



Pose-Tracking Forschungsbericht

Dietrich Paulus
Lutz Prieße
Peter Decker
Frank Schmitt

Nr. 17/2009

**Arbeitsberichte aus dem
Fachbereich Informatik**

Die Arbeitsberichte aus dem Fachbereich Informatik dienen der Darstellung vorläufiger Ergebnisse, die in der Regel noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar. Alle Rechte vorbehalten, insbesondere die der Übersetzung, des Nachdruckes, des Vortrags, der Entnahme von Abbildungen und Tabellen – auch bei nur auszugsweiser Verwertung.

The “Arbeitsberichte aus dem Fachbereich Informatik“ comprise preliminary results which will usually be revised for subsequent publication. Critical comments are appreciated by the authors. All rights reserved. No part of this report may be reproduced by any means or translated.

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346

ISSN (Online): 1864-0850

Herausgeber / Edited by:

Der Dekan:
Prof. Dr. Zöbel

Die Professoren des Fachbereichs:

Prof. Dr. Bátori, Prof. Dr. Beckert, Prof. Dr. Burkhardt, Prof. Dr. Diller, Prof. Dr. Ebert, Prof. Dr. Furbach, Prof. Dr. Grimm, Prof. Dr. Hampe, Prof. Dr. Harbusch, Prof. Dr. Sure, Prof. Dr. Lämmel, Prof. Dr. Lautenbach, Prof. Dr. Müller, Prof. Dr. Oppermann, Prof. Dr. Paulus, Prof. Dr. Priese, Prof. Dr. Rosendahl, Prof. Dr. Schubert, Prof. Dr. Staab, Prof. Dr. Steigner, Prof. Dr. Troitzsch, Prof. Dr. von Kortzfleisch, Prof. Dr. Walsh, Prof. Dr. Wimmer, Prof. Dr. Zöbel

Kontakt Daten der Verfasser

Dietrich Paulus, Lutz Priese, Peter Decker, Frank Schmitt

Institut für Computervisualistik

Fachbereich Informatik

Universität Koblenz-Landau

Universitätsstraße 1

D-56070 Koblenz

E-Mail: paulus@uni-koblenz.de, priese@uni-koblenz.de, decker@uni-koblenz.de, fschmitt@uni-koblenz.de

Inhaltsverzeichnis

1	Zusammenfassung	3
2	Semantik	3
2.1	Aufgabenstellung	3
2.2	Aufbau eines 3-D-Modells des Campus Koblenz	4
2.3	Semantische Annotation an das 3-D-Modell	6
2.4	Aufbau einer semantischen Bilddatenbank	7
2.5	Evaluationen mit Hilfe der semantischen Bilddatenbank	8
3	Elementare markante Merkmale	8
3.1	Detektion von Geraden	8
3.2	Verschmelzung räumlich naher und ähnlicher CSC-Segmente	9
3.3	Farb-Texturanalyse auf Fassadenkandidaten	10
3.4	Texturanalyse zur Detektion wilder Strukturen	10
3.5	Analyse von SIFT-Merkmalen	11
4	Höhere markante Merkmale	16
4.1	Gruppierungen auf SIFT-Merkmalen	16
4.2	Himmelsdetektion	20
4.3	Berechnung von Fluchtpunkten und Fluchtlinien	21
4.4	Blueprint-Rekonstruktion	24
5	Elementare markante Merkmale aus höheren markanten Merkmalen	25
5.1	Orte für Punktmerkmale aus semantischem Wissen	25
6	Matching	27
6.1	Allgemeines	27
6.2	Matching von Horizontlinien	27
6.3	Matching von SIFT-Merkmalgruppen	27
6.4	Matching markanter Geraden	28
6.5	Detektion von Fassaden	29
6.6	Erweiterungen von SIFT-Merkmalen	29
7	Modellgenerierung	30
7.1	Semiautomatische Modellierung	30
7.2	Automatische Erzeugung von CG-Modellen	31
7.3	3-D Laserscanner zur Rekonstruktion	35
7.4	Inpainting	35
8	Pose-Rückrechnung	35
8.1	Relative Pose	36
8.2	Absolute Pose	37
8.3	Absolute Pose durch Lineare Kalibrierung	37
8.4	Alternative Verfahren zur Bestimmung der absoluten Pose aus Punktkorrespondenzen	39
8.5	Vergleich der Rauschsensitivität der Verfahren	40
8.6	Absolute Pose aus Linien	41

9 Matching	41
9.1 Wissensbas. Graphmatching von Segmentierungsergebnissen	41
10 Laufende Qualifikationsarbeiten	42
10.1 Pose-Schätzung	43
11 Gemeinsame Arbeiten	43
12 Modell Schemata - Anforderungen an Sprache, Modell, Werkzeuge	43
13 Automatisierung der Modellgenerierung durch Matching markanter Eckpunkte	45
14 Glossar der Bildverarbeitung	46
A Technische Umsetzungsdetails der automatischen Erzeugung von CG-Modellen	46
B Rauschverhalten der untersuchten Verfahren zur Posebestimmung	47

1 Zusammenfassung

Die bildbasierte automatische Bestimmung der Pose, d. h. der Position und Blickrichtung, einer Kamera in der Welt, ist eine relevante, aber immer noch unzureichend gelöste Aufgabe im Rechnersehen.

In dem diesem Bericht zugrunde liegenden Projekt werden höhere markante Merkmale in den Bildern sicherer lokalisiert, sowie deren semantische Signifikanz vom Rechner bestimmt. Eine Posebestimmung wird durch eine Registrierung der elementaren Bestandteile dieser markanten Merkmale im Bild mit Merkmalen im 3-D-Modell erreicht.

Dazu werden neue Algorithmen zur Detektion, Lokalisation und Registrierung der markanten Merkmale entwickelt bzw. vorhandene Algorithmen weiter verbessert. Modelle, wie sie aus der Rekonstruktion von Mehrfachansichten entstehen, werden durch weitere Semantik angereichert.

Als Anwendungsszenario wird die Posebestimmung auf dem Campusgelände unter Verwendung von Bildern und einem semantischen CG-Modell des Campus gewählt. Die allgemeinen Verfahren werden an diesem Beispiel getestet und ihre Tragfähigkeit wird in Experimenten belegt.

Im modularen System entstehen problemunabhängige Einzelbausteine zur Detektion markanter Merkmale und zur 3-D-Rekonstruktion und Posebestimmung aus Merkmalen, Punkten und Linien. Damit wird belegt, dass eine robuste Detektion markanter Merkmale möglich ist und zu einer effektiven Rekonstruktion und Posebestimmung auch in teilweise wenig strukturierten Außengebieten genutzt werden kann.

2 Semantik

2.1 Aufgabenstellung

Für eine merkmalsbasierte Pose-Schätzung ist ein möglichst realistisches 3-D-Modell der Umgebung notwendig. Da unser Forschungsansatz das Matchen markanter Merkmale zwischen Modell und Kamerabildern der Umgebung beinhaltet, muss ein Ansatz gewählt werden, der es erlaubt, dass an das gewählte Modell generell Semantik leicht bzw. automatisch annotierbar ist. Wir sprechen dann von semantischen CG-Modellen.

Die erste Aufgabenstellung umfasste die Erstellung eines graphischen CG-Modells des Campus Koblenz in einer geeigneten Modellierungsumgebung. Hierzu wurden alle Gebäude des Campus Koblenz manuell detailgetreu mittels des Modellierungswerkzeugs Google Sketchup modelliert und in ein Geländemodell des Campus eingepasst, siehe Abschnitt 2.2.

Basierend auf diesem Modell wurden Verfahren entwickelt, um das graphische CG-Modell mit Hilfe semantischer Annotationen zu einem semantischen CG-Modell des Campus Koblenz zu erweitern. Dazu wurde auf das Komponentenkonzept von Google Sketchup zurückgegriffen. Die verwendeten Verfahren werden detailliert im Abschnitt 2.3 vorgestellt.

Ein „semantisches Rendering“ des so um Semantik erweiterten Modells ist möglich. Sowohl die Grafik des Modells als auch seine Semantik können in Echtzeit (momentan etwa 200 Frames pro Sekunde) gerendert werden.

Parallel zum Aufbau des 3-D-Modells wurde eine Datenbank von semantisch annotierten Bildern des Campus aufgebaut. Die Datenbank umfasst etwa 300 Bilder die mit verschiedenen Kameramodellen zu unterschiedlichen Tageszeiten und Wetterbedingungen aufgenommen wurden. Es wurden

verschiedene Werkzeuge zum Aufbau und zur Nutzung der Datenbank entwickelt, die im Abschnitt 2.4 vorgestellt werden.

2.2 Aufbau eines 3-D-Modells des Campus Koblenz

Für die Erstellung des Campusmodells wurde die Modellierungssoftware Google Sketchup gewählt. Die Gründe für diese Wahl waren hauptsächlich die unkomplizierte und schnelle Bedienung der Software sowie die Möglichkeit eines Exports in die meisten gebräuchlichen 3-D-Formate. Dadurch wurde es möglich, dass eine Gruppe von 10 Studierenden ohne vorherige Erfahrung mit der Software im Rahmen eines Praktikums mit 6 Wochenstunden in wenigen Wochen den gesamten Campus Koblenz modellieren konnte.

Es wurden alle Gebäude des Campus basierend auf Bauplänen, Photoaufnahmen und manuellen Ausmessungen modelliert. Dabei wurde sich zunächst auf die Modellierung der Außenflächen der Gebäude beschränkt. Bei der Modellierung wurde sehr detailliert vorgegangen. Objekte wie Fensterbänke, Führungsschienen von Sonnenrollos oder Fensterrahmen wurden nicht flach als Textur auf einer Fassade gezeichnet sondern als eigenständige geometrische Objekte modelliert. Glasscheiben wurden teilweise transparent und bei größeren Glasflächen auch einschließlich der dahinterliegenden Innenarchitektur modelliert.

Nach Abschluss der Modellierung wurde die geometrische und optische Korrektheit der Modelle aller Gebäude durch Überlagerung von Modellrenderings und Photos der Gebäude überprüft, um Abweichungen der Realität von den dem Modell zugrunde liegenden Bauplänen zu erkennen und beheben.

Ebenfalls wurde von uns auf Basis von Laser-Höhendaten des Landesamts für Vermessung und Geobasisinformation LVerGeo Rheinland-Pfalz in Koblenz ein digitales Höhenmodell des Campus generiert. Die Höhendaten lagen dabei zunächst als 3-D-Punktwolke vor. Diese wurden auf Punkte innerhalb des Campus reduziert und dann mittels der Software MeshLab¹ trianguliert. In das entstandene Drahtgitter-Geländemodell wurden die Gebäudemodelle eingesetzt und die Übergänge zwischen Geländemodell und Gebäudemodellen manuell angepasst. Im fertigen Modell ist die Zuordnung zwischen Modellkoordinaten und den Koordinaten der Realität in Gauß-Krüger-Koordinaten und Höhe über NN bekannt.

In einem letzten Schritt wurde das Gelände mittels Orthophotos (ebenfalls vom LVerGeo zur Verfügung gestellt) texturiert, markante Objekte wie Büsche und Bäume modelliert und ebenfalls dem Gesamtmodell hinzugefügt. Das fertige integrierte Modell ist unter <http://www.uni-koblenz.de/campusmodelle> für die Öffentlichkeit zugänglich und wurde in einem hochschulöffentlichen Vortrag vorgestellt.

Sketchup dient hauptsächlich zur 3-D-Modellierung und bietet wenig Möglichkeiten zum Rendern und Verarbeiten der fertigen Modelle. Hierzu kann das Modell in der freien 3-D-Grafiksoftware Blender² geladen werden. In Blender ist es möglich, das Modell mit modernsten Algorithmen der Computergrafik zu erweitern, manipulieren und rendern. Außerdem kann man automatisch eine vereinfachte Version des Modells generieren, die mit Hilfe der 3-D-Engine OGRE³ extrem schnell gerendert wird (momentan werden etwa 200 Bilder pro Sekunde erreicht).

¹<http://meshlab.sourceforge.net/>

²<http://www.blender.org/>

³<http://www.ogre3d.org/>



(a) Kamerabild



(b) Modell-Rendering

Abbildung 1: Bild-zu-Bild-Vergleich zwischen Kamerabild und Modell-Rendering

Als Alternativen zu Blender wurden verschiedene andere Rendering-Umgebungen getestet (unter anderem Unreal Engine, Irrlicht, g3d, ColladaDOMViewer, Kerkythea). Allerdings zeigte sich, dass Sketchups Exportfilter teilweise fehlerhafte Ergebnisse erzeugen, so dass beim Import und Rendern des Modells eine Software nötig ist, die robust ist und die Möglichkeit bietet, die Fehler automatisiert durch eine Script-Schnittstelle zu beheben. Auch hierzu besitzt Blender die notwendigen Werkzeuge.

Der sehr hohe Detaillierungsgrad des Modells lässt prinzipiell einen direkten Bild-zu-Bild-Vergleiche zwischen Renderings des Modells und Kamerabildern der Realität zu. Es wurden erste Arbeiten durchgeführt, das Modell mit einer in Position und Intensität der Sonne entsprechenden Lichtquelle zu beleuchten, um die dazu notwendigen realistische Schattenwürfe und Beleuchtung zu gewährleisten.

Im Rahmen der Öffentlichkeitsarbeit wurde eine vereinfachte Version des Campus-Modells erstellt, die für die Darstellung in der Software „Google Earth“ optimiert ist. Das Modell wurde von Google zur Aufnahme in Google Earth angenommen und ist somit bei der Betrachtung von Koblenz in Google Earth sichtbar.

2.3 Semantische Annotation an das 3-D-Modell

Basierend auf dem graphischen Modell wurde ein semantisches CG-Modell des Campus entwickelt. Dabei wurde ausgenutzt, dass das Sketchup-Modell komponentenbasiert aufgebaut ist.

Bei der komponentenbasierten Modellierung in Sketchup werden optisch und semantisch identische Objekte wie z.B. „Fenster Stockwerke 0-2 A-Gebäude Süden“ nur einmal als sogenannte Komponente modelliert. Im Modell werden dann Instanzen (z.B. „drittes Fenster von links im ersten Stock der Südfassade des Gebäudes A“) der Komponenten angelegt. Die Instanzen behalten dabei die Verbindung zur entsprechenden Komponente und übernehmen dynamisch alle Änderungen in dieser.

Im graphischen Modell sind somit sämtliche semantisch wichtigen Objekte wie Fenster, Türen und Fassaden über den Bezeichner der Komponente, der sie zugeordnet sind, ansprechbar. Die Bezeichner der Komponenten enthalten dabei alle Informationen, um die semantische Bedeutung des Objektes rekonstruieren zu können.

Ein solcher Bezeichner ist beispielsweise „ANwin002“, aus dem rekonstruiert werden kann, dass es sich um ein Fenster des Typs 002 in der Nordfassade des A-Gebäudes handelt. Generell ist in jedem Bezeichner eindeutig kodiert, um welche Objektgattung (Fenster, Türen, Fassaden etc.) und welchen Typ innerhalb der Gattung es sich bei dem Objekt handelt, zu welchem der Gebäude das Objekt gehört, und an welcher der Gebäudeseiten sich das Objekt befindet.

Statt den Bezeichnernamen direkt auszuwerten, kann der Name jedoch auch als Schlüssel zum Nachschlagen in einer Lookup-Tabelle dienen, die jedem Bezeichner prinzipiell beliebige Attribute zuweisen kann. Auch ein hierarchischer Aufbau ist hier trivial realisierbar, indem die Lookup-Tabelle z.B. für eine Fassade Verweise auf alle bekannten Objekte innerhalb der Fassade enthält.

In Blender können die Objekt-Bezeichner genutzt werden, um spezifische semantische Objekte anzusprechen. Dadurch wird zum Beispiel ein selektives Rendern nur semantisch bedeutsamer Objekte möglich.

Zudem kann jedes semantische Objekt in einer wohldefinierten Farbe eingefärbt werden. Dazu wird für alle Objekte in der Lookup-Tabelle die entsprechende eindeutige Farbe nachgeschlagen. Objekte im Modell ohne semantische Annotation werden entweder transparent oder grau gesetzt. Wird ein derart „semantisch“ eingefärbtes Modell gerendert (siehe Abbildung 2), kann anhand der im Ren-

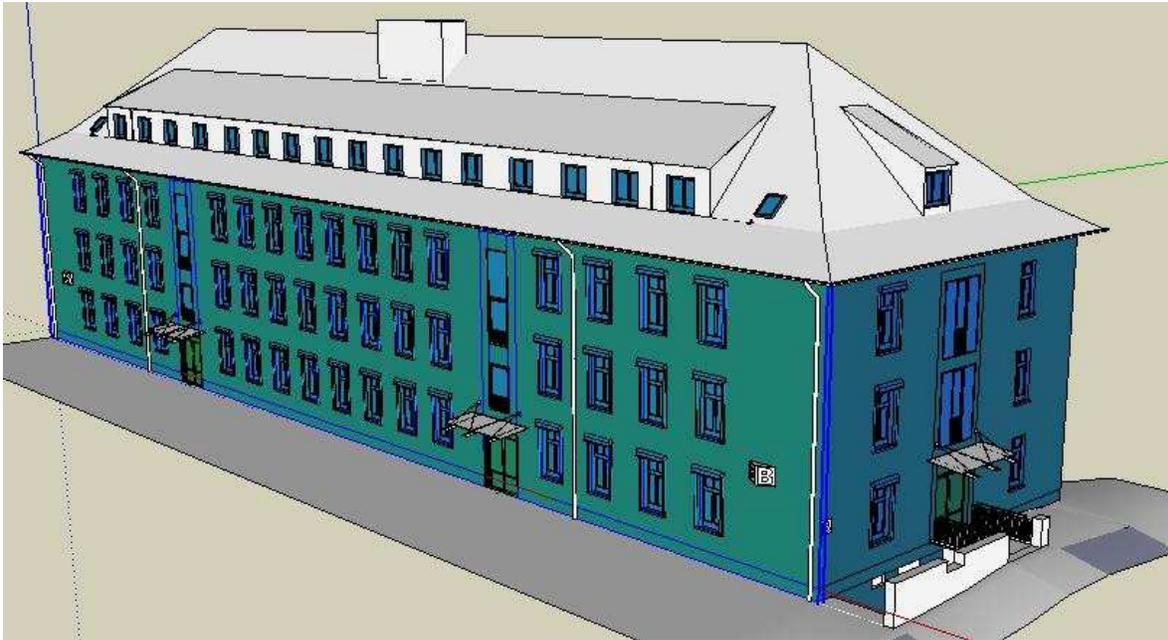


Abbildung 2: Semantisches Einfärben

dering vorkommenden Farben über eine Rückwärtssuche in der Lookup-Tabelle bestimmt werden, welche Objekte sichtbar sind. Diesen Vorgang bezeichnen wir als „semantisches Modell-Rendering“. Damit ist eine Überprüfung einer geschätzten Pose mittels der von dieser Pose tatsächlich sichtbaren semantischen Objekte aus dem Modell möglich.

2.4 Aufbau einer semantischen Bilddatenbank

Um Bilder komfortabel und genau semantischen annotieren zu können, wurde eine Werkzeugkette zur Bildannotation erstellt. Grundlage der Kette ist eine elementare Beschreibung der relevanten Objekttypen wie Fenster, Türen, Fassaden, Himmel etc. Diese Beschreibung liegt in Form einer Textdatei vor und wird dynamisch zur Laufzeit geladen.

In der Beschreibung wird jedem Objekttyp ein Deskriptor zugeordnet, der die Geometrie einer Instanz des Objekttyps im Bild beschreibt. Solche Deskriptoren können zum Beispiel Polygone oder Regionenbeschreibungen mit inneren und äußeren Konturen sein. Weiterhin kann jeder Objekttyp beliebige Attribute besitzen. Ein solches Attribut des Objekttyps “Fenster” wäre zum Beispiel der Grad der Verdeckung durch einen Sonnenschutz.

Zum Anfertigen der Annotationen wurde eine graphische Toolbox entwickelt, die über einen Touchscreen-Monitor bedient wird. Der Nutzer kann damit komfortabel und genau die Deskriptoren der zu annotierenden Objekte erstellen. Für Objekte, die über Konturen beschrieben werden, ist es zum Beispiel möglich, die Kontur frei Hand mittels eines Stiftes auf dem Touchdisplay zu zeichnen. Alternativ können Geradenstücke durch Klicken auf die Endpunkte gezeichnet werden oder es können die Konturen von Segmenten aus einer CSC-Segmentierung übernommen werden.

2.5 Evaluationen mit Hilfe der semantischen Bilddatenbank

Für alle Bilder wurde manuell die konkreten Positionen von Fassaden, Fenstern, Dächern und Türen annotiert, aber auch von indirekten Objekten wie Himmel und Fluchtpunkten. Für alle diese Objekte ist - soweit sinnvoll - annotiert, zu welchem Gebäude und zu welcher Gebäudeseite sie gehören. Außerdem wurden weitere Objektattribute annotiert. So wird zum Beispiel für Fenster annotiert, ob ein eventuell vorhandener Sonnenschutz herabgelassen ist. Über eine Korrespondenztabelle können die in den Bildern annotierten Objekte den Objekten im semantischen 3-D-Campusmodell zugeordnet werden. Die annotierten Semantiken im 3-D-Modell und der 2-D-Bilddatenbank dienen als Ground-Truth zur Evaluation aller entwickelten und zu entwickelnden Algorithmen.

Die Evaluation unserer entwickelten Bildverarbeitungsalgorithmen gegen die in der Datenbank gespeicherte Ground-Truth geschieht generell automatisch. In allen unseren Publikationen hat eine solche Evaluation stattgefunden. Um die Qualität einer Menge von berechneten Objekten S im Vergleich zu der Objektmenge in der Ground-Truth GT zu bewerten, berechnen wir die *Coverability Rate* (CR) und *Error Rate* (ER). Die Coverability Rate misst, wie viele Ground-Truth-Objekte aus GT von den Bildverarbeitungsalgorithmen korrekt gefunden werden. Die Error Rate misst, wie viele der berechneten Objekte in S nicht Teil der Ground-Truth GT sind. CR und ER sind dabei mathematisch definiert als:

$$CR(S, GT) := \frac{|S \cap GT|}{|GT|}, \quad ER(S, GT) := \frac{|S - GT|}{|S|}.$$

3 Elementare markante Merkmale

Es wurden Algorithmen entwickelt, um niederdimensionale Merkmale wie Geraden, Eckpunkte und Flächen sicher aus Bildern zu extrahieren und anschließend zu verarbeiten und zu analysieren.

3.1 Detektion von Geraden

Zur Detektion von Geraden im Bild ist die schon seit Jahrzehnten bekannte Hough-Transformation immer noch das robusteste Verfahren. Trotz intensiver Recherche konnten jedoch keine frei verfügbare Implementation der Hough-Transformation für Geraden gefunden werden, die unseren Qualitätsanforderungen genügen. Deshalb wurde nicht nur eine eigene Implementation entwickelt, sondern es wurden auch neue theoretische Überlegungen zur Verbesserung der Hough-Transformation getestet und einer Implementierung hinzugefügt.

Entscheidend für die Qualität der Geradenextraktion ist die Analyse des Akkumulatorraums. Hier wird zum einen auf so genannte Schmetterlingsfilter [LB87] und zum anderen auf neu entwickelte eigene Verfahren zur Non-Maxima-Unterdrückung zurück gegriffen. Ferner werden in einem neuen Vorverarbeitungsschritt sogenannte "wilde" Strukturen ausgeblendet, siehe den Absatz 3.4 zu Wildheitskarten. Der aktuelle Stand der Implementation liefert sehr robust die markanten Geraden im Bild, was für die weiteren Verarbeitungsschritte (siehe Abschnitte 4.3, 4.4, 5.1) von essentieller Wichtigkeit ist.

Problematisch ist momentan noch, dass Geraden, die aus wenigen, relativ kurzen Teilstücken bestehen, entsprechend der Natur der Hough-Transformation und des Schmetterlings-Filters nicht detek-

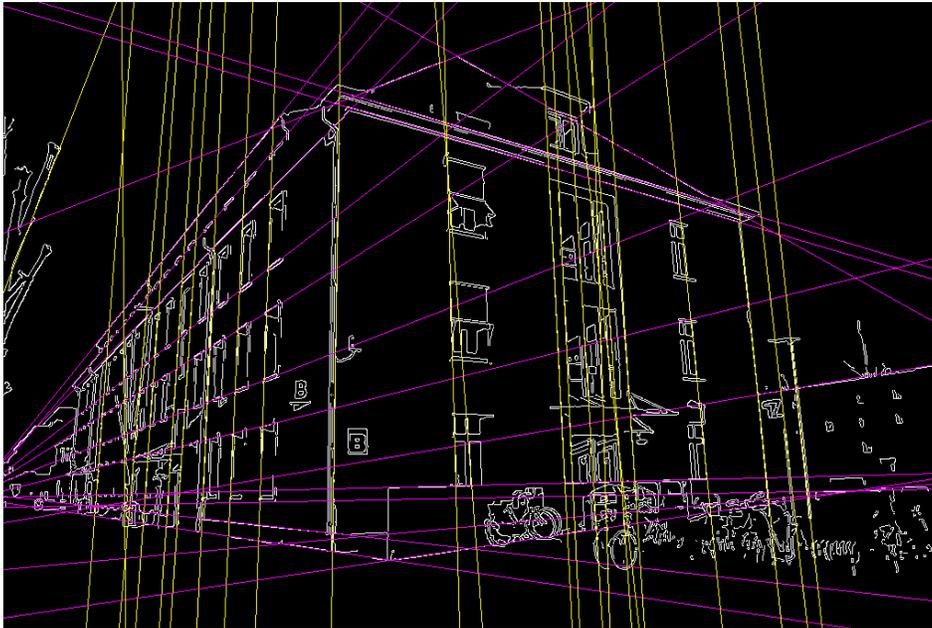


Abbildung 3: Geradenextraktion

tiert werden. So fehlen oft die Geraden an den Ober- und Unterkanten der Fenster in Stirnseiten von Gebäuden (vgl. Abbildung 3.), da diese Geradenstücke im Bild relativ kurz sind. Dieses Problem soll in Zukunft über höhere Merkmale gelöst werden. Dazu wurden bereits einige Techniken untersucht, höhere Merkmale wie Fenster unabhängig von Geraden zu detektieren (über eine Analyse von Löchern in CSC-Segmenten oder Mustererkennung). Aus Verläufen von iterierten Fenster können fehlende Geraden nachträglich berechnet und weiteren Analysetechniken (etwa für Fluchtpunkte) hinzugefügt werden.

3.2 Verschmelzung räumlich naher und ähnlicher CSC-Segmente

Neben Geraden sind auch farblich homogene Flächen charakteristische Merkmale in Aufnahmen bebauter Umgebungen. Solche Flächen werden typischerweise durch Segmentierungsalgorithmen gewonnen. Der in der Arbeitsgruppe von Professor Priese entwickelte CSC (Color Structure Code) ist ein sehr robuster und schneller Segmentierer für Farb- oder Grauwertbilder mit ausgezeichneten Ergebnissen.

Ein Segment ist per Definition eine Menge räumlich zusammenhängender, farbähnlicher Pixel. Somit sind auch alle durch den CSC generierten Segmente per Definition farblich homogen und räumlich zusammenhängend.

Semantisch zusammengehörende Strukturen wie Fassaden eines Gebäudes sind jedoch nicht immer räumlich zusammenhängend, sondern zerfallen, auch wenn sie farblich homogen sind, oft in mehrere Teilregionen. Grund dafür sind in der Regel Drittobjekte wie Fallrohre oder Laternenmäste, die den Zusammenhang zerstören.

Als ersten Schritt, um Fassaden und andere großflächige, semantisch zusammengehörende Strukturen automatisch zu einem Objekt zusammenzufassen, wurden deshalb Algorithmen entwickelt, um

in Teilregionen zerfallene Strukturen wieder zusammen zu fügen. Das Verfahren sucht an Hand von Segmentgrößen und -farben Kandidatensegmente für eine Verschmelzung und fasst diese zu Makrosegmenten zusammen, wenn anhand ihrer Farbe, Form und Nachbarschaft zueinander von einer Zusammengehörigkeit ausgegangen werden kann.

Dieser erste Ansatz muss weiter ausgebaut werden. Hier können die Informationen aus der Fluchtpunktanalyse (siehe Abschnitt 4.3) und Texturanalyse (siehe Abschnitt 3.3) genutzt werden.

3.3 Farb-Texturanalyse auf Fassadenkandidaten

Es wurde untersucht, ob es über eine Analyse von Texturmerkmalen möglich ist, Gebäudefassaden zu identifizieren. Hierzu wurden in Photos aus der Bilddatenbank (vgl. Abschnitt 2.4) 249 Bildregionen manuell extrahiert, die die Fassaden von sechs unterschiedlichen Gebäudetypen zeigen. Jede Region wurde manuell dem passenden Gebäudetyp zugewiesen.

Die gebräuchlichsten Verfahren zur Texturanalyse arbeiten nur auf Grauwertbildern. Im Anwendungsfall der Fassaden-Identifizierung ist jedoch Farbe ein sehr charakteristisches Merkmal. Aus diesem Grund wurde das Farbkovarianz-Texturmodell [5], [6], ein statistisches Texturmodell zweiter Ordnung, verwendet.

Im Gegensatz zu den üblichen Texturmodellen für Grauwertbilder werden in diesem Texturmodell sowohl die räumlichen als auch die farblichen Relationen von Bildpunkten in lokalen Nachbarschaften statistisch analysiert.

Für jede extrahierte Region wird entsprechend des Texturmodells eine Farbkovarianz-Matrix berechnet auf Basis derer 14 Texturmerkmale bestimmt werden, die an die bekannten Haralick-Merkmale [Har79] angelehnt sind.

Es ergibt sich somit eine Merkmalsdatenbank mit 249 Einträgen. Zur Evaluation der Güte des Texturmodells für die Fassadenidentifikation wurde jedes Objekt in dieser Datenbank gegen eine Trainingsmenge bestehend aus den anderen 248 Einträgen mittels einer k-NN-Klassifikation klassifiziert. Es ergab sich eine Rate von 75% richtig klassifizierten Merkmalen, 18% Merkmalen in der Rückweisklasse und 7% falsch klassifizierten Merkmalen.

3.4 Texturanalyse zur Detektion wilder Strukturen

In vielen Algorithmen ist es hilfreich Strukturen auszublenden, die stark und unregelmäßig strukturiert sind. Wir nennen solche Strukturen „wild“.

Wilde Strukturen treten typischerweise in Baumkronen, Ästen und Büschen oder durch Moiré-Effekte auf. In kantenbasierten Algorithmen bewirken sie viele starke Gradienten, die jedoch durch die fehlende Regelmäßigkeit zur weiteren Analyse nicht hilfreich sind. Oft bewirken sie sogar Fehler in nachfolgenden Verarbeitungsschritten.

Bei der Canny-Kantendetektion können solche überflüssige starke Gradienten die Berechnung der Schwellwerte für die Kantenverfolgung sogar negativ beeinflussen, da die absoluten Schwellwerte in der Regel nicht direkt festgelegt sind, sondern auf Basis der Verteilung der Gradientenstärken im aktuellen Bild berechnet werden.

Bei der Hough-Transformation zur Detektion von Geraden im Bild können unerwünschte, starke Kanten dazu führen, dass Geraden im Bild detektiert werden, die keiner bedeutsamen Objektkante

entsprechen. Diese falschen Kanten wiederum erschweren höhere Algorithmen, wie z.B. die Fluchtpunktberechnung (Abs. 4.3).

Um wilde Strukturen in einem Bild I zu detektieren, wird ein Filter angewendet, der die Wildheit an einem Bildpunkt $p \in Loc_I$ über zwei Werte charakterisiert, die in einem Fenster w_p der Größe 11×11 um p berechnet werden:

- die Standardabweichung σ aller Werte in w_p ,
- die Summe der Differenzen in den Werten an einer Position $p' \in w_p$ und der vier direkten Nachbarpositionen wird über alle $p' \in w_p$ gemittelt und ergibt $\mu_D(w_p)$.

$c_1 \cdot \mu_D(w_p) - c_2 \cdot \sigma$ ist der Grad der Wildheit an der Position p , mit Konstanten $c_1, c_2 \in \mathbb{R}$, die sicherstellen, dass beide Werte im selben Zahlenbereich liegen. Um zu einem binären Prädikat Wild/Unwild zu gelangen, wird für das gesamte Bild nach dem oben beschriebenen Verfahren der Grad der Wildheit gemessen und mittels des Verfahrens von Otsu [Ots79] ein absoluter Schwellwert bestimmt.

Abbildung 5 demonstriert den Effekt des Entferns wilder Strukturen auf das Ergebnis des Canny-Kantendetektors, Abbildung 6 zeigt auf dem selben Bild die Auswirkungen auf die Geradendetektion mittels der Hough-Transformation.

3.5 Analyse von SIFT-Merkmalen

Motivation

SIFT (scale-invariant feature transform [Low04]) ist ein Algorithmus zur Lokalisation markanter Positionen in Grauwertbildern und zur skalierungs- und rotationsinvarianten Beschreibung der Umgebungen dieser Bildpositionen. Ein SIFT-Merkmal f ist ein Tupel $f = (l_f, s_f, o_f, v_f)$ bestehend aus der Bildposition l_f , einer Hauptorientierung o_f , einer Skalierung s_f , die die Größe der betrachteten Umgebung angibt, sowie aus einem 128-dimensionalen Beschreibungsvektor v_f , der die Gradientenrichtungen auf verschiedenen Skalierungsstufen um die betrachtete Bildposition beschreibt.

Es ist bekannt, dass sich SIFT-Merkmale sehr gut zur Punktkorrespondenz zwischen räumlich oder zeitlich versetzt aufgenommenen Bildern eignen. Dabei werden in der Regel aber nur solche SIFT-Merkmale zur Korrespondenzbestimmung verwendet, die innerhalb eines Bildes gewisse Alleinstellungsmerkmale besitzen.

Betrachtet man jedoch innerhalb von Bildern in urbanen Umgebungen alle SIFT-Merkmale, und nicht nur markante mit Alleinstellungsmerkmal, so stellt man fest, dass häufig SIFT-Merkmale an sich wiederholenden Strukturen im Bild gefunden werden (wie etwa in den Fensterkreuzen in Abbildung 7). Daher ergab sich folgende Fragestellung:

- Sind die SIFT-Merkmale an sich wiederholenden semantisch ähnlichen Bildpositionen ähnlich hinsichtlich des SIFT-Beschreibungsvektors?
- Wie können die semantisch ähnlichen Bildpositionen mit Hilfe der SIFT-Merkmale automatisch gruppiert werden?
- Wie können zusammengehörende Gruppen erkannt und automatisch zusammen gefasst werden?

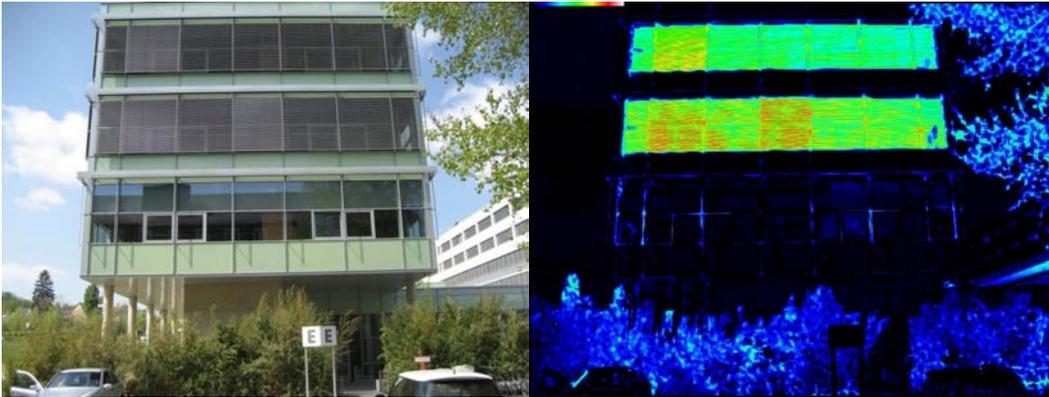


Abbildung 4: Originalbild und Visualisierung Wildheit als Heatmap von schwarz nach weiß

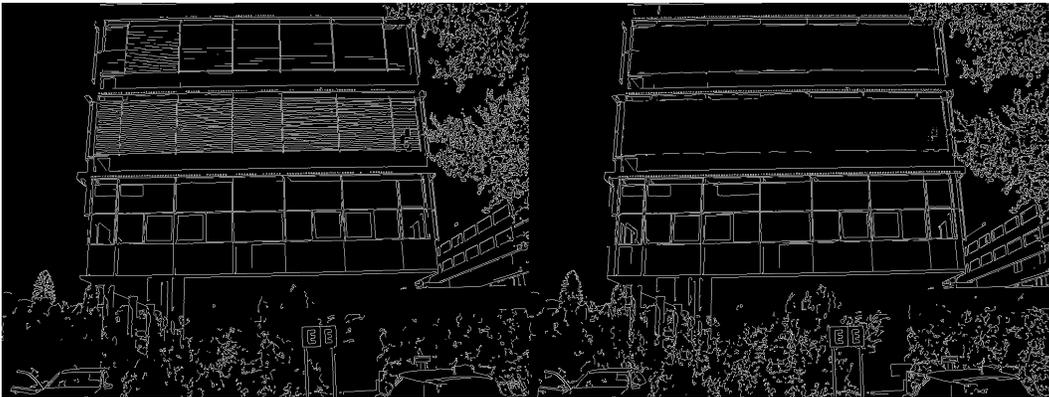


Abbildung 5: Ergebnis des Canny-Kantendetektors vor (links) und nach (rechts) Entfernung wilder Strukturen

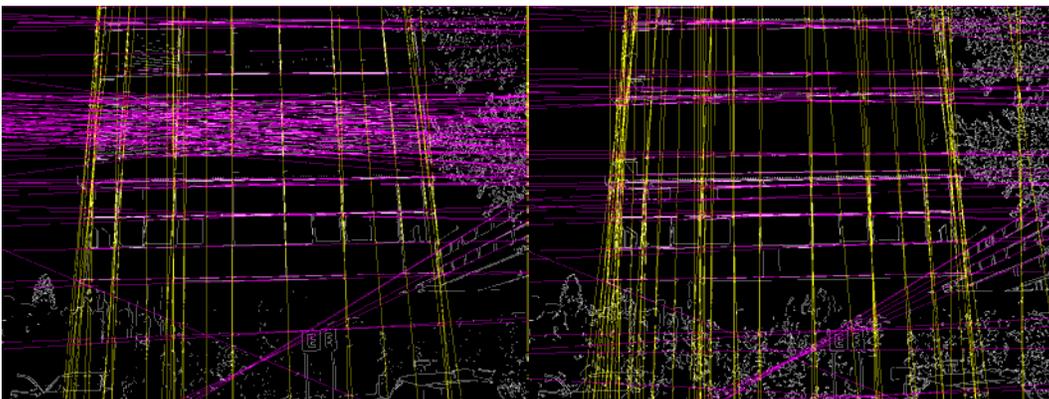


Abbildung 6: Ergebnis der Hough-Transformation vor (links) und nach (rechts) Entfernung wilder Strukturen



Abbildung 7: Markante Bildpositionen aus SIFT

Nachbarschaften auf SIFT-Merkmalen

In einer Studienarbeit von Nils Hering [3] wurde innerhalb von Bildern in urbanen Umgebungen untersucht, wie SIFT-Merkmale gefunden werden können, die zu einem Referenzmerkmal im selben Bild semantisch und optisch ähnlich sind. Die Ergebnisse dieser Studienarbeit wurden zu einem Artikel zusammengefasst und auf der Konferenz „Fourth International Conference on Computer Vision Theory and Applications (VISAPP) 2009“ vorgestellt und in deren Proceedings publiziert [4].

Merkmale an semantisch und optisch ähnlichen Bildpositionen besitzen zwar Beschreibungsvektoren mit einem geringem euklidischem Abstand, dennoch bilden die zu einem Referenzmerkmal f bzgl. Euklid nahen SIFT-Merkmale nicht die Nachbarschaft der zu f semantisch ähnlichen Merkmale. Der Grund dafür ist, dass es häufig semantisch unähnliche Merkmale gibt, deren euklidischer Abstand zum Referenzmerkmal kleiner ist als der Abstand zwischen semantisch ähnlichen Merkmalen und dem Referenzmerkmal, siehe das Beispiel in Abbildung 8.

Wir konnten einen Algorithmus für die automatische Erzeugung einer semantischer Nachbarschaft $N(f)$ zu jedem gegebenen Referenzmerkmal f entwickeln. Dieser Algorithmus berücksichtigt neben der euklidischen Distanz eine weitere sogenannte *7-Distanz* und die Skalierung und Orientierung der Merkmale. So sind die Merkmalsvektoren der beiden SIFT-Merkmale in Abbildung 9 sehr ähnlich, jedoch ist die Orientierung der Merkmale derart unterschiedlich, dass darüber eine unterschiedliche Semantik erkannt wird. Die in SIFT gewollte Orientierungsinvarianz zum Matchen in unterschiedlichen Aufnahmen ist für die semantische Ähnlichkeit in einem Bild eher hinderlich. Ähnliches gilt für die Skalierungsinvarianz, jedoch in einem anderen Rahmen, da gleiche Objekte in unterschiedlicher

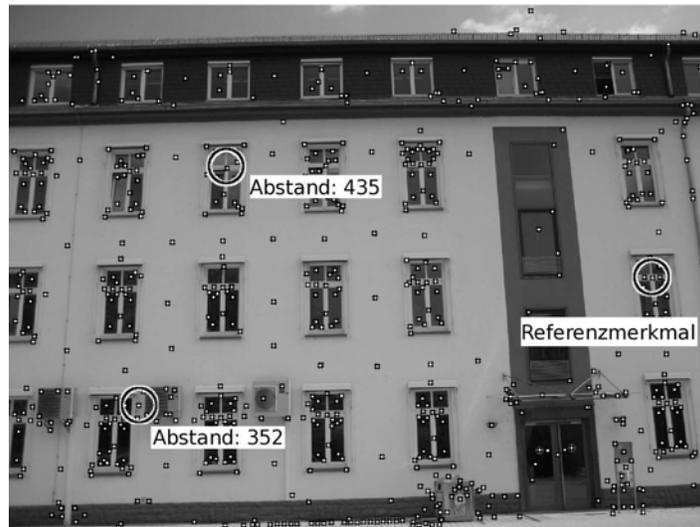


Abbildung 8: Euklidische Abstände zwischen einem Referenzmerkmal und einem semantisch ähnlichem bzw. semantisch unähnlichem Merkmal.



Abbildung 9: Zwei Merkmale mit unterschiedlicher Semantik aber ähnlichem 128-dimensionalen Merkmalsvektor

Entfernung zur Kamera eine gleiche Semantik besitzen, aber eine unterschiedliche Skalierung in ihren SIFT-Merkmalen aufweisen.

Eine Angabe etwa absoluter Schwellwerte für die Unterschiede in Orientierung und Skalierung ist daher nicht ausreichend. Es müssen kontinuierlich ansteigenden Unterschiede erlaubt werden.

Die automatisch definierten Nachbarschaften auf SIFT-Merkmalen bilden eine naive Topologie \mathcal{T} , bei der der Ort der Merkmale keine Rolle spielt sondern nur deren semantische Bedeutung. Dies ist allerdings keine Topologie im formalen Sinn der Mathematik.

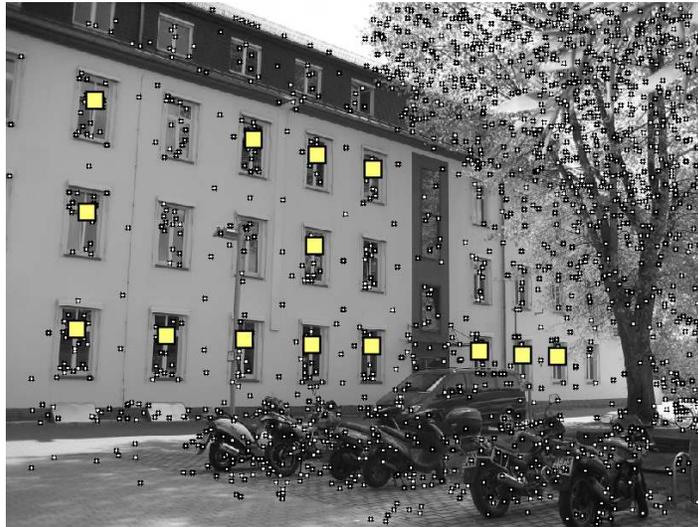


Abbildung 10: Gruppierung in einem Bild mit starker perspektivischer Verzerrung

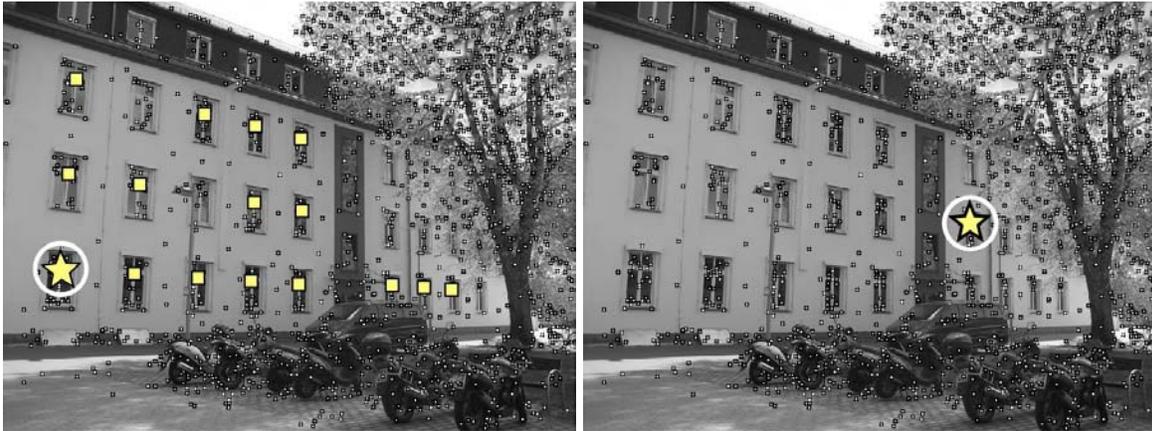


Abbildung 11: Eine gute (16 Mitglieder) und eine sehr schlechte (nur ein Mitglied) Nachbarschaft basierend auf unterschiedlichen Referenzmerkmalen (mit Stern markiert)

Das Verfahren wurde auf verschiedenen Bildern aus der Testdatenbank angewandt. Dabei zeigte sich, dass die Nachbarschaft eines SIFT-Merkmals f nur sehr selten Merkmale enthält die nicht semantisch ähnlich zu f sind. Allerdings umfasst die Nachbarschaft eines Merkmals f oft nicht alle zu f semantisch ähnlichen Merkmale. Die Größe der Nachbarschaft ist dabei stark abhängig davon welches Merkmal als Referenz gewählt wurde. Dies ist in Abbildung 11 demonstriert wo abhängig vom gewählten Referenzmerkmal einmal 16 Merkmale der Semantik Fensterkreuz zu einer Nachbarschaft zusammengefügt wurden, während für ein anderes Referenzmerkmal der selben semantischen Bedeutung keine semantisch ähnlichen Merkmale gefunden wurden.

Varianten der SIFT-Merkmale

SIFT-Merkmale normieren die Beschreibungsvektoren auf die Hauptorientierung des Merkmals, um Rotationsinvarianz zu erreichen. In unserem Anwendungsfall ist dies jedoch eher störend, da wir nicht Merkmale aus verschiedenen Bildern mit potentiell unterschiedlicher Kameraausrichtung sondern Merkmale innerhalb eines Bildes vergleichen. Für unseren Anwendungsfall erwarten wir, dass alle Merkmale, die zusammen gruppiert werden sollen, die gleiche Ausrichtung zeigen. Deshalb haben wir eine Variante von SIFT-Merkmalen definiert, in der die Normierung des Beschreibungsvektors auf die Hauptorientierung des Merkmals nicht vorgenommen wird. Wir nennen diese Variante $SIFT_{noi}$ wobei das *NOI* für **n**o **o**rientation **i**nvariance steht. Ein Tool zu Erzeugung der $SIFT_{noi}$ -Merkmale wurde implementiert.

4 Höhere markante Merkmale

Es wurde untersucht, welche höheren Merkmale charakteristische Informationen enthalten, über die ein Matching von Bildern im gewählten Kontext möglich erscheint.

Basierend auf den im Kapitel 3 vorgestellten Methoden zur Gewinnung, Verarbeitung und Analyse elementarer Merkmale wurden verschiedene Verfahren entwickelt, um höhere Merkmale aus elementaren Merkmalen zu gewinnen und sie anschließend zu analysieren.

4.1 Gruppierungen auf SIFT-Merkmalen

Gruppieren von Nachbarschaften auf SIFT-Merkmalen zu Merkmalsgruppen gleicher Semantik

Basierend auf der Nachbarschaft semantisch ähnlicher SIFT-Merkmale (vgl. Absatz 3.5) wurde ein Algorithmus zur Gruppierung von semantisch ähnlichen SIFT-Merkmalen entwickelt. Die Methode kommt dabei ohne jedes Vorwissen über die Domäne des Bildes aus und arbeitet komplett automatisch. Ein Artikel, in dem das Verfahren vorgestellt wird, wurde zur Konferenz „16th Scandinavian Conference on Image Analysis SCIA 2009“ in Oslo angenommen [10].

Der Algorithmus basiert darauf, überlappende Nachbarschaften von SIFT-Merkmalen iterativ zu Gruppen zu verschmelzen. Dabei werden Ideen und Techniken des CSC [11] verwendet. Der Algorithmus zum Verschmelzen der Nachbarschaften wurde von uns *AGS* für „Automatic Grouping of Semantics“ genannt und war zunächst speziell für den genannten Anwendungszweck entwickelt worden. Es stellte sich jedoch heraus, dass er generell ein hilfreiches Werkzeug zum Clustern in nicht topologischen Räumen - also auch ohne eine Metrik - ist, in denen ein Ersatz für „Nähe“ und „Zusammenhang“ über Nachbarschaften definierbar ist. Der *AGS* wurde von uns zur Verarbeitung von SIFT-Merkmalen in mehreren Kontexten eingesetzt, außerdem verwenden wir ihn in unserem Algorithmus zur Berechnung von Fluchtpunkten, siehe Abschnitt 4.3.

Im Prinzip arbeitet *AGS* wie folgt: Um die Gruppen von semantisch ähnlichen SIFT-Merkmalen zu generieren, wird zunächst eine initiale Gruppierung erzeugt. Nun werden iterativ bereits erzeugte *ähnliche* Gruppen G, G' zu einer neuen Gruppe *verschmolzen*. Hier sind „Ähnlichkeit“ und „Verschmelzen“ zwei Freiheitsgrade. Verschmelzen könnte etwa die Bildung von $G \cup G'$ bedeuten, aber auch von

$N(G \cap G')$ oder eine andere Region-Growing Technik innerhalb von $G \cup G'$ aus $G \cap G'$ heraus. Wir stellen hier zwei Ähnlichkeitsmaße für semantische Gruppierungen auf verschiedenen Ebenen vor.

Mikrosemantik

Bei dieser ersten semantischen Gruppierung wird zunächst SIFT auf das Bild angewendet. Anschließend wird für jedes SIFT-Merkmal f die Nachbarschaft $N(f)$ berechnet. Die Menge aller Nachbarschaften bildet eine initiale Gruppierung für den nachfolgenden AGS. Dabei wählen wir als Maß für die Ähnlichkeit von G und G' deren Überlappungsrate

$$o(G, G') := \frac{|G \cap G'|}{\min(|G|, |G'|)}.$$

Ähnlichkeit zwischen G und G' liegt vor bei $o(G, G') > t_o$ für einen festen Schwellwert t_o . Verschmelzen ist hier einfach der Übergang zu $G \cup G'$.

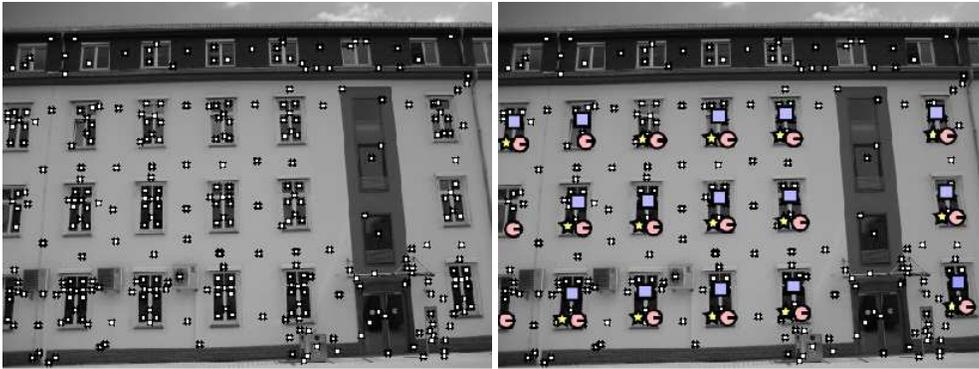


Abbildung 12: Anzahl Gruppen = 10; davon gezeigt: Fensterkreuze, untere rechte Scheiben, untere linke Scheiben

Mit dieser Variante des AGS entstehen Gruppen von Merkmalen, die auf einer niedrigen Ebene die gleiche Semantik haben. Derartige Semantiken sind etwa “Fensterkreuz”, “linke untere Scheibe” oder “Regenrinne”. Wir sprechen hier von *Mikrosemantik*.

Abbildung 12 zeigt ein Ergebnis des Algorithmus auf einem Beispielbild. In dem Bild wurden insgesamt 10 Gruppen von SIFT-Merkmalen generiert von denen drei Gruppen dargestellt sind, und zwar Fensterkreuze, untere rechte Scheiben, untere linke Scheiben.

Eine Evaluation auf Bildern mit einer handannotierten Ground-Truth zeigt, dass mit diesem einfachen Verfahren bereits gute Ergebnisse möglich sind. Dabei konnte bei den drei Klassen Fensterkreuze, untere rechte Scheiben, untere linke Scheiben im Mittel eine Coverability Rate von 86% bei einer Error Rate von lediglich 5% erreicht werden.

Durch Verwendung der SIFT_{noi}-Variante konnten bei der Bildung von Merkmalsgruppen gleicher Mikrosemantik leicht bessere Ergebnisse erreicht werden. Im Mittel stieg die Coverability Rate von 0.86% auf 89% und die Error Rate sank von 5% auf 4%.

Weiterhin wurde der Algorithmus auf Merkmale aus dem SURF-Algorithmus [BTVG06] getestet. SURF-Merkmale sind sehr ähnlich zu SIFT-Merkmalen, wie diese sind sie ein Tupel $f =$

(l_f, s_f, o_f, v_f) , allerdings mit einem anderen, jetzt 64-dimensionalen Beschreibungsvektor. Wegen der Ähnlichkeit der Merkmale konnten die für SIFT entwickelten Algorithmen unverändert übernommen werden. Die Ergebnisse waren visuell vergleichbar, eine quantitativ vergleichende Evaluation wurde allerdings noch nicht vorgenommen.

Makrosemantik

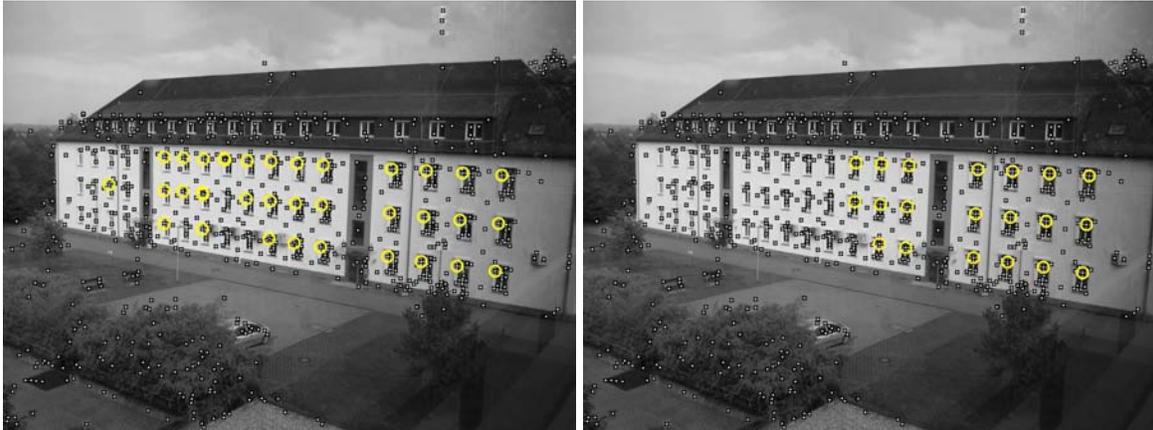


Abbildung 13: Zwei räumlich ähnliche Merkmalsgruppen

Bei den bisher vorgestellten Verfahren zur Analyse von SIFT-Merkmalen haben die Positionen der Merkmale keinerlei Rolle gespielt. Der im folgenden vorgestellte Algorithmus hingegen erlaubt das Verschmelzen räumlich ähnlicher Merkmalsgruppen, sogar falls diese Merkmalsgruppen anhand unterschiedlicher Merkmalsklassen gefunden wurden. Eine Publikation dieses Verfahrens ist aktuell in Arbeit (Arbeitstitel: Semantikclustering in spaces without an topology, Autoren: Lutz Priebe, Frank Schmitt, Nils Hering). Hier sind wir in der Evaluationsphase.

Die Idee ist, Ähnlichkeit zweier Gruppen G und G' (initial aus berechneten unterschiedlichen Mikrosemantiken) über deren räumlicher Translation zu definieren. In Abbildung 13 sind zwei solcher räumlich ähnlicher Merkmalsgruppen dargestellt. Die Gruppe im linken Bild hat die semantische Bedeutung „linke obere Scheibe“ während die Gruppe im rechten Bild die Bedeutung „Fensterkreuz“ hat. Beide Gruppen sind offensichtlich räumlich ähnlich. Diese räumliche Ähnlichkeit wurde formal definiert und als abfragbares Prädikat implementiert.

Ausgangspunkte des Custeralgorithmus AGS sind nun nicht Merkmale sondern bereits gebildete Gruppen von Merkmalen gleicher Mikrosemantik. Die räumliche Nachbarschaft $N(G)$ einer Merkmalsgruppe G ist die Menge aller zu G räumlich ähnlichen Merkmalsgruppen. Die Menge der räumlichen Nachbarschaften aller berechneten Merkmalsgruppen dient nun als initiale Gruppierung für einen zweiten Lauf des AGS.

Das Ergebnis ist eine semantische Gruppierung auf einer höheren Abstraktionsebene. Wurden zuvor Merkmalsgruppen mit Mikrosemantiken wie untere linke Scheibe, untere rechte Scheibe, obere linke Scheibe, obere rechte Scheibe und Fensterkreuz gebildet, so werden diese nun bei räumlicher Ähnlichkeit zu einer Makrogruppe mit einer „höheren“ Semantik verschmolzen, in diesem Beispiel mit der Semantik „Fenster“. Diese aggregierten Semantiken nennen wir *Makrosemantik*. Abbildung 14 zeigt ein Ergebnisbeispiel dieses Algorithmus.



Abbildung 14: Makrogruppe „Fenster“

Clustering innerhalb von Makrogruppen

Im Beispielbild 14 ist ein semantisches Objekt „Fenster“ durch ein Cluster von Orten in der Makrogruppe „Fenster“ gegeben, und nicht durch ein einzelnes Objekt. Dieser Nachteil lässt sich durch eine weitere Clusteranalyse beheben. Auch dazu kann wiederum der AGS-Algorithmus verwendet werden. Es sei L die Menge der Orte aller Merkmale in der aktuell untersuchten Makrogruppe. Die Nachbarschaft $N(p)$ eines Ortes $p \in L$ ist einfach die Menge aller weiteren Orte in L innerhalb einer gewissen euklidischen Distanz d . Die Menge $\mathcal{N} := \{N_d(p) \mid p \in L\}$ ist die initiale Gruppierung für den AGS. Das Ähnlichkeitsmaß ist wieder die Überlappungsrate. Die durch AGS gefundenen Gruppen sind die gewünschten Cluster in L . Hier steuert d die Größe der Cluster und muss auf die Größe der semantischen Objekte im Bild angepasst werden. Dies ist der erste und einzige Parameter des gesamten Verfahrens der mit Hilfe von Domänenwissen angepasst werden muss.

Abbildung 15 zeigt das Ergebnis des beschriebenen Clusterings angewandt auf die Makrogruppe „Fenster“ aus Abbildung 14. Dabei sind die Mittelpunkte der Cluster blau eingefärbt. Als einziges Domänenwissen wurde in diesem Beispiel die grobe Entfernung zum Gebäude verwendet. Ansonsten geschieht die gesamte Bildung der Mikro- und Makrosemantiken und die Berechnung derer Clusterschwerpunkte automatisch und unabhängig vom Bildinhalt und unserem Wissen über die im Bild vorkommenden Semantiken.

Ausblick und Anwendungen

Eine Evaluation zur Clusteranalyse in Makrosemantiken ist aktuell in Arbeit. Die ersten Ergebnisse sind vielversprechend, so dass in diesem Komplex eine weiterführende Diplomarbeit vergeben wurde. Die nächsten Schritte sind zum einen die Erweiterung des Verfahrens auf andere Merkmale als SIFT, siehe Abschnitt 6.6 und zum anderen der Einsatz der entwickelten Algorithmen zum Matchen zwischen Kamerabildern und einem Modell der Umgebung, siehe Abschnitt 6.3.



Abbildung 15: Clustering der Orte in der Makrogruppe „Fenster“ aus Abbildung 14

4.2 Himmelsdetektion

Im Rahmen der Analyse von Outdoor-Aufnahmen ist es für viele Zwecke wichtig, in Bilddaten schnell und zuverlässig den Himmel erkennen zu können. Ist der Himmel im Bild detektiert, dann sind Rückschlüsse über den Inhalt des Bildes möglich und es lassen sich Informationen über die Wetter- und somit auch Beleuchtungsverhältnisse zur Bildaufnahme extrahieren. Dies verbessert ein realistisches Rendering im CG-Modell und erleichtert damit Vergleiche vom 2-D-Bild zu einem aus der geschätzten Pose gerenderten Bildausschnitt des CG-Modells. Auch kann die Kenntnis über die zum Himmel gehörenden Bildbereiche dazu dienen, den Suchraum für weitere, kompliziertere Bildanalyse-Algorithmen einzuschränken.

Ein weiterer und für uns besonders relevanter Punkt ist, dass in bebauten Gebieten die Unterkante der zum Himmel gehörenden Bereiche oft der Oberkante von Gebäuden entspricht. Der Verlauf dieser Gebäudeoberkanten liefert verwertbare erste Hinweise auf die Position der Kamera im CG-Modell.

Es existieren bereits Himmelsdetektoren für blauen, unbewölkten Himmel. Im Szenario der Pose-detektion in Mitteleuropa es aber wichtig, den Himmel bei allen Witterungen inklusive eventueller Wolken zu finden. Nicht wichtig hingegen ist ein Erkennen des Himmels unterhalb von größeren Gebäudeteilen, wie etwa innerhalb von Torbögen. Es wurde daher ein neues Verfahren entwickelt, das für diesen Anwendungsfall der Posedetektion optimiert ist.

Eine erste Version des Algorithmus wurde auf dem „Farbworkshop 2008“ in Aachen [12] vorgestellt. Das endgültige Verfahren wurde in einem Vortrag auf der Konferenz „Fourth International Conference on Computer Vision Theory and Applications (VISAPP) 2009“ in Lissabon, Portugal der Fachwelt präsentiert und im Rahmen der Konferenz-Proceedings publiziert [14].

Das Verfahrens arbeitet wie folgt: Das Bild wird zunächst mit dem Kuwahara-Filter [M. 76] geglättet und anschließend CSC-segmentiert. Die Segmentierung wird dabei im $L^*a^*b^*$ -Farbraum durchgeführt, wobei die Ähnlichkeit zweier Farben über den euklidischen Abstand bestimmt wird.

Die entstehenden Segmente werden dann über ihre mittlere Farbe in Himmel und Nicht-Himmel vor-klassifiziert. Dabei wird davon ausgegangen, dass jedes Himmelsegment eine direkte, senkrecht nach

oben gerichtete Verbindung zu anderen Himmelsegmenten oder dem oberen Bildrand hat. Diese Einschränkung hat sich für unseren Anwendungsfall als unproblematisch gezeigt.

Anschließend werden Segmente unmittelbar unterhalb des Himmels ausgesondert, die aufgrund ihrer Farbe als Himmel klassifiziert würden, die jedoch tatsächlich zu Gebäuden gehören. Dazu wird die Segmentform im Hinblick auf die bei Gebäudekanten typischen geraden Segmentgrenzen analysiert. Dafür ist es wichtig, dass in Segmentgrenzen die bei der Trennung von Farbverläufen entstehen keine künstlichen geraden Linien auftreten. Dies wäre zum Beispiel beim Segmentierungsalgorithmus „Split and Merge“ [HP76] der Fall, der deshalb hier im Gegensatz zum CSC nicht einsetzbar ist.

Das Verfahren wurde auf 179 Photos des Campus Koblenz aus der Bilddatenbank angewandt, die mit verschiedenen Kameras bei unterschiedlichen Wetterbedingungen aufgenommen wurden und in denen die Grenzen des Himmels im Bild durch die Handannotation bekannt ist. Anschließend wurden Coverability Rate (CR) und Error Rate (ER) für alle Bilder bestimmt (vgl. Abschnitt 2.5).

In 80% aller Photos lag die CR über 0.9 und gleichzeitig die ER unter 0.1. In 75% aller Photos konnte sogar eine CR von über 0.95 und gleichzeitig eine ER von unter 0.05 erreicht werden.



Abbildung 16: Erfolgreiche Himmelsdetektion

Abbildung 16 zeigt ein Beispiel einer erfolgreichen Himmelsdetektion, wobei der detektierte Himmel im rechten Teilbild weiß und ausgesonderte Himmelkandidaten grau dargestellt sind.

4.3 Berechnung von Fluchtpunkten und Fluchtlinien

Motivation

In städtischen Umgebungen bestehen Gebäudefassaden in der Regel aus planaren Flächen, darin liegen die meisten Kanten in senkrechter oder waagerechter Ausrichtung. Bei einer Bildanalyse werden gerade Strecken üblicherweise zu Geraden ohne Endpunkte detektiert. Diese Geraden streben in Architekturumgebungen, wie etwa in unserem Campus, in der Regel entweder zwei oder drei Fluchtpunkten zu. Die Fluchtpunkte können im Bild liegen, z.B. bei einer Zentralperspektive, oder auch potentiell unendlich weit außerhalb des Bildes, und zwar dann, wenn die Blickrichtung des Betrachters orthogonal zur Fläche ist, in der die Geraden liegen.

Die Kenntnis der Anzahl und Position der Fluchtpunkte in einem Bild liefert Informationen über die im Bild sichtbaren Flächen und der räumlichen Ausrichtung des Betrachters zu ihnen. Durch die Zuordnung von Geraden zu detektierten Fluchtpunkten können planare Flächen leichter detektiert und Fassaden leichter lokalisiert werden. Außerdem ist es so möglich, andere Objekte, deren Ausrichtung nicht zu den Hauptrichtungen im Bild passt, wie z.B. Fahrzeuge, Hecken etc., zu lokalisieren und auszublenden.

Für die Bestimmung der Fluchtpunkte und die Zuordnung von Geraden zu den Fluchtpunkten existiert eine Vielzahl an Algorithmen in der Literatur. Allerdings konnten trotz intensiver Suche keine frei verfügbaren Algorithmen gefunden werden, die unseren Qualitätsanforderungen gerecht werden. Deshalb wurde ein neuer Algorithmus entwickelt, der Ideen aus mehreren Methoden aus der Literatur mit einer neuen Nachbarschaftsbeziehung auf Geradenschnittpunkten kombiniert.

Der Algorithmus wurde in einem Artikel beschrieben, der für die Konferenz „15-th IAPR International Conference on Discrete Geometry for Computer Imagery“ in Montreal, Kanada akzeptiert wurde [13]. Im folgenden werden die Grundzüge des Verfahrens vorgestellt, die genauen Details können dem beigegeführten Artikel entnommen werden.

Problemstellung und Stand der Forschung

Bei der Detektion von Fluchtpunkten sind im wesentlichen drei Teilprobleme zu lösen: i) das Finden von Geraden im Bild, ii) die Konstruktion eines Akkumulatorraumes von Geradenschnittpunkten und iii) Interpretation der Werte im Akkumulatorraum der Schnittpunkte.

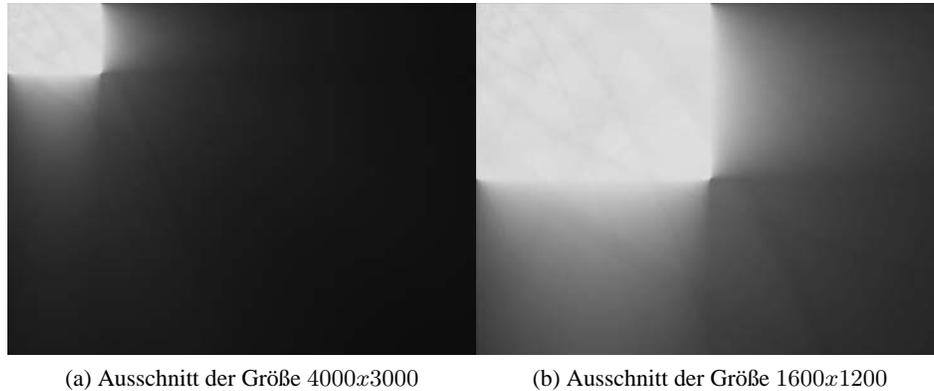
Zur Lösung von Problem i) haben wir eine sehr robuste Implementation der Hough-Transformation für Geraden entwickelt, siehe Abschnitt 3.1.

Als ersten Schritt zur Lösung von Teilproblem ii) werden alle detektierten Geraden entsprechend ihrer Richtung mit dem k-Harmonic-Means-Clustering-Algorithmus [ZHD01] geclustert. Dabei werden zwei Clustering-Durchläufe durchgeführt, einmal mit zwei und das andere mal mit drei Clustern. Falls beim zweiten Durchlauf die Intra-Cluster-Varianz nicht deutlich abnimmt wird das Ergebnis des ersten Durchlauf genutzt, ansonsten das Ergebnis des zweiten Laufes mit drei Clustern. Diese Idee des Clusters der Geraden nach Richtung stammt von Seo [SLC06]. Sie trägt immer dann, wenn die Zahl der zu erwartenden Fluchtpunkte eingeschränkt werden kann.

Nach dem Clustern bildet Seo den potentiell unendlichen Akkumulatorraum der Schnittpunkte von detektierten Geraden über eine reziproke Transformation auf ein endliches Bild bestehend aus drei Teilbildern ab. Die Transformation ist jedoch ad hoc, ohne theoretische Begründung, und bewirkt Probleme im Randbereich zwischen den Subbildern.

Eine andere Methode zum Aufbau eines Akkumulatorraumes, in dem Häufungen von Geradenschnittpunkten detektiert werden können, stammt von Almansa et al [ADV03]. Sie unterscheiden Positionen innerhalb und außerhalb des analysierten Bildes. Der Bildraum aller Positionen innerhalb des analysierten Bildes wird in quadratische bins gleicher Größe aufgeteilt, wobei jedes bin einer Akkumulatorzelle zugeordnet wird. Außerhalb des Bildes wird der Raum in Teilstücke von Kreisringen aufgeteilt die so konstruiert werden, dass für eine zufällige Gerade die den Bildraum berührt alle Teilstücke mit der selben Wahrscheinlichkeit geschnitten werden.

Allerdings wird hierbei von allen theoretisch möglichen Geraden im \mathbb{R}^2 ausgegangen und nicht von den Geraden, die mittels analytischen Methoden wie der Hough-Transformation in diskreten Bildern

Abbildung 17: Φ , messbare Geraden pro Pixel

detektiert werden können. Außerdem erscheint es uns sinnvoller, nicht den Verlauf von Geraden zu akkumulieren, sondern deren Schnittpunkte.

Auf Basis dieser Überlegungen wurde ein neues Verfahren entwickelt, das durch die Definition von Nachbarschaften auf Geradenschnittpunkten ein Akkulatorbild überflüssig macht. Diese Nachbarschaft beruht auf der Analyse der mit dem gewählten Geradendetektor detektierbaren Geraden, die wir *messbare* Geraden nennen.

Neues Verfahren

Die Geradenfrequenz $\Phi : \mathbb{Z}^2 \rightarrow \mathbb{N}$ gibt für eine Position $p \in \mathbb{Z}^2$ die Anzahl aller messbarer Geraden an, die durch p laufen können. Abbildung 17 visualisiert einen Teil von Φ für eine Hough-Transformation mit Akkulatorgröße (500, 500) und einer Größe des analysierten Bildes von 800×600 . Gezeigt wird in Teilbild 17a ein Ausschnitt von Φ der Größe von 4000×3000 und in Teilbild 17b der Größe von 1600×1200 . Die Positionen innerhalb des analysierten Bildes befinden sich in beiden Fällen in der oberen linken Ecke.

Es zeigt sich, dass Φ innerhalb des Bildes weitgehend einem konstanter Wert entspricht, außerhalb des Bildes jedoch sehr stark variiert. Die Idee der Nachbarschaft ist es nun, für jede Position $p \in \mathbb{Z}^2$ eine Nachbarschaft $N(p)$ so zu bestimmen, dass innerhalb aller $N(p)$ die aggregierte Geradenfrequenz konstant ist. Diese Nachbarschaft definiert zwar keine Topologie im Sinn der Mathematik aber eine naive Topologie, die einen hinreichend guten Ersatz für "Nähe" liefert. Formeln zur Bestimmung der Nachbarschaft wurden entwickelt und implementiert.

Zur Lösung von Teilproblem iii), also der Berechnung der Fluchtpunkte, wird für alle Schnittpunkte detektierter Geraden innerhalb einer Richtungsgruppe die Nachbarschaft berechnet. Anschließend werden die Nachbarschaften mit dem AGS-Algorithmus (siehe Abschnitt 3.5) geclustert. Die Mittelpunkte der entstehenden Cluster sind Kandidaten für Fluchtpunkte, aus denen mittels geometrischer Einschränkungen, die an die Ideen Rothers [Rot00] angelehnt sind, die endgültigen Fluchtpunkte gewonnen werden.

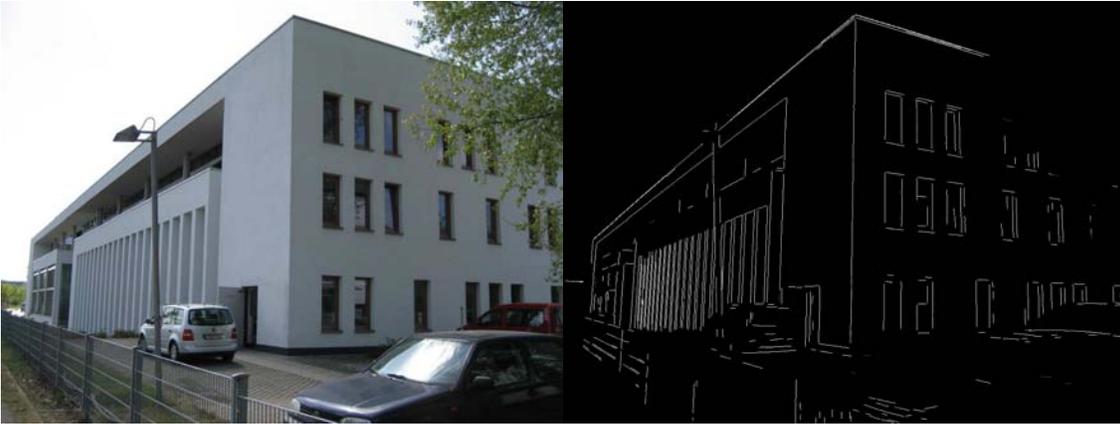


Abbildung 18: Originalbild und Blueprint

Das Verfahren wurde auf Bilder aus der Bilddatenbank mit manuell annotierten Fluchtpunkten evaluiert, dabei konnten 154 von 189 annotierten Fluchtpunkten erfolgreich detektiert werden was einer Erfolgsquote von 81.48% entspricht. Die Fehlerrate beträgt weniger als 4%.

4.4 Blueprint-Rekonstruktion

Es wurde ein Verfahren zur Detektion markanter Fassadenkanten entwickelt, dass auf den berechneten Fluchtpunkten basiert. Das Ergebnis des Verfahrens bezeichnen wir als *Blueprint*. Abbildung 18 zeigt ein Beispiel mit Originalbild links und Blueprint rechts. Das Verfahren wurde auf dem „2009 International Workshop on 3-D Digital Imaging and Modeling“ in Kyoto, Japan eingereicht [9], der im Rahmen der ICCV 2009 stattfindet.

Die erzeugten Blueprints dienen mehreren Zwecken: Zum einen als Grundlage zur Detektion markanter Eckpunkte wie in Abschnitt 5.1 beschrieben, zum anderen, um Fassaden im Bild zu lokalisieren. Ferner ist noch zu untersuchen, ob eine unmittelbare Posebestimmung aus einem Matching der markanten Fassadenkanten des Blueprints mit denen des CG-Modells möglich ist, und falls ja - wovon wir ausgehen -, mit welchen Fehlerraten. Zuletzt sehen wir hier auch eine Möglichkeit, aus mehreren Blueprintbildern aus unterschiedlichen Positionen um ein Gebäude herum, einen 3-D-Blueprint direkt zu rekonstruieren. Existierende Verfahren registrieren hier Punktmerkmale. Eine direkte Registrierung von Geraden in unterschiedlichen Projektionen erscheint interessant.

Der Algorithmus arbeitet wie folgt: Zunächst werden mit dem Canny-Algorithmus [Can86] markante Kantenpunkte im Bild lokalisiert. Das Ergebnis des Canny-Algorithmus ist eine Menge von nicht notwendig geraden Linien entlang der Kanten mit hohen Gradienten im Bild. Wir bezeichnen diese Linien als *C-Linien*. Mit Hilfe der Algorithmen aus Abschnitt 3.4 werden C-Linien in wilden Strukturen entfernt. Die übrigen C-Linien sind die Eingabe einer Hough-Transformation (siehe Abschnitt 3.1), mit der markante Geraden im Bild gefunden werden. Auf Basis dieser Geraden werden mit dem Algorithmus aus Abschnitt 4.3 die Fluchtpunkte im Bild bestimmt.

Während des Canny-Algorithmus wird für jeden Bildpunkt die Richtung und Stärke des Gradienten an diesem Punkt bestimmt. Die Tangente an einer Bildposition p ist die Gerade durch p , deren Richtung orthogonal zur Gradientenrichtung an p ist. Diese Informationen werden genutzt, um Kanten zu

finden, die zu einem Fluchtpunkt zeigen. Dies gelingt jetzt selbst dann, falls diese Kanten nicht hinreichend lang sind, um mittels Hough zu detektierbaren Linien zu führen. Diese Kanten bilden den Blueprint des Bildes.

Oft sind die Dachlinien von Gebäuden (vor allem bei Sattel- oder Walmdächern) nicht parallel zu den Hauptrichtungen der Gebäudefassaden. Deshalb wird der Himmel im Bild berechnet (siehe Abschnitt 4.2) und die Grenze zwischen Himmel und Nicht-Himmel analysiert, um sogenannte *Dachlinien* zu finden. Diese werden dem Blueprint hinzugefügt.

In einem Nachverarbeitungsschritt werden kurze Kanten die parallel zu längeren verlaufen aus dem Blueprint entfernt und kurze Lücken zwischen benachbarten Kanten die auf einer Geraden liegen geschlossen. Oftmals enthält der Blueprint Pfosten, Lampen oder Antennen, die senkrecht in den Himmel ragen. Da der Himmel bekannt ist, können solche Objekte identifiziert und entfernt werden. In einem letzten Schritt werden alle kurzen Kanten in deren Umgebung keine anderen Kanten vorhanden sind entfernt.

Wir haben den Blueprint-Algorithmus auf einer großen Zahl von Bildern getestet und konnten dabei feststellen, dass er robust die markanten Fassadenkanten findet und dabei die allermeisten Kanten in anderen Objekten ausblendet.

5 Elementare markante Merkmale aus höheren markanten Merkmalen

Elementare markante Merkmale ohne semantische Bedeutung wie Geraden, Punktmerkmale oder Flächen werden in der Regel direkt über elementare Operationen aus Bildern bestimmt. Es wurden von uns alternative Vorgehensweisen untersucht, bei denen elementare markante Merkmale nicht direkt berechnet sondern aus höheren markanten Merkmalen gewonnen werden.

5.1 Orte für Punktmerkmale aus semantischem Wissen

Ein klassisches Herangehen zum Matching zwischen räumlich oder zeitlich versetzt aufgenommenen Bildern ist die Suche nach korrespondierenden Punktmerkmalen in beiden Bildern. Punktmerkmale bestehen aus einem Ort und einer Beschreibung der Umgebung des Ortes. Die Beschreibung erfolgt dabei üblicherweise numerisch als Vektor oder Matrix. Verbreitete Punktmerkmale sind Merkmale aus dem Harris Eckendetektor [HS88] und heute vor allem SIFT [Low99] und darauf aufbauende Verfahren wie SURF [BTVG06].

Bei all diesen Verfahren wird sowohl der Ort bestimmt, an dem Merkmale angelegt werden, als auch deren Merkmalsbeschreibung generiert. Dabei ist die Entscheidung, ob an einer Bildposition ein Punktmerkmal erstellt wird, nur von der lokalen Umgebung des Merkmals abhängig. Dies hat zur Folge, dass häufig viele Merkmale in unsicheren (aber lokal stark strukturierten) Strukturen wie Baumästen gefunden werden. Außerdem ist speziell bei SIFT auffällig, dass zwar die Beschreibung der Merkmalsumgebung robust gegen Änderungen in der Skalierung ist, die Bestimmung der Merkmalsorte selbst jedoch stark von der Skalierung abhängig ist.

Wir haben es uns deshalb zum Ziel gesetzt, einen neuen Algorithmus zu entwickeln, der bessere Orte für Punktmerkmale erzeugt. Hierbei soll die Entscheidung, ob eine Bildposition Ort eines Punktmerkmals wird, nicht wie in bekannten Verfahren lokal von dem Inhalt eines Fenster rund um die

Bildposition abhängen. Stattdessen werden die Orte über semantisches Wissen über den Inhalt des Bildes bestimmt.

In einem ersten Schritt werden hierzu Eckpunkte in Gebäudefassaden gesucht und als Orte für Punktmerkmale bestimmt. Dafür wird auf die Blueprint-Rekonstruktion (siehe Abschnitt 4.4) zurückgegriffen, um markante Eckpunkte zu lokalisieren.

Der Output der Blueprint-Rekonstruktion ist eine Liste von Liniensegmenten, die markanten Kanten in Gebäudefassaden entsprechen. Für jedes Liniensegment ist bekannt:

- Startpunkt,
- Endpunkt,
- Länge,
- der Fluchtpunkt, auf den das Liniensegment zeigt.

Als Gerade durch ein Liniensegment bezeichnen wir die Gerade durch deren Start- und Endpunkt. Für jedes Liniensegment L , dessen Länge über einem Schwellwert t liegt, wird die Menge $Cand_L$ aller anderen Liniensegmente gebildet, deren Länge ebenfalls über t liegt und die zu einem anderen Fluchtpunkt als L zeigen.

Für alle Liniensegmente L_i in $Cand_L$ wird der Schnittpunkt zwischen der Geraden durch L und der Geraden durch L_i berechnet. Falls mindestens einer dieser Schnittpunkte nahe dem Startpunkt von L liegt, werden die Positionen aller Schnittpunkte die nahe am Startpunkt von L liegen gemittelt und ergeben einen markanten Eckpunkt. Entsprechend werden anschließend alle Schnittpunkte nahe L s Endpunkt zu einem zweiten markanten Eckpunkt gemittelt.

Zusätzlich werden markante Eckpunkte in der im Blueprint enthaltenen Horizontlinie bestimmt. Dazu werden Punkte in der Horizontlinie gesucht, an denen längere, gerade Teilstrecken unterschiedlicher Richtung aufeinander stoßen. Außerdem werden an vertikalen Sprüngen in der Horizontlinie markante Eckpunkte angelegt.

Für jeden der so gewonnenen markanten Eckpunkte wird eine Güte bestimmt, die ausdrückt, wie sicher der Eckpunkt ist. Die Güte setzt sich aus vier Kriterien zusammen:

- Je mehr Liniensegmente bei der Konstruktion des Punktes beteiligt waren, desto größer die Sicherheit.
- Je länger die bei der Konstruktion des Punktes beteiligten Liniensegmente, desto höher die Sicherheit.
- Je näher der Punkt an den Enden der beteiligten Liniensegmenten, desto höher die Sicherheit.
- Je höher der Punkt im Bild, desto höher die Sicherheit, da bodennahe Kanten oft durch Fahrzeuge, Personen, Büsche etc verdeckt werden.

Aktuell wird daran geforscht, wie die so erzeugten markanten Eckpunkte genutzt werden können, um Bilder miteinander in Korrespondenz zu setzen, siehe Abschnitt 11.

6 Matching

6.1 Allgemeines

Durch die Verfahren in Abschnitt 4 werden höhere markante Merkmale in Bildern bestimmt. Diese werden zum Matching zwischen Modell und Bilddaten eingesetzt. Hier können im wesentlichen zwei Ansätze unterschieden werden: Zum einen die Berechnung der Merkmale aus einem Rendering des Modells und anschließende Korrespondenzberechnung mit Merkmalen, die in Bildern berechnet wurden. Zum anderen die Auswertung einer direkten Annotation von Merkmalen im semantischen Modell, so dass nur im Kamerabild eine Berechnung der im Modell annotierten Merkmale stattfinden muss.

Der zweite Ansatz bietet offensichtliche Vorteile, ist jedoch nur für solche Merkmale möglich, die unabhängig von der Sicht der Kamera auf den Merkmalsort sind. So ist es zum Beispiel nicht möglich, Beschreibungen des Horizonts statisch im Modell zu annotieren, da die Horizontlinie abhängig ist von Standpunkt und Blickrichtung des Betrachters.

6.2 Matching von Horizontlinien

Basierend auf den Algorithmen aus Abschnitt 4.2 wurde ein erstes Verfahren untersucht, um eine in einem Bild detektierte Horizontlinie mit der Horizontlinie in einem Rendering des Umgebungsmodells zu vergleichen. Dieses Verfahren wird genutzt, um zu einem Photo, das an einer grob bekannten Position mit unbekannter Kameraausrichtung aufgenommen wurde, die Kameraausrichtung zu schätzen. Dazu werden von der angenommenen Position im CG-Modell Bilder aus unterschiedlicher Kameraausrichtung gerendert, im Photo und in allen Renderings werden die Horizontlinien bestimmt, transformiert und verglichen. Dasjenige Rendering, bei dem nach der Transformation die beste Übereinstimmung erreicht wurde, wird als wahr angesehen und die Kameraausrichtung aus der es aufgenommen wurde zurückgeliefert. Dieses Verfahren hat allerdings momentan eher den Stand eines Proof-of-Concept. Es ist jedoch vielversprechend und wird in den nächsten Monaten eingehender untersucht werden.

6.3 Matching von SIFT-Merkmalgruppen

Es wurden erste Experimente durchgeführt, wie die in Abschnitt 3.5 beschriebenen Verfahren zur Analyse und Gruppierung von SIFT-Merkmalen zum Matching zwischen einem 3-D-Modell der Umgebung und Kamerabildern genutzt werden können. Diese Untersuchungen stehen allerdings noch am Anfang.

Es wurde untersucht, ob SIFT-Merkmale in einem Kamerabild und SIFT-Merkmale aus einem Rendering des 3-D-Modell aus einer Pose, die ungefähr dem Bild entspricht, benachbart sind. Bei korrekt gewählter Beleuchtung des 3-D-Modells ist dies tatsächlich der Fall. Abbildung 19 zeigt ein Beispiel. Das mit einem roten Pfeil gekennzeichnete SIFT-Merkmal aus dem gerenderten Bild des CG-Modells (rechtes Bild) besitzt alle mit gelb umrandeten Nachbarn im linken und rechten Bild. Links ist das Originalbild, aus dem die Pose für das gerenderte Bild geschätzt wurde.

Dieses Ergebnis ist durchaus überraschend. Es konnte nicht damit gerechnet werden, dass sich Nachbarschaften so gut über mehrere Bilder fortsetzen, die mit derart unterschiedlichen Techniken (Kame-



Abbildung 19: Semantisch ähnliche SIFT-Merkmale in Kamerabild und Rendering

ra, gerendert) gewonnen wurden. Daher muss dieser Ansatz zu einer möglichen neuen Matchingtechnik weiter ausgebaut werden.

Diese Art des Matchens von Merkmalsgruppen sehen wir als Ergänzung zur klassischen Nutzung von SIFT, bei der einzelne ausgewählte Merkmale aus zwei Bildern in Korrespondenz gesetzt werden. Bei diesem klassischen Ansatz können nur solche Merkmale verwendet werden, die in gewisser Weise ausgezeichnet sind, da ansonsten keine eindeutige Zuordnung möglich wird. Mittels unserer Gruppierung müssen wir uns nicht auf ausgezeichnete Merkmale beschränken, sondern können Merkmale untersuchen, die mehrfach im Bild vorkommende Strukturen repräsentieren. Wir erhoffen, so zu mehr und zu stabileren Zuordnungen zu gelangen.

6.4 Matching markanter Geraden

In unseren Untersuchungen konnten wir feststellen, dass Geraden entlang markanter Kanten ein sehr charakteristisches Merkmal von Fassaden sind, und sich somit gut zur Orientierung in Umgebungsmodellen eignen. In ersten Ansätzen wurden deshalb Verfahren untersucht, um Geraden in einem Modell zu Geraden, die mittels der Hough-Transformation in einem Kamerabild detektiert wurden, in Korrespondenz zu setzen.

Problematisch hierbei ist, dass nicht alle Geraden, die in einem Kamerabild detektiert werden, auch geometrischen Kanten des Modells entsprechen. Oft entstehen Kanten aufgrund der Textur von Objekten, bei Fassaden, z.B. durch unterschiedliche angestrichene Gebäudeteile oder durch Objekte wie Fallrohre, Laternen, Schattenwurf. Andererseits sind nicht alle Kanten, die im Modell geometrisch charakteristisch sind, im Kamerabild auffindbar. So bilden z.B. Vorsprünge in homogen gefärbten Fassaden oft nur geringe Gradienten im Bild.

Dieses Problem sollen im weiteren Verlauf durch auf Grund unserer gewonnenen Erfahrung verbesserte Matchingverfahren gelöst werden.

6.5 Detektion von Fassaden

Es werden z.Zt. Algorithmen zur sicheren Detektion von Fassaden von Gebäuden entwickelt. Dazu werden die bereits vorhandenen linienbasierten Algorithmen aus der Blueprint-Rekonstruktion mit neuen flächenbasierten Verfahren kombiniert. Ergebnis des Algorithmus soll eine Menge von Beschreibungen von Gebäudefronten im Bild werden, die jeweils aus der Kontur der Region sowie einer geeigneten Beschreibung des Inhalts bestehen. Diese Beschreibung kann z.B. Informationen über die mittlere Farbe der Region, charakteristische Geraden, die Anzahl der Stockwerke, Textur etc. enthalten.

6.6 Erweiterungen von SIFT-Merkmalen

Die Algorithmen zur Analyse von SIFT-Merkmalen (siehe Abschnitt 3.5 und 6.3) werden aktuell erweitert. Dieser Arbeitsbereich gliedert sich im Wesentlichen in vier Einzelbereiche.

Zunächst soll eine Trennung des Verfahrens zur Gewinnung von SIFT-Merkmalen in den Teil der reinen Detektion der Orte und den Teil der Beschreibung der gefundenen Orte implementiert werden. Anschließend wird überprüft, ob es für die bisher auf SIFT-Merkmalen operierenden Algorithmen (siehe Abschnitte 3.5 und 4.1) hilfreich ist, gefundene Orte anderer Detektoren (z.B. Harris-Eckendetektor [HS88] oder markante Orte aus semantischem Wissen, siehe Abschnitt 5.1) durch SIFT-Merkmale zu beschreiben und diese zusätzlich zu den "herkömmlichen" SIFT-Merkmalen zu betrachten. Speziell in diesem Bereich gibt es einige Vorarbeiten, deren Erkenntnisse wir in unserer Arbeit berücksichtigen wollen.

Im zweiten Bereich werden die SIFT-Deskriptoren durch Farbinformationen ergänzt. Die bei SIFT nicht beachtete Farbe in der Umgebung des Merkmals soll als zusätzliche beschreibende Information des Merkmals zur Verfügung stehen und zur Bestimmung der Ähnlichkeit zwischen Merkmalen mitbenutzt werden. Da bereits wissenschaftliche Arbeiten zu ähnlichen Arbeitsbereichen existieren, werden zunächst die dort vorgestellten Verfahren auf Eignung für unsere Zwecke geprüft. Weiterhin wird ermittelt, ob es auch sinnvoll sein kann, diese Farbinformation nur von bestimmten Merkmalen zu verwenden, und ob man diese Merkmale gezielt finden kann.

Als dritter Punkt wird untersucht, welche Art von Merkmalen (dies können auch Merkmale sein, die aus anderen Verfahren als SIFT gewonnen wurden) an welchen Punkten zur Gewinnung bestimmter Makrosemantiken am besten benutzt werden. Ein konkretes Beispiel wären Merkmale für eine Makrosemantik, die alle Gesichter in Bildern von Menschenmengen (z.B. an öffentlichen Plätzen) beschreibt, was mit SIFT-Merkmalen nicht möglich ist.

Weiterhin ist ein Vergleich zwischen unseren Methoden und dem Verfahren von Wenzel, Drauschke und Förstner [WDF08] geplant. Soweit möglich soll eine Integration beider Verfahren die jeweils besten Ansätze kombinieren. U.a. soll ein Vergleich unserer Abstandsmaße bzgl. SIFT-Merkmalen mit denen von Wenzel et al. durchgeführt werden. Ebenso wird der vorgestellte Deskriptor für die sich wiederholenden Strukturen auf Verwendbarkeit für unsere Anwendung geprüft werden. In der Rektifizierung der abgebildeten Fassaden vor der eigentlichen Untersuchung liegt ein weiterer für uns interessanter Ansatz. Das Verfahren als Ganzes ist jedoch nicht für unsere Zwecke geeignet, da die Autoren anmerken, dass ihr Algorithmus nur auf Bildern arbeitet, die nur einen regelmäßigen Ausschnitt einer Fassade zeigen, und dass weitere regelmäßige Strukturen im Bild die Methode beeinträchtigen. Weiterhin fehlt eine quantitative Evaluation des Verfahrens.

7 Modellgenerierung

Modelle die sich zur Poseschätzung eignen können auf vielfältige Weise hergestellt werden. Es wurden neben manuellen Methoden (Kapitel 2) auch semiautomatische Methoden (Kapitel 7.1, Kapitel 13) und die vollautomatische Erstellung von Modellen (Kapitel 7.2) auf Tauglichkeit untersucht. Im Vorfeld wurden Kriterien festgelegt, welche von dem Modell und den damit verbundenen Sprachen und möglichen Repräsentationen erfüllt werden müssen (Kapitel 12). Die Ergebnisse sind den Arbeitspaketen „ASCG-Abstrakte semantische CG-Modelle“, „ISCG-Initiales semantisches CG-Modell“ und „Semantisches CG-Modell“ aus dem ursprünglichen Antrag zuzuordnen.

7.1 Semiautomatische Modellierung

Um Umgebungsmodelle möglichst schnell in möglichst großem Umfang herzustellen, sind stark automatisierte Methoden wünschenswert. Voll automatisierte Erstellung von semantischen Modellen ist zur Zeit generell noch nicht möglich, weshalb ein Fokus auf der semiautomatischen Erzeugung von CityGML Modellen [GKCN08] lag.

Durch das Photostitching Tool Hugin⁴ unterstützt wurden für die Geometrie von Gebäuden relevante Punktkorrespondenzen in zwei Aufnahmen ausgewählt (siehe Abbildung 20).

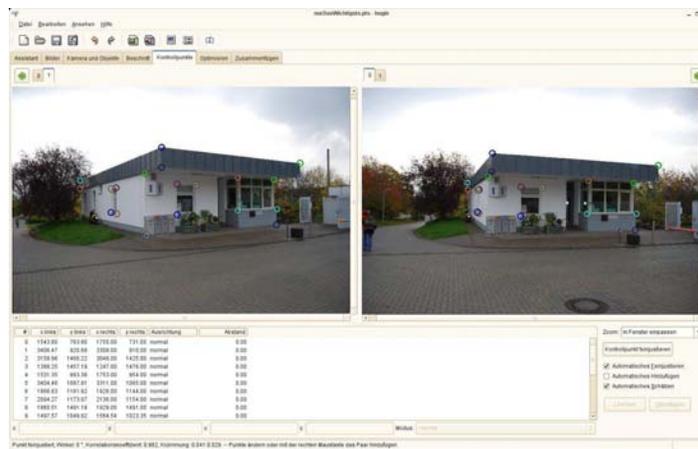


Abbildung 20: Punktkorrespondenzen in Hugin mit automatischer subpixelgenauer Positionsverfeinerung.

Aus diesen Korrespondenzen wurden die relativen Kamerapositionen errechnet und die Punkte in 3-D rekonstruiert. In einem abschließenden Verarbeitungsschritt wurden die Punkte zu Fassaden, Fenstern, Türen, Dächern usw. verbunden und schließlich als texturiertes CityGML Modell exportiert. Auf dieses Modell wurden vollautomatische Methoden der Modellverfeinerung angewendet, um die Qualität zu erhöhen und das Modell zu erweitern (vgl. Arbeitspaket „MV-Automatische bildbasierte Modellverfeinerung“). Das Resultat wurde im IFC-Explorer [For09] visualisiert (siehe Abbildung 21).

Diese Arbeit wurde in enger Kooperation mit der Arbeitsgruppe Ebert und dem DFG-Projekt ”Komponentenorientiertes Konzept zur Nutzung von Modellen und Wissen bei der Objektwiedererkennung

⁴<http://hugin.sourceforge.net>

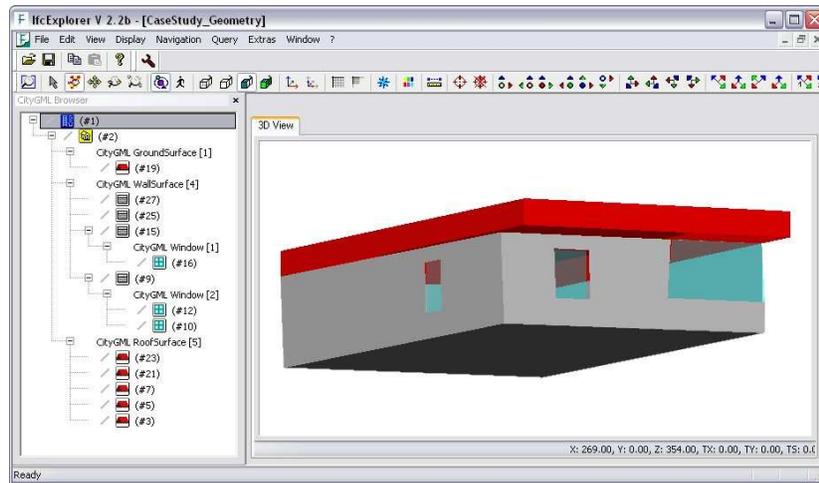


Abbildung 21: Visualisierung des semiautomatisch erzeugten CityGML Modells im IFC-Explorer. Rechts: Untexturierte, orthographisch projizierte Ansicht des Gebäudes mit unterschiedlicher Färbung der semantisch unterschiedlichen Gebäudeteile. Links: Hierarchische Struktur der Gebäudeelemente.

in Bildern und Bildfolgen" (EB 119/3-1;PA 599/8-1) erstellt. Die Ergebnisse wurden auf der Geoinformatik 2009 veröffentlicht ([2], siehe Anhang). Eine genauere Besprechung der Inhalte findet sich im Bericht des DFG-Projekts EB 119/3-1.

Auf diesen Ansätzen und Modellen aufbauend entstanden in der Arbeitsgruppe Ebert weitere Veröffentlichungen im Bereich graphbasierter Modellverfeinerung [FE09].

7.2 Automatische Erzeugung von CG-Modellen

Eine vollständig automatische Erzeugung texturierter 3-D Modelle aus Bilddaten bietet eine Alternative zu der Modellierung von Hand. Die Vorteile sind die höhere Geschwindigkeit mit der Modelle erzeugt werden können, geringerer personeller Aufwand und dass keine Voraussetzungen wie Grundrisse oder extra erhobene Vermessungsdaten der Gebäude nötig sind.

Die Nachteile liegen in der fehlenden Semantik, den höheren Ansprüchen an die Algorithmen zur Modellgenerierung und einer im allgemeinen geringeren zu erwartenden visuellen Qualität der resultierenden Modelle.

Diesen Ansatz verfolgend wurde im Rahmen einer Diplomarbeit [8] ein System entwickelt, welches texturierte 3-D Meshes aus Stereobildsequenzen erzeugt. Das System besteht aus fünf aufeinanderfolgenden, unabhängigen Schritten:

1. Erstellung von dichten Tiefenkarten aus Stereobildern
2. Schätzen der relativen Aufnahmepositionen mittels ICP
3. Fusion der Tiefenkarten
4. Meshgenerierung aus den Punktwolken
5. Ausdünnung und Texturierung des Meshs

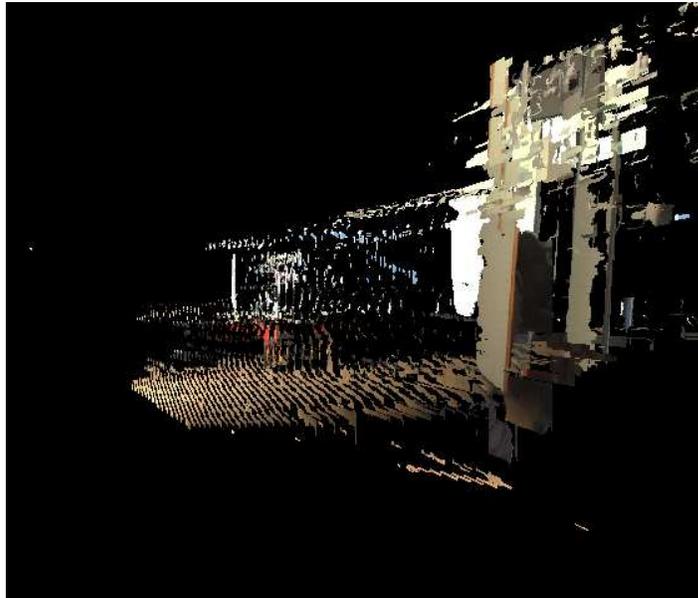


Abbildung 22: Grob quantisierte Disparitätswerte führen zu Ebenenbildung orthogonal zur Blickrichtung, welche eine erfolgreiche Registrierung von Tiefenkarten mittels ICP erschweren.

Die *Erstellung von dichten Tiefenkarten aus Stereobildern* erfordert eine vorausgehende Rektifikation der Bilder, um korrespondierende Scanlines auf gleicher Höhe auszurichten. Hier wurde ein Homographie-Matrix basiertes Verfahren verwendet [FTV97]. Eine Alternative stellen Verfahren dar, die die Bilder durch Polarkoordinaten parametrisieren [PKG99]. Zur Erstellung der Disparitätskarten wurde der Algorithmus nach Birchfield [BT98] verwendet.

Um die *relative Pose zwischen den Stereosystemen*, bzw. derer Tiefenkarten zu bestimmen wurde auf den Iterative Closest Point (ICP [Nüc06]) Algorithmus zurückgegriffen. Der ICP ist ein iteratives Verfahren welches versucht, Punktwolken gegeneinander zu registrieren. Im Vergleich zur Bestimmung der relativen Kamera Pose über den 8-Punkte-Algorithmus und die Epipolargeometrie hat es den Vorteil, dass keine Korrespondenzen bekannt sein müssen da diese vom Verfahren gefunden werden und dass sofort eine korrekte Skalierung des Translationsvektors vorliegt (die Zerlegung der E -Matrix liefert lediglich eine Richtung). Dafür benötigt der ICP eine initiale Schätzung, da er sonst Gefahr läuft in lokalen Minima zu terminieren und sehr viele Iterationsschritte zu benötigen.

Versuche der Registrierung von Tiefenbildern direkt mittels ICP haben gezeigt, dass sich das Verfahren hier nicht eignet. Die ausschlaggebende Grund hierfür ist diskrete Natur der von Algorithmen zur Tiefenbildgenerierung gelieferten 3-D Koordinaten. Da in den meisten Fällen lediglich ganzzahlige Disparitäten zugelassen werden entstehen in den aus dem Tiefenbild generierten 3-D-Punktwolken Flächen orthogonal zur Blickrichtung der Referenzkamera (siehe Abbildung 22). Diese Flächen werden durch den ICP stets aufeinander registriert, da diese Konstellation ein stabiles Minimum bildet. So können keine kleinen Rotationen mehr erkannt werden.

Um diese Fragestellung zu lösen, wurden auf den Stereobildpaaren SIFT-Merkmale [Low04] (vgl. auch Kapitel 3.5) extrahiert, die dann zur Positionsbestimmung mittels ICP verwendet wurden. Zur Entfernung von Ausreißern und zu stark verrauschte Werte aus den Punktkorrespondenzen wird ei-

ne Variante des ICP verwendet, die explizit Ausreißer modelliert [RL01]. Die gefundenen Merkmale werden in eine 3-D Karte eingetragen, wo sie als Referenzpunkte bei der Positionesbestimmung aller folgenden Stereobildpaare dienen. Ihre Position in der Karte wird dabei weiter durch ein eindimensionales Kalman Filter mit festem Mess-, Positions und Systemrausche verfeinert (entspricht einem einfachen Mittelwertfilter).

Bei der anschließenden *Fusion der Tiefenkarten* ist das Ziel, Konflikte in Folge falsch geschätzter Tiefenwerte aufzulösen und redundante Punkte zu entfernen um schließlich alle Tiefenkarten zusammen zu setzen. Hierzu wurde eine vereinfachte Version des Confidence-Based-Fusion Algorithmus [MAW⁺07] implementiert, welche ohne initiale Konfidenz der Tiefenwerte auskommt und die Konfliktarten Free-Space und Occlusion zu einer einzigen Art zusammenfasst, indem Konflikte erst behandelt werden, wenn bereits alle Daten der Tiefenkarten fusioniert wurden.

Die *Meshgenerierung* aus den resultierenden Punktwolken hat das Ziel, Punkte zu Dreiecken zu verbinden, um so Oberflächen zu generieren. Da eine 3-D Triangulierung sehr komplex ist und oft nicht die gewünschten Ergebnisse liefert, wurde auf die 2-D Delauny Triangulierung zurückgegriffen. Hierzu wurden die fusionierten Tiefenkarten in unterschiedliche Ansichten projiziert und dort trianguliert. Verbindungen zwischen Punkten mit zu starken Tiefensprüngen in einzelnen Ansichten wurden durch eine Segmentierung der Tiefenwerte verhindert um Rubber-Sheet Artefakten vorzubeugen. Diese starken Sprünge werden im anschließenden Schritt der Fusionierung der einzelnen Meshes wieder geschlossen, wenn sich an den Stellen neue Dreiecke ergeben.

Als letztes wurde das resultierende Mesh *ausgedünnt und texturiert*. Zur Ausdünnung wurde der Memoryless-Simplification Algorithmus nach Lindstrom und Turk [LT98] verwendet. Dieser Algorithmus ersetzt Kanten und ihre adjazenten Vertices durch einen einzelnen neuen Vertex, der dann zu den umgebenden Vertices neue Kanten bildet. Es werden nur Kanten entfernt, durch deren Fehlen Annahmen bezüglich der Dreiecksform, Umrandung und Volumen des Meshes nicht zu stark verletzt werden.

Der abschließende Schritt der Texturierung des Meshes wird durch Projektion der Vertices in die Ansichten der Stereokameras gelöst, aus denen Texturkoordinaten entstehen. Ist das Dreieck in mehreren Ansichten sichtbar, wird die Textur mit dem geringsten abstand zur Mediantextur gewählt wie in [NfnaodS01] vorgeschlagen.

Visualisierte Ergebnisse des implementierten Verfahrens sind in Abbildung 23 und Abbildung 24 zu sehen.

Eine abschließende Evaluation hat unter anderem folgende Grenzen des Verfahrens offenbart:

- Die Stereobasis muss der Szene angepasst sein. Dies ist einer der Hauptgründe, warum das Verfahren nur sehr begrenzt für Outdoor Szenarien geeignet ist. Eine Stereobasis bei der angemessene Tiefenkarten von Gebäuden entstehen ist in der praxis wenig praktikabel.
- Es ist eine ausreichende Texturierung des Szene notwendig, um genügend SIFT-Korrespondenzen finden zu können.
- Beleuchtungsänderungen führen zu ungleichmäßigen Texturen im Modell.

Für technische Details der Umsetzung siehe Anhang A.

Stereobasis in Outdoorumgebungen

Im Folgenden wird die theoretische Grundlage für die Nutzung von Stereokameras in Outdoorumge-

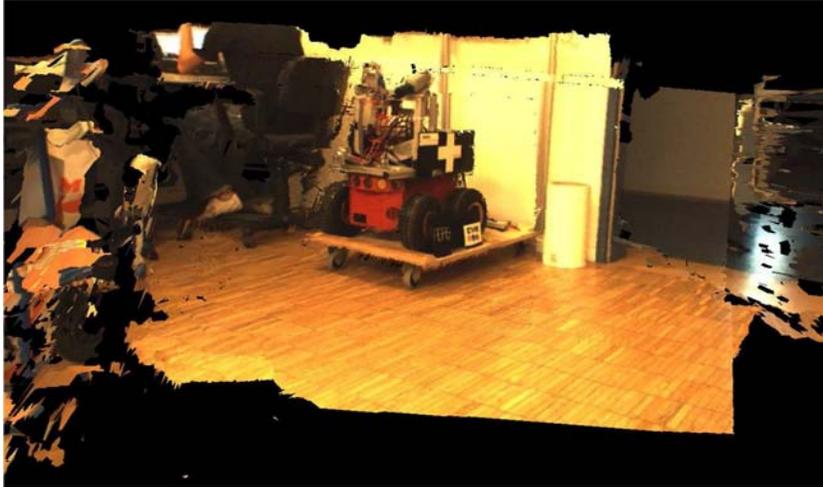


Abbildung 23: Screenshot einer neuen Ansicht des aus einer Stereobildpaarfolge automatisch generierten 3-D Modells (aus [8]).

bungen zur Rekonstruktion und zur Poseschätzung besprochen, welche die Ergebnisse des vorhergehenden Kapitels erklärt.

Ausgehend von einem optimal parallel ausgerichteten Stereokamerasystem: Sei F die Brennweite, b die Stereobasis, p_z die z-Koordinate des Objektes im Kamerakoordinatensystem und d die Disparität ebendieses Objektes. Es ergibt sich folgender Zusammenhang:

$$p_z = \frac{Fb}{d} \quad (1)$$

Für einen beispielhaften Stereoaufbau wird eine Stereobasis von $b = 30\text{cm}$, eine Brennweite von $F = 6\text{mm}$ und einer Pixelgröße von $4,65\mu\text{m}$ angenommen. Aus diesen Werten ergibt sich ein Zusammenhang zwischen Disparität und Entfernung zum Objekt, der bei typischen Outdooranwendungen (Objektentfernung über 15m) eine wesentlich zu grobe Quantisierung aufweist (vgl. Abbildung 25). Die Effekte sind starke Stufenbildungen in dichten Tiefenkarten der Gebäudefassaden, welche eine Rückführung auf Ebenen quasi unmöglich machen und sich so auch nicht für automatische Modellgenerierung eignen. Mögliche Änderungen an dem Aufbau sind:

Größere Brennweite: Teleobjektive verfügen nur über einen wesentlich geringeren Öffnungswinkel und bilden daher einen kleineren Bereich der Welt ab. Rekonstruktion und Poseschätzung werden stark negativ beeinflusst.

Größere Stereobasis: Eine größere Stereobasis ist umständlich in der Handhabung und unrealistisch für eine Applikation. Objekte die sich vergleichsweise nah an den Kameras befinden (Passanten, der nahe Boden) würden eine sehr hohe Disparität aufweisen, d. h. die Algorithmen zur Tiefenkarten Erstellung müssten mit großen Disparitäten rechnen was sich negativ auf die Laufzeit auswirkt, da es sonst zu starken Fehlern in den Tiefenkarten kommt.

Trotz dieser Schwachstellen für die direkte Rekonstruktion bietet ein Stereosystem vielleicht eine gute Möglichkeit für die Pose-Schätzung, für welche auf Subpixelgenau lokalisierte Merkmale im Bild zurückgegriffen und damit der Quantisierungsfehler verringert werden kann.

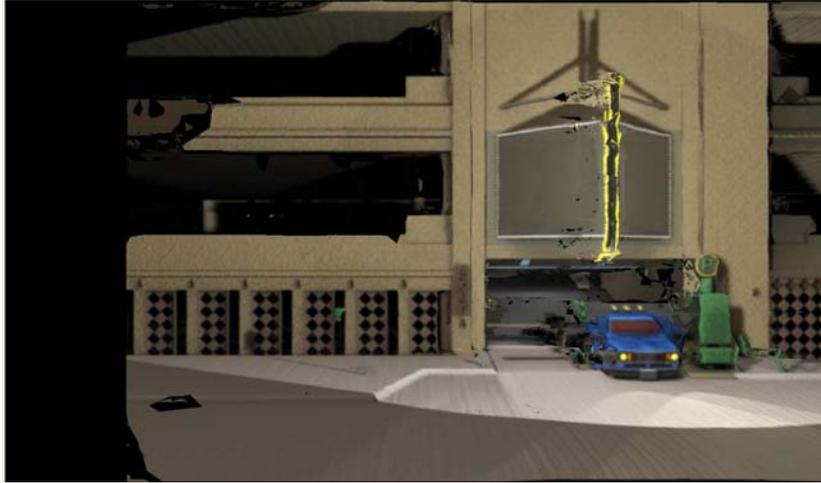


Abbildung 24: Screenshot einer neuen Ansicht des aus einer synthetischen Stereobildpaarfolge automatisch generierten 3-D Modells (aus [8]).

7.3 3-D Laserscanner zur Rekonstruktion

In weiteren Projekten der Arbeitsgruppe Aktives Sehen wurde unter anderem neue Technologie auf dem Gebiet der 3-D Laserscanner untersucht. Konkret wurden ein *Velodyne HDL-64E* Laserscanner und drei *Phillips 1300NC* USB-Kameras auf ein Fahrzeug montiert. Die Kameras sind nach vorne und zu den Seiten ausgerichtet. Mit diesem Aufbau wurden Aufnahmen des Campus Geländes gemacht. Abbildung 26 zeigt Screenshots der Applikation, die die gesammelten Daten visualisiert.

Mit über einer Million Messpunkte pro Sekunden und einem entfernungsunabhängigen Distanzrauschen der Messungen von unter 5cm ist diese Technologie geeignet, dichte Punktwolken der Umgebung zu liefern, aus welchen Modelle von Gebäuden entstehen können.

7.4 Inpainting

Bei der Datenakquisition im Außenbereich zeigt es sich, dass häufig Personen und zusätzliche Objekte im Bild zu sehen sind. Diese müssen bei der Modellerstellung erkannt und dann ignoriert werden. Alternativ können die Bilder interaktiv nachbearbeitet werden und die störenden Bildbereiche werden entfernt. In der Diplomarbeit [15] wurden mehrere Verfahren untersucht und evaluiert, mit denen ein so genanntes *Inpainting* durchgeführt werden kann. Unter *Inpainting* versteht man das Auffüllen von maskierten Bereichen im Bild mit Bildinformation, ohne dass Artefakte entstehen. Ergebnisse für die Frontansicht der Bibliothek mit einem in [15] verbesserten Verfahren nach [IJ07] sind in Abbildung 27 zu sehen.

8 Pose-Rückrechnung

Das mathematische Problem der Rückrechnung einer relativen Pose (zwischen mehreren Kamerasystemen über 2-D / 2-D Bild zu Bild Korrespondenzen) oder einer absoluten Pose (einer Kamera

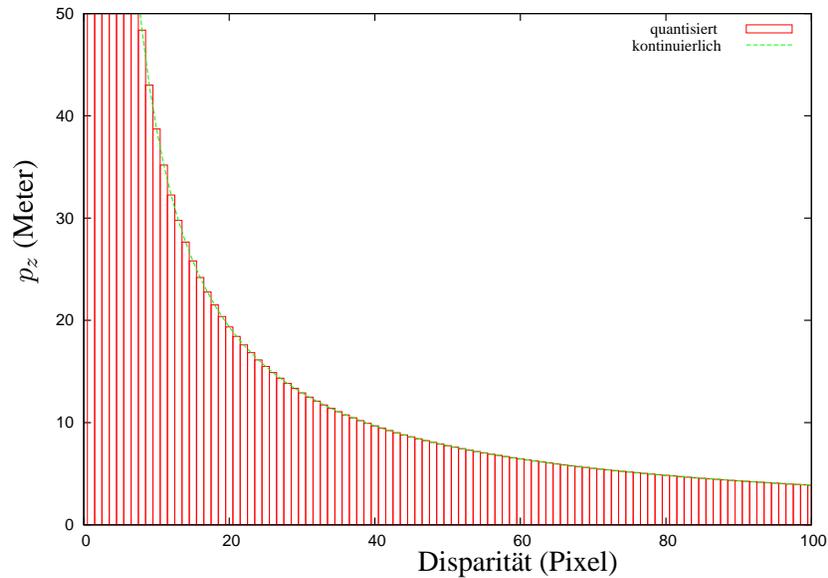


Abbildung 25: Umrechnung von Disparitäten in Entfernungen für einen gewöhnlichen Stereo-Aufbau. Die Quantisierung für höhere Entfernungen ist sehr grob.

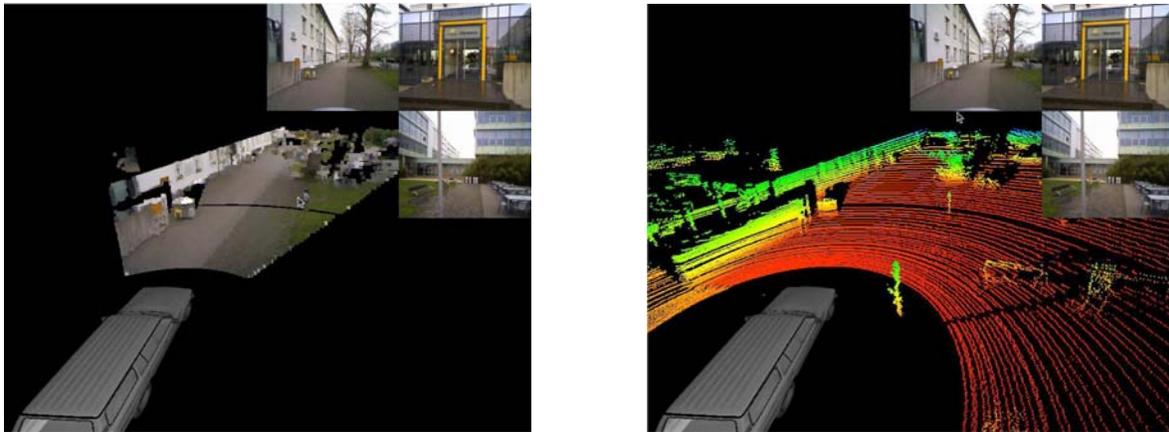


Abbildung 26: 3-D Laserscan auf dem Campus, links mit Textur der vorderen Kamera.

zu einem Weltkoordinatensystem über 2-D / 3-D Bild zu Welt Korrespondenzen) ist ein essenzielles Teilgebiet der Poserekonstruktion und des Posetrackings. Die folgenden Ergebnisse sind hauptsächlich dem Arbeitspaket „PRR-Pose Rückrechnung“ des ursprünglichen Antrags zuzuordnen.

8.1 Relative Pose

Die Berechnung der relativen Pose zwischen zwei oder mehr Bildern mit automatischen Verfahren ist eine wichtige Voraussetzung für automatische und semiautomatische Modellgenerierung aus monokularen Kamerabildpaaren und -folgen. Die automatische Detektion von geeigneten Punktmerkmalen

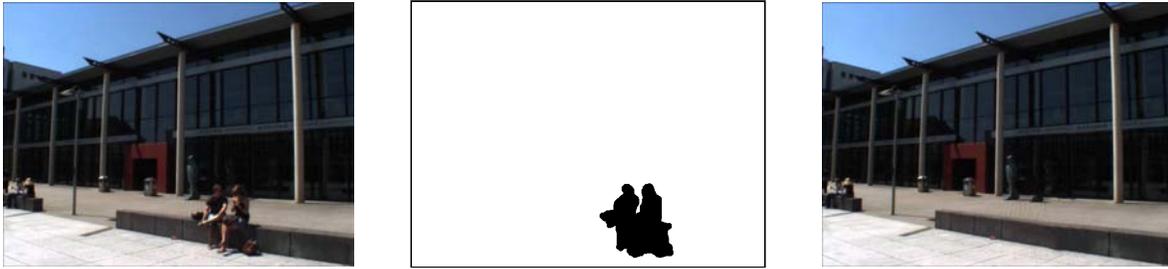


Abbildung 27: Inpainting: links das Originalbild, in der Mitte die interaktiv erstellte Maske, rechts das Ergebnis des Verfahrens

wie SIFT [Low04] oder SURF [BTVG06] ermöglicht eine Zuordnung von Punkten zwischen Bildern anhand derer Deskriptoren. Diese noch fehlerbehafteten Korrespondenzmengen werden mit Hilfe von RANSAC [FB81] oder RANSAC-derivaten ([MC02], [CMK03], [CM05], [FP06]) zur Schätzung der Epipolargeometrie durch den 8-Punkte-Algorithmus genutzt, welche in der F - bzw. E -Matrix resultiert. Aus dieser wird die relative Kamerabewegung extrahiert.

Es existieren jedoch degenerierte Fälle in Bezug auf Kamerabewegung und Merkmalskorrespondenzen, welche die Bewegungsschätzung mathematisch korrekt erscheinen lassen, jedoch nicht die Wirklichkeit wiedergeben. Um diese degenerierten Fällen erkennen zu können und ihnen zu begegnen wurden zwei Verfahren ([Shi06], [TZM95]) kombiniert und erweitert. Die Ergebnisse wurden auf der International Conference on Image Processing 2008 veröffentlicht ([1], siehe Anhang, vlg. Abbildung 28).

8.2 Absolute Pose

Die Rückrechnung der absoluten Pose aus 2-D / 3-D Merkmalskorrespondenzen ist der letzte Schritt in der Posebestimmung und resultiert in einer Kamerapose relativ zu dem verwendeten Weltkoordinatensystem. Als Merkmale für die Rückrechnung eignen sich vor allem Punkte, aber auch Geraden bieten diese Möglichkeit. Im Rahmen des Projektes wurden für beide Primitive Möglichkeiten untersucht, sie für die Modellbasiertes Posetracking zu verwenden.

8.3 Absolute Pose durch Lineare Kalibrierung

Das Verfahren zu Linearen Kalibrierung berechnet aus einer Menge von gegebenen Korrespondenzen zwischen homogenen 2-D Punkten $\tilde{p}_i^i \in \mathbb{IP}^2$ und homogenen 3-D Punkten $\tilde{p}_i^w \in \mathbb{IP}^3$ eine Projektionsmatrix $P \in \mathbb{R}^{3,4}$, $P = (R, t)$ welche die Weltpunkte durch eine Rotation um R , Translation um t und anschließende perspektivische Projektion auf die Bildpunkte abbildet [TV98]:

$$\tilde{p}_i^i = P \tilde{p}_i^w \quad (2)$$

Gleichung 2 lässt sich so umformen, dass die Elemente von P die Lösung eines überbestimmten homogenen linearen Gleichungssystems sind.

$$\begin{aligned} M q &= 0 \\ q &= (P_{1,1}, \dots, P_{3,4})^T \end{aligned} \quad (3)$$

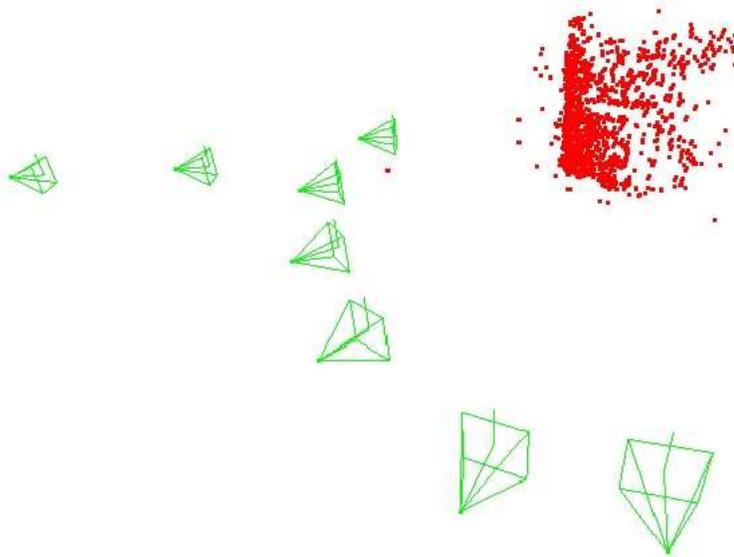


Abbildung 28: Mit dem Verfahren nach [1] rekonstruierte Trajektorie einer Kamerafahrt und 3-D Weltpunkte.

Dieses kann mit Standardverfahren wie der Singulärwertzerlegung gelöst werden. Dabei tritt das Problem auf, dass die resultierende Projektionsmatrix P nur bis auf einen Skalierungsfaktor genau bestimmt werden kann.

Um die *korrekte Skalierung* der Projektionsmatrix zu berechnen, kann man sich die Eigenschaften der enthaltenen Rotationsmatrix zu Nutze machen. Die Determinante einer Rotationsmatrix ist stets 1. Somit lässt sich der Skalierungsfaktor zwischen der berechneten Projektionsmatrix P und der Projektionsmatrix \bar{P} mit der "korrekten" Skalierung (d. h. mit valider Rotationsmatrix und korrekter Translation in Weltkoordinaten) ermitteln. Sei $\det(P_{1:3;1:3}) = \delta$ die Determinante der in P enthaltenen Rotationsmatrix. Da es sich bei der Determinanten um die n -fache Linearform handelt lässt sich \bar{P} durch division durch den gesuchten Skalierungsfaktor berechnen:

$$\bar{P} = \frac{P}{\sqrt[3]{\delta}} \quad (4)$$

Ein weiteres Problem des Verfahrens liegt in den *stark unterschiedlichen Wertebereichen* der Dimensionen des zu berechnenden Nullraums welcher die Projektionsmatrix repräsentiert. Die Werte der Einträge die zur Rotationsmatrix gehören bewegen sich stets im Intervall $[-1 \dots 1]$. Die Werte welche den Translationsvektor formen sind jedoch abhängig von der Skalierung der Welt sowie der Lage der Punkte und der Kamera. Es kann also nicht ausgeschlossen werden, dass diese Werte in absolut anderen Größenordnungen liegen. Dies wird erst zum Problem, wenn Rauschen die Daten verunreinigt. In diesem Fall wird die implizite Minimierung des quadratischen Fehlers durch die Nullraumbestimmung eine prozentual größere Abweichung bei den wesentlich sensibleren Werten der Rotationsmatrix verursachen.

In [Har97] wurde für den 8-Punkte-Algorithmus ein Verfahren vorgeschlagen, welches die Eingangsdaten durch Translation und Skalierung auf einen einheitlichen und numerisch stabileren Bereich abbildet. Dieses Verfahren lässt sich prinzipiell auch auf die Lineare Kalibrierung anwenden: Gleichung 2 wird durch die Matrizen A und U erweitert, welche die Menge der Welt- respektive Bildpunkte so verschieben und skalieren, dass der Schwerpunkt im Ursprung liegt und die Standardabweichung $\sqrt{2}$ beträgt:

$$\begin{aligned} U \tilde{p}_i^i &= (U P A^{-1}) A \tilde{p}_i^w & (5) \\ \sigma(U \tilde{p}_i^i) &= \sqrt{2} \\ \mu(U \tilde{p}_i^i) &= 0 \\ \sigma(A \tilde{p}_i^w) &= \sqrt{2} \\ \mu(A \tilde{p}_i^w) &= 0 \end{aligned}$$

Das Ergebnis der Nullraumbestimmung ist also $(U P A^{-1})$, woraus sich P leicht zurückgewinnen lässt.

Leider behebt dieser Normierungsschritt das Problem nicht, da es sich nicht ausschließlich um eine numerische Instabilität aufgrund der Eingangsdaten handelt wie beim 8-Punkte-Algorithmus (wo die Pixelkoordinaten einen Wertebereich von ca. $[0 \dots 1000]$ hatten), sondern die Wertebereiche der einzelnen Dimensionen des Nullraums sich so stark unterscheiden. Die Lösung mit der minimalen quadratischen Fehlersumme in den Komponenten von P kann also weiterhin zu unbrauchbaren Rotationsmatrizen führen, falls die Translation sich in einem zur Rotationsmatrix stark unterschiedlichen Wertebereich bewegt und die Daten zusätzlich verrauscht sind.

8.4 Alternative Verfahren zur Bestimmung der absoluten Pose aus Punktkorrespondenzen

Adnan Ansar und Kostas Daniilidis beschreiben in [AD02] ein Verfahren zur Poseschätzung, dass entweder mit Punkt- oder mit Geradenkorrespondenzen arbeiten kann. Die zentrale Idee ist eine Linearisierung eines quadratischen Gleichungssystems 6 durch Substitution der quadratischen Terme durch neue Variablen. Für n Variablen x_i bedeutet dies:

$$b_i = \sum_{j=1}^n \sum_{k=j}^n a_{ijk} x_i x_j \quad (i = 1 \dots m) \quad (6)$$

$$0 = \sum_{j=1}^n \sum_{k=j}^n a_{ijk} x_{ij} - b_i \rho \quad (i = 1 \dots m; \rho = 1; x_{ij} = x_i x_j) \quad (7)$$

So entsteht ein überbestimmtes homogenes lineares Gleichungssystem 7, welches direkt gelöst werden kann. Da hier der Nullraum in der Regel eine höhere Dimension als 1 haben wird, werden in einem zweiten Schritt die aus der Substitution bekannten Randbedingungen als lineares Gleichungssystem wieder eingebracht und aufgelöst.

Dies resultiert in einer eindeutigen Lösung für die Tiefenwerte der Bildpunkte, welche die Differenz von jeweils zwei Bildpunkten mit geschätzter Tiefeninformation zu dem tatsächlichen Abstand der korrespondierenden Weltpunkte minimiert (siehe Abbildung 29):

$$\begin{aligned} d_{ij}^2 &= (t_i p_i - t_j p_j)^T (t_i p_i - t_j p_j) \\ &= t_i^2 p_i^T p_i + t_j^2 p_j^T p_j - 2t_i t_j p_i^T p_j \end{aligned} \quad (8)$$

Gleichung 8 hat die gleiche Form wie Gleichung 6 und lässt sich somit mit dem oben beschriebenen Verfahren lösen. Das Zurückrechnen der eigentlichen Pose wird so zu einem 3-D / 3-D relativen Pose-Problem vereinfacht, für welches einfache lineare Lösungsverfahren existieren.

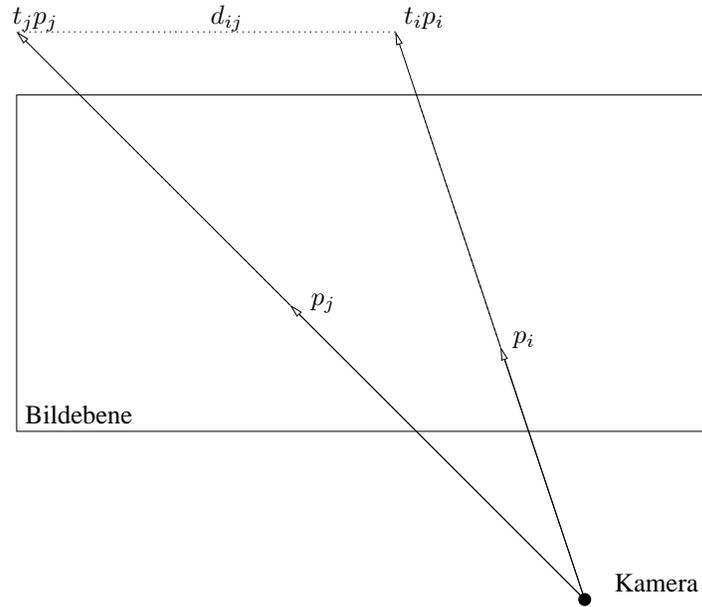


Abbildung 29: Geometrische Bedingung (Gleichung 8) für die Poserekonstruktion aus Punktkorrespondenzen nach [AD02]: Der bekannte Abstand d_{ij} der Weltpunkte muss gleich dem Abstand der Bildpunkte p_i, p_j mit ihren rekonstruierten Tiefen t_i, t_j im Kamerakoordinatensystem sein.

8.5 Vergleich der Rauschsensitivität der Verfahren

Alle betrachteten Verfahren liefern bei exakten korrekten Eingangsdaten auch in jedem Fall die korrekte Pose zurück. In realen Anwendungsfällen werden die Eingangsdaten durch ungenaue Lokalisation der Merkmale vor allem im Bild verrauscht sein. Um die Sensitivität der Verfahren in Bezug auf Rauschen in der Position der Bildmerkmale zu analysieren wurden umfangreiche Testserien mit zufällig generierten Daten durchgeführt. Als Fehlermaße werden der prozentuale Fehler in der Translation Δ_t , sowie die Winkelabweichung der Drehachse Δ_{Axis} und des Drehwinkels Δ_{Angle} in Grad verwendet.

Für jedes mittlere Rauschen im Bereich von 1 bis 10 Pixel wurden je 25 Ergebnisse der Verfahren auf zufälligen Daten gemittelt. Abbildung 31 im Anhang B zeigt die Ergebnisse. Danach bringt der Normalisierungsschritt bei der Linearen Kalibrierung nur marginale Verbesserungen, das Verfahren

nach Ansar [AD02] scheint vor allem bei der Schätzung der Rotationsachsen und des Rotationswinkels weniger rauschanfällig als die Lineare Kalibrierung zu sein.

8.6 Absolute Pose aus Linien

Die Bestimmung der Pose aus korrespondierenden Linien (Geraden) zwischen Bild und Modell ist ein in der Literatur deutlich weniger behandeltes Thema als die Pose aus Punktkorrespondenzen. Doignon versucht ein Lineares Gleichungssystem aufzustellen, indem er die Projektion von Linien mittels Projektions- und Plückermatrizen beschreibt [Doi07], leider ist mathematische Herleitung sehr kurz gehalten und so schwer nachvollziehbar. Das Verfahren von Ansar [AD02] ist im Prinzip auch für Linien geeignet und wird gerade behandelt. Der Ansatz von Liu et al. [LHF90] wurde implementiert und hat eine Einschränkung in den maximalen Eulerwinkeln bis zu welchen das Verfahren funktioniert. Es eignet sich also eventuell als Verfahren das im Tracking Schritt verwendet werden kann, im Allgemeinen jedoch nicht für die Initialisierung einer Pose.

9 Matching

Bei der Korrespondenzsuche von bekannten Merkmalen zu Merkmalen die aus dem Bild extrahiert wurden ist es sinnvoll, über eine einfache Ähnlichkeit der einzelnen Merkmale hinausgehend auch die *Lage der Merkmale zueinander* miteinzubeziehen. Dieses zusätzlich mit eingebrachtes Wissen verringert den Suchraum und erhöht die Zuverlässigkeit der Ergebnisse. Die folgende Arbeit entstand im Rahmen des Arbeitspakets „TWB-Wissensbasiertes Tracking über Graphmatching“ des ursprünglichen Antrags.

9.1 Wissensbas. Graphmatching von Segmentierungsergebnissen

Eine Zuordnung zwischen einem vorhandenen Modell und einem eingehenden Signal in Form eines Bildes zur Posebestimmung kann auf mehreren unterschiedlichen Schichten der Signalverarbeitung erfolgen, unter anderem auch nah an der Modellschicht. Ein Ansatz hierzu ist es, aus dem Signal elementare Primitive zu extrahieren und diese miteinander in Relation zu setzen, um darin höhere Primitive wieder zu finden, wie sie im Modell vorkommen.

Dieses Problem lässt sich auf ein Graphmatching abbilden, wobei die extrahierten Primitive die Knoten eines Graphen bilden. Attributierte Kanten, welche die Relationen repräsentieren, verbinden die Knoten. Das Problem der Identifikation eines bekannten Modells oder Modellteils in diesem Graphen kann als Suche nach einer Subgraphisomorphie angesehen werden, wobei das gesuchte Modell ein Subgraph des aus dem Signal generierten Datengraphen darstellt.

Im folgenden wird die Notation von Tittmann [Tit03] verwendet. Sei $G = (V, E)$ ein Graph mit einer Knotenmenge V und einer Kantenmenge E . Eine gerichtete Kante $e = (u, v)$ hat den Knoten u als direkten Vorgänger, und v als direkten Nachfolger. Zwei Graphen G_m und G_d sind *isomorph* zueinander, wenn eine bijektive Abbildung

$$M : V_m \rightarrow V_d \tag{9}$$

existiert, mit

$$(u, v) \in E_m \Leftrightarrow (M(u), M(v)) \in E_d. \tag{10}$$

Von *Subgraphisomorphie* spricht man im Fall einer injektiven Abbildung:

$$M : V_m \rightarrow V_d \quad (11)$$

mit

$$(u, v) \in E_m \Rightarrow (M(u), M(v)) \in E_d. \quad (12)$$

Sei also für das gegebene Szenario G_m ein Modellgraph. Dieser besteht aus attribuierten Knoten V_m welche stabil extrahierbare niedere Primitive (vgl. Kapitel 3) repräsentieren. Die Attribute des Knotens geben hierbei Auskunft über die Art des Primitivs. Die attribuierten Kanten E_m modellieren die Relationen zwischen den Primitiven. G_d ist ein zweiter, aus einem Bild automatisch generierter Datengraph.

Als Relationen wurden topologische Zusammenhänge sowie Umschließungsrelationen gewählt. Außerdem wurde eine Relation vorgesehen um Verbindungen zwischen Knoten zu kennzeichnen, denen durch ihren Abstand im Bild kein inhaltlicher Zusammenhang zugesprochen wird.

Ist G_M ein Subgraph von G_d , so wurde das Modell verifiziert, d. h. ein höheres Primitiv gefunden. Die Position und Orientierung des höheren Primitivs lässt sich direkt aus den Komponenten aus denen es besteht ableiten.

Zur Validierung des Ansatzes wurde im Rahmen einer Diplomarbeit [7] ein System entwickelt, welches handgezeichnete komplexe Objekte erkennt. Diese bestehen aus Primitiven wie Rechtecken, Dreiecken und Kreisen, welche sich durch Segmentierung aus dem Bild gewinnen lassen. Zwischen den Primitiven werden die o. g. topologischen Zusammenhänge instanziiert und so der Datengraph generiert. In diesem wird nach Subgraphisomorphien mit den zuvor festgelegten Modellgraphen gesucht, welche komplexe Objekte repräsentieren: Zwei Kreise nebeneinander, die sich unterhalb eines Rechtecks befinden, bilden beispielsweise ein schematisches Auto welches erkannt werden kann. Zur Berechnung der Subgraphisomorphien wurde der VF Graph Matching Algorithmus nach Cordella et al [CFSV04] verwendet.

10 Laufende Qualifikationsarbeiten

Drei weitere Qualifikationsarbeiten wurden im Rahmen dieses Projektes angemeldet und werden noch vor Ende des Jahres fertiggestellt:

Modellbasierte Posebestimmung aus 2-D / 3-D SIFT-Korrespondenzen

Ziel der Arbeit ist es, bei einem gegebenen Modell in Form von einer Menge von SIFT-Merkmalen mit bekannter 3-D-Position und einer Menge aus einem Kamerabild extrahierten SIFT-Merkmale die Position der Kamera zurückzurechnen.

Umgesetzt werden sollen folgende Schritte:

- Erzeugen einer Menge von SIFT-Merkmalen mit bekannter 3-D Position als Modell, z.B. aus einem Stereoalgorithmus, aus Bildern mit bekannter Kamerapose oder synthetischen Daten.
- Zuordnung von im Bild gefundenen SIFT-Merkmalen zu denen aus dem Modell. Ein besonderer Schwerpunkt liegt hier auf dem verwendeten Ähnlichkeitsmaß (vgl. Kapitel 3.5) und ggf. einer effektiven Datenstruktur zur nächsten Nachbar Suche im Modell.

- Rückrechnung der Pose aus den 2-D / 3-D Punktkorrespondenzen. (Hier soll auf Ergebnisse aus Kapitel 8.2 zurückgegriffen werden)

Objekterkennung durch Hough-Transform-Clustering von SURF-Features

Ziel der Arbeit ist eine Objekterkennung mit SURF-Features und Farbinformation.

Die Zuordnung der aus einem Bild extrahierten Feature-Matches zu den gelernten Objekten soll durch Hough-Transform-Clustering umgesetzt werden. Die gematchten Feature-Vektoren führen so zu einer Bewertung der Objekte aus der Datenbank aufgrund von globalen Eigenschaften. Es sollen gleichzeitig Translation, Rotation, Skalierung und Farbe berücksichtigt werden, um falsche Matches auszusortieren und damit das Ergebnis der Objekterkennung zu verbessern. Die Herausforderung bei der Implementierung des neuen Verfahrens besteht darin, die rechenintensive Hough-Transformation für die Aufgabe angemessen umzusetzen und zu parametrisieren, dabei gleichzeitig ein robustes Verfahren zur Clusterbildung und Maximadetektion im Houghraum zu entwickeln.

Zwar ist das Anwendungsgebiet nicht die Outdoor-Objekterkennung von Gebäuden, wir erwarten dennoch nützliche Erkenntnisse für die Pose-Schätzung.

Implementation des eines generellen RANSAC Verfahrens

Ziel der Arbeit ist die Implementation eines Generellen RANSAC Verfahrens, welche sich für verschiedenste Probleme, im speziellen aus dem Bereich Struktur aus Bewegung, einsetzen lässt.

Das RANSAC-Verfahren [FB81] ist ein Etabliertes Verfahren in der Computer Vision, um mit Ausreißer in Messdaten umzugehen. Es wurde in den letzten Jahren regelmäßig Vorschläge zur Verbesserung und Beschleunigung des Verfahrens gemacht. Unter anderem QDEGSAC [FP06], welches sich speziell der Problematik von quasi-degenerierten Konfigurationen annimmt. Dabei ist keinerlei Wissen über das zu schätzende Modell nötig. Dieses Verfahren soll in C++ so allgemein umgesetzt werden, dass die Algorithmen zur Modellberechnung frei austauschbar sind.

10.1 Pose-Schätzung

Im Bereich der Pose-Schätzung sollen die Verfahren zur Posebestimmung aus Linien weiter evaluiert und mit realen Daten getestet werden. Hierzu wird das Verfahren nach Ansar [AD02] für Linien implementiert und mit dem Verfahren nach Liu [LHF90], für welches bereits eine Implementation vorliegt, verglichen.

11 Gemeinsame Arbeiten

12 Modell Schemata - Anforderungen an Sprache, Modell, Werkzeuge

Um eine geeignete Repräsentation von 3-D-Umgebungsmodellen zu gewährleisten, wurde in enger Zusammenarbeit mit der Arbeitsgruppe Ebert (s.u.) eine Liste von Anforderungen aufgestellt, die von

dem Modell, der Sprache in der es repräsentiert wird und den Werkzeugen die zur Erstellung, Visualisierung und Anfrage genutzt werden erfüllt werden müssen. Als zentrale Punkte wurden erachtet:

Die Sprache in der das Modell repräsentiert wird muss:

- geeignet zur Beschreibung von Umgebungsmodellen sein
- ein offenes Format sein *und*
- gut dokumentiert sein *und*
- Möglichkeiten zur Erweiterung bieten, damit eigene Verfahren zur Verarbeitung der Modell entwickelt werden können

Das Modell selber muss:

- Geometrie und Semantik verknüpfen, da nur so unser Ansatz des semantischen Posetrackings möglich ist
- einen Bezug zum Bild aufweisen
- objektorientiert aufgebaut sein⁵ damit einzelne semantische Objekte als eine gemeinsame Entität repräsentiert werden können
- renderbar sein *und*
- für “semantisches rendering” geeignet sein, d.h. die geometrischen Primitive und die Semantik müssen in einer Form abgelegt sein, die Anfragen an Optik und Semantik in Echtzeit ermöglichen
- mit vertretbarem Aufwand erstellbar sein

Entsprechend müssen die Werkzeuge zur Modellerstellung und -verarbeitung

- die Möglichkeit zum Import/Export der gewünschten Modellformate bieten
- einfach bedienbar sein
- gut dokumentiert sein
- mit vertretbaren Kosten zu beschaffen sein

Als mögliche Modellierungssprachen wurden in Betracht gezogen: CityGML (City Geographic Markup Language [Kol08, OGC08, KB06, UK08, KS07]), COLLADA (Collaborative Design Activity [Com]), IFC (Industry Foundation Classes [Lie06]), sowie KML (Keyhole Markup Language [OGC, Goo08]).

Den Anforderungen an die Sprache wurden von CityGML am besten erfüllt. Als einfach zu erlernendes und mächtiges Werkzeug stellte sich Google SketchUp heraus, welches einen Export in verschiedene Formate (jedoch nicht CityGML) bietet. Es wurde beschlossen beide Alternativen parallel weiter zu verfolgen (vgl. Kapitel 7.1 zur semiautomatischen Modellierung und Kapitel 2.2 zur Modellierung des Campus-Modells) mit der Hoffnung, dass durch die zunehmend stärkere Verbreitung und Nutzung des CityGML Standards auch mehr Werkzeuge zur Konvertierung zwischen den Formaten entstehen.

Die Forschungsarbeiten auf dem Gebiet der Modelle und Modellschemata fanden in enger Kooperation mit der Arbeitsgruppe Ebert statt, eine ausführlichere Gegenüberstellung der Ansätze und Besprechung der Formate wird in dem Bericht des DFG-Projekts EB 119/3-1 zu lesen sein.

⁵Referenzierbarkeit, hierarchische Strukturen, Typisierung

13 Automatisierung der Modellgenerierung durch Matching markanter Eckpunkte

Die in Kapitel 7.1 angesprochenen Verfahren zur semiautomatischen Modellgenerierung wurden mit den Algorithmen zur Extraktion markanter Eckpunkte aus Blueprints (Kapitel 4.4, 5.1) kombiniert um die Modellgenerierung weiter zu automatisieren. Im Speziellen bedeutet das, dass die manuelle Selektion der für die Geometrie relevanten Punkte durch die automatische Extraktion der markanten Eckpunkte aus Blueprints ersetzt werden sollte.

Gleichzeitig verspricht dieser Ansatz, dass auch Teile der Semantik für die Rekonstruktion automatisch erzeugt werden können: Von den Eckpunkten ist beispielsweise bekannt, ob sie an den Himmel grenzen (also Vermutlich zum Dach eines Gebäudes gehören). Weiterhin ist vorstellbar, dass markante Eckpunkte die sich um ein Element einer SIFT-Markosemantik gruppieren die Eckpunkte eines Fensters beschreiben (vgl. Kapitel 4.1).

Eine neue Herausforderungen bei diesem Ansatz ist das Matching zwischen markanten Eckpunkten zwischen mehreren Bildern um eine Rekonstruktion zu ermöglichen. Hierzu wurden als erstes verschiedene Deskriptoren an den Punkten erzeugt und zunächst ohne Betrachtung der Geometrie versucht in einem Bildpaar Korrespondenzen zu finden.

Als intensitätsbasiertes Abstandsmaß zwischen zwei rechteckigen Regionen um die Punkte wurde die Normalisierte Kreuzkorrelation evaluiert, als Wavelet basierter Desriptor SURF[BTVG06].

Die durch die normalisierte Kreuzkorrelation bestimmte Ähnlichkeit S zwischen zwei gleichgroßen rechteckigen Regionen R_1 und R_2 definiert sich als

$$S(R_1, R_2) = \frac{\sum_{(i,j)} (R_1(i, j) - \mu(R_1))(R_2(i, j) - \mu(R_2))}{\sqrt{\sum_{(i,j)} (R_1(i, j) - \mu(R_1))^2} \sqrt{\sum_{(i,j)} (R_2(i, j) - \mu(R_2))^2}}$$

mit $\mu(R)$ als dem Mittelwert der Region und i, j Laufvariablen über die Pixel der Regionen. Die resultierende Ähnlichkeit $S \in [-1 \dots 1]$ gibt Auskunft über die Korrelation der beiden Regionen. 1 bedeutet vollständige Korrelation, 0 keine Korrelation und -1 anti-Korrelation.

In ersten Versuchen hat sich dieses intensitätsbasierte Ähnlichkeitsmaß als nicht destinktiv genug und damit für die Anwendung als nicht geeignet erwiesen, wesshalb SURF-Deskriptoren als Alternative betrachtet wurden. An Korrespondenzen von SURF-Deskriptoren wurde neben einem gegenseitig minimalen Deskriptorabstand die Anforderung gestellt, dass der Distance Ratio Threshold von $r = 0.8$ wie in [Low04] vorgeschlagen unterschritten werden muss.

Diese rein auf den Deskriptoren aufbauende Korrespondenzsuche wirkt vielversprechend, reicht aber alleine nicht aus. In weiteren Schritten sollen geometrische Validierungen herangezogen werden, um Ausreißer aus den Korrespondenzmengen zu entfernen. Neben offensichtlichen Herangehensweisen wie der Konsistenzbedingung durch die Epipolargeometrie sind weitere geometrische Randbedingung durch die Extraktionsverfahren der markanten Eckpunkte gegeben, die mit einbezogen werden können. Unter anderem die Reihenfolge, in welcher markante Punkte auf einer Fluchtlinie vorkommen, außerdem die Information zu welchen beiden Fluchtpunkten die sich im Punkt schneidenden Geraden fliehen.

Probleme des Ansatzes liegen im Moment noch an der noch nicht genügend robusten Extraktion der markanten Eckpunkte, sowie dem zu stark einschränkenden Distance Ratio Threshold bzw. des noch nicht stabil genügenden Matchings aufgrund der SURF-Deskriptoren. Vielleicht ist es hier nötig, die Geometrie und Deskriptoren *gleichzeitig* zu betrachten und nicht in zwei getrennten aufeinanderfolgenden Schritten.



Abbildung 30: Positionen an denen markante Eckpunkte gefunden wurden und die auf SURF-Deskriptoren basierenden Korrelationen.

14 Glossar der Bildverarbeitung

In enger Zusammenarbeit mit der Arbeitsgruppe Ebert der Universität Koblenz-Landau wurde ein Glossar definiert, das für elementare Objekte und Algorithmen der Bildverarbeitung eine gemeinsame Sprache und von allen Beteiligten akzeptierte mathematische Definitionen und Notationen festgelegt.

Die Einigung auf gemeinsame Sprache und gemeinsame Definitionen zwischen allen beteiligten Arbeitsgruppen vereinfacht die Festlegung von Schnittstellen zwischen den entworfenen Programmmodulen und verhindert Missverständnisse in der Kommunikation.

A Technische Umsetzungsdetails der automatischen Erzeugung von CG-Modellen

Es wurde auf folgende Bibliotheken zurückgegriffen die bereits Teilprobleme ausreichend gut lösen:

Triangle: 2-D Triangulierung⁶

GNU Triangulated Surface Library: Mesh Ausdünnung⁷

OpenCV: Rektifikation, Tiefenkarte aus Stereobildpaaren⁸

Das Stereosystem wurde aus zwei baugleichen Firewire Kameras vom Typ *Guppy F-146C* mit 6.0 mm Objektiven auf *Manfrotto MA 482 Micro Kugelköpfen* aufgebaut. Die Kugelköpfe wurden variabel auf

⁶<http://www.cs.cmu.edu/~quake/triangle.html>

⁷<http://gts.sourceforge.net/>

⁸<http://sourceforge.net/projects/opencvlibrary/>

einer Aluminiumleiste angebracht, um schnell die Stereobasis und Ausrichtung der Kameras ändern zu können.

B Rauschverhalten der untersuchten Verfahren zur Posebestimmung

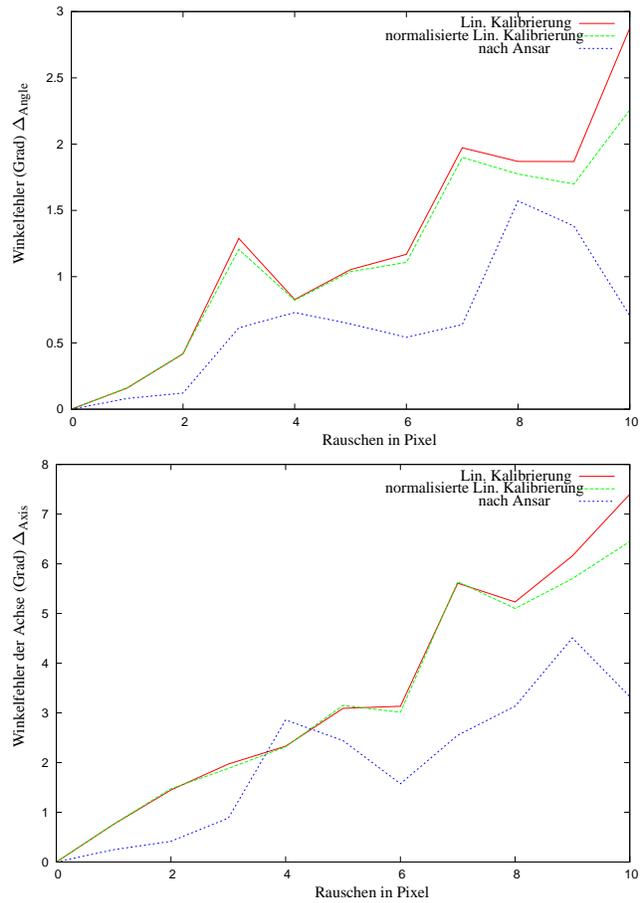


Abbildung 31: Rauschverhalten der Verfahren zur Bestimmung der absoluten Pose (Kapitel 8.2) aus Punktkorrespondenzen.

Literaturangaben

- [AD02] ANSAR, ADNAN und KOSTAS DANIILIDIS: *Linear Pose Estimation from Points or Lines*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25:282–296, 2002.
- [ADV03] ALMANSA, ANDRÉS, AGNÈS DESOLNEUX und SÉBASTIEN VAMECH: *Vanishing Point Detection without Any A Priori Information*. IEEE Trans. Pattern Anal. Mach. Intell., 25(4):502–507, 2003.
- [BT98] BIRCHFIELD, STAN und CARLO TOMASI: *Depth Discontinuities by Pixel-to-Pixel Stereo*. In: *ICCV*, Seiten 1073–1080, 1998.
- [BTVG06] BAY, HERBERT, TINNE TUYTELAARS und LUC VAN GOOL: *SURF: Speeded Up Robust Features*. ECCV, Seiten 404–417, 2006.
- [Can86] CANNY, J.: *A computational approach to edge detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, November 1986.
- [CFSV04] CORDELLA, L.P., P. FOGGIA, C. SANSONE und M. VENTO: *A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs*. IEEE TPAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence, 26, 2004.
- [CM05] CHUM, O. und J. MATAS: *Matching with PROSAC – Progressive Sample Consensus*. In: SCHMID, CORDELLA, STEFANO SOATTO und CARLO TOMASI (Herausgeber): *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Band 1, Seiten 220–226, Los Alamitos, USA, 6 2005. IEEE Computer Society.
- [CMK03] CHUM, O., J. MATAS und J. KITTLER: *Locally optimized RANSAC*. In: G. GOOS, J. HARTMANIS, J. VAN LEEUWEN (Herausgeber): *DAGM 2003: Proceedings of the 25th DAGM Symposium*, Nummer 2781 in *Lecture Notes in Computer Science (LNCS)*, Seiten 236–243, Heidelberger Platz 3, 14197, Berlin, Germany, 9 2003. Springer-Verlag, <http://www.springer.com>.
- [Com] COMMUNITY, COLLADA: *COLLADA Wiki*. Wiki.
- [Doi07] DOIGNON, CHRISTOPHE: *Scene Reconstruction, Pose Estimation and Tracking*, Kapitel Chapitre 20: An Introduction to Model-Based Pose E. Vision Systems. published by the International Journal of Advanced Robotic Systems, édité by Rustam Stolkin Auflage, 2007.
- [FB81] FISCHLER, MARTIN A. und ROBERT C. BOLLES: *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. Communications of the ACM, 24(6):381–395, 1981.
- [FE09] FALKOWSKI, KERSTIN und JÜRGEN EBERT: *Graph-based urban object model processing*. In: *City Models, Roads and Traffic (CMRT): Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms and Evaluation - Accepted for publication*. International Society for Photogrammetry and Remote Sensing (ISPRS), 2009.
- [For09] FORSCHUNGSZENTRUM KARLSRUHE.: *IFCE Explorer for CityGML*: <http://www.iai.fzk.de/www-extern/index.php?id=1570>, 02 2009.
- [FP06] FRAHM, JAN-MICHAEL und MARC POLLEFEYS: *RANSAC for (Quasi-)Degenerate data (QDEGSAC)*. In: *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seiten 453–460, Washington, DC, USA, 2006. IEEE Computer Society.
- [FTV97] FUSIELLO, A., E. TRUCCO und A. VERRI: *Rectification with unconstrained stereo geometry*. In: *British Machine Vision Conference*, Seiten 400–409, 1997.
- [GKCN08] GROEGER, GERHARD, THOMAS H. KOLBE, ANGELA CZERWINSKI und CLAUS NAGEL: *OpenGIS City Geography Markup Language (CityGML) Encoding Standard*. Technischer Bericht 1.0.0, Open Geospatial Consortium Inc., 8 2008.
- [Goo08] GOOGLE: *KML*. Description of KML, 2008.
- [Har79] HARALICK, R.M.: *Statistical and structural approaches to texture*. Proceedings of the IEEE, 67(5):786–804, May 1979.
- [Har97] HARTLEY, RICHARD I.: *In Defense of the Eight-Point Algorithm*. Pattern Analysis and Machine Intelligence, 19(6):580–593, 6 1997.
- [HP76] HOROWITZ, STEVEN L. und THEODOSIOS PAVLIDIS: *Picture Segmentation by a Tree Traversal Algorithm*. J. ACM, 23(2):368–388, 1976.
- [HS88] HARRIS, C. und M. STEPHENS: *A combined corner and edge detector*. In: *4th Alvey Vision Conference*, Seiten 147–151, 1988.
- [IJ07] IGNATIUS, CIO, UBIRAT, A. und CLAUDIO R. JUNG: *Block-based image inpainting in the wavelet domain*. Vis. Comput., 23(9):733–741, 2007.

- [KB06] KOLBE, THOMAS und SAM BACHARACH: *CityGML: An Open Standard for 3D City Models*. Description of CityGML, 6 2006.
- [Kol08] KOLBE, THOMAS: *CityGML*. Virtual 3D City Models, 6 2008.
- [KS07] KOLBE, THOMAS und ALEXANDRA STADLER: *CityGML für die 3D-Navigation*. Institut für Geodäsie und Geoinformation Technische Universität Berlin, 9 2007.
- [LB87] LEAVERS, V. F. und J. F. BOYCE: *The Radon transform and its application to shape parametrization in machine vision*. *Image Vision Comput.*, 5(2):161–166, 1987.
- [LHF90] LIU, Y., T.S. HUANG und O.D. FAUGERAS: *Determination of Camera Location from 2- D to 3-D Line and Point Correspondences*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, 1 1990.
- [Lie06] LIEBICH, THOMAS: *IFC Industry Foundation Classes*. IFC2x Edition 3, 2006.
- [Low99] LOWE, DAVID G.: *Object Recognition from Local Scale-Invariant Features*. In: *Proc. of the International Conference on Computer Vision ICCV, Corfu*, Seiten 1150–1157, 1999.
- [Low04] LOWE, DAVID G.: *Distinctive Image Features from Scale-Invariant Keypoints*. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [LT98] LINDSTROM, P. und G. TURK: *Fast and memory efficient polygonal simplification*. In: *Visualization '98. Proceedings*, Seiten 279–286, 1998.
- [M. 76] M. KUWAHARA, K. HACHIMURA, S. EIHO, AND M. KINOSHITA: *Processing of ri-angiocardigraphic images*. In: PRESTON, K. und M. ONOE (Herausgeber): *Digital Processing of Biomedical Images*, Seiten 187–202, 1976.
- [MAW⁺07] MERRELL, P., A. AKBARZADEH, L. WANG, P. MORDOHAI, J.M. FRAHM, R. YANG, DAVID NISTÉR und M. POLLEFEYS: *Real-Time Visibility-Based Fusion of Depth Maps*. In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Seiten 1–8, 2007.
- [MC02] MATAS, J. und O. CHUM: *Randomized RANSAC*. In: WILDENAUER, H. und W. KROPATSCH (Herausgeber): *Proceedings of the Computer Vision Winter Workshop '02*, Seiten 49–58, Wien, Austria, 2 2002.
- [NfnaodS01] NISTÉR, DAVID und INSTITUTIONEN) FÖR NUMERISK ANALYS OCH DATALOGI (STOCKHOLM: *Automatic dense reconstruction from uncalibrated video sequences*. Tekniska högsk., 2001.
- [Nüc06] NÜCHTER, ANDREAS: *Semantische dreidimensionale Karten für autonome mobile Roboter*. Doktorarbeit, Rheinische Friedrich-Wilhelms-Universität Bonn, <http://www.uni-bonn.de/>, 2006.
- [OGC] OGC: *KML Keyhole Markup Language*. KML Overview.
- [OGC08] OGC: *CityGML*. Request for Comment on CityGML Standard, 4 2008.
- [Ots79] OTSU, N.: *A threshold selection method from gray level histograms*. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, März 1979. minimize inter class variance.
- [PKG99] POLLEFEYS, MARC, REINHARD KOCH und LUC J. VAN GOOL: *A Simple and Efficient Rectification Method for General Motion*. In: *ICCV*, Seiten 496–501, 1999.
- [RL01] RUSINKIEWICZ, SZYMON und MARC LEVOY: *Efficient Variants of the ICP Algorithm*. In: *3rd International Conference on 3D Digital Imaging and Modeling (3DIM)*, Seiten 145–152, Quebec City, Canada, 5 2001. IEEE Computer Society.
- [Rot00] ROTHER, CARSTEN: *A new approach for vanishing point detection in architectural environments*. In: *In Proc. 11th British Machine Vision Conference*, Seiten 382–391, 2000.
- [Shi06] SHIMSHONI, ILAN: *Balanced Exploration and Exploitation Model Search for Efficient Epipolar Geometry Estimations (BEEM)*. In: *9th European Conference on Computer Vision (ECCV 2006)*, Band II, Seiten 151–164, 5 2006. BEEM, Code-Demo available at Shimshoni's Webpage.
- [SLC06] SEO, KYUNG-SEOK, JUNG-HWA LEE und HEUNG-MOON CHOI: *An efficient detection of vanishing points using inverted coordinates image space*. *Pattern Recogn. Lett.*, 27(2):102–108, 2006.
- [Tit03] TITTMANN, PETER: *Graphentheorie: Eine anwendungsorientierte Einführung*. Fachbuchverlag Leipzig im Carl Hanser Verlag, <http://www.fachbuch-leipzig.hanser.de>, 2003.
- [TV98] TRUCCO, E. und A. VERRI: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, New York, 1998.
- [TZM95] TORR, P. H. S., A. ZISSERMAN und S. J. MAYBANK: *Robust detection of degenerate configurations for the fundamental matrix*. *Fifth International Conference on Computer Vision*, Seiten 1037–1042, 1995.
- [UK08] UMBACH, EBERHARD und ALEXANDER KURZ: *CityGML - City Geography Markup Language*. CityGML Wiki, 6 2008.

- [WDF08] WENZEL, S., M. DRAUSCHKE und W. FÖRSTNER: *Detection of repeated structures in facade images*. Pattern Recognition and Image Analysis, 18(3):406–411, 2008.
- [ZHD01] ZHANG, BIN, MEICHUN HSU und UMESHWAR DAYAL: *K-Harmonic Means - A Spatial Clustering Algorithm with Boosting*. In: *TSDM '00*, Band 2007/2001 der Reihe *Lecture Notes in Computer Science*, Seiten 31–45, London, UK, 2001. Springer-Verlag.

Eigene Veröffentlichungen

- [1] DECKER, PETER, DIETRICH PAULUS und TOBIAS FELDMANN: *Dealing with degeneracy in essential matrix estimation*. In: *Proceedings of the International Conference on Image Processing (ICIP 2008)*, Seiten 1964–1967, San Diego, California, U.S.A., 10 2008. IEEE Signal Processing Society, IEEE Press.
- [2] FALKOWSKI, KERSTIN, EBERT JÜRGEN, DECKER PETER, WIRTZ STEFAN und PAULUS DIETRICH: *Semi-automatic generation of full CityGML models from images*. In: *Geoinformatik 2009*, Band 35, Seiten 101–110, Münster, 4 2009. Institut für Geoinformatik Westfälische Wilhelms-Universität.
- [3] HERING, NILS: *Bildverstehen mittels selbstähnlichen SIFT-Features*. Diplomarbeit, Universität Koblenz-Landau, 2008.
- [4] HERING, NILS, FRANK SCHMITT und LUTZ PRIESE: *Image Understanding Using Self-similar SIFT features*. In: *Fourth International Conference on Computer Vision Theory and Applications (VISAPP) 2009, Lisboa, Portugal*, Band 2, Seiten 114–119, 2009.
- [5] LAKMANN, RAIMUND: *Statistische Modellierung von Farbtexturen*. Doktorarbeit, Universität Koblenz-Landau, 1998. Fölbach Verlag, Koblenz.
- [6] LAKMANN, RAIMUND und LUTZ PRIESE: *Klassifikation von Farbtexturen mit Farbkovarianzmerkmalen*. In: *Proceedings 3. Workshop Farbbildverarbeitung, Erlangen, 25.-26. September 1997*, Seiten 17–23, 9 1997.
- [7] LATSCH, CHRISTIAN: *Erkennung zusammengesetzter Zeichen zur Interaktion auf einem handelsüblichen Whiteboard*. Diplomarbeit, Universität Koblenz-Landau, Campus Koblenz, Fachbereich 4 Informatik, Institut für Computervisualistik, 2009.
- [8] MENGELKOCH, MARCO: *Texturierte 3-D-Mesh Generierung aus Stereobildsequenzen*. Diplomarbeit, Universität Koblenz-Landau, Campus Koblenz, Fachbereich 4 Informatik, Institut für Computervisualistik, 10 2008.
- [9] PRIESE, LUTZ und FRANK SCHMITT: *On Blueprint Reconstruction*. Submitted to: The 2009 International Workshop on 3-D Digital Imaging and Modeling, Kyoto, Japan, 2009.
- [10] PRIESE, LUTZ, FRANK SCHMITT und NILS HERING: *Grouping of Semantically Similar Image Positions*. In: SALBERG, ARNT-BORRE, JON YNGVE HARDEBERG und ROBERT JENSSEN (Herausgeber): *16th Scandinavian Conference, SCIA 2009, Oslo, Norway, June 15-18, Proceedings*, Band 5575 der Reihe *Lecture Notes in Computer Science*, Seiten 726–734, 2009.
- [11] REHRMANN, VOLKER und LUTZ PRIESE: *Fast and Robust Segmentation of Natural Color Scenes*. In: CHIN, ROLAND T. und TING-CHUEN PONG (Herausgeber): *3rd Asian Conference on Computer Vision (ACCV'98)*, Nummer 1351 in *LNCS*, Seiten 598–606. Springer Verlag, 1998.
- [12] SCHMITT, FRANK und LUTZ PRIESE: *Himmelsdetektion in CSC-segmentierten Farbbildern*. In: HELLING, STEPHAN, JOHANNES BRAUERS, BERNHARD HILL und TIL AACH (Herausgeber): *14. Workshop Farbbildverarbeitung*, Seiten 77–84. Shaker Verlag, 2008.
- [13] SCHMITT, FRANK und LUTZ PRIESE: *Intersection Point Topology for Vanishing Point Detection*. In: *Accepted for publication in: Discrete Geometry for Computer Imagery 2009, Montreal, Canada, 2009*.
- [14] SCHMITT, FRANK und LUTZ PRIESE: *Sky detection in CSC-segmented color images*. In: *Fourth International Conference on Computer Vision Theory and Applications (VISAPP) 2009, Lisboa, Portugal*, Band 2, Seiten 101–106, 2009.
- [15] VETTER, SEBASTIAN: *Object Removal from Still Images Employing Inpainting Techniques*. Diplomarbeit, Universität Koblenz-Landau, www.uni-koblenz.de, 2009.

Patente und Erfindungsmeldungen

- [P4] Darstellung eines Objekts mittels einer Durchstrahlung sowie Rekonstruktion unter Verwendung von simulierten Durchstrahlungsdaten (angemeldet in EU, USA, China) WO 02/31767 A2 Anmelder Fraunhofer-Gesellschaft zur Förde-

rung der angewandten Forschung Randolph Hanke; Stefan Schröpfer; Heinz Gerhäuser; Dietrich Paulus; Anmeldedatum 11.10.2001 ; Anmelde­nummer 2062002 ; Veröffentlichungsdatum 22.04.2002

[P5] Optische Erfassung von Freiformflächen; (DP 197 39 250.4-52) Anmelder Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V., 80636 München; Spinnler, Klaus; Paulus, Dietrich; Lang, Peter; Wagner, Thomas; Bauer, Norbert; Schramm, Ullrich; Anmeldedatum 08.09.1997, Veröffentlichungsdatum 02.01.2003

[P6] Verfahren zur Untersuchung des Augenhintergrundes; (Deutschland, DE000010217858C1, International AU002003221527A8); Anmelder: Heidelberg Engineering Georg Michelson; Paulus, Dietrich;

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik

(<http://www.uni-koblenz.de/fb4/publikationen/arbeitsberichte>)

Dietrich Paulus, Lutz Prieße, Peter Decker, Frank Schmitt, Pose-Tracking Forschungsbericht, Arbeitsberichte aus dem Fachbereich Informatik 17/2009

Andreas Fuhr, Tassilo Horn, Andreas Winter, Model-Driven Software Migration Extending SOMA, Arbeitsberichte aus dem Fachbereich Informatik 16/2009

Eckhard Großmann, Sascha Strauß, Tassilo Horn, Volker Riediger, Abbildung von grUML nach XSD soamig, Arbeitsberichte aus dem Fachbereich Informatik 15/2009

Kerstin Falkowski, Jürgen Ebert, The STOR Component System Interim Report, Arbeitsberichte aus dem Fachbereich Informatik 14/2009

Sebastian Magnus, Markus Maron, An Empirical Study to Evaluate the Location of Advertisement Panels by Using a Mobile Marketing Tool, Arbeitsberichte aus dem Fachbereich Informatik 13/2009

Sebastian Magnus, Markus Maron, Konzept einer Public Key Infrastruktur in iCity, Arbeitsberichte aus dem Fachbereich Informatik 12/2009

Sebastian Magnus, Markus Maron, A Public Key Infrastructure in Ambient Information and Transaction Systems, Arbeitsberichte aus dem Fachbereich Informatik 11/2009

Ammar Mohammed, Ulrich Furbach, Multi-agent systems: Modeling and Virification using Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 10/2009

Andreas Sprotte, Performance Measurement auf der Basis von Kennzahlen aus betrieblichen Anwendungssystemen: Entwurf eines kennzahlengestützten Informationssystems für einen Logistikdienstleister, Arbeitsberichte aus dem Fachbereich Informatik 9/2009

Gwendolin Garbe, Tobias Hausen, Process Commodities: Entwicklung eines Reifegradmodells als Basis für Outsourcingentscheidungen, Arbeitsberichte aus dem Fachbereich Informatik 8/2009

Petra Schubert et. al., Open-Source-Software für das Enterprise Resource Planning, Arbeitsberichte aus dem Fachbereich Informatik 7/2009

Ammar Mohammed, Frieder Stolzenburg, Using Constraint Logic Programming for Modeling and Verifying Hierarchical Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 6/2009

Tobias Kippert, Anastasia Meletiadou, Rüdiger Grimm, Entwurf eines Common Criteria-Schutzprofils für Router zur Abwehr von Online-Überwachung, Arbeitsberichte aus dem Fachbereich Informatik 5/2009

Hannes Schwarz, Jürgen Ebert, Andreas Winter, Graph-based Traceability – A Comprehensive Approach. Arbeitsberichte aus dem Fachbereich Informatik 4/2009

Anastasia Meletiadou, Simone Müller, Rüdiger Grimm, Anforderungsanalyse für Risk-Management-Informationssysteme (RMIS), Arbeitsberichte aus dem Fachbereich Informatik 3/2009

Ansgar Scherp, Thomas Franz, Carsten Saathoff, Steffen Staab, A Model of Events based on a Foundational Ontology, Arbeitsberichte aus dem Fachbereich Informatik 2/2009

Frank Bohdanovicz, Harald Dickel, Christoph Steigner, Avoidance of Routing Loops, Arbeitsberichte aus dem Fachbereich Informatik 1/2009

Stefan Ameling, Stephan Wirth, Dietrich Paulus, Methods for Polyp Detection in Colonoscopy Videos: A Review, Arbeitsberichte aus dem Fachbereich Informatik 14/2008

Tassilo Horn, Jürgen Ebert, Ein Referenzschema für die Sprachen der IEC 61131-3, Arbeitsberichte aus dem Fachbereich Informatik 13/2008

Thomas Franz, Ansgar Scherp, Steffen Staab, Does a Semantic Web Facilitate Your Daily Tasks?, Arbeitsberichte aus dem Fachbereich Informatik 12/2008

Norbert Frick, Künftige Anforderungen an ERP-Systeme: Deutsche Anbieter im Fokus, Arbeitsberichte aus dem Fachbereich Informatik 11/2008

Jürgen Ebert, Rüdiger Grimm, Alexander Hug, Lehramtsbezogene Bachelor- und Masterstudiengänge im Fach Informatik an der Universität Koblenz-Landau, Campus Koblenz, Arbeitsberichte aus dem Fachbereich Informatik 10/2008

Mario Schaarschmidt, Harald von Kortzfleisch, Social Networking Platforms as Creativity Fostering Systems: Research Model and Exploratory Study, Arbeitsberichte aus dem Fachbereich Informatik 9/2008

Bernhard Schueler, Sergej Sizov, Steffen Staab, Querying for Meta Knowledge, Arbeitsberichte aus dem Fachbereich Informatik 8/2008

Stefan Stein, Entwicklung einer Architektur für komplexe kontextbezogene Dienste im mobilen Umfeld, Arbeitsberichte aus dem Fachbereich Informatik 7/2008

Matthias Bohnen, Lina Brühl, Sebastian Bzdak, RoboCup 2008 Mixed Reality League Team Description, Arbeitsberichte aus dem Fachbereich Informatik 6/2008

Bernhard Beckert, Reiner Hähnle, Tests and Proofs: Papers Presented at the Second International Conference, TAP 2008, Prato, Italy, April 2008, Arbeitsberichte aus dem Fachbereich Informatik 5/2008

Klaas Dellschaft, Steffen Staab, Unterstützung und Dokumentation kollaborativer Entwurfs- und Entscheidungsprozesse, Arbeitsberichte aus dem Fachbereich Informatik 4/2008

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese, Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priese: Finite Automata on Unranked and Unordered DAGs Extended Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aus dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description: "E-KRHyper", Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Information systems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priese, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen“, Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“

(<http://www.uni-koblenz.de/fb4/publikationen/gelbereihe>)

Lutz Prieze: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißel: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005