Höhe** (nicht für Flachdach)
  - **Dicke** (nicht für Flachdach)
  - **Überhang am Giebel** (nur für Satteldach)
  - **Überhang an der Traufe** (nicht für Flachdach)
  - **Ausrichtung** (wird vom Grundstück übernommen)
- **Stockwerk-Parameter**  
Die Parameter der einzelnen Stockwerke werden später in Abhängigkeit zu den allgemeinen stadtgeographischen Parametern berechnet und anschließend in einer Liste gespeichert. Über die Geschossnummer kann dann auf die einzelnen Einträge zugegriffen werden. Wie dieser Parametersatz definiert ist, wird zur besseren Übersichtlichkeit in Kapitel 6.6.1 beschrieben.
- **LOD-Stufen**  
Die LOD-Liste für dieses Objekt enthält Einträge<sup>75</sup> aus folgender Parameter-Struktur (siehe Kapitel 5.3.4):
  - Stufennummer
  - Entfernung
  - (Textur<sup>76</sup>)
  - Transparenz

---

<sup>75</sup>In der zugehörigen Tabelle als 6-Tupel: (*Stufennummer, Entfernung, Transparenz, Innenräume, Außenwände, Dach*).

<sup>76</sup>Die Textur wird aus den objektspezifischen-Parametern übernommen.

- Innenräume (mit Angaben für Fenster- und Türrahmen)
  - Außenwände (mit Angaben für Fenster, Eingänge, Fenster- und Türrahmen)
  - Dach
- **Textur**  
Dieser Parameter legt die verwendete Textur des Gebäudes fest. Die Textur-remap eines Gebäudes wurde so definiert, das pro Bild ( $768 \times 1024$  Pixel) ein Dach, 4 Wände (mit einer Teil-Überlappung), 8 Türen, 12 Fenster, 2 Böden, 2 Decken und diverse andere Motive in zweckmäßiger Qualität eingefügt werden können (vgl. Abb. 44 in Kapitel 5.3.2).  
Für eine realistisch wirkende Darstellung bzw. Oberflächenvielfalt der Gebäude werden mindestens vier dieser Texturen (Industrie-, Büro-, Wohn- und Mischgebäude) benötigt und je nach vorherrschender Flächennutzung ausgewählt<sup>77</sup>.

Die Werte dieser Parameter werden wie in Tabelle 12 beispielhaft zugewiesen.

Parametername	vordefinierter Wert
Höhe	<i>wird berechnet</i>
Anzahl der Stockwerke	<i>wird berechnet</i>
Wandbreite	30cm
Flächengröße des Treppenhauses	$5m^2$
Dach-Parameter	<i>werden berechnet</i>
Stockwerk-Parameter	<i>werden berechnet</i>
LOD-Stufen	(0, 0, ja, vorhanden & komplex, vorhanden & komplex, komplex), (1, 50, nein, nicht vorhanden, vorhanden & komplex, einfach), (2, 100, nein, nicht vorhanden, vorhanden & einfach, einfach)
Textur	building_living.1.tga

Tabelle 12: Objektspezifische Parameter eines Gebäude-Objekts.

### Grundriss-Parameter

Diese Grundriss-Parameter werden erst auf Stockwerk-Ebene verwendet. Dabei wird eine Wachstumsfunktion für die Anzahl der Begrenzungs-Knoten definiert, deren Ergebniswert bei größerer Gebäudefläche immer langsamer steigt (vgl. Abb. 49).

Die entstehenden Teilflächen können dabei konkav sein (außer an konkaven Ecken). Als Zwischenraum wird ein Hausflur generiert, dessen Eingang ( $loaction_{entrance}$ ) je nach Flächennutzung des Erdgeschosses ( $FN_{storey_0}$ ) platziert wird (Geschäfts- und Gastronomie-Räumlichkeiten besitzen jeweils eigene, sepa-

<sup>77</sup>Dabei werden die Dateinamen der Texturen für die Zuordnung verwendet, z.B. 'building\_industry....' für Industrie- oder 'building\_living....' für Wohngebäude.

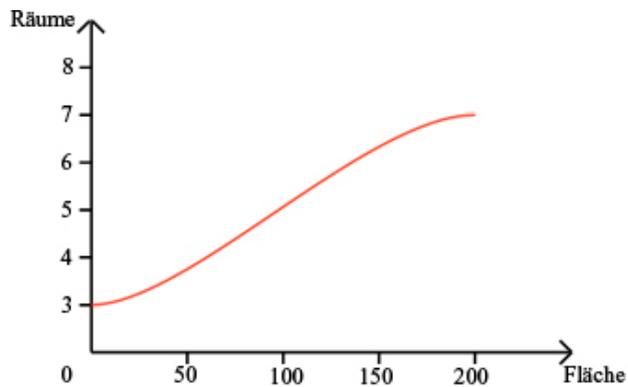


Abbildung 49: Beispielhafte Wachstumsfunktion für die Anzahl von Räumen pro Stockwerksfläche.

rate Eingänge zur Straßenseite):

$$loaction_{entrance} = \begin{cases} \text{Rückseite} & FN_{storey_0} = \text{Einzelhandel} \\ \text{Rückseite} & FN_{storey_0} = \text{Gastronomie} \\ \text{Straßenseite} & \text{sonst} \end{cases} \quad (58)$$

Alle weiteren Grundriss-Parameter werden beispielhaft zugewiesen. (vgl. Tab. 13).

Parametername	vordefinierter Wert
Anzahl der Begrenzungs-Knoten	wird berechnet
Typ der Berechnungsvorschrift für die Anzahl der Begrenzungs-Knoten	Wachstumsfunktion
Durchschnittlicher Flächeninhalt einer Teilfläche	30m <sup>2</sup>
Seitenverhältnis der Teilflächen	1 : 2
Skalierungsfaktor und Exponent für Wachstumsfunktion	15; 2
Minimale Anzahl von Begrenzungs-Knoten	1
Positions-Abweichung der Begrenzungs-Knoten	0.05
Typ der Zwischenräume	Hausflur
Breite der Zwischenräume	1.5m
Start-Knoten für Zwischenräume	wird berechnet
Kürzeste und längste Kante	wird berechnet
Maximale Anzahl von überstumpfen Winkeln für Pfade am Außenrand, Außenrand mit Ecke und Außenrand mit konkaver Ecke	1; 1; 0

Tabelle 13: Grundriss-Parameter eines Gebäude-Objekts.

### Allgemeine stadtgeographische Parameter

Diese Parameter werden durch das Oberobjekt *Hausblock* gesetzt. Ist dies nicht der Fall, also das Gebäude wurde anderweitig erstellt, müssen die Parameterwerte vordefiniert werden. Tabelle 14 veranschaulicht eine beispielhafte Zuweisung.

Parametername	vordefinierter Wert
Flächennutzung Industrie	0%
Flächennutzung Wohnen	50%
Flächennutzung Büro	40%
Flächennutzung Einzelhandel	10%
Flächennutzung Gastronomie	0%
Flächennutzung Freiflächen (Grünflächen)	0%
Gebietslage	Stadtmantel
Kulturraum	Europäisch

Tabelle 14: Allgemeine stadtgeographische Parameter eines Gebäude-Objekts.

Durch diese Werte können nun die objektspezifischen Parameter des Gebäudes berechnet werden.

1. Als ersten Schritt wird dafür die *Anzahl der Stockwerke* ( $num_{storeys}$ ) mit Hilfe der vorherrschenden Gebäudenutzung, der Gebietslage und der Stadtgröße<sup>78</sup> bestimmt. Als zusätzlicher Anhaltspunkt sollen dafür die (in Deutschland gültigen) Werte der Geschossflächenanzahl (GFZ) aus Kapitel 3.3 verwendet werden (z.B. Stadtkern-GFZ = 3.0). Generell sollten dabei Gebäude, die näher am Stadtkern liegen, eine größere Geschossanzahl besitzen (mit  $\{x = rand(a, b) | x \in \mathbb{N} \wedge a \leq x \leq b\}$  und  $(FN_{general, locality})$  als Paar von vorherrschender Flächennutzung, siehe Kapitel 6.2.2, und Gebietslage).

Die Anzahl der Stockwerke in einer *Großstadt*:

$$num_{storeys} = \begin{cases} 3 + rand(0, 2) & (\text{Industriegebiet}, *) \\ 5 + rand(0, 10) & (\text{Gewerbegebiet}, \text{Stadtkern}) \\ 4 + rand(0, 3) & (\text{Gewerbegebiet}, \text{Stadtmantel}) \\ 3 + rand(0, 2) & (\text{Gewerbegebiet}, \text{Stadtrand}) \\ 4 + rand(0, 3) & (\text{Wohngebiet}, \text{Stadtkern}) \\ 3 + rand(0, 2) & (\text{Wohngebiet}, \text{Stadtmantel}) \\ 2 + rand(0, 1) & (\text{Wohngebiet}, \text{Stadtrand}) \\ 5 + rand(0, 2) & (\text{Mischgebiet}, \text{Stadtkern}) \\ 4 + rand(0, 2) & (\text{Mischgebiet}, \text{Stadtmantel}) \\ 3 + rand(0, 1) & (\text{Mischgebiet}, \text{Stadtrand}) \end{cases} \quad (59)$$

<sup>78</sup>Falls das Gebäude zu keinem Stadtobjekt gehört, wird für den Stadtgröße-Parameter der Wert *Mittelstadt* angenommen.

Die Anzahl der Stockwerke in einer *Mittelstadt*:

$$num_{storeys} = \begin{cases} 2 + rand(0, 2) & (\text{Industriegebiet, *}) \\ 4 + rand(0, 4) & (\text{Gewerbegebiet, Stadtkern}) \\ 3 + rand(0, 3) & (\text{Gewerbegebiet, Stadtmantel}) \\ 2 + rand(0, 2) & (\text{Gewerbegebiet, Stadtrand}) \\ 3 + rand(0, 3) & (\text{Wohngebiet, Stadtkern}) \\ 2 + rand(0, 2) & (\text{Wohngebiet, Stadtmantel}) \\ 1 + rand(0, 1) & (\text{Wohngebiet, Stadtrand}) \\ 4 + rand(0, 3) & (\text{Mischgebiet, Stadtkern}) \\ 3 + rand(0, 2) & (\text{Mischgebiet, Stadtmantel}) \\ 2 + rand(0, 1) & (\text{Mischgebiet, Stadtrand}) \end{cases} \quad (60)$$

Die Anzahl der Stockwerke in einer *Kleinstadt*:

$$num_{storeys} = \begin{cases} 1 + rand(0, 2) & (\text{Industriegebiet, *}) \\ 3 + rand(0, 3) & (\text{Gewerbegebiet, Stadtkern}) \\ 3 + rand(0, 2) & (\text{Gewerbegebiet, Stadtmantel}) \\ 2 + rand(0, 1) & (\text{Gewerbegebiet, Stadtrand}) \\ 2 + rand(0, 3) & (\text{Wohngebiet, Stadtkern}) \\ 2 + rand(0, 2) & (\text{Wohngebiet, Stadtmantel}) \\ 1 + rand(0, 1) & (\text{Wohngebiet, Stadtrand}) \\ 3 + rand(0, 2) & (\text{Mischgebiet, Stadtkern}) \\ 2 + rand(0, 2) & (\text{Mischgebiet, Stadtmantel}) \\ 2 + rand(0, 1) & (\text{Mischgebiet, Stadtrand}) \end{cases} \quad (61)$$

2. Danach kann die Anzahl der Stockwerke *pro Flächennutzungstyp* ( $num_{storeys_i}$ ) bestimmt werden, d.h. dass z.B. ein sechsstöckiges Gebäude mit 50% Wohn- und 50% Büroanteil, 3 übereinander liegende Etagen mit Wohnungen und 3 Etagen mit Büroräumen besitzt. Sinnvollerweise wird hier die maximale Anzahl von Geschäfts- und Gastronomiestockwerken (in Mischgebäuden) begrenzt und festgelegt, dass pro Gebäude nur einer der beiden Flächennutzungstypen verwendet wird. Mit folgenden nacheinander ausgeführten Berechnungen werden die einzelnen Stockwerksanzahlen *pro Flächennutzungstyp* bestimmt (mit  $FN_i$

für die einzelnen Flächennutzungsstypen):

$$\begin{aligned}
FN_{total} &= \sum_{i=0}^{n-1} FN_i \\
num_{storeys_{Einzelhandel}} &= \lfloor \min(num_{storeys} * \frac{FN_{Einzelhandel}}{FN_{total}}, 2) + 0.5 \rfloor \\
num_{storeys_{Gastronomie}} &= \lfloor \min(num_{storeys} * \frac{FN_{Gastronomie}}{FN_{total}}, 1) + 0.5 \rfloor \\
FN_{total} &= FN_{total} - FN_{Einzelhandel} - FN_{Gastronomie} \\
num_{storeys_{Einzelhandel}} &= 0 \Leftrightarrow FN_{Einzelhandel} \leq FN_{Gastronomie} \\
num_{storeys_{Gastronomie}} &= 0 \Leftrightarrow FN_{Gastronomie} < FN_{Einzelhandel} \\
num_{storeys} &= num_{storeys} - num_{storeys_{Einzelhandel}} \\
num_{storeys} &= num_{storeys} - num_{storeys_{Gastronomie}} \\
\\
num_{storeys_{Industrie}} &= \lfloor num_{storeys} * \frac{FN_{Industrie}}{FN_{total}} + 0.5 \rfloor \\
FN_{total} &= FN_{total} - FN_{Industrie} \\
num_{storeys} &= num_{storeys} - num_{storeys_{Industrie}} \\
\\
num_{storeys_{Büro}} &= \lfloor num_{storeys} * \frac{FN_{Büro}}{FN_{total}} + 0.5 \rfloor \\
FN_{total} &= FN_{total} - FN_{Büro} \\
num_{storeys} &= num_{storeys} - num_{storeys_{Büro}} \\
\\
num_{storeys_{Wohnen}} &= \lfloor num_{storeys} * \frac{FN_{Wohnen}}{FN_{total}} + 0.5 \rfloor \quad (62)
\end{aligned}$$

Bei diesen Berechnungsschritten werden nach jedem Zwischenergebnis die jeweilige Stockwerksanzahl von der Gesamtmenge der Stockwerke und der zugehörige Flächennutzungstyp von der gesamten Flächennutzung abgezogen, um sonst zwangsläufig entstehende Rundungsfehler zu vermeiden.

3. Nachdem die Anzahl der funktional verschiedenen Stockwerke bestimmt wurde, werden aus diesen Werten die *Parameter der einzelnen Stockwerke* abgeleitet. Dabei werden die (vordefinierten) Parameter der einzelnen Unterobjekte nach ihrer Flächennutzung in folgender Reihenfolge von unten nach oben in das spätere Gebäude eingefügt (umgesetzt wird dieses Verfahren in einer Schleife über alle Stockwerke):

- (a) Einzelhandel, Gastronomie oder (ausschließlich) Industrie
- (b) Büro
- (c) Wohnen

Dabei wird jedem Stockwerk folglich nur ein stadtgeographischer Flächennutzungstyp als *funktionaler Stockwerktyp* zugewiesen.

Besitzt ein Gebäude z.B. zwei Stockwerke für Wohnungen und eins für Gastronomieflächen, werden dem Erdgeschoss die Parameter für Gastronomie und den oberen beiden Stockwerken die Parameter für Wohnungen zugewiesen.

Damit die Parameter-Liste des Gebäude-Objekts gefüllt werden kann, müssen für sämtliche funktionale Stockwerkstypen vordefinierte Parametersätze vorhanden sein (siehe Kapitel 6.6.1). Die Parameter der Stockwerkstypen werden hier zusätzlich in Erd- und Obergeschosse unterteilt, da das Erdgeschoss eines Gebäudes i.A. eine gesonderte Gestaltung besitzt (siehe Kapitel 3.4).

4. Abschließend werden die *Parameter des Dachs* berechnet, wobei der Dachtyp wiederum in Abhängigkeit der stadtgeographischen Parameter bestimmt wird (mit  $\{x = rand(a, b) | x \in \mathbb{N} \wedge a \leq x \leq b\}$  und  $FN_i$  für die einzelnen Flächennutzungstypen):

$$roof_{type} = \begin{cases} Flachdach & FN_{Industrie} > 0\% \wedge rand(0, 1) = 0 \\ Satteldach & FN_{Industrie} > 0\% \wedge rand(0, 1) = 1 \\ Flachdach & FN_{Büro} > FN_{Wohnen} \\ Satteldach & sonst \end{cases} \quad (63)$$

Der Typ *Zeltdach* wird immer dann zugewiesen, falls das Gebäude ein *Satteldach* besitzt und seine Grundfläche nicht durch vier Eckpunkte beschrieben werden kann.

Die restlichen Dach-Parameter werden wie folgt bestimmt (mit  $\{x = rand(a, b) | x \in \mathbb{R} \wedge a \leq x \leq b\}$ ):

$$\begin{aligned} roof_{height} &= rand(2.5, 3.5) \\ roof_{width} &= 0.15 \\ roof_{overlap_{gable}} &= rand(0.1, 0.25) \\ roof_{overlap_{eaves}} &= rand(1, 1.5) \end{aligned} \quad (64)$$

## 6.5.2 Allgemeine stadtgeographische Parameter von Unterobjekten

Für jedes Stockwerk eines Gebäudes wurde bereits im vorigen Kapitel, bei der Zuweisung der Stockwerk-Parameter, ein funktionaler Typ (Flächennutzungstyp) zugewiesen, der die allgemeinen stadtgeographischen Parameter eines solchen Unterobjekts *ersetzt*. Damit entfällt die Berechnung dieser Parameter auf der Gebäude-Stufe.

## 6.6 Stockwerk

Ein Stockwerk wird hier immer als Unterobjekt eines Gebäudes erstellt und seine Flächenausmaße somit durch dessen Grundfläche festgelegt. Für eine möglichst realistische Aufriss-Gestaltung besitzt jedes Stockwerk-Objekt einen

eigenen SLG<sup>79</sup> bzw. Grundriss, sowie eine Menge von objektspezifischen Parametern. Die stadtgeographischen Parameter werden durch einen einzelnen Flächennutzungstyp repräsentiert, da ein Stockwerk auf genau eine Art (industriell, gewerblich für Büros, gewerblich für Einzelhandel, gewerblich für Gastronomie oder zum Wohnen) genutzt wird und keine weiteren stadtgeographischen Zusammenhänge benötigt werden.

In den folgenden Unterkapiteln werden insbesondere die spezifischen Stockwerk-Parameter definiert und beschrieben.

### 6.6.1 Parameter und vordefinierte Werte eines Stockwerks

Für ein *Stockwerk*-Objekt werden die Grundriss-Parameter des Gebäudes übernommen, die ausschließlich für die Stockwerke bestimmt wurden. Die allgemeinen stadtgeographischen Parameter werden, wie weiter oben bereits beschrieben wurde, durch einen einzelnen objektspezifischen Parameter abgedeckt. Folglich werden hier nur die objektspezifischen Parameter und ihre vordefinierten Werte behandelt:

#### Objektspezifische Parameter

Für ein Stockwerk werden folgende spezifische Parameter definiert:

- **Etagennummer und Parametervererbung**  
Die Etagennummer wird für jedes Stockwerk nach seiner vertikalen Position im Gebäude festgelegt (0 für das Erdgeschoss, 1 für das erste Obergeschoss usw.).  
Um die Parameterzuweisung von Stockwerken mit gleicher Flächennutzung in einem Gebäude zu vereinfachen, kann zusätzlich ein Stockwerk als Parameter-Referenz angegeben werden. In diesem Fall werden die restlichen objektspezifischen Parameter von diesem Referenz-Stockwerk übernommen.
- **Funktionaler Typ** [ *Industrie* | *Büro* | *Wohnen* | *Einzelhandel* | *Gastronomie* ]  
Dieser Wert legt die Flächennutzung des Stockwerks fest und beeinflusst damit maßgeblich die restlichen Parameter, sowie die Texturauswahl bei der Aufriss-Generierung.
- **Höhe**  
Dieser Parameter gibt die Stockwerkshöhe an.  
In diesem Verfahren bekommen Stockwerke von Industrie- und Bürogebäuden im Vergleich zu Wohngebäuden größere Höhen zugewiesen. Bei Mischgebäuden (z.B. mit Geschäft im Erdgeschoss) wird dabei dem Erdgeschoss immer ein größerer Wert als den restlichen Obergeschossen zugewiesen.
- **Grund-Abstand**  
Dieser Parameter bezeichnet den vertikalen Abstand zwischen Stockwerk- und Erdgeschoss-Boden. Dieser Betrag ( $d_{ground_i}$ ) wird mit Hilfe der jeweiligen Stockwerkshöhe ( $h_i$ ) wie folgt berechnet (mit der Etagennummer als

---

<sup>79</sup>Jedes Stockwerk wird zwar durch die Grundriss-Parameter des zugehörigen Gebäudes auf die gleiche Art unterteilt, aber der entstehende Grundriss mit individuellen Parametern erweitert (Fenster, Türen usw. — siehe Kapitel 5.2).

Index):

$$d_{ground_i} = \sum_{j=0}^{i-1} h_j \quad (65)$$

- **Innenausbau** [ *Raum* | *Wohnung* | *Stockwerk* ]

Dieser Parameter bestimmt, auf welche Weise bei der späteren Aufriss-Generierung die Innenwände eines Stockwerks erstellt werden. Dabei können drei verschiedene Arten umgesetzt werden:

- **Raum**

Es werden alle Innenwände für die Wohnungen und Räume des Stockwerks erstellt.

- **Wohnung**

Es werden nur Innenwände auf den Grundriss-Kanten der Stockwerks-Teilflächen (Wohnungen) erstellt. Folglich werden Innenwände, die eine Wohnung in einzelne Räume aufteilen, nicht generiert.

- **Stockwerk**

Es werden nur Innenwände an den Rückseiten der Stockwerks-Außenwände erstellt. Die restliche Innenaufteilung des Stockwerks wird nicht beachtet.

Mit diesem Parameter kann die Raumaufteilung eines Stockwerks insbesondere für unterschiedliche Flächennutzungen variiert werden, was durch die notwendig einheitliche Grundriss-Unterteilung aller Stockwerke durch den SLG nicht möglich ist. Z.B. kann damit ein Gebäude im Erdgeschoss einen einzelnen großen Geschäftsraum und in den oberen Stockwerken mehrere Wohnungen mit Räumen besitzen.

- **Fenster-Parameter für Vorder- und Rückseite des Gebäudes**

Für die Grundriss-Erweiterung und Modell-Generierung in den Kapiteln 5.2 und 5.3.4 werden folgende Fenster-Parameter<sup>80</sup> definiert:

- **Fensterhöhe**

- **Fensterbankhöhe**

- **Breite**

- **Platzierungstyp** [ *einfach getrennt* | *mehrfach getrennt* | *mehrfach zusammenhängend* ]

Dieser Parameter bestimmt die Positionierung von Fenstern in einer Wand (siehe Kapitel 5.2.3).

- **Abstand zu Nachbarfenstern**

Für die Platzierung wird ggf. der Zwischenabstand von mehreren Fenstern in einem Wandstück benötigt.

- **Abstand zu Nachbarwand**

Für die Platzierung wird der Abstand zwischen Fenster und nächstem Wandstück benötigt.

---

<sup>80</sup>In der zugehörigen Tabelle als 9-Tupel: (*Fensterhöhe, Fensterbankhöhe, Breite, Platzierungstyp, Abstand zu Nachbarfenstern, Abstand zu Nachbarwand, Fensterglas vorhanden, Vordach vorhanden, Tiefe des Vordachs*).

- **Fensterglas vorhanden** [ *ja* | *nein* ]
- **Vordach vorhanden** [ *ja* | *nein* ]
- **Tiefe des Vordachs**

In diesem Verfahren können für Fenster an der Gebäude-Vorderseite und -Rückseite unterschiedliche Parameter angegeben werden. Z.B. haben Geschäfte nur an ihrer Straßenseite große Schaufenster und kleinere normale Fenster an den restlichen Seiten.

- **Eingangs-Parameter für Gebäude-, Außen- und Inneneingänge**  
Wie bei Fenstern, werden auch für die Eingänge eines Stockwerks folgende Parameter<sup>81</sup> definiert:

- **Türhöhe**
- **Breite**
- **Platzierungsausrichtung** [ *links* | *mitte* | *rechts* ]  
Dieser Parameter bestimmt, ob die Tür mittig oder an der linken/rechten Seite des Wandstücks eingefügt wird (siehe Kapitel 5.2.2).
- **Tür vorhanden** [ *ja* | *nein* ]
- **Vordach vorhanden** [ *ja* | *nein* ]
- **Tiefe des Vordachs**

Für ein Stockwerk können hier drei verschiedene Eingänge spezifiziert werden:

- *Gebäudeeingang* (für Haustür)
- *Inneneingang* (für Wohnungs- und Zimmertür)
- *Außeneingang*  
Ein Außeneingang wird wie der Gebäudeeingang nur in das Erdgeschoss eingefügt. Dort kann er, insbesondere für Geschäftsräume sinnvoll, wie ein Fenster an einer beliebigen Außen-Kante des SLG platziert werden.

Die Parameterwerte eines Erdgeschosses mit mehreren Geschäften werden wie in Tabelle 15 beispielhaft zugewiesen.

## 6.7 Wohnung

Das Wohnungs-Objekt stellt die niedrigste Hierarchiestufe und damit die letzte behandelte Ebene des Gesamtverfahrens dar. Dieses Objekt ist immer Unterobjekt bzw. Bestandteil eines Stockwerks, wobei seine Ausmaße durch die zugehörige Stockwerk-Teilfläche bestimmt werden. Bei der Unterteilung einer Wohnung entstehen Teilflächen, die den Wohnungsflur sowie die einzelnen Räume repräsentieren.

Eine Wohnung wird hier nicht weiter behandelt und besitzt somit keine objekt-spezifischen und allgemeinen stadtgeographischen Parameter. Eine Erweiterung

---

<sup>81</sup>In der zugehörigen Tabelle als 6-Tupel: (*Türhöhe, Breite, Platzierungsausrichtung, Tür vorhanden, Vordach vorhanden, Tiefe des Vordachs*).

Parametername	vordefinierter Wert
Etagennummer	0
Funktionaler Typ	Einzelhandel
Höhe	3m
Grund-Abstand	0
Innenausbau	Wohnung
Fenster	(2.5m,0m,2m, mehrfach zusammenhängend, 0m, 0.5m, ja), (1.5m,0.5m,1.2m, mehrfach getrennt, 0.5m, 0.5m, ja, ja, 1.7m)
Eingänge (Gebäudeeingang, Inneneingang, Au- ßeneingang)	(2.3m,1.8m, mitte, ja), (2.1m,1.5m, mitte, ja), (2.5m,1.7m, links, ja, ja, 1.2m)

Tabelle 15: Objektspezifische Parameter eines Stockwerk-Objekts.

hinsichtlich der funktionalen Raumnutzung ist hier aber denkbar und prinzipiell möglich. Damit können den einzelnen Zimmern einer Wohnung ihre verschiedenen Funktionen, wie z.B. Küche, Bad oder Schlafzimmer, zugewiesen werden und ggf. in die Grundriss- und Aufriss-Gestaltung mit einfließen (z.B. in Form von Inneneinrichtungen).

### 6.7.1 Parameter und vordefinierte Werte einer Wohnung

Für ein *Wohnung*-Objekt werden die folgenden Parameter und vordefinierten Werte verwendet:

Parametername	vordefinierter Wert
Anzahl der Begrenzungs-Knoten	wird berechnet
Typ der Berechnungsvorschrift für die Anzahl der Begrenzungs-Knoten	Wachstumsfunktion
Durchschnittlicher Flächeninhalt einer Teilfläche	10m <sup>2</sup>
Seitenverhältnis der Teilflächen	1 : 3
Skalierungsfaktor und Exponent für Wachstumsfunktion	1; 3
Minimale Anzahl von Begrenzungs-Knoten	3
Positions-Abweichung der Begrenzungs-Knoten	0.05
Typ der Zwischenräume	Wohnungsflur
Breite der Zwischenräume	1.5m
Start-Knoten für Zwischenräume	wird berechnet
Kürzeste und längste Kante	wird berechnet
Maximale Anzahl von überstumpfen Winkeln für Pfade am Außenrand, Außenrand mit Ecke und Außenrand mit konkaver Ecke	1; 1; 0

Tabelle 16: Grundriss-Parameter eines Wohnungs-Objekts.

#### Grundriss-Parameter

Wie bei einem Stockwerk-Objekt, wird die Anzahl der Teilflächen einer Wohnung durch eine Wachstumsfunktion berechnet. Dabei steigt die Anzahl der Zimmer

nur langsam an und wird sinnvoll zwischen ihren Ober- und Untergrenzen<sup>82</sup> verteilt. Die Grundriss-Parameter können wie in Tabelle 16 beispielhaft zugewiesen werden.

---

<sup>82</sup>Die Untergrenze ist hier die *minimale Anzahl von Begrenzungs-Knoten* und die Obergrenze der Wert gegen den die Wachstumsfunktion konvergiert (mit einem Faktor 1.0 und Exponenten von 3.0 ist das Maximum bei  $\approx 7$  erreicht).

## 7 Praktische Umsetzung: CITYLIFE

Im Rahmen dieser Arbeit wurde das prozedural-unterstützte Modellierungsprogramm CITYLIFE entwickelt. Mit Hilfe dieses Programms kann ein Benutzer 3D-Modelle von einzelnen begehbaren Gebäuden, aber auch komplexen städtischen Gebieten nahezu automatisch, d.h. mit nur wenigen Interaktionsschritten, erstellen. Mit der integrierten 3D-Echtzeit-Visualisierung können die generierten Modelle in ihrer einsetzbaren Form und Gestaltung begutachtet sowie über zusätzlich angezeigte Bedienelemente ihre Eigenschaften bzw. Parameter durch den Benutzer angepasst werden.

Implementiert wurde CITYLIFE in den objektorientierten Programmiersprachen C++ und Managed C++ (*Microsoft® .NET Framework*). Ausführbar ist das Programm auf dem Betriebssystem *Microsoft Windows® 2000/XP* und wurde u.a. auf einem Testsystem mit *Intel® Celeron® 2.8GHz*, *1GB RAM* und *ATI Radeon™ X800XT* verwendet.

Aufgrund der hier behandelten, komplexen Thematik sowie der Berücksichtigung aller entwickelten Verfahren besteht CITYLIFE aus 20000 Zeilen Programmcode.

In den folgenden Unterkapiteln werden Konzept, Aufbau, Funktionalitäten und die Bedienung von CITYLIFE behandelt und dabei beschrieben, wie die hier entwickelten Verfahren von Stadtgrundriss und -aufriss umgesetzt wurden.

Die 3D-Echtzeit-Visualisierung wurde auf Basis der OGRE-Engine [OGRE] entwickelt, die in einem eigenen Unterkapitel vorgestellt und hinsichtlich Verwendung und Funktionalitäten bewertet wird.

### 7.1 Konzept, Aufbau und Funktionalitäten

Die in Kapitel 4 und 5 beschriebenen Verfahren zur Generierung von Stadtgrundriss und -aufriss wurden speziell für eine spätere Implementierung entwickelt und ihre Komplexität (gleichzusetzen mit der Menge des Programm-Codes) an den Zeitrahmen dieser Arbeit angepasst. Dazu zählt auch die Hierarchie der einzelnen Stadtobjekte gemäß Kapitel 6, die durch eine entsprechende Struktur ebenfalls direkt umgesetzt wird.

Das Programm CITYLIFE<sup>83</sup> ist allgemein in einzelne (ggf. austauschbare) Module<sup>84</sup> unterteilt. Alle implementierten Module können in drei informationstechnisch unabhängige Hauptbereiche eingeordnet werden:

#### 1. CLEngine

Alle hier enthaltenen Module bzw. Klassen behandeln die entwickelten Verfahren für den Stadtgrundriss und -aufriss. Folglich werden hier eine erweiterte Graphenstruktur — der SLG — mit sämtlichen Funktionalitäten für die Flächenunterteilung, alle geometrischen Basis-Elemente, die Module für 3D-Meshs und -Modelle sowie alle Stadtobjekte mit entsprechenden Parametern implementiert.

---

<sup>83</sup>Im Folgenden als Prefix *CL* verwendet.

<sup>84</sup>Der Begriff Modul kann in der objektorientierten Programmierung für einzelne oder semantisch zusammenhängende Klassen verwendet werden.

Die CLEngine ist somit der programmiertechnisch anspruchsvollste und größte Programmteil von CITYLIFE. Alle hier implementierten Module sind in C++ geschrieben und wurden hochgradig objektorientiert umgesetzt.

## 2. CLEditor

Der CLEditor implementiert die grafische Benutzeroberfläche (GUI) und verarbeitet die Eingaben des Benutzers. Dieser Teil ist in Managed C++ geschrieben und basiert auf den *Windows®Forms*-Bibliotheken. Damit kann dieser Programmteil nur auf dem Betriebssystem Windows 2000/XP ausgeführt werden. Für die Schnittstellen zu den Modulen CLEngine und CLGraphics, bzw. für den Datenaustausch von C++ und Managed C++, werden zusätzliche Typkonvertierungen benötigt und im CLEditor umgesetzt.

## 3. CLGraphics

Dieser Teil von CITYLIFE ist ausschließlich für die dreidimensionale Darstellung der Stadtszene mit ihren Grundrissen und Modellen zuständig. Als Middleware wird hier das OGRE-SDK<sup>85</sup> verwendet um die komplexe Steuerung einer Grafik-API<sup>86</sup> zu vereinfachen [OGRE]. Die Module von CLGraphics werden hauptsächlich zur Erweiterung und Anpassung der OGRE-Bibliotheken verwendet.

Alle Klassen von CLGraphics und die OGRE-Bibliotheken sind in betriebs-systemunabhängigen C++ geschrieben.

Der detaillierte **Aufbau** von CITYLIFE mit allen verwendeten Klassen wird durch ein Klassendiagramm in Abb. 50 veranschaulicht und im Folgenden kurz erläutert:

- **CLEngine** (Initialisierung und Verwaltung aller Stadtobjekte)
  - **CLSLSGraph** (SLG mit Flächenunterteilung und Strukturen zur Speicherung seiner Bestandteile)
    - \* **CLNode** (Knoten-Objekt)
    - \* **CLEdge** (Kanten-Objekt)
    - \* **CLPath** (Pfad-Objekt)
    - \* **CLArea** (Teilflächen-Objekt)
  - **CLObject** (Basisklasse für alle Stadtobjekte)
    - \* **CLCity** (Objekt für eine Stadt, Ausführung der zugehörigen, spezifischen Grundriss- und Aufriss-Erstellung, abgeleitet von CLObject)
    - \* **CLDistrict** (Objekt für einen Stadtteil, vgl. CLCity)
    - \* **CLBlock** (Objekt für einen Hausblock, vgl. CLCity)
    - \* **CLLot** (Objekt für ein Grundstück, vgl. CLCity)
    - \* **CLBuilding** (Objekt für ein Gebäude, vgl. CLCity)
    - \* **CLStorey** (Objekt für ein Stockwerk, vgl. CLCity)
    - \* **CLApartment** (Objekt für eine Wohnung, vgl. CLCity)

---

<sup>85</sup>Ein Software Development Kit (SDK) einer bestimmten Software ist eine Ansammlung von Programmen, Bibliotheken oder Klassen sowie einer Dokumentation, um basierend auf dieser Software weitere Anwendungen zu erstellen.

<sup>86</sup>DirectX® oder OpenGL.

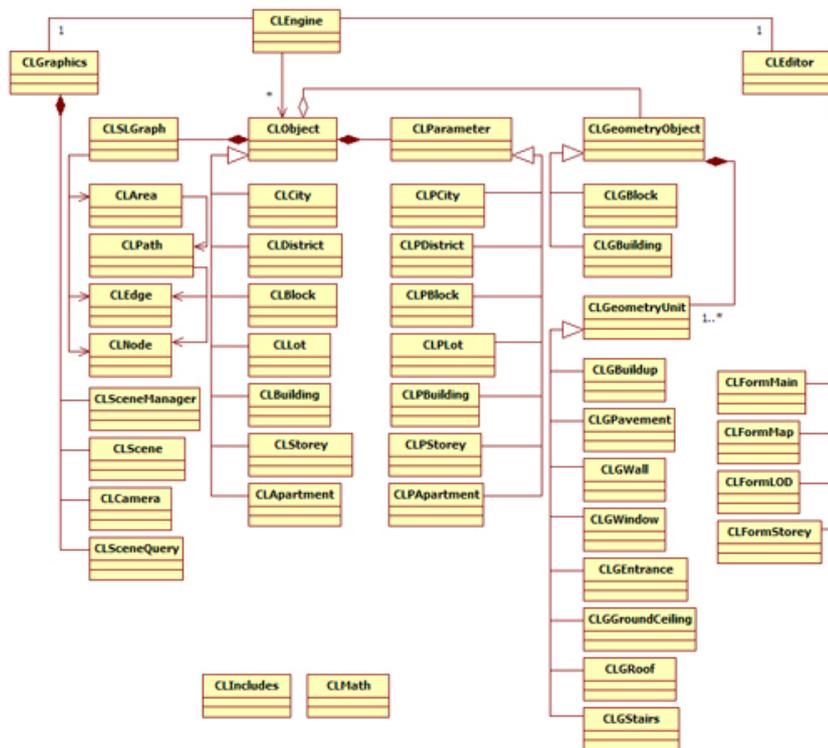


Abbildung 50: UML-Klassendiagramm von CITYLIFE.

- **CLParameter** (Basisklasse für alle Parameter der Stadtobjekte)
  - \* **CLPCity** (Parameter-Objekt für eine Stadt zur Speicherung und Berechnung der eigenen und Unterobjekt-Parameter, abgeleitet von CLParameter)
  - \* **CLPDistrict** (Parameter-Objekt für einen Stadtteil, vgl. CLPCity)
  - \* **CLPDistrict** (Parameter-Objekt für einen Stadtteil, vgl. CLPCity)
  - \* **CLPBlock** (Parameter-Objekt für einen Hausblock, vgl. CLPCity)
  - \* **CLPlot** (Parameter-Objekt für ein Grundstück, vgl. CLPCity)
  - \* **CLPBuilding** (Parameter-Objekt für ein Gebäude, vgl. CLPCity)
  - \* **CLPStorey** (Parameter-Objekt für ein Stockwerk, vgl. CLPCity)
  - \* **CLPApartment** (Parameter-Objekt für eine Wohnung, vgl. CLP-City)
- **CLGeometryObject** (Basisklasse zum Erstellen von 3D-Meshs und -Modellen)
  - \* **CLGBlock** (Steuerung der Erstellung eines Hausblock-Modells mit den geometrischen Basis-Elementen CLGPavement und CLGBuildup, abgeleitet von CLGeometryObject)
  - \* **CLGBuilding** (Steuerung der Erstellung eines Gebäude-Modells mit den geometrischen Basis-Elementen CLGWall, CLGEntrance, CLGWindow, CLGGroundCeiling, CLGRoof und CLGStairs, abgeleitet von CLGeometryObject)

- **CLGeometryUnit** (Basisklasse der geometrischen Basis-Elemente)
  - \* **CLGBuildup** (geschlossene Hausblock- bzw. Grundstücks-Oberfläche, abgeleitet von CLGeometryUnit)
  - \* **CLGPavement** (Gehwege an Straßenrändern, abgeleitet von CLGeometryUnit)
  - \* **CLGWall** (Innen- und Außenwände eines Gebäudes, abgeleitet von CLGeometryUnit)
  - \* **CLGEntrance** (Innen- und Außen-Eingänge, abgeleitet von CLGeometryUnit)
  - \* **CLGWindow** (Fenster, abgeleitet von CLGeometryUnit)
  - \* **CLGGroundCeiling** (Decken und Böden, abgeleitet von CLGeometryUnit)
  - \* **CLGRoof** (Sattel-, Zelt- und Flachdach, abgeleitet von CLGeometryUnit)
  - \* **CLGStairs** (vereinfachte Treppenhaus-Geometrien, abgeleitet von CLGeometryUnit)
- **CLEditor** (Steuerung und Aufbau der Formulare<sup>87</sup>, Auslesen und Zuweisen von Parametern sowie Typkonvertierungen)
  - **CLFormMain** (Hauptfenster)
  - **CLFormMap & CLFormParamsLOD & CLFormParamsStorey** (diverse Dialogfenster für Hintergrundbild, LOD- und Stockwerks-Parameter)
- **CLGraphics** (Erweiterung, Initialisierung und Steuerung der 3D-Grafik-Engine)
  - **CLCamera** (Blickrichtung und Navigation)
  - **CLScene** (3D-Umgebung mit Lichtquellen, Bodenfläche, Maßstabs-Gitter usw.)
  - **CLSceneManager** (Organisation und Anordnung aller 3D-Modelle)
  - **CLSceneQuery** (benötigte Verfahren für das 3D-Picking und Setzen von Umriss-Punkten)
- **CLMath** (ausgelagerte mathematische Funktionen)
- **CLIncludes & CLUtilities** (Einbindung benötigter Bibliotheken, Definition von statischen Strukturen und diverse Hilfsklassen)

Im Programm CITYLIFE wurden die folgenden **Funktionalitäten** integriert und umgesetzt:

- Zur vollständigen Steuerung und übersichtlichen Verwendung des Programms werden die einzelnen Bedienungs- und Informationselemente intuitiv und übersichtlich angeordnet bzw. angezeigt.

---

<sup>87</sup>Programmfenster, GUI.

- Der Benutzer kann per Maus und Tastatur in der 3D-Szene frei navigieren. Dafür werden zwei verschiedene Kamera-Steuerungen — *Modell*<sup>88</sup> und *FirstPerson*<sup>89</sup> — angeboten.
- Die manuelle Erstellung der Stadtobjekte Stadt, Stadtteil, Hausblock und Gebäude ist möglich. Dabei können die jeweiligen Umrisse durch Mausclicks im 3D-Fenster definiert werden. Alle weiteren Stadtobjekte werden im Zuge der Grundriss-Generierung automatisch erzeugt.
- Die Grundrisse und 3D-Modelle aller Stadtobjekte können automatisch generiert werden. Dabei kann zusätzlich die Stufentiefe der Hierarchie gesteuert werden. Z.B. kann eine Grundriss-Generierung nur in den Ebenen Stadtteil bis Hausblock oder ausschließlich auf Wohnungs-Ebene durchgeführt werden. Damit können unpassende Grundrisse neu erstellt werden.
- Die Grundriss- und Aufriss-Generierung kann durch eine Vielzahl von Parametern beeinflusst werden, wobei in den meisten Fällen die einmalige Angabe der stadtgeographischen Parameter ausreicht, um ansprechende Grundrisse und/ oder Modelle zu generieren (siehe Kapitel 6).
- Alle Grundrisse und 3D-Modelle werden in Echtzeit angezeigt. Die 3D-Modelle werden dabei texturiert, korrekt beleuchtet und mit Schatten dargestellt. Dabei können auch semi-transparente Texturen, z.B. für Fenster, verwendet werden.
- Alle 3D-Modelle werden für die performante Echtzeit-Darstellung in verschiedenen Detailtiefen bzw. LOD-Stufen generiert und angezeigt.
- Die entstandenen 3D-Modelle können für eine spätere Verwendung in anderen Echtzeit-Simulationen als OGRE-XML-Dateien exportiert werden. Für eine anschließende Format-Konvertierung kann z.B. das Modellierungsprogramm *Blender* benutzt werden.

## 7.2 Benutzeroberfläche und Bedienung

Die **Benutzeroberfläche** von CITYLIFE wurde mit den *Windows® Forms*-Bibliotheken des zugehörigen *Windows® .Net Framework* implementiert. Die gesamte Programmoberfläche kann in vier Hauptbereiche eingeteilt werden (vgl. Abb. 51):

- (A) Das Hauptmenü mit Funktionen für den Export, Anzeigeeoptionen (Wireframe, Nebel usw.) und dem Laden von Hintergrundbildern<sup>90</sup>.
- (B) Die hierarchische Anzeige aller vorhandenen Stadtobjekte.
- (C) Bedienungselemente für die Grundriss- und Aufriss-Generierung sowie die vollständige Parametersteuerung.

---

<sup>88</sup>Die Kamera bewegt sich auf einer Kugel um einen zentralen in der Ebene gelegenen Punkt. Vergleichbar ist dieser Modus mit der Kamera-Steuerung in einem Modellierungsprogramm.

<sup>89</sup>Die Kamera ist an die Steuerung von *First-Person-Shooter*-Computerspielen angelehnt.

<sup>90</sup>Dafür wird ein weiteres Dialogfenster geöffnet und eine beliebige Bilddatei kann als (ebener) Hintergrund geladen werden.

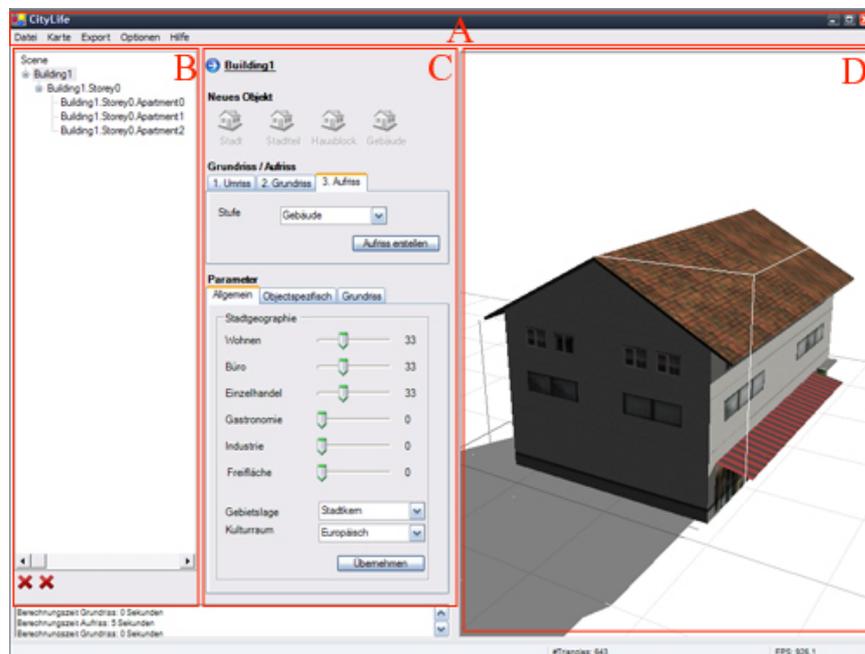


Abbildung 51: Die Benutzeroberfläche von CITYLIFE.

(D) Der 3D-Anzeigebereich in Echtzeit.

Die Erstellung von Stadt- und Gebäudemodellen wird in CITYLIFE durch drei aufeinander folgende, benutzergesteuerte Schritte eingeteilt:

1. Umriss der Objektfläche definieren.
2. Objekt-Grundriss (Flächenunterteilung) erstellen.
3. Objekt-Aufriss (3D-Modelle) erstellen.

Dieses **Bedienungs-Konzept** von CITYLIFE soll am Beispiel der Generierung eines vollständigen Hausblock-Modells (mit einzelnen Gebäuden) veranschaulicht werden (die Bedienelemente sind nach Abb. 51 beschriftet.):

1. Nach dem Starten des Programms wird die Benutzeroberfläche mit 3D-Anzeigebereich angezeigt und eine leere Stadtszene initialisiert.
2. Durch Klicken auf den Button 'Hausblock' im mittleren Bereich (C) wird ein initiales Hausblock-Objekt erstellt und im linken Bereich in der Hierarchieansicht (B) angezeigt.
3. Nachdem dieses Objekt ausgewählt wurde<sup>91</sup>, können seine initialen Parameterwerte in den Kategorien *Allgemein*<sup>92</sup>, *Objektspezifisch* und *Grundriss* geprüft und ggf. angepasst werden (C).

<sup>91</sup>Über die hierarchische Baumansicht (B). Hat das Objekt einen Grund- oder Aufriss, kann es auch direkt im 3D-Anzeigebereich ausgewählt werden (D).

<sup>92</sup>Hier befinden sich u.a. die stadtgeographischen Parameter gemäß Kapitel 6

4. Anschließend kann die Gesamtfläche des Grundrisses definiert werden, wofür die Eingabe der Eckpunkte über den Button *Start* (Reiter: *1. Umriss*) gestartet wird (C). Der Benutzer kann nun im 3D-Anzeigebereich den Umriss der Hausblock-Fläche, ggf. auf Grundlage einer vorher geladenen Stadt- bzw. Gebietskarte, mit Mausklicks bestimmen (D). Wurden alle Eck-Punkte platziert, wird dieser Prozess über den Button *Ende* (Reiter: *1. Umriss*) beendet (C).
5. Nun kann die Flächenunterteilung des SLG mit dem Button *Grundriss erstellen* (Reiter: *2. Grundriss*) gestartet werden. Für eine vollständige Unterteilung bis auf Wohnungs-Ebene muss aus der Auswahlliste *Stufen* der Eintrag *Räume* selektiert werden (C).
6. Wurde der Grundriss vollständig unterteilt und (ggf. durch erneutes Erstellen) in die gewünschte Form gebracht, können die 3D-Modelle dieses Stadtobjekts über den Button *Aufriss erstellen* (Reiter: *3. Aufriss*) generiert werden (C).

Mit Schritt 6 ist die vollständige Generierung eines Hausblocks mit seinen (begehbaren) Gebäudemodellen abgeschlossen und die Ergebnisse können im 3D-Anzeigebereich interaktiv betrachtet werden. Selbst ein unerfahrener Benutzer kann so in nur einigen Sekunden 10-20 komplexe urbane 3D-Modelle für Echtzeit-Simulationen erstellen. Für mehrere bzw. größere Modelle steigt dabei nicht zwangsläufig die Interaktionsdauer, sondern nur die Berechnungszeit von CITYLIFE.

Für größere Stadtgebilde werden die oben aufgeführten Schritte ab dem Objekt Stadt oder Stadtteil ausgeführt. Dabei wird es dem Benutzer überlassen, ob z.B. die Stadtteile in einem Stadtgebiet automatisch angelegt oder manuell definiert werden. Im letzteren Fall kann dabei wie folgt vorgegangen werden:

1. Zunächst wird ein Stadtobjekt angelegt und seine Gesamtfläche mitsamt Parametern definiert (der Stadtmittelpunkt wird dabei automatisch berechnet).
2. Die Stadtteile werden nun aber nicht mit dem Grundriss-Verfahren erstellt, sondern nach den Vorstellungen des Benutzers als Unterobjekte der Stadt eingeben. Z.B. ein zentraler Stadtteil, der von weiteren umgeben ist.
3. Nun können über das Stadtobjekt Grund- und Aufriss aller Stadtteile erstellt werden. Dabei werden alle Parameter für die Unterobjekte in Abhängigkeit nach ihrer Lage und der vorgegebenen Stadt-Parameter berechnet. D.h., dass der zentrale Stadtteil im Normalfall den Stadtkern mit hohen Gebäuden repräsentiert und die restlichen Stadtteile am Stadtmantel und Stadtrand mit entsprechend niedrigeren Bauformen liegen. Zusätzlich wird die Verteilung der Flächennutzungstypen an Stadtkern, Stadtmantel und Stadtrand angepasst.

Des Weiteren kann hier die Flexibilität der Stadtobjekt-Hierarchie ausgenutzt werden, indem in ein Stadtobjekt auch Hausblöcke oder einzelne Gebäude als Unterobjekte eingefügt werden.

### 7.3 Verwendete 3D-Grafik-Engine

Für die Echtzeit-Darstellung von dreidimensionalen Szenen bzw. Objekten wird i.A. eine der beiden verbreiteten Grafik-APIs — DirectX® oder OpenGL — verwendet. Die Grafik-API stellt dabei eine Schnittstelle zur Grafikhardware bereit und kann in Form von Klassen(-Bibliotheken) oder Funktionen bei der 3D-Softwareentwicklung verwendet werden. Die meisten dabei bereitgestellten Funktionalitäten sind allerdings auf einem programmiertechnisch sehr niedrigem Level und müssen z.B. für die Grafik-Initialisierung oder das Laden und Anzeigen von 3D-Modellen zu komplexen nicht trivialen Modulen bzw. Codesegmenten zusammengesetzt werden. Der dabei entstehende Aufwand wird z.B. in der Arbeit von SCHRADER für die Grafik-API DirectX® behandelt [RoSc06].

Um den somit zusätzlich entstehenden Programmieraufwand zu reduzieren, wurde in dieser Arbeit die Open-Source<sup>93</sup> 3D-Grafik-Engine<sup>94</sup> OGRE in der Version 1.4.0 verwendet [OGRE]. Diese Engine stellt für den Entwickler komfortable Klassen und Funktionen für alle grundlegenden Problemstellungen bei der Darstellung von 3D-Szenen bereit. U.a. sind folgende Funktionalitäten integriert:

- Initialisierung der Grafik-API und ggf. Erstellung eines Anzeigefensters. Dabei kann entweder DirectX® oder OpenGL verwendet werden.
- Einfaches Laden und Anzeigen von (mitunter auch animierten) 3D-Modellen aus Dateien.
- Bereitstellung verschiedener Szenengraphen zur Strukturierung und Anordnung der Objekte.
- Aufwändiges Materialsystem zur Texturierung und Gestaltung von Oberflächen.
- Vorhandene Klassen für die manuelle Erstellung von 3D-Geometrien.

Zusätzlich werden noch weitere wichtige, allgemeine Eigenschaften benannt:

- Ständige Weiterentwicklung mit häufigen Releases.
- Vollständige und umfangreiche Dokumentation.
- Schneller und guter Support sowie ein große *Community*.
- Akzeptanz in der (kommerziellen) Computerspiel-Industrie.
- Die bei einer Grafik-Engine i.A. fehlenden Simulations-Bestandteile, wie z.B. KI, Scripting, Physik oder Benutzeroberfläche, können als Add-Ons (von Drittanbietern) hinzugefügt werden.

Das OGRE-SDK ist eine in C++ implementierte, performante Ansammlung von Klassenbibliotheken, die sich u.a. durch eine objektorientierte Struktur mit häufiger Verwendung der *Standard Template Library*<sup>95</sup> auszeichnet. Die einzelnen

<sup>93</sup>Die OGRE-Engine ist frei erhältlich und kann aufgrund der *GNU Lesser General Public License (LGPL)* auch in kommerziellen Software-Anwendungen verwendet werden.

<sup>94</sup>Eine *Grafik-Engine* unterstützt hauptsächlich die Darstellung von visuellen Daten und behandelt nicht, im Gegensatz zu einer *Gameengine*, die weiteren Themenbereiche der künstlichen Intelligenz (KI), Physik, Modell- und Level-Editoren, Scripting usw.

<sup>95</sup>Die für C++ standardisierte *Standard Template Library (STL)* ist eine Ansammlung von effizienten template-basierten Datenstrukturen, wie z.B. Zeichenketten (Strings), Vektoren, Listen oder Hashmaps.

Module wurden so konzipiert, dass sie durch einen Entwickler<sup>96</sup> einfach erweitert und somit exakt auf die vorhandenen, spezifischen Problemstellungen angepasst werden können.

Für die Entwicklung einer allgemeinen 3D-Echtzeit-Anwendung kann daher die Verwendung der OGRE-Engine als Visualisierungsbasis empfohlen werden.

## 7.4 Ergebnisbilder und Leistungsverhalten

Die folgenden Bilder veranschaulichen den Konstruktionsprozess der morphogenetischen und städtebaulichen Elemente von der Erstellung der Umrissflächen, über die automatisch erzeugten Grundrisse bis hin zu den fertigen 3D-Modellen in verschiedenen LOD-Stufen. Dabei wurden die stadtgeographischen Parameter so eingestellt, dass Industrie-, Gewerbe-, Wohn- und Mischgebiete repräsentiert werden (dazu zählt auch die vertikale Nutzungsverteilung innerhalb von Gebäude), sowie Gebäudeplatzierungen und -ausmaße sich an der Lage im Stadtgebiet orientieren. Des Weiteren wird ein urbanes Gelände durch die manuelle Erstellung von Hausblöcken auf der Grundlage einer *Google-Earth*-Karte der Stadt Berlin gezeigt (vgl. Abb. 52 bis 58).

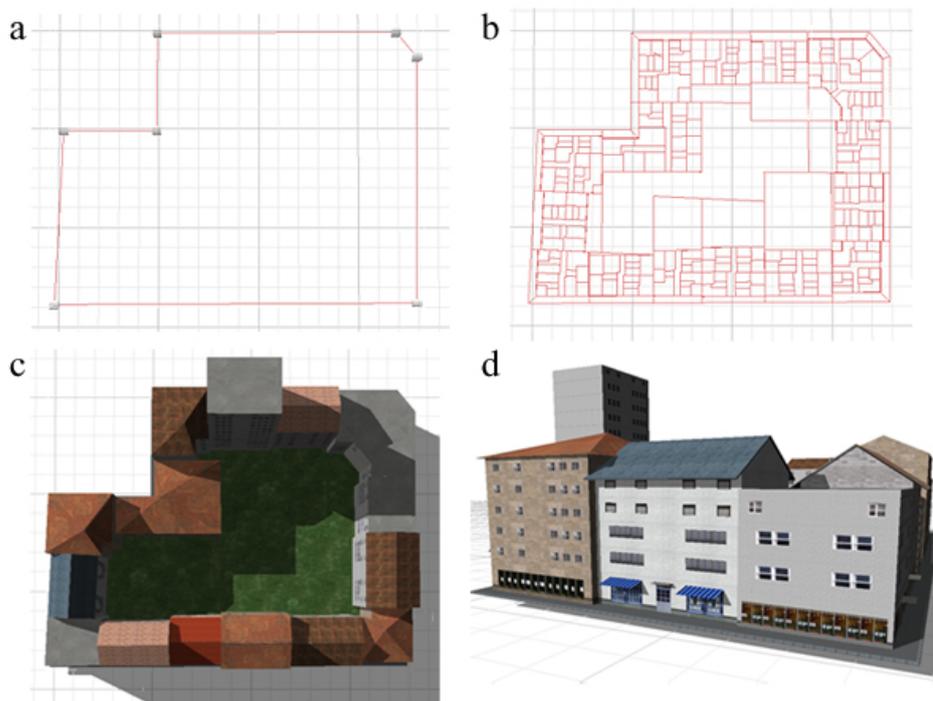


Abbildung 52: Konstruktionsprozess eines Hausblocks: a) Umriss-Eingabe, b) Grundriss-Generierung, c) Aufriss-Generierung, d) Interaktive Betrachtung der Ergebnisse.

<sup>96</sup>Fortgeschrittene Kenntnisse in der Programmiersprache C++ vorausgesetzt.



Abbildung 53: Ein Gebäudemodell in vier LOD-Stufen.



Abbildung 54: Verschiedene Gebäudetypen.



Abbildung 55: Innere Konstruktion zweier Gebäude (bzgl. der Raumaufteilung und Größe von Innenräumen in Stockwerken).

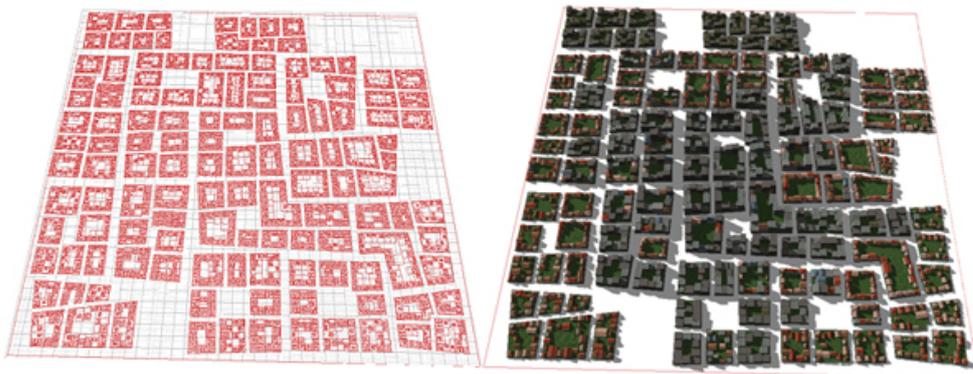


Abbildung 56: Grund- und Aufriss einer Stadt mit ca. 1500 Gebäuden.

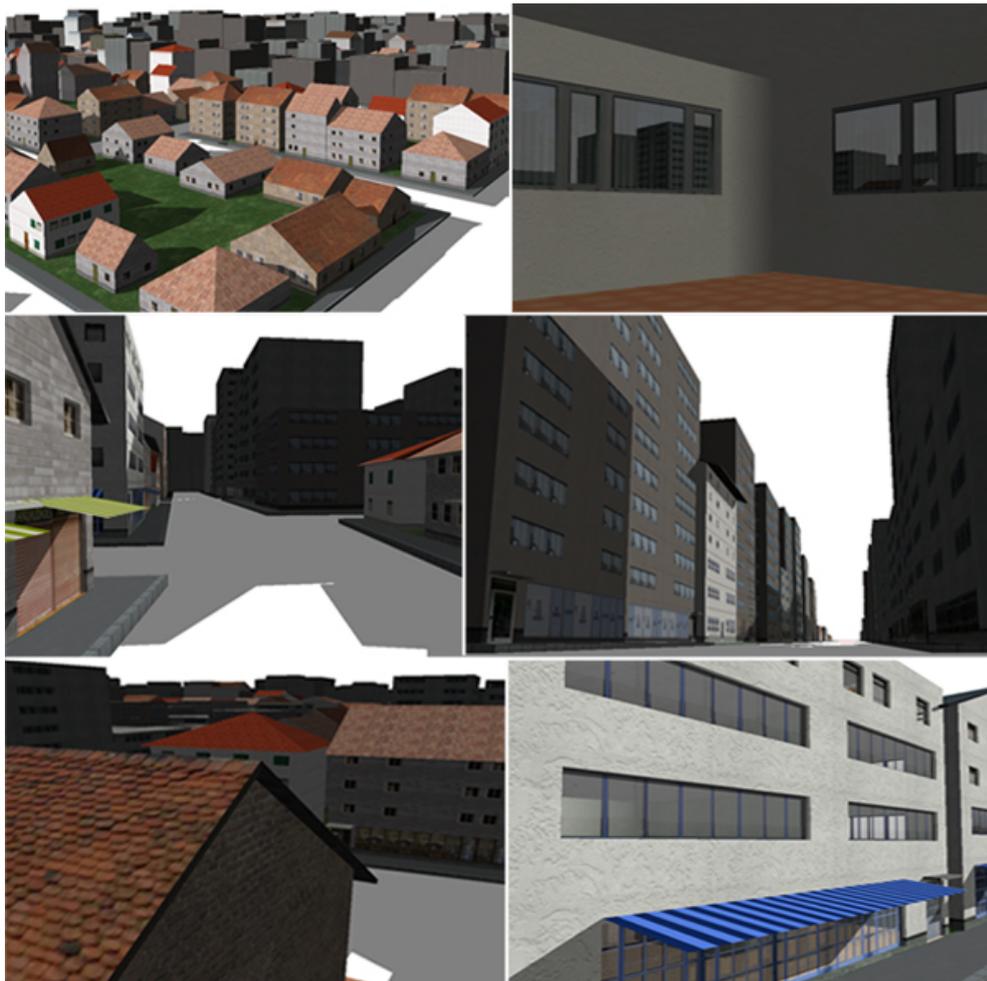


Abbildung 57: Verschiedene Betrachtungspositionen innerhalb eines großen Stadtmodells.

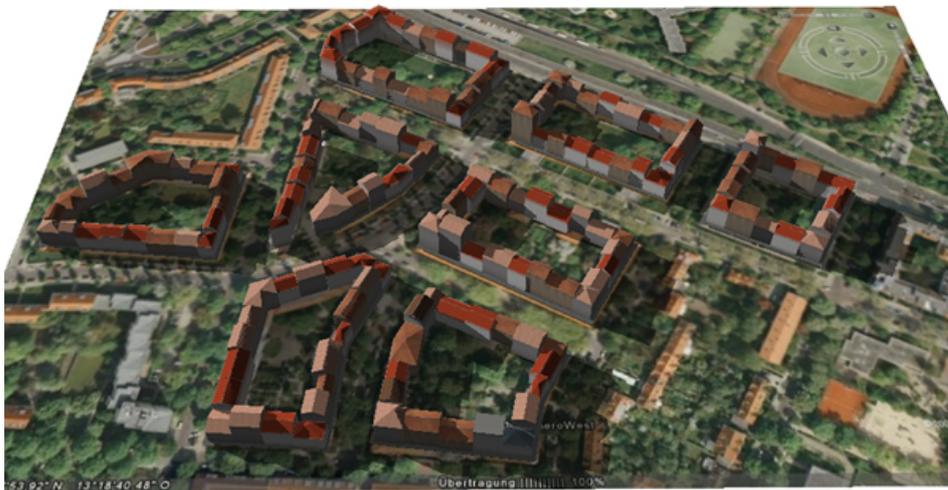


Abbildung 58: Generierter Grund- und Aufriss auf der Grundlage eines Luftbilds von Berlin aus Google Earth.

In Tabelle 17 wird durch die Angabe von Berechnungszeiten (*Zeit* in Sekunden) für die Konstruktion von unterschiedlichen Stadtobjekten das Leistungsverhalten von CITYLIFE bei der Grundriss- und Aufriss-Generierung aufgezeigt. Zusätzlich werden zur Vollständigkeit die Anzahl der durchschnittlichen *FPS* bei der Modell-Darstellung, sowie die minimale und maximale Anzahl der angezeigten *Polygone* (bzgl. der LOD-Stufen) angegeben.

Die Tests wurden auf einem System mit *Intel® Core™ 2 T2700 @ 2.00GHz*, *2GB RAM* und *NVIDIA GeForce Go 7700* ausgeführt. Alle dabei entstandenen Gebäudemodelle wurden mit *fünf* LOD-Stufen generiert (dabei werden sinnvollerweise nur für die LOD0-Meshs die Innenräume erstellt), durch eine Richtungs-Lichtquelle beleuchtet und mit Schatten<sup>97</sup> sowie semi-transparenten Texturen gerendert.

Testfall	Zeit (sek.)	FPS	Polygone
Grund- und Aufriss eines vierstöckigen Gebäudes	< 1	> 60	14 – 3000
Grund- und Aufriss eines Hausblocks mit 12 Gebäuden	5	> 60	60 – 10000
Grund- und Aufriss eines Stadtteil-Objekts mit 100 Gebäuden	20	> 60	3000 – 20000
Grund- und Aufriss eines Stadt-Objekts mit 1500 Gebäuden	950	10 – 60	60000 – 70000

Tabelle 17: Leistungsverhalten von CITYLIFE.

---

<sup>97</sup>Die Schatten werden hier mit relativ aufwändigen *Modulative Stencil Shadows* erzeugt. Da die Modelle nicht animiert sind, kann ein zusätzlicher Leistungszuwachs durch Texturbasierte Schatten (*Texture-based Shadows*) erreicht werden. Allerdings soll die CITYLIFE-Darstellung nur eine beispielhafte Testumgebung repräsentieren und weiterführende Optimierungen beim Rendering den Anwendungen überlassen werden, welche die Modelle im Nachhinein verwenden.

## 8 Gesamtfazit

In diesem abschließenden Kapitel werden die im Zeitrahmen einer Diplomarbeit entstandenen Ergebnisse zusammengefasst und bewertet, sowie weiterführende Konzepte und Ausblicke für zukünftige Arbeiten skizziert.

### 8.1 Zusammenfassung

Ein Stadtgebiet ist aufgrund seiner vielschichtigen horizontalen und vertikalen Strukturen, bedingt durch eine Vielzahl von ökonomischen, historischen und kulturellen Entwicklungsprozessen, ein hoch komplexes räumliches Gebilde. Die prozedurale Generierung realistischer, digitaler Nachbildungen dieser Formen stellt somit eine besondere Herausforderung für die notwendigerweise interdisziplinären Verfahren dar.

Im Rahmen dieser Arbeit ist aufgrund dieser Thematik ein Gesamtsystem entwickelt worden, das realistisch wirkende Stadt- und Gebäudemodelle speziell für 3D-Echtzeit-Anwendungen generiert.

Um die prozedurale Behandlung von horizontaler und vertikaler Struktur bzw. Grund- und Aufriss einer Stadt zu vereinfachen, wurde zunächst ihre räumliche Ausdehnung in verschiedene, hierarchisch angeordnete Elemente gegliedert:

1. Stadt
2. Stadtteil
3. Hausblock
4. Grundstück
5. Gebäude
6. Stockwerk
7. Wohnung (mit Räumen)

Damit konnte die Thematik sinnvoll in Teilprobleme zerlegt und die jeweiligen Besonderheiten einzeln bearbeitet werden. Alle aufgeführten Stadtobjekte sind an stadtgeographische und städtebauliche Untersuchungen bzw. Gesetzmäßigkeiten angelehnt, um zusätzlich in einem wissenschaftlichen bzw. realitätsnahen Zusammenhang betrachtet zu werden.

Für den Grundriss eines städtischen Gebiets wurde ein Verfahren entwickelt, dass

- die Fläche einer Stadt in Stadtteile,
- die Fläche eines Stadtteils in Hausblöcke,
- die Fläche eines Hausblocks in Grundstücke mit Gebäuden,
- die Fläche eines Gebäudes in Wohnungen und
- die Fläche einer Wohnung in Räume

iterativ unterteilt. Dabei wird insbesondere auf die realistische Grundriss-Erstellung von Hausblöcken, Grundstücke, Gebäuden und Wohnungen eingegangen. Das somit notwendige, informationstechnische Flächenunterteilungsverfahren wurde mit einer erweiterten Graphenstruktur — dem hierfür entwickelten *Spatial Landmark Graph (SLG)* — realisiert, der möglichst rechtwinklige und am jeweiligen Flächenrand angeordnete Teilflächen produziert. Alle dabei entstehenden Ergebnisse werden als gut strukturierte Richtungs- und Positionsvektoren für das spätere Aufrissverfahren bereitgestellt.

Als Datenbasis für den SLG dienen planare, beliebig konvexe und konkave Polygone, die den jeweiligen Umriss bzw. die Ausmaße der zu unterteilenden Gesamtfläche repräsentieren.

Der Stadtaufriss wurde durch ein hierfür entwickeltes Verfahren realisiert, dass aus den hervorgegangenen Grundriss-Flächendaten des SLG die eigentlichen 3D-Modelle generiert. Hierbei wird insbesondere die Optimierung der entstehenden dreidimensionalen Formen für Echtzeit-Bedingungen in den Mittelpunkt gestellt. Bevor jedoch diese Modelle generiert werden, wurde ein weiteres Verfahren vorgestellt, dass zum Einfügen von Obergeschossen und Platzieren der Fenster, Türen sowie Treppenhäuser in die zugrunde liegenden, ebenen Grundrisse verwendet wird.

Die vertikalen Formen der Stadtobjekte werden dann aus einzelnen geometrischen Bausteinen zusammengesetzt, mit Texturen versehen und auf Wunsch in verschiedenen Detailtiefen (LOD-Stufen) zu qualitativ angemessenen 3D-Modellen vereint.

Die hierbei entstehenden Ergebnisse können ohne weitere Bearbeitungsschritte in einer 3D-Echtzeit-Anwendung visualisiert und selbst in großer Anzahl interaktiv betrachtet werden.

Die Parametrisierung der Stadtobjekte wurde auf Grundlage der behandelten Disziplinen *allgemeine Stadtgeographie* und (*deutscher*) *Städtebau* konzipiert und umgesetzt. Dabei ist insbesondere auf die Theorien der funktionalen Stadtgliederung eingegangen worden, wodurch neben allgemeinen Gebäudetypen (Wohnen, Büro, Industrie usw.) auch Gebäude mit unterschiedlich genutzten Stockwerksflächen generiert werden können (z.B. Einzelhandelsbetrieb im Erdgeschoss und Büro- und/ oder Wohnbereiche in den Obergeschossen).

Alle hierbei entstehenden 3D-Modelle sind nahezu vollständig parametergesteuert und können aufgrund dessen in einer unbegrenzten Vielfalt generiert werden.

Alle prozeduralen Verfahren sowie die Hierarchie der Stadtobjekte wurden letztendlich in einer praktischen Umsetzung — dem Programm CITYLIFE — implementiert und mit einer grafischen Benutzeroberfläche und 3D-Echtzeit-Visualisierung gekoppelt.

## 8.2 Bewertung

Ziel dieser Arbeit war es, ein (ggf. aus Teilverfahren zusammengesetztes) Gesamtsystem zur Generierung von Stadt- und Gebäudemodellen zu entwickeln und umzusetzen. Dabei war insbesondere wichtig, dass die baulichen Elemente einer Stadt durch realistisch wirkende, dreidimensionale Formen repräsentiert und

auch in großer Anzahl unter Echtzeit-Bedingungen dargestellt werden können. Die Erstellung von Gebäude-Innenräumen, eine ansprechende Benutzersteuerung und ein schneller Konstruktionsprozess wurden hier als weitere Anforderungen definiert.

Für die daraus resultierenden, komplexen Problemstellungen wurden zunächst grundlegende Lösungsansätze gesichtet und bewertet. Dabei fiel die Wahl auf den prozeduralen Ansatz, um u.a. eine notwendige Bindung des Systems an die zeit- und kostenintensive Sammlung und aufwendige Verarbeitung von Geodaten zu vermeiden.

Das entwickelte prozedurale Gesamtverfahren wurde in die Teilprobleme

- Stadtgrundriss und
- Stadtaufriss

zerlegt und mit den Gesetzmäßigkeiten der allgemeinen Stadtgeographie sowie des (deutschen) Städtebaus verknüpft. Dadurch konnte das gesamte Verfahren an eine theoretische und wissenschaftlich fundierte Grundlage angelehnt und durch sinnvolle Parameter kontrolliert werden. Die beiden so entstandenen Teilverfahren werden für alle beschriebenen Stadtobjekte (*Stadt, Stadtteil, Hausblock, Grundstück, Gebäude, Stockwerk* und *Wohnung*) verwendet und dabei die jeweiligen Besonderheiten behandelt. Bemerkenswert ist hier, dass das gesamte Verfahren auch mit nur *sieben* Parametern nahezu vollständig gesteuert werden kann.

Das Besondere beim Stadtgrundriss-Verfahren — dem *Spatial Landmark Graph (SLG)* — ist, dass es die Flächen der Stadtobjekte *Hausblock* bis *Wohnung* sehr realistisch und mit allen benötigten zusätzlichen Elementen, wie Gehwegen, Gebäudeplatzierung, Haus- oder Wohnungsfluren unterteilt. Die Flächen der Stadtobjekte *Stadt* und *Stadtteil* werden zwar vereinfacht, aber ebenfalls unterteilt und mit Straßenzügen versehen.

Die eigentliche Generierung von 3D-Modellen im Stadtaufriss-Verfahren, bzw. die Visualisierung der vorher erstellten Grundrissdaten, wurde hinsichtlich der *computergrafischen Aspekte*

- Modellierung,
- Texturierung,
- Beleuchtung und
- Detailstufen

vollständig umgesetzt. Für die optimierte Darstellung in Echtzeit werden dabei alle 3D-Modelle bei vergleichsweise hoher Qualität mit einer minimalen Anzahl von Punkten und Polygonen modelliert, ihre Oberflächen jeweils mit nur einer Texturmap texturiert und diese in verschiedenen LOD- bzw. Detailstufen erzeugt.

Das Besondere des Stadtaufriss-Verfahrens ist, neben diesen informationstechnischen Umsetzungen, dass hier unterschiedliche funktionale Gebäudetypen generiert werden, die in sich eine *vertikale Nutzungsverteilung*<sup>98</sup> besitzen und vor allem mit *Innenräumen* ausgestattet sind, welche außerdem die zugehörigen

---

<sup>98</sup>D.h. innerhalb eines Gebäudes können die einzelnen Stockwerke bzgl. ihrer Raumfunktion unterschiedlich genutzt bzw. gestaltet werden.

Fassadenstrukturen entscheidend beeinflussen.

Die Ergebnisse der hier entwickelten Verfahren, bzw. des im Rahmen dieser Arbeit entstandenen Gesamtsystems, resultieren in realistisch wirkenden Stadt- und Gebäudemodellen, die bzgl. Aufbau und Gestaltung den gestellten Anforderungen entsprechen.

Der im Programm CITYLIFE *vollständig* umgesetzte Konstruktionsprozess, kann benutzerfreundlich und vor allem zügig ausgeführt werden. Die Grund- und Aufrisse der verwendeten Stadtobjekte können dabei auf jeder Stufe durch eine Vielzahl von sinnvoll gewählten Parametern mit selbsterklärenden Werten gesteuert werden. Des Weiteren können alle Ergebnisse in Echtzeit und aus frei definierten Perspektiven interaktiv betrachtet und in Dateien für eine weitere Verwendung gespeichert werden.

Die allgemeinen stadtgeographischen Theorien und städtebaulichen Gesetzmäßigkeiten konnten in allen Teilverfahren, zwar vereinfacht aber durchaus nutzbringend, integriert werden. Diese Wissensgrundlage unterstützt maßgeblich die realistische Form der Ergebnisse, wobei ihr Potenzial, bedingt durch den beschränkten Zeitrahmen dieser Arbeit, nur ansatzweise ausgeschöpft werden konnte.

Durch alle Besonderheiten der Stadtgrundriss und -aufriss-Verfahren sowie der einzelnen Stadtobjekte, kann eine klare Abgrenzung zu den bereits vorhandenen prozeduralen Verfahren für die Generierung von Stadt- und Gebäudemodellen ermöglicht und durch die Berücksichtigung der dort nicht oder nur teilweise behandelten Problemstellungen ein sinnvoller Mehrwert bereitgestellt werden.

Damit lag die Herausforderung dieser Arbeit u.a. darin, eine Vielzahl von Teilbereichen der Informatik bzw. Computergrafik mit den Erkenntnissen der allgemeinen Stadtgeographie zu verbinden und folglich für die hier behandelte Thematik gewinnbringend zu bereichern.

### 8.3 Ausblick

Ein logischer Anknüpfungspunkt für zukünftige Arbeiten wäre die realistische Grundriss-Generierung auf Stadt- und Stadtteil-Ebene. Dafür könnten z.B. die Arbeiten von MÜLLER oder KELLY aufgegriffen werden, die basierend auf L-Systemen bzw. Voronoi-Diagrammen realistisch wirkende Straßennetzmuster generieren [PaMu03], [ToKe06]. Eine neue Herangehensweise wäre die Verwendung der prinzipiell vielversprechenden Bin Packing bzw. Slicing-Tree Verfahren, mit denen ebenfalls Flächen nach bestimmten Kriterien unterteilt werden können. Ebenfalls denkbar ist die Anbindung an ein photogrammetrisches Verfahren, das Flächendaten in Form von Vektoren direkt an das hier entwickelte Aufriss-Verfahren liefert.

Des Weiteren ist es sinnvoll, die Umrissformen der Gebäude-Grundrisse durch zusätzliche Ein- und Ausbuchtungen (z.B. für Erker, Nebengebäude, usw.) zu erweitern, um die Grundflächen vielfältiger und interessanter zu gestalten (alle grundlegenden, dafür notwendigen Funktionen sind bereits vorhanden).

Für die Aufriss-Generierung könnten zunächst die hier noch fehlenden architektonischen Elemente bzw. geometrischen Bausteine, wie z.B. Balkone, Schornsteine, Zäune, Straßenlaternen usw. erstellt und in das Verfahren integriert werden, um

mit relativ wenigen zusätzlichen Polygonen die Stadt- bzw. Gebäude-Silhouetten interessanter zu gestalten. Des Weiteren kann die Integration eines Straight-Skeleton-Verfahrens sinnvoll sein, falls die Formen von Walmdächern generiert werden sollen.

Denkbar ist auch die weitere Behandlung der Teilflächen von Wohnungen, um einzelne Raumobjekte mit funktionalen Eigenschaften, wie z.B. Küche, Bad usw., zu erzeugen. Dafür kann das vorhandene Konzept mit minimalen Erweiterungen verwendet werden. Dabei ist auch das Einfügen bzw. Platzierung von Einrichtungsgegenständen möglich, die entweder aus externen Quellen geladen oder durch weitere geometrische Bausteine erzeugt werden.

Um, neben dem Europäischen, auch andere Kulturräume zu repräsentieren, wäre nur eine Anpassung der vorhandenen Parameterwerte sowie die Erstellung von entsprechenden Texturen notwendig.

Prinzipiell kann über die Gestaltung neuer bzw. weiterer Texturen der visuelle Eindruck der 3D-Modelle schnell und einfach an die verschiedensten (realen) Umgebungen angepasst und so ein ggf. gewünschter Wiedererkennungswert erhöht werden.

Zusammenfassend können durch Texturvariationen, verschiedene Parametersätze und kleineren System-Erweiterungen die unterschiedlichsten städtischen Gebiete nachgebildet werden (z.B. ganze Städtesysteme oder aber auch dörfliche Siedlungen.).

Für die spätere externe Verwendung der generierten 3D-Modelle würde sich auch die Implementation eines Exporters z.B. für die verbreiteten Formate .3ds, .obj oder .fbx lohnen, um die jetzige Bindung an das eigene, aber gut strukturierte Dateiformat der OGRE-Engine zu lockern.

In den konkreten weiterführenden Arbeitsschritten werden primär die folgenden Aspekte berücksichtigt:

- Anpassung der Grundriss-Generierung für die Objekte Stadt und Stadtteil.
- Erstellung zusätzlicher architektonischer Aufriss-Elemente.
- Erstellung von Parametersätzen für verschiedene (kulturelle) Umgebungen.

## Literatur

- [AiAu01] AICHHOLZER, AURENHAMMER, *Straight Skeletons for General Polygonal Figures in the Plane*. Computing and Combinatorics, 1996.
- [AnVi01] A. VIGNERON, *Computing a Delaunay triangulation*. National of University of Singapore, 2004.
- [AuKl01] F. AURENHAMMER, R. KLEIN, *Voronoi Diagrams*. Special Issue on Foundations of Information Processing of TELEMATIK, 1:7-11, 2002.
- [BauBG60] Bundesministerium der Justiz, *Deutsches Baugesetzbuch*. Berlin, 1960.
- [BauNVO93] Bundesministerium der Justiz, *Verordnung über die bauliche Nutzung der Grundstücke (Baunutzungsverordnung - BauNVO)*. Berlin, 1993.
- [Blote03] Blotevogel Kultur- u. Sozialgeographie, *Kap. 07 Raumsystem Stadt*, 2003.
- [BeBr01] B. BRADLEY, *Procedural Generation of Architectural Game Environments*. Computer Games Technology BSc Hons, University of Abertay Dundee.
- [BeBr05] B. BRADLEY, *Towards the Procedural Generation of Urban Building Interiors*. University of Hull, 2005.
- [BrHa01] BRENNER, HAALA *Fast Production Of Virtual Reality City Models*. Universität Stuttgart, 1998.
- [BrHa02] BRENNER, HAALA *Erfassung von 3D Stadtmodellen*. Institut für Photogrammetrie, Universität Stuttgart, 2000.
- [BrHaFr01] BRENNER, HAALA, FRITSCH *Towards Fully Automated 3D City Model Generation*. Institut für Photogrammetrie, Universität Stuttgart, 2000.
- [ChFuLiRo02] P. CHEN, Z. FU, A. LIM, B. RODRIGUES, *Two-Dimensional Packing For Irregular Shaped Objects*. National University of Singapore, Singapore Management University, Hong Kong University of Science and Technology, 2002.
- [ChKuHu00] C. CHENG, E. KUH, T. HU, *High Performance Interconnect-Dominated Circuit Analysis and Synthesis*. University of California, 2000.
- [ClBr00] C. BRENNER, *Dreidimensionale Gebäuderekonstruktion aus digitalen Oberflächenmodellen und Grundrissen*. Universität Stuttgart, 2000.
- [EpEr98] D. EPPSTEIN, J. ERICKSON, *Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions*. Symposium on Computational Geometry, 1998.

- [ErBr05] E. BRUNETON, *Modeling and Rendering Rama*. Séminaires hebdomadaires EVASION, 2005.
- [ErRa02] E. RAITH, *Stadt morphologie*. Verlag Springer Wien, 2002.
- [FeCa01] F. CACCIOLA, *A CGAL implementation of the Straight Skeleton of a Simple 2D Polygon with Holes*. 2004, Online unter: [www.cgal.org/UserWorkshop/2004/straight\\_skeleton.pdf](http://www.cgal.org/UserWorkshop/2004/straight_skeleton.pdf), letzter Zugriff 02.04.07.
- [FeOb01] FELKEL, OBDRŽÁLEK, *Straight Skeleton Implementation*. 14th Spring Conference on Computer Graphics, 1998.
- [FrGra01] BTU Cottbus, *Fraktale und regelbasierte Grafikprogrammierung*, Online unter: [http://www-gs.informatik.tu-cottbus.de/lfcg\\_s5.pdf](http://www-gs.informatik.tu-cottbus.de/lfcg_s5.pdf), letzter Zugriff, 02.04.07.
- [GaOc01] G. OCHOA, *An Introduction to Lindenmayer Systems*. University of Sussex, Online unter: [http://www.biologie.uni-hamburg.de/b-online/e28\\_3/lsys.html](http://www.biologie.uni-hamburg.de/b-online/e28_3/lsys.html), letzter Zugriff 02.04.07.
- [GrPaStLe01] S. GREUTER, J. PARKER, N. STEWART, G. LEACH, *Real-time Procedural Generation of 'Pseudo Infinite' Cities*. RMIT University, Melbourne, Victoria, Australia, 2003.
- [GrPaStLe03] S. GREUTER, J. PARKER, N. STEWART, G. LEACH, *Undiscovered Worlds - Towards a Framework for Real-Time Procedural World Generation*. RMIT University, Melbourne, Victoria, Australia, 2003.
- [GrStLe04] S. GREUTER, N. STEWART, G. LEACH, *Beyond the horizon, Computer generated, three-dimensional, infinite virtual worlds without repetition in real-time*. Image Text and Sound Conference, RMIT University, Melbourne, Victoria, Australia, 2004.
- [Gruen04] E. GRÜNBECK, *Die dritte Dimension im Kataster - eine neue Herausforderung für das amtliche Vermessungswesen*. Ansbach, 2004.
- [HaBe05] HANSCHKE, BEDDIG, *Eine Beschreibung der im Informationssystem Stadt und Umwelt (ISU) der Senatsverwaltung für Stadtentwicklung erfassten und verwalteten Struktur- und Flächennutzungskategorien von Berlin*. 1995-2005.
- [HaGu01] M. HACHET, P. GUITTON, *From Cadastres to Urban Environments for 3D Geomarketing*. Remote Sensing and Data Fusion over Urban Areas, IEEE/ISPRS Joint Workshop, 2001.
- [HaNi04] H. NILLER *Stadt und Verstädterung*. Universität Regensburg, 2004.
- [HaSe04] J. HAUNERT, M. SESTER, *Using the Straight Skeleton for Generalisation in a Multiple Representation Environment*. University of Hannover, 2004.
- [HHCGDCG01] X. HONG, G. HUANG, Y. CAI, J. GU, S. DONG, C. CHENG, J. GU, *Corner Block List: An Effective and Efficient Topological Representation of Non-Slicing Floorplan*. Computer Aided Design, ICCAD-2000, IEEE/ACM International Conference on, 2000.

- [HeHe06] H. HEINEBERG, *Stadtgeographie*. Verlag Ferdinand Schöningh, 2006.
- [Hoeng01] HÖNGGERBERG, *CyberCity Modeler und TerrainView - Werkzeuge zur Visualisierung von 3D-Stadt- und Werksmodellen*. VPK (Vermessung, Photogrammetrie und Kulturtechnik), 2001.
- [JaPu05] J. PUCHINGER, *Algorithmen für das Zweidimensionale Bin-Packing Problem*. Technische Universität Wien, 2005.
- [JeMa01] J. MARTIN, *The Algorithmic Beauty of Buildings, Methods for Procedural Building Generation*. Trinity University, 2004.
- [JeMa02] J. MARTIN, *Procedural House Generation: A method for dynamically generating floor plans*. University of North Carolina, 2006.
- [JoNo03] J. NOEL, *Dynamic Building Plan Generation*. Department of Computer Science, University of Sheffield, 2003.
- [KiUlm01] K. ULM, *Reality-based 3D city models with CyberCity-Modeler (CC-Modeller<sup>TM</sup>) And laserscanner data*. Optical 3D Measurement Techniques, ETH Zürich, 2003.
- [KoGr03] T. KOLBE, G. GRÖGER, *Towards Unified 3D City Models*. Universität Bonn, 2003.
- [KuKe01] K. KEUTZER, *The Physical Placement Problem in Integrated Circuits*, EECS University of California Berkeley, CA.
- [LaDa01] R. LAYCOCK, A. DAY, *Automatically Generating Roof Models from Building Footprints*. Journal of WSCG - FULL Papers, 2003.
- [LaDa02] R. LAYCOCK, A. DAY, *Automatically Generating Large Urban Environments based on the Footprint Data of Buildings*. SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications, 2003.
- [LaDu05] M. LARIVE, Y. DUPUY, V. GAILDRAT, *Automatic Generation of Urban Zones*. OktalSE, IRT-UPS, 2005.
- [LaFe01] M. LANCELLE, D. FELLNER, *Current issues on 3D city models*. Technische Universität Braunschweig, 2003/2004.
- [LiLuYe03] C. LIAO, H. LU, H. YEN, *Compact Floor-Planing via Orderly Spanning Trees*. National Taiwan University, 2003.
- [LoKo02] L. KOPPERS, *Generierung von Objekten für 3D-Stadtmodelle*. Universität der Bundeswehr München, 2002.
- [MaLa01] M. LANCELLE, *Automatische Generierung und Visualisierung von 3D-Stadtmodellen*. Technische Universität Braunschweig, 2003/2004.
- [MeGe01] F. MEYER AUF DER HEIDE, J. GEHWEILER, *Bin Packing oder 'Wie bekomme ich die Klamotten in die Kisten?'*. 24. Algorithmus der Woche - Informatikjahr 2006, Online unter: <http://www.informatikjahr.de/algorithmus/>, letzter Zugriff 06.04.07.

- [MJMcB05] M. MCBRYDE, *Generation of office buildings in large scale virtual worlds*. Trinity University, 2005.
- [MuVeUIVG01] P. MÜLLER, T. VEREENOOGHE, A. ULMER, L. VAN GOOL, *Automatic reconstruction of Roman housing architecture. Modeling and Visualization of Cultural Heritage*, Ascona, Switzerland, 2005.
- [MuWoHaUI05] P. MÜLLER, P. WONKA, S. HAEGLER, A. ULMER, L. VAN GOOL. *Procedural Modeling of Buildings*. ETH Zürich, Arizona State University, 2005.
- [OGRE] OGRE (Object-Oriented Graphics Rendering Engine), OGRE-SDK und OGRE API Reference, Online unter <http://www.ogre3d.org/>, letzter Zugriff 06.04.07.
- [OIKn03] O. KNIEMEYER, *Von L-Systemen zu Graphgrammatiken: Ein neuer Ansatz für die Modellierung von Artificial Life*. Arbeit im Rahmen eines Projektes der DFG-Forscherguppe Virtual Crops, TU Cottbus, 2003.
- [PaMi05] P. MIN, *3D Modelling, Lecture 12, Procedural Modelling*. Utrecht University, 2005.
- [PaMu01] P. MÜLLER, *Prozedurales Modellieren einer Stadt*. Semesterarbeit, ETH Zürich, 1999.
- [PaMu02] Y. PARISH, P. MÜLLER, *Procedural Modeling of Cities*. ETH Zürich, 2001.
- [PaMu03] P. MÜLLER, *Design und Implementation einer Preprocessing Pipeline zur Visualisierung prozedural erzeugter Stadtmodelle*. Diplomarbeit, ETH Zürich, 2000.
- [PeMa06] P. MARWEDEL, *Platzierung - Algorithmen von Breuer & Lauther; SA & GA-Verfahren -*. Universität Dortmund, 2006.
- [PlanzV90] Bundesministerium der Justiz, *Verordnung über die Ausarbeitung der Bauleitpläne und die Darstellung des Planinhalts (Planzeichnverordnung - PlanzV)*. Berlin, 1990.
- [PIDo04] *Planliche Dokumentation, Flächenwidmungsplan 2002 der Landeshauptstadt Graz*. 2002.
- [PrHaHaMe94] P. PRUSINKIEWICZ, M. HAMMEL, J. HANAN, R. MĚCH, *L-Systems: From The Theory To Visual Models Of Plants*. University of Calgary, Alberta, Canada, 1994.
- [RaWa01] C. RATHGEB, S. WALKNER, *Constructive Solid Geometry mit Depth- und Stencilbuffer*. Universität Salzburg, 2005.
- [ReKl01] R. KLEIN, *Construction Of The Constrained Delaunay Triangulation Of A Polygonal Domain*. Universität Tübingen, 1997.
- [RoSc01] N. ROSENBUSCH, G. SCHLEUPEN, *Graphisch interaktive Eingabe und Darstellung eines Geländemodells mittels Delaunay-Triangulierung*. Universität Stuttgart, 2001.

- [RoSc06] R. SCHRADER, *Framework für die Verwaltung und Visualisierung von animierten 3D-Modellen auf der Basis von DirectX®*. Institut für Computervisualistik, Universität Koblenz-Landau, 2006.
- [SaSc05] SAAKE, SCHALLEHN, *Java: Datenstrukturen für Graphen*. Universität Magdeburg, 2005/2006.
- [SiHa05] S. HAEGLER, *Modeling Architecture with String Grammars*. Diplomarbeit, Technische Hochschule Zürich, 2005.
- [StFi03] S. FISCHER, *Algorithmen & Datenstrukturen, Graphen*. Technische Universität Braunschweig, 2003.
- [StMa05] S. MARSHALL, *The Probabilistic Generation of Characteristic Urban Structure*. Bartlett School of Planning, University College London, 2005.
- [StVe05] Lehrstuhl für Raumentwicklung *Stadt morphologie: Verständnis von Strukturen und Prozessen*. Technische Universität Dresden, 2005.
- [ThOt01] T. OTTMANN, *Vorlesung Geometrische Algorithmen - Das Voronoi Diagramm. Geometrische Algorithmen -*. Universität Freiburg, 2001.
- [ToKe06] T. KELLY, *City Architecture Generation*. University of Bristol, 2006.
- [TrSy03] T. SYNNOTT, *Alternatives To Traditional 3D Modelling Methods*. Univ. of Queensland, Dept. of Electrical and Computer Engineering, 2003.
- [VeDev01] VESTRI, DEVERNAY, *Using Robust Methods for Automatic Extraction of Buildings*. Hawaii, USA, 2001.
- [WaPaNa01] R. WACK, G. PAAR, B. NAUSCHNEGG, H. URBAN, *Erzeugung von 3D Stadtmodellen*. Proc. 12th International Geodesic Week, 2003.
- [WeZo04] WESTPHAL, ZORNOW, *Gestaltwandel der Stadt*, Technische Universität Berlin, 2004.
- [WoWiSi01] P. WONKA, M. WIMMER, F. SILLION, W. RIBARSKY, *Instant Architecture*. ACM Transactions on Graphics, July 2003.
- [YoChSh01] E. YOUNG, C. CHU, Z. SHEN, *Twin Binary Sequences: A Non-Redundant Representation for General Nonslicing Floorplan*. Int. Symp. Physical Design, 2002.
- [YoYaWo01] F. YOUNG, H. YANG, D. WONG, *On extending slicing floorplans to handle L/T-shaped modules and abutment constraints*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2000.
- [ZhLi94] Z. LI, *Compaction Algorithms For Non-convex Polygons and Thier Applications*. Harvard University, Cambridge, Massachusetts, 1994.