

Automatisierte Erzeugung synthetischer Luftbilder auf Basis von Geodaten

RETO C. STUBER

DIPLOMARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

INTERACTIVE MEDIA

in Hagenberg

im Juli 2011

© Copyright 2011 Reto C. Stuber

Diese Arbeit wird unter den Bedingungen der *Creative Commons Lizenz Namensnennung–NichtKommerziell–KeineBearbeitung Österreich* (CC BY-NC-ND) veröffentlicht – siehe <http://creativecommons.org/licenses/by-nc-nd/3.0/at/>.

Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet habe.

Hagenberg, am 21. Juni 2011

Reto C. Stuber

Inhaltsverzeichnis

Erklärung	iii
Kurzfassung	vi
Abstract	vii
1 Einleitung	1
1.1 Aufbau	5
2 Verwandte Forschung	6
2.1 Prozedurales Modelling	6
2.2 Beispielbasierte Synthese	8
3 Verfahren zur Flächensynthese in urbanen Layouts	11
3.1 Stilebenen in Layouts	11
3.1.1 Beispielbasierte Synthese	13
3.1.2 Fokus auf Flächensynthese	15
3.2 Feature Extraktion und Abgleich	17
3.2.1 Strukturelle Features	17
3.2.1.1 Geometrische Form	17
3.2.1.2 Topologische Relationen	19
3.2.1.3 Grenzfeatures	22
3.2.2 Semantische Features	24
3.2.2.1 Klassifizierung und Matching	25
3.2.2.2 Semantische Taxonomie	27
3.2.2.3 Berechnung semantischer Nähe	28
3.2.2.4 Relationen	29
3.2.2.5 Topologisch-semantische Relationen	29
3.3 Strukturelles Syntheseverfahren	31
3.3.1 Energie-basiertes Lösungsverfahren	33
3.3.1.1 Optimale Layoutkonfiguration	35
3.3.2 Elementeplatzierung	36
3.3.2.1 Approximierte Ortsbestimmung	37
3.3.2.2 Geometrische Passgenauigkeit	37

3.3.2.3	Platzierungsenergie	39
3.3.2.4	Aktualisierung des Containers	41
3.3.3	Lokale Randbedingungen	42
3.3.3.1	Bestimmung von Nachbarschaft	42
3.3.3.2	Feature Matching	45
4	Implementierung	48
4.1	Programmablauf	48
4.1.1	Preprocessing Phase	48
4.1.2	Synthesis Phase	49
4.2	Geometric Hashing	49
4.2.1	Datenstruktur	50
4.2.2	Rasteralgorithmus	52
4.2.3	XOR-Maskenoperator	52
5	Auswertung	54
5.1	Quantitative Analyse	54
5.1.1	Qualitätsmetrik	55
5.1.2	Versuchsreihen zur Flächensynthese	55
5.1.2.1	Datenvorbereitung	55
5.1.2.2	Durchführung	56
5.1.2.3	Auswertung	57
5.2	Qualitative Analyse	59
6	Zusammenfassung	64
6.1	Ausblick & Verbesserungen	66
A	Detaillierte Auswertung	69
B	Testprogramm	71
C	Inhalt der CD-ROM	73
C.1	PDF-Dateien	73
C.2	Quelltext	73
C.3	Versuchsdaten	73
	Literaturverzeichnis	74

Kurzfassung

In dieser Arbeit wird ein neuartiger Ansatz zur Erzeugung und Darstellung virtueller Städte thematisiert.

Das präsentierte Verfahren ermöglicht die vollautomatische Herstellung synthetischer Luft- und Satellitenbilder. Dazu werden aus GIS Datenbanken bezogene Geodaten gemeinsam mit Satellitenbildern in einem Beispielbasierten Syntheseverfahren verarbeitet. So können auf einfache Weise neue Bildansichten von Städtelayouts produziert werden. Eine derartige Technologie kann u.a. in Bereichen wie der Städteplanung, urbanen Simulation und in der Unterhaltungsindustrie Anwendung finden.

Das vorgestellte Syntheseverfahren baut dabei auf bekannte Techniken aus dem featurebasierten Image-Retrieval und der Erzeugung von Bildmosaiken auf. Als besondere Erweiterung dieser Techniken werden der Einbezug semantischer Geodaten in das Syntheseverfahren behandelt und hierzu neue Featuremetriken vorgestellt.

Entgegen dem aktuellen Forschungstrend in der Städtevisualisierung (*Urban Visualization*) hin zum *Prozeduralen Modelling*, verfolgt diese Arbeit bewusst einen Ansatz aus der Beispielbasierten Synthese. In diesem Rahmen werden die Vorteile beispielbasierter Methoden erörtert.

Zum Abschluss werden mit dem Verfahren erzeugte Resultate präsentiert und künftige Verbesserungen diskutiert.

Abstract

In this work a novel approach towards the creation and visualization of virtual urban environments is presented.

This technique makes it possible to produce artificial aerial and satellite imagery in a fully automated fashion. This is done by processing geographical data and satellite imagery from GIS databases in an example-based synthesis framework. Novel views of urban layouts can be easily created with this. The technology may be of use in applications such as urban planning, urban simulation and the entertainment industry.

The presented synthesis framework enhances upon well known techniques from feature-based image retrieval and the production of computer-generated image mosaics. A notable extension has been made by including semantic geographical data into the synthesis process and by introducing new feature distance metrics for these.

Against current trends in *urban visualization* towards *procedural modelling* an example-based synthesis approach is pursued in this work. In that scope the benefits of example-based methods will be examined.

At the end some results produced with this framework will be presented and future enhancements will be discussed.

Kapitel 1

Einleitung

Die Erfassung und Verarbeitung geographischer Daten hat in den vergangenen Jahrzehnten einen beachtlichen Wandel vollzogen. Wurden derartige Informationen früher noch nach traditionellen Methoden der Kartographie erfasst, aufbereitet und dargestellt, so finden sich heute kaum mehr Systeme, die nicht vollständig auf eine digitale Form der Speicherung und Weiterverarbeitung umgestellt worden sind.

Während sich die Verarbeitung geographischer Daten durch sogenannte Geographische Informationssysteme (GIS) in klassischen Anwendungsszenarien, wie der Landvermessung, Grenzbestimmung und insbesondere der Navigation durchgesetzt hat, ist das Potenzial dieser Technologie in anderen Bereichen größtenteils ungenutzt geblieben. Hierzu kann insbesondere die Visualisierung geographischer Daten gezählt werden. Bekannte Formen heutiger Visualisierungsansätze, etwa in Form von Vektorkarten (siehe Abb. 1.1 a), digitaler Satellitenbilddatenbanken (Abb. 1.1 b) oder dreidimensionaler Globen (Abb. 1.1 c), finden sich dabei alle in traditionellen Entsprechungen wieder.

Neuartige Ansätze zur Visualisierung können aber gerade für andere Anwendungsgebiete als die oben genannten von Interesse sein. Dazu zählen u.a. die Städte- und Landschaftsplanung, die urbane Simulation, die Rekonstruktion historischer Gebiete und seit wenigen Jahren der wachsende Markt ortsbasierter Spiele- und Multimediaanwendungen. Insbesondere die Visualisierung in realitätsnaher Darstellungsform zählt hier zu den am wenigsten entwickelten Feldern.

Neue Visualisierungsansätze

Unter den wenigen Ansätzen, die seit längerem existieren, findet sich u.a. die virtuelle Darstellung urbaner Landschaften durch einfache, automatisch generierte, dreidimensionale Gebäudemodelle oder deren komplexeren, dafür meist manuell erstellten Pendanten (s. Abb. 1.1 d). Insbesondere automatisierte Verfahren beschränken sich bisher darauf, realweltlich existierende

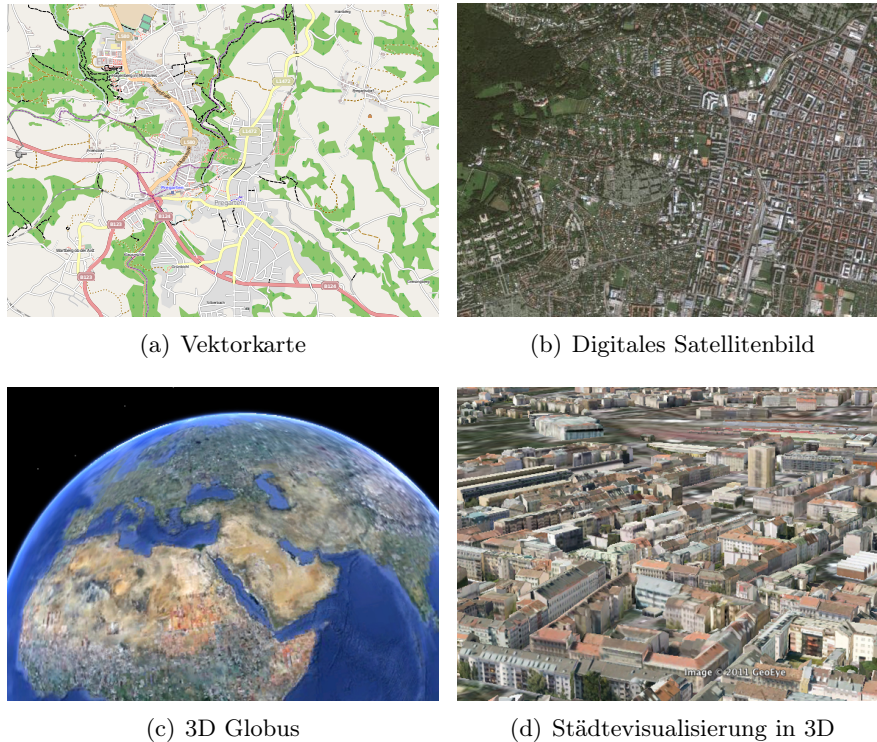


Abbildung 1.1: Verschiedene Visualisierungsformen von Geodaten. (Bildquelle: OpenStreetMap.org (a), Google Inc. (b–d))

Geodaten mit Hilfe verschiedener Techniken zu reproduzieren.

Zu diesen Techniken zählen beispielsweise die Gewinnung architektonischer Grundformen aus strukturiertem Kartenmaterial oder über die Anwendung von Computer-Vision Methoden. Ergänzt wird dies häufig durch die ästhetische Aufwertung der erzeugten (3D-)Modelle mittels Bilddaten aus verschiedenen Quellen (u.a. Straßenrandfotografie).

Erst seit wenigen Jahren finden sich vereinzelt Bemühungen zur computergestützten Erzeugung voll- oder teilsynthetischer Visualisierungsinhalte von Geodaten. Dienten vorherige Ansätze der bloßen Reproduktion bestehender Geostrukturen (wie z.B. existierender Städtelayouts), so liegt der Fokus dieser Methoden darauf, vollständig oder teilweise neue, hypothetische Ansichten zu erzeugen. Die Abbildung von Städten steht im dabei Fokus vieler Anwendungen.

Städtevisualisierung

Gerade in der Städteplanung und Simulation sind neuartige Methoden zur Visualisierung vielversprechend. Dort müssen i.d.R. auf Basis rein künstlicher Datensätze – beispielsweise von Bauplänen – nützliche Darstellungsfor-

men für Geostrukturen gefunden werden. So müssen beispielsweise großflächige Bauvorhaben in Städten vorab veranschaulicht werden. Zudem sollen strukturelle Veränderungen im zeitlichen Verlauf simuliert oder von Grund auf künstliche Visualisierungen von zukünftigen urbanen Lebensräumen geschaffen werden.

Gerade der fotorealistischen Darstellung der stark abstrakten Geodaten aus GI-Systemen ist besondere Bedeutung zuzumessen. Diese Darstellungsform liegt der menschlichen Wahrnehmung von Örtlichkeit am nächsten und birgt daher ein besonders großes Nutzungspotenzial.

Ansätze in der Städtevisualisierung (engl.: *Urban Visualization*) haben dabei in der Vergangenheit verschiedenste Fokusse gehabt. Darunter die Erzeugung und Visualisierung neuartiger urbaner Szenarien (u.a. [Parish and Müller, 2001]), das Bearbeiten und Erweitern bestehender urbaner Layouts (u.a. [Aliaga et al., 2008]), das Replizieren des Erscheinungsbildes einer Vorlage in neu generierten Städtelayouts [Aliaga et al., 2008], sowie das räumliche Zusammenführen von Layouts unterschiedlicher Stile. Dazu wurden Methoden aus dem *Prozeduralen Modelling* und der *Beispielbasierten Synthese* verwendet (näheres dazu in Kapitel 2).

Diese großteils experimentellen Ansätze unterscheiden sich sowohl in der verwendeten technologischen Methodik (prozedural oder beispielbasiert) als auch in weiteren signifikanten Merkmalen. Dazu zählen insbesondere die strukturelle Ebene in urbanen Layouts auf denen sie operieren, deren Anforderungen an die verwendeten Inputdaten, sowie die Art, der Detailgrad und die Qualität ihres Outputs. Außerdem ist von Belang, ob es sich um voll- oder teilautomatische Verfahren handelt.

Während prozedurale Verfahren inzwischen die Produktion und Reproduktion von Geostukturdaten auf fast allen Ebenen urbaner Layouts ermöglichen – von der Visualisierung kompletter Infrastrukturnetze bis hin zu den Details einzelner Gebäudefassaden [Müller et al., 2006, 2007, Parish and Müller, 2001] – wurden beispielbasierte Ansätze bisher nur vollständig auf hochstruktureller Ebene zur Produktion von Straßennetzen in solchen Layouts angewendet [Aliaga et al., 2008]. Gerade aber für vollautomatische Anwendungen sind beispielbasierte Verfahren aufgrund ihrer autarken Funktionsweise besonders effektiv. Der Umfang und die Komplexität großräumiger Geolayouts wiederum sprechen für eine Vollautomatisierung der Produktionskette.

Beispielbasiertes Syntheseverfahren

Die Ausweitung beispielbasierter Verfahren auf andere Ebenen urbaner Layouts scheint aus genannten Gründen eine vielversprechende Richtung zu sein.

In dieser Arbeit soll daher ein neuartiger, beispielbasierter Ansatz erörtert werden, um Städtelayouts auf Ebene der Nutzflächen zu visualisieren. Mit diesem Verfahren kann erstmals der Stil bestehender urbaner Layouts auf

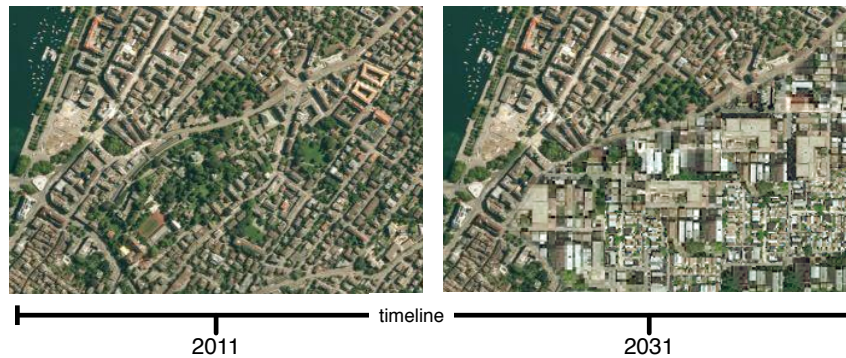


Abbildung 1.2: Urbane Flächennutzung im Wandel: Links die heutige Ansicht einer Stadt. Rechts eine hypothetische Ansicht in der Zukunft. Zur Erzeugung des rechten Bildes wurde das hier beschriebene Beispielbasierte Syntheseverfahren verwendet. Mehr Resultate finden sich in Kap. 5.

genannter Ebene auf neue Layouts übertragen und das Ergebnis visualisiert werden. Eines vieler möglicher Anwendungsszenarien ist beispielsweise die Simulation von Flächennutzung in urbanen Lebensräumen, wie in Abb. 1.2 dargestellt.

Die vorliegende Arbeit beschränkt sich dabei auf eine Visualisierung in Form synthetisch erzeugter, zweidimensionaler Satellitenbilder. Im Rahmen dieses Ansatzes werden u.a. die Vorteile der Nutzung beispielbasierter Lösungsansätze diskutiert. Zu den wichtigsten Beiträgen dieser Arbeit gehören:

- Ein vollautomatisiertes Syntheseverfahren zur künstlichen Erzeugung und Visualisierung von Nutzflächen urbaner Layouts;
- Die Verwendung von Techniken zur Reproduktion eines Stils in neuen Layouts anhand von Beispieldaten. Darunter die Verwendung eines Feature-Retrieval & Matching Frameworks;
- Die erstmalige Betrachtung der Synthese von strukturierten Flächen als *Shape Packing* Problemstellung und der Wahl eines dazu geeigneten Synthesearchgorithmus;
- Der Einbezug verschiedener Randbedingungen in den Synthesearchgorithmus – insbesondere der Einbezug semantischer Beziehungen zwischen Strukturelementen in deren räumlichen Kontext;
- Die Einführung neuer Metriken zur Distanzbestimmung komplexer Eigenschaften von Strukturdaten, u.a. von Semantikattributen und topologischen Relationen.

1.1 Aufbau

Der Aufbau der Arbeit ist dabei wie folgt gegliedert. In Kapitel 2 wird der Status quo im Forschungsumfeld der Städtevisualisierung und artverwandter Gebiete aufgezeigt. Dabei werden die wichtigsten Strömungen, namentlich das *Prozedurale Modelling* und die *Beispielbasierte Synthese*, getrennt untersucht. Dabei wird auch auf die Entwicklung innerhalb dieser näher eingegangen.

In Kapitel 3 wird das neuartige Verfahren dieser Arbeit ausführlich erläutert. Nach einer Einleitung, in der die Wahl einer geeigneten Methodik begründet wird, sind in Abschnitt 3.2 die Gewinnung und der Abgleich von Merkmalen der Strukturdaten als Teil des Feature-Frameworks im Detail beschrieben. In Abschnitt 3.3 wird der Kern des eigentlichen Syntheseverfahrens vorgestellt. Skizziert wird ein geometrisches Platzierungsverfahren für Strukturelemente in urbanen Layouts. Außerdem wird der Einbezug der im vorherigen Abschnitt ausgewiesenen Merkmale als Teil der Synthese thematisiert.

Kapitel 4 widmet sich kurz der Implementierung des Verfahrens als Programm zu Versuchszwecken. In Kapitel 5 werden dann die Resultate ausgewertet, die durch Anwendung des Verfahrens erzielt werden konnten. Dazu wird die Tauglichkeit des Verfahrens sowohl anhand objektiv messbarer Kriterien (quantitative Analyse), als auch mittels subjektiver Kriterien (qualitative Analyse) untersucht. Kapitel 6 gibt schließlich eine Zusammenfassung der einzelnen Teile dieser Arbeit und endet mit einem Ausblick auf mögliche zukünftige Verbesserungen des Verfahrens.

Kapitel 2

Verwandte Forschung

Das in dieser Arbeit vorgestellte Verfahren ermöglicht die automatisierte Erzeugung ästhetisch und logisch konsistenter synthetischer Satellitenbilder von Städten. Vergleichbare Forschungsbemühungen der Vergangenheit zur Visualisierung geographischer Daten können unter dem Begriff der *Urban Visualization* zusammengefasst werden, einem bedeutenden Teilgebiet des *Urban Modelling*.

Urban Visualization umfasst all jene computergestützte Verfahren, die der – meist fotorealistischen – virtuellen (Re-)produktion und Darstellung urbaner Lebensräume dienen. Dies beinhaltet es u.a., räumliche Strukturdaten wie Gebäude, Straßen und vergleichbare menschgemachte Strukturen zu generieren und zu rendern. Urban Visualization grenzt sich dadurch deutlich von der explorativen Geovisualisierung etwa in Form thematischer Karten ab. Ebenso besteht nur geringe Überlappung zum Gebiet des Digital Terrain Modelling, das sich ausschließlich dem Darstellen geologischer Geländedaten widmet.

Bisherige Ansätze basierten dabei meist auf zwei prominenten Techniken zur Umsetzung der Visualisierung. Zum einen ist dies das *Prozedurale Modelling*, zum anderen die *Beispielbasierte Synthese* (vgl. [Vanegas et al., 2010]). Neben diesen existieren zahlreiche Verfahren aus dem Bereich der Computer Vision zur Rekonstruktion von Geostrukturen auf Basis von Bilddaten (Satelliten-/Straßenrandfotografie, u.a. [Debevec et al., 1996, Huertas and Nevatia, 1988]), deren Output zumindest als Grundlage für weitere Verfahren genutzt werden können.

2.1 Prozedurales Modelling

Um neue Strukturen wie Gebäudemodelle zu erschaffen, kommen im Urban Modelling hauptsächlich prozedurale Verfahren zum Einsatz. Diese operieren auf Basis einer überschaubaren Anzahl einfacher Produktionsregeln zur automatisierten Erzeugung. Meist sind diese Regeln in Form von Gramma-

tiken gehalten und können manuell verfasst sein oder aber aus Beispieldaten gewonnen werden (siehe [Aliaga et al., 2007]).

Stiny führte erstmals sogenannte Shape Grammars ein [Stiny, 1975, 1980]. Shape Grammars operieren direkt auf geometrischen Primitiven und können mit Hilfe von Ableitungsregeln (auch Produktionsregeln genannt) die iterative Veränderung geometrischer Strukturen steuern. Insbesondere zur prozeduralen Modellierung komplexer architektonischer Formen (z.B. Gebäuden) sind Shape Grammars hervorragend geeignet [Vanegas et al., 2010].

Das Prinzip der Shape Grammars wurde später von Wonka et al. [2003] in deren Set- bzw. Split Grammars verfeinert. Diese Grammatiken erweiterten Shape Grammars um die bisher fehlende Möglichkeit einer präzise steuerbaren Ableitung im Produktionssystem. Dies geschieht durch eine regelbasierte Auswahl der für die im jeweiligen Iterationsschritt geeignetsten Produktionsregel. Des Weiteren wurde das bloße Ersetzen von Formen um das Teilen (Split) von Flächen oder Volumen ergänzt.

Müller et al. führten darauf aufbauend eine ähnliche Grammatik ein. Diese wurde bei der Erzeugung von Gebäudemassemodellen [Müller et al., 2006] und später zur prozeduralen Erzeugung von Gebäudefassaden [Müller et al., 2007] eingesetzt.

Neben Shape und Set Grammars existiert noch eine Reihe weiterer Grammatiken und deren Produktionssysteme. Diese können zur automatisierten geometrischen Modellierung verwendet werden, sind jedoch nicht speziell zu diesem Zweck entstanden. Nach [Vanegas et al., 2010] sind dies u.a. Chomsky Grammars [Chomsky, 1959], Graph Grammars [Ehrig et al., 1999], Attributed Grammars [Knuth, 1968] und L-Systems [Prusinkiewicz and Lindenmayer, 1990].

Insbesondere L-Systems nehmen eine bedeutende Rolle im Urban Modelling ein. L-Systems basieren ebenfalls auf iterativ anwendbaren Produktionsregeln. Sie wurden ursprünglich zur Modellierung von Pflanzen, später aber auch in einer parametrisch-stochastischen Variante zur Erzeugung von Straßennetzwerken und Gebäuden verwendet (s. [Parish and Müller, 2001]). Zur Produktion von Gebäuden wurde sie inzwischen vollständig durch andere, weiter oben genannte, Grammatiken ersetzt. Dies ist aufgrund der nicht wachstumsähnlichen Charakteristik architektonischer Formen (vgl. Pflanzen).

Neben der Anwendung regelbasierter, prozeduraler Verfahren zur Erzeugung neuer Strukturentitäten gab es auch vielzählige Ansätze zur Arrangierung und Restrukturierung bestehender Objekte, um so neue Layouts zu generieren. Neben anderen wendeten Smith et al. [2001] und Xu et al. [2002] Constraint-Regeln auf Objekte bzw. Flächen in virtuellen Szenen an, um eine logisch konsistente Anordnung unter manuell platzierten Objekten sicher zustellen.

Diese Regeln umfassen Anweisungen hinsichtlich erlaubter räumlicher Beziehungen. Eine legitime Anordnung aller Objekte zueinander wird dabei

über einen prozeduralen Constraint Solver ermittelt. Andere [Lopes et al., 2010, Sanchez et al., 2003, Tutenel et al., 2009] verwendeten ähnliche Verfahren zur automatisierten Erstellung von vollständigen Layouts. Erzeugt wurden u.a. Innenräumen und deren Einrichtungen für virtuelle Anwendungen, wie etwa Computerspiele.

Alle genannten Verfahren haben gemeinsam, dass sie zwar ein beliebig hohes Maß an Gestaltungsfreiheit und Detailgrad ermöglichen, dies jedoch auf Kosten der Arbeitsintensität zur Erstellung notwendiger Regeln geht. Des Weiteren erfordert das Schreiben von Produktionsregeln ein tiefes Verständnis geometrischer Zusammenhänge in architektonischen Formen oder Layouts. Es gestaltet sich dadurch umso schwieriger, bestehende urbane Szenarien zu reproduzieren.

Ansätze zur Ableitung von prozeduralen Regeln aus Beispieldaten beschränken sich bis dato auf semi-automatische Verfahren (s. [Aliaga et al., 2007, Müller et al., 2007]) und sind daher noch unzureichend für vollautomatisierte Prozesse.

2.2 Beispielbasierte Synthese

Techniken der Beispielbasierten Synthese bilden einen alternativen Ansatz zum Prozeduralen Modelling. Alle Syntheseverfahren haben gemein, dass sie einen initialen Datensatz benötigen, auf dem basierend sich neue oder erweiterte Daten erzeugen lassen. Dies hat den entscheidenden Vorteil, dass keine Produktionsregeln wie beim Prozeduralen Modelling a priori verfasst werden müssen. Für die Erzeugung urbaner Strukturen in der Urban Visualization bedeutet dies, dass sich neue, hypothetische Strukturentitäten (z.B. Gebäude) und Layouts (Straßennetze, Grundstücksflächen etc.) anhand solcher Vorlagen erzeugen lassen.

Syntheseverfahren können grob in zwei Gattungen getrennt werden. Jene, die auf unstrukturierten und jene, die auf strukturierten Daten operieren. Die Synthese unstrukturierter Daten bezieht sich darauf, rohe Bildpixel ohne weitere Kenntnis über Inhaltsstrukturen in Bildern zu verarbeiten. Darunter fallen insbesondere Techniken der Textursynthese (z.B. [Efros and Leung, 1999, Heeger and Bergen, 1995, Popat and Picard, 1993]) und der Bildsynthese im Allgemeinen.

Unstrukturierte Daten

Die Textursynthese erlaubt, meist beliebig große Texturmengen anhand kleiner Beispielsamples zu erzeugen. Auch wenn viele dieser Verfahren keine explizite Anwendung in der Urban Visualization vorsehen, sind ihre Problemstellungen häufig analog und ihre Herangehensweisen leicht übertragbar.

Avidan und Shamir [2007] haben mit Seam Carving ein Verfahren zum verzerrungsfreien, nicht-linearen Skalieren von Bildern vorgestellt, das auf in-

telligenter Interpolation und Extrapolation beruht. Damit ließen sich neue Layouts mit verändertem Seitenverhältnis konstruieren. Fang et al. führten eine auf Textursynthese basierende Technik zum Deformieren einzelner Bildsegmente vor. Diese minimiert Verzerrung, indem neue Texturregionen – wo nötig – synthetisiert werden [Fang and Hart, 2007]. Zhang et al. [2003] zeigten einen Ansatz zum Synthetisieren nicht-homogener, inhaltlich variierender Texturbilder. Atomare Texturelemente müssen hier jedoch noch als so genannte Texton Masks manuell annotiert werden.

Wie bei Texton Masks gehen weitere Syntheseverfahren von größeren, zusammenhängenden Bildeinheiten aus, ohne aber Rücksicht auf deren semantische und kontextuelle Bedeutung zu nehmen. Kim und Pellacini [2002] stellten ein Verfahren zur mosaikartigen Anordnung beliebiger Formen und [Cohen et al., 2003, Efros and Freeman, 2001, Kwatra et al., 2003, Stam, 1997] Verfahren zum aperiodischen, nahtlosen ‚Kacheln‘ von größeren Bildfragmenten vor.

Alle Verfahren können theoretisch auch auf Bilder urbaner Szenerien angewandt werden, womit sich neue, hypothetische Ansichten generieren ließen. Hertzmann et al. [2001] haben mit ihren Image Analogies ein weiteres, mächtiges Syntheseverfahren vorgestellt. Es erlaubt, über vorab annotierte Bildbeziehungen zwischen zwei Bildern, ein neues Bild analog zur Vorlage zu erzeugen. U.a. präsentierten sie in ihrer Arbeit diverse damit produzierte Bilder von Landschaften und urbanen Szenerien.

Kritisch sind genannte Techniken jedoch in der Anwendung auf hochstrukturierte Inhalte, die gerade in Bilddaten urbaner Szenerien vorkommen. Selbst bei manuell annotierten Bildern (vgl. Textures-by-Numbers, [Hertzmann et al., 2001]) werden nur semantisch unzureichende Informationen in die Synthese mit einbezogen, wie etwa die unterscheidbaren Bildregionen. Die annotierten Regionen bleiben aber für den jeweiligen Algorithmus nur eine Sammlung unstrukturierter Pixeldaten.

So lassen sich inkonsistente Darstellungen von makroskopischen Strukturelementen, etwa von Straßenzügen oder Gebäudeanordnungen, in den synthetisch erzeugten Bildern mit diesen Verfahren kaum vermeiden. Auf Detailstrukturen und Bildinhalte stochastischer Natur (Grünflächen, Wälder etc.) lassen sich viele der Verfahren jedoch problemlos anwenden.

Strukturierte Daten

Für Aliaga et al. [2008] waren genannte Unzulänglichkeiten Anlass, Strukturdaten in das Syntheseverfahren einzubeziehen. Grund war auch die Tatsache, dass Geostrukturdaten in Form von Vektorkarten heute vielfach und frei verfügbar sind.

Ihr Verfahren ermöglicht u.a. die Synthese von ganzen Straßennetzen und Grundstücksparzellen urbaner Layouts. Ihre Algorithmen untersuchen dazu die grundlegenden Charakteristika der Straßennetze und Flächen eines Lern-

datensatzes. Sie nutzen diese Informationen, um neue Straßen und Flächen in einem vergleichbaren Stil generieren zu lassen. Dadurch wird eine logische Konsistenz der Strukturen im Bild gewährleistet, sodass Straßen nicht mehr abrupt enden oder Grundstücksflächen ungewöhnliche Formen oder Größen aufweisen.

Neben der Synthese der Strukturdaten werden zur Visualisierung außerdem Bildfragmente von Parzellen – verfügbar in Form geo-referenzierter Satellitenbilder – aus dem Lerndatensatz extrahiert und neu generierte Grundstücksparzellen damit gefüllt. Für die Auswahl geeigneter Quellbildfragmente wird eine Ähnlichkeitsmetrik verwendet, die auf der Übereinstimmung der Flächeninhalte und dem Seitenverhältnis der orientierten Bounding Boxen (OOBs) zweier Flächen basiert.

In einer früheren [Aliaga et al., 2008] und späteren Arbeit [Vanegas et al., 2009] der gleichen Autoren kommen außerdem einfache semantische Attribute als Übereinstimmungskriterien zum Tragen. Darunter beispielsweise die Flächennutzung oder die durchschnittliche Anzahl der Bewohner je Grundstück. Zum Einpassen des ähnlichsten Bildfragments verwenden Aliaga et al. schließlich eine Bildwarping-Technik. Diese versucht eine räumliche Ein-zu-eins-Korrespondenz zwischen den Grundstücken mittels Flächensegmentierung herzustellen.

Während das von Aliaga et al. präsentierte Verfahren plausible Ergebnisse in der Synthese der Strukturdaten (Straßen, Parzellen) liefert, findet eine Synthese der zugrunde liegenden Bilddaten nur unzureichend statt.

Bei genauer Betrachtung wirken die Bildfragmente der Parzellen beliebig zueinander angeordnet, da keinerlei lokale Nachbarschaftsbeziehungen in die Vergleichsmetrik mit einfließen. Dies wiederum führt zu meist harten Kanten zwischen und – aufgrund des verwendeten Warpings – zu teils starken Verzerrungen innerhalb der Bildfragmente. Eine ästhetisch konsistente, detailgetreue Visualisierung ist durch dieses Verfahren nicht möglich.

Das Hauptaugenmerk des hier beschriebenen Verfahrens liegt jedoch im Besonderen auf einer – auch im Detail konsistenten – Visualisierung synthetischer Satellitenbilder. Es bleibt hinzuzufügen, dass der Fokus der Arbeit von Aliaga et al. auf der Visualisierung umfangreicher, großflächiger urbaner Datensätze und weniger auf örtlichen Details liegt.

Kapitel 3

Verfahren zur Flächensynthese in urbanen Layouts

3.1 Stilebenen in Layouts

Urbane Layouts können vereinfacht als eine Anordnung aus Transportnetzen, Nutzflächen sowie Gebäudestrukturen verstanden werden. Dabei bedeuten jede jeweils eine räumliche Untergliederung (Segmentierung) des anderen. Abb. 3.1 a zeigt das vollständige Layout einer Stadt in Form eines georeferenzierten Satellitenbildes, wie sie in gängigen GIS Datenbanken erfasst sind. Abb. 3.1 b hingegen bildet nur die strukturellen Ebenen der Straßen- und Grundstückstopologie gewonnen aus strukturiertem Vektorkartenmaterial ab, während Abb. 3.1 c einzelne, unzusammenhängende Fragmente der visuellen Detailebene darstellt.

Die Detailebene bezeichnet dabei die wahrnehmbare Geometrie-, Material- und Farbkombination der zugrundeliegenden Strukturelemente, also etwa der Straßen und Gebäude. Die strukturelle Ebene hingegen definiert sich ausschließlich durch die räumliche Anordnung und Verknüpfung dieser Elemente im Ganzen. Durch eine getrennte Betrachtung wie in Abb. 3.1 b und 3.1 c lassen sich dabei stilprägende Eigenschaften auf beiden dieser Ebenen erkennen.

Trennung von Stilebenen

Zwei Straßennetze können beispielsweise den gleichen strukturellen Stil in Form einer ähnlichen Netzwerktopologie aufweisen. Dennoch können sie grundlegend verschiedene visuelle Ausprägung besitzen, etwa einmal als Asphaltstraßen und einmal als Feldwege. Andererseits können zwei Städtelayouts gleiche Gebäudetypen besitzen, einmal aber in einem quadratischen Raster, das andere Mal in einer natürlich gewachsenen Anordnung aufweisen.

Beide Ebenen prägen nach dieser Beobachtung folglich gemeinsam das Erscheinungsbild eines urbanen Layouts. Dass unterschiedliche Kombinations-

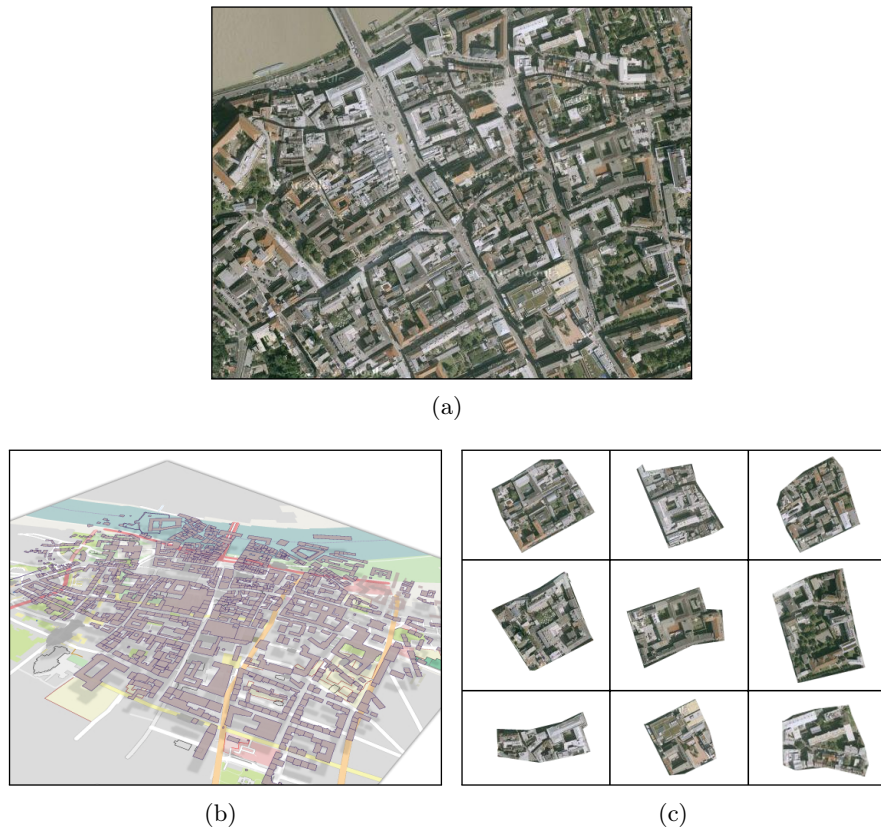


Abbildung 3.1: a) Satellitenbild eines urbanen Layouts. Die strukturelle Ebene und die Detailebene verschmelzen zu einem konsistenten Gesamteindruck. b) Dieselbe Stadt in Form einer Vektorkarte. Straßen- und Grundstückstopologien geben die strukturelle Ebene des Layouts vor. c) Einzelne Bildfragmente aus dem Layout. Ohne eine topologische Anordnung dieser lässt sich kein Stil auf Ebene des gesamten Layouts ausmachen. Die Aufspaltung von a) in b) und c) verdeutlicht dabei die verschiedenen Ebenen eines Layouts und ihre formale Trennbarkeit. (Bildquelle: Google Inc. (a, c), OpenStreetMap.org (b))

nen zwischen Struktur- und Detailebenen in Layouts auftreten, deutet darauf hin, dass beide formell getrennt werden können. Dies wiederum macht eine Synthese zwischen Ebenen unterschiedlicher Ausprägung denkbar. Struktur- und Detailebenen könnten demnach voneinander getrennt und in unterschiedlichen Variationen neu zusammengeführt werden. Damit ließen sich völlig neuartige Erscheinungsbilder von Layouts erzeugen. Diese Erkenntnis bildet den Kerngedanken des hier dargelegten Verfahrens.

Indem eine der zwei tragenden Stilebenen eines Layouts durch eine Ebene anderer Ausprägung ersetzt wird, lässt sich eine Art Hybridlayout erzeugen. Die Ersetzung kann dabei ebenfalls nur partiell auf gewisse Bereiche eines

Layouts angewendet werden. Anstelle von Substitution kann auch vom *Klonen* eines Stils gesprochen werden, sofern es ein Quelllayout gibt, aus dem der Stil übernommen werden kann.

Strukturdaten sind in GIS-Systemen typischerweise als Vektordaten verfügbar. Sie umfassen u.a. die Straßen-, Nutzflächen- und Grundstückstopologie und Geometrie. Die Detailebene kann implizit in Form von Rasterdaten, z.B. georeferenzierter Satellitenbilder angenommen werden. Das ‚Stil-Klonen‘ erfolgt, indem Fragmente geokodierter Satellitenbilder aus einem Layout mit den Vektordaten aus einem zweiten Layout zusammengeführt werden. So wird ein neues, synthetisches Layout generiert.

3.1.1 Beispielbasierte Synthese

Die Synthese zwischen Stilebenen unterliegt allerdings einer Vielzahl an Einschränkungen. Geographische Layouts umfassen äußerst komplexe, kontextsensitive Strukturelemente¹. Dies betrifft insbesondere deren visuelle, geometrische und semantische Ausprägung unter Berücksichtigung ihrer Anordnung zueinander (→ Räumlicher Kontext).

Übergeordnet kann dieses vielschichtige Zusammenspiel unterschiedlichster Faktoren als die Konsistenz eines Layouts wahrgenommen werden. Zwingende Bedingung ist, dass die Konsistenz eines geographischen Layouts durch das Syntheseverfahren sichergestellt werden muss. Andererseits würde dies zu inkonsistenten Anordnungen innerhalb von Layouts führen, die wiederum in der Wahrnehmung durch einen Betrachter als falsch empfunden würden.

So kann beispielsweise eine asphaltierte Straßenkreuzung mit drei Zugangswegen nur sinnvoll durch eine Kreuzung von drei Feldwegen ersetzt werden, ohne dass in diesem Fall die topologische Konsistenz des Layouts verletzt werden würde.

Analogien zwischen Layouts

Das Arrangement verschiedenster Strukturelemente mit unterschiedlichen Eigenschaften gibt als Ganzes den Kontext im Layout vor (s.o.). Dabei beeinflussen sich die Elemente gegenseitig. Die Substitution als Teil der Synthese kann daher nur erfolgen, wenn man die kontextuellen Beziehungen zwischen Elementen eines Layouts berücksichtigt.

Diese Beziehungen sind wesentlich durch die Eigenschaften der Strukturelemente bestimmt. Das bedeutet de facto, dass im Idealfall nur Strukturelemente mit großteils identischen Eigenschaften gegeneinander ausgetauscht werden können. Dabei lassen sich Strukturelemente durch eine Reihe solcher Eigenschaften eindeutig in einem Layout einordnen. Insbesondere sind diese Eigenschaften die:

¹Als Strukturelemente sollen alle räumlich-geometrischen Entitäten eines Layouts verstanden werden, also etwa Grundstücksflächen.



Abbildung 3.2: Zergliederung eines Quelldatensatzes bis auf Ebene der Grundstücke. Detaillierte Datensätze wie diese werden häufig von Städten und Kommunen verwaltet. (Quelle: <http://geoaccess.co.dutchess.ny.us/parcelaccess/>)

- Geometrische Struktur,
- Topologische Beziehung,
- Funktion (Semantik),
- und visuelle Ausprägung.

Die Komplexität kontextsensitiver Beziehungen zwischen Elementen, deren Erfassbarkeit sowie Übertragbarkeit auf andere Layouts setzt ein hohes Maß an Kontextwissen voraus. Es wird hier vorgeschlagen, diesen Kontext über Analogien zwischen Datensätzen verschiedener Layouts zu bestimmen. Eine solche Vorgehensweise legt ein Verfahren aus der Beispielbasierten Synthese nahe.

Werden Analogien zwischen Strukturelementen in verschiedenen Datensätzen auf Basis o.g. kontextrelevanter Eigenschaften gefunden, so können auch zur Substitution geeignete Elementpaare ermittelt werden. Anders ausgedrückt können zwei Elemente trotz unterschiedlicher Stilausprägung meist dann einander ersetzen, wenn sie eine vergleichbare Funktion (Semantik) im Layout wahrnehmen und weitere Randbedingungen (Form, topologische Relation etc.) im Layout befriedigen.

3.1.2 Fokus auf Flächensynthese

Diese Arbeit beschränkt sich auf die Synthese von Flächen mit hochstrukturierten Inhalten. In urbanen Layouts sind dies im Wesentlichen Nutzflächen, welche sich meist in Grundstücke mit darin befindlichen Gebäudestrukturen untergliedern lassen. Sie machen den flächenmäßig größten Anteil eines urbanen Layouts aus.

Es wird hier von einem im Quelldatensatz verfügbaren Detailgrad ausgegangen, der bis auf Ebene der Segmentierung dieser Nutzflächen in einzelne Grundstückspartellen reicht. Das bedeutet, dass Grundstücke als atomare Einheiten eines Layouts betrachtet werden, welche – der Allgemeingültigkeit wegen – im Weiteren gleichbedeutend als *Elemente*, *Formelemente* oder *Strukturelemente* bezeichnet werden. Die ihnen übergeordneten Nutzflächen, beispielsweise Häuserblocks, werden als so genannte *Container* bezeichnet.

Übersicht des Verfahrens

Es wird hier ein Ansatz verfolgt, bei dem alle, in einem Ziellayout vorhandenen, Container mit Elementen aus einem Quelldatensatz befüllt werden. Das bedeutet konkret, dass einzelne Grundstückspartellen in einer größeren Nutzfläche platziert werden. Der Quelldatensatz liefert somit die Detailebene, das Ziellayout wiederum gibt mit seinen Containern die übergeordnete Strukturebene vor.

Das vollautomatische Syntheseverfahren berücksichtigt dabei insbesondere die genannten kontextsensitiven Beziehungen zwischen Elementen, um so ohne Einflussnahme durch den Nutzer selbstständig konsistente Layouts zu produzieren.

Es umfasst hierbei zwei wesentliche Kernaspekte, welche gemeinsam die Synthese bilden. Zum einen ist dies die Gewinnung und Aufbereitung der Eigenschaften (Features), die für das Finden von Analogien zwischen Quell- und Zieldatensätzen relevant sind. Zum anderen sind dies Techniken, welche die effiziente und optimierte Befüllung der beliebigen Containerstrukturen mit Elementen ermöglichen. Beide Aspekte sind dabei eng miteinander verknüpft.

Die Bestimmung von Analogien erfolgt über sogenannte Features. Features stellen Eigenschaften von Elementen in vereinfachter, algorithmisch verwertbarer Form dar. Diese können durch spezielle Distanzmetriken miteinander verglichen werden. Die Bestimmung relevanter Featuremerkmale in Layouts stellt eine wesentliche Schwierigkeit dar und wird im Verlauf dieses Kapitels thematisiert. Sind Featuremerkmale einmal definiert, machen die Gewinnung (Extraktion) der Features einzelner Strukturelemente aus den Layouts und der Abgleich (Retrieval & Matching) der Features verschiedener Elemente untereinander einen Hauptanteil des Syntheseverfahrens aus. Abschnitt 3.2 des Kapitels befasst sich ausführlich mit der Featuregewinnung und den verwendeten Metriken zum Abgleich.

Die sogenannte strukturelle Synthese beschreibt jenen Prozess, der im Rahmen des Syntheseverfahrens die Elemente der Detailebene im Layout platziert und hierbei die Konsistenz eines synthetischen Layouts sicherstellt. Die strukturelle Synthese berücksichtigt u.a. die räumlich-geometrische Komponente und stellt außerdem Methoden zur Verfügung, mit denen der abstrakte Feature-Abgleich auf räumliche Dimensionen angewendet werden kann. Somit können neben geometrischen Randbedingungen auch weitere signifikante Merkmale von Layouts in die Synthese mit einbezogen werden. Die strukturelle Synthese wird in Abschnitt 3.3 im Detail erörtert.

3.2 Feature Extraktion und Abgleich

Die Feature Extraktion und das Matching stellen gemeinsam eine der zwei Säulen des Syntheseverfahrens dar. Unter Features werden alle für die Synthese relevanten Eigenschaften der geographischen Strukturelemente verstanden. Wie in der Einleitung (3.1.1) skizziert, dienen Features zur Erkennung von Analogien zwischen Elementen aus dem Ziellayout mit jenen aus dem Quelldatensatz. Analogien können theoretisch anhand zahlreicher Merkmale ausgemacht werden. Darunter Merkmale geometrischer, topologischer, semantischer und visueller Art.

Hier wird ein effizienter Abgleich (Retrieval) auf Basis einer überschaubaren Anzahl an Features vorgeschlagen – darunter u.a. die Geometrie der Elemente sowie strukturelle und visuelle Constraints an topologischen Kanten. Außerdem werden semantische Attribute berücksichtigt. Dazu ist dieser Teil des Kapitels wie folgt gegliedert.

In Abschnitt 3.2.1 werden sogenannte Strukturelle Features betrachtet. Dabei wird jeweils deren Bedeutung für die Synthese, deren Gewinnung (*Extraktion*) und eine Metrik zum Ähnlichkeitsvergleich (*Matching*) erläutert. Im zweiten Teil werden dann semantische Features ausführlich betrachtet und auf theoretische Ansätze zur Gewinnung, Aufbereitung und zum Abgleich dieser eingegangen (Abschn. 3.2.2). Zum Schluss wird auf eine der wichtigsten Eigenschaften in Layouts, den topologisch-semantischen Relationen, näher eingegangen.

3.2.1 Strukturelle Features

3.2.1.1 Geometrische Form

Die geometrische Form von Elementen ist für das Verfahren von großer Bedeutung. Wie in Abschnitt 3.3.2 genauer erläutert, werden einzelne Elemente sukzessive in einem Container platziert, um so ein Layout zu erzeugen. Um eine optimale Platzierung zu erreichen, muss im Algorithmus Kenntnis von der geometrischen Form aller Elemente bestehen, um diese hinsichtlich ihrer Passgenauigkeit in gegebenen Flächen evaluieren können.

Die Elemente liegen i.d.R. als geschlossene, einfache Polygone im Datensatz vor und können als n-Tupel wie folgt beschrieben werden:

$$P := (P_1, P_2, \dots, P_n), P_i \in \mathbb{R}^2, 1 \leq i \leq n. \quad (3.1)$$

Die Repräsentation der geometrischen Form als Streckenzug (Polygonzug) legt für das Matching eine Metrik basierend auf Kantenübereinstimmung nahe. Zu beachten ist, dass die Elemente unabhängig von ihrer Orientierung und ggf. Skalierung vergleichbar sein sollten, d.h. der Abgleich invariant zu affinen Transformationen sein muss. Hier wird *Geometric Hashing* (vgl. [Cohen and Guibas, 1997]) als eine effiziente Technik angewandt, um dies zu bewerkstelligen.

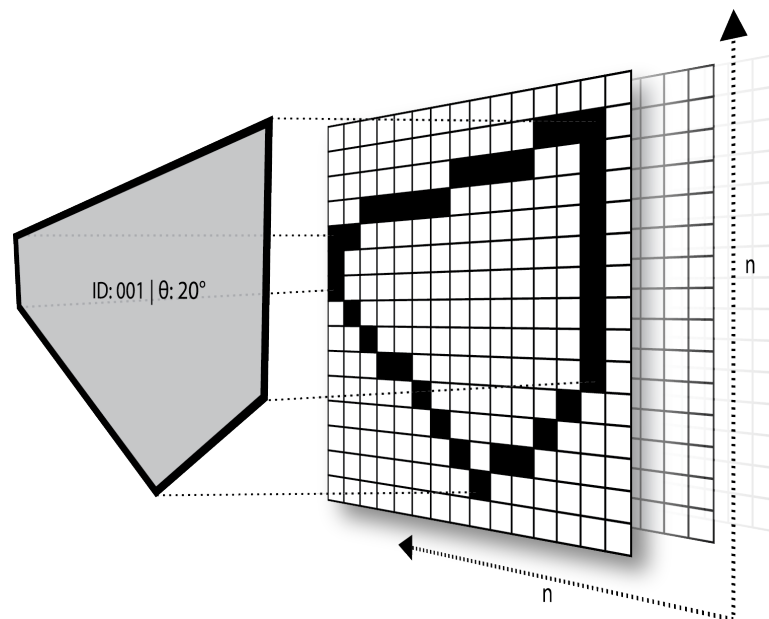


Abbildung 3.3: Geometric Hashing: Ein Polygon (grau) wird in diskreten Orientierungen auf ein $n \times n$ Raster gemapped.

Gegeben sei ein $n \times n$ Einheitsraster in Form einer Hashtabelle, das den Raum in zwei Dimensionen diskret quantifiziert. Jedes Elementpolygon im Quelldatensatz wird:

1. So proportional skaliert, dass es vollständig in das Einheitsraster passt.
2. Um w diskrete Winkelschritte rotiert. Für jeden Winkelschritt gilt: Jede Überschneidung einer Polygonkanten mit einem Rasterfeld generiert einen Eintrag in der Hashtabelle. Dieser Eintrag ist ein 2-Tupel und umfasst die ID² und die momentane Orientierung (Rotation als Winkel θ) des jeweiligen Elements.

In der Vorverarbeitungsphase wird für alle im Quelldatensatz vorhandenen Elemente (bzw. deren Polygone) eine solch diskrete Beschreibung ihrer geometrischen Form in allen erlaubten Orientierungen extrahiert und zum Zwecke des späteren Abgleichs gespeichert.

Distanzmetrik Das Matching erfolgt großteils analog zur Extraktion. Zuerst wird ein abzugleichendes Element (siehe Abschn. 3.3.2) P_q ebenfalls nach

²Jedes Element aus dem Quelldatensatz erhält eine eindeutige Identifikationsnummer (ID) in einer Datenbank mit der dieses eindeutig referenziert werden kann.

o.g. Schema im Einheitsraster registriert. Der Abgleich erfolgt dann, indem in allen von P_q berührten Rasterfeldern die dort in der Hashtabelle stehenden Einträge untersucht werden. Jeder registrierte Eintrag, als eindeutiges Paar $(ID, Orientierung)$, wird hierbei getrennt gezählt. Jenes Eintragspaar, das über alle untersuchten Rasterfelder am häufigsten auftritt, referenziert jenes Element, das die höchste geometrische Übereinstimmung besitzt – und zwar unter Berücksichtigung einer bestimmten Orientierung. Alternativ können k Elemente nach dem Grad ihrer Übereinstimmung sortiert zurückgeliefert werden.

Weil der Abgleich u.a. invariant zu Skalierungen ist, kann zusätzlich noch eine Gewichtungsfunktion ausgehend von dem zuvor angewendeten proportionalen Skalierungsfaktor hinzugezogen werden. Die Gewichtung benachteiligt dann bewusst Elemente, die zwar eine hohe Formähnlichkeit aufweisen, aber stark abweichende Größenverhältnisse besitzen.

3.2.1.2 Topologische Relationen

Die Topologie der Elemente liefert implizit eine Reihe wichtiger, meist semantischer Bezüge, die als Features extrahiert werden können. Voranging betreffen diese die strukturelle Kohärenz von Elementen gleicher oder verschiedener semantischer Klasse (siehe Abschn. 3.2.2.1)

Ein Beispiel sei etwa ein Grundstück an einer Straße, von dem man naturgemäß ausgehen kann, dass es eine Zufahrt besitzt (\rightarrow Semantische Beziehung). Betrachtet man beide Elemente getrennt voneinander, so ist diese semantische Beziehung in einem typischen Geodatensatz nicht zu finden, da es sich um eine **implizite** Beziehung handelt. Für das Syntheseverfahren muss daraus eine **explizite** semantische Beziehung erzeugt und als Feature gespeichert werden. Eine Analyse der räumlich-topologischen Relationen zwischen Strukturelementen liefert daher den Ausgangspunkt für umfassendere Betrachtungen, wie beispielsweise der Herstellung semantischer Analogien im räumlichen Kontext (vgl. *Geosemantik*).

Zur Beschreibung topologischer Relationen zwischen Strukturelementen wird eine Variation des *9-Intersection*-Modells [Egenhofer et al., 1993] verwendet. In diesem Modell werden die möglichen räumlichen Relationen zwischen zwei Entitäten (Punkte, Linien oder Flächen) A und B als die Überschneidungen der Hülle von A (∂A), dessen Inneres (A^0) und Äußeres/Komplement (A^-) und den Entsprechungen von B respektive in einer Matrix wie folgt festgehalten:

$$R_9(A, B) = \begin{pmatrix} \partial A \cap \partial B & \partial A \cap B^0 & \partial A \cap B^- \\ A^0 \cap \partial B & A^0 \cap B^0 & A^0 \cap B^- \\ A^- \cap \partial B & A^- \cap B^0 & A^- \cap B^- \end{pmatrix}. \quad (3.2)$$

Zur Bestimmung der Überschneidungen werden Hülle, Inneres und Äußeres als Punktmenge aus \mathbb{R}^2 erachtet. Für diese müssen jeweils die Schnitt-

Tabelle 3.1: Eine V9-Intersection Bit-Matrix.

$$\emptyset \cong 0, \neg \emptyset \cong 1$$

	∂A	A^0	A^-
∂B	1	0	1
B^0	0	0	0
B^-	1	0	1

mengen mit den entsprechenden Punktmengen der jeweils anderen geometrischen Entitäten bestimmt werden. Relevant ist dabei nur, ob die Schnittmengen leer (\emptyset), respektive nicht leer ($\neg \emptyset$) sind. Mit dieser Darstellung können folgende topologischen Beziehungen zwischen A und B kompakt erfasst werden: *disjunkt* (disjoint), *trifft* (meet), *gleich* (equal), *innerhalb* (inside), *enthält* (contains), *überlappt* (overlap), *deckt* (covers) und *ist bedeckt von* (covered-by).

Anders als bei [Egenhofer et al., 1993] wird als Äußeres einer Entität nicht deren vollständiges Komplement ($\mathbb{R}^2 \setminus (A^0 \cup A^-)$) betrachtet, sondern eine sie umschließende *Voronoi-Region* (siehe [Jun et al., 2000]). Damit können topologische Relationen sich nicht berührender Elemente genauer bestimmt werden als im klassischen 9-Intersection Modell. Es werden hier nur Elemente mit einer Voronoi-Distanz³ $vd(A, B) = 1$ bei der Erfassung topologischer Relationen weiter berücksichtigt. Also nur unmittelbare Nachbarelemente. Dies verringert die Zahl der Relationen eines Strukturelements deutlich, was den Abgleich effizienter macht.

Abbildung 3.4 zeigt zwei Elemente und deren Komplement in Form ihrer Voronoi-Regionen. Tabelle 3.1 zeigt ihre topologischen Relation $R_{V9}(A, B)$ nach dem 9-Intersection Modell in Form einer Bit-Matrix. Die vereinfachte Form, in welcher Relationen als Feature gespeichert werden, entspricht dann der *row-major* Darstellung einer solchen Matrix als Bitstring

$$R_{V9}(A, B) = (101000101). \quad (3.3)$$

Die topologischen Relationen werden dabei jeweils für die einzelnen Kanten eines Elements getrennt ermittelt.

Distanzmetrik Zwei topologische Relationen R_1 und R_2 lassen sich prinzipiell nur schwer vergleichen. In der einfachsten Betrachtung lässt sich eine Übereinstimmung nur dann ausmachen, wenn die Relationen vollkommen identisch sind. Diese Betrachtungsweise ist stark einschränkend und für das Syntheseverfahren ungeeignet. Vereinfachend wird angenommen, dass nach

³Die Voronoi-Distanz beschreibt den kürzesten Weg zwischen zwei Voronoi-Regionen, wobei das Betreten jeder Region gezählt wird (s. [Zhao and Chen, 1999]).

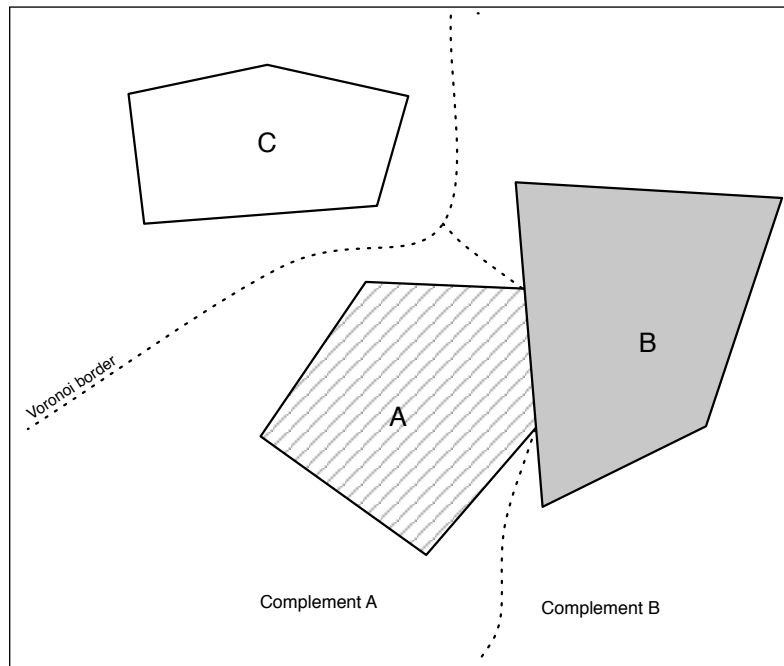


Abbildung 3.4: Veranschaulichung der topologischen Relation zweier Strukturelemente. Die Hüllen der Elemente A und B berühren sich hierbei. Die gepunkteten Linien stellen nach dem erweiterten V9-Intersection Modell dabei die Grenzen der Komplemente dar. Zusätzlich ist ein drittes – zu A und B disjunktes – Element C abgebildet.

dem 9-Intersection Modell nahe gelegene Relationen eine hohe Übereinstimmung in Bezug auf die Schnittmengen der einzelnen Komponenten (Hülle, Inneres und Äußeres) einer Entität aufweisen.

Zur Bestimmung der Distanz zweier Relationen werden demnach diese Übereinstimmungen errechnet. Die topologische Relation ist als einfaches Bitmasken-Feature gespeichert. Die Übereinstimmung lässt sich mit Anwendung eines XOR-Gatter der Form R_2 auf R_1 wie folgt bestimmen:

$$R_2 \text{ XOR } R_1.$$

Danach kann aus dem resultierenden Bitstring die Anzahl der gefundenen Übereinstimmungen mittels des *Hamming-Gewichts* [Hamming, 1950] errechnet werden. Der ermittelte Skalar d entspricht dabei der Distanz beider Relationen, wobei eine Distanz $d = 0$ einer identischen Relation entspricht. In Tab. 3.2 sei ein Beispiel zur Berechnung der Distanz zweier unterschiedlicher Relationen gegeben.

Das Matching topologischer Relationen unter Berücksichtigung der Semantik beteiligter Strukturelemente wird als zusammengesetzte Berechnung dieser und anderer (semantischer) Distanzmetriken realisiert. Näheres dazu

Tabelle 3.2: Einfache Distanzbestimmung zwischen zwei topologischen Relationen. Das ermittelte Hamming-Gewicht der beiden Bitstrings entspricht hierbei der Distanz d der beiden Relationen.

XOR	101000101	$f_{hamming}(101000100) = 3$
	000000001	
=	101000100	

findet sich unter Abschnitt 3.2.2.5 der semantischen Features.

3.2.1.3 Grenzfeatures

Globale Features eines Strukturelements zu extrahieren, ist nur in seltenen Fällen sinnvoll. Häufig ermöglicht die Zerlegung eines Elements bzw. des Polygons, das es repräsentiert, eine exaktere Bestimmung von Eigenschaften, vor allem in deren räumlichen Kontext. Die Trennung nach individuellen Objekten entspräche dabei einer bestmöglichen Segmentierung⁴.

Es wird hier jedoch auf eine detaillierte Zerlegung verzichtet. Stattdessen wird eine einfache Segmentierung auf Grundlage von Kanten vorgenommen. Kanten beschreiben den Grenzbereich eines Elements und spielen in räumlichen Layouts eine entscheidende Rolle, insbesondere beim Zusammenspiel mehrerer Elemente. Dort, wo sich Kanten verschiedener Elemente annähern oder berühren, müssen Features eine möglichst hohe Übereinstimmung besitzen. Nur so kann ein kontinuierliches und konsistentes Arrangement sichergestellt werden.

Kantenfeatures lassen sich am besten aus Regionen eines Elements bestimmen, da es sich in erster Linie um Low-Level Features handelt. Dazu wird ein Element folglich in eine diskrete Anzahl nicht-überlappender Regionen mit Kantenbezug zergliedert, welche als *Kantenregionen* bezeichnet werden.

Die Segmentierung eines beliebigen Polygons in Kantenregionen erfolgt hier über ein Verfahren zur Bestimmung des *Straight Skeleton* (nach [Oswin et al., 1995]). Durch uniforme Parallelverschiebung der Kanten um eine Distanz s_{offset} zu sich selbst ergibt sich für jede der n Kanten des Polygons je eine Region, deren Inneres näher an dieser Kante liegt als an jeder anderen Kante des Polygons (vgl. *Voronoi-Diagramm*). Diese räumliche Beziehung

⁴Zahlreiche Methoden zur Low-Level-Segmentierung sind bekannt, darunter solche basierend auf Farbsegmentierung (u.a. [Swain and Ballard, 1991]), Kantenerkennung (u.a. [Ma and Manjunath, 1997]), Modellbasierte Verfahren (u.a. [Hoogs, 1996]) und viele mehr. Die vielzähligen, unterschiedlichen Objektstrukturen in Elementen machen eine gute Segmentierung generell schwierig. Zudem sind für manche Segmentierungsalgorithmen (etwa Modellbasierte Verfahren) a priori Zusammenhänge vonnöten, die häufig im Datensatz nicht zur Verfügung stehen.

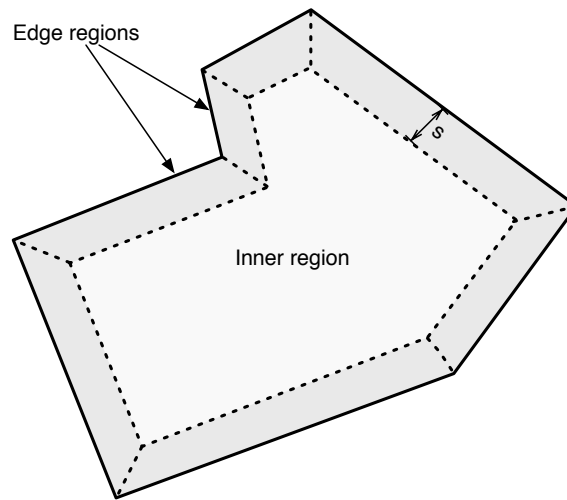


Abbildung 3.5: Segmentierung in Kantenregionen.

macht ersichtlich, warum eine Region je einer bestimmten Kante zugeordnet werden kann. Abb. 3.5 zeigt eine Zerlegung für ein beliebiges Polygon. Insgesamt ergeben sich $n + 1$ Regionen, wobei eine Region der Zentralregion des Polygons entspricht. Die aus einer Kantenregion ermittelten Features werden hierbei in den Featurevektor der jeweiligen Kante übernommen.

Farbregionen Farbe und Textur bilden zwei wichtige Faktoren zur Bestimmung von Kontinuität bzw. Kohärenz an topologischen Kanten. Meist weisen Grenzen benachbarter Flächen ein bestimmtes Schema an farblicher oder textueller Konsonanz oder aber Dissonanz auf. Über diese Faktoren lässt sich – vereinfacht angenommen – bestimmen, ob die Anordnung von Elementen plausibel ist. Ein Beispiel: Eine Rasenfläche eines Grundstücks weist einen typischerweise sehr hohen Grünwert auf, Gebäude tun dies i.d.R. nicht. Wenn zwei Elemente zusammengeführt werden, deren Kanten einen ähnliche Farbverteilung (hier grün) aufweisen, kann zumindest von einer ästhetischen Kontinuität ausgegangen werden. Im Idealfall stellt dies sogar eine semantische Kohärenz dar – d.h. eine Rasenfläche geht in eine andere Rasenfläche über.

Um die Farb-Features zu bestimmen, wird hier auf das in diskrete Kantenregionen segmentierte Polygon zurückgegriffen (siehe Abschn. 3.2.1.3). Ähnliche Ansätze werden im *Content-based Image Retrieval (CBIR)* verwendet (vgl. [Stricker et al., 1996]). Unter den vielzähligen Möglichkeiten zur Ermittlung von Farb-Feature-Deskriptoren, darunter *Color Moments* [Stricker and Orenge, 1995], *Color Sets* [Smith and Chang, 1996] und einfacher Farbmittelwerte, haben sich Farbhistogramme bewährt. Deshalb wird hier das Histogramm der Farben einer Kantenregion ermittelt. Das Histogramm be-

schreibt die statistische Häufigkeit von Farben einer Region in diskret quantifizierten Farbwerten (Klassen). Die absolute Häufigkeit einer quantifizierten Farbe m in einem Bild mit Größe $W \times H$ lässt sich formal beschreiben als

$$h_c[m] = \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} \delta(I^q(i, j), y_m), (m = 1, \dots, M), \quad (3.4)$$

wobei $I^q(i, j)$ dem Farbwert eines Pixels an Stelle i, j entspricht und y_m der m -ten Farbklasse (siehe [Castelli and Bergman, 2002]). Um die Histogramme verschiedener, unterschiedlich großer Regionen vergleichbar zu machen, muss zudem eine Normalisierung der Histogramme durchgeführt werden.

In der Vorverarbeitungsphase werden für jedes Element die normalisierten Histogramme seiner n Kantenregionen errechnet. Diese werden je logisch verknüpft mit der zugehörigen Kante als Teil deren Featurevektors in einer Feature-Datenbank gespeichert.

Distanzmetrik Durch das Anwenden einer Distanzfunktion auf zwei Histogramme, lassen sich Regionen hinsichtlich ihrer farblichen Übereinstimmung bewerten. Eine gängige Metrik stellt die Euklidische Distanz der Histogramme dar:

$$D_2(q, t) = \sum_{m=0}^{M-1} [h_q(m) - h_t(m)]^2. \quad (3.5)$$

Dabei entsprechen $h_q(m)$ und $h_t(m)$ jeweils der Häufigkeit der m -ten Farbklasse in den beiden Regionen q und t . Die Differenz ergibt eine positive Maßzahl und lässt sich somit leicht in diesem Verfahren einsetzen. Neben vielzähligen anderen Metriken (*Quadric Distance Measures*, *Hamming Distanz* etc.) gilt die hier verwendete quadrierte, Euklidische Histogrammdistanz als äußerst effektiv und performant zugleich (s. [Castelli and Bergman, 2002, S. 308]).

3.2.2 Semantische Features

Die Verfügbarkeit von semantischen Merkmalen (Features) und häufig deren Relationen sind neben der Strukturierung ein wesentliches Unterscheidungsmerkmal von Geodaten zu herkömmlichen Bilddaten. Durch diese kann die logisch-semantische Integrität von Elementen geprüft und ggf. korrigiert werden (vgl. [ISO, 2002]).

Ob zwei Objekte (z.B. Grundstück und Straße) gleicher oder unterschiedlicher semantischer Bedeutung in eine bestimmte räumliche Anordnung zueinander gebracht werden dürfen, ist in der Beispielbasierten Synthese durch den Quelldatensatz implizit vorgegeben. Findet sich dort eine spezifische Konstellation von Objekten, so soll diese im Ziellayout ebenfalls gestattet sein. Des Weiteren kann das Ziellayout explizit bestimmte Konstellationen

Tabelle 3.3: Kleine Auswahl semantischer Daten aus der Geokarte in Abb. 3.6 (Quelle: <http://openstreetmap.org>)

<i>Object ID</i>	Attribute	Value
33740973	highway	secondary
22795831	landuse	residential
80599751	landuse	forest
413481562	amenity	fountain

erwirken⁵. Der Einbezug semantischer Daten in den Synthesealgorithmus ist hier daher ein wichtiger Aspekt. Genauer wird die Verwendung semantischer Daten in Abschnitt 3.3.3.2 des Synthesealgorithmus erläutert.

An dieser Stelle soll die Erfassung (Extrahieren), Aufbereitung (Transformieren) und Verarbeitung (Abgleichen) der semantischen Daten an sich thematisiert werden. Die Problemstellung umfasst dabei *a)* woraus eine semantische Zuordnung von Objekten geschlossen werden kann, *b)* wie zwei Objekte in ihrer semantischen Bedeutung verglichen, sowie *c)* wie eine semantische Relation, also die Beziehung zwischen zwei Objekten, erfasst und verarbeitet werden kann.

Abb. 3.6 zeigt einen Ausschnitt einer indexierten Karte. Tabelle 3.3 zeigt den Inhalt derselben Karte in Form von Metadaten⁶.

Es soll hier von semantischen Merkmalen im Datensatz in Form von Attribut-Wertpaaren ausgegangen werden. Diese können als geordnetes Paar vom Schema

$$R = (\textit{attribute}, \textit{value}). \quad (3.6)$$

formuliert werden.

3.2.2.1 Klassifizierung und Matching

Der Schluss, ob zwei Objekte eine semantische Gemeinsamkeit haben, lässt sich nur bewerkstelligen, wenn *1)* beide Objekte mindestens ein identisches Attribut besitzen und *2)* die Werte des Attributs aus einer wohldefinierten

⁵Beispielsweise durch einen Container, der eine bestimmte Klasse an Elementen voraussetzt.

⁶Metadaten sind klassischerweise Attribute in Textform, die u.a. dazu genutzt werden können, einem Objekt (hier: einer geographischen Form) eine semantische Bedeutung zuzuweisen. Entweder liegen diese als singuläre, ungeordnete Attribute (sog. Tags), als Attribut-Wertpaare oder als semantische Tripel vor. Erstere sind stark eingeschränkt in ihrer Fähigkeit, Semantik zu transportieren. Tripel liefern die exakteste semantische Beschreibung, da sie eng mit der Prädikatenlogik verknüpft sind. Dennoch finden sich in den meisten Geodatensätzen heute primär Attribut-Wertpaare. Eine ausführliche Analyse von Metadaten findet sich in [Lobin and Lemnitzer, 2004].



Abbildung 3.6: Ausschnitt einer Karte eines indextierten Geodatenatzes (Quelle: <http://openstreetmap.org>).

Wertemenge stammen. Meist besitzen Objekte mehrere Attribute, die es erst in Kombination ermöglichen, ein Objekt semantisch sinnvoll zuzuordnen.

Zur Vereinfachung der Ähnlichkeitsbestimmung zweier semantischer Objekte, wird vorzugsweise keine Distanzmetrik auf Attributpaare selbst angewendet. Stattdessen werden die Objekte entsprechend ihrer Attribut-Wertpaare k diskreten *Klassen* zugeordnet. Dies geschieht durch einen *Klassifikator*, der stark an das semantische Vokabular des verwendeten Datensatzes gebunden ist. Deshalb wird hier nicht näher auf eine ggf. komplexere Klassifikation eingegangen. Die einfachste Annahme ist, dass eine sog. Attributsdomäne existiert, die anhand eines Attribut-Wertpaares eine plausible Klassifizierung ermöglicht. Das bedeutet, dass ein Attributwert eines bestimmten Attributs äquivalent zu einer Klasse erachtet wird.

Mindestens eine Attributdomäne, die eine solche Zuordnung erlaubt, findet sich in allen vollständigen Geodatenätzen wieder. Gängigerweise ist das als *Typattribut* bezeichnete Merkmal eines Elements Teil einer geeigneten Attributdomäne. Ein Typattribut nimmt einen von k diskreten Werten einer Attributwertemenge M_{type} an, wobei beispielsweise

$$M_{type} = \{highway, country\ road, industrial\ site, residential\ estate\} \quad (3.7)$$

gilt. Jeder der möglichen Attributwerte entspricht hierbei einer Klasse an realweltlichen Objekten (Konzepten), deren Semantik sich aus ihrer Bezeichnung und dem menschlichen Kontextwissen erschließt. Wie in diesem Beispiel

ersichtlich, sind Klassenbezeichner meist in natürlicher Sprache gehalten und besitzen im Gegensatz dazu für einen Algorithmus keinen weiteren semantischen Kontext. Es kann von einer horizontalen Klassenstruktur gesprochen werden. Eine Ähnlichkeitsmetrik beläuft sich dabei i.d.R. auf einen trivialen String-Vergleich, bei dem in diesem Anwendungsfall kein sinnvolles Distanzmaß existiert, mit Ausnahme eines booleschen Vergleichs. Stimmen zwei Klassenbezeichner syntaktisch vollständig überein, sind die Klassen identisch, in jedem anderen Fall sind sie verschieden.

3.2.2.2 Semantische Taxonomie

Eine horizontale Klassenstruktur schränkt den *Matchingprozess* insofern stark ein, dass dieser nur exakte Übereinstimmungen semantischer Klassen erkennen kann. Es ist daher von großem Nutzen, die Attributklassen in eine hierarchische Relation zu überführen, sofern eine solche existiert. Der Einfachheit halber wird nach einer monohierarchischen Struktur, also einer Taxonomie, der Elemente gesucht. In einigen GIS ist diese Struktur bereits explizit im Attributesystem modelliert⁷.

Im Fall, dass es keine weitere semantische Relation zwischen einzelnen Werten der Definitionsmenge gibt, wäre der Feature-Abgleich – wie bereits erwähnt – darauf begrenzt, exakte Übereinstimmung zu finden. Existierende hingegen eine Beziehung, die über diese Definitionsmenge hinausgeht, so könnte im Fall, dass kein exakter Treffer im Matchingprozess gefunden wird, auf eine nächst bessere Übereinstimmung zurückgegriffen werden. Mit dem nötigen Kontextwissen kann intuitiv eine Taxonomie für die Menge in 3.7 wie folgt formuliert werden:

$$M_{street} = \{highway, country\ road\} \quad (3.8)$$

$$M_{landuse} = \{industrial\ site, residential\ estate\}. \quad (3.9)$$

Dabei stellen $M_{street} \subset M_{type}$ und $M_{landuse} \subset M_{type}$ je echte Teilmengen von M_{type} dar. Die beiden Teilmengen generalisieren dabei die Klassen ihrer Elemente. Diese Generalisierung bedeutet die Herstellung einer semantischen ‚ist-ein‘-Relation (vgl. *Hyponymie*). Dabei sind die neu eingeführten Mengen auch Oberklassen ihrer jeweiligen Elemente. Klasse C_{street} beispielsweise generalisiert bzw. verallgemeinert die Elementklassen *highway* und *country road* zu einer abstrakten Straßenklasse. Die Elemente stellen im Gegenzug eine Spezialisierung ihrer Oberklasse dar.

Eine anschaulichere Darstellung sei in Abb. 3.7 in Form eines relationalen Graphen gegeben. Zur Zuordnung von Klassen in eine Taxonomie, muss ein hierarchisches *Klassifikationsschema* für den Klassifikator existieren. Die gemeinsame Basisklasse aller Elemente wird stets als C_{object} bezeichnet.

⁷Im *OpenStreetMap* Projekt werden Kartenfeatures über ihren Typattributsschlüssel explizit einer semantischen Oberklasse zugeordnet. Die Zuordnung der spezifischen Klasse erfolgt über den Attributwert.

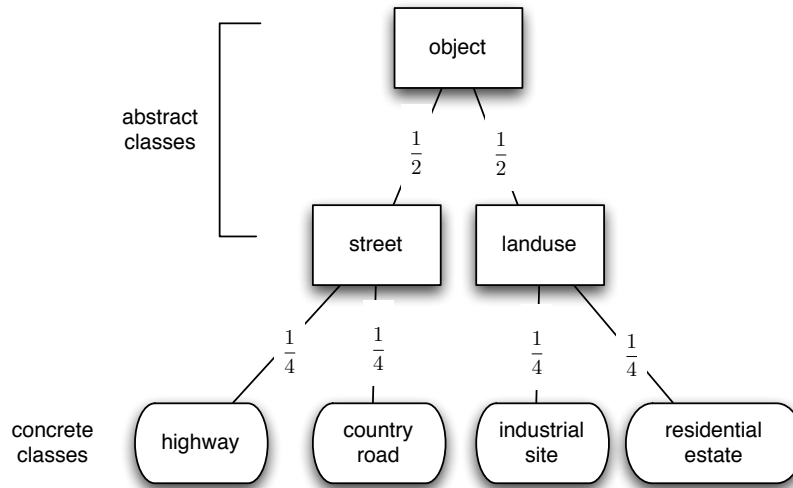


Abbildung 3.7: Semantische Relationen als Graph

3.2.2.3 Berechnung semantischer Nähe

Durch die Taxonomie ergeben sich neue Möglichkeiten einer geeigneten Distanzmetrik. Die Metrik dient dabei zur Bestimmung der semantischen Nähe zweier Objekte.

In der Vorverarbeitungsphase wird – ausgehend von einem datensatzgebundenen Klassifikationsschema – ein gerichteter, relationaler Graph der Klassenhierarchie ermittelt. Tatsächliche Klassen (mit einer Repräsentation im Datensatz) befinden sich ausschließlich an den Blättern des Graphenbaumes. Alle weiteren Knoten sind abstrakte Klassen. Jeder Übergang einer Hierarchieebene zur nächsten wird dabei mit jeweils der Hälfte der Kosten zur nächst höheren Ebene gewichtet. Dadurch wird der zunehmenden Verfeinerung und somit größeren semantischen Nähe der Klassen in tieferen Hierarchieebenen Rechnung getragen. Abb. 3.7 zeigt u.a. dieses Gewichtungsschema.

Zur Bestimmung der semantischen Nähe zweier beliebiger Klassen C_1 und beispielsweise C_3 muss der kostengünstigste Weg zwischen beiden Knoten im Graph errechnet werden. In der Praxis hingegen wird aber i.d.R. eine nächst beste semantische Übereinstimmung zu einer Klasse C_i gesucht. Dazu muss der kürzeste Weg zu einem beliebigen anderen Blattknoten ermittelt werden. Die Klasse des gefundenen Blattknotens entspricht somit nach o.g. Definition jener mit der höchsten semantischen Übereinstimmung, wobei es andere Klassen mit gleich großer Übereinstimmung geben kann. Zum Ermitteln des kürzesten Weges kann der Algorithmus von Dijkstra verwendet werden [Dijkstra, 1959].

Die Qualität der Ergebnisse ist stark an das zugrundeliegende Klassifi-

kationsschema gekoppelt.

3.2.2.4 Relationen

Neben der Taxonomie (,ist-ein'-Beziehung) können weitere Relationen zwischen Objekten semantischer Klassen existieren. Im Kontext von Geolayouts lassen sich vielfältige solcher Relationen erfassen, wobei insbesondere topologisch-semantische Relationen von Interesse sind (s. Abschn. 3.2.2.5).

Das Beschreiben semantischer Beziehungen ist ein weitreichendes Feld und zahlreiche Datenmodelle und Sprachen existieren (u.a. RDF [Manola and Miller, 2004], OWL [Bechhofer et al., 2004] u.v.m.). Hier soll ein allgemeines Schema zur Beschreibung einer semantischen Relation gegeben werden, ohne ein bestimmtes Modell zu bevorzugen. Dazu wird ein Tripel bestehend aus (s, p, o) verwendet, wobei s das Subjekt, p das Prädikat und o das Objekt der Relation ist.

Als einfaches Beispiel dient ein Grundstück eines Wohngebietes, das mit einer Grundstückskante an eine Ortsstraße grenzt. Die Relation, die sich daraus für diese Kante ergibt, lautet etwa wie folgt:

$$R = (\textit{residential estate}, \mathbf{meets}, \textit{town road}). \quad (3.10)$$

Subjekt und Objekt stammen hier aus der Menge an Elementklassen. Das Prädikat (*meets*) beschreibt die konkrete Beziehung zwischen Subjekt und Objekt und stammt hier aus der Menge der Prädikate für topologische Relationen.

Distanzmetrik bei Relationen Die Distanz zwischen semantischen Relationen kann ebenfalls durch die Bestimmung semantischer Nähe berechnet werden. Für jedes der drei Elemente einer Relationsbeschreibung (Subjekt, Prädikat und Objekt) wird die Distanz getrennt berechnet. Über die Summe der einzelnen Distanzen kann dann die Gesamtdistanz zweier semantischer Relationen ermittelt werden.

$$D(R_1, R_2) = D(s_1, s_2) + D(p_1, p_2) + D(o_1, o_2). \quad (3.11)$$

Dabei stehen s_1 und s_2 für die Subjekte, p_1 und p_2 für die Prädikate, sowie o_1 und o_2 für die Objekte der Relationen R_1 und R_2 . Als Voraussetzung gilt, dass Subjekt, Prädikat und Objekt beider Relationen jeweils aus den gleichen Definitionsmengen stammen müssen, ansonsten ist deren Distanz undefiniert.

3.2.2.5 Topologisch-semantische Relationen

Als topologisch-semantische Relationen werden hier die topologischen Beziehungen zwischen Strukturelementen unter Berücksichtigung ihrer Semantik

bezeichnet. Für die Synthese urbaner Layouts ist eine solche Betrachtungsweise unerlässlich. Die Erfassung topologischer Beziehungen im Raum alleine ist unvollständig. Denn sie vernachlässigt die Bedeutung der in Relation stehenden Elemente. Ein Haus mit Anschluss an eine Straße wäre dabei identisch zu einem Haus an einem Fluss. Straße und Fluss wären als Linien, ein Haus entsprechend als Fläche im Datensatz beschrieben. Die Relation würde sich nur aus der Topologie dieser geometrischen Primitiven zueinander ergeben.

Aus diesem Grund werden topologische Beziehungen zwischen Strukturelementen in Layouts als semantische Relation (siehe oben) erfasst. Dabei wird als Subjekt die *semantische Klasse* des Elements hergenommen, von welchem die Betrachtung ausgeht. Als Objekt wird die semantische Klasse des Elements festgelegt, das zu ihm in Relation steht. Sowohl Subjekt als auch Objekt stammen aus der Menge Elementklassen (M_{object}), welche im Klassifikationsschema definiert sind. Die Prädikate werden nach dem *g-Intersection-Modell* aus der Menge topologischer Relationen (M_{gI}) entnommen. Die topologisch-semantische Relation zweier Strukturelemente S_A und S_B lässt sich dann verallgemeinert ausdrücken als

$$R(S_A, S_B) = (class(S_A), p, class(S_B)), class(S_A, S_B) \in M_{object}, p \in M_{gI}. \quad (3.12)$$

Wie in Abschnitt 3.2.1.2 erläutert, werden topologische Relationen jeweils auf Kantenbasis ermittelt. Darauf aufbauend werden topologisch-semantische Relationen für jede Kante eines Elements getrennt erfasst und als Feature der jeweiligen Kante gespeichert.

Distanzmetrik Die Distanzbestimmung erfolgt entsprechend dem Matching semantischer Relationen. Die Distanzen zwischen Subjekten und Objekten zweier zu vergleichender topologisch-semantischer Relationen ergeben sich aus deren Distanz innerhalb der Klassen-Taxonomie. Die Distanz des topologischen Prädikats wird nach der in Abschnitt 3.2.1.2 geschilderten Distanzmetrik bestimmt. Eine Gesamtdistanz lässt sich dann aus der Summe der einzelnen Teildistanzen ermitteln.

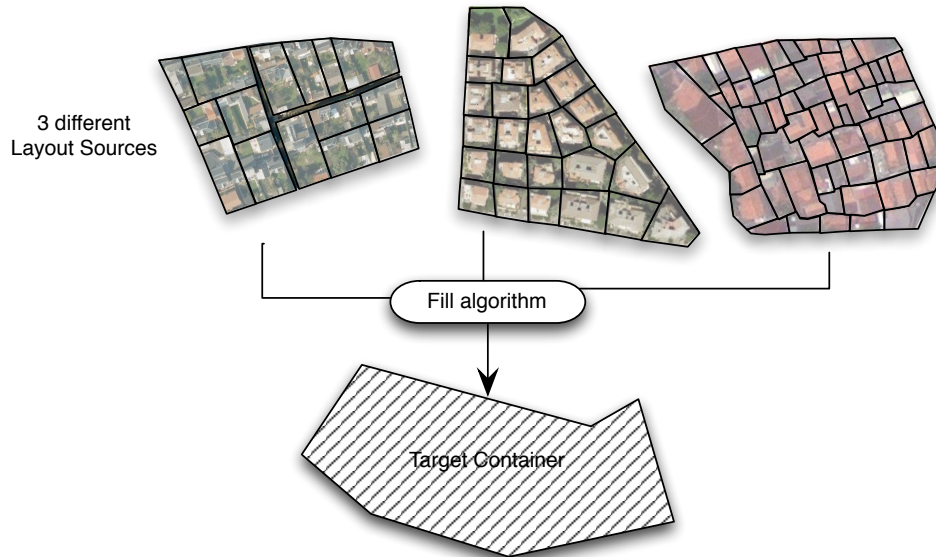


Abbildung 3.8: Die Segmentierung drei unterschiedlicher Container aus verschiedenen Quelldatensätze bei Anwendung auf einen Zielcontainer. Die unterschiedlichen Muster in der Anordnung der Grundstücke machen eine vorab Segmentierung des Zielcontainers nach festem Schema so gut wie unmöglich. Stattdessen empfiehlt sich eine Befüllung des Zielcontainers nach dem hier beschriebenen Verfahren.

3.3 Strukturelles Syntheseverfahren

Das Strukturelle Syntheseverfahren beschreibt den eigentlichen Prozess zur Inferenz neuer Layouts anhand von Beispieldaten und baut auf den in Abschnitt 3.2 ausgeführten Techniken auf.

In Abbildung 3.8 ist u.a. beispielhaft die im Quelldatensatz vorhandene Segmentierung mehrerer Container (hier: Häuserblocks) in eine Vielzahl an Formelementen (hier: Grundstücksparzellen) zu sehen. Die zu Beginn von Kapitel 3 erläuterte Trennung von Ebenen in Layouts weisen darauf hin, dass ein Container als Teil einer vorgegebenen strukturellen Ebene erachtet werden kann. Dessen Inhalt wird allerdings der visuellen Detailebene zugeordnet. Die strukturelle Synthese beinhaltet vereinfacht gesagt die intelligente Überführung der Detailstrukturen aus einem Quelldatensatz in das Ziellayout.

Das Problem bei der strukturellen Synthese von Flächen besteht folglich darin, einen beliebigen im Ziellayout vorhandenen Container (Zielcontainer) mit Elementen aus dem Quelldatensatz zu befüllen, sodass sich ein logisches und ästhetisch stimmiges Arrangement der Elemente ergibt, das am Quell-

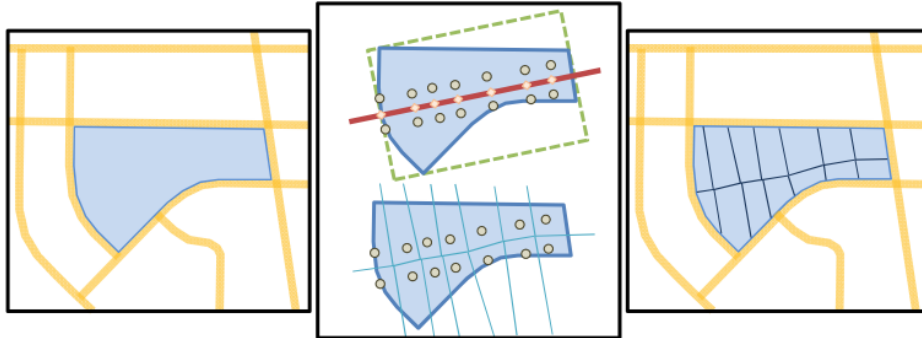


Abbildung 3.9: Segmentierung eines Häuserblocks nach Aliaga et al. Komplexere Grundstücksformen lassen sich beispielsweise nicht mit dieser Segmentierung in Einklang bringen. (Quelle: [Aliaga et al., 2008, S. 5])

datensatz gemessen werden kann.

Aliaga et al. segmentierten dazu den Zielcontainer in einzelne, meist rechteckige Parzellen nach einem einheitlichen Schema. Dazu sampeln sie entlang der approximierten Medialen Achse des Containers mehrere Punkte und legen die Grundstücke als die sich daraus ergebenden Voronoiregionen fest (s. [Aliaga et al., 2008]). Insbesondere sind in ihrem Ansatz nur jeweils höchstens zweireihige Grundstücksanordnungen vorgesehen. Das führt dazu, dass jedes Grundstück ein Randelement eines Containers darstellt. Im Kontext städtischer Layouts bedeutet das, dass Grundstücke einen zwingenden Straßenzugang besitzen müssen. Dieser Ansatz führt nicht nur zu einer starren⁸, sich wiederholende Segmentierung, sondern vernachlässigt auch die Vielfalt an Grundstücksformen und deren unterschiedliche Anordnungen (s. Abb. 3.9), die so möglicherweise im Quelldatensatz vorhanden sind.

Die strukturelle Synthese in diesem Verfahren hingegen basiert grundlegend auf den von Kim und Pellacini beschriebenen *Jigsaw Image Mosaics* [Kim and Pellacini, 2002]. Anders als bei Aliaga et al. wird beim *Image Mosaics* Verfahren keine a priori Segmentierung vorgenommen, sondern Flächen mittels der Formelemente segmentiert, die tatsächlich im Quelldatensatz vorhandenen sind. Da die Formelemente und der Zielcontainer per se keine Beziehung besitzen, muss für die Befüllung ein minimales Ziel definiert sein.

Für ein konsistentes Layout lässt sich intuitiv festhalten, dass *a)* Lücken zwischen und Überlappungen von Strukturelementen im Container und zum Container selbst zu vermeiden sind, *b)* die räumliche Anordnung wichtige logisch-semantische Beziehungen zwischen Strukturelementen keinesfalls ver-

⁸ Aliaga et al. verwenden im Weiteren ein Warpingverfahren, um Bildfragmente in die Parzellen einzupassen. Das führt zu oft merkbaren Verzerrungen, abhängig davon, wie stark sich die Geometrien der Parzellen zwischen Quell- und Zieldatensatz unterscheiden.

letzen darf, sowie c) als Umkehrschluss die Anordnung räumliche (und somit implizit auch die semantische) Beziehungen von Elementen im Quelldatensatz möglichst akkurat replizieren sollten.

Die genannten Minimalanforderungen legen dabei zwei Teilprobleme offen. Zum einen der rein geometrische Aspekt. Dieser fordert eine im euklidischen Raum verankerte konsistente topologische Anordnung der Elemente. Zum anderen der semantische Aspekt. Er gibt einen Kontext auf Metaebene vor und erlegt dadurch weitere Constraints für die Anordnung von Elementen auf.

Der Synthesealgorithmus arbeitet, indem er sukzessive Elemente (z.B. Grundstücksflächen) in einem Container (z.B. Häuserblock) platziert. Bei jedem Element wird geprüft, ob dessen Platzierung stimmig ist, d.h. ob sämtliche Randbedingungen erfüllt werden. Die Randbedingungen ergeben sich dabei sowohl aus dem momentanen Zustand des Zielcontainers als auch aus den Constraints, die sich aus dem Quelldatensatz ergeben. Neben lokalen Randbedingungen mit Auswirkung auf einzelne Elemente wird zudem auch das Ergebnis der Platzierung aller Elemente als Randbedingung aufgefasst. Der Algorithmus optimiert auch dahingehend und sucht nach einer Konfiguration, die ein näherungsweise Optimum darstellt, sowohl lokal, als auch global (auf Containererebene).

Details des Platzierungsverfahrens der Elemente werden nun ausführlich in Abschnitt 3.3.2 behandelt und widmen sich dem rein geometrischen Aspekt der Flächensynthese. Die Berücksichtigung weiterer struktureller Randbedingungen im Layout wird dann in Abschnitt 3.3.3.2 thematisiert. Dazu wird das Zusammenspiel mehrerer Elemente eines Layouts untersucht. Hierzu gehört deren Einflussnahme auf ihre nähere Umgebung und die damit verbundenen Metriken zur Bestimmung von Einflussgewichtung in lokalen Nachbarschaften (Abschn. 3.3.3). Zudem wird die eigentliche Vorgehensweise des Feature-Matchings zwischen Elementen betrachtet (Abschn. 3.3.3.2) und auf zwei relevante Eigenschaften (Features) zur Sicherstellung eines konsistenten Layouts näher eingegangen.

Zuerst wird jedoch eine Einführung in den Aufbau des Verfahrens gegeben, das in den o.g. Abschnitten angewendet wird. Darunter ist die Verwendung eines *Energie-basierten* Lösungsansatzes (Abschn. 3.3.1), um die unterschiedlichen Einflussfaktoren effizient zu beurteilen und optimale Konfigurationen in der Flächensynthese zu finden.

3.3.1 Energie-basiertes Lösungsverfahren

In der Einleitung wurde verdeutlicht, dass die Synthese einer Reihe unterschiedlicher Randbedingungen unterliegt, darunter solche geometrischer, topologischer als auch semantischer Natur. Da nie alle Randbedingungen in vollem Umfang erfüllt werden können, wird hier zumindest eine Optimierung dieser angestrebt.

Zur Bestimmung eines erzielbaren Optimums bietet sich ein Energiebasiertes Lösungsverfahren an (s. auch [Kim and Pellacini, 2002]). Die Platzierung jedes Elements im Container kann dabei über einen Energiewert beurteilt werden. Die Summe aller Energiewerte der Elemente stellt die Gesamtenergie einer Konfiguration des Layouts dar. Der Energiewert ermittelt sich dabei über eine Energiefunktion, welche sich aus der Summe gewichteter Energieterme ergibt.

$$E = w_C \cdot E_C + w_G \cdot E_G + w_O \cdot E_O + w_D \cdot E_D \quad (3.13)$$

Formel 3.13 zeigt die ursprüngliche Energiefunktion, wie sie von Kim und Pellacini vorgeschlagen wurde. E_C , E_G und E_O entsprechen dabei einzelnen Energietermen, welche jeweils durch einen Faktor w_i zueinander in ein Verhältnis gesetzt werden. Unter einer optimalen Konfiguration muss jene Anordnung der Elemente innerhalb des Containers verstanden werden, welche die Summe der Energiewerte aller Elemente minimiert. Sei X die Anzahl Elemente im Container und M die Zahl verwendeter Energieterme je Element, dann ist

$$E = \sum_{x=0}^{X-1} \sum_{m=0}^{M-1} (w_m \cdot E_m) \quad (3.14)$$

die im Container auftretende Gesamtenergie. Aus dieser Betrachtung lässt sich folgern, dass jedes Element so platziert werden soll, dass sein eigener Energiewert minimal wird.

Der allgemeine Aufbau dieses Energiebasierten Verfahrens erlaubt es, Energiefunktionen mit beliebigen Energietermen zu konstruieren, ohne dabei das Lösungsverfahren selbst zu verändern. Dies macht es flexibel und mächtig zugleich. In Formel 3.13 beschreibt Energieterm E_C beispielsweise die Übereinstimmung des Formelements mit dem Container hinsichtlich der Farbe⁹. E_G und E_O bewerten den Abstand (*gap*) respektive die Überlappung (*overlap*) des Formelementes mit anderen Elementen oder mit der Hülle des Containers. Term E_D gibt den Grad der geometrischen Deformation (vgl. *Warping*) an, die notwendig wäre, um Abstände bzw. Überlappungen weiter zu minimieren und so kompaktere Resultate zu erzielen.

Im Weiteren werden zum Teil abweichende Energieterme verwendet und das ursprüngliche Verfahren so erweitert, dass eine Anwendung auf geographische Layouts möglich wird. In Abschnitt 3.3.2.3 wird die Energiebestimmung im Platzierungsverfahren genauer erläutert. In Abschnitt 3.3.3 werden die Energieterme weiterer Randbedingungen im Detail vorgestellt.

⁹Bei *Jigsaw Image Mosaics* ist der Container eine Rastergrafik, sodass jedem diskreten Punkt eine Farbe zugeordnet werden kann.

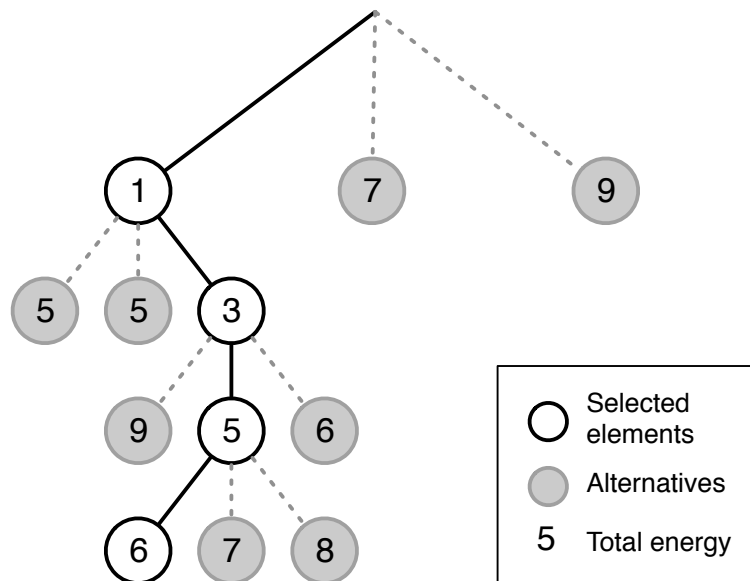


Abbildung 3.10: Ein Energiebaum des Verfahrens. Schwarz markiert jeweils die tatsächlich platzierten Elemente. In Grau jeweils die zur Platzierung in Betracht gezogenen Alternativen mit höherer Platzierungsenergie.

3.3.1.1 Optimale Layoutkonfiguration

Die Flächensynthese setzt neben der Energieminimierung der einzelnen Elemente im Layout auch eine optimale, möglichst vollständige Flächennutzung voraus. Eine näherungsweise optimale Flächennutzung kann erst nach Platzierung aller Elemente bestimmt werden, sodass es nötig sein kann, durch verschiedene Platzierungskonfigurationen iterieren zu müssen, bis ein Optimum gefunden wird.

Zu diesem Zweck wird jeder Schritt im Platzierungsvorgang in einem ‚Energie‘-Suchbaum (*Energiebaum*) festgehalten. Dazu werden je Platzierungsschritt jeweils die zur Platzierung in die engere Auswahl genommenen Elemente (siehe Abschnitt 3.3.2) als Knoten in den Energiebaum übernommen. Ihre Platzierungsenergie wird zur Energie des Elternknotens addiert und im jeweiligen Knoten vermerkt (vgl. [Kim and Pellacini, 2002]). So kann an jedem Knoten die momentane Gesamtenergie des Layouts ermittelt werden. Abb. 3.10 zeigt beispielhaft einen Energiebaum nach drei Platzierungsschritten. Die Platzierung erfolgt analog zur Vorgehensweise des Best First Search (s. [Russell and Norvig, 1995]) Algorithmus. Dabei wird jeweils das Element mit minimalster Energie bevorzugt platziert.

Wird keine vollständige Flächennutzung in der Endkonfiguration erzielt oder kein geeignetes Element für den nächsten Platzierungsschritt gefunden, so wird im Energiebaum eine Konfiguration mit der nächst geringeren Ge-

samtenergie ermittelt und das Verfahren dort fortgesetzt. Durch diese Vorgehensweise kann eine näherungsweise optimale Flächennutzung mit gleichzeitig minimaler Gesamtenergie im Layout ermittelt werden.

3.3.2 Elementeplatzierung

Die Platzierung der Elemente im Container macht den geometrischen Aspekt der Flächensynthese aus. Ein Container stellt hier ein Polygon in der zweidimensionalen Ebene \mathbb{R}^2 dar. Das Containerpolygon ist im Regelfall konvex und nur in seltenen Fällen nicht-konvex. Für die Formelemente gilt das Gleiche, auch sie sind beliebige Polygone in der Ebene und i.d.R. konvex. Die Konvexität kann jedoch nicht als gegeben vorausgesetzt werden und darf daher keinen Einfluss auf die Tauglichkeit des Algorithmus haben.

Der Füllalgorithmus platziert Elemente sukzessive in einem Container, solange bis dieser (möglichst) vollständig befüllt ist. Die zwei sich aus der geometrischen Betrachtung ergebenden Randbedingungen stellen somit dar, dass 1) Elemente sich nicht gegenseitig oder mit der Hülle des Containers überschneiden sollen und 2) die Leerräume („Lücken“) zwischen Elementen und zur Hülle des Containers minimiert werden müssen, was analog zur Maximierung der Flächenausnutzung des Containers steht. Beides wäre eine Verletzung der Konsistenz des Layouts. Zusammen ergibt sich daraus ein Optimierungsproblem, welches als *Shape packing* Problem bekannt ist – in diesem Fall von irregulären Formen. Zur Lösung wird eine leicht veränderte Variante des Algorithmus verwendet, den Kim und Pellacini in ihrer Arbeit vorgeschlagen haben.

Für die Platzierung eines Elements im Container finden sich dort folgende sukzessive Teilschritte wieder:

1. Finde eine annähernd geeignete freie Stelle im Container;
2. Ermittle eine kleine Anzahl möglicher Elemente, die aufgrund ihrer Geometrie eine hohe Passgenauigkeit an gewählter Stelle besitzen;
3. Berechne für jedes dieser Elemente seine Energie unter Berücksichtigung bisherig platzierter Elemente. Die Energie ergibt sich aus dem Platzierungsgleichgewicht und weiterer Constraints;
4. Platziere jenes Element, das die niedrigste Energie aufweist und subtrahiere dessen Fläche vom aktuellen Container, woraus sich ein neuer Container ergibt;
5. Wiederhole Schritte 1–4 solange, bis der Container vollständig gefüllt (bzw. dessen Fläche Null) ist oder kein passendes Element mehr gefunden wird. In letzterem Fall gehe zurück zum Zustand mit der niedrigsten Gesamtenergie und wähle ein anderes Element. Danach fahre fort mit Schritt 1.

Die einzelnen Teilschritte werden im Folgenden genauer erläutert.

3.3.2.1 Approximierte Ortsbestimmung

In jeder Iteration gilt es dem Container ein weiteres Element hinzuzufügen. Elemente dabei an beliebiger Stelle im Container zu platzieren würde zu prinzipiell unendlich vielen möglichen Kombinationen des Layouts führen. Das wiederum würde den Suchaufwand zum Finden eines näherungsweise Optimums dramatisch erhöhen. Es ist daher wichtig, eine geeignete Annahme zu treffen, wo ein Element sinnvoll zu platzieren ist.

Kim und Pellacini schlagen zur vereinfachten Ortsbestimmung die Zuhilfenahme einer *Zentroidale Voronoi* Tesselierung (ZVT) des Containers vor. Zentroidale Voronoi-Diagramme stellen eine Sonderform einfacher Voronoi-Diagramme dar. Das Zentrum jeder Voronoi-Region ist dabei gleichzeitig deren geometrischer Schwerpunkt. Zur Erzeugung der ZVT wird die durchschnittliche Fläche ΔA_{El} aller Elemente im Quelldatensatz ermittelt und mit Kenntnis der verbleibenden Fläche A_C des noch zu befüllenden Containers $n = \frac{A_C}{\Delta A_{El}}$ Voronoi-Zentren in diesem erzeugt. Die Anwendung eines Algorithmus zur Bestimmung der ZVT (z.B. Lloyd's Algorithmus) führt dann zu n Regionen, die den Container gleichmäßig aufteilen (\rightarrow Platzierungsregionen). Jedes der (neuen) Zentren dieser Regionen ist damit ein plausibler Platzierungsort.

Die Wahl des Zentrums wird jedoch weiter eingeschränkt, wenn man annimmt, dass jene Regionen mit der geringsten Anzahl an Nachbarregionen den Container möglichst einfach segmentieren. Dies sind immer die Randregionen des Containers. Es wird folglich jene Region ausgewählt, die die geringste Anzahl an Nachbarregionen besitzt. Sind dies mehrere, so entscheidet der Zufall. Der Schwerpunkt (Voronoi-Zentrum) der gewählten Region wird als neuer Platzierungsort bestimmt. Abb. 3.11 veranschaulicht die ZVT und zeigt eine dadurch bestimmte Platzierungsregion auf.

3.3.2.2 Geometrische Passgenauigkeit

Am approximierten Platzierungsort muss im Weiteren die ungefähre Passgenauigkeit der Elemente evaluiert werden. Dabei handelt es sich um eine Näherungsbestimmung, bei dem nur die k besten Elemente ermittelt und weiter untersucht werden.

Die geometrischen Rahmenbedingungen am Platzierungsort sind gegeben durch die Containerhülle. Wie in Abschnitt 3.2.1.1 der Feature Extraktion erläutert, kann jedes Element über das *Geometric Hashing* Verfahren auf dessen geometrische Ähnlichkeit zu anderen Elementen hin untersucht werden. Eine geometrische Übereinstimmung mit der Fläche des verbleibenden Leerraumes im Container ist dabei äquivalent zur Passgenauigkeit eines Elements in diesem. Anstelle des gesamten Containers wird jedoch nur jener Teil

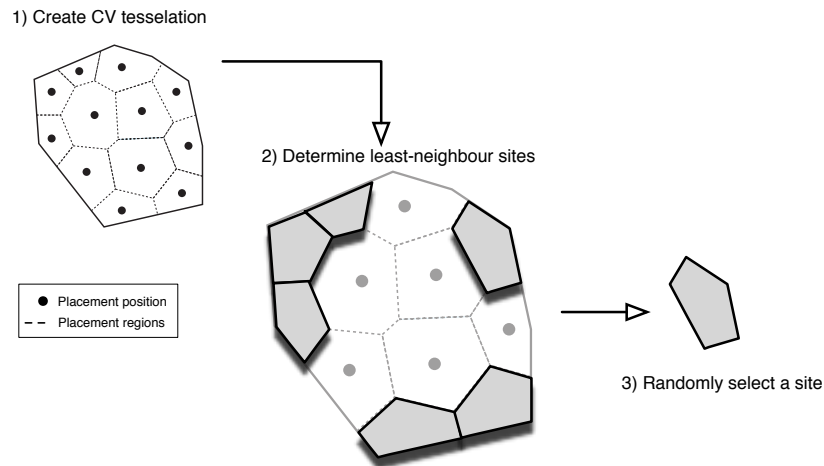


Abbildung 3.11: Ablauf der approximierten Ortsbestimmung: Zuerst wird eine Unterteilung des Containers in n Voronoi Regionen vorgenommen. Jede Region hat die ungefähre Größe eines durchschnittlichen Elements. Danach werden die geeignetsten Regionen bestimmt und per Zufall eine von ihnen ausgewählt.

hergenommen, welcher der Platzierungsregion entspricht, die bei der Platzierungsbestimmung ermittelt wurde. Dadurch wird der Abgleich auf einen örtlichen Bereich reduziert, der ungefähr den durchschnittlichen Flächeninhalt der Elemente besitzt und somit implizit Rücksicht auf deren Skalierung genommen.

Der Ablauf ist dann wie folgt: Zuerst wird die zum momentanen Platzierungsort zugehörige Voronoi-Region mittels Geometric Hashing im Einheitsraster der Hashtabelle registriert. Da sämtliche Elemente des Quelldatensatzes in der Feature-Extraktionsphase bereits ebenfalls dort erfasst wurden, kann unmittelbar ein Abgleich durchgeführt werden (für Details siehe Abschn. 3.2.1.1). Der Matching-Algorithmus liefert somit sämtliche Elemente sortiert nach dem Grad ihrer Übereinstimmung zurück. Wie erläutert, werden bereits bei der Erzeugung der Hashtabelle die für jedes Element möglichen (diskreten) Orientierungen berücksichtigt. Das Matchingverfahren findet somit auch jene Elemente, die beispielsweise erst nach einer Rotation um ihren Mittelpunkt eine hohe Passgenauigkeit aufweisen.

Aus den gefundenen Elementen werden im Weiteren nur jene k Elemente mit der höchsten Übereinstimmung berücksichtigt. Alle anderen Elemente werden für diesen Platzierungsort nicht weiter in Betracht gezogen. Faktor k ist dabei ein vom Nutzer bestimmter Wert. Abbildung 3.12 veranschaulicht den gesamten Prozess noch einmal.

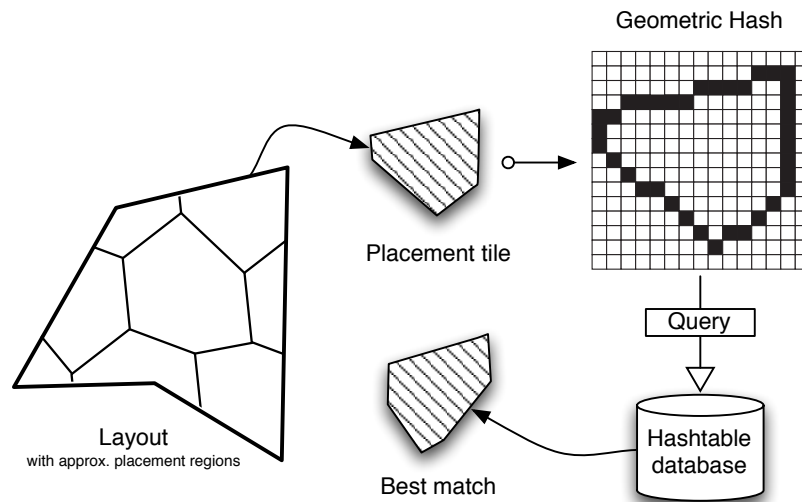


Abbildung 3.12: Bestimmung der Passgenauigkeit eines Elements im Container mittels des Formabgleichs (Geometric Hashing). Die Datenbank liefert die k Elemente mit den höchsten Übereinstimmungen zurück. Im Schaubild ist dies vereinfacht dargestellt.

3.3.2.3 Platzierungsenergie

Die Platzierungsenergie des Elements setzt sich zusammen aus dem geometrischen Platzierungsgleichgewicht und den sich durch die Umgebung des Elements ergebenden Constraints. Letztere Einflussfaktoren werden in einem gesonderten Abschnitt behandelt (s. 3.3.3). An dieser Stelle soll nur auf das Platzierungsgleichgewicht eingegangen werden.

Trotz Geometrieabgleich mittels des Geometric Hashing Verfahrens weisen platzierte Elemente im Regelfall keine exakte Passgenauigkeit an gegebener Stelle im Container auf. Es treten häufig Überschneidungen oder Abstände auf. Beide Faktoren stellen eine geeignete Metrik zur Beurteilung der Passgenauigkeit eines Formelements im Container dar. Für ein konsistentes Layout ist es i. Allg. erstrebenswert, beide Größen soweit zu minimieren, sodass ein Equilibrium entsteht. Das bedeutet, die Elemente möglichst dicht beieinander zu platzieren, ohne dass diese sich oder die Containerhülle überschneiden. Die exakte Berechnung der tatsächlichen Schnitt- und Freiflächen ist jedoch sehr rechenaufwändig. Daher soll eine einfache Methode zur Näherungsberechnung zur Hilfe genommen werden. Zur Bestimmung beider Größen wird die *Federenergie*

$$E_{Spring} = \frac{d^2}{2} \quad (3.15)$$

einer hypothetischen Feder zwischen jedem Eckpunkt V_i des Formelements und eines jeweils am nächsten liegenden Ankerpunktes P_i auf einer Kante

der lokalen Platzierungsregion berechnet (vgl. [Moore and Wilhelms, 1988]). Faktor d entspricht dabei der Distanz zwischen Eckpunkt und Ankerpunkt. Ist diese positiv, so wird die errechnete Federenergie zur Abstandsenergie E_G addiert. Existiert hingegen eine Überlappung zwischen Elementen, ist diese also negativ, wird sie zur Überlappungsenergie E_O gezählt. Der vollständige Term zur Evaluierung des Platzierungsgleichgewichts lautet somit:

$$E_{Placement} = w_G \cdot E_G + w_O \cdot E_O. \quad (3.16)$$

Es wird bewusst darauf verzichtet, beide Energiewerte von vorne herein in einen zu überführen, da sonst eine ggf. getrennte Gewichtung dieser wegfallen würde. Abb. 3.13 veranschaulicht die auftretende *Spannung* bei einem nicht optimal platzierten Element. Das Platzierungsgleichgewicht als ein Teil der Platzierungsenergie trägt somit maßgeblich zur Gesamtenergie eines Elements bei. Durch die Energie wiederum können die k zur Platzierung in Betracht gezogenen Elemente miteinander verglichen und jenes mit der geringsten Energie ausfindig gemacht werden.

Das ermittelte Platzierungsgleichgewicht kann des Weiteren dazu genutzt werden, die Platzierung zu verfeinern. Eine Verfeinerung kann beispielsweise durch iteratives Umarrangieren im örtlichen Bereich des Elements bewerkstelligt werden. Dabei wird auf der Suche nach einem lokalen Minimum für jede Platzierung die Platzierungsenergie neu evaluiert. Auf weitere Details der Verfeinerung soll an dieser Stelle jedoch nicht näher eingegangen werden.

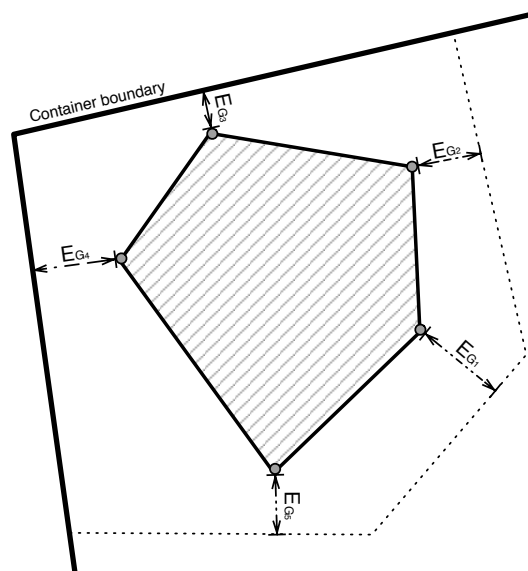


Abbildung 3.13: Formelement im Spannungsfeld. Die hier auftretenden Federenergien E_{G_i} ergeben in der Summe das Platzierungsgleichgewicht des Elements.

3.3.2.4 Aktualisierung des Containers

Nach der endgültigen Bestimmung eines Elements zur Platzierung wird dessen Form vom Container geometrisch subtrahiert. Der modifizierte Container, der sich daraus ergibt, wird für den jeweils nächsten Platzierungsschritt weiterverwendet. Die durch die Intersektion des Elements mit dem Container resultierenden neuen Kanten bekommen jeweils die Eigenschaften der jeweiligen Elementkanten des subtrahierten Elements übertragen. Hierdurch bleiben sämtliche Constraints zwischen Elementen für den nächsten Platzierungsschritt erhalten, ohne dass die vollständigen Datenstrukturen sämtlicher platzierter Elemente verwaltet werden müssen.

Zum besseren Verständnis wird in folgenden Abschnitten dennoch von einzelnen Elementen gesprochen, obwohl die interne Struktur sich tatsächlich auf einen einzelnen, zusammengeführten Container beschränkt.

3.3.3 Lokale Randbedingungen

Kontinuität spielt eine entscheidende Rolle für die Glaubwürdigkeit von synthetisch erzeugten Arrangements. Bei den hochstrukturierten Inhalten urbaner Layouts bedeutet das im Wesentlichen eine strukturelle, implizit semantische, Kontinuität. So müssen beispielsweise Grundstücke mit Zufahrten einen korrekt ausgerichteten, direkten Zugang zu einer Straße aufweisen. Um strukturelle Kontinuität zwischen Formelementen in synthetisierten Layouts herzustellen, bedarf es prinzipiell bis auf Objektebene segmentierter Quelldaten. Die Absenz dieser Datengenauigkeit zwingt einen auf weniger verlässliche und weitaus weniger präzise Faktoren zur Bestimmung von Kontinuität zurückzugreifen, z.B. der Farbkontinuität. Da es sich bei Layouts naturgemäß um räumliche Beziehungen handelt, kann im Weiteren von der *strukturellen Nähe* als Metrik zur Beschreibung von struktureller Zusammengehörigkeit gesprochen werden.

Die Eigenschaften (*Features*), auf welche sich diese strukturelle Nähe auswirkt, sind zwar für deren Gewichtung von großem Interesse, ändern aber an der grundlegenden Problemstellung bei der Beurteilung struktureller Nähe nichts.

3.3.3.1 Bestimmung von Nachbarschaft

Diverse Verfahren Beispielbasierter Synthese haben gezeigt, dass örtliche Nachbarschaftsbeziehungen ein ideales Maß für strukturelle Nähe bilden¹⁰ und dass das Abgleichen von räumlichen Nachbarschaftsbeziehungen zu guten Ergebnissen in der Synthese führt. In der Pixel-basierten Bildsynthese werden neu zu befüllende Pixel im Zielbild hinsichtlich ihrer Nachbarpixel untersucht. Dann wird im Quellbild ein äquivalentes Pixel gesucht (vgl. *nearest-neighbour-search*), dessen benachbarte Pixel den eigenen Nachbarn am nächsten kommen (meist ist hier Farbe die ausschlaggebende Eigenschaft) – übertragen formuliert besitzen angrenzende Pixel ein einheitliches Stimmgewicht und stimmen entsprechend ihrer Eigenschaft (z.B. ihrer Farbe) ab. Das Pixel im Zielbild wird schließlich durch jenes Pixel mit der höchsten Übereinstimmung (ähnlichstes Stimmergebnis) ersetzt.

In diesem Verfahren wird eine ähnliche Vorgehensweise für die Flächensynthese gewählt, welche *a)* mit dem besonderen Platzierungsverfahren der *Image Mosaics* und *b)* mit dem Energie-basierten Framework in Einklang gebracht werden kann. Ziel- und Quelldatensatz verhalten sich dabei analog zu den Rasterbildern in der Pixel-basierten Synthese. Die Ausnahme ist, dass hier der Quelldatensatz bereits vorab segmentiert und Elemente auf ihre Features untersucht werden.

¹⁰vgl. [Efros and Leung, 1999, Hertzmann et al., 2001, Ijiri et al., 2008]

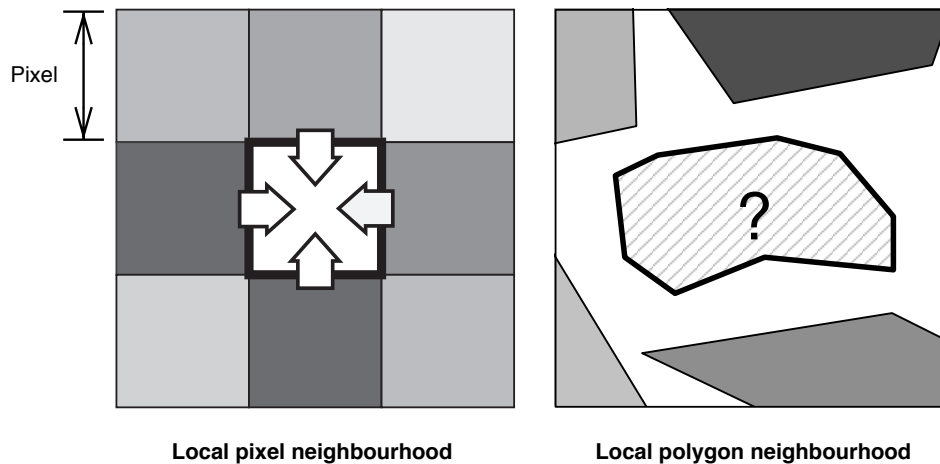


Abbildung 3.14: Lokale Nachbarschaftsgewichtung im Vergleich: **1)** Links eine typischen Nachbarschaft in Rasterbildern mit quadratischen Pixeln. Die das Zentrum umschließenden Pixel üben den Einfluss aus. Die Bestimmung der Nachbarn und deren Gewichtung ist dabei relativ eindeutig. **2)** Rechts eine ähnliche Situation mit frei platzierten Polygonen. Die Bestimmung der örtlichen Nachbarn und deren Stimmgewicht ist hier keinesfalls eindeutig und bedarf einer eigenen Metrik.

Lokale Einflusszonen Aufgrund der Tatsache, dass das Verfahren das freie Platzieren und Arrangieren beliebig geformter Elemente voraussetzt, entfällt eine triviale Nachbarschaftsbeurteilung, wie es bei quadratischen Pixeln mit i.d.R. gleich gewichteten Kanten der Fall ist (siehe Abb. 3.14). Es ergibt sich demnach die Problematik, ob und wie stark nahegelegene Elemente ein Stimmgewicht in der lokalen Nachbarschaft eines Elements besitzen. Eine Metrik beruhend auf der Distanz der Zentren oder eine auf der *Federenergie*-Analogie beruhender Ansatz, wie beim Platzierungsgleichgewicht, scheint dazu ungeeignet – beide vernachlässigen die topologische Beziehung zwischen Kanten der Elemente.

Elemente sind geometrisch betrachtet Polygone. Deshalb werden deren Inhalt, anders als bei Pixeln, hier als nicht homogen erachtet. Inhomogenität zeigt sich bereits bei o.g. Beispiel der Straßenzufahrt, da diese i.d.R. nur eine Seite einer Grundstücksparzelle betrifft. Das Stimmgewicht darf deshalb nicht von einem zentralen Punkt des Elements ausgehen. Eine ideale Lösung dieses Problems scheint generell ausgeschlossen, sodass im Weiteren auf eine annähernd geeignete Metrik gesetzt wird. Dazu werden die Kanten der Elemente als lokale Schnittstellen verwendet, über diese sie örtlich Einfluss ausüben können.

Elemente in der Platzierungsphase stellen einfache, geschlossene Polygone ohne gemeinsame geometrische Kanten dar, was es verbietet, die Kanten

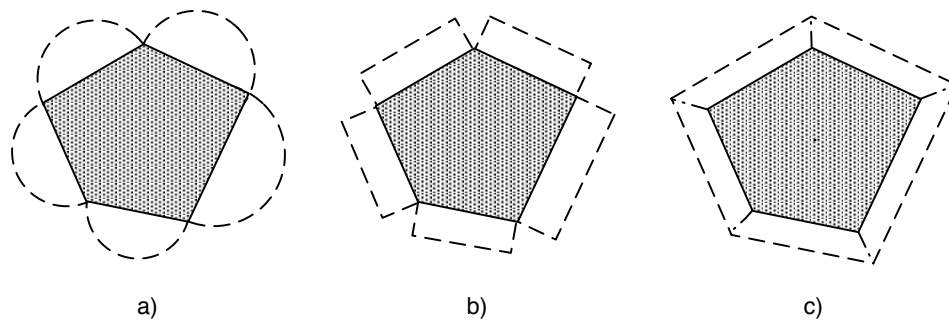


Abbildung 3.15: Drei denkbare Einflussregionen.

selbst als Schnittstellen zu verwenden. Stattdessen sind sogenannte Kanten-Einflusszonen ein praktikabler Ansatz. Abb. 3.15 zeigt drei denkbare Einflussregionen von Kanten. Das Element in Abb. 3.15 a verwendet einen Halbkreis, dessen Mittelpunkt der Kantenmittelpunkt ist, in 3.15 b werden nach außen extrudierte Rechtecke verwendet und in 3.15 c schließlich die *Thiessen-Polygone* (siehe auch Voronoi-Diagramm) der Kanten.

Die auf Rechtecken basierenden Einflussregionen vernachlässigen jedoch die Diagonalbeziehung zwischen Elementen an deren Eckpunkten. Bei der auf Halbkreisen basierenden Variante nimmt das Einflussgebiet zu den Endpunkten einer Kante hin ab, was in vielen Anwendungsfällen unbegründet scheint. Die Variante, die auf den *Thiessen-Polygonen* basiert, besitzt beide Nachteile nicht. Dafür ist sie aber am aufwändigsten zu ermitteln. Dennoch soll die Konstruktion dieser Variante kurz erläutert werden:

Um das Einflussgebiet jeder Kante der Kontur eines Elements zu bestimmen, wird das *Voronoi-Diagramm* aller seiner Kanten gebildet, approximiert durch das äußere *Straight Skeleton* [Oswin et al., 1995]. Man erhält je Kante k_i des ursprünglichen Polygons genau eine Region außerhalb dessen Kontur, deren innere Punkte näher an k_i liegen als an jeder anderen Kante des Polygons. Solch eine Region soll vereinfachend als Einflussregion einer Kante angenommen werden. Dies entspricht dem Gegenstück zu den in Abschnitt 3.2.1.3 vorgestellten inneren Kantenregionen. Vergleichbare auf Einflusszonen basierende Ansätze finden sich u.a. in [Tutenel et al., 2009].

Einflussfaktor Die Einflussregionen stehen in einer räumlichen Relation zueinander, die entsprechend zur Anordnung der Elemente ist. Distanz und Winkel zwischen Elementkanten spielen eine wichtige Rolle, was sich im Einflussfaktor widerspiegeln muss. Dieser wird berechnet aus der Schnittfläche – sofern vorhanden – einer Einflusszone mit der eines anderen Elements. Die Schnittfläche wird zusätzlich ins Verhältnis zum Flächeninhalt der ersten Zone gesetzt. Konkret bedeutet dies, dass der Einfluss proportional zur Abdeckung einer Einflusszone durch eine andere ist.

Wird ein Strukturelement zur Platzierung im Zielcontainer in Betracht gezogen, können dessen Features mit den im Zielcontainer bereits vorhandenen Elementen abgeglichen und mit dem jeweiligen Einflussfaktor gewichtet werden. Zur Reduktion der Suche werden nur benachbarte Elemente berücksichtigt.

Der Einflussfaktor ermöglicht die Gewichtung des Stimmgewichts lose benachbarter Formelemente (bzw. derer Kanten) beim Feature-Matching und nimmt somit indirekt Einfluss auf die Beurteilung der Tauglichkeit eines Elements zur Platzierung an gegebener Stelle. Die Gewichtung erfolgt nach o.g. Schema aufgrund der räumlichen Beziehung. Dieser Umstand setzt voraus, dass gewichtete Features im weiteren Sinne einen räumlichen Ursprung besitzen müssen, ansonsten ist der Einflussfaktor zur Gewichtung ungeeignet. Eine genauere Erläuterung des Abgleichens (Matching) von Features wird im nächsten Abschnitt gegeben.

3.3.3.2 Feature Matching

Das Maß der Sättigung lokaler Randbedingungen im Layout kann über den Abgleich relevanter Kohärenzeigenschaften zwischen den Strukturelementen bestimmt werden. Diese wurden in Abschnitt 3.2 als strukturelle und semantische *Features* identifiziert und ausführlich beschrieben. Die Menge der *Features* eines Elements werden als dessen *Featurevektor* bezeichnet. Genauer existieren n solcher Featurevektoren je Element, einer je Elementkante.

Der Abgleich erfolgt nach der in vorherigen Abschnitten beschriebenen Herangehensweise im Gültigkeitsbereich der Einflusszonen eines Elements¹¹. Konkret findet der Abgleich folglich in den p sich einander beeinflussenden Zonen eines Elements S_i und seiner k nächsten Nachbarn statt.

Es wird hier die Erfüllung der Randbedingungen zwischen den Elementen über die Bestimmung derer Feature-Vektor Distanz je Einflusszone ermittelt und diese dann als Energieterm formuliert. Das bedeutet beispielsweise, je geringer die Distanz zwischen zwei Feature-Vektoren, desto geringer die dort auftretende Energie, respektive desto höher das Maß der Sättigung lokaler Randbedingungen für diese Zone. Die Gesamtenergie eines Elements ergibt sich – unter Berücksichtigung der Gewichtung durch die *Einflussfaktoren* – aus der Summe aller Teilenergien seiner n Zonen.

Der konkrete Matchingvorgang ist abhängig vom jeweiligen Feature. Dabei muss ggf. die wechselseitige Einflussnahme der Strukturelemente berücksichtigt werden. Zur eigentlichen Distanzberechnung von identischen Features werden jedoch stets die im Kapitel der Feature Extraktion erläuterten Metriken zur Hilfe genommen.

Im Folgenden werden zwei Kohärenzaspekte betrachtet, für die eine hohe

¹¹Die Anzahl und Bedeutung der Einflusszonen ist großteils analog zu den Kanten eines Elements (s. Abschn. 3.3.3.1), weshalb beide Bezeichnungen im Weiteren gleichbedeutend verwendet werden.

Relevanz zur Schaffung von Kontinuität im synthetisierten Layout angenommen wird. Zum einen werden die topologisch-semantischen Beziehungen zwischen Strukturelementen, zum anderen die an Elementgrenzen auftretende Farbkontinuität untersucht. Insbesondere wird dabei ihre Formulierung als Energierterme beschrieben, die sie für den Synthesealgorithmus verwertbar machen.

Grenzbedingungen Kontinuität an Elementkanten kann durch den Abgleich der in den Kantenregionen (s. Abschn. 3.2.1.3) zu beobachtenden Eigenschaften (Features) bestimmt werden. Die Bestimmung erfolgt trivial über die Berechnung der Distanz unter Zuhilfenahme einer geeigneten Distanzmetrik. Die Distanz kann dann als Energie zwischen den Kanten in Bezug auf das jeweilige Feature betrachtet werden.

Die Bestimmung der Farbkontinuität zwischen zwei Kanten, respektive Kantenregionen (q und t), lässt sich einfach über die jeweilige Distanz der Farbhistogramme beider Strukturelemente bestimmen. Die auftretende Energie ergibt sich dann als

$$E(q, t) = \sum_{m=0}^{M-1} [h_q(m) - h_t(m)]^2. \quad (3.17)$$

Topologisch-Semantische Relationen Die Platzierung eines Strukturelements an gegebener Stelle in einem Container führt zu einer unmittelbaren Veränderung der örtlichen topologischen Beziehungen im Layout. Um die Konsistenz des Layouts sicher zu stellen, muss die veränderte Situation der topologischen Relationen zwischen Elementen unter Beachtung des semantischen Kontexts beurteilt werden. Ziel ist es, jenes Element zu ermitteln, welches den höchsten Grad an Kohärenz aufweist, d.h. die größte Anzahl topologisch-semantischer Randbedingungen an dieser Position im Layout erfüllt.

Es lässt sich beobachten, dass das Hinzufügen eines neuen Strukturelements zu einer wechselseitigen Abhängigkeit zwischen diesem und seinen sich bereits im Layout befindlichen, lokalen Nachbarn führt. Sowohl das neu hinzugefügte, als auch die bereits platzierten Elemente sind jeweils bestrebt, ihre topologisch-semantischen Randbedingungen zu befriedigen. Die Wechselseitigkeit der Randbedingungen sichert dabei die strukturelle Kohärenz zwischen Elementen in beide Richtungen.

Um die entstandene Situation im Layout zu beurteilen, wird die Energie der auftretenden topologisch-semantischen Relationen nach Platzierung des neuen Elements bestimmt. Diese wird zu dessen Gesamtenergie addiert.

Eine Relation existiert jeweils zwischen zwei Kanten (K_{A_i} und K_{B_i}) verschiedener Strukturelemente (siehe Abschn. 3.3.3.1). Um festzustellen, ob die, durch das Quelllayout vorgegebenen topologische-semantischen Relationen, eines Elements im Ziellayout erfüllt werden und in welchem Maße, muss

Tabelle 3.4: Wechselseitige Beziehung zwischen zwei Kanten. Für jede Kante muss jeweils die auftretende Energie zwischen ihrem Ist- und Sollzustand topologisch-semantischer Relationen ermittelt werden. Die Summe dieser Energie über beide Kanten ergibt dann die auftretende Gesamtenergie zwischen ihnen.

	<i>Class</i>	<i>Topological relation</i>	<i>Class</i>	<i>Energy</i>
Current state	residential estate	meets	commercial site	
Target state	residential estate	disjoint	country road	
Distance		3	4	7
Current state	commercial site	meets	residential estate	
Target state	commercial site	meets	industrial site	
Distance		0	1	1
Total energy				8

sein Ist-Zustand im Ziellayout dem Soll-Zustand der jeweiligen Kantenrelation gegenübergestellt werden. Diese Betrachtung muss dabei einmal aus Sicht einer Kante K_{A_i} zu K_{B_i} und einmal aus Sicht von K_{B_i} zu K_{A_i} erfolgen. Es ist wichtig zu beachten, dass Relationen nicht in beide Richtungen identisch sind. Dies kann aufgrund der möglicherweise abweichenden Soll-Zustände begründet werden.

Insgesamt lassen sich folglich vier Relationen zwischen zwei Kanten ausmachen, von denen jeweils zwei je Kante (Ist und Soll) miteinander verglichen werden müssen. Die Summe der Energien (ermittelt als Feature-Distanzen) zwischen beiden Relationspaaren entspricht dann der Gesamtenergie der Relation zwischen den zwei Kanten. Tabelle 3.4 soll diese komplexe Beziehung anhand von Beispieldaten anschaulich darstellen.

Die Ermittlung der Energie zweier Relationen erfolgt nach dem bereits in Abschnitt 3.2.1.2 beschriebenen Verfahren zur Berechnung der Feature-Distanz für topologisch-semantische Relationen.

Bestimmung des Ist-Zustand Während der Soll-Zustand einer Relation als Feature jeder Kante gespeichert ist, muss eine Beschreibung des Ist-Zustands erst ermittelt werden. Die Bestimmung fällt jedoch trivial aus. Die Klassen der an der Relation beteiligten Strukturelemente können aus deren Feature-Vektor direkt ermittelt werden. Eine tatsächliche Bestimmung der topologischen Relation ist ebenfalls nicht notwendig. In diesem Anwendungsfall wird davon ausgegangen, dass zwei unmittelbar benachbarte Strukturelemente einander im endgültigen Layout berühren sollen. Eine topologisch-semantische Relation des Ist-Zustandes zweier Elemente S_A und S_B lässt sich demnach wie folgt formulieren:

$$R(S_A, S_B) = (\text{class}(S_A), \text{meets}, \text{class}(S_B)). \quad (3.18)$$

Kapitel 4

Implementierung

Die Kernlogik des vorgestellten Syntheseverfahrens wurde in MATLAB 2008 realisiert. Zusätzlich wurde ein eigenständiges Framework in C++ entwickelt, welches das Betrachten der Ergebnisse in einer hardwarebeschleunigten OpenGL-Renderumgebung erlaubt. Der Fokus wird hier auf die Implementierung des Kernprogrammteils gelegt.

In Abb. 4.1 wird eine Übersicht der wichtigsten Datenquellen und Abläufe des Programms veranschaulicht. Das Programm ist analog zu der in Kapitel 3 erörterten Vorgehensweise in zwei primäre Verarbeitungsschritte gegliedert. Dies ist zum Ersten eine *Preprocessing Phase*, zum Zweiten die eigentliche *Synthesis Phase* zur Generierung neuer Layouts.

4.1 Programmablauf

4.1.1 Preprocessing Phase

In der Preprocessing Phase wird eine Datenbank mit den für die Synthese notwendigen Elementen und deren Features erstellt. Als Input dienen geographisches Kartenmaterial und georeferenzierte Satellitenbilder des Quelllayouts (1.1), die durch den Nutzer zur Verfügung gestellt werden müssen. Unterstützt werden derzeit *Shapefiles* und *OSM* als Kartenformate. Die Satellitenbilddaten werden aus einer offline Bilddatenbank gewonnen.

In einem ersten internen Vorverarbeitungsschritt werden die Elemente aus den geographischen Layoutdaten zusammen mit ihren Bilddaten aufbereitet (1.2) und zu Fragmentdatenstrukturen zusammengeführt (1.3). Der Hauptteil der Preprocessing Phase machen dann die Feature Extraktion (1.4) und deren Speicherung (1.5) aus. Während der Feature Extraktion werden die in Kapitel 3 beschriebenen Features aus den Geofragmenten algorithmisch ermittelt und anschließend in einer Datenbank gespeichert. Hierdurch kann in der späteren Synthesis Phase effizient darauf zurückgegriffen werden. In Fig. 4.1 ist der grobe Programmablauf der ersten Phase als Pseudocode abgebildet.

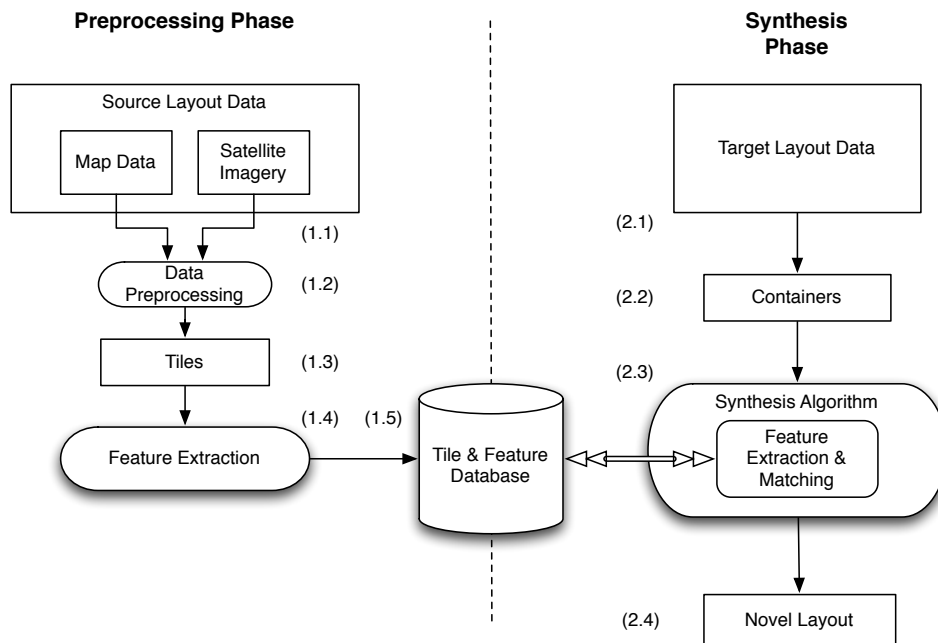


Abbildung 4.1: Schematische Übersicht des Programmablaufes.

4.1.2 Synthesis Phase

Die Synthesis Phase umfasst die Erzeugung des neuen Layouts unter Verwendung der in Abschnitt 3.3 erläuterten Techniken der strukturellen Synthese. Aus einem geographischen Datensatz für das Ziellayout (2.1) werden alle für die Synthese relevanten Container des Layouts ermittelt (2.2). Auf jeden dieser Container wird jeweils der vollständige Synthese Algorithmus angewendet (2.3).

Dieser beinhaltet u.a. die Segmentierung des Containers, das Feature-Matching und die Platzierung der ermittelten Elemente im Container. Beim Feature-Matching werden dazu zum Teil gleiche Programmmethoden wie in der Preprocessing Phase wiederverwendet, z.B. das Geometric Hashing. Insbesondere greift der Algorithmus auf die in der Preprocessing Phase erzeugte Datenbank zu. Nach Vollendung der Synthese wird das neu generierte Layout über einen Renderer visualisiert (2.4). Der Programmablauf ist in Fig. 4.2 ebenfalls als Pseudocode dargestellt.

4.2 Geometric Hashing

Das Geometric Hashing ist eine Kernfunktion des Algorithmus, sodass dessen Umsetzung hier kurz vorgestellt werden soll. Die Implementierung ist

Algorithmus 4.1: Feature Extraktion

```

1: FEATUREEXTRACTIONPHASE( $M, I$ )
   Creates a tile feature database from map  $M$  and image source  $I$ 
2:   for each closed polygon  $P_i$  in  $M$  do
3:      $T_i \leftarrow$  CreateTileFragment( $P_i, I$ )
4:      $FT_i \leftarrow$  ExtractFeatures( $T_i$ )
5:     Store ( $T_i, FT_i$ ) together in database  $D$ 
6:   end for
7: end

1: CREATETILEFRAGMENT( $P, I$ )
2:   if coordinates of  $P$  are geographic then
   transform vertices  $V$  of  $P$  using map projection  $Q$ 
3:   end if
4:   for each vertex  $v_i$  in  $V$  do
   Determine texture  $t$  from  $I$  and calculate texture coordinate
    $uv$ . Create edge  $e(v_i, v_{i+1})$ 
5:   end for
6:   return tile fragment with ( $V, V_e, V_t, V_{uv}$ )
7: end

1: EXTRACTFEATURES( $T$ )
2:   Determine geometric hash  $H$ , color border regions  $C$  and topological
   relations  $R$  of  $T$ 
3:   return feature vector ( $H, C, R$ )
4: end

```

realisiert unter Verwendung folgender Mittel:

- 3-dimensionale Array-Datenstruktur,
- Rasteralgorithmus,
- XOR-Maskenoperator.

4.2.1 Datenstruktur

Als Datenstruktur der Hashtabelle dient ein $32 \times 32 \times P$ Array. Die ersten beiden Dimensionen im Array adressieren das diskrete Rastergitter. Die dritte Dimension wird als eine Art dynamische Liste verwaltet und dient der Aufzeichnung und Speicherung von Einträgen in der Hashtabelle an Stelle des jeweiligen Rasterfeldes.

P ist bestimmt durch die maximale Anzahl an Einträgen über alle Rasterfelder, i.d.R. ist diese identisch zur Anzahl an Elementen in der Quelldatenbank. Für die ersten beiden Dimensionen kann eine beliebige, abwei-

Algorithmus 4.2: Synthesis Phase

```

1: SYNTHESISPHASE( $M, D$ )
   Creates a novel layout for map  $M$  using database  $D$  as source
2:   for each container  $C_i$  in  $M$  do
3:     while  $C_i$  not filled do
4:        $R \leftarrow$  SegmentContainer( $C_i$ )
5:        $R_p \leftarrow$  find placement region with least neighbours in  $R$ 
6:        $T \leftarrow$  FindBestGeometricMatches( $R_p$ )
7:       for each tile  $T_i$  in  $T$  do
8:          $E \leftarrow$  CalculateEnergy( $T_i, R_p$ )
9:         if  $E < E_{least}$  then
10:           $B \leftarrow$  assign  $T_i$  as tile with least energy
11:        end if
12:      end for
13:      Place tile  $B$  in  $C_i$  and update container
14:    end while
15:  end for
16: end

1: SEGMENTCONTAINER( $C$ )
2:    $n \leftarrow$  determine approx. sites left from container area and average tile
   size
3:   Construct Centroidal Voronoi diagram with  $n$  sites. Clip Voronoi re-
   gions  $R$  with boundary of  $C$ .
4:   return Voronoi regions  $R$ 
5: end

1: FINDBESTGEOMETRICMATCHES( $R$ )
2:    $H \leftarrow$  create geometric hash of  $R$ 
3:    $T \leftarrow$  find  $k$  best matches for  $H$  in database  $D$ 
4:   return tiles  $T$ 
5: end

1: CALCULATEENERGY( $T, R$ )
2:   Calculate  $T$ 's energy for placement equilibrium  $E_{EQ}$ , color border
   matches  $E_C$  and topological relations  $E_R$  at position of  $R$ 
3:   return  $E \leftarrow E_{EQ} + E_C + E_R$ 
4: end

```

chende Auflösung $m \times n$ gewählt werden, was Auswirkung auf die Präzision des Matching hat und gleichzeitig zu einem proportional skalierten Speicher- und Suchaufwand führt.

4.2.2 Rasteralgorithmus

Der Rasteralgorithmus dient dazu, die Hüllen der im kontinuierlichen Raum \mathbb{R}^2 beschriebenen Polygone in ein diskret quantifiziertes Rastergitter \mathbb{N}^2 zu überführen, der so genannten Hashmaske. Dadurch kann der Vergleichsaufwand zwischen zwei beliebigen Polygonen in einer Datenbank von $O(N_{tiles})$ auf $O(h_{grid})$ reduziert werden¹ (siehe [Kim and Pellacini, 2002]). Zur Bildung der Hashmaske wird ein Scanline Algorithmus angewendet und eine binäre Bitmaske mit identischer (zweidimensionaler) Größe zur Hashtabelle erzeugt. Einträge mit 1 entsprechen dabei einer Überschneidung des Polygons mit dem Raster an Stelle x/y , 0 entspricht keiner Überschneidung. Schaubild 4.2 zeigt die Rasterung eines Polygons.

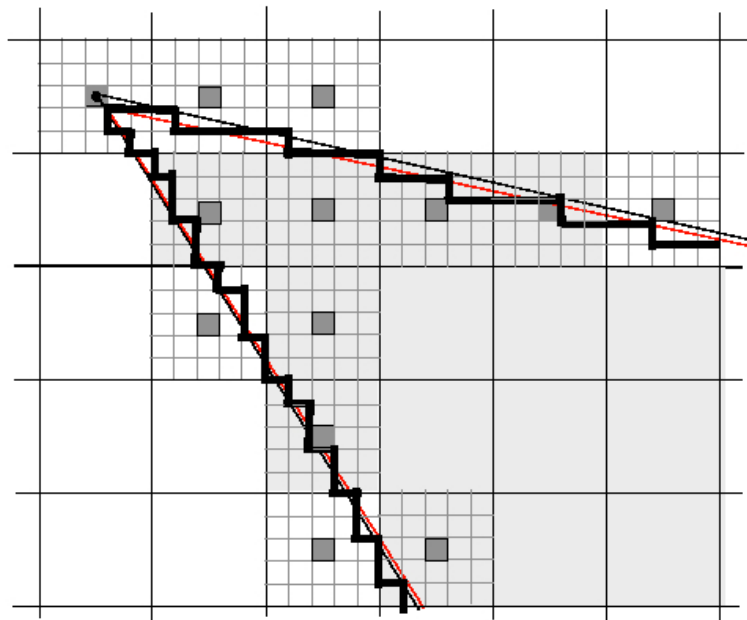


Abbildung 4.2: Rasterung eines Polygons. Hellgrau markiert jeweils die in der Rasterung durch die Polygonkanten berührten Felder. Zur Bestimmung der Überschneidung werden sog. Subpixel verwendet (Quelle: MATLAB 2008).

4.2.3 XOR-Maskenoperator

Zur Bewertung der Übereinstimmung einer Hashmaske (Suchmaske) mit einer zweiten (Abgleichmaske) muss zunächst die binäre XOR-Differenz beider Masken über das gesamte Rastergitter gebildet werden. Anschließend

¹Faktor N_{tiles} entspricht dabei der Anzahl an Elementen in der Datenbank. h_{grid} entspricht der Auflösung des Rastergitters (s.o.)

werden die übrig gebliebenen Einträge gezählt – sie entsprechen der Fläche nicht überlappender Bereiche zweier Polygone. Je geringer hierbei die Anzahl der Einträge, desto höher ist die Übereinstimmung einer abgeglichenen Maske mit der jeweiligen Suchmaske. Der Abgleich zwischen Suchmaske und Abgleichmaske wird für alle in der Hashtabelle gespeicherten Elemente durchgeführt.

Kapitel 5

Auswertung

In Kapitel 3 wurde ausführlich ein neuartiger Ansatz zur computergestützten synthetischen Erzeugung und Visualisierung urbaner Layouts erläutert. Insbesondere wurde dabei eine Technik aus dem Bereich der Bildverarbeitung, namentlich das *Image Mosaics* Verfahren, adaptiert. Für die Auswertung wurde daher sowohl eine quantitative Analyse des gewählten Ansatzes durchgeführt, als auch eine qualitative Betrachtung der Resultate berücksichtigt.

In der quantitativen Analyse wurde die objektive Tauglichkeit des gewählten Ansatzes bei Anwendung auf Geodatenätze untersucht. Dazu wurde ein messbares Qualitätskriterium definiert und das Syntheseverfahren in automatisierten Versuchsreihen wiederholt unter Verwendung verschiedener Datensätze ausgeführt. Die ausführliche Beschreibung und Auswertung der quantitativen Analyse findet sich in Abschnitt 5.1.

Die qualitative Analyse entspricht im Gegensatz dazu der subjektiven Beurteilung der Ergebnisse. Daher werden in Abschnitt 5.2 die Resultate verschiedener Anwendungsbeispiele des Verfahrens auf reale Datensätze gezeigt und kurz kommentiert. Ansonsten wird dort dem Betrachter die Beurteilung der Qualität der Ergebnisse selbst überlassen.

5.1 Quantitative Analyse

Die quantitative Analyse dient zur Evaluierung der Versuchsergebnisse anhand objektiver Kriterien. Die verwendeten Beurteilungskriterien sind im nächsten Abschnitt 5.1.1 genauer beschrieben.

Auf Grundlage ‚ausgewogener‘ Datensätze wurde in einer umfassenden Versuchsreihe untersucht, wie der Algorithmus in mehreren Standardszenarien abschneidet. Dabei wurde vorwiegend die Auswirkung unterschiedlicher Datensatzmerkmale auf die Qualität der Resultate analysiert. Hierbei stand insbesondere die Erfassung möglicher vorhandener Korrelationen zur Datensatzgröße im Vordergrund. Desweiteren wurde versucht, Grenzwerte – etwa

die kritische Datenmasse und maximal erzielbare Ausbeute – zu bestimmen. In dieser Versuchsreihe wurden dabei ausschließlich Techniken aus dem original *Image Mosaics* Verfahren berücksichtigt¹.

5.1.1 Qualitätsmetrik

Zur Beurteilung der Resultate diene ein vorab festgelegtes, quantitativ erfassbares Qualitätskriterium. Das Qualitätskriterium wurde in Anlehnung an die Kriterien zur Energiebestimmung in Layouts gewählt. Es wird aus der relativen Überlappung der Elemente und der am Container gemessenen relativ verbleibenden Freifläche errechnet. Der als Qualitätsfaktor Q bezeichnete Faktor wurde somit bestimmt als

$$\begin{aligned} Q &= 1 - (Q_{overlap} + Q_{space}) \\ &= 1 - \left(\frac{A_O}{A_T} + \frac{A_S}{A_C} \right). \end{aligned}$$

Darin entspricht A_O den aufsummierten Überlappungsflächen der Elemente, A_S den summierten Freiflächen, A_T der Fläche aller Elemente und A_C der Fläche des Containers. Die Berechnungen aller Größen wurden dabei exakt durchgeführt.

5.1.2 Versuchsreihen zur Flächensynthese

5.1.2.1 Datenvorbereitung

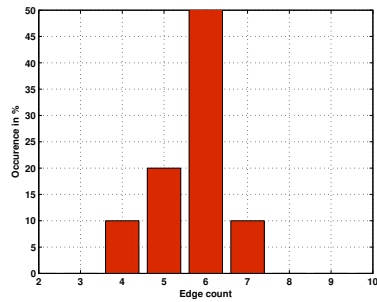
Als Grundlage der Versuchsdurchführung wurden 5 Elementdatensätze verschiedener Größen generiert (siehe Tabelle 5.1). Zur Befüllung der Datensätze mit 'durchschnittlichen' Strukturelementen wurde ein einfaches Verfahren angewendet. Es wurden 1.) in einem Einheitsraster n Punkte per Zufall platziert und deren Voronoi-Diagramm gebildet. Hierbei galt $20 \geq n \geq 2, n \in \mathbb{N}$, 2.) die resultierenden Voronoi-Regionen in den jeweiligen Datensatz als Elemente übernommen und 3.) wurden Schritte 1 und 2 so lange wiederholt, bis die gewünschte Größe des Datensatzes erreicht wurde. Zusätzlich wurde nach gleichem Schema ein Zieldatensatz mit 15 Containern erzeugt.

In Tab. 5.1 findet sich eine kurze Übersicht der produzierten Datensätze. In Abb. 5.1 wird überdies die Verteilung der Kantenkomplexitäten (\rightarrow Anzahl Kanten) der Elemente in den einzelnen Datensätzen grafisch veranschaulicht.

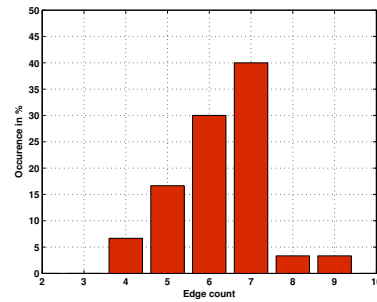
¹Eine ausführliche Analyse des vollständigen Verfahrens kann an dieser Stelle nicht gegeben werden. Um die Resultate zu gewinnen, wurde eine vereinfachte Variante des in dieser Arbeit beschriebenen, umfangreichen Verfahrens realisiert und angewendet. Gezeigt werden soll deshalb lediglich die tendenzielle Anwendbarkeit des gewählten Ansatzes auf geographische Daten.

Tabelle 5.1: Merkmale der generierten Versuchsdatensätze.

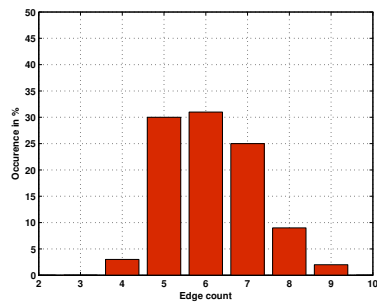
	<i>Dataset 1</i>	<i>Dataset 2</i>	<i>Dataset 3</i>	<i>Dataset 4</i>	<i>Dataset 5</i>
Number of tiles	10	30	100	500	1500



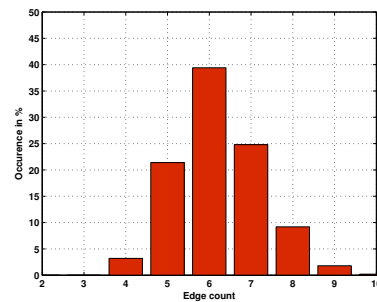
(b) Dataset 1



(c) Dataset 2



(e) Dataset 3



(f) Dataset 4

Abbildung 5.1: Verteilung der Kantenkomplexität in den generierten Datensätzen. Bei größer werdenden Datensätzen (1–4) ist eine zunehmende Gauß'sche Verteilung zu beobachten, die als wichtiges Merkmal eines ‚ausgeglichenen‘ Datensatzes betrachtet wird.

5.1.2.2 Durchführung

In der Versuchsdurchführung wurde das Syntheseverfahren unter Verwendung jeder der 5 Elementdatensätze einmal auf das Ziellayout angewendet. Für alle der im Zieldatensatz synthetisierten Containerlayouts wurde dabei jeweils der Qualitätsfaktor Q bestimmt und aufgezeichnet.

Tabelle 5.2: Ergebnisse der Versuchsreihen.

	<i>Dataset 1</i>	<i>Dataset 2</i>	<i>Dataset 3</i>	<i>Dataset 4</i>	<i>Dataset 5</i>
Average free space in %	13.6	11.3	10.7	8.3	7.8
Average tile overlap in %	7.6	5.9	6.2	4.9	4.3
Minimum quality factor	0.67	0.79	0.79	0.79	0.85
Maximum quality factor	0.83	0.85	0.88	0.89	0.89
Average quality factor	0.79	0.83	0.83	0.86	0.87

5.1.2.3 Auswertung

Tabelle 5.2 zeigt eine Übersicht der Ergebnisse, die in den Versuchsreihen erzielt wurden. Dort sind – nach verwendetem Datensatz aufgeschlüsselt – sowohl der durchschnittliche, minimale als auch maximale Qualitätsfaktor Q , als auch dessen Teilkomponenten $Q_{overlap}$ und Q_{space} gesondert aufgeführt. In Abb. 5.2 ist eine grafische Gegenüberstellung der erzielten Qualitäten bei Verwendung der diversen Datensätze für jeden Container im Ziellayout gegeben. Im Anhang wurde zusätzlich unter Abb. A.1 für vier der Datensätze die relative Freifläche und Überlappung in den jeweiligen Zielcontainern als Grafiken zusammengefasst.

Anhand dieser Ergebnisse lässt sich eine Reihe an Beobachtungen machen, die für die Beurteilung des Verfahrens besonders aufschlussreich sind.

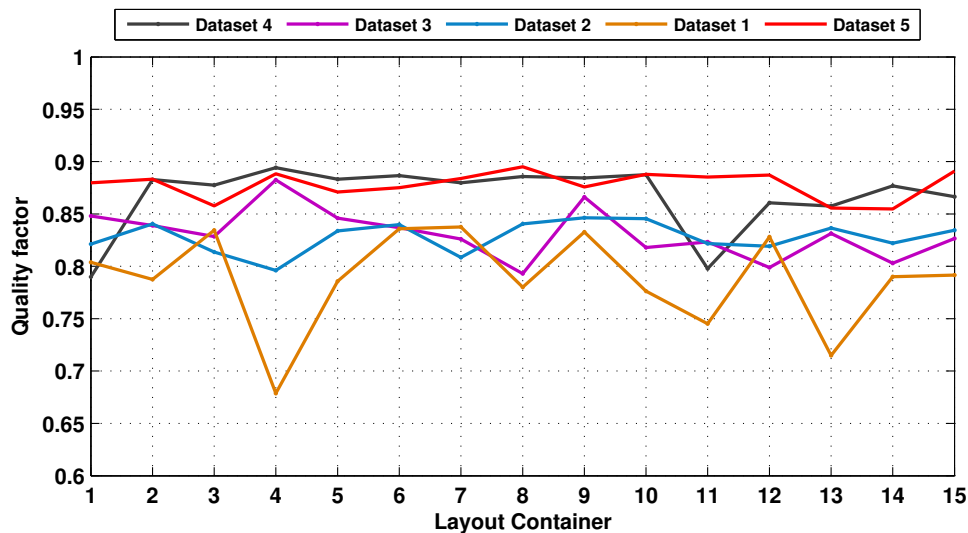


Abbildung 5.2: Gegenüberstellung des Qualitätsfaktors bei unterschiedlichen Datensätzen.

Grundeffizienz

In der Durchführung der Synthese unter Verwendung von Datensatz 1 ließ sich eine durchschnittliche Ausbeute von rund 80% (Qualitätsfaktor: 0.79) der Qualität einer optimalen Layoutkonfiguration erzielen. Diese Kennzahl setzt sich hier zusammen aus einer durchschnittlichen relativen Freifläche von unter 14% in den befüllten Containern und einer durchschnittlichen relativen Überlappung der Formelemente von unter 8% (siehe Tab. 5.2).

Datensatz 1 ist mit einer Größe von nur 10 Elementen dabei als außerordentlich klein einzustufen. Der Algorithmus liefert somit auch bei sehr kleinen Datensätzen ausgesprochen gute Ergebnisse in der strukturellen Synthese² – hier mit einem maximalen Qualitätsfaktor von 0.83.

Korrelation zur Datensatzgröße

Durch die Erhöhung der Datensatzgröße konnte eine geringe, aber wahrnehmbare Steigerung der erzielbaren Qualität verzeichnet werden. So ist etwa eine Qualitätszunahme von rund 4% bei einer Verdreifachung der Datensatzgröße zu beobachten. Bei einer Vergrößerung des Datensatzes um Faktor 50 wurde eine Zunahme der Qualität um insgesamt nur ca. 7% verzeichnet.

Auf Basis dieser Resultate lässt sich folglich ein Sättigungsverlauf hinsichtlich der erreichbaren Füllqualität vermuten. Näherungsweise liegt dieser Grenzwert nach Beobachtungen bei rund 87% einer optimalen Layoutkonfiguration. Abb. 5.3 zeigt die vermutete Korrelation zwischen Datensatzgröße und erzielbaren Ergebnissen anhand einer angepassten Kurve. Dabei handelt es sich um eine logistische Funktion.

Normalverhalten und Abweichungen

Abbildung 5.4 a zeigt eine Gegenüberstellung zweier befüllter Container, deren Ergebnisse der beobachteten statistischen Beziehung zwischen Datensatzgröße und durchschnittlich erzielter Qualität folgen.

Der größere Datensatz D4 (rechts) erzielt dabei ein besseres mess- und optisch wahrnehmbares Resultat gegenüber dem kleineren Datensatz D1 (links) bei Anwendung auf den gleichen Layoutcontainer C4. Während dies dem Regelfall entspricht, konnten vereinzelt Abweichungen in den Versuchsreihen registriert werden (siehe Abbildung 5.4 b). Der kleinere Datensatz D1 (links) erzielte dort einen höheren Qualitätsfaktor gegenüber dem um Faktor 10 größeren Datensatz D3 (rechts) bei Anwendung auf denselben Container C12.

Diese Beobachtung verdeutlicht, dass die Vielzahl der Kombinationsmöglichkeiten der unterschiedlichen Einflussfaktoren – wie Elementgrößen und Formen – bei endlich großen Datensätzen zu derartigen Abweichungen führen können. Eine solche Abweichung kann jedoch dann ausgeschlossen wer-

²Es ist zu beachten, dass eine Größe der Elemente (vgl. Skalierung) nicht in die Beurteilung der Qualität mit einbezogen wurde. Unter Einbezug dieses Faktors ist eine Veränderung der Ergebnisse zu erwarten.

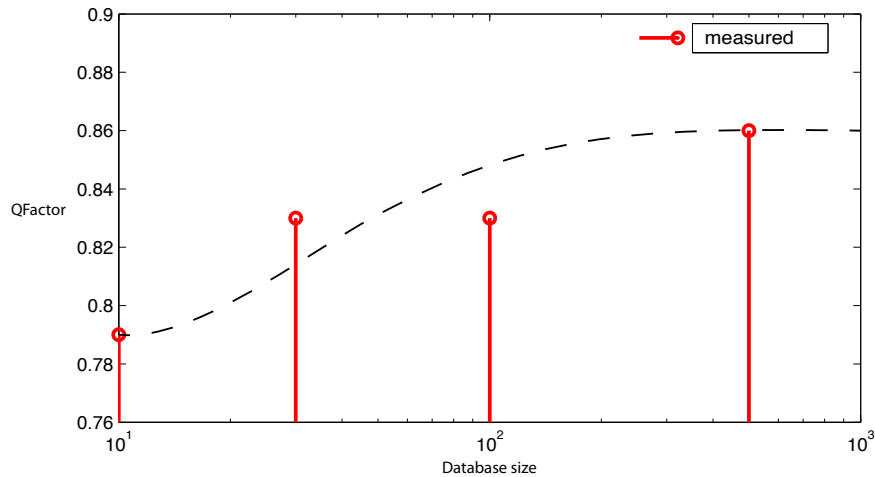


Abbildung 5.3: Vermutete Korrelation zwischen Datensatzgröße und erzielter Qualität der Synthese in Form einer Sigmoid Kurve. Die Messwerte folgen näherungsweise einer logistischen Funktion.

den, wenn der kleinere Datensatz stets eine echte Teilmenge des Größeren ist.

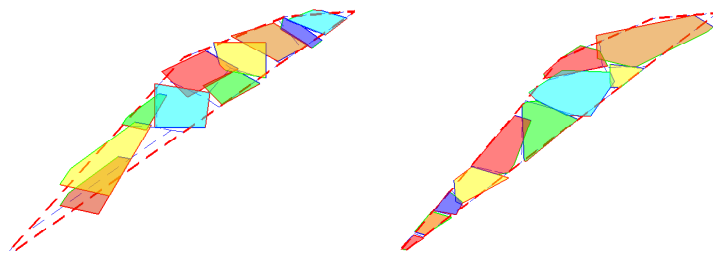
5.2 Qualitative Analyse

Die mit Hilfe des Syntheseverfahrens erzielbaren Resultate werden hier anhand drei verschiedener Layouts betrachtet. Diese wurden unter Verwendung unterschiedlicher realer Quelldatensätze erzeugt. Die Quelldatensätze wurden hierbei aus geographischem Kartenmaterial und Satellitenbilddaten erstellt.

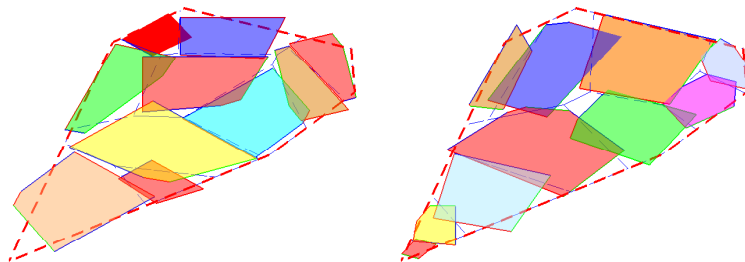
Gezeigt werden sollen drei unterschiedliche Anwendungsmöglichkeiten des Verfahrens. Darunter die Vergrößerung einer Stadt durch ‚Bebauung‘ freier Flächen (Abb. 5.5), die Veränderung eines Städtebilds durch das Umgestalten bereits bestehender Nutzflächen (Abb. 5.6) und zuletzt die vollständigen Erzeugung einer fiktiven Metropole (Abb. 5.7). Tabelle 5.3 listet die wichtigsten Merkmale der dazu verwendeten Datensätze.

Tabelle 5.3: Merkmale der realen Quelldatensätze.

	<i>Rome</i>	<i>Chicago</i>	<i>Rio de Janeiro</i>
Number of tiles	~ 30	~ 50	~ 60



(a) Ergebnisse der Anwendung von D1 (links) und D4 (rechts) jeweils auf C4.



(b) Ergebnisse der Anwendung von D1 (links) und D3 (rechts) jeweils auf C12.

Abbildung 5.4: Einfluss der Datensatzinhalte auf die Qualität der Synthese. Einmal als zu erwartendes Normalverhalten (oben) und einmal als Abweichung (unten). Die erzielten Qualitätsfaktoren können Tab. 5.2 entnommen werden.

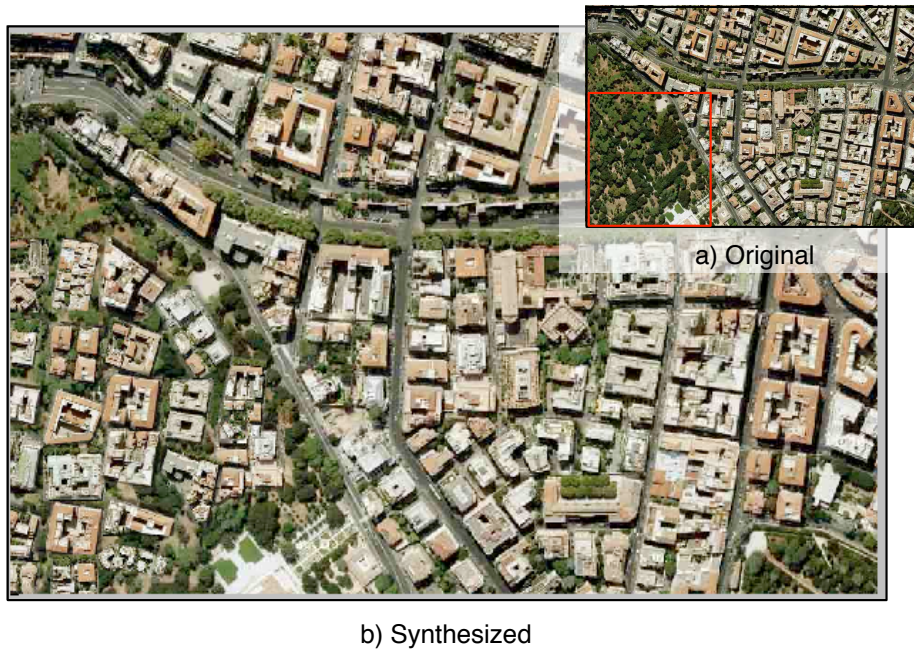


Abbildung 5.5: Erweiterte Stadt. Gezeigt wird ein Vorort von Rom. In diesem Anwendungsbeispiel wurde ein bestehendes Städtelayout künstlich erweitert. Dazu wurde eine unbebaute Nutzfläche im Ziellayout als Container bestimmt und die Synthese darauf angewendet. Besonders ist hierbei, dass das Ziellayout und das Quelllayout (a) identisch sind. Somit konnte die Erweiterung der Stadt in einem einheitlichen Stil ermöglicht werden. Der Algorithmus verwendet also dieselben Bildfragmente mehrmals, was bei sehr ‚generischen‘ Gebäudetypen jedoch kaum bemerkbar ist. Die größeren Abstände zwischen den Gebäuden sind dabei gewollt und wurden durch Parameter in der Synthese bewusst herbeigeführt. Eine Schwäche des Verfahrens zeigt sich allerdings in den Leerstellen, da dort kein plausibler Inhalt vorhanden ist. Mehrere Ansätze zur Lösung dieser Problematik sind denkbar, z.B. die Verwendung von ‚Graph-cut‘ Techniken (mehr dazu im Ausblick unter Abschn. 6.1).

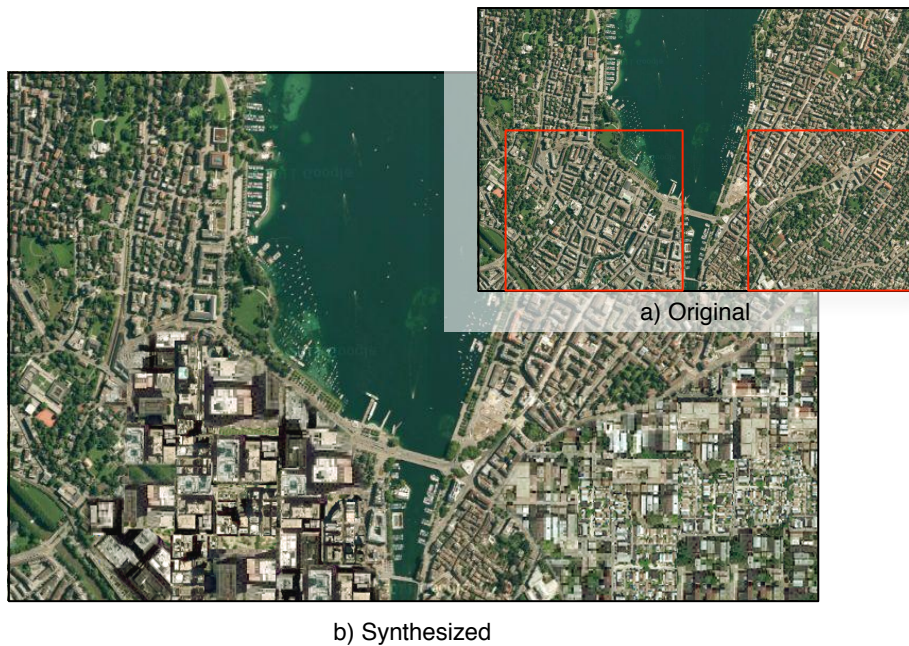


Abbildung 5.6: Veränderte Stadt. Anhand dieses Beispiels soll das Potenzial des Verfahrens zur Simulation von Flächennutzung in urbanen Lebensräumen verdeutlicht werden. Abgebildet ist die mittelgroße Stadt Zürich. Mit Hilfe des Testprogramms wurden zwei größere Flächenabschnitte aus dem Layout markiert und die Synthese darauf ausgeführt. Als Vorlage für den Quelldatensatz diente die Stadt Chicago, die mit Hochhäusern zum einen und dicht bebauten Einfamilienhäusern zum anderen ein deutlich abweichendes Städtebild bietet. Das Resultat ist eine hypothetische Sicht auf Zürich in einigen Jahren. Zur Verbesserung des Bildeindrucks wurden die Randbereiche der Regionen nachträglich manuell bearbeitet und außerdem eine Farbanpassung vorgenommen. Das Innere der synthetisierten Regionen ist jedoch voll automatisch erzeugt worden. Insgesamt bietet das Resultat eine gute Visualisierung. Lediglich die Beschneidung einiger Häuser an den Randbereichen ist störend. Hier müssen Verbesserungen im Platzierungsalgorithmus in Betracht gezogen werden.

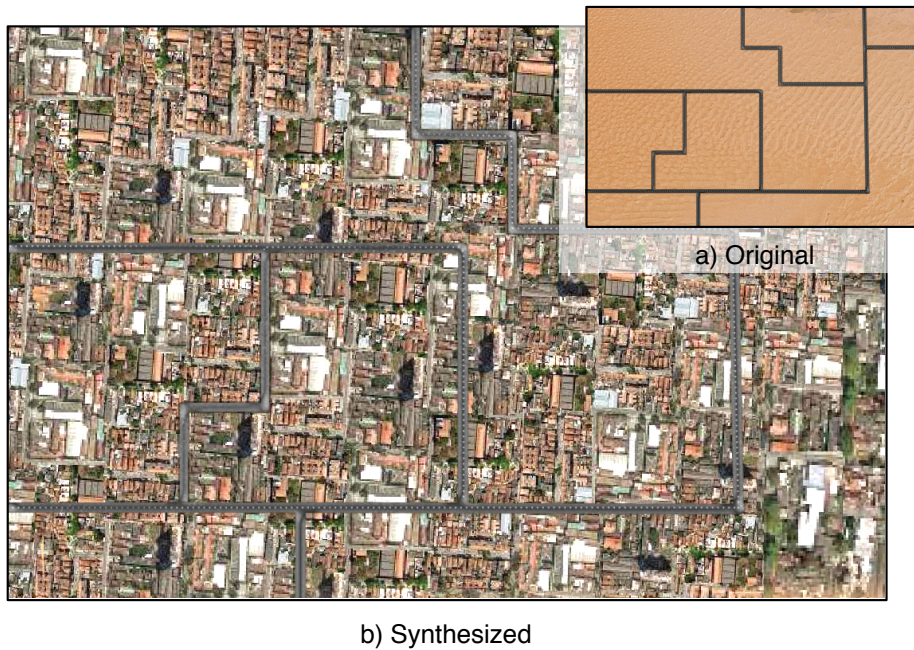


Abbildung 5.7: Fiktive Stadt. Die Abbildung zeigt einen Ausschnitt eines vollständig synthetisch erzeugten Städtelayouts. Das original Ziellayout (a) wurde mit einem Grafikprogramm erstellt und beinhaltet lediglich ein einfaches Straßennetz auf einer Wüstenfläche. Mit Anwendung des Syntheseverfahrens unter Verwendung des Datensatzes ‚Rio‘ konnte das Abbild einer dicht bebauten Metropole erzeugt werden. Obwohl das Bild auf Distanz einen guten Eindruck vermittelt, existieren einige störende Mängel. Zum einen wurden bewusst Überlappungen in Kauf genommen (\rightarrow geringer gewichtet), um keine Lücken im Layout entstehen zu lassen. Dies führt teilweise zu unfeinen Kanten. Um den Effekt abzumildern, wurde ein geringfügiges *Blending* zwischen den Elementen angewendet. Zum anderen zeigt sich eine erkennbare Wiederholung einiger herausstechender Gebäude auf Makroebene. Dies kann mit der geringen Größe des Datensatzes begründet werden und lässt sich daher nur effektiv mit einer Vergrößerung des Quelldatensatzes beseitigen.

Kapitel 6

Zusammenfassung

Der Fokus dieser Arbeit richtete sich auf Methoden und Techniken zur Erzeugung und Visualisierung von Städtelayouts. Eingangs wurden dazu in Kapitel 2 die größten Strömungen im Forschungsumfeld des Urban Modelling bzw. der Urban Visualization gegenübergestellt – namentlich das *Prozedurale Modelling* und die *Beispielbasierte Synthese*.

Im weiteren Verlauf dieser Arbeit wurde dann ein neuartiger Ansatz zur automatisierten Erstellung eines Teilbereiches künstlicher Städtelayouts beschrieben. Mit diesem Verfahren lassen sich hypothetische Ansichten von strukturierten Flächen (z.B. Wohn- oder Industrieflächen) in urbanen Layouts in Form synthetisch generierter Satellitenbilder produzieren. Derartige Daten können vor allem für die Anwendung in der Städteplanung und Simulation, aber auch in der Unterhaltungsindustrie von Nutzen sein.

Neues Syntheseverfahren

Das beschriebene Verfahren operiert hierbei auf Basis von Analogien zwischen geographischen Beispieldatensätzen und ermöglicht es, visuelle Inhalte bestehender urbaner Layouts intelligent auf neue Layouts zu übertragen. Damit weist es Ähnlichkeiten zu einigen Verfahren aus dem Bereich der Pixelbasierten Bild- und Textursynthese auf (vgl. [Hertzmann et al., 2001]).

Zu den größten Neuerungen im Vergleich zu artverwandten Ansätzen aus dem Urban Modelling (vgl. [Aliaga et al., 2007]) zählen dabei, dass erstmals komplexe Anordnungen von Grundstücksflächen in Layouts Berücksichtigung fanden, indem auf a priori Segmentierungen der Layouts verzichtet wurden. Hierzu wurde ein Teil der Synthese als *Shape-packing* Problem aufgefasst und mit dem *Image Mosaics* Verfahren (siehe [Kim and Pellacini, 2002]) ein effizienter Ansatz zur Lösung dieser Problemstellung adaptiert. Damit lassen sich u.a. komplexe und asymmetrische Layouts nach europäischem Vorbild erzeugen.

Als zweite Neuerungen gilt der vollständige Einbezug unterschiedlicher äußerer Randbedingungen in die Synthese, insbesondere in Form semanti-

scher Daten. Um das Platzierungsverfahren für die komplexen Beziehungen in urbanen Layouts tauglich zu machen, wurde das aus der Bildsynthese stammende Verfahren um einen Constraint-basierten Aufbau erweitert. Insbesondere wurde ein eigens entwickelter Ansatz zum Abgleich von Kantenfeatures in räumlichen Anordnungen in Kombination mit einem energie-basierten Lösungsansatz zur Bestimmung optimaler Layoutkonfigurationen vorgestellt.

Als Ausgangssituation für dieses erweiterbare Framework sind einige Features, die als relevant erachtete wurden, ausgewählt und genauer erörtert worden. Für alle Features wurden dabei jeweils im Detail Methoden zur Gewinnung, Speicherung und zum Abgleich dieser mit Hilfe geeigneter Distanzmetriken vorgestellt. Neben Low-Level Features in Form von Farbüber-einstimmungen (siehe Abschn. 3.2.1.3) an Kanten, wurde ausführlich auf die Bedeutung der ganzheitlichen Betrachtung topologischer und semantischer Eigenschaften in Layouts eingegangen (Abschn. 3.2.1.2 / 3.2.2.5). Insbesondere die Verknüpfung semantischer Daten mit einer räumlichen Komponente und der Einbezug sich daraus ergebender Constraints in das Syntheseverfahren sind in bisherigen Ansätzen häufig vernachlässigt worden und stellen eine Besonderheit dieses Verfahrens dar.

Für die Implementierung wurde auf effiziente Algorithmen und kompakte Datenstrukturen zurückgegriffen. Nennenswert ist hierbei das *Geometric Hashing*, das den Rechenaufwand zum Vergleich beliebig komplexer Formen mittels einer diskret quantifizierten Hashmasken merklich reduziert.

Ergebnisse

Die Auswertung hat gezeigt, dass sich mit dem Verfahren schon bei sehr kleinen Quelldatensätzen gute Ergebnisse erzielen lassen.

Bei größeren Datensätzen wurde sogar ein Fehler von im Schnitt nur ca. 10% Abweichung zu einer optimalen Layoutkonfiguration ermittelt. Allerdings konnte im Rahmen dieser Arbeit keine ausführliche Analyse unter Verwendung verschiedener Datensatztypen vorgenommen werden, weshalb an dieser Stelle kein abschließendes Fazit über die objektive Qualität des Verfahrens gezogen wird.

Neben der Quantitativen Analyse konnte durch Abbildung diverser mit dem Verfahren produzierter Resultate ein Bild über die Funktionstüchtigkeit des Verfahrens geliefert werden. Es wurden dabei verschiedene Anwendungsszenarien betrachtet. Darunter u.a. die Erweiterung eines bestehenden Städtelayouts im gleichen Stil, die Veränderung einzelner Flächen eines Layouts nach einem abweichenden Stil und schließlich die Erzeugung einer kompletten fiktiven Stadt von Grund auf.

Zum Großteil konnten dabei plausible und ansprechende Ergebnisse produziert werden. Die dem Verfahren intrinsischen Fehler in der Platzierung – die stets auftreten, messbar jedoch gering sind – führen allerdings zu einer

unverhältnismäßig schlechteren, subjektiven Wahrnehmung. Es wird jedoch angenommen, dass sich diese Fehler (etwa Lücken im Layout) relativ leicht durch weitere Techniken als Ergänzung zu diesem Verfahren beheben lassen (s. Abschn. 6.1).

Es kann abschließend festgehalten werden, dass das dargelegte Produktionsverfahren im Bereich der Visualisierung von Städten einen bisher kaum berücksichtigten Ansatz verfolgt.

Diese Nische zwischen der manuellen Herstellung virtueller, urbaner Layouts und den regelbasierten Techniken des Prozeduralen Modelling, birgt großes Potential. Wie gezeigt werden konnte, können mit einem beispielbasierten Verfahren wie diesem eine Vielzahl an Anwendungsgebieten ohne zusätzlichen Mehraufwand abgedeckt werden. Neben dieser Vielfältigkeit, zählen insbesondere der geringe nötige Arbeitsaufwand, sowie der geringe initiale Input¹ zu seinen größten Stärken.

Die produzierten Beispiele haben zudem gezeigt, dass eine durchaus annehmbare Qualität in den Resultaten erzielt werden kann. Ob diese Qualität ausreichend ist, ist hierbei allerdings vom jeweiligen Anwendungsfall abhängig. So kann vermutet werden, dass sich das Verfahren zumindest zur groben Visualisierung von Städtelayouts einsetzen lässt. Selbstkritisch bleibt anzumerken, dass gerade zur Herstellung detailgetreuer und ästhetisch vollkommen stimmiger Layouts noch viele Verbesserungen notwendig sind. Der beschriebene Ansatz bildet hierzu jedoch den ersten nötigen Schritt.

6.1 Ausblick & Verbesserungen

Der Fokus im Bereich der Städtevisualisierung liegt derzeit unzweifelhaft auf dem *Prozeduralen Modelling*. Wie in Kapitel 2 zur Verwandten Forschung bereits erläutert, haben diese Ansätze den Nachteil, dass sie ein sehr hohes Maß an Kontextwissen und manueller Vorbereitung voraussetzen, um überhaupt plausible Resultate damit erzielen zu können. Zudem lassen sich damit nur schwer der hohe Detailgrad und die Vielfalt der mittels *Beispielbasierter Synthese* produzierbaren Ergebnisse erreichen.

Derartige Nachteile sprechen für die Daseinsberechtigung des in dieser Arbeit erläuterten sowie vergleichbarer, beispielbasierter Syntheseverfahren zur Erzeugung und Visualisierung urbaner Szenerien. Langfristig betrachtet wird allerdings eine Verschmelzung der beiden Herangehensweisen stattfinden müssen, welche die Vorteile beider vereint. Auf diese Entwicklung kann an dieser Stelle nicht näher eingegangen werden. Stattdessen sollen nur einige sinnvolle konkrete Erweiterungen und Verbesserungen des vorgestellten Verfahrens genannt werden:

¹Sofern ausreichend detaillierte GIS Daten bereits verfügbar sind.

- **Befüllung von Leerstellen**

Der Platzierungsalgorithmus kann in den wenigsten Fällen eine hundertprozentige Flächenausnutzung erzielen. So verbleiben meist geringe Freiflächen in den generierten Layouts.

Eine denkbare Erweiterung wäre im einfachsten Fall ein geeignetes Warping-Verfahren. Dieses verformt die platzierten Elemente so, dass die Freifläche vollständig verschwindet. Optimalerweise muss die Deformation dazu möglichst ausgeglichen über alle Elemente verteilt werden, um deren lokale Deformation zu minimieren. Anstelle von Warping könnten aber auch komplexere Verfahrensweisen Anwendung finden. Insbesondere die Verwendung von Graph-cut Techniken (s. [Kwatra et al., 2003]) an überlappenden Elementkanten und/oder die Verwendung von Methoden aus der Textursynthese (vgl. [Efros and Freeman, 2001]) zur künstlichen Befüllung der Leerstellen sind hier denkbar.

- **Verbesserte Stilreproduktion**

Ein wesentliches Merkmal des beschriebenen Syntheseverfahrens ist dessen Anwendbarkeit auf unterschiedlichste Datensätze. Diese Flexibilität wird u.a. durch den Verzicht auf vorab segmentierte Layouts erreicht. Der strukturelle Stil des Layouts wird dabei implizit über die Merkmale des Quelldatensatzes transportiert und durch die Techniken zur Wahrung der Layoutkonsistenz (z.B. Abgleich der Randbedingungen) im Ziellayout wieder hergestellt.

Diese implizite Vorgehensweise kann jedoch auch zu stilistisch abweichenden Anordnungen führen, wenn diese durch den Algorithmus als gleichfalls konsistent eingestuft werden.

Eine denkbare Erweiterung wäre die Extraktion der Platzierungstopologien aus dem Quelldatensatz und der Einbezug dieser Information in den Platzierungsalgorithmus – etwa bei der Bestimmung eines geeigneten Platzierungsortes. Ohne die Nachteile vorab segmentierter Layouts in Kauf nehmen zu müssen, könnte das Syntheseverfahren so besser gesteuert werden, wodurch der Stil des Quelllayouts im Ziellayout exakter reproduziert werden könnte.

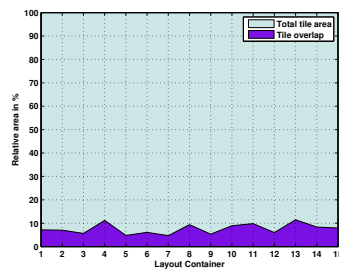
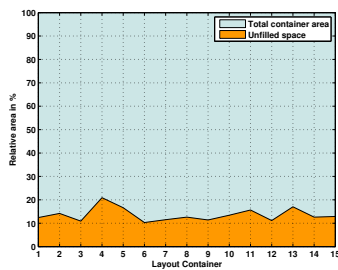
- **Effizienzsteigerung im Matching**

Die Auswertung (siehe Abschn. 5.2) hat gezeigt, dass bereits kleine Datensätze in der Gegenüberstellung mit größeren zu ähnlich guten Resultaten führen. Dieser Umstand kann insbesondere auf eine i.d.R. geringe geometrische Variation der in ihnen enthaltenen Formelemente zurückgeführt werden (siehe Normalverteilung in Abb. 5.1). Bei großen Datensätzen ist es daher von Nachteil, einen Formabgleich (hier durch Geometric Hashing) mit sämtlichen Elementen eines Datensatzes durchzuführen. Stattdessen könnte die Matching-Effizienz durch vorab Ermittlung repräsentativer Formelemente verbessert werden.

Die verschiedenen Elemente könnten dann durch Clustering dieser unter einer der repräsentativen Formelemente in einer hierarchischen Baumstruktur eingeordnet werden, wodurch der Abgleich auf einzelne Teile des Datensatzes beschränkt werden würde. Dies würde zu einer deutlich gesteigerten Sucheﬃzienz führen.

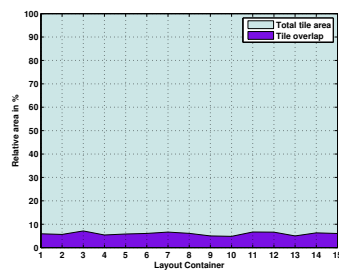
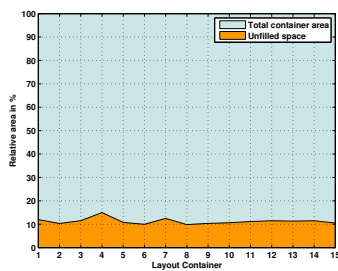
Anhang A

Detaillierte Auswertung



$\epsilon_{min} : 10.31 \mid \epsilon_{max} : 20.91 \mid \sigma : 2.88$ $\epsilon_{min} : 4.71 \mid \epsilon_{max} : 11.49 \mid \sigma : 2.22$

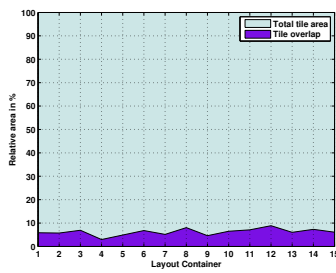
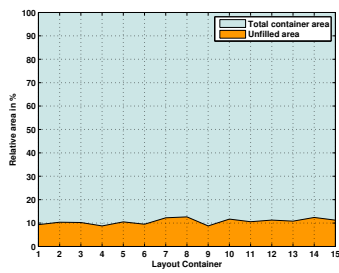
Datensatz 1



$\epsilon_{min} : 9.84 \mid \epsilon_{max} : 14.98 \mid \sigma : 1.27$ $\epsilon_{min} : 4.82 \mid \epsilon_{max} : 7.13 \mid \sigma : 0.68$

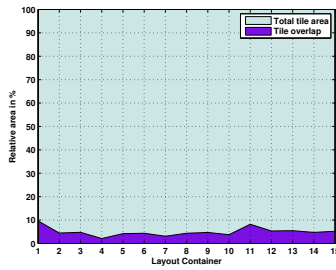
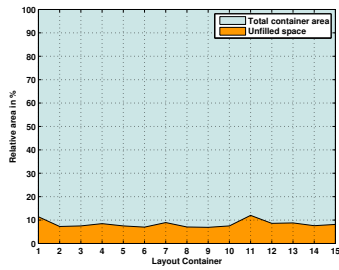
Datensatz 2

Abbildung A.1: Graphische Auswertung von Flächennutzung und Überlappung bei Anwendung der unterschiedlichen Datensätze (1–4) im Syntheseverfahren. Das ermittelte Werteminimum ϵ_{min} , respektive Maximum ϵ_{max} sowie die Standardabweichung σ der Werte sind dabei zusätzlich aufgeführt.



$\epsilon_{min} : 8.77 \mid \epsilon_{max} : 12.65 \mid \sigma : 1.24$ $\epsilon_{min} : 2.95 \mid \epsilon_{max} : 8.85 \mid \sigma : 1.45$

Datensatz 3



$\epsilon_{min} : 6.87 \mid \epsilon_{max} : 12.01 \mid \sigma : 1.53$ $\epsilon_{min} : 2.11 \mid \epsilon_{max} : 9.62 \mid \sigma : 1.84$

Datensatz 4

Abbildung A.1: s.o.

Anhang B

Testprogramm

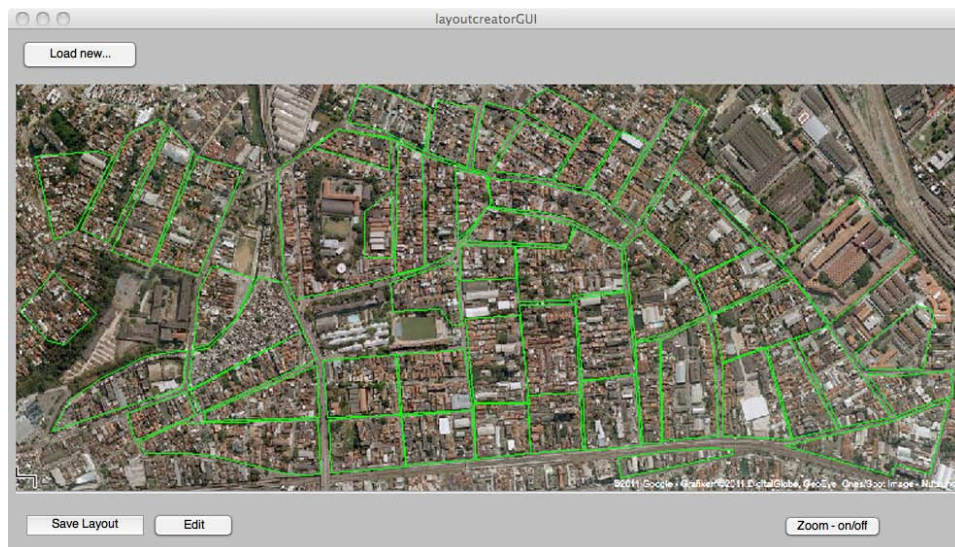


Abbildung B.1: Als Zusatzfeature bietet das Framework einen interaktiven Modus zum Erzeugen von Quelldatensätzen. Dazu kann über *Load new* ein Satellitenbild von der Festplatte geladen und angezeigt werden. Unterstützt werden alle gängigen Bildformate (z.B. *.jpg, *.png, *.tiff). Der Nutzer kann über den *Edit* Knopf ein Tool zum interaktiven Einzeichnen von Grundstücksbegrenzung o.ä. geographischen Flächen aufrufen. Über *Save Layout* kann nach Beendigung das komplette Layout in eine Datenbank Datei (*.db) gespeichert werden. Diese Datenbank kann vom Hauptprogramm gelesen und als Quelldatensatz für die Synthese verwendet werden.

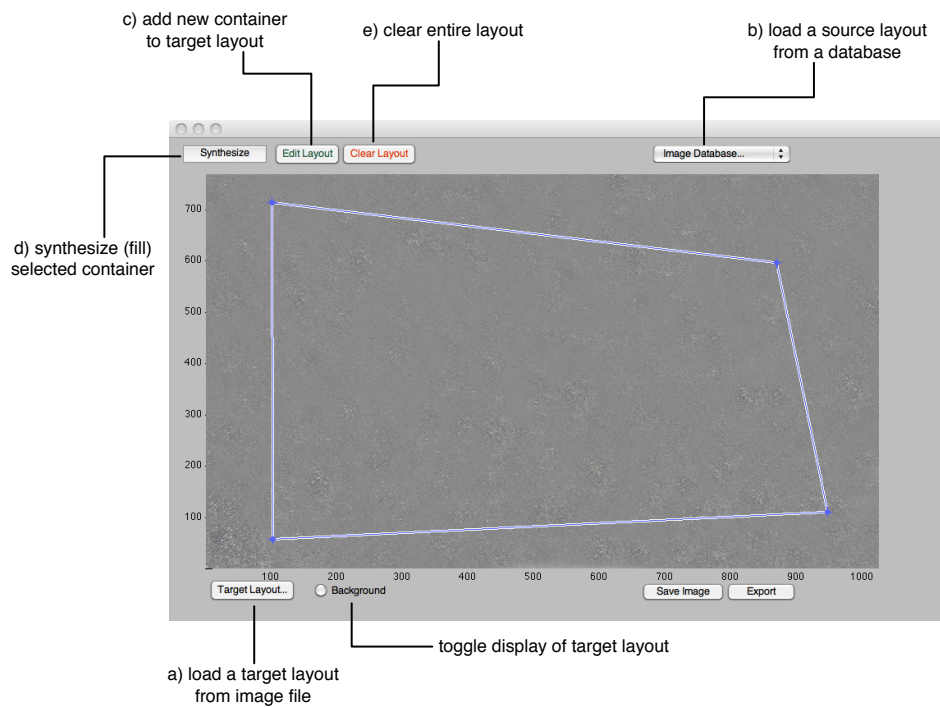


Abbildung B.2: Mit dem Testprogramm können interaktiv Ziellayouts erstellt, bearbeitet und das Syntheseverfahren selektiv auf einzelne Container angewendet werden. Es bietet eine gute Umgebung zum Testen von verschiedenen Aspekten des Algorithmus. Die Abbildung zeigt die wichtigsten Bedienelemente der Benutzeroberfläche. Zuerst muss über *a)* ein Bild als Ziellayout geladen werden. Über *b)* muss dann eine bestehende Quelldatenbank gewählt werden. Über *c)* wird schließlich ein interaktives Tool zum Platzieren von Zielcontainern im Layout aufgerufen. Dieser Vorgang kann beliebig oft wiederholt werden. Mittels *d)* kann jeweils auf den aktuell ausgewählten Zielcontainer die Synthese angewendet werden. Optional lässt sich mit *e)* das erzeugte Layout vollständig zurücksetzen.

Anhang C

Inhalt der CD-ROM

Format: CD-ROM, Single Layer, ISO9660-Format

C.1 PDF-Dateien

Pfad: /

Diplomarbeit_Stuber_Reto_2011.pdf Diplomarbeit (Druckqualität)

C.2 Quelltext

Pfad: /source

/projectI/SatSynthesizer C++ Quelltext & Applikation
(Masterprojekt I)

/projectII *MATLAB* Quelltext (Masterprojekt II)

C.3 Versuchsdaten

Pfad: /testing

*.db Quelldatensätze der Versuchsreihen
(s. Abschn. 5.1)

/images Bilddateien (Datensatz Rom)

/results/*.result Ergebnisse der Versuchsreihen

/results/*.analysis Auswertung der Ergebnisse

Hinweis: Diese Dateien sind im Standard *MATLAB* Binärformat kodiert.

Literaturverzeichnis

- ALIAGA, D. G., BENEŠ, B., VANEGAS, C. A., AND ANDRYSKO, N. 2008. Interactive reconfiguration of urban layouts. *IEEE Comput. Graph. Appl.* 28, 38–47.
- ALIAGA, D. G., ROSEN, P. A., AND BEKINS, D. R. 2007. Style grammars for interactive visualization of architecture. *IEEE Transactions on Visualization and Computer Graphics* 13, 786–797.
- ALIAGA, D. G., VANEGAS, C. A., AND BENEŠ, B. 2008. Interactive example-based urban layout synthesis. *ACM Trans. Graph.* 27, 160:1–160:10.
- AVIDAN, S. AND SHAMIR, A. 2007. Seam carving for content-aware image resizing. *ACM Trans. Graph.* 26, 3, 10:1–10:9.
- BECHHOFFER, S., VAN HARMELEN, F., AND HENDLER, J. 2004. OWL web ontology language reference. <http://www.w3.org/TR/owl-ref/>.
- CASTELLI, V. AND BERGMAN, D. L. 2002. *Image Databases: Search and Retrieval of Digital Imagery*. John Wiley & Sons, Inc.
- CHOMSKY, N. 1959. On certain formal properties of grammars. *Information and Control* 2, 2, 137–167.
- COHEN, M. F., SHADE, J., HILLER, S., AND DEUSSEN, O. 2003. Wang tiles for image and texture generation. In *ACM SIGGRAPH 2003 Papers*. SIGGRAPH '03. ACM, New York, NY, USA, 287–294.
- COHEN, S. D. AND GUIBAS, L. J. 1997. Shape-based image retrieval using geometric hashing. In *Proceedings of the ARPA Image Understanding Workshop*. Morgan Kaufmann Publishers Inc., 669–674.
- DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. SIGGRAPH '96. ACM, New York, NY, USA, 11–20.

- DIJKSTRA, E. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 1, 269–271.
- EFROS, A. A. AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '01. ACM, New York, NY, USA, 341–346.
- EFROS, A. A. AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision*. ICCV '99 Series, vol. 2. IEEE Computer Society, Washington, DC, USA, 1033–1038.
- EGENHOFER, M., SHARMA, J., AND MARK, D. 1993. A critical comparison of the 4-intersection and 9-intersection models for spatial relations: Formal analysis. *Auto-Carto 11*, 1–11.
- EHRIG, H., ENGELS, G., KREOWSKI, H.-J., AND ROZENBERG, G., Eds. 1999. *Handbook of graph grammars and computing by graph transformation: vol. 2: applications, languages, and tools*. World Scientific Publishing Co., Inc., River Edge, NJ, USA.
- FANG, H. AND HART, J. C. 2007. Detail preserving shape deformation in image editing. *ACM Trans. Graph.* 26, 3, 12:1–12:5.
- HAMMING, R. W. 1950. Error detecting and error correcting codes. *Bell System Technical Journal* 26, 2, 147–160.
- HEEGER, D. J. AND BERGEN, J. R. 1995. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. SIGGRAPH '95. ACM, New York, NY, USA, 229–238.
- HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLISS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '01. ACM, New York, NY, USA, 327–340.
- HOOGS, A. 1996. Model-based learning of segmentations. In *Proceedings of the International Conference on Pattern Recognition (ICPR '96)*. Vol. 4. IEEE Computer Society, Washington, DC, USA, 494–499.
- HUERTAS, A. AND NEVATIA, R. 1988. Detecting buildings in aerial images. *Computer Vision, Graphics and Image Processing Journal* 41, 131–152.
- IJIRI, T., MĚCH, R., IGARASHI, T., AND MILLER, G. 2008. An example-based procedural system for element arrangement. *Computer Graphics Forum* 27, 2, 429–436.

- ISO. 2002. ISO 19113:2002: Geographic information – quality principles. http://www.iso.org/iso/catalogue_detail.htm?csnumber=26018.
- JUN, C., CHENGMING, L., ZHILIN, L., AND CM, G. 2000. Improving 9-intersection model by replacing the complement with voronoi region. *Geo-Spatial Information Science* 3, 1–10. 10.1007/BF02826800.
- KIM, J. AND PELLACINI, F. 2002. Jigsaw image mosaics. *ACM Trans. Graph.* 21, 657–664.
- KNUTH, D. 1968. Semantics of context-free languages. *Theory of Computing Systems* 2, 2, 127–145.
- KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.* 22, 277–286.
- LOBIN, H. AND LEMNITZER, L. 2004. *Texttechnologie. Perspektiven und Anwendungen*. Tübingen, Stauffenburg.
- LOPES, R., TUTENEL, T., SMELIK, R., DE KRAKER, K., AND BIDARRA, R. 2010. A constrained growth method for procedural floor plan generation. In *Proceedings of GAME-ON 2010*. EUROSIS, Leicester, United Kingdom, 13–23.
- MA, W. Y. AND MANJUNATH, B. S. 1997. Edge flow: A framework of boundary detection and image segmentation. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*. CVPR '97. IEEE Computer Society, Washington, DC, USA, 744–749.
- MANOLA, F. AND MILLER, E. 2004. RDF primer. <http://www.w3.org/TR/rdf-primer/>.
- MOORE, M. AND WILHELMS, J. 1988. Collision detection and response for computer animation. *SIGGRAPH Comput. Graph.* 22, 289–298.
- MÜLLER, P., WONKA, P., HAEGLER, S., ULMER, A., AND VAN GOOL, L. 2006. Procedural modeling of buildings. In *ACM SIGGRAPH 2006 Papers*. SIGGRAPH '06 Series, vol. 25. ACM, New York, NY, USA, 614–623.
- MÜLLER, P., ZENG, G., WONKA, P., AND VAN GOOL, L. 2007. Image-based procedural modeling of facades. *ACM Trans. Graph.* 26, 3, 85:1–85:9.
- OSWIN, A., FRANZ, A., DAVID, A., AND BERND, G. 1995. A novel type of skeleton for polygons. *Journal of Universal Computer Science* 1, 752–761.

- PARISH, Y. I. H. AND MÜLLER, P. 2001. Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '01. ACM, New York, NY, USA, 301–308.
- POPAT, K. AND PICARD, R. W. 1993. Novel cluster-based probability model for texture synthesis, classification, and compression. In *Visual Communications and Image Processing*. 756–768.
- PRUSINKIEWICZ, P. AND LINDENMAYER, A. 1990. *The algorithmic beauty of plants*. Springer-Verlag Inc., New York, NY, USA.
- RUSSELL, S. AND NORVIG, P. 1995. *Artificial Intelligence: A Modern Approach* 1 Ed. Prentice Hall.
- SANCHEZ, S., LE ROUX, O., LUGA, H., AND GAILDRAT, V. 2003. Constraint-based 3d-object layout using a genetic algorithm. In *Proceedings of the 6th 3IA*. 3IA - International Conference on Computer Graphics and Artificial Intelligence, Limoges.
- SMITH, G., SALZMAN, T., AND STUERZLINGER, W. 2001. 3d scene manipulation with 2d devices and constraints. In *Graphics Interface 2001*. GRIN'01. Canadian Information Processing Society, Toronto, Ont., Canada, 135–142.
- SMITH, J. R. AND CHANG, S.-F. 1996. Tools and techniques for color image retrieval. In *Storage and Retrieval for Image and Video Databases IV*. Vol. 2670. SPIE, Image Rochester, NY, 426–437.
- STAM, J. 1997. Aperiodic texture mapping. Tech. rep., European Research Consortium for Informatics and Mathematics (ERCIM).
- STINY, G. 1975. *Pictorial and Formal Aspects of Shape and Shape Grammars*. Birkhauser Verlag, Basel.
- STINY, G. 1980. *Introduction to shape and shape grammars*. Vol. 7. Pion Ltd., London, 343–361.
- STRICKER, M., DIMAI, A., AND DIMAI, E. 1996. Color indexing with weak spatial constraints. In *Storage and Retrieval for Image and Video Databases IV*. Vol. 2670. SPIE, 29–40.
- STRICKER, M. AND ORENGO, M. 1995. Similarity of color images. In *Storage and Retrieval for Image and Video Databases III*. Vol. 2420. SPIE, 381–392.
- SWAIN, M. J. AND BALLARD, D. H. 1991. Color indexing. *International Journal of Computer Vision* 7, 1, 11–32.

- TUTENEL, T., BIDARRA, R., SMELIK, R., AND DE KRAKER, K. 2009. Rule-based layout solving and its application to procedural interior generation. In *Proceedings of the CASA'09 Workshop on 3D Advanced Media in Gaming and Simulation*. 3AMIGAS, Amsterdam, 15–24.
- VANEGAS, C. A., ALIAGA, D. G., BENEŠ, B., AND WADDELL, P. 2009. Visualization of simulated urban spaces: Inferring parameterized generation of streets, parcels, and aerial imagery. *IEEE Transactions on Visualization and Computer Graphics* 15, 424–435.
- VANEGAS, C. A., ALIAGA, D. G., WONKA, P., MÜLLER, P., WADDELL, P., AND WATSON, B. 2010. Modeling the appearance and behavior of urban spaces. *Computer Graphics Forum* 29, 1, 25–42.
- WONKA, P., WIMMER, M., SILLION, F., AND RIBARSKY, W. 2003. Instant architecture. *ACM Trans. Graph.* 22, 669–677.
- XU, K., STEWART, J., AND FIUME, E. 2002. Constraint-based automatic placement for scene composition. In *Graphics Interface 2002*. 25–34.
- ZHANG, J., ZHOU, K., VELHO, L., GUO, B., AND SHUM, H.-Y. 2003. Synthesis of progressively-variant textures on arbitrary surfaces. *ACM Trans. Graph.* 22, 295–302.
- ZHAO, R. AND CHEN, J. 1999. Enhancing v9i model for spatial relationships with voronoi distance. In *Proceedings of the International Symposium on Digital Earth*. Vol. 2. Science Press.