

Modellierung und Oberflächenerzeugung architektonischer Gebäude aus Photographien

Seminararbeit
von
Marcel Busse
aus
Hemmingen

vorgelegt am
Lehrstuhl für Praktische Informatik IV
Prof. Dr. W. Effelsberg
Fakultät für Mathematik und Informatik
Universität Mannheim

Dezember 2002

Betreuer: Dipl.-Inf. Dirk Farin

Inhaltsverzeichnis

Abbildungsverzeichnis	iv
Kapitel 1 Einleitung.....	1
Kapitel 2 Das interaktive Modellierungssystem	5
2.1 Das Benutzerinterface von <i>Façade</i>	5
2.2 Die Modellierung.....	8
2.3 Der Rekonstruktionsalgorithmus.....	12
2.4 Schätzung guter Startwerte	15
Kapitel 3 Texturerzeugung abhängig vom Blickwinkel	17
3.1 Projektion von Einzelbildern.....	17
3.2 Zusammenführung mehrerer projizierter Einzelbilder	18
Kapitel 4 Detailkonstruktion mit einem modell-basierten Stereo-Algorithmus.....	21
4.1 Bestimmung der Korrespondenzpunkte.....	24
Kapitel 5 Zusammenfassung.....	27
Literaturverzeichnis	v

Abbildungsverzeichnis

Abbildung 1:	Das hybride System als Kombination aus geometrie- und bild-basierten Systemen	2
Abbildung 2:	Eine in <i>Façade</i> verwendete Fotografie mit dazugehörigem Grobmodell	7
Abbildung 3:	Das auf das Originalbild projizierte Grobmodell und eine synthetisch generierte neue Ansicht des Modells.....	7
Abbildung 4:	Ein Block mit seinen internen Parametern und einer umliegenden Begrenzungsbox.....	9
Abbildung 5:	Ein einfaches Modell mit entsprechendem Modellbaum	9
Abbildung 6:	Referenzbild einzelner Blockparameter	10
Abbildung 7:	Drei von zwölf Photographien mit markierten Bildkanten.....	11
Abbildung 8:	Vergleich zwischen Photographien und Modell	11
Abbildung 9:	Aus zwölf Photographien erstelltes Modell mit Gesamtsicht.....	12
Abbildung 10:	Projektion einer einzelnen Linie auf die Bildebene der Kamera	13
Abbildung 11:	Abweichung zwischen einer Modell- und entsprechender Bildkante	14
Abbildung 12:	Beispiel einer Texturerzeugung, die abhängig vom Blickwinkel des Betrachters ist	18
Abbildung 13:	Verwendete Gewichtsfunktion.....	19
Abbildung 14:	Ausmaskierte Bereiche.....	20
Abbildung 15:	Die Epipolar-Geometrie von Korrespondenzpunkten	22
Abbildung 16:	<i>Key</i> -, <i>Warped Offset</i> - und <i>Offset</i> -Bild	24
Abbildung 17:	Epipolar-Geometrie bei Verwendung des modell-basierten Stereo-Algorithmus	25

Kapitel 1

Einleitung

Diese Seminararbeit beschäftigt sich mit dem Ziel, akkurate und realistische 3D-Modelle von großflächigen architektonischen Szenen zu erstellen, die es Benutzern ermöglichen, in eine virtuelle Umgebung dieser Szenerie einzutauchen, ohne einen merkbaren Unterschied zur Realität zu verspüren. Weiterhin sind solche Modelle auch für die Filmproduktion hilfreich, die es Akteuren gestatten, sich in einer virtuellen Welt zu bewegen. Teile der hier vorgestellten Methoden wurden bspw. für den Film „Matrix“ verwendet.

Basierend auf einer Menge von wenigen Photographien architektonischer Gebäude wird zunächst ein Basismodell erzeugt, das später mit den Texturen der Bilder versehen und verfeinert werden kann. Dabei ist es natürlich nicht entscheidend, wie alt die vorliegenden Photographien sind, so dass es auch möglich sein wird, Modelle von sehr alten und möglicherweise nicht mehr existierenden Gebäuden zu erstellen. Traditionelle Ansätze lassen sich dahingegen unterscheiden, ob sie geometrie- oder bild-basierend aufgebaut sind. Geometrie-basierte Algorithmen verwenden hauptsächlich Benutzerdaten, die zur Konstruktion des Modells notwendig sind, bspw. Daten über räumliche Vermessungen der Gebäude, CAD-Daten oder digitalisierte architektonische Pläne. Die Texturerzeugung wird ausschließlich vom Computer generiert. Teilweise gibt es Ansätze, die neben den Benutzereingaben auch als unterstützenden Teil Bilder verwenden. Bild-basierte Algorithmen setzen demgegenüber hauptsächlich die eigentlichen Bilder zur Modellerzeugung ein. Um daraus das räumliche Modell zu konstruieren, ist es notwendig, das Korrespondenzproblem zu lösen, d.h. es müssen bei zwei gegebenen Bildern die Punkte identifiziert werden, die denselben Punkt im Realobjekt repräsentieren. Hierzu werden Stereo-Algorithmen eingesetzt, wobei die Güte dieser Verfahren sehr stark die Korrektheit der bild-basierten Verfah-

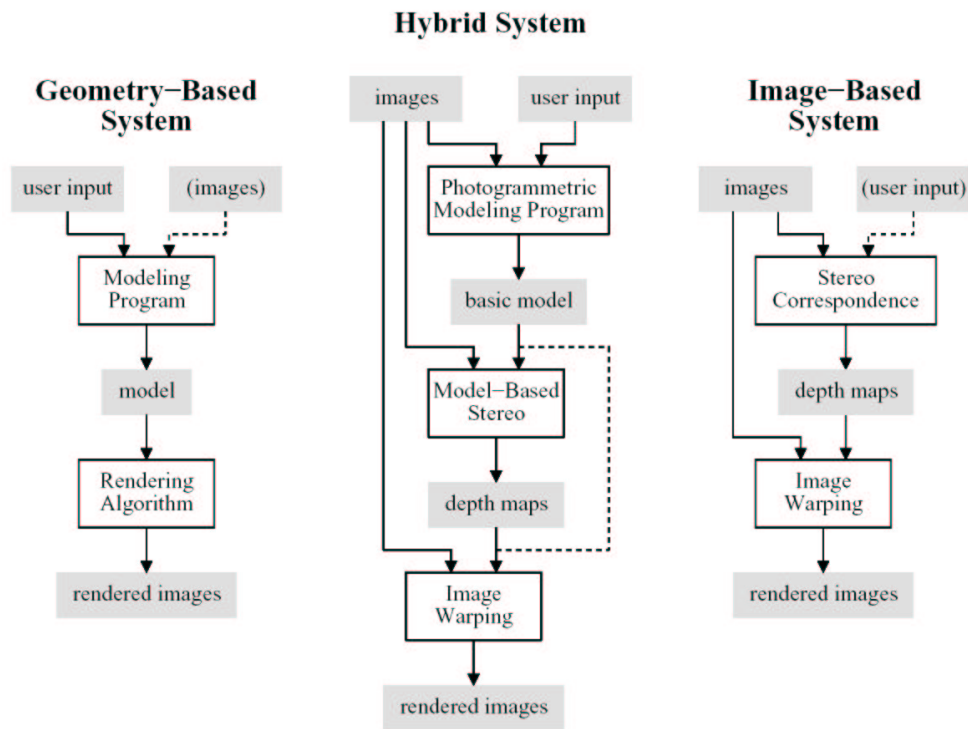


Abbildung 1: Das hybride System als Kombination aus geometrie- und bild-basierten Systemen

ren beeinflusst. Deshalb ist es für allgemeine und traditionelle Stereo-Algorithmen meist erforderlich, dass die gegebenen Bilder sehr ähnlich sind, so dass die entstehenden Fehler bei der Korrespondenzbestimmung möglichst gering gehalten werden können, wodurch das konstruierte Modell akkurater wird. Genau genommen wird durch den Stereo-Algorithmus kein geschlossenes Modell konstruiert, sondern lediglich für jeden Punkt Tiefenangaben ermittelt, so dass zweidimensionale Bilder räumlich dargestellt werden können. Auch bei diesen Verfahren gibt es Ansätze, in denen zusätzlich zur Unterstützung Benutzereingaben herangezogen werden können, bspw. konkrete Angaben von Korrespondenzpunkten. Als Konsequenz ergibt sich hieraus, dass aufgrund der sehr vielen Benutzereingaben in geometrie-basierten Systemen und der Einschränkung bzgl. der Ähnlichkeit von Photographien bei bild-basierten Systemen, weshalb sehr viele solcher Bilder notwendig werden, kein System isoliert eingesetzt werden kann. Deshalb wird im Folgenden ein hybrider Ansatz gewählt, in dem die Vorteile aus beiden Systemen kombiniert werden. Abbildung 1 zeigt, wie im ersten Schritt des hybriden Systems, hauptsächlich durch Benutzereingaben getrieben, in einem Modellierungsprogramm ein Basismodell der architektonischen Szene erzeugt wird, das anschließend in einem automatischen zweiten Schritt mit einem modell-basierten Stereo-Algorithmus weiter verfeinert wird, indem zu jedem Punkt entsprechende Tiefenangaben ermittelt werden. Daraus kann anschließend die Texturerzeugung des Basismodells generiert werden, wobei in allen Schritten die vorliegenden Bilder als Eingabe

verwendet werden. Für die Texturerzeugung ist es allerdings auch möglich, auf den modellbasierten Stereo-Algorithmus zu verzichten und ohne weitere Tiefeninformationen, ausgehend vom Basismodell, neue Ansichten der Szene zu erzeugen (vgl. Kapitel 3).

Da die hier vorgestellten Algorithmen hauptsächlich für architektonische Szenen entwickelt wurden, können viele vorkommende Probleme vereinfacht und gelöst werden, indem bestimmte Einschränkungen und Annahmen an das Modell gestellt werden, die in den folgenden Kapiteln noch näher beschrieben werden. Besonders entscheidend für die Korrektheit und realistische Darstellung des Modells ist die Unterteilung des Konstruktionsproblems in zwei Modelle unterschiedlicher Auflösung:

1. **Ein Grobmodell der Szene.** Mit einfachen geometrischen Objekten soll versucht werden, die Architektur der Szene nachzubauen, indem mittels eines Modellierungssystems Objekte durch einen Benutzer erzeugt und entsprechend angeordnet werden. Außerdem werden Größe, Form und Position der Objekte an spezielle geometrische Bedingungen geknüpft, z.B. Angaben bzgl. der Symmetrie. Entscheidend hierbei ist, dass viele Details vernachlässigt werden, woraus sich ein Modell mit relativ wenigen Parametern und ein robuster Konstruktionsalgorithmus ergeben.
2. **Ein detailliertes Modell der Szene.** Ausgehend vom vorher konstruierten Grobmodell werden in diesem Schritt die fehlenden Details mittels eines modellbasierten Stereo-Algorithmus ermittelt. Durch die Verwendung des Grobmodells können viele Probleme von traditionellen Stereo-Algorithmen vermieden werden, indem das Korrespondenzproblem nicht direkt zwischen zwei Bildern, sondern zwischen einem Bild und einem auf das Grobmodell projizierten neuen Bild, gelöst wird.

Zunächst wird in Kapitel 2 das interaktive Modellierungssystem *Façade* vorgestellt, das speziell für den ersten Schritt, also die Konstruktion eines Grobmodells, implementiert wurde. Als ein Zwischenresultat wird in Kapitel 3 eine Erweiterung der bildbasierten Texturerzeugung präsentiert, die den Blickwinkel eines Betrachters explizit berücksichtigt, um so realistischere neuartige Bilder zu erzeugen. Kapitel 4 stellt am Ende den modellbasierten Stereo-Algorithmus vor, mit dem das detaillierte Modell generiert wird. Außerdem wird hier untersucht, wie die blickwinkelabhängigen Texturerzeugungen, angepasst an das detaillierte Modell, angewendet werden können. Eine kurze Zusammenfassung der Vorgehensweise und vorgestellten Methoden liefert Kapitel 5.

Kapitel 2

Das interaktive Modellierungssystem

Dieses Kapitel soll die Funktionsweise des interaktiven Modellierungssystems *Façade* vorstellen, mit dessen Hilfe ein Benutzer auf der Basis von Photographien architektonischer Gebäude ein dreidimensionales Grobmodell der Szene erstellen und in weiteren Schritten detaillieren kann. Hierfür erzeugt er zunächst einfache primitive Objekte, wobei die Bestimmung der entsprechenden Parameter von *Façade* übernommen wird, um eine möglichst genaue Übereinstimmung mit dem Realmodell zu erreichen. Zuerst werden das Benutzerinterface und anschließend die Modellierung und der Rekonstruktionsalgorithmus, die in *Façade* zum Einsatz kommen, vorgestellt.

2.1 Das Benutzerinterface von *Façade*

Ausgehend von einer Menge vorliegender Photographien erzeugt ein Benutzer in *Façade* ein Grobmodell der Szene. Dieses ist ein inkrementeller und leicht durchzuführender Prozess, in dem der Benutzer nach und nach einzelne Photographien bearbeitet und hinzufügt, um so das Grobmodell weiter zu verfeinern. Anhand der vorliegenden Bilder kann er bestimmte Objekte im Bild markieren und mit dem Grobmodell verknüpfen, so dass festgelegt ist, bspw. welche Kante im Grobmodell mit welcher Kante im Realmodell übereinstimmt. Außerdem können an Objekte Bedingungen gestellt werden, die angeben, wie sie im Grobmodell dargestellt werden soll und in welcher Beziehung sie zu anderen Objekten stehen sollen.

Abbildung 2 zeigt zwei Fenster, die in *Façade* verwendet werden. Das linke Fenster ist eine einzelne Photographie, in der Kanten markiert wurden. Anhand dieser Markierungen konnte das dazugehörige Grobmodell erstellt werden, das im rechten Teil dargestellt ist. Die Parameter der jeweiligen Objekte des Grobmodells, wie z.B. Größe und Position, wurden automatisch generiert, so dass eine bestmögliche Übereinstimmung mit dem Realmodell erreicht werden kann. Zu bemerken sei, dass der Benutzer nur jeweils eine Seitenkante des Turms markiert hat, die anderen drei (inklusive der nicht sichtbaren) wurden aufgrund von Symmetriebedingungen generiert. In der Anzahl der zu verwendenden Photographien ist ein Benutzer nicht eingeschränkt. Dadurch, dass er stets eine Verknüpfung zwischen Bild- und Modellkante herstellt, kann er Schritt für Schritt die gesamte Szene modellieren. Mit Hilfe von geometrischen Bedingungen können die Anzahl der Photographien verringert und Objekte im Grobmodell in korrekte Beziehung zueinander gesetzt werden. So war es in dem Beispiel aus Abbildung 2 sogar möglich, das Grobmodell nur aufgrund des linken Bildes vollständig zu erzeugen.

Die Objekte des Grobmodells, Blöcke genannt, setzen sich aus einfachen geometrischen Körpern zusammen, die durch mehrere Parameter charakterisiert werden, bspw. durch die Länge, Breite und Höhe. Der Benutzer erstellt das Grobmodell mittels entsprechender Blöcke, um so die gesamte Szene zu modellieren. Falls gewünscht, kann er neue Klassen von Blöcken erzeugen und in das Grobmodell aufnehmen. Das einzige, das er bei der Modellierung angeben muss, sind Verknüpfungen zu den Bildkanten; die Werte der jeweiligen Parameter werden automatisch bestimmt. Viele dieser Parameter unterliegen ebenfalls gewissen Bedingungen. So haben einige Blöcke in Abbildung 2 die gleiche Länge, Breite oder ihre Position ist durch einen unterliegenden Block festgelegt. Bedingungen an die Position von Blöcken dienen später auch dem Rekonstruktionsalgorithmus, um ein korrektes Abbild des Realmodells zu erzeugen. Zu erkennen ist auch, dass im Bild nur eine der vier Spitzen markiert wurde; die anderen wurden durch Symmetriebedingungen automatisch hinzugefügt. Ein entscheidender Vorteil durch die Einschränkung der Parameterwahl ist es, dass sich ihre Anzahl immens verringert und der Rekonstruktionsalgorithmus robust und effizient realisiert werden kann.

Das Markieren von Bildkanten in *Façade* ist durch einfaches Anklicken im Bild realisiert worden. Der Benutzer kann auch einzelne Punkte einer Kante hinzufügen und hierfür das Bild beliebig heranzoomen. Schon mit wenigen markierten Bildkanten kann ein Grobmodell erzeugt werden, besonders dann, wenn die Anordnung einzelner Blöcke an festgelegte Bedingungen geknüpft ist. Die Herstellung einer Verknüpfung zwischen Modell- und Bildkante erfolgt, indem zuerst eine beliebige Kante im Grobmodell und anschließend die dazugehörige Bildkante markiert werden. Um diesen Vorgang für den Benutzer zu vereinfachen, kann das Grobmodell in jede beliebige Lage gebracht werden.

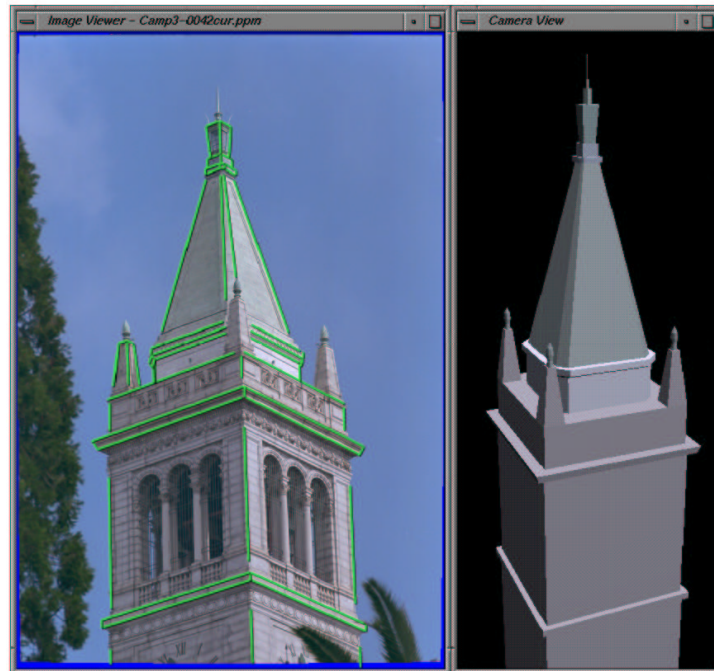


Abbildung 2: Eine in *Façade* verwendete Fotografie mit dazugehörigem Grobmodell

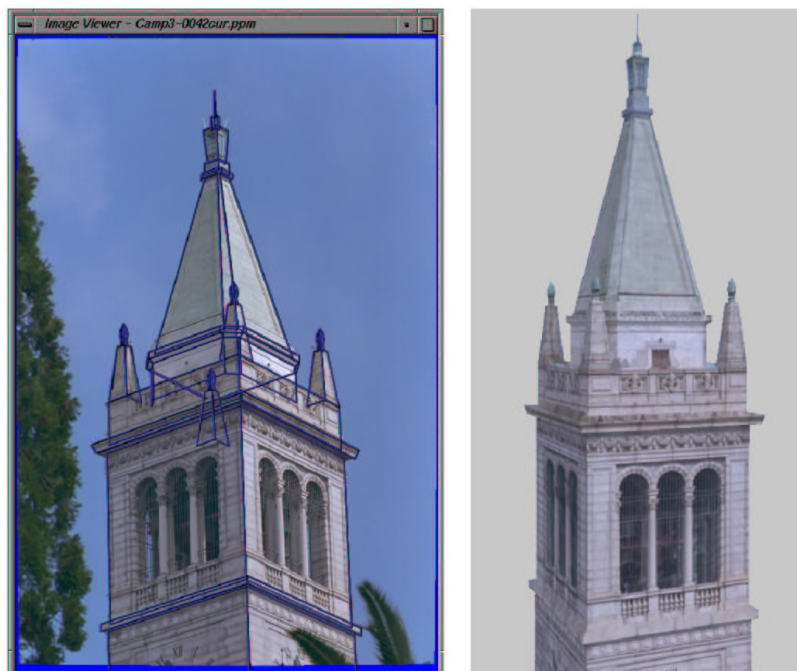


Abbildung 3: Das auf das Originalbild projizierte Grobmodell und eine synthetisch generierte neue Ansicht des Modells

Die Generierung des Grobmodells durch den Rekonstruktionsalgorithmus kann zu jeder Zeit durch den Benutzer angestoßen werden. So erhält er ständig eine aktuelle Modellsicht und kann weitere Bereiche im Bild modellieren. Der Rekonstruktionsalgorithmus bestimmt neben den Objektparametern auch die Kamerapositionen, aus denen die Photographien aufgenommen wurden und stellt sie in einer Gesamtsicht der Szene mit dar. Der Genauigkeitsgrad des Rekonstruktionsalgorithmus kann benutzerspezifisch eingestellt werden. Abhängig von der Anzahl der verwendeten Photographien dauert die Berechnung zwischen einigen Sekunden und mehreren Minuten. Deshalb ist es ggf. vorteilhaft, während der Modellierungsphase eine gröbere Auflösung für die Rekonstruktion zu wählen.

Im linken Teil von Abbildung 3 wurde das Grobmodell aus Abbildung 2 über das Originalbild aus Sicht der berechneten Kameraposition projiziert, woran man erkennen kann, wie genau der Rekonstruktionsalgorithmus das Modell erstellt. Die Abweichungen betragen in den meisten Fällen maximal einen Pixel. Nachdem der Benutzer die Modellierung des Grobmodells abgeschlossen hat, kann er die in den Kapiteln 3 und 4 vorgestellten Verfahren zur Texturerzeugung und Detailkonstruktion verwenden, um so neuartige Ansichten des erstellten Modells zu erzeugen. Der rechte Teil von Abbildung 3 zeigt hierfür ein Beispiel, wobei die Texturerzeugung abhängig vom Blickwinkel des Betrachters bestimmt wurde, um so einen besseren räumlichen Eindruck zu erhalten.

2.2 Die Modellierung

Façade stellt architektonische Gebäude aus einer Menge von geometrischen Blöcken dar, deren Parameter ihre Größe und Form bestimmen. Die Koordinaten eines Blockes können durch lineare Funktionen dieser Parameter bestimmt werden. Der Punkt P_0 aus Abbildung 4 kann bspw. mittels $(-width, -height, -length)^T$ dargestellt werden. Jeder Block wird durch eine umliegende Box begrenzt, deren Größe sich aus den internen Parametern des Blockes ergeben. Z.B. wird die minimale x-Ausdehnung $width_x^{MIN}$ durch $-width$ bestimmt. Anhand dieser Parameter kann jeder Block charakterisiert und leicht für die Modellierung verwendet werden. Die einzelnen Parameter werden für jeden Blocktyp separat in einer Datei gespeichert und geben an, wie sich der Block und die umliegende Begrenzungsbox zusammensetzen. Falls es erforderlich sein sollte, können so weitere Klassen von Blöcken hinzugefügt werden.

Abbildung 5(a) zeigt ein einfaches Modell eines Gebäudes, das aus drei Blöcken besteht. Die Beziehungen der Blöcke zueinander werden in *Façade* durch einen Modellbaum dargestellt

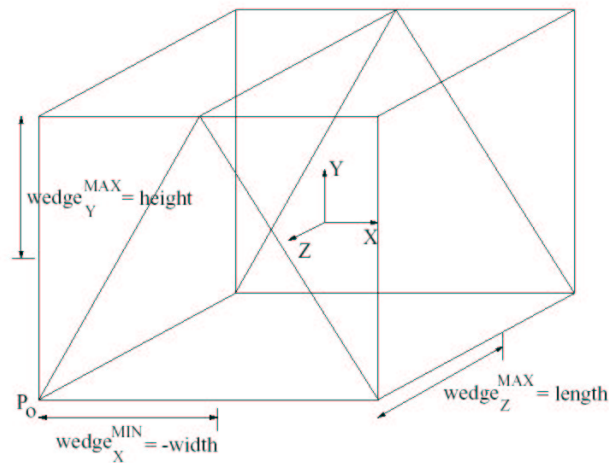


Abbildung 4: Ein Block mit seinen internen Parametern und einer umliegenden Begrenzungsbox

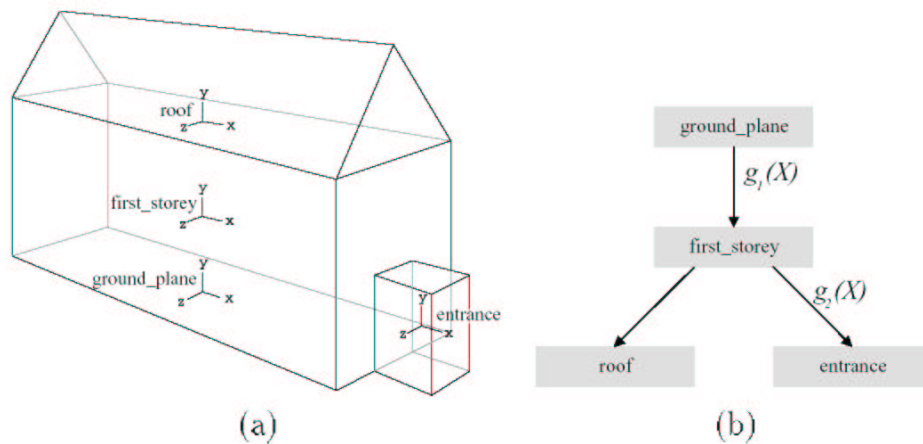


Abbildung 5: Ein einfaches Modell mit entsprechendem Modellbaum

(vgl. Abbildung 5(b)). Jeder Knoten im Baum repräsentiert einen einzelnen Block. Die Wurzel jedoch wird nur als Referenz für die Weltkoordinaten verwendet. Die Kanten des Baumes entsprechen den vorhandenen Beziehungen. Bspw. gibt der Modellbaum an, dass der Eingang und das Dach des Gebäudes mit dem ersten Stockwerk in Beziehung stehen. Daraus lassen sich gewisse Einschränkungen an die Parameter der Blöcke ableiten.

Jede Beziehung zwischen zwei Blöcken wird im Modellbaum als Vater-Sohn-Beziehung dargestellt und kann durch eine Rotationsmatrix R und einem Verschiebungsvektor t beschrieben werden. Daraus ergeben sich sechs unterschiedliche Freiheitsgrade, je drei für R und t . Ein Freiheitsgrad gibt dabei an, anhand welcher Koordinate die Rotation oder Verschiebung erfolgen soll. Vor allem bei architektonischen Gebäuden kann die Anzahl der Freiheitsgrade

entscheidend verringert werden, was die Gesamtzahl der Parameter einschränkt und somit die Effizienz des Rekonstruktionsalgorithmus erhöht. Folgende drei Fälle können für die Rotationsmatrix R unterschieden werden. (i) Die Rotation erfolgt nur an einer Achse entlang. (ii) Die Rotation ist fix oder wird nicht durchgeführt. (iii) Es werden keine Einschränkungen an die Rotation gestellt. Analog können solche Einschränkungen auch an den Verschiebungsvektor t gestellt werden. Auf Abbildung 5 bezogen gilt dann für die Verschiebung des Daches relativ zum ersten Stockwerk: $t_y = (first_storey_y^{MAX} - roof_y^{MAX})$.

Jeder Parameter eines Blockes wird in *Façade* als Referenz auf einen symbolischen Variablenwert abgebildet, wodurch erzwungene Bedingungen an Blöcke, wie Symmetrieeigenschaften oder Duplizierung dergleichen Blockstruktur, leicht und komfortable durch den Benutzer festgelegt werden können. Für das obige Beispiel ist dieses in Abbildung 6 gezeigt. Auch hierdurch kann die Anzahl der Freiheitsgrade bzgl. der Parameterwahl verringert werden.

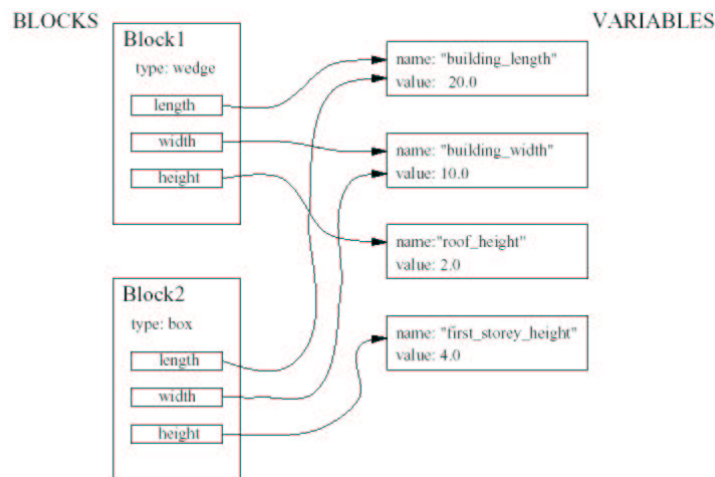


Abbildung 6: Referenzbild einzelner Blockparameter

Nachdem alle Blöcke und deren Beziehungen modelliert wurden, können anhand des Modellbaums sehr leicht die Koordinaten eines Blockes oder einer einzelnen Linie relativ zu den Koordinaten der Wurzel berechnet werden. Sei X ein Vektor, der die gesamten Modellparameter eines Blockes repräsentiert und seien $g_1(X) \dots g_n(X)$ die jeweiligen Transformationen (Rotation und Verschiebung) auf dem Pfad von der Wurzel zu einem bestimmten Block $P(X)$ (vgl. Abbildung 5(b)). Dann ergeben sich für dessen Koordinaten $P_w(X)$:

$$P_w(X) = g_1(X) \dots g_n(X) P(X) \quad (1)$$

Entsprechend ergeben sich die Koordinaten $v_w(X)$ einer Linie $v(X)$ der Stufe n :

$$v_w(X) = g_1(X) \dots g_n(X) v(X) \quad (2)$$

Die Parameterreduktion, die mit dem obigen Vorgehen erhalten wird, soll zum Abschluss noch einmal anhand der Abbildung 3 verdeutlicht werden. Das berechnete Modell konnte mit nur 45 Variablen parametrisiert werden, wohingegen bei Betrachtung von nicht eingeschränkten Parameterwerten sich 240 Variablen ergeben. Wird anstatt kompletter Blöcke jede Linie unabhängig voneinander verarbeitet, sind 2.896 Variablen notwendig.

Ergebnisse von der Verwendung von *Façade* geben die Abbildung 3, 4, 7, 8 und 9. Sehr gut zu erkennen ist, wie realistisch der im nächsten Abschnitt vorgestellte Rekonstruktionsalgorithmus das Modell erzeugt und aus welchen Kamerapositionen die einzelnen Photographien aufgenommen wurden. Die Abweichungen zwischen dem rekonstruierten Modell und dem Realmodell betragen bei allen zwölf Bildern maximal einen Pixel. Abbildung 9 zeigt eine komplette Übersicht der Szene. Aufgrund der errechneten Höhe der Kameraposition gegenüber des Haupteinganges kann vermutet werden, dass dieses Bild aus dem ersten Stock eines Nachbarhauses gemacht worden ist.

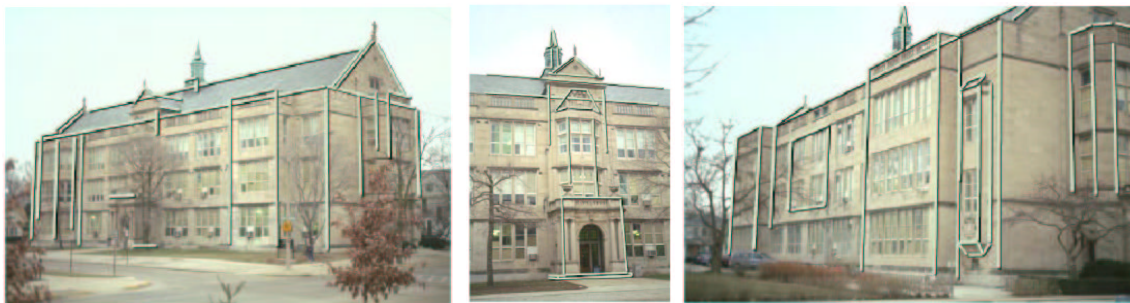


Abbildung 7: Drei von zwölf Photographien mit markierten Bildkanten

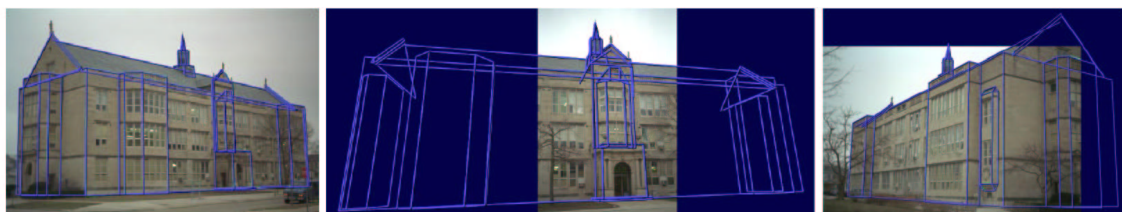


Abbildung 8: Vergleich zwischen Photographien und Modell

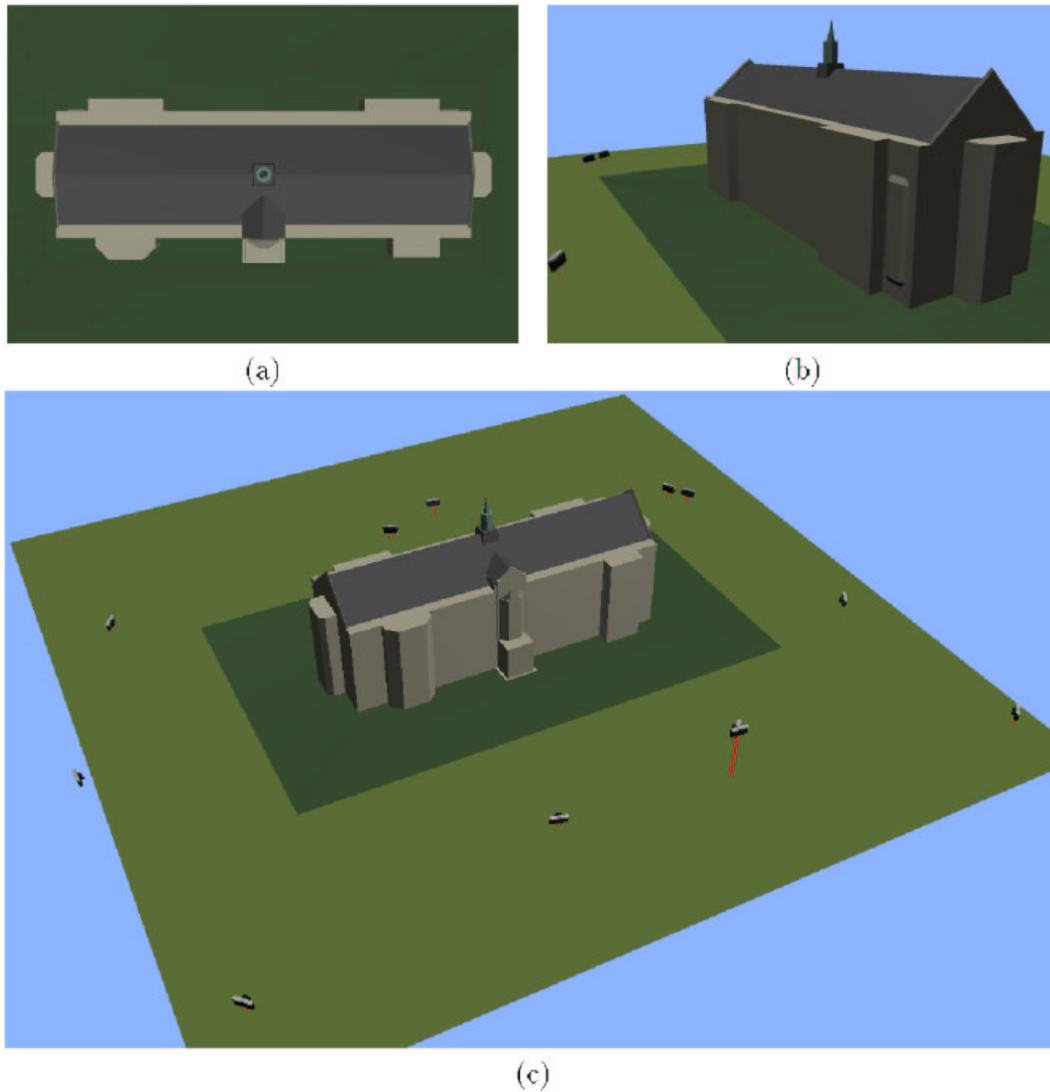


Abbildung 9: Aus zwölf Photographien erstelltes Modell mit Gesamtsicht

2.3 Der Rekonstruktionsalgorithmus

Ziel des Rekonstruktionsalgorithmus, der in *Façade* zum Einsatz kommt, ist es, die Abweichungen zwischen den Modell- und entsprechenden Bildkanten zu minimieren. Hierfür werden alle Kantenabweichungen aufsummiert, so dass sich als zu minimierende Zielfunktion $\Theta = \sum_i Err_i$ ergibt, mit Err_i als Maß für die Abweichung der Modellkante i . Mittels der Minimierung der Zielfunktion ergeben sich sowohl sehr gute Schätzungen für die Modellparameter, als auch für die einzelnen Kamerapositionen, aus denen die vorliegenden Photographien aufgenommen wurden.

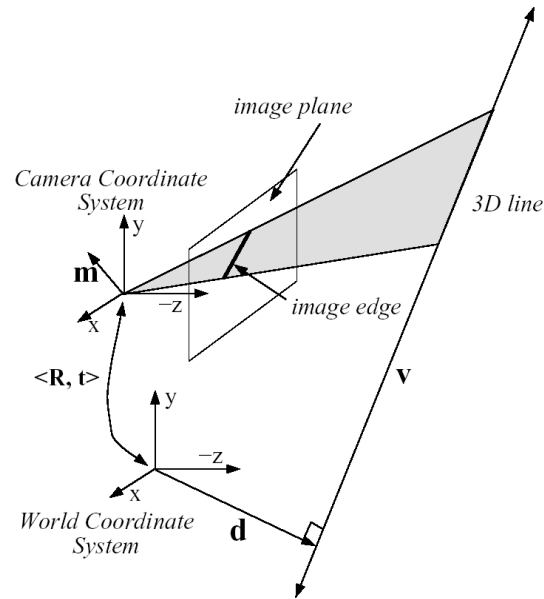


Abbildung 10: Projektion einer einzelnen Linie auf die Bildebene der Kamera

Wie eine räumliche Gerade in die Bildebene der Kamera projiziert wird, zeigt Abbildung 10. Die Gerade kann durch einen Vektor $\langle v, d \rangle$ beschrieben werden. Dabei sei v die Richtung und d ein Punkt auf der Geraden. Die Bildkante ergibt sich dann als einfacher Schnitt zwischen der Bildebene und der Ebene, die durch die Gerade und der Kameraposition gebildet wird. Sei m der Normalenvektor dieser Ebene. Die Parameter v und d lassen sich mit den Formeln (1) und (2) berechnen. Die Position der Kamera j bzgl. den Weltkoordinaten ist durch die Rotationsmatrix R_j und den Verschiebungsvektor t_j gegeben, so dass sich für den Normalenvektor m ergibt:

$$m = R_j(v \times (d - t_j)) \quad (3)$$

Daraus folgt schließlich für die Bildkante die Bedingung $m_x x + m_y y + m_z z = 0$, wobei für z die Brennweite der Kamera f verwendet wird.

Die Abweichung zwischen Modell- und Bildkante, die vom Benutzer markiert wurde, kann mittels Abbildung 11 veranschaulicht werden. Seien (x_1, y_1) und (x_2, y_2) die Eckpunkte einer Bildkante. Dann kann jeder Punkt auf dieser Geraden mit einem Parameter $s \in [0, l]$ charakterisiert werden, wobei l der Länge der Geraden entspricht. Sei $h(s)$ dann der kürzeste Abstand zwischen dem Punkt $p(s)$ und der Bildkante.

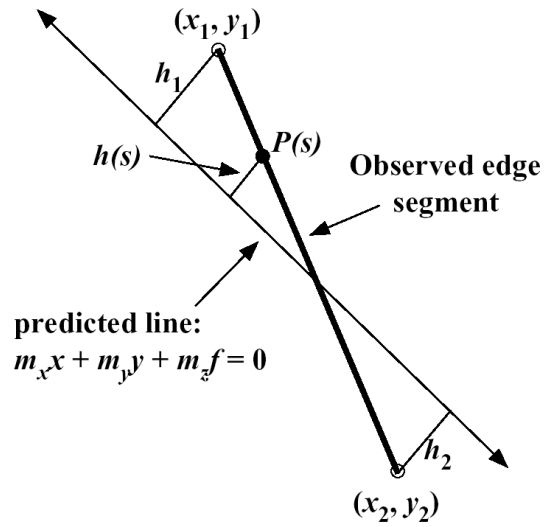


Abbildung 11: Abweichung zwischen einer Modell- und entsprechender Bildkante

Für den Fehler Err_i als Abweichung der Modellkante i ergibt sich anhand dieser Definitionen:

$$Err_i = \int_0^l h^2(s) ds = \frac{l}{3} (h_1^2 + h_1 h_2 + h_2^2) = m^T (A^T B A) m \quad (4)$$

mit

$$m = (m_x, m_y, m_z)^T$$

$$A = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix}$$

$$B = \frac{l}{3(m_x^2 + m_y^2)} \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix}$$

Minimiert wird die Zielfunktion Θ durch eine Variante der Newton-Raphson-Methode. Hierfür wird iterativ der Gradient und die Hesse-Matrix berechnet, um die gesamten Modellparameter und Kamerapositionen unter der Optimierung zu erhalten. Dieses Verfahren kann aufgrund der einfachen Ausdrücke und Formeln recht effizient realisiert werden.

2.4 Schätzung guter Startwerte

Ein Problem, das sich durch die Minimierung ergibt, ist die Nichtlinearität der Zielfunktion Θ , wodurch der Algorithmus bei zufälliger Wahl der Startwerte in ein lokales Optimum konvergieren kann. Um dieses zu vermeiden, soll in diesem Abschnitt untersucht werden, wie man geeignete Startwerte für die Modellparameter und Kamerapositionen wählen sollte, so dass das globale Optimum erreicht werden kann. In der Praxis konnte es immer erreicht werden, was anhand der minimalen Abweichungen in den zahlreichen Ergebnisse zu erkennen ist.

Die Schätzung guter Startwerte, die schon von Beginn an sehr nahe an der optimalen Lösung liegen sollten, resultiert aus einer genaueren Untersuchung der Gleichung (3) und Abbildung 10. Daraus lässt sich erkennen, dass jede Modellkante, charakterisiert durch ihren Richtungsvektor v und die Rotationsmatrix R , senkrecht zu dem Normalenvektor m stehen muss. Außerdem muss auch der Richtungsvektor $(d-t)$, rotiert mittels R , senkrecht zu m stehen. Daraus folgt, dass die entsprechenden Skalarprodukte null sein müssen, woraus sich die Gleichungen (5) und (6) ergeben:

$$m^T R_j v = 0 \quad (5)$$

$$m^T R_j (d - t_j) = 0 \quad (6)$$

Für den Normalenvektor m' , der durch eine Bildkante mit den Eckpunkten (x_1, y_1) und (x_2, y_2) gebildet wird, gilt:

$$m' = (x_1, y_1, -f)^T \times (x_2, y_2, -f)^T \quad (7)$$

Mit Hilfe dieser Gleichungen können nun in einem zweistufigen Verfahren mögliche Startwerte berechnet werden. Zuerst werden die Rotationsmatrizen R_j und anschließend die Verschiebungsvektoren t_j und Modellparameter eines Blockes j geschätzt. Hierzu werden zwei Hilfsfunktionen Θ_1 und Θ_2 minimiert; Θ_1 soll die Verletzung von Gleichung (5) und Θ_2 die Verletzung von Gleichung (6) für jeden Block j minimieren, so dass unter zu Hilfenahme von Gleichung (7) folgt:

$$\Theta_1 = \sum_i (m_i'^T R_j v_i)^2 \quad (8)$$

Sind alle Startwerte für R_j bestimmt, kann mittels Gleichung (6) die zweite Hilfsfunktion Θ_2 minimiert werden. Seien dafür $P_i(X)$ und $Q_i(X)$ die Eckpunkte der Modellkante i , dargestellt als Vektoren, dann lassen sich die Verletzungen von Gleichung (6) folgendermaßen darstellen:

$$\Theta_2 = \sum_i \left((m_i'^T R_j (P_i(X) - t_j))^2 + (m_i'^T R_j (Q_i(X) - t_j))^2 \right) \quad (9)$$

Θ_2 hat wegen der bereits geschätzten Rotationsmatrizen R_j eine quadratische Form und kann deshalb mit einer Menge von linearen Gleichungen leicht gelöst werden.

Kapitel 3

Texturerzeugung abhängig vom Blickwinkel

Dieses Kapitel beschäftigt sich mit der Texturerzeugung des Modells, auch „texture-mapping“ genannt, wodurch die Oberfläche auf Basis der vorliegenden Photographien erstellt wird. Im Gegensatz von traditionellen Ansätzen, soll je nach Blickwinkel, aus dem das Modell gezeigt wird, eine angepasste Textur gewählt werden. So können viele geometrische Details noch besser dargestellt werden, und für den Betrachter ergibt sich ein realistischerer Eindruck. Hierfür ist es vorteilhaft, wenn alle Photographien mit gleichen Lichtverhältnissen erstellt wurden und das zuvor konstruierte Grobmodell gut mit dem Realmodell übereinstimmt. Im nächsten Kapitel soll dann untersucht werden, wie die hier vorgestellte Methode mit einem modellbasierten Stereo-Algorithmus bei approximierten Modellen eingesetzt werden kann. Zunächst wird die Texturerzeugung, abhängig vom Blickwinkel, anhand einer einzelnen Photographie, anschließend bei Verwendung mehrerer Vorlagen, erläutert.

3.1 Projektion von Einzelbildern

Die oberen Bilder von Abbildung 12 zeigen zwei Beispiele, in denen jeweils aus Sicht der dargestellten Kamerapositionen die entsprechenden Photographien auf das Grobmodell projiziert wurden. Da die Kamerapositionen, wie im letzten Kapitel beschrieben, automatisch während der Modellkonstruktion berechnet werden, ist dies ein einfacher Vorgang, den man sich wie die Projektion mittels eines Overhead-Projektors vorstellen kann. Wie in beiden Bildern zu erken-

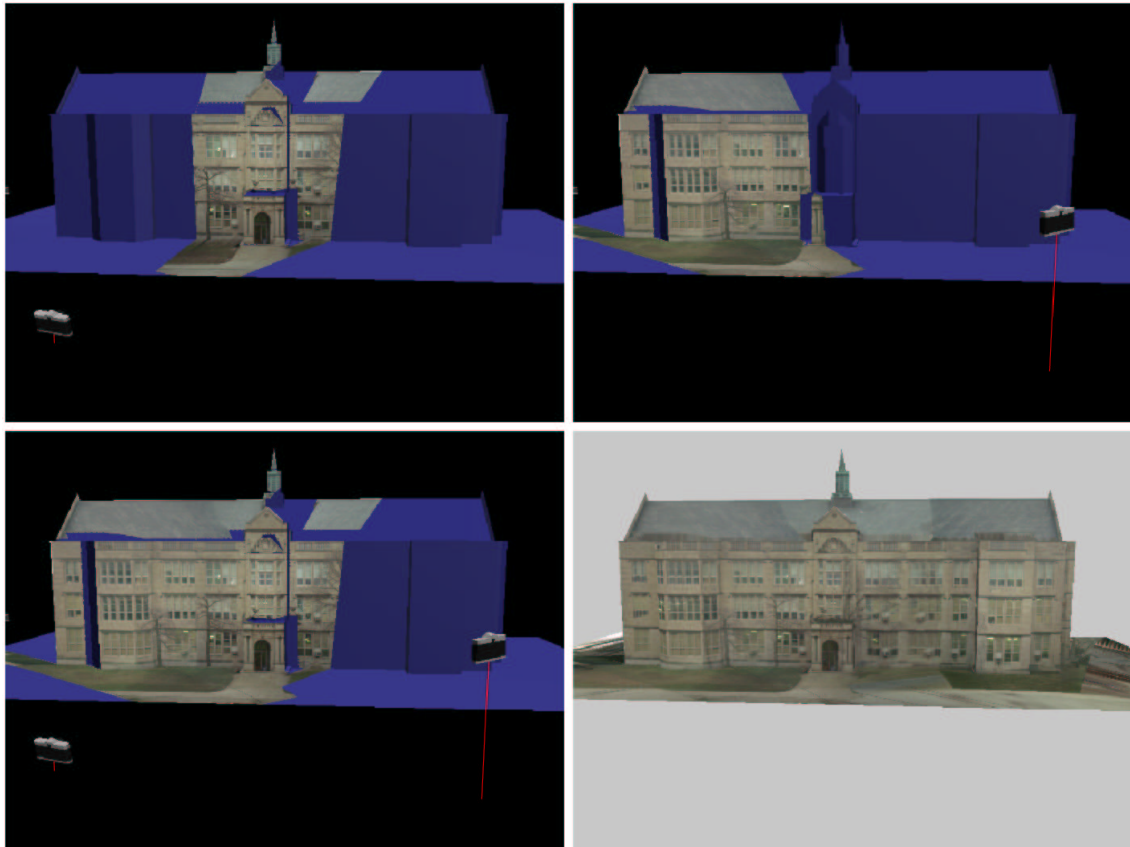


Abbildung 12: Beispiel einer Texturerzeugung, die abhängig vom Blickwinkel des Betrachters ist

en, kommt es bei nicht-konvexen Modellen vor, dass bestimmte Bereiche durch andere verdeckt werden und es somit zur Schattenbildung kommt. Hierzu gibt es eine Reihe von Verfahren, um die entstehenden Schatten im Bild zu bestimmen. In *Façade* wurde ein „image-space shadow map algorithm“ eingesetzt.

3.2 Zusammenführung mehrerer projizierter Einzelbilder

Um die gesamte Oberflächentextur des Modells zu erzeugen, ist es notwendig, die im vorherigen Abschnitt projizierten Einzelbilder zu kombinieren. Das linke, untere Bild in Abbildung 12 zeigt, wie dies für die oberen beiden Bilder geschehen kann. Pixel, die nur in einem Bild vorkommen, können eins zu eins übernommen werden. Bei Bereichen, die durch zwei Bilder überdeckt werden, muss entschieden werden, aus welchem Bild Pixel projiziert werden sollen. Hierfür muss überprüft werden, welche Pixel denselben Punkt auf dem Modell repräsentieren (Korrespondenzproblem). Wünschenswert wäre es, wenn übereinander liegende

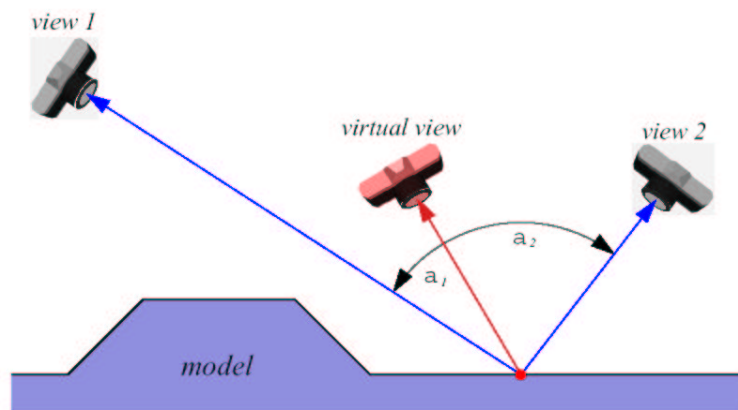


Abbildung 13: Verwendete Gewichtungsfunktion

Bereiche perfekt übereinstimmen würden. Dies ist aber meist nicht der Fall, was aus der fehlenden Modellierung sämtlicher Details und oft unterschiedlicher Lichtverhältnisse resultiert. Deshalb versucht man, den dominierenden Pixel so zu wählen, dass dessen Kameraposition der aktuellen Position des Betrachters am nächsten liegt. Entscheidend dabei ist der Blickwinkel des Betrachters (vgl. Abbildung 13). Werden jedoch nebeneinander liegende Pixel aus unterschiedlichen Photographien auf das Modell projiziert, können sich scharfe Kanten ergeben, die den Texturverlauf unnatürlich erscheinen lassen. Deshalb wird der resultierende Pixelwert als gewichteter Durchschnitt der Korrespondenzpixel bestimmt. Die Gewichte w_1 und w_2 verhalten sich antiproportional zu den Winkeln α_1 und α_2 , so dass Pixel, mit einem kleinerem α -Wert stärker ins Gewicht fallen.

Trotz der Einführung von Gewichten ist eine scharfe Kantenbildung weiterhin möglich, nämlich dann, wenn ein Pixel außerhalb des Gebäudes liegt und folglich sein Gewicht w mit null zu bewerten ist. Daher versucht man, die Gewichte nahe der Modellgrenze auf null absinken zu lassen. Auch wenn die Kantenglättung nicht für alle Fälle garantiert werden kann, so können hiermit gute Resultate erzielt werden.

Mit Hilfe der Gewichtungsfunktion ist es jetzt auch möglich, alle ungewünschten Objekte vor einem architektonischen Gebäude auszublenden. Bspw. wurden in Abbildung 14 vom Benutzer einzelne Bereiche blau markiert, wodurch die Gewichte dieser Pixel auf null gesetzt wurden. Werden später diese Bilder zur Texturerzeugung herangezogen, so werden die maskierten Bereiche aufgrund ihres Gewichtes nicht berücksichtigt und durch Pixel anderer Bilder ersetzt. Entstehen am Ende der Texturerzeugung nicht gefüllte Bereiche, weil keine Photographie diese abdecken konnte, müssen so genannte „hole-filling methods“ eingesetzt werden, die durch Interpolation möglichst passende Pixelwerte generieren.

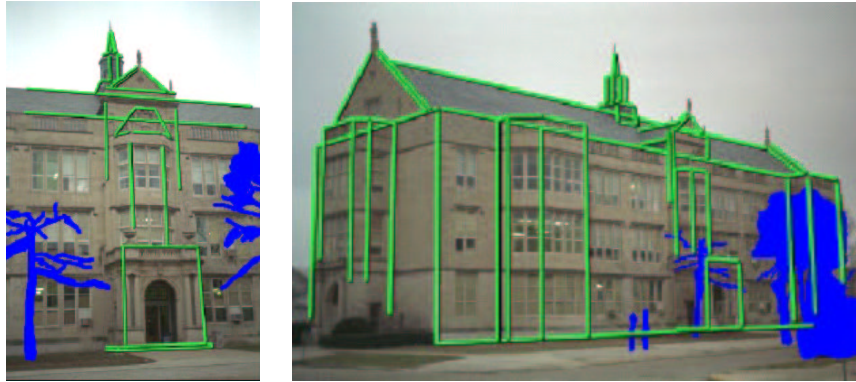


Abbildung 14: Ausmaskierte Bereiche

Um die Effizienz der hier vorgestellten Methode zu erhöhen, wird nicht für jedes Pixel ein einzelnes Gewicht berechnet, sondern versucht, Pixel zu Gruppen zusammenzufassen. Anhand des Grobmodells können für alle Pixel, die zu einer Fläche gehören, ein gemeinsames Gewicht berechnet werden. Dieses könnte z.B. auf die Mitte der Fläche bezogen werden und aus dem resultierendem α -Wert dieses Punktes, wie oben beschrieben, berechnet werden. In der Praxis ergeben sich keine visuell erkennbaren Nachteile; eine Implementierung könnte allerdings so in Echtzeit realisiert werden. Weiterhin ist es bei komplexen Modellen mit vielen Fotografien, deren Objekte aus dem zu betrachtendem Blickwinkel nicht sichtbar sind, nicht empfehlenswert, jedes Bild auf das Modell zu projizieren. Hierfür können spezielle Algorithmen eingesetzt werden, die schon *a priori* solche Gegebenheiten erkennen.

Kapitel 4

Detailkonstruktion mit einem modell-basierten Stereo-Algorithmus

Ausgehend von dem im Kapitel 2 erstellten Grobmodell sollen die fehlenden geometrischen Details, die nicht modelliert wurden, mit dem hier vorgestellten modell-basierten Stereo-Algorithmus automatisch konstruiert werden. Dazu werden im Gegensatz zu traditionellen Stereo-Algorithmen Abweichungen zwischen der Szene und dem Grobmodell ermittelt. Zunächst müssen wie bei den meisten Verfahren die Korrespondenzpunkte zwischen zwei Bildern gefunden werden, die projiziert auf das Modell, demselben Punkt entsprechen. Bezeichnet man die zwei Bilder als „*key*“- und „*offset*“-Bild, berechnet der modell-basierte Stereo-Algorithmus die Tiefen der Bildpunkte ausgehend von der Sicht des *key*-Bildes. Um die Korrespondenzpunkte zweier Bilder zu bestimmen, versucht man zu einem Punkt im *key*-Bild den passenden Korrespondenzpunkt im *offset*-Bild zu finden. Hierfür werden häufig die umliegenden Nachbarschaftspunkte verglichen, um so feststellen zu können, wie gut die Übereinstimmung der beiden Punkte ist. Jedoch reicht dieses Kriterium alleine meist nicht aus, da aufgrund von Verdeckungen, perspektivischen Verzerrungen und fehlender Textur im Modell die Güte dieses Kriterium stark nachlässt. Deshalb bedient man sich weiteren Bedingungen, die an Korrespondenzpunkte gestellt werden können:

1. **Die Epipolar-Bedingung.** Für alle Punkte P eines Realmodells kann eine Epipolar-Ebene gefunden werden, die durch P und den Kamerapositionen des *key*- und *offset*-Bildes bestimmt ist. Wie in Abbildung 15 zu erkennen ist, ergeben sich die Epipolar-Linien dann als Schnitt zwischen der Epipolar-Ebene und dem *key*- bzw. *offset*-Bild.

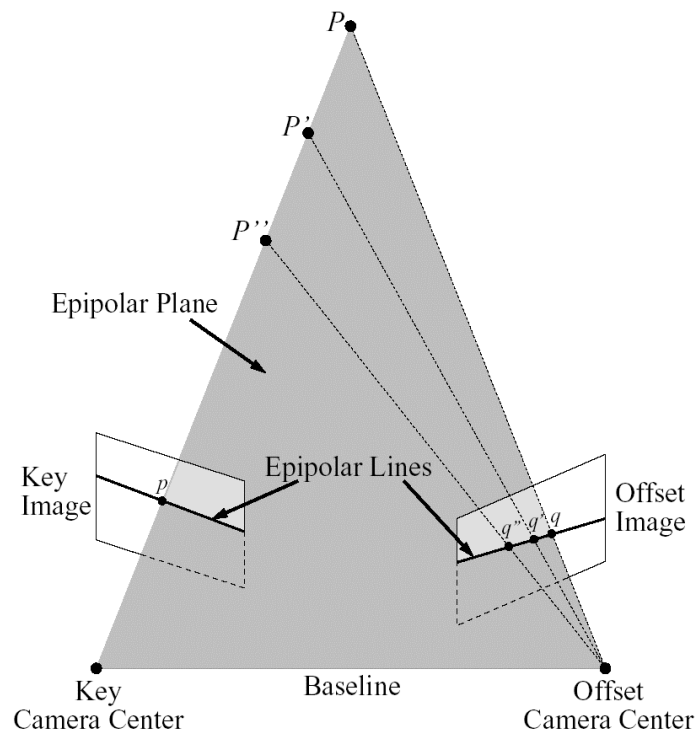


Abbildung 15: Die Epipolar-Geometrie von Korrespondenzpunkten

Wegen der unbekanntem Tiefe des Punktes P kann nicht eindeutig ermittelt werden, welchem Punkt der projizierte Punkt p entspricht: Mögliche Punkte wären bspw. P , P' oder P'' . Abhängig davon ergeben sich als potentielle Korrespondenzpunkte q , q' oder q'' . Allgemein kommen also nur solche Punkte in Frage, die auf der korrespondierenden Epipolar-Linie liegen. Somit verringert sich die Suche von zwei auf eine Dimension. Für den modell-basierten Stereo-Algorithmus kann diese Bedingung sehr vorteilhaft eingesetzt werden.

2. **Einschränkungen bzgl. der Reihenfolge.** Wie im vorherigen Punkt beschrieben, liegen die Korrespondenzpunkte auf den zugehörigen Epipolar-Linien. Ergeben die Tiefenwerte nebeneinander liegender Punkte eine monotone Funktion, so ist die Reihenfolge, in der die Punkte auf den Epipolar-Linien liegen, gleich. Daraus ergibt sich für einen Algorithmus eine weitere Effizienzsteigerung hinsichtlich der zu untersuchenden Punkte. Leider gilt diese Einschränkung nicht allgemein, bspw. nicht für ein ausreichend schmales Objekt, das vor einem großflächigen Objekt im Hintergrund steht. Mit Hilfe des Grobmodells, das hier als Grundlage für den Stereo-Algorithmus verwendet wird, ist es möglich, die Szene in Flächen zu unterteilen, die die Bedin-

gung bzgl. der Reihenfolge normalerweise erfüllen.

3. **Teilweise Kontinuität der Tiefen.** Betrachtet man alle Tiefen einer Szene, so ergeben sich Bereiche, in denen Tiefenänderungen kontinuierlich verlaufen, getrennt von Bereichen mit diskontinuierlichen Verläufen. Diese Beobachtung kann für die Bestimmung der Korrespondenzpunkte dahingegen ausgenutzt werden, dass benachbarte Punkte ähnliche Tiefenwerte aufweisen sollten. Ansonsten ist es ein gutes Zeichen dafür, dass sie zu unterschiedlichen Objekten gehören. Da große relative Tiefenänderungen hauptsächlich an Objektgrenzen auftreten. Besonders zutreffend ist dieses bei architektonischen Gebäuden, so dass mit dieser Bedingung gute Resultate erzielt werden können.

Für einen traditionellen Stereo-Algorithmus ist es relativ schwierig die Korrespondenzpunkte zwischen den vorliegenden Photographien zu finden, die in der Regel aus sehr unterschiedlichen Richtungen hinsichtlich der entstehenden Tiefen aufgenommen wurden. Gründe hierfür sind:

- Zwischen dem *key*- und *offset*-Bild können große Abweichungen auftreten, so dass Überschneidungen nur schwer gefunden werden und der Algorithmus viele potentielle Korrespondenzpunkte untersuchen muss.
- Dadurch, dass viele Photographien aus unterschiedlichen Richtungen aufgenommen wurden, ergeben sich perspektivische Verzerrungen, wodurch gleiche Objekte und Nachbarschaftsstrukturen unterschiedlich skaliert sind.
- Das *key*- und das *offset*-Bild enthalten oft Objekte, die nur auf einem der beiden Bilder verdeckt sind und für die keine Korrespondenzpunkte existieren.

Deshalb verwendet der hier vorgestellte Stereo-Algorithmus das aus Kapitel 2 erstellte Grobmodell und vergleicht potentielle Korrespondenzpunkte nicht direkt miteinander, sondern erstellt zuvor aus dem *offset*-Bild ein so genanntes „*warped offset*“-Bild, das anschließend mit dem *key*-Bild verglichen wird. Abbildung 16(a) und (c) zeigen ein *key*- und ein *offset*-Bild als flache Oberfläche aus der ursprünglichen Kameraposition. Das in (b) dargestellte *warped offset*-Bild wurde erzeugt, indem das *offset*-Bild als Textur für das Grobmodell diente und aus Sicht der Kameraposition des *key*-Bildes neu erstellt wurde. Die Bilder (a) und (b) erscheinen dadurch

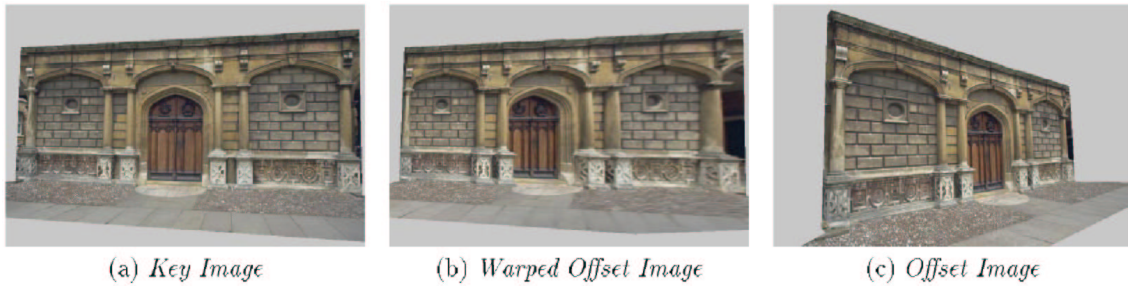


Abbildung 16: Key-, Warped Offset- und Offset-Bild

viel ähnlicher als (a) und (c). Die vorhandenen Unterschiede resultieren jetzt nur noch durch die fehlenden geometrischen Details im Grobmodell. Das *warped offset*-Bild hat folgende vorteilhafte Eigenschaften:

- Bereiche, die sich in der Szene auf dem Grobmodell befinden, verursachen keine Abweichung zwischen *key*- und *warped offset*-Bild.
- Abweichungen resultieren also aus den Unterschieden zwischen Modell und Szene.
- Perspektivische Verzerrungen bezogen auf das *key*-Bild können vermieden werden.
- Alle Objektverdeckungen können durch das Grobmodell identifiziert werden.
- Die Epipolar-Linien stimmen mit denen des *offset*-Bildes überein.

4.1 Bestimmung der Korrespondenzpunkte

Nachdem das *warped offset*-Bild erstellt wurde, versucht der Stereo-Algorithmus durch Untersuchung der Nachbarschaftsumgebungen die Korrespondenzpunkte zu finden. Traditionelle Stereo-Algorithmen benutzen dabei die im vorherigen Abschnitt beschriebene Tatsache, dass Korrespondenzpunkte, wie in Abbildung 15 zu sehen ist, in derselben Epipolar-Ebene liegen, so dass zu einem bestimmten Punkt nur die korrespondierende Epipolar-Linie betrachtet werden muss. Dadurch, dass das *offset*-Bild zuerst auf das Grobmodell projiziert und anschließend aus Sicht des *key*-Bildes neu dargestellt wurde, muss analysiert werden, in welcher Weise sich die Epipolar-Geometrie der Bilder verändert hat. Mathematisch kann bewiesen werden, dass die Epipolar-

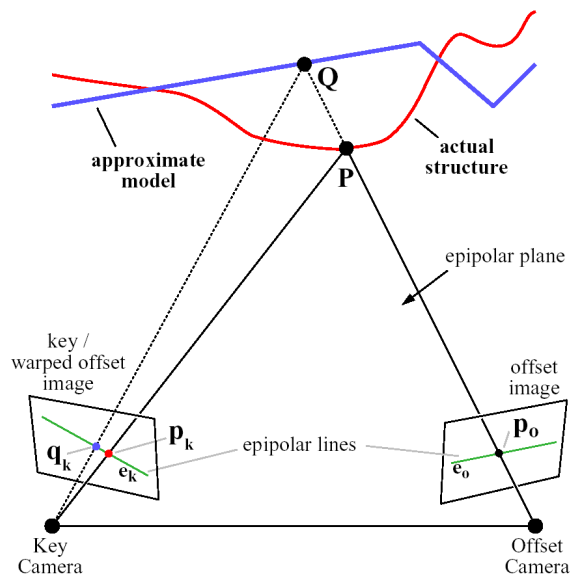


Abbildung 17: Epipolar-Geometrie bei Verwendung des modell-basierten Stereo-Algorithmus

Geometrie, unabhängig von der Form der Objekte, linear bleibt und dass die Epipolar-Linien identisch mit denen des *key*-Bildes sind.

Abbildung 17 zeigt die Epipolar-Geometrie bei Verwendung des modell-basierten Stereo-Algorithmus. Sei P ein Punkt auf einem realen Objekt, der im *offset*-Bild dem Punkt p_{offset} entspricht. Da das *key*-Bild ebenfalls das reale Objekt darstellt, entspricht hier dem Punkt P der Punkt p_{key} . Alle drei Punkte liegen in derselben Epipolar-Ebene und die Punkte p_{offset} und p_{key} auf den dadurch definierten Epipolar-Linien. Bei der Erzeugung des *warped offset*-Bildes wird p_{offset} auf das Grobmodell projiziert und entspricht jetzt nicht dem Punkt P , sondern Q . Da Q aber ebenfalls in derselben Epipolar-Ebene liegt, wird er in dem *warped offset*-Bild durch den Punkt q_{key} repräsentiert, der aber folglich auf der Epipolar-Linie liegt, die auch p_{key} schneidet.

Diese Tatsache kann jetzt vom Stereo-Algorithmus ausgenutzt werden, indem das *key*-Bild in viele Epipolar-Linien aufgeteilt und unabhängig versucht wird, Korrespondenzpunkte zu identifizieren. Gleichzeitig können die Kriterien aus dem vorherigen Abschnitt verwendet werden, um Abweichungen zwischen korrespondierenden Nachbarschaftsgebieten abzuschätzen.

Die daraus resultierenden Tiefenangaben können schließlich dazu verwendet werden, das Grobmodell mit entsprechender Textur detailgetreu darzustellen, wobei auch andere Ansichten des Modells möglich werden, als nur die durch die Photographien vorgegebenen. Liegen viele solcher Photographien mit den dazugehörigen Tiefeninformationen vor, so kann die in Kapitel 3 entwickelte Gewichtsfunktion der blickwinkelabhängigen Texturerzeugung verwendet werden, um mehrere solcher Bilder zu kombinieren.

Kapitel 5

Zusammenfassung

Die in dieser Seminararbeit vorgestellten Techniken beschreiben einen Ansatz, der es ermöglicht, aus wenigen Photographien architektonischer Gebäude ein realistisches und akkurates räumliches Modell zu erstellen. Verwendet wurde dazu ein hybrider Ansatz, der geometrie- und bild-basierten Methoden kombiniert. Mit dem Modellierungsprogramm *Façade* wurde es auf einfache Weise möglich, ein vorläufiges Grobmodell der Szene zu erzeugen, das als Basis für eine spätere Verfeinerung diente. Der Benutzer brauchte lediglich in einem inkrementellen Prozess Photographien bearbeiten, indem er durch Anordnung primitiver geometrischer Objekte das Grobmodell erzeugt und markierte Bildkanten mit den entsprechenden Modellkanten verknüpft. Anschließend konnten durch Minimierung einer Zielfunktion, die den aufsummierten Abweichungen zwischen Bild- und Modellkante entsprach, alle Modellparameter und Kamerapositionen der aufgenommenen Photographien bestimmt werden. Hierfür war es notwendig, geeignete Startwerte zu bestimmen, so dass der Minimierungsalgorithmus gegen das globale Optimum konvergiert. Als ein Zwischenresultat der Seminararbeit wurde die blickwinkelabhängige Texturerzeugung vorgestellt, wodurch sich verbesserte und realistischere Ansichten des Modells ergeben. Die fehlenden Details, die durch das Grobmodell noch nicht erfasst wurden, konnten schließlich am Ende mit Hilfe eines modell-basierten Stereo-Algorithmus modelliert werden. Hierzu mussten ausgehend von zwei Photographien alle Korrespondenzpunkte gefunden werden, woraus für alle Punkte Tiefenangaben abgeleitet werden konnten. Da die vorliegenden Bilder nur architektonische Gebäude zeigen, konnten mehrere Bedingungen an die Korrespondenzpunkte gestellt werden, wodurch der entstehende Suchraum entscheidend eingeschränkt werden konnte. Vor allem konnte das zuvor konstruierte Grobmodell dazu verwendet werden, einzelne

Fotographien nicht direkt miteinander zu vergleichen, sondern vorher eines von beiden auf das Grobmodell zu projizieren und aus Sicht des zweiten Bildes darzustellen. Die Suche nach den Korrespondenzpunkten wurde dann zwischen diesem und dem zweiten Bild durchgeführt. Ein besonders wichtiges Ergebnis dabei war, dass die Epipolar-Geometrie durch diese Transformation weiterhin linear blieb und dass die Epipolar-Linien aufeinander fielen. Dadurch konnten die zuvor aufgestellten Bedingungen erhalten werden. Zuletzt wurde erläutert, wie die Methodik der blickwinkelabhängigen Texturerzeugung auf das detaillierte Modell übertragen werden kann, um daraus neue Ansichten des Modells zu generieren.

Literaturverzeichnis

- [1] Paul Ernest Debevec. **Modeling and Rendering Architecture from Photographs**. PhD thesis, University of California at Berkeley, 1996.