



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik



Orientierung Aus Silhouetten

Diplomarbeit
zur Erlangung des Grades
DIPLOM-INFORMATIKER
im Studiengang Computervisualistik

vorgelegt von

Matthias Scharek

Betreuer: Dipl.-Inform. Peter Decker, Institut für Computervisualistik,
Fachbereich Informatik, Universität Koblenz-Landau

Erstgutachter: Dipl.-Inform. Peter Decker, Institut für
Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau

Zweitgutachter: Prof. Dr.-Ing. Dietrich Paulus, Institut für
Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau

Koblenz, im Mai 2010

Kurzfassung

Anhand der sogenannten „Analyse durch Synthese“ soll in der folgenden Qualifikationsarbeit versucht werden, zum Zeitpunkt einer Aufnahme auf dem Campus der Universität Koblenz die Orientierung der Kamera zu bestimmen. Die Lösungsidee für diese Problemstellung liegt in Form eines merkmalsbasierten Ansatzes vor. Zu diesem Zweck ist die grobe Position im 3D-Campusmodell der Universität Koblenz zum Zeitpunkt der entsprechenden Aufnahme gegeben. Um das Bild einer realen Aufnahme mit einem künstlichen Bild des 3D-Modells vergleichen zu können, ist ein Ähnlichkeitsmaß notwendig, welches die Ähnlichkeit zwischen diesen beiden Bildern mit einem Wert ausdrückt. Der Vergleich erfolgt anhand der Silhouette der einzelnen Bilder, während die Orientierung der Kamera mit Hilfe eines nicht linearen Optimierungsverfahrens bestimmt wird.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Vereinbarung der Arbeitsgruppe für Studien- und Abschlussarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. ja nein

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja nein

Koblenz, den 26. Mai 2010

Inhaltsverzeichnis

1	Einleitung	15
2	Anwendungsmöglichkeiten von Silhouetten	17
2.1	Shape Matching	17
2.2	3D Shape Matching	19
2.3	Horizon Matching	20
3	Orientierung aus Silhouetten	25
3.1	Extraktion der Silhouetten	26
3.2	Generierung von Einzelbildern aus dem Panorama	29
3.3	Detektion von Eckpunkten und Aufstellung von Hypothesen	33
3.4	Berechnung der Blickrichtung	36
3.5	Das Ähnlichkeitsmaß	42
3.6	Das nicht lineare Optimierungsverfahren	46
4	Experimente und Ergebnisse	51
4.1	Implementation	51
4.2	Versuchsreihen und Ergebnisse	52
4.2.1	Künstlich erzeugte Einzelbilder	52
4.2.2	Künstlich verzerrte Einzelbilder	56
4.2.3	Bilder realer Aufnahmen	64
4.3	Laufzeit des Verfahrens	73
4.4	Grenzen des Verfahrens	75
5	Zusammenfassung und Ausblick	77
5.1	Zusammenfassung	77
5.2	Ausblick	79
A	Mathematische Bezeichner und Symbole	81
B	Notwendige Bibliotheken	85

C Aufbau der CD

87

Tabellenverzeichnis

4.1	Künstlich generierte Einzelbilder aus dem Panorama eins.	55
4.2	Künstlich generierte Einzelbilder aus dem Panorama zwei.	55
4.3	Künstlich generierte Einzelbilder aus dem Panorama drei.	55
4.4	Künstlich generierte Einzelbilder aus dem Panorama vier.	56
4.5	Die fünf Einzelbilder aus dem Panorama eins mit minimaler Varianz $\sigma^2 = 1$ verzerrt.	58
4.6	Die fünf Einzelbilder aus dem Panorama eins mit mittlerer Varianz $\sigma^2 = 7$ verzerrt.	58
4.7	Die fünf Einzelbilder aus dem Panorama eins mit maximaler Varianz $\sigma^2 = 15$ verzerrt.	58
4.8	Die fünf Einzelbilder aus dem Panorama zwei mit minimaler Vari- anz $\sigma^2 = 1$ verzerrt.	60
4.9	Die fünf Einzelbilder aus dem Panorama zwei mit mittlerer Varianz $\sigma^2 = 7$ verzerrt.	60
4.10	Die fünf Einzelbilder aus dem Panorama zwei mit maximaler Vari- anz $\sigma^2 = 15$ verzerrt.	60
4.11	Die fünf Einzelbilder aus dem Panorama drei mit minimaler Varianz $\sigma^2 = 1$ verzerrt.	60
4.12	Die fünf Einzelbilder aus dem Panorama drei mit mittlerer Varianz $\sigma^2 = 7$ verzerrt.	61
4.13	Die fünf Einzelbilder aus dem Panorama drei mit maximaler Varianz $\sigma^2 = 15$ verzerrt.	61
4.14	Die fünf Einzelbilder aus dem Panorama vier mit minimaler Varianz $\sigma^2 = 1$ verzerrt.	63
4.15	Die fünf Einzelbilder aus dem Panorama vier mit mittlerer Varianz $\sigma^2 = 7$ verzerrt.	63
4.16	Die fünf Einzelbilder aus dem Panorama vier mit maximaler Varianz $\sigma^2 = 15$ verzerrt.	63
4.17	Die ersten 20 Ergebnisse für die 40 Bilder realer Aufnahmen, die eine 360° Sicht um den Betrachter zeigen.	65

4.18	Die letzten 20 Ergebnisse für die 40 Bilder realer Aufnahmen, die eine 360° Sicht um den Betrachter zeigen.	66
4.19	Die zusätzlichen Ergebnissen für die Bilder Nummer 28 und 29. . .	69
4.20	Die zusätzlichen Ergebnissen für die Bildserie Nummer zehn, 18, sowie 20 bis 23.	69
A.1	Mathematische Bezeichner	81
A.2	Fortsetzung 1 der Mathematischen Bezeichner.	82
A.3	Fortsetzung 2 der Mathematischen Bezeichner.	83

Abbildungsverzeichnis

2.1	Abbildung (a) zeigt das gerenderte Terrain, Abbildung (b) zeigt die extrahierten Features und Abbildung (c) den Abgleich der Horizontlinie bei [BBGL96].	21
2.2	Horizont bei Fridtjof Stein und Gérald Medioni [SM95].	22
2.3	Scheibenschnitt durch die Landschaft bei Fridtjof Stein und Gérald Medioni [SM95].	22
2.4	Supersegment bei Fridtjof Stein und Gérald Medioni [SM95].	23
2.5	Die Generierung einer Hypothese bei Behringer [Beh99].	24
3.1	Aktivitätsdiagramm des Systems mit der Annahme, dass in beiden Bildern Eckpunkte detektiert werden konnten. Levmar bezeichnet die Implementierung der Levenberg-Marquardt Methode.	26
3.2	Eine Aufnahme, welche auf dem Campus der Universität gemacht wurde mit dem entsprechenden Silhouettenbild.	28
3.3	Das Panoramabild einer 360° Rundumsicht mit dem entsprechenden Silhouettenbild.	28
3.4	Die Zylinderkoordinaten.	30
3.5	Horizontaler Öffnungswinkel θ_{hor} und vertikaler Öffnungswinkel θ_{ver} der Kamera.	30
3.6	Berechnung des Schnittpunktes mit dem Zylinderpanorama.	31
3.7	Zwei Ausschnitte aus dem 360° Panorama, links noch verzerrt und rechts entzerrt.	33
3.8	Der Vergleich zwischen zwei Horizontlinien, links unverändert und rechts approximiert.	34
3.9	Die zwei Arten von linken Eckpunkten	35
3.10	Die zwei Arten von rechten Eckpunkten	35
3.11	Korrespondenz zwischen einem Punkt der Liste L_A und einem Punkt der Liste L_P	36
3.12	Berechnung des Winkels κ_{hor} auf dem aufgeklappten Zylinderpanorama Z_a	37

3.13	Berechnung des Winkels φ_{hor} auf dem Bild B_A der realen Aufnahme A.	38
3.14	Fall eins und zwei für die Berechnung von α_{hor}	41
3.15	Fall drei und vier für die Berechnung von α_{hor}	41
3.16	Fall fünf und sechs für die Berechnung von α_{hor}	42
3.17	Die Silhouettenbilder, die miteinander durch die XOR-Operation zu einem Resultatbild verbunden werden.	43
3.18	Die Ähnlichkeit zwischen einem Bild mit festgelegter Blickrichtung und Bildern, bei denen die Blickrichtung zwischen 0° bis maximal 10° und 20° abweicht.	44
3.19	Die Ähnlichkeit zwischen einem Bild mit festgelegter Blickrichtung und Bildern, bei denen die Blickrichtung zwischen 0° bis maximal 30° und 40° abweicht.	45
3.20	Die Ähnlichkeit zwischen einem Bild mit festgelegter Blickrichtung und Bildern, bei denen die Blickrichtung zwischen 0° bis maximal 50° und 60° abweicht.	45
3.21	Das Herantasten durch die Levenberg-Marquardt Methode an die gesuchte Blickrichtung Nummer fünf.	50
4.1	Zwei unterschiedliche Szenen, die aber durch das Ähnlichkeitsmaß einen hohen Wert für die Ähnlichkeit zugewiesen bekommen.	54
4.2	Das Bild einer Silhouette verzerrt mit maximaler Varianz von $\sigma^2 = 15$	57
4.3	Zusammenfassung der Ergebnisse des ersten Panoramas mit künst- lich verzerrten Bildern.	59
4.4	Zusammenfassung der Ergebnisse des zweiten Panoramas mit künst- lich verzerrten Bildern.	61
4.5	Zusammenfassung der Ergebnisse des dritten Panoramas mit künst- lich verzerrten Bildern.	62
4.6	Zusammenfassung der Ergebnisse des vierten Panoramas mit künst- lich verzerrten Bildern.	64
4.7	Die Silhouetten der Bilder Nummer 25 und 26, in denen keine Eck- punkte extrahiert werden konnten.	67
4.8	Der Vergleich zwischen zwei Approximationen der Horizontlinie des gleichen Bildes Nummer 27.	68
4.9	Abbildung (a) zeigt die gesuchte Szene, Abbildung (b) die vom Al- gorithmus fälschlicherweise ausgegebene Szene und Abbildung (c) die Szene nach Anpassen der Parameter.	68
4.10	Abbildung (a) zeigt die gesuchte Szene, Abbildung (b) die vom Al- gorithmus fälschlicherweise ausgegebene Szene und Abbildung (c) die Szene, die in den Evaluationsdaten zu finden war, aber nicht ausgegeben wurde.	70

4.11	Abbildung (a) zeigt die gesuchte Szene, Abbildung (b) die vom Algorithmus fälschlicherweise ausgegebene Szene und Abbildung (c) die Szene, die in den Evaluationsdaten zu finden war, aber nicht ausgegeben wurde.	70
4.12	Abbildung (a) zeigt die gesuchte Szene, Abbildung (b) die vom Algorithmus fälschlicherweise ausgegebene Szene und Abbildung (c) die Szene, die in den Evaluationsdaten zu finden war, aber nicht ausgegeben wurde.	70
4.13	Eines der besseren Ergebnisse mit Bild Nummer 38 für diese Versuchsreihe.	71
4.14	Einer der Ausreißer mit Bild Nummer 15 für diese Versuchsreihe.	71
4.15	Zusammenfassung der Ergebnisse für die Abweichungen in horizontaler Blickrichtung.	72
4.16	Zusammenfassung der Ergebnisse für die Abweichungen in vertikaler Blickrichtung.	72
4.17	Zusammenfassung der Ergebnisse für die Laufzeit bei Bildern, für die das nicht lineare Optimierungsverfahren mit einer richtigen Lösung terminierte.	74
4.18	Zusammenfassung der Ergebnisse für die Laufzeit bei Bildern, für die das nicht lineare Optimierungsverfahren mit einer falschen Lösung terminierte.	74

Kapitel 1

Einleitung

In den letzten Jahren häuften sich die Möglichkeiten 3D Modelle zu erzeugen. Zahlreiche Software Programme wie 3ds Max¹, Blender² oder Ogre³ ermöglichen es heute mit etwas Übung 3D Szenen zu erzeugen. Aufgrund dieser Tatsache gewann die sogenannte „Analyse durch Synthese“ in den letzten Jahren mehr und mehr an Bedeutung. Mittlerweile versuchen viele Autoren mit verschiedenen Ansätzen das Lokalisierungsproblem - oder auch „Drop Off“ Problem genannt - mit Hilfe dieser Forschungsmethode zu lösen. Dabei geht man von der 3D Szene einer realen Umgebung aus und vergleicht Bilder realer Aufnahmen dieser Umgebung mit Einzelbildern, die man aus dem 3D Modell generieren kann. Gleichen sich die Szenen in beiden Bildern, kann man durch die Koordinaten im 3D Modell die Position der Kamera zum Zeitpunkt der Aufnahme bestimmen.

In dieser Qualifikationsarbeit wird ein merkmalsbasierter Ansatz vorgestellt, der zu einer groben Position im 3D Modell einer Szene versucht die Orientierung der Kamera zum Zeitpunkt verschiedener Aufnahmen zu bestimmen. Bei der Szene handelt es sich um den Campus der Universität Koblenz⁴. Um die Orientierung zum Zeitpunkt einer Aufnahme bestimmen zu können, müssen die Bilder der realen Aufnahmen mit den Einzelbildern aus dem 3D Modell verglichen werden. Der Vergleich erfolgt anhand der Silhouette eines Bildes. Silhouetten haben im Vergleich zu texturbasierten Ansätzen wie SIFT den Vorteil, dass beispielsweise weiße Wände, die überhaupt keine Merkmale bieten oder sich wiederholende Fensterreihen, die man schwer auseinander halten kann, keine Probleme verursachen. Desweiteren ist die Herstellung von Texturen immer mit einem gewissen Zeitaufwand verbunden. Dies kann bei Silhouetten komplett außer Acht gelassen werden. Der Nachteil von Silhouetten hingegen besteht in ihrer Mehrdeutigkeit. Da Gebäude mit großer

¹<http://www.autodesk.de>

²<http://www.blender.org/>

³<http://www.ogre3d.org/>

⁴<http://www.uni-koblenz-landau.de/koblenz/>

Wahrscheinlichkeit aus vier Wänden bestehen, ist es mit Hilfe der Silhouette ohne zusätzliche Informationen nicht möglich zu entscheiden, auf welcher Seite des Gebäudes man sich befindet. Weiterhin sind Silhouetten störanfällig gegenüber Objekten, die sich zwar in Bildern der realen Aufnahmen befinden, aber nicht im 3D Modell. Man ist sozusagen auf die Aktualität des 3D Modells angewiesen, um überhaupt ähnliche Bilder vergleichen zu können. Dies ist aber eher ein grundsätzlicher Nachteil an der Forschungsmethode „Analyse durch Synthese“. Da man letztendlich an der Orientierung der Kamera interessiert ist, wird mit Hilfe eines nicht linearen Optimierungsverfahrens versucht, diese in mehreren Iterationen so gut wie möglich an die gesuchte Orientierung anzunähern.

Die Gliederung dieser Qualifikationsarbeit setzt sich aus insgesamt fünf Kapiteln zusammen. Das zweite Kapitel gibt einen umfassenden Einblick in das Forschungsgebiet der Silhouetten der letzten 15 Jahre. Kapitel drei stellt den merkmalsbasierten Algorithmus und alle notwendigen Komponenten, die zur Lösung des Problems benötigt werden, vor. Kapitel vier ist unterteilt in drei Experimente und deren Ergebnisse, die dazu dienen den Algorithmus zu testen. Darüber hinaus werden in diesem Kapitel die Grenzen des Verfahrens, die Laufzeit des Algorithmus, sowie die Implementantation beschrieben. Den Schluß dieser Arbeit bildet das Kapitel fünf, welches die Probleme, Lösungen und Ergebnisse noch einmal zusammenfasst und einen Ausblick auf mögliche Verbesserungen des Ansatzes schildert.

Kapitel 2

Anwendungsmöglichkeiten von Silhouetten

Das Forschungsgebiet im Bereich der Silhouetten hat im Laufe der letzten 15 Jahre stark an Umfang zugenommen. Umso verwunderlicher ist es, dass während der Literaturrecherche dieser Qualifikationsarbeit keine Ansätze gefunden wurden, die sich ausschließlich nur mit Silhouetten von Gebäuden befassen. Stattdessen wird im folgenden Kapitel dem Leser ein umfassender Einblick in den Stand der Technik dieses Forschungsgebietes gegeben und somit verschiedene Anwendungsmöglichkeiten von Silhouetten präsentiert, die zwar von der Aufgabenstellung dieser Arbeit abweichen können, aber sie dennoch geprägt haben. Nach Ansicht des Verfassers lassen sich die hier relevanten Teilgebiete des Forschungsgebietes wie folgt unterteilen:

- Shape Matching
- 3D-Shape Matching
- Horizon Matching

2.1 Shape Matching

Shape Matching wird größtenteils in der Objekterkennung verwendet. Hierbei lassen sich die Shape Matching Algorithmen im wesentlichen in globale und lokale Algorithmen unterteilen. Die Gemeinsamkeit dieser Ansätze besteht darin, nach einer geeigneten Repräsentation für den Umriss eines Objekts zu suchen, um darauf basierend Shapes miteinander vergleichen zu können.

So versuchen die Autoren Ronen Basri et al. einen Umriss mit Hilfe von Kurven zu beschreiben, der dann mit einer Distanzfunktion verglichen werden kann.

Diese Distanzfunktion drückt die Ähnlichkeit von zwei Objekten aus, indem sie die Kosten zurückliefert, die man benötigt, um ein Objekt in ein anderes zu deformieren. Dafür legen die Autoren bestimmte Bedingungen für die Kostenfunktion fest. Beispielsweise sollen viele kleine Deformierungen den Umriss weniger beeinflussen, als eine Deformierung gleichen Ausmaßes [BCGJ95].

Um den Umriss eines Objektes zu repräsentieren und zu vergleichen, verwendet Saliha Bougar einen Deskriptor, der die Eigenschaft besitzt ein Objekt in verschiedene Teile zu zerlegen. Der Vergleich von zwei Objekten erfolgt in zwei Schritten. Zum einen wird der globale Deskriptor und zum anderen der lokale Deskriptor verglichen. Der globale Deskriptor gibt beispielsweise Auskunft über die Anzahl der Teile, die ein Objekt zerlegen. Der lokale Deskriptor hingegen beschreibt die geometrische Form der Kontur oder die Orientierung. Die Deskriptoren wiederum werden mit Hilfe von LWDOS (Language for Writing Descriptors of Outline Shapes) [LBTFL03] - einer Sprache für Deskriptoren - die eine Kontur beschreiben sollen, beschrieben [Bou05].

Serge Belongie et al. versuchen primär mit einem einfachen Algorithmus Korrespondenzen zwischen Umrisen zu finden. Hierbei wird der Umriss durch eine Vielzahl von Punkten, welche mit Hilfe eines Eckendetektors detektiert wurden, beschrieben. Weiterhin stellen die Autoren „shape context“, einen Shape Deskriptor, der zu einem Punkt auf dem Shape grob den Verlauf des restlichen Shapes beschreiben soll, vor. Die Ähnlichkeit zwischen zwei Objekten wird mit Hilfe von zwei Kostenfunktionen und der Menge an Transformationen, die nötig sind, um die Shapes richtig auszurichten, berechnet. Die erste Kostenfunktion beschreibt die Kosten, die notwendig sind, um zwei korrespondierende Punkte abzugleichen. Die zweite Kostenfunktion wird durch die quadrierte Differenz in der Helligkeit um korrespondierende Punkte herum beschrieben [BMP02].

Die Arbeit von Dengsheng Zhang und Guojun Lu besteht darin, lokale und globale Deskriptoren vorzustellen, um daraus einen Deskriptor abzuleiten, der die Vorteile beider Arten kombiniert. Als lokalen Deskriptor wählen die Autoren einen Fourier Deskriptor. Diesen erhalten sie, indem die diskrete Fourier-Transformation auf die Außenpunkte des Umrisses angewendet wird. Der globale Deskriptor besteht aus komplexen Zernike Momenten, die sich aus Zernike Polynomen ableiten lassen. Bei der Berechnung eines Zernike Moment Deskriptors sind keine Informationen über den Umriss notwendig. Letztendlich kombinieren die Autoren beide Deskriptoren in einem Vektor, der aus Fourierkoeffizienten und Zernike Momenten besteht, die sich mit zwei unterschiedlichen Faktoren gewichten lassen. Die Ähnlichkeit zwischen zwei Vektoren läßt sich mit der euklidischen Distanz ausdrücken [ZL02].

Haibin Ling und David W. Jacobs verwenden die innere Distanz zwischen zwei Punkten eines Shapes, um einen Deskriptor zu erstellen. Die innere Distanz wird

dabei durch den kürzesten Pfad zwischen zwei Punkten in einem geschlossenen Shape definiert. Die Ähnlichkeit zwischen zwei Shapes wird wie in [BMP02] berechnet, mit dem Unterschied, dass die innere Distanz in den „shape context“ mit einfließt [LJ05].

Der Algorithmus, den die Autoren Michael Donoser et al. vorstellen, heißt „IS-Match“. Dieser bekommt zwei Umrisse als Eingabe und liefert teilweise erkannte Übereinstimmungen und einen Ähnlichkeitswert als Ergebnis. Ein Umriss ist durch die Reihenfolge der gesammelten Punkte einer Kontur repräsentiert. „IS-Match“ orientiert sich an sogenannten „chord“ Ansätzen. Ein „chord“ ist eine Linie zwischen zwei Punkten, die sich an der Außenseite des Shapes befinden. Der Deskriptor setzt sich somit aus den Winkeln zwischen zwei ausgewählten „chords“ zusammen und die Ähnlichkeit zwischen zwei Deskriptoren wird mit der Salukwadze Distanz berechnet [DRB09].

2.2 3D Shape Matching

Der Übergang zwischen Shape Matching und 3D Shape Matching ist fließend. Viele der vorgestellten Ansätze aus dem Bereich des Shape Matchings lassen sich auch auf 3D Objekte übertragen. Dessen ungeachtet gibt es Anwendungsmöglichkeiten, wie die Erkennung von Gesichtern oder die Lokalisierung von Personen in 3D, die unter dem Begriff 3D Shape Matching besser zusammengefasst werden können. Im Gegensatz zum Shape Matching kommt es in diesem Teilgebiet der Forschung weniger auf die Repräsentation einer Silhouette, als mehr auf den Vergleich zweier Silhouetten an.

Hendrik P. A. Lensch et al. vergleichen die Silhouette eines Fotos von einem Objekt mit der Silhouette des entsprechenden 3D Modells, um dann möglichst einfach Texturen aus dem Foto auf das 3D Modell übertragen zu können. Hierfür benötigen die Autoren Kameraparameter, die sie mit Hilfe der Downhill Simplex Methode [NM65] aus dem Foto bestimmen, um das 3D Modell aus der gleichen Sicht rendern zu können. Der Vergleich zwischen den beiden Silhouetten erfolgt mit der XOR-Operation, auf die in Kapitel drei genauer eingegangen wird [LHS01].

Die Arbeit von Jinho Lee et al. beschäftigt sich mit der Rekonstruktion von Gesichtern anhand der Silhouette. Da die Autoren nur an bestimmten Bereichen des Gesichtes interessiert sind, müssen die 3D Modelle eines echten Gesichtes zugeschnitten werden. Hierfür werden 3D Modelle von Eigen Faces verwendet, die zuerst in die gleiche Position gebracht werden. Dies erfolgt mit dem Levenberg-Marquardt Verfahren [MNT04], welches in Kapitel drei explizit behandelt wird. Nach Extraktion der Silhouette aus dem zugeschnittenen 3D Modell und dem Aufnehmen der echten Silhouette mit einer Kamera werden die beiden Silhouetten mit einer angepassten XOR-Funktion verglichen. Da die Silhouetten, die mit einer

Kamera aufgenommen wurden, größere Teile des Gesichtes abdecken, als die Silhouetten aus dem 3D Modell, werden Ungleichheiten an der Kontur des Gesichtes stärker gewichtet [LMP⁺03].

Thomas B. Moeslund und Erik Granum verwenden in ihrer Arbeit, im Gegensatz zu vielen Ansätzen aus dem 3D Shape Matching, anstelle der XOR-Funktion die AND Funktion, um aus 2D Daten die 3D Pose von Menschen zu bestimmen. Bei dieser Funktion können fehlerhafte Ähnlichkeiten entstehen, da die AND Funktion nicht die Überdeckung von Punkten zwischen Silhouettenbildern angibt, sondern lediglich nur eine Aussage über die Anzahl der Punkte eines Objektes trifft, wenn sich das Objekt im Silhouettenbild befindet. Dieses Problem versuchen die Autoren zu umgehen, indem sie zusätzlich noch Informationen über den Umriss einer Silhouette mit Hilfe der Bounding Box in Betracht ziehen [MG00].

Ding-Yun Chen et al. gehen bei ihrem Ansatz davon aus, dass falls zwei 3D Objekte ähnlich sind, sie auch aus unterschiedlichen Blickwinkeln ähnlich aussehen. Auf Basis dieser Idee werden Silhouetten eines 3D Objektes aus unterschiedlichen Blickwinkeln um das 3D Objekt gebildet. Diese Bilder ergeben einen Lichtfeld Deskriptor. Aus diesem werden wie in [ZL02] Deskriptoren extrahiert, deren Ähnlichkeit durch eine einfache L1-Distanz zwischen den einzelnen Koeffizienten ausgedrückt wird [CTSO03].

Ähnlich dem Ansatz von Ding-Yun et al. werden bei Ameesh Makadia et al. die Silhouetten des 3D Modells mit Hilfe einer Sphere um das Objekt herum erzeugt. Dabei wird die Sphere in verschiedene Samples aufgeteilt, auf denen dann die Silhouetten orthografisch auf eine Tangentialebene abgebildet werden. Mehrere Silhouetten eines Objektes werden zu einem Lichtmodell zusammengefasst, welches mit der Zentroid Distanz und Zernike Momenten verglichen wird [MVD07].

2.3 Horizon Matching

Horizon Matching ist das Teilgebiet im Bereich von Silhouetten, welches der Thematik der Aufgabenstellung dieser Qualifikationsarbeit am nächsten kommt. Grundsätzlich wird versucht die Position eines Objektes in Outdoor Szenarien zu bestimmen. Bei diesen Outdoor Szenarien handelt es sich meistens um Landschaften, die beispielsweise in näherer Umgebung Hügel aufweisen. Zwei wichtige Teilaspekte des Horizon Matchings sind die Repräsentation einer Horizontlinie sowie der Vergleich zwischen zwei Horizontlinien.

J. Ross Beveridge et al. präsentieren in ihrer Arbeit einen lokalen Algorithmus, der bruchstückartige Horizontlinien abgleicht. Als 3D Modell verwenden die Autoren eine sogenannte „Digital Elevation Map“ (DEM) von Lockheed-Martin. Der Vorteil hierbei besteht darin, dass es Testbilder zu dieser Digital Elevation Map

mit bekannten Positionen gibt. Das Verfahren lässt sich grob mit drei Schritten beschreiben:

1. Das 3D Modell wird mit einer bekannten Position gerendert (siehe Abbildung 2.1 (a)).
2. Aus dem 3D Modell werden mit Hilfe einer eigenen Implementation des Burns Algorithmus [BHR86] Features extrahiert (siehe Abbildung 2.1 (b)).
3. Die Features werden miteinander verglichen (siehe Abbildung 2.1 (c)).

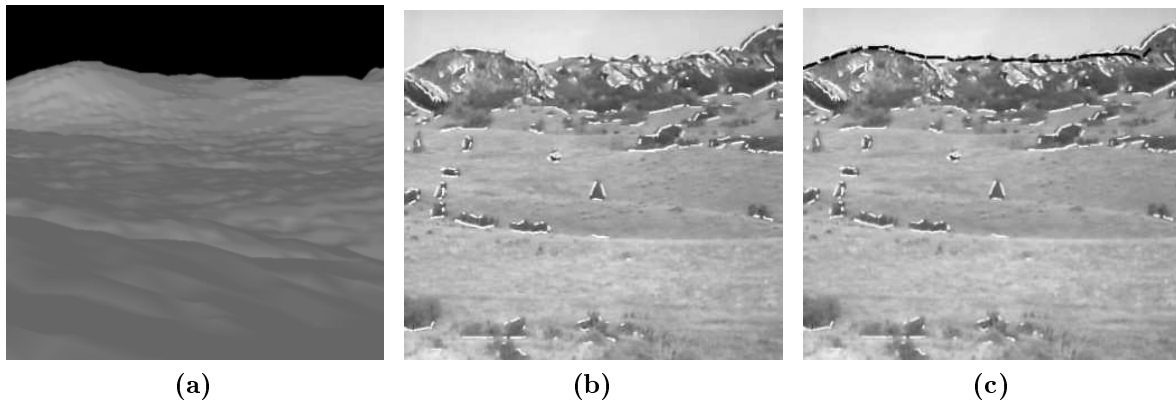


Abbildung 2.1: Abbildung (a) zeigt das gerenderte Terrain, Abbildung (b) zeigt die extrahierten Features und Abbildung (c) den Abgleich der Horizontlinie bei [BBGL96].

Um die Anzahl von möglichen Vergleichen zwischen dem Horizontmodell und den Bildlinienssegmenten zu limitieren, werden zwei Hypothesen aufgestellt. Zum einen wird angenommen, dass sich der Horizont zentriert im Bereich einer halben Höhe des Bildes von der Position befindet, des Weiteren darf der Winkel zwischen zwei Segmenten nicht mehr als 17 Grad betragen. Der Abgleich zwischen dem Horizont und den Bildlinienssegmenten wird mit einem Fehlermaß angegeben [BBGL96].

Fridtjof Stein und Gérald Medioni präsentieren mit ihrem Ansatz eine Lösung für das Lokalisierungsproblem, bei dem ein Betrachter mit einer topografischen Karte einer Umgebung an einer unbekannt Position ausgesetzt wird. Zu diesem Zweck sind Korrespondenzen zwischen Merkmalen in der Karte und den Bildern, die der Betrachter mit einer Kamera fotografiert, notwendig. Als Merkmal wählen die Autoren eine panoramische Horizontkurve. Der Horizont wird dabei als obere Grenze einer Projektion aller Punkte einer Landschaft auf einen Zylinder um den Betrachter herum definiert (siehe Abbildung 2.2).

Für den Vergleich zwischen der Horizontkurve einer topografischen Karte und der vom Beobachter gesehenen Horizontkurve, müssen die Horizonte zuerst extrahiert werden. Ersteres wird mit Hilfe von abgetasteten Scheibenschnitten durch

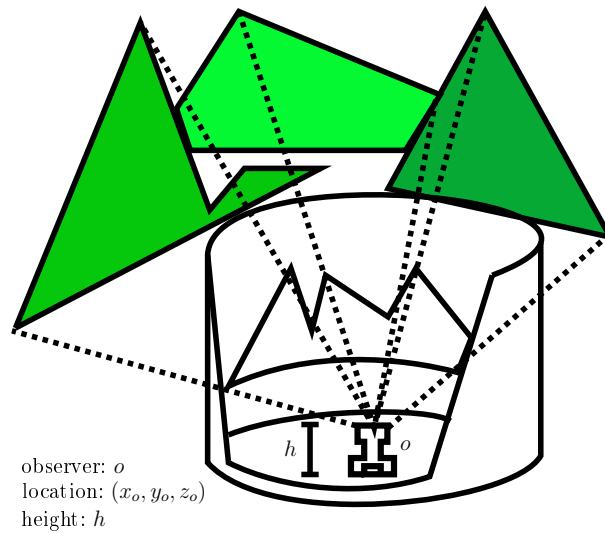


Abbildung 2.2: Horizont bei Fridtjof Stein und Gérald Medioni [SM95].

die Landschaft um die Position des Beobachters herum generiert (siehe Abbildung 2.3) und letzteres ergibt sich durch die Tatsache, dass der Beobachter mit einer Kamera ausgestattet ist, mit der er Panoramafotos schießt.

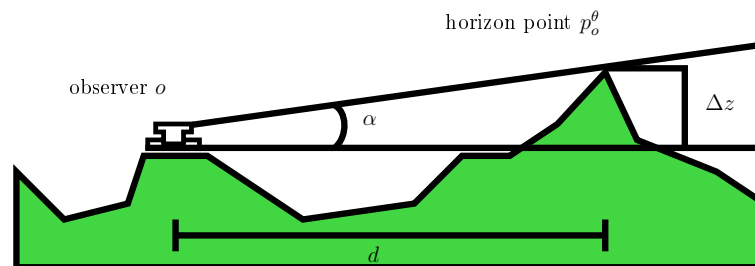


Abbildung 2.3: Scheibenschnitt durch die Landschaft bei Fridtjof Stein und Gérald Medioni [SM95].

Die Horizontkurven selbst werden durch eine polygonale Approximation beschrieben. Dies hat den Vorteil, dass bestimmte Eigenschaften von Kurven durch eine polygonale Approximation in Form von Winkeln zwischen fortlaufenden Segmenten erfasst werden können. Da aber einzelne, individuelle Segmente zu lokal für einen Vergleich sind, fassen die Autoren angrenzende Segmente zu Supersegmenten zusammen. Ein Supersegment ss lässt sich dabei mit folgenden Eigenschaften beschreiben (siehe Abbildung 2.4):

- Die Kardinalität ist die Anzahl der Segmente, aus denen das Supersegment ss besteht.

- Ein Supersegment ss , welches aus n Segmenten besteht, besitzt $n - 1$ Winkel zwischen den fortlaufenden Segmenten.
- Die Richtung dir des Supersegmentes wird durch die Mitte zwischen θ_{min} und θ_{max} definiert.
- Die relative Höhenreichweite Δr des Supersegmentes ist die Differenz zwischen r_{max} und r_{min} .

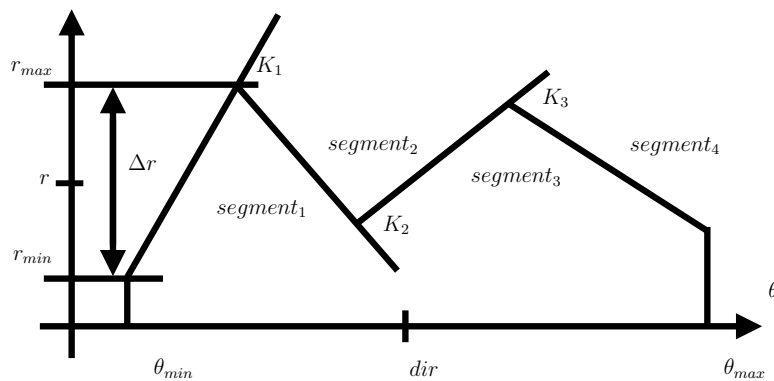


Abbildung 2.4: Supersegment bei Fridtjof Stein und Gérald Medioni [SM95].

Um gute von schlechten Hypothesen bei dem Vergleich zwischen zwei Horizontkurven unterscheiden zu können, fließen die letzten drei Eigenschaften als Fehlermaße mit ein. Der Fehler zwischen zwei Supersegmenten bezüglich der Richtung ist beispielsweise durch $\epsilon_d(h) = ||dir(ss_m) - dir(ss)||$ definiert. Eine gute Hypothese besteht somit aus zwei Supersegmenten mit der gleichen Kardinalität und möglichst kleinen Fehlermaßen. Falls mehr als eine Hypothese für einen Horizont in Frage kommt, stellen die Autoren geometrische Bedingungen auf, die die Anzahl an möglichen Hypothesen einschränken sollen [SM95].

Auch Reinhold Behringer versucht mit seiner Arbeit eine Lösung für das „Drop-Off“ Problem zu finden. Wie im Ansatz von Stein und Medioni, benutzt auch Behringer Digital Elevation Maps (DEM), aus denen er die Silhouette extrahiert und dann mit der Silhouette eines einzelnen Videobildes aus Sicht des Betrachters vergleicht. Um die Silhouette aus dem Videobild zu extrahieren, werden Algorithmen zur Eckendetektion benutzt. Die Position des Beobachters wird beispielsweise mit Hilfe von GPS Daten als bekannt angenommen, so dass basierend auf diesen Daten eine 360 Grad Horizont Silhouette aus der Digital Elevation Map (DEM) erstellt werden kann. Nun werden Hypothesen für die Korrespondenzen in beiden Silhouetten aufgestellt. Als Korrespondenzkandidaten wählt der Autor Minima und Maxima in beiden Silhouetten (siehe Abbildung 2.5). Die Hypothese wird

mit der Summe der quadratischen Distanz zwischen dem am besten interpolierten Punkt einer Video Silhouette und allen Punkten auf der DEM innerhalb des Blickfeldes der Kamera berechnet [Beh99].

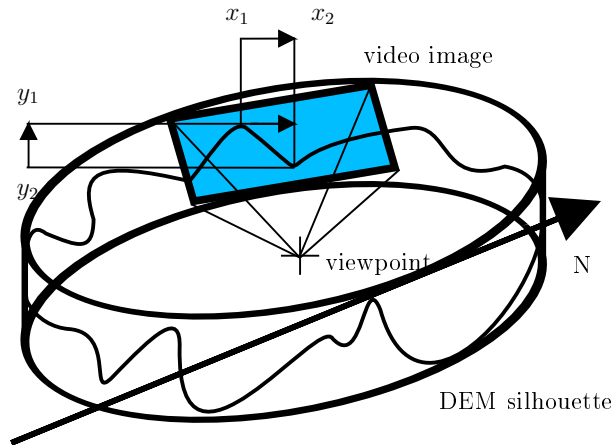


Abbildung 2.5: Die Generierung einer Hypothese bei Behringer [Beh99].

Kapitel 3

Orientierung aus Silhouetten

Im Folgenden wird ein merkmalsbasierter Algorithmus präsentiert, der zur Lösung des vorgestellten Problems führen soll. Dessen Grundidee orientiert sich an der sogenannten „Analyse durch Synthese“. Hierbei wird ein reales Bild, mit den künstlichen Bildern eines 3D Modells verglichen. Anhand der Koordinaten im 3D Modell läßt sich die Position der Kamera zum Zeitpunkt der Aufnahme des realen Bildes bestimmen. Hierfür müssen beide Bilder die gleiche Szene zeigen. Aus der Aufgabenstellung dieser Qualifikationsarbeit ergeben sich folgende Voraussetzung und Bedingungen für die Lösung dieses Problems:

- Die Bilder, zu denen die Orientierung der Kamera bestimmt werden soll, müssen auf dem Campus der Universität Koblenz aufgenommen sein.
- Die grobe Position zum Zeitpunkt der Aufnahme ist bekannt.
- Um die Bilder aus dem 3D Modell mit den realen Bildern vergleichen zu können ist ein geeignetes Ähnlichkeitsmaß notwendig.
- Der Vergleich zwischen zwei Bildern erfolgt anhand der Silhouetten.
- Die Orientierung der Kamera wird mit Hilfe eines nicht linearen Optimierungsverfahrens bestimmt.

Der Algorithmus bekommt als Eingabe echte Aufnahmen vom Campus der Universität Koblenz, die nach [Zha00] entzerrt worden sind, sowie das Panoramabild einer 360° Rundumsicht, welches mit der groben Position der Aufnahme aus dem 3D Modell gerendert wurde. Das 3D Modell stammt aus dem Praktikum „Markante Merkmale“ der Arbeitsgruppe Priese. Durch die Detektion von Eckpunkten in der realen Aufnahme wird versucht Hypothesen aufzustellen, die zur Detektion von Kandidatenpunkten aus den Einzelbildern des Panoramas herangezogen werden. Die Detektion von Eckpunkten wird auf einer approximierten

Horizontlinie der realen und auf einzelnen Aufnahmen, die aus dem Panoramabild gewonnen werden, durchgeführt. Mit Hilfe dieser Punkte kann die Blickrichtung der Kamera geschätzt werden, die dann durch das nicht lineare Optimierungsverfahren Levenberg-Marquardt [MNT04] optimiert wird. Einen groben Einblick in die Aktivitäten des Systems liefert Abbildung 3.1. Hierbei ist anzunehmen, dass sowohl in den realen Aufnahmen Eckpunkte, als auch in den einzelnen Bildern des Panoramas Kandidatenpunkte detektiert werden konnten.

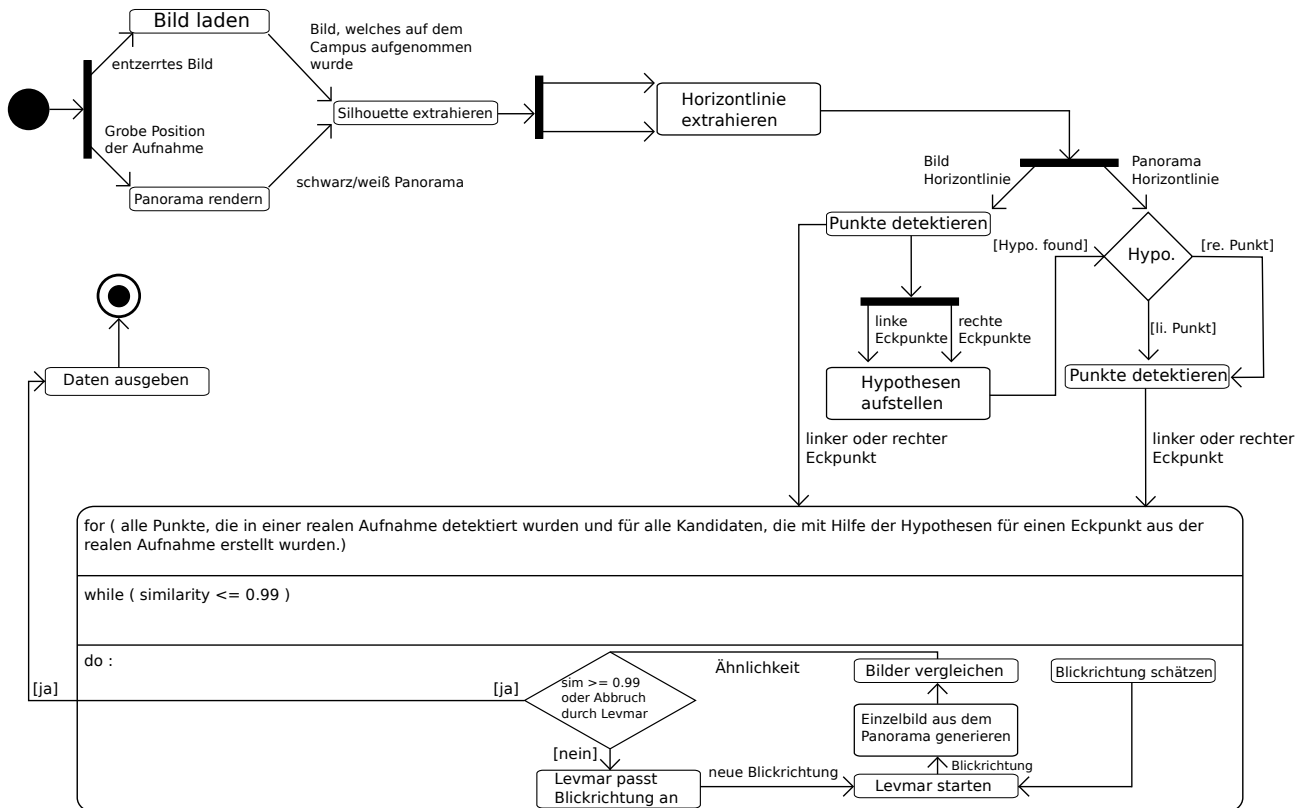


Abbildung 3.1: Aktivitätsdiagramm des Systems mit der Annahme, dass in beiden Bildern Eckpunkte detektiert werden konnten. Levmar bezeichnet die Implementierung der Levenberg-Marquardt Methode.

3.1 Extraktion der Silhouetten

Der Vergleich zwischen zwei Bildern erfolgt anhand der Silhouetten. Dafür müssen die Silhouetten aus den realen Aufnahmen und den Panoramabildern extrahiert werden. Die Extraktion erfolgt mit Hilfe des von Frank Schmitt und Lutz Priese vorgestellten Algorithmus in [SP09]. Um die Stabilität des Algorithmus gewährleisten

zu können, werden Annahmen über Farbe und Position des Himmels aufgestellt. Die Bilder sollten horizontal aufgenommen sein, so dass sich der Himmel im oberen Bereich des Bildes befinden wird. Die Farbe des Himmels kann von weiß über verschiedene Arten von grau innerhalb möglicher Wolken bis hin zu unterschiedlichen Blautönen innerhalb wolkenfreier Regionen variieren. Desweiteren wird angenommen, dass alle Segmente des Himmels eine vertikale Verbindung zum oberen Rand des Bildes haben. Diese Verbindung kann über den direkten Weg oder über andere Segmente, die als Himmel erkannt worden sind, erfolgen. Bevor der Himmel extrahiert wird, ist eine Vorverarbeitung der Bilder notwendig. Zum einem wird der Kuwahara Filter auf die Bilder angewendet und zum anderen werden die Bilder mit der CSC-Segmentierung segmentiert. Bei der CSC-Segmentierung handelt es sich um eine sogenannte „Region Growing“ Segmentierungsmethode, bei der auf Basis der Ähnlichkeit zwischen zwei überlappenden Teilsegmenten, diese zu einem Segment zusammengefasst werden. Ansonsten wird die Subregion zwischen den Segmenten aufgeteilt. Das Resultat ist ein Labelbild, bei dem jeder Wert eines Pixels auf das Label des entsprechenden Segmentes gesetzt ist. Ob ein Segment die Farbe des Himmels trägt, wird mit Hilfe der folgenden Bedingungen für einen HSV Wert geprüft:

- $hsv_{saturation} < 13$ und $hsv_{value} > 216$
- $hsv_{saturation} < 25$ und $hsv_{value} > 204$ und $hsv_{hue} > 190$ und $hsv_{hue} < 250$
- $hsv_{saturation} < 128$ und $hsv_{value} > 153$ und $hsv_{hue} > 200$ und $hsv_{hue} < 230$
- $hsv_{value} < 88$ und $hsv_{hue} > 210$ und $hsv_{hue} > 220$

Die Autoren bezeichnen in diesem Zusammenhang einen HSV Wert, der mindestens eine dieser Bedingungen erfüllt, als HSV Sky Colored. Segmente deren Durchschnittsfarbe Sky Colored ist und bei denen mindestens die Hälfte der oberen Grenzen den oberen Rand des Bildes berühren, kommen in die Liste L_s für Segmente, die als Himmel erkannt wurden. Segmente, die mindestens die untere Kante eines Segmentes aus der Liste L_s berühren, kommen in die Liste L_c für Himmelskandidaten. Für diese Kandidaten müssen zusätzlich folgende Kriterien überprüft werden:

1. Das Segment S aus der Liste L_c ist himmelfarbig.
2. Mindestens zweidrittel der oberen Grenze des Segmentes berühren entweder den oberen Rand des Bildes oder Segmente aus der Liste L_s .
3. Mindestens eine der folgenden Bedingungen ist erfüllt:
 - (a) Die Umgebung von Segment S beinhaltet weniger als 500 Pixel.

(b) Der Mean Vertical Gradient von Segment S ist kleiner 25.

(c) Die Mean Bounded Second Derivative an der oberen Grenze $b_u(S)$ ist kleiner 0.3.

Dabei beschreibt der Mean Vertical Gradient die Helligkeit an der gesamten Grenze zwischen einem Segment und seinen oberen Nachbarn, während die Mean Bounded Second Derivative ein Maß für die Form der Grenze $b_u(S)$ ist. Falls alle drei Kriterien erfüllt sein sollten wird das Segment S als Himmel klassifiziert und die Liste L_s aktualisiert. Der Algorithmus terminiert wenn die Kandidatenliste L_c leer ist [SP09]. Abbildung 3.2 zeigt eine Aufnahme, welche auf dem Campus der Universität Koblenz geschossen wurde, sowie das entsprechende Silhouettenbild. Abbildung 3.3 zeigt wiederum das Panoramabild einer 360° Rundumsicht mit dem dazugehörigen Silhouettenbild.

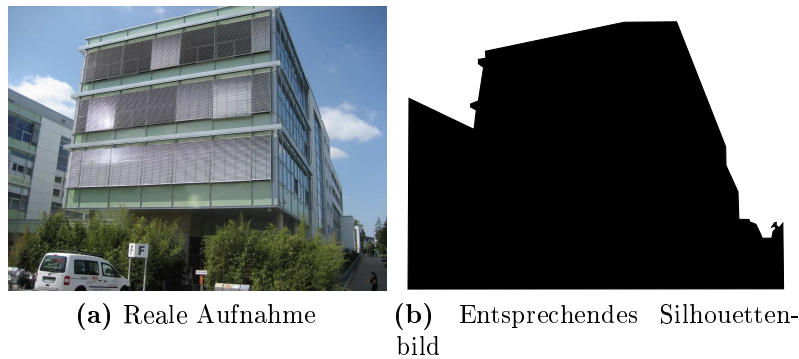


Abbildung 3.2: Eine Aufnahme, welche auf dem Campus der Universität gemacht wurde mit dem entsprechenden Silhouettenbild.



Abbildung 3.3: Das Panoramabild einer 360° Rundumsicht mit dem entsprechenden Silhouettenbild.

3.2 Generierung von Einzelbildern aus dem Panorama

Bei der Verwendung eines nicht linearen Optimierungsverfahrens muß man davon ausgehen, dass mehrere Bilder miteinander verglichen werden. Um nicht ständig immer wieder jedes neue Bild aus dem 3D Modell rendern zu müssen, wird mit der groben Position zum Zeitpunkt der Aufnahme ein einziges 360° Panoramabild gerendert, aus dem man sozusagen jedes Einzelbild, welches die Szene um den Betrachter herum zeigt, generieren kann.

Dies hat den Vorteil, dass es viel kürzer dauert ein Einzelbild aus dem Panorama zu generieren, als ein Einzelbild aus dem 3D Modell zu rendern. Der Nachteil von Panoramaaufnahmen ist jedoch, dass horizontale Geraden (siehe Abbildung 3.3) nicht gerade, sondern gebogen sind. Aus diesem Grund müssen die Einzelbilder des 360° Panoramabildes entzerrt generiert werden.

Da es sich bei der Panoramaaufnahme um ein Zylinderpanorama handelt, genügt es mit Hilfe des Raytracings-Prinzips vom Mittelpunkt des Zylinders aus Strahlen durch eine Bildebene zu senden, die dann auf dem Zylinderpanorama auftreffen. Vereinfacht wird angenommen, dass sich der Mittelpunkt des Zylinders im Ursprung des kartesischen Koordinatensystems befindet.

Benötigt werden hierfür die Blickrichtung und sowohl der horizontale, als auch der vertikale Öffnungswinkel der Kamera. Die Blickrichtung wird mit Hilfe der Zylinderkoordinaten durch die Angabe von zwei Winkeln (siehe Abbildung 3.4) in einen Vektor umgerechnet:

$$x_{view} = r_e \cdot \cos(\alpha_{hor}) \quad (3.1)$$

$$y_{view} = r_e \cdot \sin(\alpha_{hor}) \quad (3.2)$$

$$z_{view} = z \quad (3.3)$$

r_e ist der Radius des Einheitszylinders Z_e . Im Falle der Umrechnung gilt $r_e = 1$. Man kann sich das so vorstellen, als ob der Einheitszylinder Z_e , sich innerhalb des Zylinderpanoramas Z_z befinden würde, mit dessen Hilfe man die Blickrichtung \vec{v} bestimmen kann. Für z_{view} ergibt sich hingegen:

$$z_{view} = r_e \cdot \tan(\alpha_{ver}) \quad (3.4)$$

Somit ergibt sich für die Blickrichtung \vec{v} :

$$\vec{v} = (x_{view} \ y_{view} \ z_{view})^T \quad (3.5)$$

Die beiden Winkel α_{hor} und α_{ver} beschreiben die Orientierung innerhalb des Zylinders. Der Winkel α_{hor} steht dabei für die Ausrichtung der Kamera in x,y -Richtung und der Winkel α_{ver} für die Höhe der Blickrichtung. Die beiden Öffnungswinkel der Kamera θ_{hor} und θ_{ver} definieren wie groß der Ausschnitt im Panorama ist, der auf die Bildebene E_i abgebildet wird (siehe Abbildung 3.5). Bestimmt werden die Öffnungswinkel anhand der Sensorgröße der Kamera und der Brennweite, mit der die Aufnahmen gemacht wurden. Die Sensorgröße wurde im Internet recherchiert.

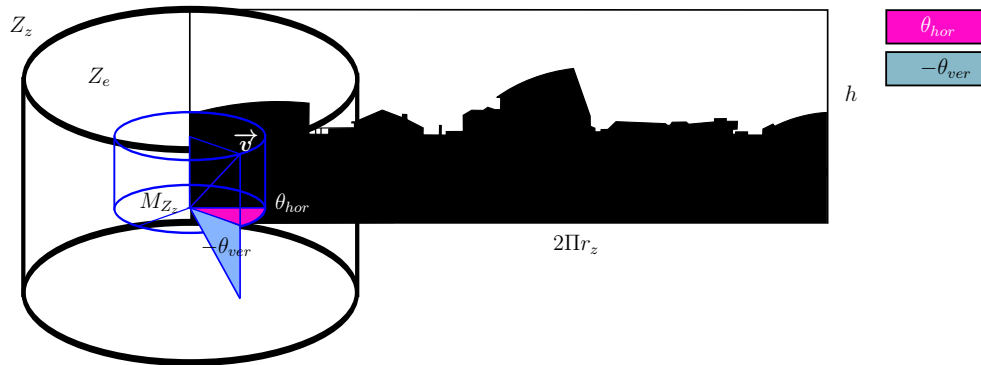


Abbildung 3.4: Die Zylinderkoordinaten.

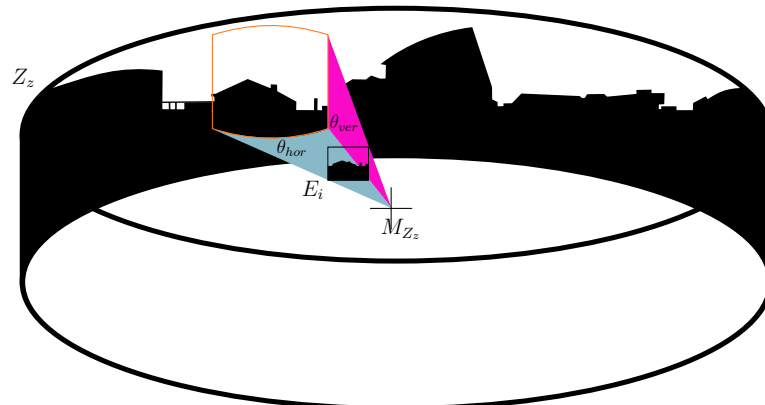


Abbildung 3.5: Horizontaler Öffnungswinkel θ_{hor} und vertikaler Öffnungswinkel θ_{ver} der Kamera.

Am Schnittpunkt S_{Z_z} eines Strahls \vec{s} mit dem Zylinderpanorama Z_z lässt sich die Farbe des jeweiligen Pixels nachschlagen und auf die entsprechende Stelle der Bildebene E_i , durch die der Strahl \vec{s} gesendet wurde, eintragen. Hierfür muss das Zylinderpanorama aufgeklappt werden (siehe Abbildung 3.6). Die Bildebene kann beliebig hoch aufgelöst sein, die Breite und Länge aber sind durch die beiden Öffnungswinkel der Kamera beschränkt. Um den Schnittpunkt S_{Z_z} auf dem Zylinderpanorama Z_z berechnen zu können, werden der Winkel α_{hor} und die Strecke zwischen dem Mittelpunkt M_{E_i} der Bildebene E_i und dem aktuellen Punkt P_{E_i} auf der Bildebene, durch den gerade ein Strahl gesendet wird, benötigt. Mit Hilfe dieser Strecke lässt sich der Winkel β berechnen (siehe Abbildung 3.6). Die Länge dieser Strecke berechnet sich durch die Koordinaten des Punktes P_{E_i} und der Koordinate der Blickrichtung \vec{v} :

$$P_{E_i} = \vec{v} - \vec{i}_x + \vec{i}_y + \left(\frac{(2 \cdot i + 1) \cdot (-2) \cdot \vec{i}_y}{(2 \cdot (v_{Auflösung} + 1))} \right) + \left(\frac{(2 \cdot j + 1) \cdot 2 \cdot \vec{i}_x}{(2 \cdot (h_{Auflösung} + 1))} \right) \quad (3.6)$$

Die Vektoren \vec{i}_x und \vec{i}_y entsprechen den beiden Vektoren, die zum einem in x-Richtung und zum anderen in y-Richtung senkrecht auf \vec{v} stehen. Wobei gilt $\|\vec{i}_x\| = \|\vec{i}_y\| = \|\vec{v}\| = 1$. $h_{Auflösung}$ und $v_{Auflösung}$ sind durch die Auflösung der Bildebene E_i , die angibt wie genau die Bildebene mit Strahlen abgetastet wird, gegeben. i und j sind die Indizes für Zeilen und Spalten der Bildebene E_i .

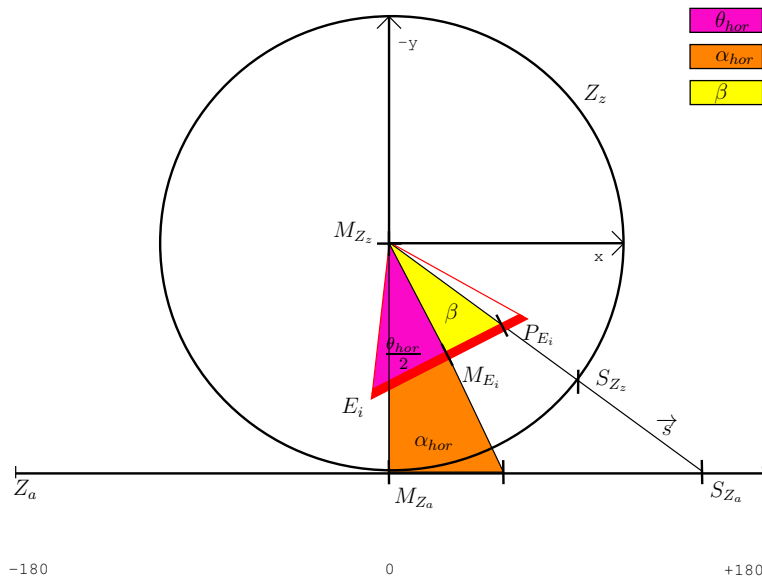


Abbildung 3.6: Berechnung des Schnittpunktes mit dem Zylinderpanorama.

Für das Nachschlagen der Farbe eines Pixels auf dem aufgeklappten Zylinderpanorama Z_a sind letztendlich nur der Winkel β und die z -Koordinate des Schnittpunktes S_{Z_z} von Bedeutung. Addiert man den Winkel α_{hor} mit dem Winkel β kann man mit Hilfe des Bogenmaßes die x -Koordinate des Punktes S_{Z_a} auf dem aufgeklappten Zylinderpanorama Z_a bestimmen:

$$x_{S_{Z_a}} = \frac{(r_z \cdot \Pi \cdot \alpha_{hor} + \beta)}{180} \quad (3.7)$$

Der Radius r_z des Zylinderpanoramas berechnet sich durch:

$$r_z = \frac{width_{Z_a}}{2 \cdot \Pi} \quad (3.8)$$

$width_{Z_a}$ ist durch die horizontale Auflösung des gerenderten Panoramabildes aus dem 3D Modell gegeben. Die y -Koordinate des Punktes S_{Z_a} auf dem aufgeklappten Zylinderpanorama Z_a kann wiederum mit Hilfe von Gleichsetzen der Strahlgleichung des Strahls \vec{s} mit der Kreisgleichung k des Kreises mit Mittelpunkt M_{Z_z} , die sich durch das Zylinderpanorama Z_z ergibt, berechnet werden:

$$k : (x - 0)^2 - (y - 0)^2 = r_z^2 \quad (3.9)$$

$$\vec{s} = \vec{0} + \lambda \cdot \vec{P}_{E_i} \quad (3.10)$$

Da der Strahl \vec{s} vom Mittelpunkt M_{Z_z} des Zylinderpanoramas aus gesendet wird und der Mittelpunkt im Ursprung $(0 \ 0 \ 0)$ des kartesischen Koordinatensystems liegt, lässt sich der Ortsvektor von \vec{s} auf den Nullvektor vereinfachen. Der Richtungsvektor von \vec{s} hingegen ist durch den aktuellen Punkt P_{E_i} auf der Bildebene E_i gegeben. Es muß letztendlich der Faktor λ berechnet werden, um diesen dann in die Strahlgleichung einzusetzen und so die Koordinaten des Schnittpunktes S_{Z_z} berechnen zu können. Somit ergibt sich für λ :

$$(\lambda \cdot P_{x_i} - 0)^2 - (\lambda \cdot P_{y_i} - 0)^2 = r_z^2 \quad (3.11)$$

$$\lambda = \sqrt{\frac{r_z^2}{(P_{x_i}^2 + P_{y_i}^2)}} \quad (3.12)$$

Die y -Koordinate des Punktes S_{Z_a} ist gleich der z -Koordinate des Punktes S_{Z_z} . Der Punkt S_{Z_a} ist sozusagen der Schnittpunkt des Strahls \vec{s} mit dem Zylinderpanorama Z_z , wenn der Zylinder an der negativen y -Achse aufgeklappt wird. Sind die Koordinaten dieses Punktes bekannt, kann man die Pixelfarbe auf dem aufgeklappten Zylinderpanorama Z_a für den aktuellen Punkt P_{E_i} auf der Ebene E_i nachschlagen und eintragen. Dadurch werden Einzelbilder aus dem 360° Panorama generiert, dessen Geraden keine Krümmungen aufweisen (siehe Abbildung 3.7).

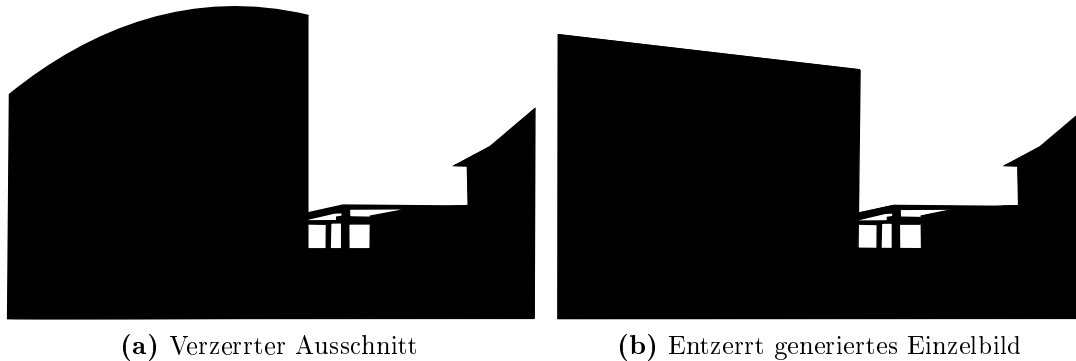


Abbildung 3.7: Zwei Ausschnitte aus dem 360° Panorama, links noch verzerrt und rechts entzerrt.

3.3 Detektion von Eckpunkten und Aufstellung von Hypothesen

Die Detektion von Eckpunkten erfolgt sowohl auf der approximierten Horizontlinie einer realen Aufnahme, als auch auf dem gesamten 360° Zylinderpanorama, nach dem Extrahieren der Silhouetten. Für die Approximation der Horizontlinien wird ein Algorithmus von Frank Schmitt aus der Bildverarbeitungsbibliothek „KIPL“ der AG Priese verwendet. Der Algorithmus bekommt zwei Parameter als Eingabe, die angeben, wie weit oder nah die Annäherung an die Horizontlinie sein soll. Abbildung 3.8 beschreibt den Vergleich zwischen zwei Horizontlinien des gleichen 360° Panoramas nachdem die Silhouette extrahiert worden ist. Für die Detektion der Eckpunkte werden folgende Bedingungen aufgestellt:

- Der Winkel γ zwischen drei aufeinander folgenden Punkten P_1 , P_2 und P_3 in realen Aufnahmen darf mindestens 75° und höchstens 150° betragen.
- Der Winkel γ zwischen drei aufeinander folgenden Punkten P_1 , P_2 und P_3 im 360° Panorama darf mindestens 75° und höchstens 170° betragen.

- Für linke Eckpunkte gilt höchstens eine der folgenden Bedingungen:

$$((P_{2y} \leq P_{1y}) \wedge (P_{2y} \geq P_{3y})) \wedge (P_{1y} > P_{3y}) \quad (3.13)$$

$$((P_{2y} \leq P_{1y}) \wedge (P_{2y} \leq P_{3y})) \wedge (P_{1y} > P_{3y}) \quad (3.14)$$

- Für rechte Eckpunkte gilt höchstens eine der folgenden Bedingungen:

$$((P_{2y} \leq P_{1y}) \wedge (P_{2y} \leq P_{3y})) \wedge (P_{1y} < P_{3y}) \quad (3.15)$$

$$((P_{2y} \geq P_{1y}) \wedge (P_{2y} \leq P_{3y})) \wedge (P_{1y} < P_{3y}) \quad (3.16)$$

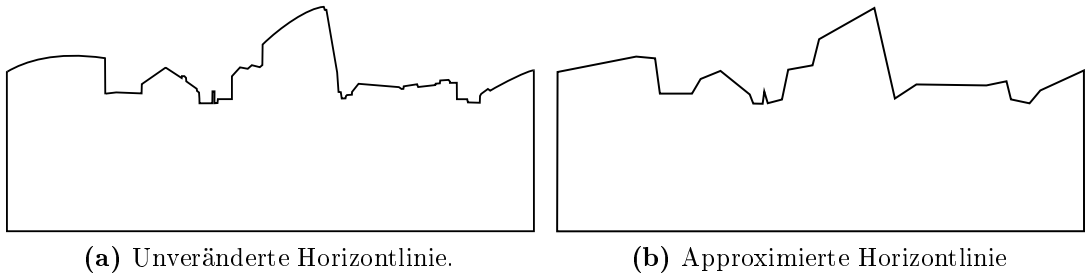


Abbildung 3.8: Der Vergleich zwischen zwei Horizontlinien, links unverändert und rechts approximiert.

Die Einschränkung der Winkelgröße soll vermeiden, dass Eckpunkte von zu spitzen Kanten detektiert werden. Hier gilt die Annahme, dass eine Gebäudekante eher einen weiten Winkel haben wird, als einen spitzen. Experimente haben gezeigt, dass bei realen Aufnahmen, deren Horizontlinie approximiert wird, ein Winkel zwischen 75° und 150° den Anforderungen einer Detektion genügt. Beim Approximieren der Horizontlinie eines Panoramas reicht der Winkel von 75° bis 170° . Berechnen lässt sich der Winkel γ zwischen zwei Vektoren mit:

$$\cos(\gamma) = \frac{\overrightarrow{P_2P_1} \circ \overrightarrow{P_2P_3}}{\|\overrightarrow{P_2P_1}\| \cdot \|\overrightarrow{P_2P_3}\|} \quad (3.17)$$

3.3. DETEKTION VON ECKPUNKTEN UND AUFSTELLUNG VON HYPOTHESEN 35

P_{1y} , P_{2y} und P_{3y} sind jeweils die y -Koordinaten des linken, mittleren und rechten Punktes von drei aufeinanderfolgenden Punkten. Abbildung 3.9 und 3.10 zeigen jeweils die unterschiedlichen Arten von linken und rechten Eckpunkten, es handelt sich dabei nur um vier exemplarische Repräsentanten der unterschiedlichen Ecken, die in realen Aufnahmen oder im Panorama existieren können.

Punkte, die entweder die erste oder zweite Bedingung für linke Eckpunkte erfüllen werden als linker Eckpunkt kategorisiert, entsprechend wird mit Punkten, die eine der beiden Bedingungen für rechte Eckpunkte erfüllen, verfahren. Detektiert man entlang der approximierten Horizontlinie einer Aufnahme, die durch eine Liste von Punktkoordinaten beschrieben wird, entsteht eine Liste L_A mit kategorisierten Punkten. Für diese Punkte werden nun Hypothesen aufgestellt nach denen Punkte im 360° Panorama detektiert werden. Handelt es sich beispielsweise beim ersten Punkt der Liste L_A um einen rechten Eckpunkt, werden im Panorama nur Punkte detektiert, die entweder 3.15 oder 3.16 erfüllen. Diese Punkte kommen in die Liste L_P .

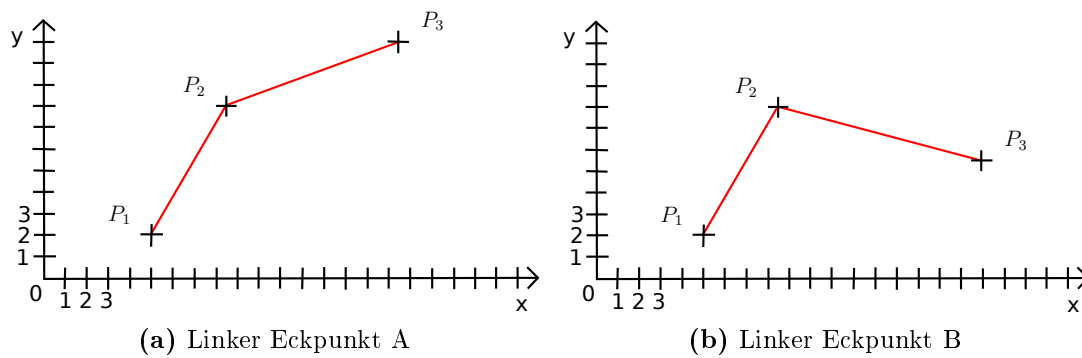


Abbildung 3.9: Die zwei Arten von linken Eckpunkten

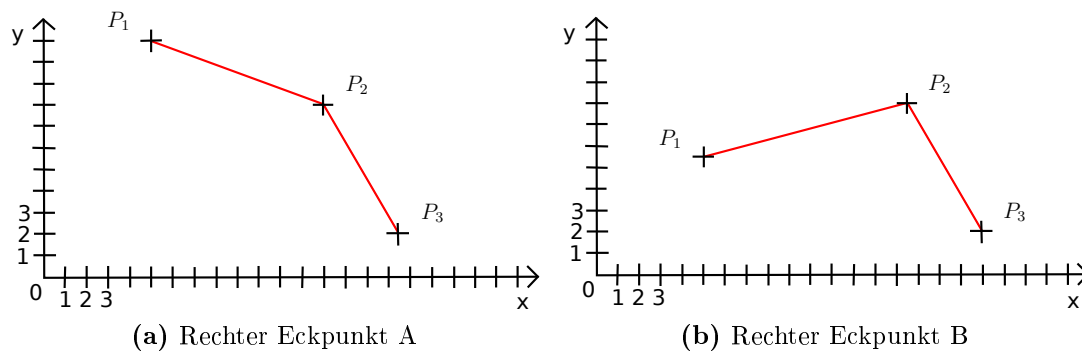


Abbildung 3.10: Die zwei Arten von rechten Eckpunkten

Die Ursache warum gerade Eckpunkte detektiert werden und nicht beispielsweise einfach nur Maxima in einem Bild, liegt in der Krümmung der horizontalen Geraden in einem 360° Panorama. In einer realen Aufnahme oder auf dem generierten Einzelbild des Panoramas existieren diese Krümmungen nicht mehr (siehe Abbildung 3.7). Da aber Punkte auch im Panorama detektiert werden und dieses bis auf die Tatsache, dass die Silhouette extrahiert wird, unverändert bleibt, können bei der Berechnung der initialen Blickrichtung für das nicht lineare Optimierungsverfahren größere Ungenauigkeiten entstehen. Denn das Maximum auf einer Krümmung korrespondiert nicht mit dem Maximum eines künstlich generierten Einzelbildes oder einer realen Aufnahme.

3.4 Berechnung der Blickrichtung

Nach der Detektion von Eckpunkten entstehen zwei Listen mit Eckpunkten. Die Liste L_A beinhaltet die Punktkoordinaten der Eckpunkte, die auf der approximierten Horizontlinie einer realen Aufnahme detektiert wurden. Liste L_P hingegen besteht aus den Punktkoordinaten der Kandidaten, die auf Basis der Hypothesen zu einem Punkt P_{L_A} aus Liste L_A im 360° Panorama detektiert wurden. Jeder Punkt aus Liste L_A hat eine Punktkandidaten Liste L_P . Mit Hilfe eines Punktes aus der Liste L_A und den Punkten der entsprechenden Liste L_P lassen sich Blickrichtungen berechnen, die zur Generierung von Einzelbildern aus dem 360° Panorama benutzt werden. Nimmt man an, dass ein Punkt aus Liste L_A mit seinen Kandidatenpunkten aus der zugehörigen Liste L_P im aufgeklappten Panorama Z_a korrespondiert, liegen beide Punkte auf dem gleichen Strahl \vec{s} (siehe Abbildung 3.11).

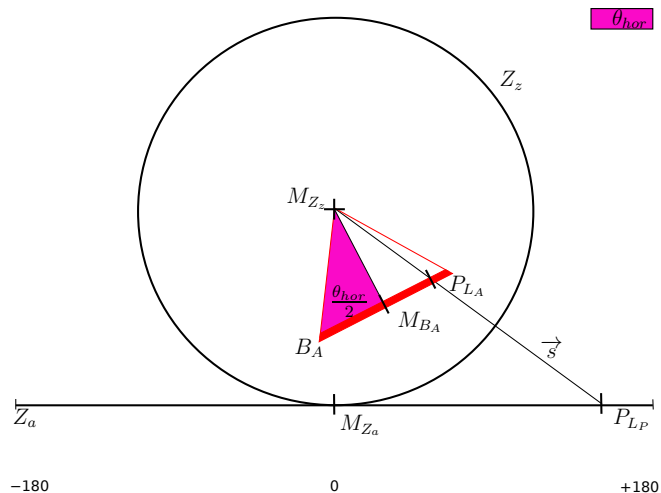


Abbildung 3.11: Korrespondenz zwischen einem Punkt der Liste L_A und einem Punkt der Liste L_P .

Dadurch kann der Winkel φ zwischen einem Punkt P_{L_A} der Liste L_A und dem Mittelpunkt M_{B_A} des Bildes der realen Aufnahme mit dem Winkel κ zwischen den Kandidaten von P_{L_A} aus der Liste L_P und dem Mittelpunkt M_{Z_a} des aufgeklappten Panoramas Z_a verrechnet werden (siehe Abbildung 3.14). Hieraus lässt sich der Winkel α herleiten. Dies geschieht unter der Voraussetzung, dass die Blickrichtung \vec{v} der Vektor vom Ursprung $(0\ 0\ 0)$ des kartesischen Koordinatensystems bis zum Mittelpunkt M_{B_A} des jeweiligen Bildes einer Aufnahme ist. Für die Berechnung des Winkels $\kappa_{hor,ver}$ ist die Strecke zwischen Mittelpunkt M_{Z_a} und dem korrespondierenden Punkt P_{L_P} zu P_{L_A} notwendig (siehe Abbildung 3.12). Der horizontale

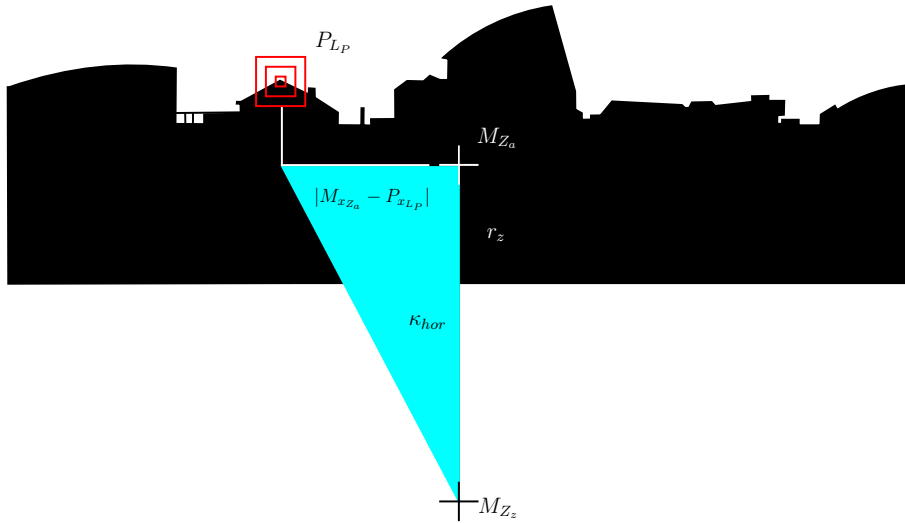


Abbildung 3.12: Berechnung des Winkels κ_{hor} auf dem aufgeklappten Zylinderpanorama Z_a .

Winkel κ_{hor} lässt sich durch:

$$\cos(\kappa_{hor}) = \frac{r_z}{|M_{xZ_a} - P_{xL_P}|} \quad (3.18)$$

bestimmen. Der vertikale Winkel κ_{ver} hingegen durch:

$$\cos(\kappa_{ver}) = \frac{r_z}{|M_{yZ_a} - P_{yL_P}|} \quad (3.19)$$

M_{xZ_a} und M_{yZ_a} sind die x -Koordinate und y -Koordinate des Mittelpunktes M_{Z_a} vom aufgespannten Zylinderpanorama Z_a . Der Mittelpunkt lässt sich durch die

Auflösung von Z_a bestimmen. r_z ist der Radius des Zylinders in Pixel. Angenommen, dass die Auflösung des aufgeklappten Zylinderpanoramas eins zu eins in kartesische Koordinaten umgerechnet wird, entspricht ein Pixel einer Längeneinheit in kartesischen Koordinaten. Dadurch sind keine weiteren Umrechnungen mehr notwendig. Unter der Annahme, dass es sich zwischen dem Punkt P_{L_A} und den Kandidatenpunkten aus L_P um Korrespondenzen handelt, lässt sich das Bild B_A der realen Aufnahme auf die Bildebene E_i abbilden (siehe Abbildung 3.11). Die Bildebene E_i ist eine Längeneinheit im kartesischen Koordinatensystem vom Ursprung $(0\ 0\ 0)$ entfernt. Nach den Formeln 3.18 und 3.19 könnte man auf die gleiche Weise φ_{hor} und φ_{ver} berechnen (siehe Abbildung 3.13). Der Unterschied liegt

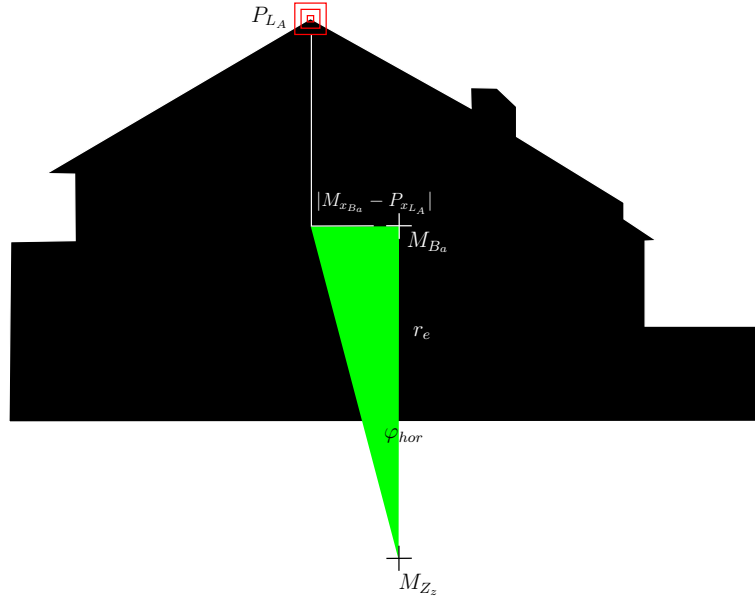


Abbildung 3.13: Berechnung des Winkels φ_{hor} auf dem Bild B_A der realen Aufnahme A .

darin, dass die Länge zwischen dem Punkt P_{L_A} und M_{B_A} in die Längeneinheit des kartesischen Koordinatensystems umgerechnet werden muss. Diese ist in Pixel angegeben und durch die Tatsache, dass die Bildebene nur eine Längeneinheit im kartesischen Koordinatensystem vom Ursprung entfernt ist, kann man nicht eins zu eins umrechnen. Es würden falsche Ergebnisse bei der Winkelberechnung nach Formel 3.18 und 3.19 entstehen. Die Umrechnung lässt sich mit Hilfe der beiden Öffnungswinkel θ_{hor} und θ_{ver} der Kamera berechnen. Die Breite $width_{E_i}$ und die Höhe $height_{E_i}$ der Bildebene E_i ist in kartesischen Längeneinheiten durch:

$$width_{E_i} = \tan\left(\frac{\theta_{hor}}{2}\right) \quad (3.20)$$

$$height_{E_i} = \tan\left(\frac{\theta_{ver}}{2}\right) \quad (3.21)$$

definiert. Damit lassen sich die beiden Faktoren f_{hor} und f_{ver} für Höhe und Breite der Umrechnung mit:

$$f_{hor} = \frac{width_{E_i}}{width_A} \quad (3.22)$$

$$f_{hor} = \frac{height_{E_i}}{height_A} \quad (3.23)$$

berechnen. $width_A$ und $height_A$ ergeben sich durch die Auflösung der realen Aufnahme. Um die Umrechnung abzuschließen muss die Strecke zwischen P_{L_A} und M_{B_A} nur noch mit dem entsprechenden Faktor multipliziert werden.

$$|(M_{x_{B_A}} - P_{x_{L_A}})| \cdot f_{hor} \quad (3.24)$$

$$|(M_{y_{B_A}} - P_{y_{L_A}})| \cdot f_{ver} \quad (3.25)$$

Dies eingesetzt in:

$$\cos(\varphi_{hor}) = \frac{1}{|(M_{x_{B_A}} - P_{x_{L_A}}) \cdot f_{hor}|} \quad (3.26)$$

$$\cos(\varphi_{ver}) = \frac{1}{|(M_{y_{B_A}} - P_{y_{L_A}}) \cdot f_{ver}|} \quad (3.27)$$

ergibt φ_{hor} und φ_{ver} . $M_{x_{B_A}}$ und $M_{y_{B_A}}$ sind die x -Koordinate und y -Koordinate des Mittelpunktes M_{B_A} auf dem Bild B_A der Aufnahme. Zusammen mit den Winkeln κ_{hor} und κ_{ver} können der horizontale α_{hor} und vertikale α_{ver} Winkel der Blickrichtung \vec{v} berechnet werden. Zu diesem Zweck müssen jeweils sechs Fälle für α_{hor} und α_{ver} betrachtet werden:

1.

$$(P_{x_{LP}} > M_{x_{Za}}) \wedge (P_{x_{LA}} > M_{x_{BA}}) \Rightarrow \alpha_{hor} = -\kappa_{hor} + \varphi_{hor} \quad (3.28)$$

2.

$$(P_{x_{LP}} > M_{x_{Za}}) \wedge (P_{x_{LA}} < M_{x_{BA}}) \Rightarrow \alpha_{hor} = -\kappa_{hor} - \varphi_{hor} \quad (3.29)$$

3.

$$(P_{x_{LP}} < M_{x_{Za}}) \wedge (P_{x_{LA}} < M_{x_{BA}}) \Rightarrow \alpha_{hor} = \kappa_{hor} - \varphi_{hor} \quad (3.30)$$

4.

$$(P_{x_{LP}} < M_{x_{Za}}) \wedge (P_{x_{LA}} > M_{x_{BA}}) \Rightarrow \alpha_{hor} = \kappa_{hor} + \varphi_{hor} \quad (3.31)$$

5.

$$(P_{x_{LP}} == M_{x_{Za}}) \wedge (P_{x_{LA}} < M_{x_{BA}}) \Rightarrow \alpha_{hor} = \kappa_{hor} \quad (3.32)$$

6.

$$(P_{x_{LP}} == M_{x_{Za}}) \wedge (P_{x_{LA}} > M_{x_{BA}}) \Rightarrow \alpha_{hor} = -\kappa_{hor} \quad (3.33)$$

Die gleichen sechs Fälle ergeben sich für α_{ver} , nur mit dem Unterschied, dass die y -Koordinate von P_{LP} und P_{LA} , sowie die Mittelpunkte $M_{y_{Za}}$, $M_{y_{BA}}$, als auch die Winkel κ_{ver} und φ_{ver} in die Gleichungen eingehen. Die Mittelpunkte $M_{x_{Za}}$, $M_{y_{Za}}$ und $M_{x_{BA}}$, $M_{y_{BA}}$ sind durch die horizontale und vertikale Auflösung des aufgeklappten Panoramas und des Bildes der realen Aufnahme gegeben. Für $M_{x_{Za}}$ und $M_{x_{BA}}$ ergibt sich beispielsweise:

$$M_{x_{Za}} = \frac{widht_{ZA}}{2} \quad (3.34)$$

$$M_{x_{BA}} = \frac{widht_{BA}}{2} \quad (3.35)$$

Die Notwendigkeit dieser Fälle wird durch die Abbildungen 3.14 bis 3.16 am Beispiel von α_{hor} verdeutlicht. Ohne die Berücksichtigung dieser Fälle wird α_{hor} falsch berechnet. Dies hat zur Folge, dass eine falsche Blickrichtung anhand von zwei korrespondierenden Punkten zur Generierung eines Einzelbildes aus dem 360° Panorama benutzt wird. Das wiederum veranlasst das nicht lineare Optimierungsverfahren dazu die reale Aufnahme mit einem falschen Einzelbild zu vergleichen.

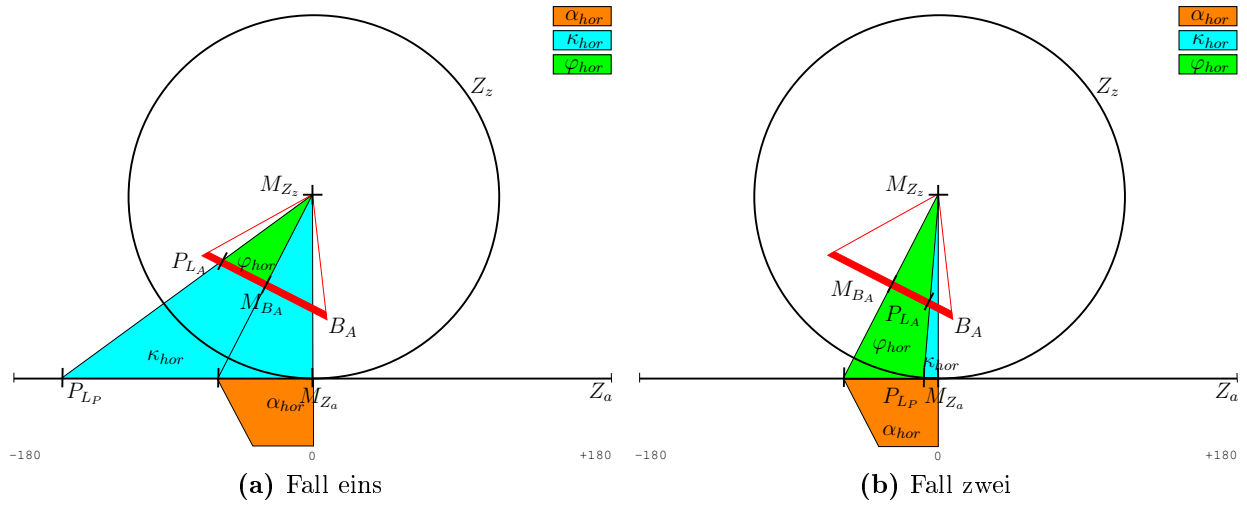


Abbildung 3.14: Fall eins und zwei für die Berechnung von α_{hor} .

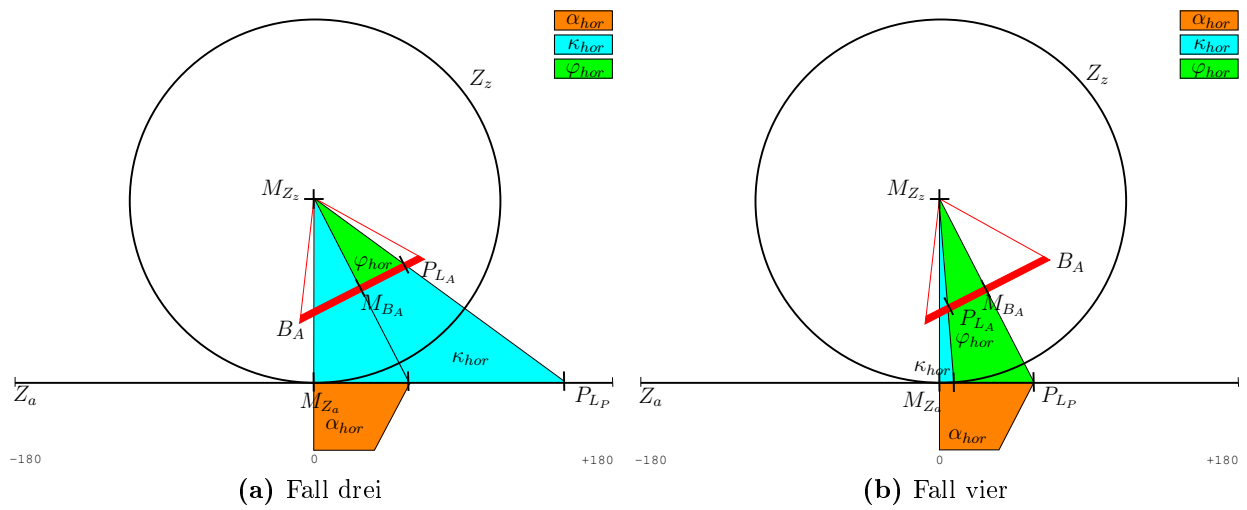


Abbildung 3.15: Fall drei und vier für die Berechnung von α_{hor} .

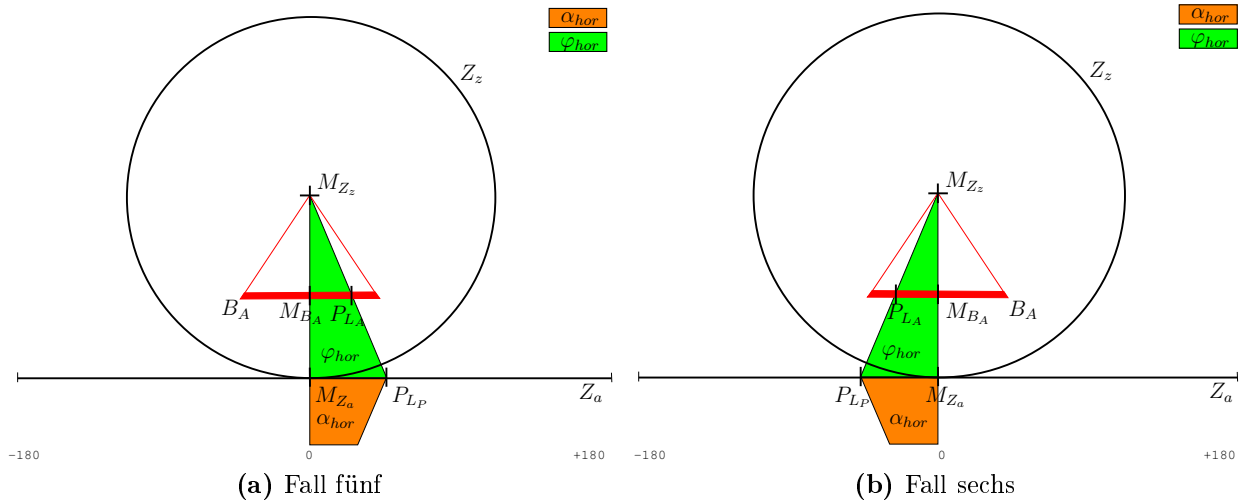


Abbildung 3.16: Fall fünf und sechs für die Berechnung von α_{hor} .

3.5 Das Ähnlichkeitsmaß

Der Vergleich zwischen zwei Bildern erfolgt anhand der Silhouetten. Die Ähnlichkeit zwischen zwei Silhouettenbildern wird aber durch ein Ähnlichkeitsmaß ausgedrückt. Im Falle dieser Qualifikationsarbeit handelt es sich um das XOR-Ähnlichkeitsmaß. Dieses beschreibt die Ähnlichkeit zwischen zwei Bildern, indem die XOR-Operation auf alle Pixel beider Bilder angewendet wird. Aus dem Resultatbild der XOR-Operation lässt sich die Anzahl der weißen Pixel berechnen.

Die Ähnlichkeit wird hierbei anhand der Gesamtanzahl aller Pixel in einem Bild B im Verhältnis zur Anzahl der weißen Pixel durch die Zahlen 0 bis 1 beschrieben. Bei einer Ähnlichkeit von 1 handelt es sich um das gleiche Bild, bei einer Ähnlichkeit von 0 wiederum um zwei komplett unterschiedliche Bilder. Sind beispielsweise in beiden Bildern an der Stelle $B_{(j,i)}$ die Pixel schwarz gefärbt, wird im Resultatbild an der Stelle $B_{(j,i)}$ das Pixel schwarz eingetragen. Die Indizes j und i stehen für die Spalten und Zeilen eines Bildes B . Sollte einer der beiden betrachteten Pixel an der Stelle $B_{(j,i)}$ weiß sein und der andere schwarz, wird im Resultatbild der Pixel an der entsprechenden Stelle weiß gefärbt. Dies gilt auch für den umgekehrten Fall, nur wenn die betrachteten Pixel unterschiedliche Farben aufweisen wird für das Pixel im Resultatbild an der jeweiligen Stelle weiß eingetragen [LHS01]. Durch diesen Zusammenhang ergeben sich Resultatbilder wie in Abbildung 3.17 zu sehen ist.

Die Ursache für die Verwendung dieses Ähnlichkeitsmaßes liegt auf der einen Seite an der einfachen Berechnung der Ähnlichkeit zwischen zwei Bildern und auf der anderen Seite an der Tatsache, dass dieses Ähnlichkeitsmaß eine Eigenschaft

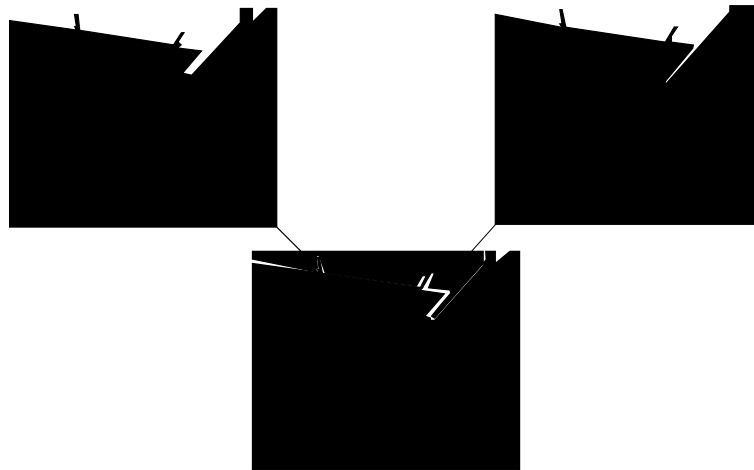


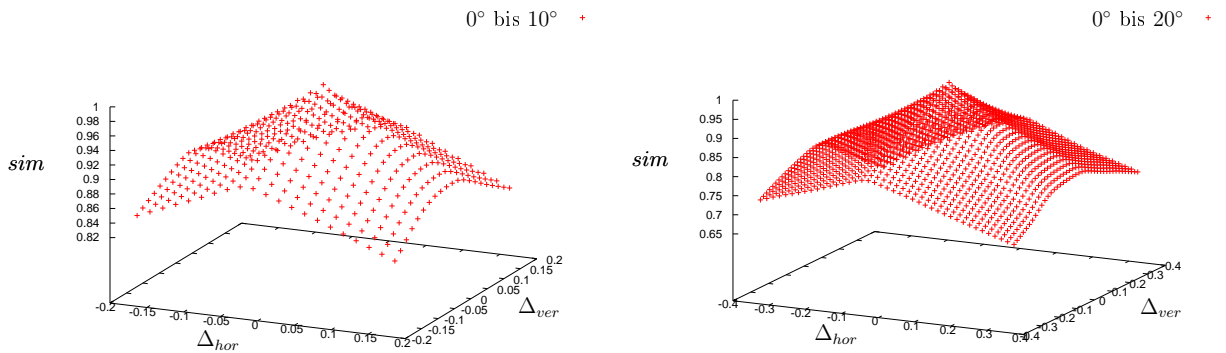
Abbildung 3.17: Die Silhouettenbilder, die miteinander durch die XOR-Operation zu einem Resultatbild verbunden werden.

besitzt, die ein nicht lineares Optimierungsverfahren ausnutzen kann. Nicht lineare Optimierungsverfahren versuchen sozusagen das Minimum, in diesem Fall das Maximum an Ähnlichkeit, zu finden, indem sie sich buchstäblich einen Hügel einzelner Werte entlang hinaufhangeln. Je nachdem ob der aktuelle Wert besser oder schlechter ist als der vorherige, tastet sich das Optimierungsverfahren in der entsprechenden Richtung weiter an dem Hügel entlang. Das Minimum kann hierbei nur erreicht werden, wenn die einzelnen Werte gleichförmig verteilt sind.

Die Abbildungen 3.18 bis 3.20 zeigen die Gleichförmigkeit der Ähnlichkeit der XOR-Operation bezüglich der Abweichung horizontaler und vertikaler Blickrichtung zu einer festgelegten Blickrichtung. Auf der z-Achse ist die Ähnlichkeit zwischen zwei Bildern eingetragen, diese liegt zwischen 0 und 1. Auf der x-Achse ist die Differenz zwischen der festgelegten horizontalen Blickrichtung und den horizontalen Blickrichtungen, die abweichen, eingetragen. Die y-Achse hingegen zeigt die Differenz zwischen der festgelegten vertikalen Blickrichtung und den vertikalen Blickrichtungen, die abweichen.

Wie man an den Abbildungen 3.18 bis 3.19 deutlich erkennen kann ist die Ähnlichkeit noch sehr gleichförmig, während es bei der Abweichung bis zu 50° oder 60° schon ausgeprägte Sprünge gibt. Der oberste Punkt in den Abbildungen 3.18 bis 3.20 würde somit bedeuten, dass das Bild, für die eine Blickrichtung festgelegt wurde, mit sich selbst verglichen wird. In Folge dieser Tatsache ergibt sich eine Ähnlichkeit von $sim = 1$ und eine Differenz in horizontaler und vertikaler Blickrichtung von $\Delta_{hor} = \Delta_{ver} = 0$. Alle anderen Punkte der entsprechenden Abbildung bedeuten, dass das Bild mit festgelegter Blickrichtung mit Bildern verglichen wird, die in horizontaler und vertikaler Blickrichtung abweichen. Somit ergeben

sich unterschiedliche Ähnlichkeiten der Bilder mit unterschiedlichen Differenzen in der Blickrichtung. Theoretisch sollte es dem nicht linearen Optimierungsverfahren anhand der Gleichförmigkeit der Ähnlichkeit bei veränderter Blickrichtung in Abbildung 3.18 bis 3.19 möglich sein, das Optimum - die gesuchte Blickrichtung - noch zu finden. Auch wenn diese bis maximal 40° abweicht.



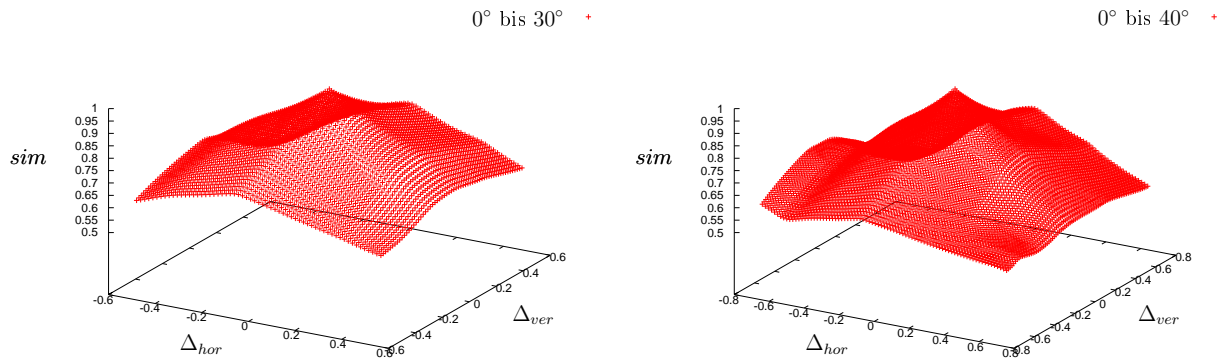
(a) Änderung der Blickrichtung bis maximal 10° . (b) Änderung der Blickrichtung bis maximal 20° .

Abbildung 3.18: Die Ähnlichkeit zwischen einem Bild mit festgelegter Blickrichtung und Bildern, bei denen die Blickrichtung zwischen 0° bis maximal 10° und 20° abweicht.

In Wahrheit spielen die Parameter, die man beispielsweise bei der Implementierung des Levenberg-Marquardt Verfahrens nach [Lou04] mit angeben kann, eine wichtige Rolle. Die Abweichung bis zu 50° und 60° sollte sich auch mit der richtigen Wahl der Parameter als schwierig erweisen. Diese Parameter, auf die im nächsten Kapitel eingegangen wird, entscheiden grundsätzlich darüber, ob das gesuchte Optimum gefunden wird oder nicht, falls eine nicht all zu große Abweichung in der Blickrichtung existiert.

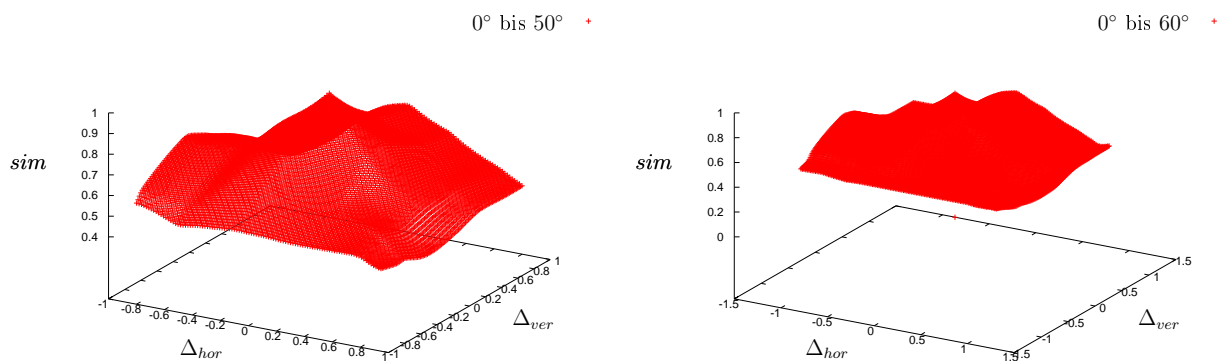
Der Nachteil an diesem Ähnlichkeitsmaß ist die Tatsache, dass die gesamte Anzahl der Pixel beim Vergleich zwischen zwei Bildern in Betracht gezogen wird. Algorithmen, die sich ausschließlich nur auf die Horizontlinie beschränken sind in der Regel beim Vergleichen von Bildern schneller, haben aber den Nachteil, dass Einschränkungen bei den Gebäudestrukturen, die verglichen werden, keine Seltenheit sind. Insbesondere, wenn die Horizontlinie noch approximiert wird, kann es verstärkt zu Mehrdeutigkeiten zwischen komplizierteren Gebäudestrukturen kommen.

Ein weiteres Problem beim Vergleich mit der XOR-Operation ist eine zu hohe Anzahl und Gleichverteilung von schwarzen Pixel im gesamten Bild. Zieht sich beispielsweise gleichförmig ein schwarzer Balken durch die gesamte Länge und $\frac{2}{3}$ der Höhe eines Bildes verursacht dies eine hohe Ähnlichkeit bei Bildern, die aber



(a) Änderung der Blickrichtung bis maximal 30°. (b) Änderung der Blickrichtung bis maximal 40°.

Abbildung 3.19: Die Ähnlichkeit zwischen einem Bild mit festgelegter Blickrichtung und Bildern, bei denen die Blickrichtung zwischen 0° bis maximal 30° und 40° abweicht.



(a) Änderung der Blickrichtung bis maximal 50°. (b) Änderung der Blickrichtung bis maximal 60°.

Abbildung 3.20: Die Ähnlichkeit zwischen einem Bild mit festgelegter Blickrichtung und Bildern, bei denen die Blickrichtung zwischen 0° bis maximal 50° und 60° abweicht.

unterschiedliche Szenen zeigen können. Da sich die Horizontlinien trotz der vielen Gleichanteile von schwarzen und weißen Pixel unterscheiden können.

Desweiteren sind Ähnlichkeiten erst ab einem Wert von $sim \geq 0.85$ interessant, da zwei Bilder sich zu einem gewissen Maß in der Anzahl von gleich farbigen Pixeln überschneiden, ohne sofort die gleiche Szene zeigen zu müssen. Dies kommt insbesondere öfter im ganz unteren und ganz oberen Teil eines Bildes vor.

3.6 Das nicht lineare Optimierungsverfahren

Das Verfahren von Levenberg und Marquardt gehört zu den nicht linearen Optimierungsverfahren, die versuchen Non-Linear Least Squares Probleme zu lösen. Der Unterschied zwischen Least Squares Problemen und Non-Linear Least Squares Problemen liegt darin, dass bei ersteren versucht wird ein lokales Minimum zu finden. Bei letzteren hingegen wird das globale Minimum:

$$x^+ = \operatorname{argmin}_x \{F(x)\} \text{ mit } F : \mathbb{R}^n \mapsto \mathbb{R} \quad (3.36)$$

gesucht. Ein Non-Linear Least Square Problem lässt sich beispielsweise durch das Finden des globalen Minimums x^+ für:

$$F(x) = \frac{1}{2} \sum_{i=1}^m (f_i(x))^2 = \frac{1}{2} \|f(x)\|^2 = \frac{1}{2} f(x)^T f(x) \quad (3.37)$$

definieren. Hierbei ist $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ eine Vektorfunktion mit $m \geq n$. Bevor das nicht lineare Optimierungsverfahren von Levenberg und Marquardt erläutert wird ist die Definition der Ableitungen von F notwendig. Vorausgesetzt, dass f stetige zweite partielle Ableitungen besitzt, kann die Taylorentwicklung auf folgende Weise ausgedrückt werden:

$$f(x+h) = f(x) + \mathbf{J}(x)h + O(\|h\|^2). \quad (3.38)$$

$\mathbf{J} \in \mathbb{R}^{n \times m}$ ist die Jacobi-Matrix. Diese Matrix beinhaltet die ersten partiellen Ableitungen der Funktionskomponenten:

$$(\mathbf{J}(x))_{ij} = \frac{\partial f_i}{\partial d_j}(x). \quad (3.39)$$

Aus der Formel 3.37 folgt, dass:

$$\frac{\partial F}{\partial x_j}(x) = \sum_{i=1}^m f_i(x) \frac{\partial f_i}{\partial x_j}(x) \quad (3.40)$$

gilt. Folglich ist der Gradient definiert durch:

$$F'(x) = \mathbf{J}(x)^T f(x). \quad (3.41)$$

Desweiteren sollte die Hessematrix von F bekannt sein. Aus 3.40 folgt, dass das Element an der Position (j, k) mit:

$$\frac{\partial^2 F}{\partial x_j \partial x_k}(x) = \sum_{i=1}^m \left(\frac{\partial f_i}{\partial x_j}(x) \frac{\partial f_i}{\partial x_k}(x) + f_i(x) \frac{\partial^2 f_i}{\partial x_j \partial x_k}(x) \right) \quad (3.42)$$

gegeben ist. Daraus lässt sich zeigen:

$$F''(x) = \mathbf{J}(x)^T \mathbf{J}(x) + \sum_{i=1}^m f_i(x) f_i''(x). \quad (3.43)$$

Das Levenberg-Marquardt Verfahren kann als eine sogenannte gedämpfte Gauß-Newton Methode kategorisiert werden. Die Gauß-Newton Methode basiert auf der linearen Approximation der Komponenten eines linearen Modells der Funktion f in der Nachbarschaft von x . Es gibt zwei Arten von Methoden, um den Schritt h zu bestimmen (siehe (3.44)). Auf der einen Seite stehen die Trust Region Methoden und auf der anderen die Damped Methoden (gedämpfte Methoden). Grundsätzlich wird auch bei diesen Methoden versucht mit einem Modell $L(h)$ das Verhalten einer Funktion in der Nachbarschaft des gerade iterierten x zu approximieren:

$$F(x+h) \simeq L(h) \equiv F(x) + h^T c + \frac{1}{2} h^T B h. \quad (3.44)$$

Bei Damped Methoden wird der Damping Parameter durch:

$$h = h_{dm} \equiv \operatorname{argmin}_h \left\{ L(h) + \frac{1}{2} \mu h^T h \right\}, \quad (3.45)$$

wobei $\mu \geq 0$ gilt, definiert. Der Term $\frac{1}{2}\mu h^T h = \frac{1}{2}\mu \|h\|^2$ sorgt für die Benachteiligung von großen Schritten h . Das Modell L_h ist eine gute Approximation von $F(x+h)$ wenn h klein ist. Die Qualität eines Modells kann mit der sogenannten Gain Ratio berechnet werden:

$$\varrho = \frac{F(x) - F(x+h)}{L(0) - L(h)}. \quad (3.46)$$

Bei einer Damped Methode bedeutet ein niedriger Wert für ϱ , dass der Damping Faktor erhöht werden sollte, um größere Schritte zu benachteiligen. Ein großer Wert hingegen beinhaltet, dass L_h mit h eine gute Approximation für $F(x+h)$ ist und der Damping Faktor herabgesetzt werden sollte.

Levenberg und Marquardt definieren die Berechnung des Schrittes h_{lm} auf folgende Weise:

$$(\mathbf{J}^T \mathbf{J} + \mu \mathbf{I})h_{lm} = -g \text{ mit } g = \mathbf{J}^T f \text{ und } \mu \geq 0. \quad (3.47)$$

Es gilt $\mathbf{J} = \mathbf{J}(x)$ und $f = f(x)$. Der Damping Faktor hat beim Levenberg-Marquardt Verfahren folgende Effekte:

1. Für alle $\mu > 0$ ist die Koeffizientenmatrix positiv definit, dies garantiert, dass h_{lm} eine steile Richtung hat.
2. Für große Werte von μ bekommt man $h_{lm} \simeq -\frac{1}{\mu}g = -\frac{1}{\mu}F'(x)$, einen kleinen Schritt. Dies ist gut, wenn die momentane Iteration des Optimierungsverfahrens weit von der gesuchten Lösung entfernt ist.
3. Wenn μ sehr klein ist, dann ist $h_{lm} \simeq h_{gn}$. Insbesondere in der finalen Phase der Iterationen, wo x nah an x^+ ist, ist dieser Schritt von Vorteil.

Der Faktor μ beeinflusst somit die Richtung und die Größe des Schrittes h . Dieser Faktor wird während der Iteration mit Hilfe der Gain Ratio, in diesem Fall:

$$\varrho = \frac{F(x) - F(x+h_{lm})}{L(0) - L(h_{lm})} \quad (3.48)$$

aktualisiert. Wie bereits weiter oben erwähnt, beinhaltet ein großer Wert für ϱ , dass $L(h_{lm})$ eine gute Approximation für $F(x+h_{lm})$ ist und man μ herabsetzen kann, um den nächsten Schritt h_{lm} näher an den Gauß-Newton Schritt h_{gn} zu bringen.

Sollte ϱ sehr kleine oder sogar negative Werte aufweisen, ist $L(h_{lm})$ eine schlechte Approximation und man sollte μ erhöhen. Mit der Erhöhung von μ verfolgt man zwei Ziele. Zum einen möchte man mit einem Schritt in eine steilere Richtung weisen und zum anderen versucht man gleichzeitig die Schrittweite zu verkleinern.

Das Abbruch Kriterium bei diesem Algorithmus sollte reflektieren, dass an einem globalen Minimum:

$$F'(x^+) = g(x^+) = 0 \quad (3.49)$$

gilt und man somit $\|g\|_\infty \leq \epsilon_1$ benutzen kann. Wobei ϵ_1 eine kleine, positive Zahl ist, die vom Benutzer eingestellt werden kann. Ein anderes Kriterium sorgt dafür, dass abgebrochen wird wenn die Veränderung in x zu klein sein sollte:

$$\|x_{new} - x\| \leq \epsilon_2(\|x\| + \epsilon_2). \quad (3.50)$$

Desweiteren wird mit $k \geq k_{max}$ dafür gesorgt, dass es nicht zu einer unendlichen Iterationschleife kommt. Sowohl ϵ_2 , als auch k_{max} können vom Benutzer gewählt werden [MNT04]. Der Algorithmus ist in Listing 3.1 grob beschrieben.

begin

$k := 0; \nu := 2; x := x_0$

$\mathbf{A} := \mathbf{J}(x)^T \mathbf{J}(x); g := \mathbf{J}(x)^T f(x)$

$found := (\|g\|_\infty \leq \epsilon_1); \mu := \tau * max\{a_{ij}\} \quad \{\tau \text{ is chosen by the user}\}$

while(not found) **and** ($k < k_{max}$)

$k := k + 1; \text{Solve } (\mathbf{A} + \mu \mathbf{I})h_{lm} = -g$

if $\|h_{lm}\| \leq \epsilon_2(\|x\| + \epsilon_2)$

$found := \mathbf{true}$

else

$x_{new} := x + h_{lm}$

$\varrho := (F(x) - F(x_{new})) / (L(0) - L(h_{lm}))$

if $\varrho > 0 \quad \{\text{step acceptable}\}$

$x := x_{new}$

$\mathbf{A} := \mathbf{J}(x)^T \mathbf{J}(x); g := \mathbf{J}(x)^T f(x)$

$found := (\|g\|_\infty \leq \epsilon_1)$

$\mu := \mu * max\{\frac{1}{3}, 1 - (2\varrho - 1)^3\}; \nu := 2$

else

$\mu := \mu * \nu; \nu := 2 * \nu$

end

Listing 3.1: Der Algorithmus für das nicht lineare Optimierungsverfahren Levenberg-Marquardt [MNT04].

Die eigentliche Aufgabe des nicht linearen Optimierungsverfahrens in dieser Qualifikationsarbeit besteht nun darin Blickrichtungen, die mit Hilfe eines detektierten Punktes P_{L_A} aus der Liste L_A und dessen Korrespondenzpunkten aus der Kandidatenliste L_P berechnet wurden, in mehreren Iterationsschritten anzupassen. Auf Basis dieser Blickrichtungen werden ständig neue Einzelbilder aus dem Panorama generiert und anhand des Ähnlichkeitsmaßes ein Wert für die Ähnlichkeit zwischen den Bildern berechnet. Dadurch kann die Levenberg-Marquardt Methode bestimmen, wie die Blickrichtungen angepasst werden, mit dem Ziel sich der gesuchten Blickrichtung zu nähern.

Dieser Sachverhalt wird in der Abbildung 3.21 dargestellt. Das Einzelbild Nummer eins wird beispielsweise als erstes mit der horizontalen und vertikalen Blickrichtung aus den detektierten Punkten generiert. Diese Bild wird mit dem gesuchten Einzelbild Nummer fünf verglichen und dementsprechend wird die Blickrichtung angepasst, so dass Einzelbild Nummer zwei entsteht. Die Ähnlichkeit zwischen Einzelbild Nummer zwei und Bild Nummer fünf ist aber schlechter als die Ähnlichkeit zwischen Bild eins und Bild fünf, sodass nun die Blickrichtung angepasst wird und das Einzelbild Nummer drei entsteht. Dies geschieht solange, bis das generierte Einzelbild mit dem Silhouettenbild der Aufnahme am ähnlichsten ist - in diesem Fall beim fünften Iterationsschritt.

Es sei angemerkt, dass es sich bei der Abbildung 3.21 nur um einen Sachverhalt handelt und dies nicht der Wirklichkeit entspricht. In der Regel braucht das nicht lineare Optimierungsverfahren mehr Iterationsschritte. Desweiteren ist es so gut wie unmöglich eine Blickrichtung, die so ein weit entferntes Einzelbild, wie Bild Nummer eins generiert, so anzupassen, das es zur richtigen Lösung kommt. Sind die Punktekorrespondenzen gut, ist die berechnete Blickrichtung nah an der gesuchten dran und es kommt zur einer guten Lösung. Da auch das Einzelbild, welches mit Hilfe dieser Blickrichtung generiert wird eine sehr ähnliche Szene zeigen wird, wie sie auf dem Bild der realen Aufnahme zu sehen ist. Die in Kapitel drei verwendeten mathematischen Symbole und Bezeichner sind zusätzlich im Anhang A aufgelistet und kurz erläutert.

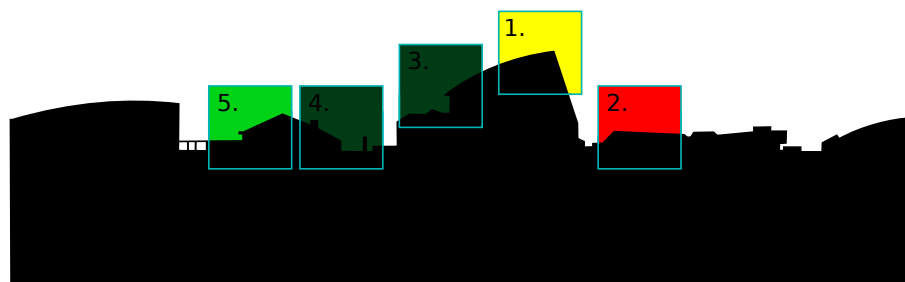


Abbildung 3.21: Das Herantasten durch die Levenberg-Marquardt Methode an die gesuchte Blickrichtung Nummer fünf.

Kapitel 4

Experimente und Ergebnisse

In diesem Kapitel werden die Möglichkeiten und Grenzen des vorgestellten Algorithmus aufgezeigt. Zu diesem Zweck wurden drei Experimente durchgeführt. Es wurden durch den Autor zufällig ausgesuchte Blickrichtungen ausgewählt, mit dessen Hilfe Einzelbilder aus dem entsprechenden 360° Panorama generiert wurden. Diese Einzelbilder sind somit künstlich erzeugt und befinden sich mit Sicherheit in dem entsprechenden Panorama, aus dem sie generiert wurden. Desweiteren wurden fünf möglichst unterschiedliche Einzelbilder eines Panoramas, die zusätzlich meistens zur gesuchten Blickrichtung die geringste Abweichung aufwiesen, nach einer Gaußverteilung verzerrt. Diese Verzerrung soll den Unterschied zwischen künstlich erzeugten Einzelbildern aus dem 3D Modell und Bildern von realen Aufnahmen simulieren. Zum Schluß wurde das Verfahren mit 40 realen Aufnahmen, dessen Positionen im 3D Modell bekannt waren, getestet.

4.1 Implementation

Für die Implementierung wurde zum größten Teil C++ und Qt 4 benutzt. Qt 4 wird dabei für die Entwicklung der Oberfläche verwendet und C++ für die Hauptfunktionalitäten, wie beispielsweise die Detektion der Eckpunkte auf einer Horizontlinie. Die Extraktion der Silhouetten und die Approximierung der Horizontlinie ist mit Hilfe der universitätseigenen Bibliothek KIPL, welche von der Arbeitsgruppe Priese entwickelt wurde, implementiert. Das nicht lineare Optimierungsverfahren ist sowohl in C++, als auch in C geschrieben. Um alle Funktionalitäten der Implementation des nicht linearen Optimierungsverfahren nutzen zu können, sind zudem weitere Bibliotheken notwendig, die in Anhang B kurz erläutert werden. Der Pseudozufallsgenerator für die künstliche Verzerrung von Einzelbildern basiert auf der C++ Bibliothek boost.

4.2 Versuchsreihen und Ergebnisse

Bei der Approximation der Horizontlinie eines Bildes lassen sich der Parameter *maxLineDistance* und *minLineDistance* einstellen. Der erste Parameter hat hierbei Einfluß auf die Genauigkeit, mit der eine Horizontlinie approximiert wird. Dieser legt fest, wie weit die Approximation von der echten Horizontlinie höchstens entfernt sein darf, wohingegen mit dem zweiten Parameter Horizontlinienssegmente mit einer kürzeren Länge verworfen werden. Während der Versuchsreihe mit künstlichen Bildern wurde für das 360° Panorama der Parameter *maxLineDistance* auf 20 und für das generierte Einzelbild aus dem Panorama auf 30 gestellt. In beiden Fällen wurde *minLineDistance* auf 0 gesetzt. Die Ursache für die genauere Approximation im 360° Panorama liegt darin, dass beim Panorama Gebäudestrukturen kleiner abgebildet sind und deswegen genauer approximiert werden müssen, um Kandidatenpunkte nicht zu verfehlen.

In Bildern realer Aufnahmen sind die Horizontlinien - beispielsweise durch das Auftreten von Vegetation - nicht unbedingt fortlaufend. Um Löcher in der Horizontlinie zu vermeiden, wird diese etwas genauer approximiert als bei künstlich generierten Einzelbildern aus dem Panorama. Der Parameter *maxLineDistance* ist hierbei auf 25 gesetzt und *minLineDistance* auf 0. Trotz der Tatsache, dass die Verzerrung bei künstlich generierten Einzelbildern den Unterschied zwischen künstlichen und realen Bildern simulieren soll, ist ein Anpassen des Parameters *maxLineDistance* nicht nötig. Denn die Verzerrung kann nie so stark sein, dass Löcher in der Horizontlinie entstehen könnten.

Während der gesamten Versuchsreihe wurden für das nicht lineare Optimierungsverfahren die Parameter nicht geändert. Der Startwert für den Faktor μ wurde beispielsweise auf $\mu = 0.01$ gesetzt. Desweiteren werden Blickrichtungen durch das nicht lineare Optimierungsverfahren bei der Versuchsreihe mit realen Aufnahmen erst ab einer Ähnlichkeit von $sim = 0.80$ zwischen dem Bild der realen Aufnahme und dem generierten Einzelbild angepasst. Bei den zwei anderen Versuchsreihen ist die Ähnlichkeit auf $sim = 0.85$ gesetzt. Dieser Unterschied liegt daran, dass Bilder realer Aufnahmen niemals 100 prozentig gleich sein können wie die Bilder aus dem 3D Modell. Die Folge ist ein niedrigerer Startwert für die Ähnlichkeit zwischen zwei Bildern, damit das nicht lineare Optimierungsverfahren mit der Anpassung der entsprechenden Blickrichtungen beginnt.

4.2.1 Künstlich erzeugte Einzelbilder

Die Tabellen 4.1 bis 4.4 zeigen die Ergebnisse aus vier gerenderten Panoramen zu denen jeweils zehn künstlich erzeugte Blickrichtungen betrachtet wurden. Aufgrund der numerischen Genauigkeit des nicht linearen Optimierungsverfahrens werden die Schritte am Ende einer Iteration so klein, dass sich die Ähnlichkeit nur

noch bei Nachkommastellen ändert. Ist diese Änderung zu klein, terminiert das Verfahren im besten Fall mit einer Ähnlichkeit von $sim = 0.9999$ anstelle einer Ähnlichkeit von $sim = 1.0$. Dies ändert nichts an der Tatsache, dass die gesuchte Szene einer realen Aufnahme bei geringer Abweichung der Blickrichtungen als gefunden betrachtet werden kann. Da die Einzelbilder aus dem Panorama künstlich erzeugt wurden, werden sie sich mit Sicherheit im entsprechenden Panorama befinden. Dies erklärt auch die durchgehend hohen Werte in der Ähnlichkeit und meistens niedrigen Werte der Differenz zwischen gesuchter und gefundener Blickrichtung.

Tabelle 4.1 bildet bis auf Bild Nummer zehn keine Ausnahme. Trotz der hohen Differenz in horizontaler $\Delta_{hor} = 9.213^\circ$ und vertikaler $\Delta_{ver} = 4.5936^\circ$ Blickrichtung ergibt sich eine Ähnlichkeit von $sim = 0.991155$ zwischen den beiden Bildern. Dies lässt sich dadurch erklären, dass die Horizontlinie im generierten Einzelbild nur bis zu $\frac{1}{3}$ Länge des Bildes geht und der Rest des Bildes schwarz ist. Aufgrund des hohen Anteils von schwarz in den beiden Bildern entsteht eine zu hohe Ähnlichkeit. Da die gefundenen Blickrichtungen α_{hor}^{res} und α_{ver}^{res} trotzdem ein Einzelbild generieren, das im Prinzip die gleiche Szene zeigt - nur ein wenig verschoben und ohne prägnante Horizontlinie - bleibt die Tatsache, dass $\frac{2}{3}$ des Bildes schwarz ist, erhalten. Dies führt zu einer hohen Ähnlichkeit ungeachtet der Verschiebung zwischen den beiden Bildern.

Tabelle 4.2 zeigt die Ergebnisse des zweiten Panoramas. Auffallend sind Bild Nummer zwei, vier und neun. Bei diesen Bildern konnten keine markanten Eckpunkte in den künstlich generierten Einzelbildern detektiert werden. Dies lag zum einen an der Approximation der Horizontlinie und zum anderen an den Gebäudestrukturen innerhalb der Einzelbilder. Bild vier und neun zeigen beispielsweise eher gerade Gebäudestrukturen, die keine Eckpunkte aufweisen. Ändert man die Parameter der Approximation ergeben sich zumindestens folgende Ergebnisse für Bild Nummer zwei:

- $\alpha_{hor}^{res} = -69.8955^\circ$
- $\alpha_{ver}^{res} = 14.995^\circ$
- $\Delta_{hor} = 0.1045$
- $\Delta_{ver} = 0.005$
- $sim = 0.9995$

Im Gegensatz zu Tabelle 4.4 zeigt Tabelle 4.3 keine Auffälligkeiten, hier wurden sämtliche Szenen der Testbilder gefunden. In Bild neun der Tabelle 4.4 wurde fälschlicherweise ein Punkt im gesuchten Einzelbild als rechter Eckpunkt kategorisiert. Dies führte aufgrund der damit verbundenen Hypothesen zur Detektion

von rechten Eckpunkten im 360° Panorama. Das Einzelbild, welches mit Hilfe des falsch kategorisierten Punktes und seinem korrespondierenden Kandidatenpunkt generiert wurde, zeigte eine ähnliche Horizontlinie, so dass eine hohe Ähnlichkeit bei eigentlich zwei unterschiedlichen Szenen entstand (siehe Abbildung 4.1). Passt man die Parameter für die Approximation einer Horizontlinie an, führt das zur richtigen Kategorisierung des betroffenen Punktes und damit zu folgenden Ergebnissen:

- $\alpha_{hor}^{res} = -120.923^\circ$
- $\alpha_{ver}^{res} = 20.1433^\circ$
- $\Delta_{hor} = 0.923$
- $\Delta_{ver} = 0.1433$
- $sim = 0.9995$

Leider ergab die Anpassung der Levmar-Parameter keine Verbesserung, so dass die Differenz der horizontalen Blickrichtung vergleichsweise höher ist als in anderen Testbildern. Grundsätzlich mussten die Levmar-Parameter bei dieser Versuchsreihe nicht angepasst werden. Es war dem nicht linearen Optimierungsverfahren ohne weiteres möglich die gesuchten Szenen der künstlich generierten Testbilder zu finden. Die Hauptvoraussetzung hierbei ist, dass die Gebäudestrukturen eine Detektion, nach den in Kapitel 3.3 beschriebenen Kriterien, von Eckpunkten zulassen.

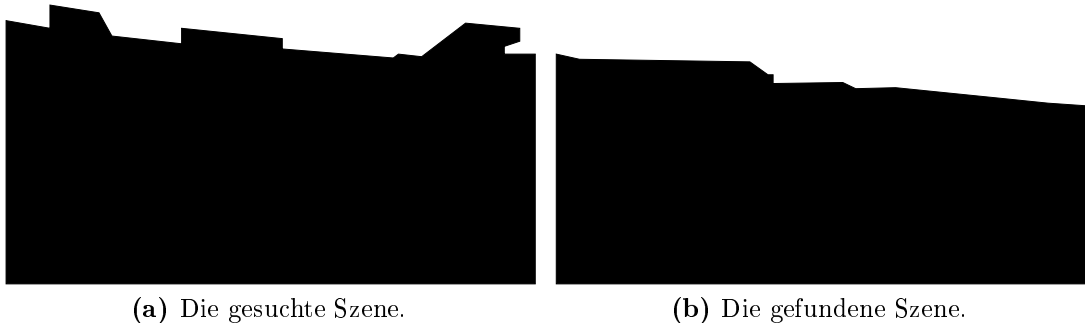


Abbildung 4.1: Zwei unterschiedliche Szenen, die aber durch das Ähnlichkeitsmaß einen hohen Wert für die Ähnlichkeit zugewiesen bekommen.

Panorama 1	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	-10.0°	30.0°	-10.0023°	30.071°	0.0023°	0.071°	0.9991
Bild 2	-90.0°	13.0°	-90.0545°	-13.0067°	0.0545°	0.0067°	0.9992
Bild 3	15.0°	30.0°	15.004°	29.9499°	0.004°	0.0501°	0.9992
Bild 4	80.0°	27.0°	80.0°	27.0051°	0.0°	0.0051°	0.9998
Bild 5	140.0°	15.0°	139.989°	14.9999°	0.011°	0.0001°	0.9999
Bild 6	112.0°	27.0°	111.981°	27.0008°	0.019°	0.0008°	0.9998
Bild 7	-55.0°	15.0°	-54.9997°	15.0°	0.0003°	0.0°	0.9999
Bild 8	-80.0°	30.0°	-79.9971°	29.9911°	0.0029°	0.0089°	0.9997
Bild 9	-90.0°	-15.0°	-89.9976°	-15.0001°	0.0024°	0.0001°	0.9999
Bild 10	-180°	-15.0°	-189.213°	-10.4064°	9.213°	4.5936°	0.9911

Tabelle 4.1: Künstlich generierte Einzelbilder aus dem Panorama eins.

Panorama 2	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	125.0°	15.0°	125.001°	14.999°	0.001°	0.0001°	0.9998
Bild 2	-70.0°	15.0°	0.0°	0.0°	70°	15.0°	0.0
Bild 3	205.0°	20.0°	205.002°	20.01°	0.002°	0.01°	0.9998
Bild 4	90.0°	25.0°	0.0°	0.0°	90.0°	25.0°	0.0
Bild 5	20.0°	35.0°	20.565°	35.4808°	0.565°	0.4808°	0.9904
Bild 6	140.0°	30.0°	140°	30.0024°	0.0°	0.0024°	0.9998
Bild 7	-13.0°	23.0°	-13.002°	23.0003°	0.002°	0.0003°	0.9999
Bild 8	120.0°	23.0°	119.997°	23.0007°	0.003°	0.0007°	0.9998
Bild 9	60.0°	35.0°	0.0°	0.0°	60.0°	35.0°	0.0
Bild 10	-50°	0.0°	-50.0005°	-0.00177°	0.0005°	0.00177°	0.9999

Tabelle 4.2: Künstlich generierte Einzelbilder aus dem Panorama zwei.

Panorama 3	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	25.0°	25.0°	25.0004°	25.0014°	0.0004°	0.0014°	0.9998
Bild 2	120.0°	33.0°	120.002°	32.9994°	0.002°	0.0006°	0.9999
Bild 3	100.0°	20.0°	99.9991°	20.0077°	0.0009°	0.0077°	0.9995
Bild 4	140.0°	20.0°	140.012°	20.041°	0.012°	0.041°	0.9994
Bild 5	0.0°	20.0°	0.00198°	20.0005°	0.00198°	0.0005°	0.9999
Bild 6	-30.0°	30.0°	-30.0008°	30.0011°	0.0008°	0.0011°	0.9999
Bild 7	-50.0°	29.0°	-50.0008°	28.9865°	0.0008°	0.0135°	0.9998
Bild 8	-80.0°	16.0°	-80.0018°	16.0032°	0.0018°	0.0032°	0.9998
Bild 9	-120.0°	10.0°	-120.001°	10.0°	0.001°	0.0°	0.9999
Bild 10	40°	40.0°	40.0005°	39.9729°	0.0005°	0.0271°	0.9993

Tabelle 4.3: Künstlich generierte Einzelbilder aus dem Panorama drei.

Panorama 4	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	-130.0°	10.0°	-129.598°	10.0151°	0.402°	0.0151°	0.9960
Bild 2	-70.0°	12.0°	-69.9979°	11.9988°	0.0021°	0.0012°	0.9999
Bild 3	-80.0°	20.0°	-79.9831°	19.9982°	0.0169°	0.001°	0.9998
Bild 4	-90.0°	10.0°	-90.2815°	10.0063°	0.2815°	0.0063°	0.9979
Bild 5	-20.0°	13.0°	-20.0371°	-12.9999°	0.0371°	0.0001°	0.9996
Bild 6	40.0°	-3.0°	39.9798°	-3.00047°	0.0202°	0.00047°	0.9999
Bild 7	-80.0°	0.0°	-79.8175°	0.00914°	0.1825°	0.00914°	0.9982
Bild 8	-170.0°	22.0°	-170.015°	21.9904°	0.015°	0.0096°	0.9994
Bild 9	-120.0°	20.0°	-205.788°	-3.58359°	85.888°	23.58259°	0.9599
Bild 10	90°	0.0°	-91.4359°	0.0595°	-1.4359°	0.0595°	0.9901

Tabelle 4.4: Künstlich generierte Einzelbilder aus dem Panorama vier.

4.2.2 Künstlich verzerrte Einzelbilder

Die nächste Versuchsreihe besteht aus künstlich generierten Einzelbildern, die mit Hilfe eines Pseudozufallsgenerators nach einer Gaußverteilung verändert wurden. Hierbei wurde entlang der Breite eines Bildes jeweils das erste Pixel an der Horizontlinie gesucht und dessen Position mit dem Pseudozufallsgenerator entweder nach oben oder unten verlagert. Dementsprechend wurden alle Pixel unter diesem ersten Pixel schwarz gefärbt (siehe Abbildung 4.2). Die Veränderung der Horizontlinie ist von der Varianz σ^2 abhängig und tritt in den folgenden Testbildern in drei Stufen auf:

- Minimale Varianz $\sigma = 1$
- Mittlere Varianz $\sigma = 7$
- Maximale Varianz $\sigma = 15$

Es wurden jeweils möglichst fünf unterschiedliche Einzelbilder aus der vorherigen Versuchsreihe für ein Panorama ausgewählt. Die Ursache der Verzerrung liegt darin begründet, dass Bilder realer Aufnahmen sich mit Sicherheit von den Einzelbildern des Panoramas aus dem 3D Modell unterscheiden werden. Diese Feststellung wird von der Tatsache unterstützt, dass beispielsweise das 3D Modell der Universität Koblenz nur eingeschränkt Vegetation aufweist. Um herauszufinden ob der Algorithmus rein theoretisch überhaupt auf Bildern realer Aufnahmen funktionieren kann, wird der Unterschied zwischen realen Aufnahmen und künstlich generierten Bildern durch die Veränderung mit dem Pseudozufallsgenerator simuliert.

Die Tabellen 4.5 bis 4.7 zeigen die Ergebnisse für das erste Panorama. Die Differenz der Blickrichtungen ergibt sich aus dem Unterschied zu den jeweiligen

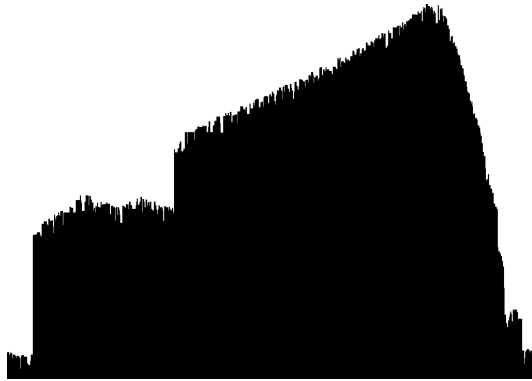


Abbildung 4.2: Das Bild einer Silhouette verzerrt mit maximaler Varianz von $\sigma^2 = 15$.

Bildern ohne Verzerrung aus. Es lässt sich annehmen, dass je stärker die Bilder verzerrt werden, desto weiter die horizontale und vertikale Blickrichtung von Bildern ohne Verzerrung entfernt ist und desto kleiner der Wert für die Ähnlichkeit zwischen diesen Bildern ist. Die Abbildung 4.3 fasst diesen Zusammenhang für alle fünf Testbilder des ersten Panoramas zusammen. Hier ist auf der x-Achse die Varianz $\sigma^2 = 0; 1; 7; 15$ und auf der y-Achse die Summe der Differenz der horizontalen und vertikalen Blickrichtung aufgetragen. Einzige Auffälligkeit bietet das Bild Nummer sieben. Hier ist deutlich zu sehen, dass die Summe der Differenz eines Bildes mit mittlerer Varianz verzerrt höher ist als die Summe der Differenz bei einer Verzerrung mit maximaler Varianz. Dies ist insbesondere der geringen horizontalen Differenz bei maximaler Verzerrung zu verdanken.

Die Ursache für dieses Verhalten liegt darin, dass das nicht lineare Optimierungsverfahren bei maximaler Verzerrung viel schneller an einen Punkt kommt, wo sich die Ähnlichkeit in den Nachkommastellen bei Veränderung der horizontalen und vertikalen Blickrichtung nicht mehr ändert und somit früher abbricht. Es ist reiner Zufall, dass dies in diesem Fall eine genauere horizontale Blickrichtung zur Folge hat.

Panorama 1	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	-10.0°	30.0°	-9.99963°	30.2586°	0.00037°	0.2586°	0.9937
Bild 5	140.0°	15.0°	140.016°	15.1859°	0.016°	0.1859°	0.9991
Bild 6	112.0°	27.0°	112.275°	27.1345°	0.275°	0.1345°	0.9954
Bild 7	-55.0°	15.0°	-54.9905°	15.0591°	0.0095°	0.0591°	0.9980
Bild 9	-90.0°	-15.0°	-89.9974°	-14.8652°	0.0026°	0.1348°	0.9973

Tabelle 4.5: Die fünf Einzelbilder aus dem Panorama eins mit minimaler Varianz $\sigma^2 = 1$ verzerrt.

Panorama 1	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	-10.0°	30.0°	-10.0329°	30.5306°	0.0329°	0.5306°	0.9903
Bild 5	140.0°	15.0°	139.878°	15.4621°	0.122°	0.4621°	0.9950
Bild 6	112.0°	27.0°	112.275°	27.5237°	0.275°	0.5237°	0.9924
Bild 7	-55.0°	15.0°	-54.8961°	15.7129°	0.1039°	0.7129°	0.9980
Bild 9	-90.0°	-15.0°	-89.9547°	-14.5492°	0.0453°	0.4508°	0.9944

Tabelle 4.6: Die fünf Einzelbilder aus dem Panorama eins mit mittlerer Varianz $\sigma^2 = 7$ verzerrt.

Panorama 1	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	-10.0°	30.0°	-9.91954°	31.0535°	0.08046°	1.0535°	0.9847
Bild 5	140.0°	15.0°	140.042°	15.9856°	0.042°	0.985°	0.9914
Bild 6	112.0°	27.0°	111.907°	28.8511°	0.093°	1.8511°	0.9879
Bild 7	-55.0°	15.0°	-55.0399°	15.7156°	0.0399°	0.7156°	0.9886
Bild 9	-90.0°	-15.0°	-89.893°	-14.1588°	0.107°	0.8412°	0.9893

Tabelle 4.7: Die fünf Einzelbilder aus dem Panorama eins mit maximaler Varianz $\sigma^2 = 15$ verzerrt.

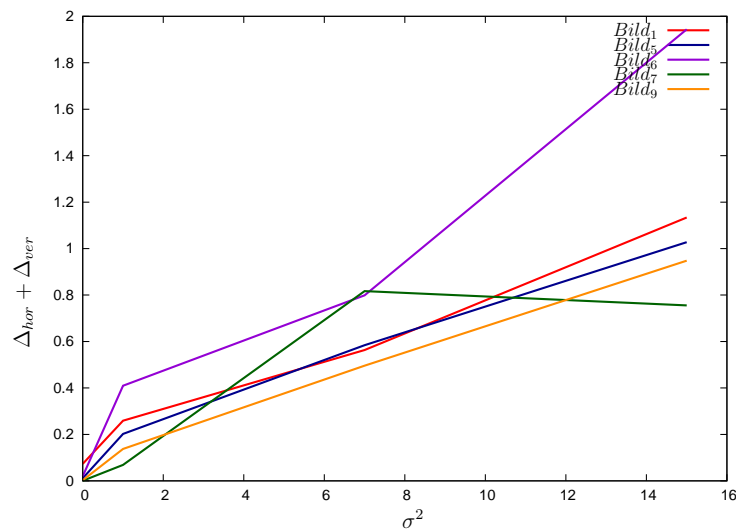


Abbildung 4.3: Zusammenfassung der Ergebnisse des ersten Panoramas mit künstlich verzerrten Bildern.

Die Tabellen 4.8 bis 4.13 zeigen die Ergebnisse für das zweite und dritte Panorama. Es lassen sich hierbei keine besonderen Auffälligkeiten beobachten (siehe Abbildung 4.4 und 4.5). Je stärker die Verzerrung wird, desto größer ist die Summe der horizontalen und vertikalen Differenz in der Blickrichtung und gleichzeitig wird der Wert für die Ähnlichkeit kleiner. Wie man deutlich in Abbildung 4.4 erkennen kann muß der Anstieg der Differenz bei steigender Verzerrung bei weitem nicht gleichförmig steigend sein - es sind durchaus größere Sprünge möglich. Da die Verzerrung mit einem Psdeudozufallsgenerator künstlich generiert wurde sind solche „Knicke“ in den Abbildungen keine Seltenheit. Dessen ungeachtet zeigen die Ergebnisse des zweiten und dritten Panoramas ein erwartendes Verhalten bezüglich der künstlichen Verzerrung.

Panorama 2	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	125.0°	15.0°	124.949°	14.8621°	0.051°	0.1379°	0.9946
Bild 3	205.0°	20.0°	204.3°	20.0502°	0.07°	0.0502°	0.9945
Bild 6	140.0°	30.0°	140.215°	30.2473°	0.215°	0.2473°	0.9969
Bild 7	-13.0°	23.0°	-12.999°	23.2006°	0.0001°	0.2006°	0.9964
Bild 8	120.0°	23.0°	119.993°	23.2095°	0.007°	0.2095°	0.9981

Tabelle 4.8: Die fünf Einzelbilder aus dem Panorama zwei mit minimaler Varianz $\sigma^2 = 1$ verzerrt.

Panorama 2	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	125.0°	15.0°	125.108°	15.4146°	0.108°	0.4146°	0.9948
Bild 3	205.0°	20.0°	205.012°	20.6067°	0.012°	0.6067°	0.9951
Bild 6	140.0°	30.0°	140.027°	140.027°	0.027°	0.478°	0.9944
Bild 7	-13.0°	23.0°	-12.9005°	23.9701°	0.0995°	0.9701°	0.9915
Bild 8	120.0°	23.0°	120.101°	23.6767°	0.101°	0.6767°	0.9949

Tabelle 4.9: Die fünf Einzelbilder aus dem Panorama zwei mit mittlerer Varianz $\sigma^2 = 7$ verzerrt.

Panorama 2	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	125.0°	15.0°	125.034°	15.9912°	0.034°	0.991216°	0.9901
Bild 3	205.0°	20.0°	205.245°	21.1129°	0.245°	1.1129°	0.9908
Bild 6	140.0°	30.0°	140.07°	31.6008°	0.07°	1.6008°	0.9904
Bild 7	-13.0°	23.0°	-12.8872°	24.2411°	0.1128°	1.2411°	0.9887
Bild 8	120.0°	23.0°	120.003°	23.9384°	0.003°	1.0616°	0.9904

Tabelle 4.10: Die fünf Einzelbilder aus dem Panorama zwei mit maximaler Varianz $\sigma^2 = 15$ verzerrt.

Panorama 3	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	25.0°	25.0°	25.0147°	25.1465°	0.0147°	0.1465°	0.9957
Bild 2	120.0°	33.0°	120.076°	33.1996°	0.076°	0.1996°	0.9986
Bild 3	100.0°	20.0°	100.013°	20.2041°	0.013°	0.2041°	0.9961
Bild 5	0.0°	20.0°	-0.00884°	20.1459°	0.00884°	0.1459°	0.9980
Bild 6	-30.0°	30.0°	-29.994°	30.2017°	0.006°	0.2017°	0.9976

Tabelle 4.11: Die fünf Einzelbilder aus dem Panorama drei mit minimaler Varianz $\sigma^2 = 1$ verzerrt.

Panorama 3	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	25.0°	25.0°	25.084°	25.6795°	0.084°	0.6795°	0.9919
Bild 2	120.0°	33.0°	120.002°	33.8216°	0.002°	0.8216°	0.9955
Bild 3	100.0°	20.0°	99.9528°	20.4648°	0.0472°	0.4648°	0.9931
Bild 5	0.0°	20.0°	0.01091°	20.4615°	0.01091°	0.4615°	0.9947
Bild 6	-30.0°	30.0°	-29.989°	30.533°	0.011°	0.533°	0.9945

Tabelle 4.12: Die fünf Einzelbilder aus dem Panorama drei mit mittlerer Varianz $\sigma^2 = 7$ verzerrt.

Panorama 3	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	25.0°	25.0°	25.0137°	26.0837°	0.0137°	1.0837°	0.9919
Bild 2	120.0°	33.0°	120.008°	34.8034°	0.008°	1.8034°	0.9905
Bild 3	100.0°	20.0°	100.014°	21.3724°	0.014°	1.3724°	0.9884
Bild 5	0.0°	20.0°	0.0012°	20.9806°	0.0012°	0.9806°	0.9905
Bild 6	-30.0°	30.0°	-29.9588°	30.6641°	0.0412°	0.6641°	0.9889

Tabelle 4.13: Die fünf Einzelbilder aus dem Panorama drei mit maximaler Varianz $\sigma^2 = 15$ verzerrt.

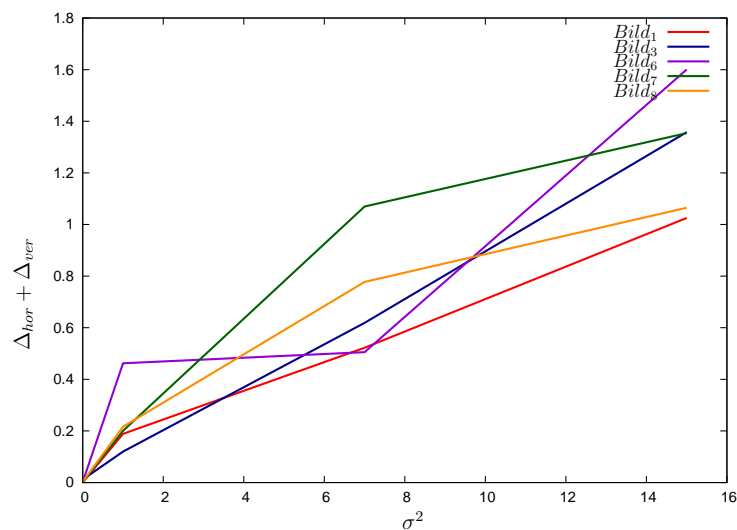


Abbildung 4.4: Zusammenfassung der Ergebnisse des zweiten Panoramas mit künstlich verzerrten Bildern.

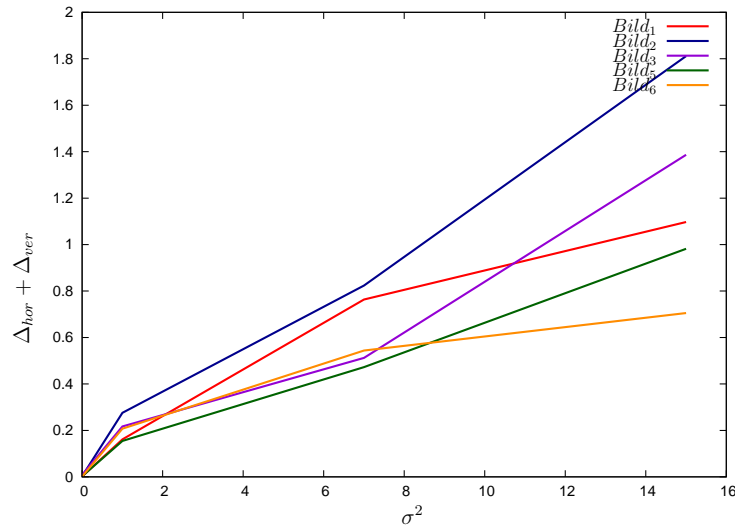


Abbildung 4.5: Zusammenfassung der Ergebnisse des dritten Panoramas mit künstlich verzerrten Bildern.

Abbildung 4.6 zeigt die Zusammenfassung der Ergebnisse für das vierte Panorama. Die einzige Auffälligkeit bildet dabei das Bild Nummer eins. Wie man deutlich erkennen kann weicht die Summe der Differenz in horizontaler und vertikaler Blickrichtung ohne Verzerrung schon stärker von der gesuchten Blickrichtung ab als in allen Testbildern davor.

Desweiteren ist die Summe der Differenz bei mittlerer Verzerrung im Gegensatz zu minimaler Verzerrung geringer. Bei der Versuchsreihe mit mittlerer Verzerrung hatte der Algorithmus bei diesem Bild Probleme einen Punkt richtig zu kategorisieren. Auf Basis der Kategorisierung dieses Punktes wurden für die Detektion falsche Hypothesen aufgestellt, so dass ein falsches Einzelbilder generiert wurde, dessen Ähnlichkeit zufälligerweise aber sehr hoch war. Nach Anpassen der Parameter für die Approximation der Horizontlinie wurde der entsprechende Eckpunkt richtig kategorisiert. Zusammen mit einem Kandidatenpunkt aus dem 360° Panorama wurde eine Blickrichtung berechnet, die sich durch das nichtlineare Optimierungsverfahren besser an die gesuchte Blickrichtung anpassen lies, als bei minimaler Verzerrung.

Interessanterweise handelt es sich bei dem Kandidatenpunkt des Panoramas sowohl bei minimaler als auch bei mittlerer Verzerrung um den gleichen Punkt. Aufgrund der Verzerrung unterscheidet sich nur die Koordinate des detektierten Punktes im verzerrten Einzelbild und somit die geschätzte Blickrichtung. Diese ermöglicht es dem nicht linearen Optimierungsverfahren sich besser an die gesuchte Blickrichtung anzunähern.

Panorama 4	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	-130.0°	10.0°	-129.567°	10.1322°	0.433°	0.1322°	0.9950
Bild 2	-70.0°	12.0°	-69.9417°	12.1743°	0.0583°	0.1743°	0.9977
Bild 5	20.0°	13.0°	-20.1551°	-12.8254°	0.1551°	0.1746°	0.9959
Bild 6	-40.0°	-3.0°	40.0635°	-2.82987°	0.0635°	0.17013°	0.9984
Bild 8	-170.0°	22.0°	-170.299°	22.0956°	0.299°	0.0956°	0.9961

Tabelle 4.14: Die fünf Einzelbilder aus dem Panorama vier mit minimaler Varianz $\sigma^2 = 1$ verzerrt.

Panorama 4	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	-130.0°	10.0°	-129.996°	10.4353°	0.004°	0.4353°	0.9944
Bild 2	-70.0°	12.0°	-69.9531°	12.5488°	0.0469°	0.5488°	0.9941
Bild 5	20.0°	13.0°	-20.0161°	-12.4961°	0.0161°	0.5039°	0.9939
Bild 6	-40.0°	-3.0°	40.224°	-2.55578°	0.224°	0.44422°	0.9948
Bild 8	-170.0°	22.0°	-169.927°	22.562°	0.073°	0.562°	0.9946

Tabelle 4.15: Die fünf Einzelbilder aus dem Panorama vier mit mittlerer Varianz $\sigma^2 = 7$ verzerrt.

Panorama 4	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	-130.0°	10.0°	-129.996°	10.9039°	0.004°	0.9039°	0.9906
Bild 2	-70.0°	12.0°	-70.016°	13.067°	0.016°	1.067°	0.9901
Bild 5	20.0°	13.0°	-20.0202°	-12.0505°	0.0202°	0.9495°	0.9894
Bild 6	-40.0°	-3.0°	40.0494°	-2.10869°	0.0494°	0.89131°	0.9907
Bild 8	-170.0°	22.0°	170.1°	22.879°	0.1°	0.879°	0.9908

Tabelle 4.16: Die fünf Einzelbilder aus dem Panorama vier mit maximaler Varianz $\sigma^2 = 15$ verzerrt.

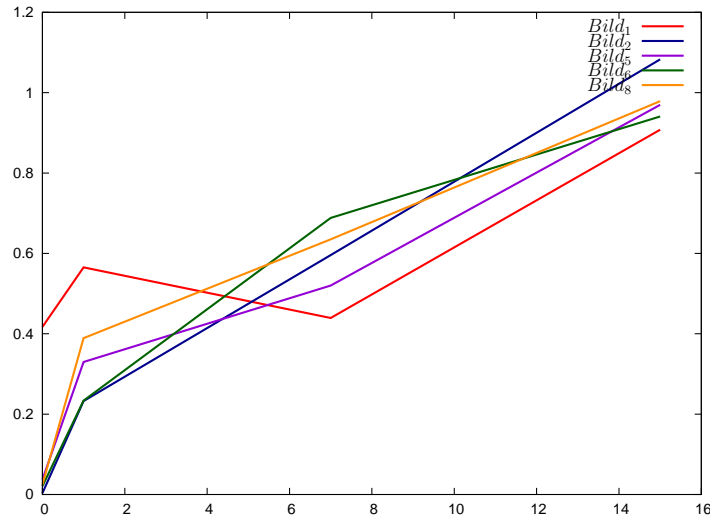


Abbildung 4.6: Zusammenfassung der Ergebnisse des vierten Panoramas mit künstlich verzerrten Bildern.

4.2.3 Bilder realer Aufnahmen

Für die Versuchsreihe mit Bildern von realen Aufnahmen wurden zu einer bekannten Position im 3D Modell 40 Bilder aufgenommen, die eine 360° Sicht um den Betrachter herum zeigen. Diese Position wurde mit einem Laser Entfernungsgesät bestimmt. Dabei wird vom Punkt aus, an dem die Bilder geschossen wurden, die Entfernung zu vier bis fünf Häuserecken gemessen. Die Weltkoordinaten dieser gemessenen Punkte kann man im 3D Modell ablesen. Der Standpunkt, ist dann der Punkt, der von allen Punkten die gemessene Entfernung besitzt. Die Ausrichtung der Kamera wurde letztendlich manuell im 3D Modell bestimmt und die daraus entstehenden Koordinaten einfach abgelesen. Da die Rotationsparameter der Kamera mit Hilfe von Quaternionen angegeben sind, mussten diese in die entsprechenden Euler Winkel umgerechnet werden:

$$x = \arctan\left(\frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \quad (4.1)$$

$$y = \arcsin(2(q_0q_2 - q_3q_1)) \quad (4.2)$$

$$z = \arctan\left(\frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)}\right) \quad (4.3)$$

Die vertikale Blickrichtung α_{ver} ist hierbei durch x und die horizontale Blickrichtung α_{hor} durch z gegeben. Die vier Parameter q_0 bis q_3 beschreiben die Rotationsparameter der Kamera in Form eines Quaternions. Da es in dieser Qualifikationsarbeit das Ziel ist die Orientierung der Kamera zu bestimmen, kann der Parameter y außer Acht gelassen werden. Dieser beschreibt die horizontale Neigung der Kamera. Die Tabellen 4.17 und 4.18 zeigen die Ergebnisse dieser Versuchsreihe.

Panorama 5	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 1	59.9304°	12.7401°	59.4367°	14.416°	0.494°	1.676°	0.988
Bild 2	60.0627°	29.3814°	58.0768°	30.5319°	1.986°	1.151°	0.984
Bild 3	44.722°	12.2707°	44.1548°	12.8198°	0.567°	0.549°	0.992
Bild 4	42.902°	27.3512°	42.7252°	28.022°	0.177°	0.671°	0.990
Bild 5	42.5882°	33.3414°	43.0352°	34.1303°	0.447°	0.789°	0.987
Bild 6	29.5113°	14.6107°	29.0813°	14.2088°	0.432°	0.402°	0.990
Bild 7	29.6415°	23.3613°	29.4695°	23.9121°	0.172°	0.551°	0.990
Bild 8	26.5215°	29.4316°	28.8368°	30.3078°	2.315°	0.876°	0.989
Bild 9	12.8706°	18.6411°	15.1597°	18.8345°	2.289°	0.193°	0.984
Bild 10	13.0006°	32.3717°	107.516°	13.4522°	94.515°	18.919°	0.958
Bild 11	1.4299°	24.4412°	-0.5769°	26.9707°	2.006°	2.529°	0.910
Bild 12	-0.7801°	30.6012°	0.256°	31.418°	1.036°	0.817°	0.889
Bild 13	-18.5907°	28.911°	-17.6409°	30.2935°	0.950°	1.383°	0.925
Bild 14	-31.5912°	27.7812°	-29.1187°	26.0307°	2.473°	1.751°	0.938
Bild 15	-29.592°	13.261°	-33.9379°	11.4685°	4.346°	1.793°	0.891
Bild 16	-29.482°	9.550°	-29.112°	9.934°	0.369°	0.384°	0.952
Bild 17	-44.6732°	9.681°	-40.3762°	8.7500°	4.297°	0.930°	0.909
Bild 18	-45.5031°	3.04065°	-168.425°	-0.797°	122.921°	3.838°	0.931
Bild 19	-38.8732°	29.2522°	-37.2038°	28.6138°	1.669°	0.638°	0.919
Bild 20	-48.4956°	4.6411°	66.5491°	21.3765°	115.004°	16.735°	0.922

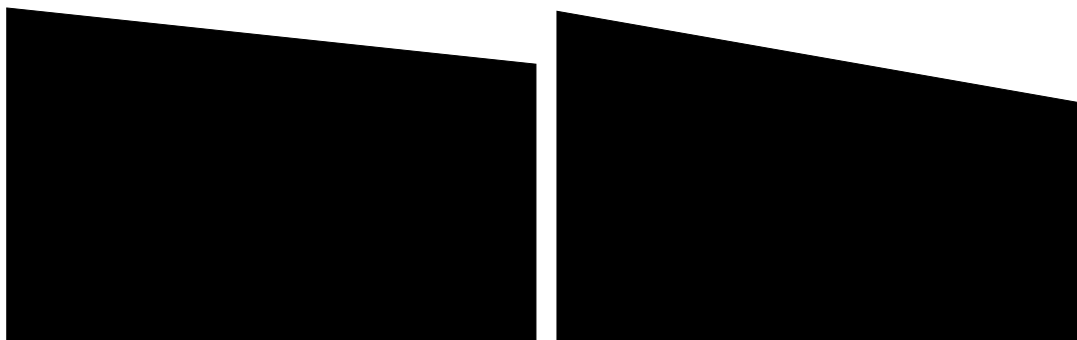
Tabelle 4.17: Die ersten 20 Ergebnisse für die 40 Bilder realer Aufnahmen, die eine 360° Sicht um den Betrachter zeigen.

Panorama 5	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 21	-61.888°	3.771°	84.0637°	19.0104°	145.953°	15.239°	0.927
Bild 22	-76.809°	4.562°	81.5536°	15.7467°	158.363°	11.184°	0.977
Bild 23	-90.824°	4.167°	-249.959°	7.38051°	159.131°	3.217°	0.933
Bild 24	-106.673°	3.596°	-106.359°	3.9972°	0.314°	0.401°	0.996
Bild 25	-123.493°	3.468°	0.0°	0.0°	123.493°	3.468°	0.0
Bild 26	-129.474°	3.598°	0.0°	0.0°	129.475°	3.599°	0.0
Bild 27	-151.577°	4.559°	0.0°	0.0°	151.578°	4.5591°	0.0
Bild 28	-166.569°	4.559°	-293.556°	23.209°	126.988°	18.650°	0.911
Bild 29	-166.569°	4.559°	-294.116°	22.223°	127.547°	17.664°	0.917
Bild 30	-182.390°	3.689°	-182.244°	3.003°	0.146°	0.687°	0.978
Bild 31	-196.821°	0.649°	-197.501°	1.128°	0.679°	0.479°	0.970
Bild 32	-211.513°	2.040°	-212.506°	1.5507°	0.9928°	0.489°	0.964
Bild 33	-224.775°	4.510°	-225.623°	5.04171°	0.848°	0.531°	0.976
Bild 34	-239.038°	10.281°	-240.075°	10.6911°	1.037°	0.410°	0.987
Bild 35	-254.035°	14.653°	-258.592°	14.9902°	4.556°	0.336°	0.986
Bild 36	89.082°	15.776°	89.2462°	16.644°	0.159°	0.867°	0.978
Bild 37	73.754°	16.168°	73.9835°	17.3297°	0.229°	1.161°	0.976
Bild 38	57.106°	17.300°	57.4532°	17.4051°	0.347°	0.105°	0.986
Bild 39	43.974°	23.401°	43.3417°	23.6119°	0.632°	0.211°	0.989
Bild 40	29.942°	19.811°	28.6097°	19.6122°	1.332°	0.200°	0.990

Tabelle 4.18: Die letzten 20 Ergebnisse für die 40 Bilder realer Aufnahmen, die eine 360° Sicht um den Betrachter zeigen.

Auf den ersten Blick wird sofort klar, dass die Szenen, die auf den Bildern Nummer 25, 26 und 27 zu sehen sind von dem Algorithmus nicht gefunden wurden. Die Ursache hierfür liegt darin begründet, dass die Detektion von Eckpunkten auf den realen Aufnahmen fehl schlug. Bei Bild Nummer 25 und 26 liegt das daran, dass die Gebäudestrukturen auf den Aufnahmen keine markanten Punkte zeigen, die nach den Kriterien aus Kapitel 3.3 detektiert werden könnten (siehe Abbildung 4.7). Bei Bild Nummer 27 ist die Approximation der Horizontlinie für die Nicht-Detektion verantwortlich (siehe Abbildung 4.8). Werden die Parameter für die Approximation der Horizontlinie angepasst, kann ein Punkt nach den Kriterien in Kapitel 3.3 detektiert werden und dies führt dann zu folgenden Ergebnissen für Bild Nummer 27:

- $\alpha_{hor}^{res} = -154.765^\circ$
- $\alpha_{ver}^{res} = 4.33183^\circ$
- $\Delta_{hor} = 3.1873$
- $\Delta_{ver} = 0.22728$
- $sim = 0.988$

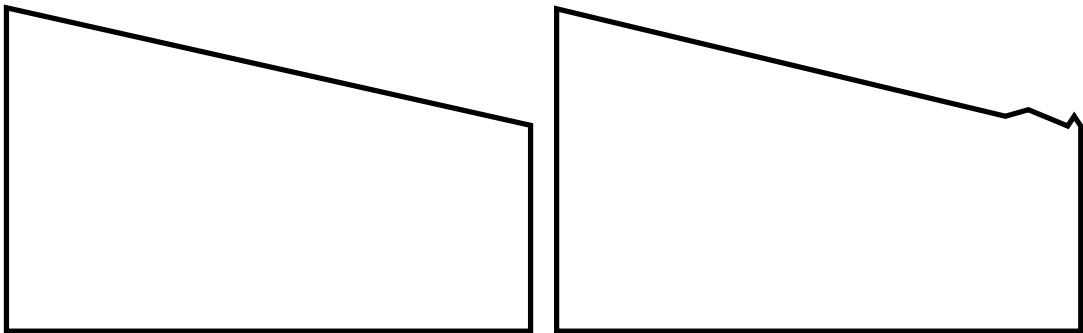


(a) Die Silhouette des Bildes Nummer 25.

(b) Die Silhouette des Bildes Nummer 26.

Abbildung 4.7: Die Silhouetten der Bilder Nummer 25 und 26, in denen keine Eckpunkte extrahiert werden konnten.

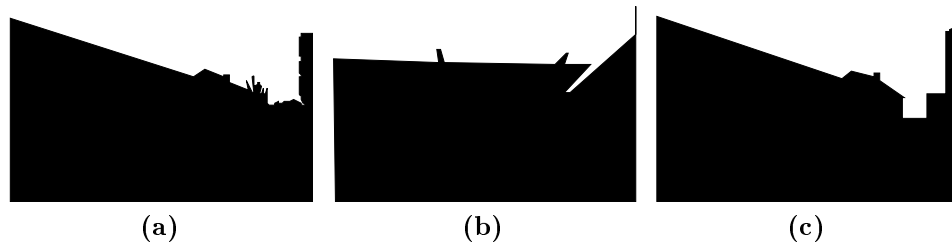
Im Unterschied zu Bild Nummer 27 konnten Punkte in Bild Nummer 28 und 29 zwar detektiert werden, jedoch hat sich das nicht lineare Optimierungsverfahren anhand eines Punktes und der geschätzten Blickrichtung dann in die falsche Richtung angenähert. Dies liegt darin begründet, dass die Ähnlichkeit zwischen den Szenen auf den Bildern trotz eigentlich falscher Blickrichtung sehr hoch wird (siehe Abbildung 4.9). Nach dem Anpassen der Parameter für die Approximation



(a) Horizontlinie ohne Anpassung der Parameter. (b) Horizontlinie nach Anpassung der Parameter.

Abbildung 4.8: Der Vergleich zwischen zwei Approximationen der Horizontlinie des gleichen Bildes Nummer 27.

der Horizontlinie können die Ergebnisse aus Tabelle 4.19 festgehalten werden. Es handelt sich bei den beiden Bildern um jeweils die gleichen Szenen, nur mit unterschiedlichen Neigungswinkel der Kamera in y -Richtung.



(a)

(b)

(c)

Abbildung 4.9: Abbildung (a) zeigt die gesuchte Szene, Abbildung (b) die vom Algorithmus fälschlicherweise ausgegebene Szene und Abbildung (c) die Szene nach Anpassen der Parameter.

Panorama 5	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 28	-166.5693°	4.5594°	-166.444°	3.96536°	0.1253°	0.59404°	0.982
Bild 29	-166.5693°	4.5594°	-166.59°	3.59809°	1.116°	0.0209°	0.961

Tabelle 4.19: Die zusätzlichen Ergebnissen für die Bilder Nummer 28 und 29.

Panorama 5	α_{hor}^{ges}	α_{ver}^{ges}	α_{hor}^{res}	α_{ver}^{res}	Δ_{hor}	Δ_{ver}	sim
Bild 10	13.0006°	32.3717°	15.6116°	32.9762°	2.616°	0.6045°	0.953
Bild 18	-45.5031°	3.04065°	-45.5881°	3.3449°	0.085°	0.30425°	0.908
Bild 20	-48.4956°	4.6411°	-48.8785°	4.97568°	0.3829°	0.3345°	0.905
Bild 21	-61.8889°	3.77178°	-60.3138°	3.28657°	1.5751°	0.48521°	0.919
Bild 22	-76.8095°	4.56264°	-76.9262°	3.6951°	0.1167°	0.86754°	0.925
Bild 23	-90.8249°	4.1677°	-93.9852°	3.23432°	3.1603°	0.93338°	0.933

Tabelle 4.20: Die zusätzlichen Ergebnissen für die Bildserie Nummer zehn, 18, sowie 20 bis 23.

Große Probleme hatte der Algorithmus auch bei den Bildserien Nummer zehn, 18, sowie 20 bis 23. Hier wurden ausschließlich falsche Blickrichtungen ausgegeben, mit denen Einzelbilder aus dem 360° Panorama generiert wurden, die zwar eine hohe Ähnlichkeit zu den gesuchten Szenen auf den Bildern der realen Aufnahmen hatten, aber eine eigentlich unterschiedliche Szene zeigten. Interessanterweise lassen sich aus den Evaluationsdaten zusätzlich die Ergebnisse aus Tabelle 4.20 zeigen. Hier wird deutlich, dass die gesuchten Szenen auf den Bildern der realen Aufnahmen mit einer geringeren Abweichung in horizontaler und vertikaler Blickrichtung gefunden wurden. Nur mit dem Unterschied, dass die Ähnlichkeit in allen Fällen für diese Bildserie unter den Werten aus Tabelle 4.17 und 4.18 liegt, so dass nach einem ähnlicheren Einzelbild weiter gesucht wurde und der Algorithmus an der entsprechenden Stelle nicht terminierte. Die Abbildungen 4.10 bis 4.12 sollen dabei helfen diesen Zusammenhang zu verstehen.

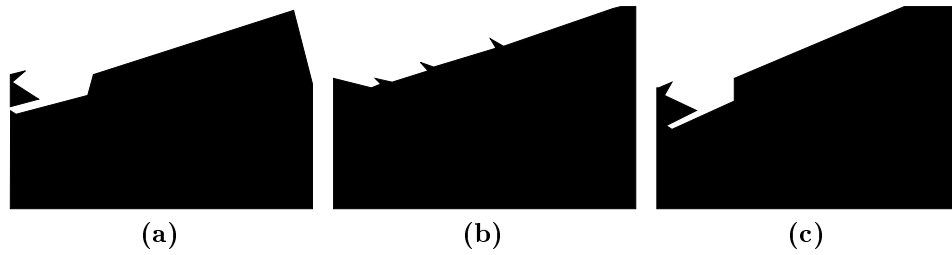


Abbildung 4.10: Abbildung (a) zeigt die gesuchte Szene, Abbildung (b) die vom Algorithmus fälschlicherweise ausgegebene Szene und Abbildung (c) die Szene, die in den Evaluationsdaten zu finden war, aber nicht ausgegeben wurde.

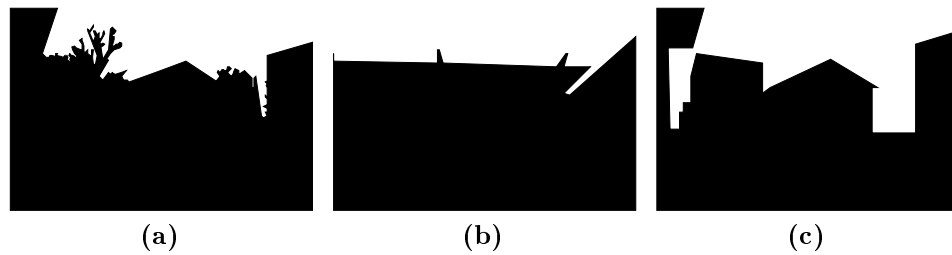


Abbildung 4.11: Abbildung (a) zeigt die gesuchte Szene, Abbildung (b) die vom Algorithmus fälschlicherweise ausgegebene Szene und Abbildung (c) die Szene, die in den Evaluationsdaten zu finden war, aber nicht ausgegeben wurde.

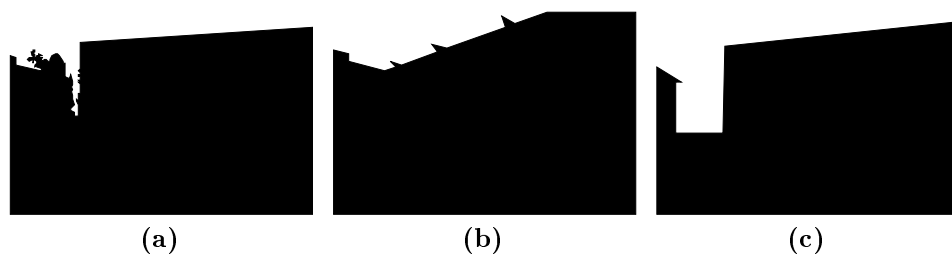


Abbildung 4.12: Abbildung (a) zeigt die gesuchte Szene, Abbildung (b) die vom Algorithmus fälschlicherweise ausgegebene Szene und Abbildung (c) die Szene, die in den Evaluationsdaten zu finden war, aber nicht ausgegeben wurde.

Bis auf einzelnen Ausreißer, deren horizontale oder vertikale Blickrichtung bis zu 4° abweichen kann, wurde die Blickrichtung in den restlichen Bildern dieser Versuchsreihe gut angenähert. Die Abbildung 4.13 zeigt hierbei eines der besseren Ergebnisse mit Bild Nummer 38, während Abbildung 4.14 mit Bild Nummer 15 einen Ausreißer zeigt. Die Abbildungen 4.15 und 4.16 fassen die Ergebnisse dieser Versuchsreihe noch einmal zusammen. Dabei sind die horizontalen und vertikalen Abweichungen jeweils auf der x -Achse eingetragen, während die y -Achse den prozentualen Anteil der entsprechenden Abweichung angibt.

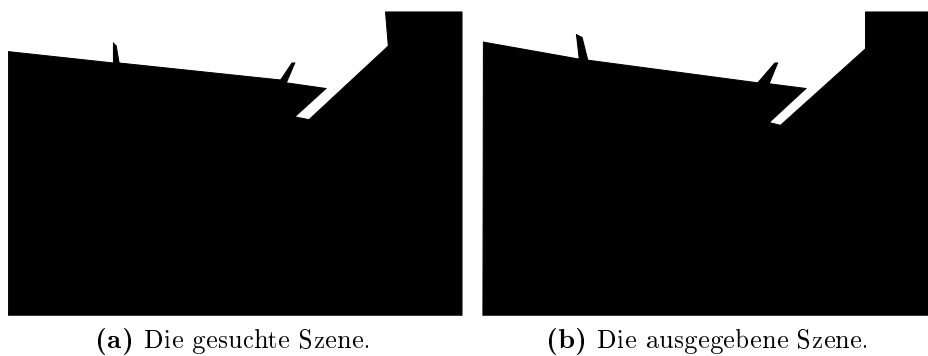


Abbildung 4.13: Eines der besseren Ergebnisse mit Bild Nummer 38 für diese Versuchsreihe.

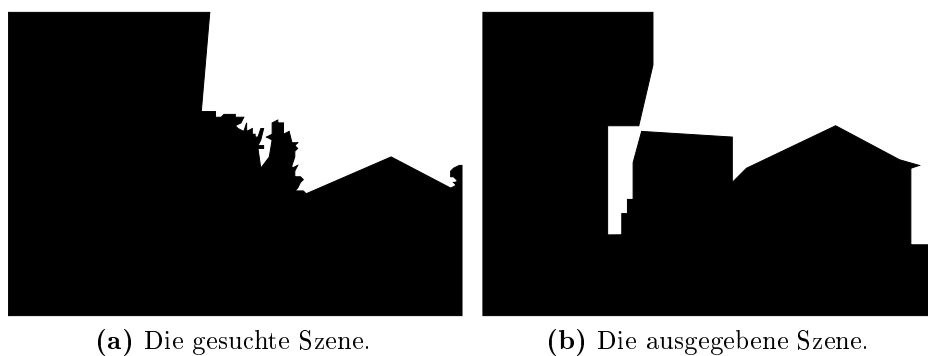


Abbildung 4.14: Einer der Ausreißer mit Bild Nummer 15 für diese Versuchsreihe.

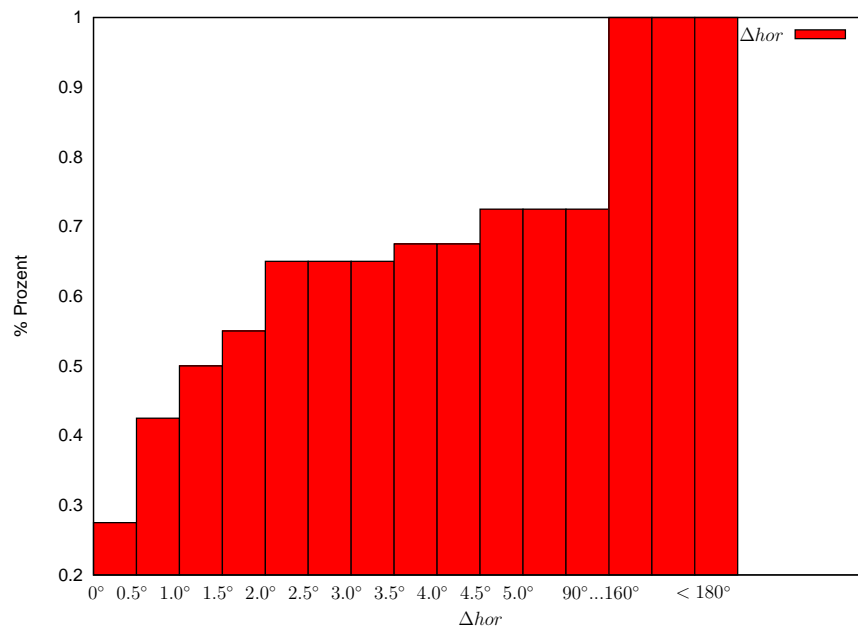


Abbildung 4.15: Zusammenfassung der Ergebnisse für die Abweichungen in horizontaler Blickrichtung.

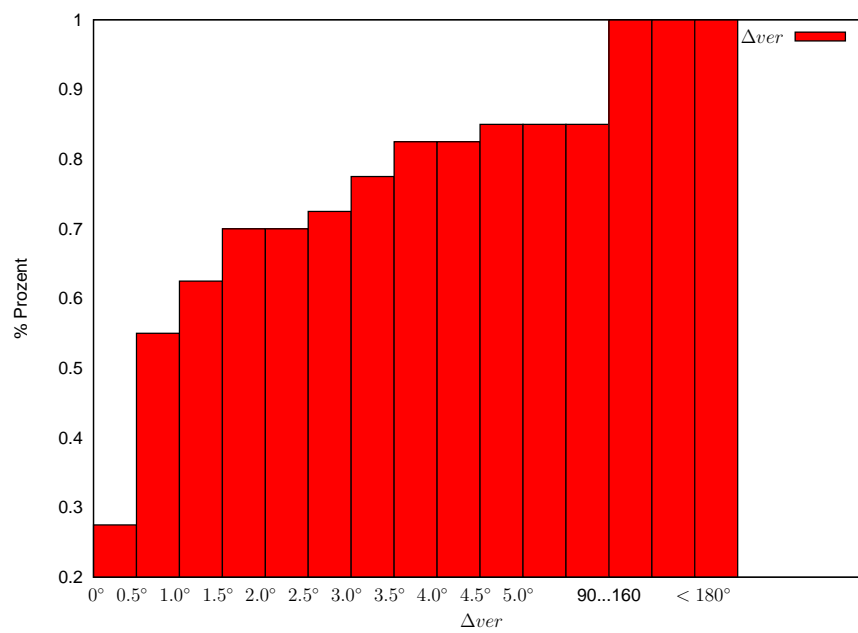


Abbildung 4.16: Zusammenfassung der Ergebnisse für die Abweichungen in vertikaler Blickrichtung.

4.3 Laufzeit des Verfahrens

Die Laufzeit des Verfahrens wurde ausschließlich auf Bildern gemessen, bei denen die Detektion von Eckpunkten ohne größere Probleme möglich war. In die erste Messung fließen sowohl die Ergebnisse von Bildern, bei denen der Algorithmus mit der richtigen Lösung terminierte, als auch die Ergebnisse von Bildern, bei denen eine Anpassung der Approximationsparameter notwendig war, ein.

Im besten Fall terminierte der Algorithmus mit einer Lösung nach zehn Sekunden und 32 Bildern, die während der Iteration des nicht linearen Optimierungsverfahrens generiert wurden. Im Durchschnitt braucht das Verfahren ungefähr drei Minuten und 404 Bilder, um mit der richtigen Lösung zu terminieren. Im schlechtesten Fall hingegen kann es bis zu neun Minuten und vier Sekunden dauern, bis der Algorithmus nach dem Vergleich von insgesamt 1480 Bildern terminiert.

Bei Bildern für die das Verfahren mit einer falschen Lösung terminiert, verhält es sich ähnlich. Im besten Fall wird die richtige Lösung nach 48 Sekunden und 65 Bildvergleichen erreicht. Im Durchschnitt braucht der Algorithmus bei diesen Bildern auch ungefähr drei Minuten und 480 Bilder. Während im schlechtesten Fall die Dauer bis zum Abbruch des nicht linearen Optimierungsverfahrens acht Minuten und 17 Sekunden betragen kann. Es wurden dabei 1480 Bilder generiert und miteinander verglichen. Die Abbildungen 4.17 und 4.18 fassen die Ergebnisse zusammen. Auf der y -Achse ist die Zeit in Sekunden eingetragen und auf der x -Achse die einzelnen Fälle.

Wie bereits zu erwarten ist die Laufzeit in erster Linie von der Anzahl der Bilder, die mit dem Bild der gesuchten Szene verglichen werden müssen, abhängig. Die Anzahl der Vergleiche hängt wiederum von der Näherungsweise des nicht linearen Optimierungsverfahrens ab. Desweiteren hängt die Laufzeit von der Genauigkeit der Approximation der Horizontlinie eines Silhouettenbildes ab. Diese Genauigkeit legt fest, wie viele Punkte auf der Horizontlinie des Panoramas und des Bildes einer realen Aufnahme detektiert werden können. Da das nicht lineare Optimierungsverfahren für alle detektierten Punkte der Liste L_A einer Aufnahme, mit Hilfe aller Punkte der Liste L_P , Blickrichtungen schätzt und anpasst, ist natürlich eine geringe Anzahl von Punkten für die Laufzeit von Vorteil. Eine geringe Anzahl kann sich aber auch negativ auf die Lösung auswirken. Falls die geschätzte Blickrichtung zu weit von der gesuchten Blickrichtung entfernt ist, kann es Schwierigkeiten bei der Annäherung geben. Falls sonst keine Punkte mehr in Betracht gezogen werden könnten.

Desweiteren hängt die Laufzeit von der Generierung und vom Vergleich der Bilder ab. Je höher die Auflösung, desto länger dauert die Generierung und der Vergleich. Es ließ sich darüberhinaus auch erkennen, dass die Approximation der Horizontlinie bei unterschiedlicher Auflösung unterschiedlich gute Ergebnisse lieferte. Deswegen wurden Bilder mit einer Auflösung von 800×535 Pixeln benutzt.

Hier war der Vergleich und die Generierung von Bildern vergleichsweise schnell und das Ergebnis der Approximation der Horizontlinie war für die Detektion von Eckpunkten ausreichend.

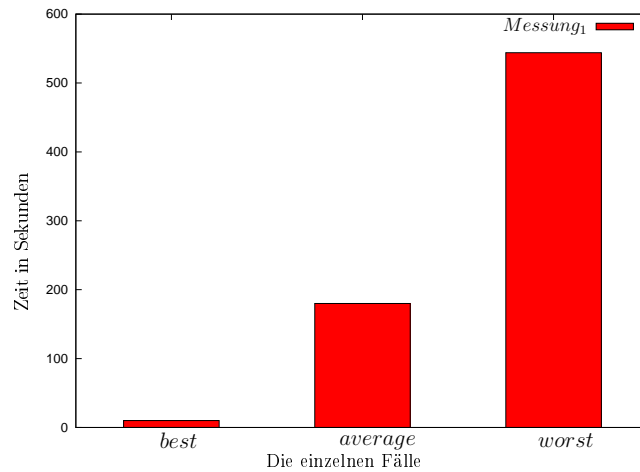


Abbildung 4.17: Zusammenfassung der Ergebnisse für die Laufzeit bei Bildern, für die das nicht lineare Optimierungsverfahren mit einer richtigen Lösung terminierte.

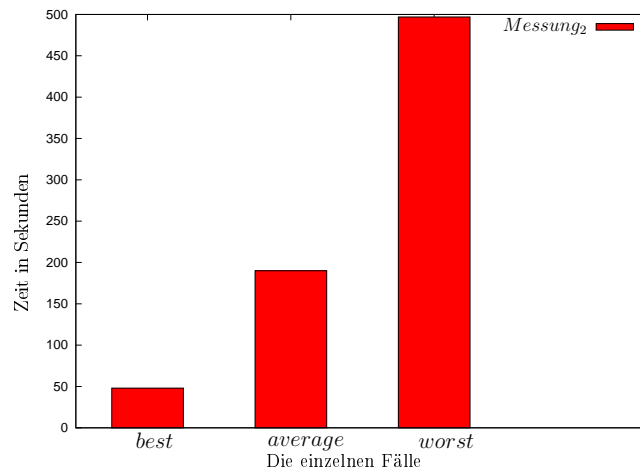


Abbildung 4.18: Zusammenfassung der Ergebnisse für die Laufzeit bei Bildern, für die das nicht lineare Optimierungsverfahren mit einer falschen Lösung terminierte.

4.4 Grenzen des Verfahrens

Aus der Versuchsreihe mit Bildern realer Aufnahmen sind deutliche Grenzen des Verfahrens zu erkennen. Es kommt in insgesamt sechs Fällen zu einer Termination des Verfahrens mit einer falschen Lösung, obwohl die richtige Lösung während der Iterationen des nicht linearen Optimierungsverfahrens vorliegt. Dies liegt an der Art und Weise wie Bilder realer Aufnahmen mit den künstlich generierten Einzelbildern eines 360° Panoramas verglichen werden. Das Ähnlichkeitsmaß hat Probleme zwei Bilder zu vergleichen, die sich zu einem großen Teil in Farbe und Stelle der Pixel überschneiden. Dies kann sowohl bei schwarzen, als auch bei weißen Pixeln vorkommen.

Angenommen die Horizontlinie in zwei Bildern ist in ungefähr gleicher Höhe abgebildet und die Bilder überschneiden sich stark im unteren und oberen Abschnitt. Dann ist es mit der XOR-Operation kaum möglich nur anhand des Abschnittes, der die Horizontlinie zeigt, die beiden Bilder zu unterscheiden. Zusätzlich zu dieser Schwierigkeit ist in realen Aufnahmen oft Vegetation abgebildet. Diese ist im 3D Modell nur durch ein Rechteck approximiert oder gar nicht abgebildet. Die Approximation durch ein Rechteck entspricht nicht der Silhouette einer Pflanze oder eines Baumes. Befindet sich also Vegetation auf dem Foto, dann ist das Einzelbild nicht ähnlich genug. Dies verursacht die unterschiedliche Silhouette an der Position der Vegetation. Dadurch wird versucht ein passenderes Einzelbild in dem 360° Panorama zu finden, dessen Horizontlinie sich ungefähr in gleicher Höhe befindet (siehe Abbildung 4.11).

Vegetation scheint eine der Hauptursachen dafür zu sein, dass das Verfahren mit einer falschen Lösung terminiert, obwohl die richtige Lösung in den Evaluationsdaten, bereits vorliegt. In Abbildung 4.12 beispielsweise fehlt die Vegetation im Einzelbild vollständig. Bis auf einen einzigen Fall bei dieser Problem Bildserie ist Vegetation den übrigen Bildern der realen Aufnahmen abgebildet. Die Folge davon ist eine falsche Lösung mit einem Einzelbild, welches an der Position der Vegetation komplett schwarz ist. Dadurch entsteht eine höhere Ähnlichkeit zwischen der falschen Lösung und der Szene in der realen Aufnahme, als zwischen der richtigen Lösung und der gesuchten Szene. Bei dem Ausnahmefall handelt es sich um zwei Bilder, die eine sehr ähnliche Horizontlinie im Verlauf und Position haben.

Kapitel 5

Zusammenfassung und Ausblick

Im folgenden Kapitel werden die wichtigsten Ergebnisse und Schlußfolgerungen dieser Arbeit zusammengefasst. Im Anschluß findet sich ein Ausblick auf mögliche Erweiterungen des vorgestellten Verfahrens.

5.1 Zusammenfassung

In dieser Qualifikationsarbeit wurde ein merkmalsbasierter Algorithmus entwickelt, der zur einer groben Position im 3D Modell des Campus der Universität Koblenz versucht die Orientierung der Kamera zum Zeitpunkt einer Aufnahme zu bestimmen. Das ganze Verfahren basiert auf der Idee der sogenannten „Analyse durch Synthese“. Hierbei werden Szenen auf Bildern realer Aufnahmen mit den Szenen künstlicher Einzelbilder aus einem 3D Modell verglichen. Ähneln sich die abgebildeten Szenen, kann man mit Hilfe der Koordinaten im entsprechenden 3D Modell auf die Position schließen, an der die Szene aufgenommen wurde. Der Vergleich zwischen zwei Bildern erfolgt in dieser Arbeit anhand der Silhouette eines Bildes.

Um die Orientierung der Kamera zum Zeitpunkt der Aufnahme zu bestimmen wird das nicht lineare Optimierungsverfahren Levenberg-Marquardt verwendet. Dieses bekommt als Eingabe eine geschätzte Blickrichtung, die im Laufe mehrerer Iterationen möglichst genau an die gesuchte Blickrichtung der Kamera angenähert werden soll. Hierfür muss die Orientierung bei den Testbildern bekannt sein. Da während einer Iteration mehrere Bilder miteinander verglichen werden sollen, ist es wegen des Zeitaufwands nicht praktikabel ständig ein Einzelbild aus dem 3D Modell zu rendern. Zu diesem Zweck wird mit der groben Position zum Zeitpunkt der Aufnahme ein 360° Panorama gerendert, aus dem man einzelne Bilder generieren kann.

Die Schätzung der Blickrichtung ergibt sich aus der Korrespondenz zwischen einem detektierten Eckpunkt in der realen Aufnahme und einem Kandidatenpunkt

auf dem 360° Panorama. Zu diesem Zweck werden Hypothesen anhand der detektierten Eckpunkte in der realen Aufnahme aufgestellt, die zu Kandidatenpunkten im 360° Panorama führen. Damit das nicht lineare Optimierungsverfahren abschätzen kann in welche Richtung es sich der gesuchten Blickrichtung annähern soll, wird die Ähnlichkeit zwischen dem Bild einer realen Aufnahme und dem durch die geschätzte Blickrichtung generierten Einzelbild aus dem 360° Panorama, durch einen Wert ausgedrückt. Dieser Wert ergibt sich durch die Anwendung der XOR-Operation auf die beiden Silhouettenbilder.

Die Versuchsreihen aus Kapitel vier haben gezeigt, dass das Verfahren bei Bildern, die aus dem Panorama generiert wurden, keinerlei Probleme hat. Auch wenn diese Bilder mit einem Pseudozufallsgenerator verzerrt wurden, ist es für den Algorithmus stets möglich die gesuchte Szene im Panorama zu finden. Dies liegt daran, dass sich im Prinzip der Verlauf der Horizontlinie nicht verändert hat. Zwar ist die Silhouette verzerrt worden, aber die Approximation der Horizontlinie ähnelt stark der einer unverzerrten Silhouette. Die Eckpunkte können ohne weiteres detektiert werden, so dass entsprechende Hypothesen aufgestellt werden können, die zu einer initialen Blickrichtung für das nicht lineare Optimierungsverfahren führen. Es kann gegebenenfalls zu einer falschen Kategorisierung für einen Eckpunkt kommen, da die Horizontlinie nicht genau genug approximiert wurde. Dieses Problem kann aber durch Anpassen der Parameter für die Approximation der Horizontlinie umgangen werden. Dies bedeutet, dass die Horizontlinie in diesem Fall genauer approximiert wurde.

Größere Probleme zeigte der Algorithmus bei Bildern realer Aufnahmen, auf denen Vegetation zu sehen war. Wie man in den Ergebnissen des Kapitels 4.2.3 sehen kann terminiert das Verfahren in sechs Fällen mit einer falschen Lösung. Hier brachte auch die Anpassung der Parameter für die Approximation der Horizontlinie keine Verbesserung des Problems. Bei zwei Fällen konnte der Algorithmus durch eine genauere Approximation der Horizontlinie im Bild der realen Aufnahme mit der richtigen Lösung terminieren. Hierzu musste zusätzlich die Horizontlinie des Panoramabildes genauer approximiert werden. Dies kam insgesamt nur diese zwei Mal während der gesamten Versuchsreihe in Kapitel vier vor. Ansonsten genügte eine genauere Approximation der Horizontlinie im Bild einer realen Aufnahme. Hierbei lag der Wertebereich für die Parameter bei zehn, 15 und 20.

In drei Fällen konnten keine Eckpunkte detektiert werden. Dies lag zwei Mal an der Tatsache, dass die Aufnahmen Gebäudestrukturen zeigten, die keine Eckpunkte aufwiesen. Im Übrigen konnte die Anpassung der Parameter für die Approximation der Horizontlinie bei der Detektion von Eckpunkten Abhilfe schaffen.

5.2 Ausblick

Grundsätzlich ist die Ursache für die Terminierung mit einer falschen Lösung, obwohl die richtige Lösung während der Iterationen des nicht linearen Optimierungsverfahrens vorliegt, der Vergleich anhand der XOR-Operation. Es kommt bei diesen Problemfällen vor, dass Bilder fälschlicherweise als Lösung ausgegeben werden, nur weil diese eine Horizontlinie zeigen, die in der Position zur gesuchten Szene ähnlich ist. Zusätzlich dazu wird der Vergleich durch das Auftreten von Vegetation bei eigentlich gleichen Szenen erschwert. Es sollte bei diesen Fällen ein Ähnlichkeitsmaß benutzt werden, welches nicht nur das ganze Bild in den Vergleich miteinbezieht, sondern insbesondere auf die Eigenschaften der Horizontlinie eingeht. Mit solch einem Ähnlichkeitsmaß sollte dieses Problem gelöst werden können.

Das Problem der Vegetation kann auf zwei Weisen gelöst werden. Das 3D Modell sollte Vegetation, die zwar in Bildern realer Aufnahmen auftaucht, aber im 3D Modell nicht modelliert ist, modellieren. Desweiteren reicht die Apporximation von Vegetation im 3D Modell, für einen Vergleich mit Bildern realer Aufnahmen, durch ein einfaches Rechteck nicht aus. Auf der anderen Seite könnte der Bereich, in dem Vegetation zu finden ist, markiert werden. Diese Markierung kann man auf das Einzelbild des 3D Modells übertragen und somit den Bereich beim Vergleich komplett ausschliessen.

Anhang A

Mathematische Bezeichner und Symbole

Im Folgenden werden die verwendeten Variablen und mathematischen Bezeichner aufgelistet und kurz erläutert.

Symbol/Bezeichner	Bedeutung
\vec{v}	Der Vektor, der die Blickrichtung der Kamera beschreibt.
x_{view}	x -Koordinate der Blickrichtung der Kamera.
y_{view}	y -Koordinate der Blickrichtung der Kamera.
z_{view}	z -Koordinate der Blickrichtung der Kamera.
α_{hor}	Horizontaler Winkel der Blickrichtung.
α_{ver}	Vertikaler Winkel der Blickrichtung.
θ_{hor}	Horizontaler Öffnungswinkel der Kamera.
θ_{ver}	Vertikaler Öffnungswinkel der Kamera.
β	Winkel für die Berechnung der x -Koordinate des Schnittpunktes S_{Z_a} .

Tabelle A.1: Mathematische Bezeichner

Symbol/Bezeichner	Bedeutung
Z_e	Der Einheitszylinder.
h	Höhe eines Zylinders.
r_e	Radius des Einheitszylinders.
Z_z	Das zugeklappte Zylinderpanorama.
r_z	Radius des zugeklappten Zylinderpanoramas Z_z .
Z_a	Das an der negativen y -Achse aufgeklappte Zylinderpanorama.
$width_{Z_a}$	Horizontale Auflösung des aufgeklappten Zylinderpanoramas Z_a .
M_{Z_a}	Mittelpunkt des aufgeklappten Zylinderpanoramas Z_a .
\vec{s}	Vektor, der den Strahl, welcher vom Mittelpunkt M_{Z_z} des Zylinderpanoramas Z_z aus gesendet wird, beschreibt.
M_{Z_z}	Mittelpunkt des Zylinderpanoramas Z_z im Ursprung (0 0 0) des kartesischen Koordinatensystems.
S_{Z_z}	Schnittpunkt von Strahl \vec{s} mit dem Zylinderpanorama Z_z .
S_{Z_a}	Schnittpunkt von Strahl \vec{s} mit dem aufgeklappten Zylinderpanorama Z_a .
$x_{S_{Z_a}}$	x -Koordinate des Schnittpunktes S_{Z_a} .
E_i	Die Bildebene, welche durch θ_{hor} und θ_{ver} aufgespannt wird.
$h_{Auflösung}$	Horizontale Auflösung der Bildebene E_i .
$v_{Auflösung}$	Vertikale Auflösung der Bildebene E_i .
M_{E_i}	Mittelpunkt der Bildebene E_i .
P_{x_i}	x -Koordinate des Punktes P_{E_i} .
P_{y_i}	y -Koordinate des Punktes P_{E_i} .
P_{E_i}	Punkt auf der Bildebene E_i .
A	Reale Aufnahme.
M_{B_A}	Mittelpunkt des Bildes der realen Aufnahme A .
\vec{i}_x	Horizontaler Vektor, der senkrecht zur Blickrichtung \vec{v} in der Bildebene E_i steht.
\vec{i}_y	Vertikaler Vektor, der senkrecht zur Blickrichtung \vec{v} in der Bildebene E_i steht.
k	Kreisgleichung mit dem Mittelpunkt M_{Z_z} des Zylinderpanoramas Z_z im Ursprung (0 0 0).
λ	Faktor für die Berechnung des Schnittpunktes S_{Z_z} durch Gleichsetzen der Kreisgleichung k mit der Strahlgleichung des Strahls \vec{s} .
γ	Winkel zwischen drei aufeinander folgenden Punkten, die eine Ecke beschreiben.
$width_A$	Horizontale Auflösung der Aufnahme A in Pixel.
$height_A$	Höhe der Aufnahme A in Pixel.
$width_{E_i}$	Breite der Bildebene E_i in kartesischen Koordinaten.
f_{ver}	Vertikaler Faktor für die Umrechnung von Pixeln in Längeneinheiten des kartesischen Koordinatensystems.
f_{hor}	Horizontaler Faktor für die Umrechnung von Pixeln in Längeneinheiten des kartesischen Koordinatensystems.

Tabelle A.2: Fortsetzung 1 der Mathematischen Bezeichner.

Symbol/Bezeichner	Bedeutung
L_A	Liste mit detektierten und kategorisierten Eckpunkten aus der Aufnahme A .
L_P	Liste mit Kandidatenpunkten zu den entsprechenden Punkten aus der Liste L_A .
P_{L_A}	Punkt aus der Liste L_A .
P_{L_P}	Kandidatenpunkt aus der Liste L_P .
$M_{x_{Z_a}}$	x -Koordinate des Mittelpunktes M_{Z_a} .
$M_{y_{Z_a}}$	y -Koordinate des Mittelpunktes M_{Z_a} .
$M_{x_{B_A}}$	x -Koordinate des Mittelpunktes M_{B_a} .
$M_{y_{B_A}}$	y -Koordinate des Mittelpunktes M_{B_a} .
P_{L_P}	Kandidatenpunkt aus der Liste L_P .
$P_{x_{L_A}}$	x -Koordinate des Punktes P_{L_A} aus der Liste L_A .
$P_{y_{L_A}}$	y -Koordinate des Punktes P_{L_A} aus der Liste L_A .
φ_{hor}	Winkel, der für die Berechnung von α_{hor} benötigt wird.
φ_{ver}	Winkel, der für die Berechnung von α_{ver} benötigt wird.
κ_{hor}	Winkel, der für die Berechnung von α_{hor} benötigt wird.
κ_{ver}	Winkel, der für die Berechnung von α_{ver} benötigt wird.
B	Das Bild B .
$B_{(j,i)}$	Die Position eines Pixels an der Stelle (j, i) in Bild B .
Δ_{hor}	Differenz der horizontalen Blickrichtung.
Δ_{ver}	Differenz der vertikalen Blickrichtung.
sim	Wert für die Ähnlichkeit zwischen zwei Bildern.
x^+	Globales Minimum der Funktion $F(x)$.
\mathbf{J}	Jacobi-Matrix, die die partiellen Ableitungen von $F(x)$ enthält.
$F'(x)$	Gradient der Funktion $F(x)$.
h	Schrittweite des nicht linearen Optimierungsverfahrens.
$L(h)$	Linear approximiertes Modell für das Verhalten einer Funktion $F(x)$.
ϱ	Gain Ratio, die die Qualität eines Modells $L(h)$ bestimmt.
μ	Dämpfungsfaktor, der die Richtung und Größe der Schrittweite h beeinflusst.

Tabelle A.3: Fortsetzung 2 der Mathematischen Bezeichner.

Anhang B

Notwendige Bibliotheken

Zum erfolgreichen Kompilieren des Qt-Projektes sind die Bibliotheken KIPL, levmar-2.5, boost_1_42.0 und das lineare Algebra Packet LAPACK notwendig. Bei KIPL handelt es sich um eine Bildverarbeitungsbibliothek der Universität Koblenz, die von der Arbeitsgruppe Priesse entwickelt wurde. Zu finden ist diese Bibliothek auf der beiliegenden CD unter dem Verzeichnis:

`Programmcode/OrientierungAusSilhouetten/libs`

Die Bibliothek levmar-2.5 ermöglicht es, das nicht lineare Optimierungsverfahrens Levenberg-Marquardt in C/C++ zu nutzen. Die Bibliothek levmar-2.5 liegt unter dem Verzeichnis:

`Programmcode/Bibliotheken/levmar-2.5.tgz`

Zusätzlich zu levmar-2.5 wird das lineare Algebra Packet LAPACK benötigt. Dieses läßt sich auf einem Linux-basierten Betriebssystem am einfachsten mit einem Paketmanager installieren. Ansonsten ist dieses Packet im Internet¹ zu finden. Der Pseudozufallsgenerator ist mit Hilfe der Bibliothek boost_1_42.0 implementiert, diese ist unter:

`Programmcode/Bibliotheken/boost_1_42_0.tar.gz`

zu finden. Nach dem Installieren der Bibliotheken muß die Datei `CMakeLists.txt` aktualisiert werden, indem alle Abhängigkeiten eingetragen werden. Das Qt-Projekt läßt sich mit der QMake-Version 2.0.1a und den Befehlen `qmake` und `make` kompilieren. Zu diesem Zweck werden die Befehle im entsprechenden Verzeichnis aus der Konsole heraus nacheinander ausgeführt. Voraussetzung ist eine Qt-Version ab 4.6.2.

¹<http://www.netlib.org/lapack/>

Anhang C

Aufbau der CD

Der Aufbau der beiliegenden CD setzt sich aus den folgenden Verzeichnissen zusammen:

- /Ausarbeitung
 - /OberSeminar
 - /OrientierungAusSilhouetten
- /Bilder
 - /extrahierteSilhouetten
 - /gerenderteSilhouetten
 - /realeAufnahmen
- /EvaluierungsDaten
 - /Laufzeit
 - /Panorama_1
 - /Panorama_2
 - /Panorama_3
 - /Panorama_4
 - /realeBilder
- /Literatur
 - /3D-Shape-Matching
 - /Horizon-Matching
 - /NichtLineareOptimierungsVerfahren
 - /Shape-Matching

- /Programmcode
 - /Bibliotheken
 - /OrientierungAusSilhouetten

In dem Ordner */Ausarbeitung* findet sich die vorliegende Arbeit, sowie der Vortrag zum Oberseminar vom 22.04.2010. Der Ordner */Bilder* und dessen Unterordner beinhalten alle Bilder, die während der Evaluation benutzt wurden. Die Evaluierungsdaten finden sich in dem entsprechenden Ordner */EvaluierungsDaten* unterteilt in die verschiedenen Panoramen, die ausgewertet wurden. Die Literatur und alle zitierten Dokumente sind im Ordner */Literatur* zu finden. Der Ordner */Programmcode* enthält den Programmcode und die notwendigen Bibliotheken.

Literaturverzeichnis

- [BBGL96] BEVERIDGE, Ross ; BEVERIDGE, J. R. ; GRAVES, Christopher ; LESHER, Christopher E.: Local Search as a Tool for Horizon Line Matching. In: *Image Understanding Workshop*, ARPA, Morgan Kaufmann, 1996, S. 683–686
- [BCGJ95] BASRI, Ronen ; COSTA, Luiz ; GEIGER, Davi ; JACOBS, David: Determining the Similarity of Deformable Shapes. In: *Vision Research* 38 (1995), S. 135–143
- [Beh99] BEHRINGER, Reinhold: Improving the Registration Precision by Visual Horizon Silhouette Matching. In: *Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes*. Bellevue, Washington, United States : A. K. Peters, Ltd., 1999, S. 225–232
- [BHR86] BURNS, J. B. ; HANSON, Allen R. ; RISEMAN, Edward M.: Extracting straight lines. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 8 (1986), Nr. 4, S. 425–455
- [BMP02] BELONGIE, Serge ; MALIK, Jitendra ; PUZICHA, Jan: Shape Matching and Object Recognition Using Shape Contexts. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (2002), Nr. 4, S. 509–522
- [Bou05] BOUAGAR, Saliha: A New Approach of Outline Shape Matching. In: ASHRAF, Aboshosha (Hrsg.): *Proc. of the first ICGST International Conference on Graphics, Vision and Image Processing GVIP 05* Bd. 05. Cairo, Egypt : ICGST, 2005, S. 347–351
- [CTSO03] CHEN, Ding-Yun ; TIAN, Xiao-Pei ; SHEN, Yu-Te ; OUHYOUNG, Ming: On Visual Similarity Based 3D Model Retrieval. In: *Comput. Graph. Forum* 22 (2003), Nr. 3, S. 223–232

- [DRB09] DONOSER, M. ; RIEMENSCHNEIDER, H. ; BISCHOF, H.: Efficient Partial Shape Matching of Outer Contours. In: *Proceedings of Asian Conference on Computer Vision (ACCV)*. Xi'an, China, 2009
- [LBTF03] LARABI, Slimane ; BOUAGAR, Saliha ; TRESPADERNE, Felix M. ; FUENTE LOPEZ, Eusebio de l.: LWDOS: language for writing descriptors of outline shapes. In: *SCIA'03: Proceedings of the 13th Scandinavian conference on Image analysis*. Berlin, Heidelberg : Springer-Verlag, 2003, S. 1014–1021
- [LHS01] LENSCH, Hendrik P. A. ; HEIDRICH, Wolfgang ; SEIDEL, Hans-Peter: A Silhouette-Based Algorithm for Texture Registration and Stitching. In: *Graphical Models* 63 (2001), S. 245–262
- [LJ05] LING, Haibin ; JACOBS, David W.: Using the Inner-Distance for Classification of Articulated Shapes. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* Bd. 2. Washington, DC, USA : IEEE Computer Society, 2005, S. 719–726
- [LMP⁺03] LEE, J. ; MOGHADDAM, B. ; PFISTER, H. ; MACHIRAJU, R. ; LEE, Jinho ; MOGHADDAM, Baback ; PFISTER, Hanspeter ; MACHIRAJU, Raghu: Silhouette-Based 3D Face Shape Recovery. In: *In Graphics Interface*, 2003, S. 21–30
- [Lou04] LOURAKIS, M.I.A.: *levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++*. 2004. – [Accessed on 31 Jan. 2005.]
- [MG00] MOESLUND, Thomas B. ; GRANUM, Erik: 3D Human Pose Estimation using 2D-Data and an Alternative Phase Space Representation. In: *in Workshop on Human Modeling, Analysis and Synthesis at CVPR, Hilton Head Island, South*, 2000, S. 26–33
- [MNT04] MADSEN, K. ; NIELSEN, H. B. ; TINGLEFF, O.: *Methods for Non-Linear Least Squares Problems*. Informatics and Mathematical Modeling, Technical University of Denmark, 2004
- [MVD07] MAKADIA, A. ; VISONTAI, M ; DANIILIDIS, K.: Harmonic Silhouette Matching for 3D Models. In: *3DTV Conference, 2007*. Kos Island, 2007, S. 1–4
- [NM65] NELDER, J. A. ; MEAD, R.: A Simplex Method for Function Minimization. In: *The Computer Journal* 7 (1965), Nr. 4, S. 308–313

- [SM95] STEIN, Fridtjof ; MEDIONI, Gerard: Map-based localization using the panoramic horizon. In: *IEEE Transactions on Robotics and Automation* Bd. 11, 1995, S. 892–896
- [SP09] SCHMITT, Frank ; PRIESE, Lutz: Sky detection in CSC-segmented color images. In: *Fourth International Conference on Computer Vision Theory and Applications (VISAPP) 2009, Lisboa, Portugal* Bd. 2, 2009, S. 101–106
- [Zha00] ZHANG, Zhengyou: A flexible new technique for camera calibration. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), Nr. 11, S. 1330–1334
- [ZL02] ZHANG, Dengsheng ; LU, Guojun: An Integrated Approach to Shape Based Image Retrieval. In: *Proc. of 5th Asian Conference on Computer Vision (ACCV)*. Melbourne, Australia, 2002, S. 652–657