Umgebungsmodellierung auf der Basis von Stereo-Kamerabildern für eine Telepräsenzanwendung

Georg Passig

Dissertation

Lehrstuhl für Realzeit-Computersysteme

Umgebungsmodellierung auf der Basis von Stereo-Kamerabildern für eine Telepräsenzanwendung

Georg Passig

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. habil. G. Rigoll
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. G. Färber
	2. Univ.-Prof. Dr.-Ing. K. Diepold

Die Dissertation wurde am 13.09.2005 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 29.03.2006 angenommen.

Danksagung

Diese Dissertation entstand als Ergebnis meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Realzeit–Computersysteme der Technischen Universität München. Teile der Arbeit wurden von der *Deutschen Forschungsgemeinschaft* (DFG) als Teil des Sonderforschungsbereichs „Wirklichkeitsnahe Telepräsenz und Teleaktion“ gefördert.

Mein besonderer Dank gilt Herrn Prof. Färber für das in mich gesetzte Vertrauen und die Betreuung dieser Promotion. Die Arbeit am Lehrstuhl habe ich als die richtige Mischung aus Projektarbeit, Lehre und Freiräumen empfunden, die es mir ermöglicht hat, mich weiterzuentwickeln und zu entfalten. Mein Dank gilt auch Herrn Professor Diepold für sein Interesse an meiner Arbeit und die bereitwillige Übernahme des Koreferates.

Das jederzeit gute Arbeitsklima am Lehrstuhl war für mich ein wichtiger Grund am LPR anzufangen und mich auch am RCS weiterhin wohlzufühlen. Dafür bedanke ich mich bei meinen früheren und heutigen Kollegen, insbesondere der *Robot-Vision*-Gruppe des Lehrstuhls.

Für die gute Zusammenarbeit bedanke ich mich speziell bei Tim Burkert, der die SFB-Jahre mit mir geteilt hat und dessen Arbeit der Dreh- und Angelpunkt des Projekts war. Bei meinem zweiten SFB-Kollegen Jan Leupold bedanke ich mich ebenfalls für die gute Zusammenarbeit und die viele Hilfe in Softwarefragen. Der wertvolle Beitrag, den er mit seinen Grundlagenbibliotheken zu unserer Arbeit geleistet hat, ist hier hervorzuheben. Ich danke meinen Diplomanden und Werkstudenten für ihre Hilfe und die vielen hilfreichen Diskussionen.

Bei meinen beiden Korrekturlesern Alexa Zierl und Kathrin Passig bedanke ich mich für ihre bereitwillige Suche nach fachlichen und sprachlichen Ungereimtheiten. Meinen Eltern möchte ich danken, dass sie mir diese Ausbildung ermöglicht haben – und mich nicht wegen Latein von der Schule genommen haben.

Die vielfältigen neuen Aufgaben und Herausforderungen, die eine junge Familie mit sich bringt, relativieren die Probleme einer Dissertation. Dieser Rahmen hat mir während meiner Arbeit sehr geholfen. Vielen Dank Konstantin und Emma. Die Kontinuität meiner Arbeit beruht auf dem Gefühl großer Zufriedenheit mit meinem Leben. Vielen Dank Nathalie.

München, im September 2005

Kurzfassung

Die Rekonstruktion dreidimensionaler Szenen aus Sensordaten spielt eine wichtige Rolle in vielen Anwendungen von der mobilen Robotik bis zur Fernerkundung. In der vorliegenden Arbeit wird die 3D-Rekonstruktion im Rahmen einer Telepräsenzanwendung eingesetzt, in der ein Bediener einen Roboter intuitiv fernsteuern soll. Um die störende Zeitverzögerung, die durch die Datenübertragung in diesem System entsteht, zu kaschieren, werden synthetische Bilder von der Umgebung des Roboters prädiziert und fotorealistisch dargestellt. So nimmt der Bediener die Folgen seiner Steuerbefehle sofort wahr, obwohl der Roboter sie erst mit einer Verzögerung tatsächlich ausführt. Dafür muß ein Umgebungsmodell aus Stereo-Kamerabildern gewonnen und aktualisiert werden, auf dessen Basis die synthetischen Bilder erzeugt werden können. Diese Arbeit beschreibt die Modellbildung aus Kamerabildern und ihre Einbettung in den Telepräsenzkontext.

Die vorliegende Arbeit hat ihre Schwerpunkte in der Gewinnung von Entfernungsinformation aus Stereo-Kamerabildern und dem Aufbau eines polygonalen dreidimensionalen Modells einer Szene aus diesen Daten. Zunächst werden drei ausgewählte Stereo-Verfahren detailliert vorgestellt und anhand von Testdaten verglichen, um ihre Leistung im angestrebten Einsatzgebiet einschätzen zu können. In der zweiten Hälfte der Arbeit steht der Aufbau polygonaler Netze aus Tiefenkarten, ihre Weiterverarbeitung und die Kombination verschiedener Ansichten in ein gemeinsames Modell im Zentrum des Interesses. Daneben wird die Einbindung der Modellgewinnung in das oben beschriebene Telepräsenzprojekt beschrieben. Die Funktion des Systems wird anhand von Testszenarien unterschiedlicher Komplexität belegt. Die Qualität der resultierenden Modelle wird nach den Gesichtspunkten Bildqualität, Aktualisierungsgeschwindigkeit und Darstellungsgeschwindigkeit beurteilt.

Das Ergebnis ist ein Komplettsystem, mit dem aus Stereo-Kamerabildern und ihrer jeweiligen Aufnahmeposition ein polygonales Modell der Szene erstellt werden kann. Das System ist speziell auf die Anforderungen im Telepräsenzkontext zugeschnitten. Das heißt, dass mit jedem neu eintreffenden Bildpaar das polygonale Modell nur soweit nötig aktualisiert wird.

Das System ist in der Lage, ein komplexes Szenenmodell aus einfachen Stereo-Kamerabildern innerhalb kurzer Zeit zu erstellen. Auf der Basis dieses Modells ist für den telepräsenten Bediener des Roboters somit immersives Arbeiten innerhalb der synthetisch erzeugten, aber fotorealistisch dargestellten Szene möglich.

Inhaltsverzeichnis

Verzeichnis der verwendeten Symbole

RCS	Lehrstuhl für Realzeit–Computersysteme
b	Bildbreite
h	Bildhöhe
d	Disparität
$\mathcal{F}_W$	Weltkoordinaten
$\mathcal{F}_K$	Kamerakoordinaten
$\mathcal{F}_B$	Bildkoordinaten der Kamera
$\mathcal{F}_P$	Technische Pixelkoordinaten der Kamera
$\mathcal{F}_{K_0}$	Kamerakoordinaten der linken Kamera in einem Stereosystem
$\mathcal{F}_{k_0}$	Kamerakoordinaten der rektifizierten linken Kamera in einem Stereosystem
$\mathcal{F}_b$	Bildkoordinaten der rektifizierten Kamera
$\mathcal{F}_p$	Technische Pixelkoordinaten der rektifizierten Kamera
${}^K\mathbf{T}_W$	Transformationsmatrix vom System $\mathcal{F}_W$ in das System $\mathcal{F}_K$ in homogenen Koordinaten
${}^{p_0}\mathbf{P}_W$	Projektionsmatrix von Weltkoordinaten in rektifizierte technische Pixelkoordinaten der linken Kamera in einem Stereosystem in homogenen Koordinaten
I_{P1}	Kamerabild am Ort P1 aufgenommen
D_{P1}	Disparitätenkarte von I_{P1}
D'_{P1}	Aus dem Modell erzeugte virtuelle Disparitätenkarte

1 Einleitung

1.1 Motivation für diese Arbeit

Die Kameratechnik blickt mittlerweile auf eine mehr als hundertjährige Geschichte zurück. Dabei war der Wunsch, bewegte Bilder aufzuzeichnen, zu transportieren, zu archivieren und darzustellen, die treibende Kraft. Die Digitalisierung hat dazu geführt, dass diese Technik heute so günstig und einfach einzusetzen ist, dass Kameras zu einem allgegenwärtigen Gebrauchsgegenstand in unterschiedlichsten Anwendungen geworden sind. Parallel dazu hat sich die inhaltliche Analyse und Verarbeitung von Bildern zu einem Forschungsgebiet mit unzähligen Anwendungen und einem lukrativen Markt entwickelt. Da digitale Bilder, insbesondere bewegte, ein enormes Datenaufkommen mit sich bringen, war die Bildverarbeitung zunächst durch die begrenzte Rechenleistung und Bandbreite verfügbarer Rechnersysteme eingeschränkt. Obwohl dies durch rasant steigende Bildauflösung und Farbtiefe gängiger Kameras immer noch noch gilt, ermöglichen die Fortschritte der Computertechnik heute immer komplexere Anwendungen der Bildverarbeitung.

Im Zentrum dieser Arbeit steht ein solcher Teilbereich der Bildverarbeitung, die Rekonstruktion von dreidimensionaler Information aus Stereo-Kamerabildern. Angewendet auf diesen Teilbereich gilt das oben Gesagte ebenfalls: Die Grundlagen der Stereosicht sind bereits seit langer Zeit bekannt. Sie wurden frühzeitig in der Fotografie und Fotogrammetrie zur Anwendung gebracht und haben über viele Jahre hinweg unzählige Forschungsarbeiten in verschiedenen Gebieten motiviert. Trotzdem ist erst in jüngerer Zeit das technische Potential verfügbar, viele der durch hohes Datenaufkommen und große Komplexität geprägten Ansätze zu implementieren und weiterzuentwickeln.

Gleichzeitig erzeugt der ebenfalls von den Fortschritten der Computertechnik beflügelte Bereich der Computergrafik heute neue Impulse für die Rekonstruktion dreidimensionaler Szenen. Die primär vom Markt der Computerspiele vorangetriebene Entwicklung, dreidimensionale Szenen in fotorealistischer Qualität auf handelsüblichen Computern darzustellen, erzeugt einen großen Bedarf für Modelldaten *realer* Szenen für neue Anwendungen. Diese Modelle wurden bis vor kurzem noch größtenteils in Handarbeit erzeugt. Heute wird diese Lücke häufig von Systemen geschlossen, die die Umgebung oder einzelne Objekte mittels Laserlicht abtasten und modellieren. Doch entsteht dank intensiver Forschungsarbeit und leistungsfähiger Computer ein wachsender Forschungsbereich, der Kamerasysteme zu diesem Zweck einsetzt.

Aus diesen Entwicklungstendenzen entstand die Idee zu einem Teilprojekt im Sonderforschungsbereich „Wirklichkeitsnahe Telepräsenz und Teleaktion“. Im Projekt „Übertragungszeitkompensation durch Szenenprädiktion“, das am Lehrstuhl für Realzeit-Computersysteme bearbeitet wurde, steht genau diese Schnittmenge von Bildverarbeitung, Geometrierekonstruktion und Computergrafik im Zentrum des Interesses.

1.2 Kontext und Aufgabenstellung

Telepräsenz wird erreicht, wenn es einem menschlichen Operator durch technische Mittel ermöglicht wird, mit seinem subjektiven Empfinden in einer anderen, entfernten oder nicht zugänglichen Umgebung präsent zu sein. **Teleaktion** bedeutet, dass der menschliche Operator nicht nur passiv präsent ist, sondern dass er am entfernten Ort auch aktiv eingreifen kann [81]. Im Sonderforschungsbereich „Wirklichkeitsnahe Telepräsenz und Teleaktion“ werden seit dem Jahr 1999 verschiedenste Fragen aus dem Bereich der Telepräsenz untersucht.

In einem typischen Telepräsenzsystem werden (Hand- und Kopf-) Bewegungen eines Bedieners in Bewegungen einer entfernten Maschine umgesetzt. Rückmeldung der Aktionen der Maschine können durch Bilder, durch haptische, taktile und akustische Eindrücke erfolgen. Übertragungszeiten erweisen sich dabei als störend, wenn der Bediener in einer Telemanipulation Bestandteil eines Regelkreises wird. Im Extremfall wartet der Bediener nach jeder Aktion auf die Rückmeldungen der Maschine, womit das Präsenzempfinden und die Arbeitsgeschwindigkeit auf ein Minimum zurückgehen.

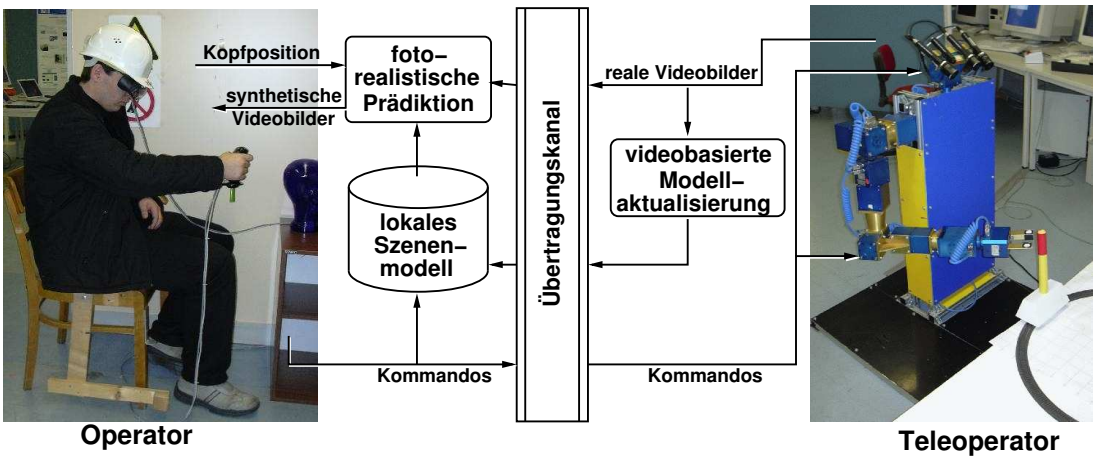


Abbildung 1.1: Übertragungszeitkompensation durch Szenenprädiktion – Übersicht.

Das Teilprojekt „Übertragungszeitreduktion durch fotorealistische Szenenprädiktion“ nutzt mit der Prädiktion eine bekannte Strategie, diese Übertragungszeit im visuellen Kanal zu kaschieren. Abbildung 1.1 stellt die Komponenten des Systems dar: der Bediener des Roboters kann diesen durch intuitive Bewegungen seiner Hand oder seines Kopfes

steuern. Doch anstatt Kamerabilder als Rückmeldung präsentiert zu bekommen, werden in einem Head-Mounted-Display synthetische Ansichten der Szene dargestellt. Dabei kommt dem Projekt die rasante Entwicklung im Bereich der Grafikkarten in Standardcomputern entgegen. Diese Subsysteme bieten die hohe Grafikleistung die zur schnellen Darstellung texturierter Polygone in dreidimensionalen Szenen *(rendering)* notwendig ist und machen sie durch eine standardisierte Schnittstelle (OpenGL) nutzbar.

Die Darstellung synthetischer Ansichten der Szene ermöglicht die Prädiktion der Bilder und damit ein Kaschieren eventuell vorhandener Übertragungszeiten. Um jedoch die Szene korrekt und möglichst fotorealistisch darstellen zu können, muss umfangreiches Wissen über die Szene vorliegen. Bezogen auf den in Abbildung 1.1 dargestellten Fall wären das folgende Komponenten:

- ein geometrisches Modell des Roboters
- die Gelenkwinkel zum Zeitpunkt der Bildaufnahme
- die Position der Kameras zum Zeitpunkt der Bildaufnahme
- die Lage des rot-gelben Zylinders
- den sonstigen Inhalt des Bildes (Hintergrund) als geometrisches Modell
- das Aussehen jedes Objekts in der Szene (Textur)

Die vorliegende Arbeit widmet sich speziell der Rekonstruktion des zunächst unbekannten Hintergrunds durch optische Methoden. Das Ziel einer *subjektiv überzeugenden Darstellung* für das vorgestellte Projekt wirkte dabei prägend auf die eingesetzten Verfahren und ihre Bewertung.

Szenario

Die Charakteristik der Szene, also die Eigenschaften des soeben vorgestellten Manipulationsszenarios sind ausschlaggebend für die Auswahl von Verfahren und ihre Implementierung. Daher soll zu diesem frühen Zeitpunkt dieses gedankliche Szenario etwas allgemeiner zusammengefasst werden.

- Die Kamera des Roboters ist beweglich und wird den Kopfbewegungen des Bedieners nachgeführt.
- Der Roboterarm ist für den Bediener bei entsprechender Kameraposition sichtbar, und seine Geometrie und Lage ist aus der Robotersteuerung bekannt.
- Der Bediener interagiert mit einer begrenzten Anzahl von Objekten, die im Zentrum seiner Aufmerksamkeit stehen.
- Die restliche Szene ist zunächst unbekannt und wird als weitgehend statisch angenommen.

- Es handelt sich um ein Szenario in Innenräumen, das heißt, es dominieren ebene Flächen, rechte Winkel und texturarme Regionen.
- Ziel ist der Aufbau und die schnelle Aktualisierung des Szenenmodells.

Grundlegende Entscheidungen

Ihre größte Leistung entfalten Grafikkarten aktueller Computer bei stabilen, polygonalen Netzen mit Texturen, da diese Datenstrukturen derzeit die größte Bedeutung im Markt besitzen. Daher steht diese Datenstruktur als Ergebnis des Modellbildungsprozesses fest.

Die Szene zerfällt in zwei Teile:

- **Der Vordergrund** befindet sich im Zentrum der Aufmerksamkeit des Bedieners. Objekte, mit denen interagiert wird, sind bekannt und als geometrische Modelle verfügbar. Das erlaubt hohe Präzision und vereinfacht die Modellbildung auf Objektlokalisation und Verfolgung. Das Modell des Vordergrunds muss schritthaltend aktualisiert werden, da Verzögerungen hier sehr störend wirken. Der Roboterarm zählt ebenfalls zu diesem Vordergrund, da sein Modell und seine Position durch die Robotersteuerung automatisch bekannt sind.
- **Der Hintergrund** ist für die wirklichkeitsnahe Darstellung von großer Wichtigkeit. Er unterstützt das Präsenzempfinden des Benutzers und erlaubt dadurch erst die intuitive Interaktion mit dem System. Da er als weitgehend statisch angenommen wird, muss die Aktualisierung nicht schritthaltend erfolgen. Trotzdem ist eine schnelle Aktualisierung und insbesondere eine Erweiterung des Modells wichtig, ganz besonders, wenn von einem mobilen Telemanipulator ausgegangen wird.

1.3 Ziele und Gliederung dieser Arbeit

Ziele

Diese Arbeit stellt ein System vor, mit dem basierend auf Stereo-Kamerabildern das polygonale, geometrische Modell einer Szene aufgebaut, erweitert und verändert werden kann. Viele Design-Entscheidungen in diesem System sind im speziellen Kontext der angestrebten Telepräsenzanwendung zu sehen. Die Modellierung des Szenenhintergrundes steht im Zentrum der Arbeit, es wird jedoch auch die Aktualisierung des Vordergrundes kurz besprochen.

Aufgrund der fächerübergreifenden Problemstellung fußt diese Arbeit auf zahlreichen Vorarbeiten verschiedener Fachgebiete. Daher liegt ihr Schwerpunkt bewusst nicht im Entwurf eines neuen Verfahrens für ein Teilproblem, sondern in der sorgfältigen Auswahl und – wo nötig – Variation existierender Verfahren. Die Arbeit hat weiterhin den Anspruch, die Teilsysteme zu einem abgeschlossenen und der Aufgabe angemessenen Gesamtsystem

zusammenzusetzen, woraus sich neue Fragestellungen an den Schnittstellen der Teilsysteme ergeben. So ist insbesondere der Schritt von den mit einem Stereo-Kamerasystem gewonnenen Entfernungsinformationen zu einer polygonalen Beschreibung ein neuralgischer Punkt im Gesamtsystem. Hier stellt die Arbeit eine neue Methode der Integration von Disparitätenkarten in polygonale Modelle vor. Im selben Kontext wird auch eine neue Datenstruktur zur parallelen Modellierung der Szene in der Form unstrukturierter 3D-Punktewolken und polygonaler Netze vorgestellt.

Der Erfolg des Gesamtsystems muss sich an verschiedenen Maßstäben messen lassen:

- Die **Modellierungsgeschwindigkeit** ist aufgrund der großen Komplexität des Rekonstruktionsproblems und des großen Datenaufkommens ein kritischer Faktor. Wie oben bereits angeführt, ist eine schnelle Modellierung des Hintergrunds notwendig. Sie muss jedoch aus zwei Gründen nicht schritthaltend erfolgen. Zum einen ist durch das Konzept der Szenenprädiktion die Darstellungsgeschwindigkeit von der Modellaktualisierung unabhängig, zum anderen werden durch die konzeptionelle Teilung der Szene in Vordergrund und Hintergrund kritische Bereiche des Modells (z. B. Roboter) schneller aktualisiert als der Hintergrund. Rechenzeiten von mehreren Minuten oder gar Stunden, wie sie in Anwendungen wie der präzisen Rekonstruktion einzelner Objekte durchaus vorkommen, sind im vorliegenden System dennoch nicht vertretbar.

- Die **Qualität der Modellierung** muss zur Erzeugung fotorealistischer Ansichten geeignet sein. Zur Beurteilung dieser Qualität kann jedoch kein objektives Qualitätsmaß herangezogen werden, vielmehr steht das Präsenzempfinden des Bedieners und somit der Mensch im Mittelpunkt. Dabei ist die subjektive Bildqualität, wie sie der Leser in dieser Arbeit leicht nachvollziehen kann, nicht alleine ausschlaggebend. Modellfehler werden durch die Texturierung häufig gut kaschiert, fallen aber durch Parallaxenfehler bei translatorischer Bewegung des Betrachters in der Szene oder der Betrachtung des Modells mit Stereosicht erst auf. So kann die in dieser Arbeit erreichte Modellierungsqualität ohne wahrnehmungspsychologische Experimente mit Versuchspersonen nur oberflächlich beurteilt werden.

- Die **Darstellungsgeschwindigkeit** stellt eine kritische Größe dar, da sie direkt in die wahrnehmbare Reaktivität des Systems eingeht. Das heißt, dass der Bediener Verzögerungen zwischen seinen Aktionen (z. B. Kopfbewegung) und der Reaktion des System (z. B. Bilder) wahrnimmt. Die Darstellungsgeschwindigkeit hängt vorwiegend von der Anzahl darzustellender Polygone ab. Somit lautet das Ziel der Modellierung die Szene mit möglichst wenig Polygonen zu modellieren.

Gliederung

Nach dieser Einführung in das Thema und der Definition des Szenarios stellt Kapitel 2 einige verwandte Arbeiten im Bereich der Verknüpfung von Modellaufbau und -darstellung vor. Hier wurde bewusst diese applikationsnahe Ebene gewählt, da die beiden folgenden

Kapitel jeweils eines der Teilprobleme dieser Arbeit umfassend behandeln. So werden im Kapitel 3 „3D-Rekonstruktion aus Stereobildern“ zunächst die Grundlagen der Bildaufnahme (Abschnitt 3.1), die Grundlagen der Stereopsis (Abschnitt 3.2) und Verfahren der 3D-Rekonstruktion aus Kamerabildern (Abschnitt 3.3) vorgestellt.

Im Abschnitt 3.4 „Auswahl eines geeigneten Stereoverfahrens“ werden drei relevante Verfahren anhand verschiedener Testaufnahmen verglichen und parametriert. In diesem Kapitel werden ebenfalls viele Details der Stereorekonstruktion besprochen, mit denen sich die Qualität der Tiefenkarten verbessern lässt. Das Kapitel schließt mit einer Diskussion der Ergebnisse und einem Abschnitt zum Thema Implementierung.

Das Kapitel 4 „Modellaufbau“ bildet den zweiten Schwerpunkt dieser Arbeit. Nach der Darstellung der Problematik des Modellaufbaus beginnt dieses Kapitel mit den Grundlagen der Thematik und einer Klärung der Terminologie. Im Abschnitt 4.3 stehen die oben angedeuteten Fragestellungen am Übergang von einer Tiefenkarte zu einem Dreiecknetz im Vordergrund. Die logisch nächsten Schritte des Modellaufbaus bilden die folgende Gliederung in „Dezimierung“ und „Netzfusion“. Im Abschnitt 4.6 wird als Abschluss des Kapitels eine neuartige Datenstruktur vorgestellt, mit der die Problematik noch besser gelöst werden kann.

Das Kapitel 5 stellt die weiteren Module und Subsysteme, die im Telepräsenzsystem des Teilprojektes implementiert wurden, kurz vor. Damit soll die Arbeit im softwaretechnischen Projektrahmen dargestellt werden.

Die mit dem vorgestellten System erzielbaren Ergebnisse werden für unterschiedliche Szenen schließlich im Kapitel 6 dargestellt. Es zeigt im Abschnitt 6.4 auch die Anwendbarkeit in einem vollkommen unterschiedlichen Szenario. Eine Bewertung der Ergebnisse schließt das Kapitel ab.

Eine Zusammenfassung beschließt die Arbeit und stellt mögliche Richtungen für eine Weiterentwicklung des Systems vor.

2 Verwandte und ergänzende Arbeiten

In diesem Kapitel werden einige verwandte Arbeiten aus dem Umfeld dieser Arbeit vorgestellt. Dabei stehen Systeme im Vordergrund, bei denen eine ähnlich integrierte Kombination von Modellakquisition und Darstellung auftritt wie in dem Telepräsenzsystem, in dessen Kontext diese Arbeit steht. Zum Stand der Forschung in einzelnen Teilbereichen wird erst in den jeweiligen Grundlagenkapiteln des Teilbereichs Stellung genommen.

Die im folgenden vorgestellten Systeme sind nach Applikationen gegliedert, da die Randbedingungen der Applikation formend auf das System wirken und so häufig ähnliche Lösungen für ähnliche Probleme entstehen. Dabei wird die jeweilige Arbeit kurz in Beziehung zum vorliegenden System gesetzt und darauf eingegangen, in welchen Bereichen Ähnlichkeiten und Unterschiede bestehen.

Gegen Ende des Kapitels wird die Arbeit zur fotorealistischen Szenenprädiktion von Tim Burkert [7] kurz zusammengefasst, da sie die vorliegende Arbeit erst zu einem Gesamtsystem aus Modellaufbau und -darstellung ergänzt. Diese Arbeit ist im selben Teilprojekt des Sonderforschungsbereichs entstanden und beschreibt die fotorealistische Visualisierung der Modelldaten.

2.1 Szenenrekonstruktion und -Darstellung

2.1.1 Video-Konferenzen und Teleimmersion

Eine sehr natürliche Form der Telepräsenz ergibt sich in Videokonferenzanwendungen. Zwei oder mehrere Teilnehmer befinden sich an weit voneinander entfernten Orten, sollen sich aber in der Konferenzsituation dessen möglichst nicht bewusst werden. Aktuelle Systeme beschränken sich dabei auf die Bild- und Sprachkommunikation. Um gesteigerten Anforderungen an die Immersionsleistung zu begegnen, werden von verschiedenen Gruppen 3D-Videokonferenzlösungen entwickelt. So wird bei Mulligan [66] ein Satz von sieben auf einem horizontalen Kreissegment angeordneten Kameras zur Rekonstruktion genutzt. Dabei wird bi- und trinokulares korrelationsbasiertes Stereo auf rektifizierte Bilder angewandt. Die erforderliche Rechenleistung wird durch ein massiv-paralleles System aus Standard PCs erbracht. Die immersive Darstellung der Szene erfolgt jedoch auf der Basis von Punktewolken. Es wird also kein stabiles polygonales Szenenmodell aufgebaut und gepflegt.

In ähnlichem Kontext bewegen sich auch der **virtuelle Verkaufsraum** und die Arbeiten zur Teleimmersion an der ETH Zürich [31, 51]. Die zu erfassende Szene besteht hier aus einer einzelnen freistehenden Person. Rekonstruktion erfolgt silhouettenbasiert, mit Hilfe mehrerer, die Szene umgebender Kameras. Die Modellierung erfolgt hier ebenfalls nicht auf Polygonbasis, sondern mit Hilfe eines animierten generischen Personenmodells.

2.1.2 Robotik und Planung

Der Bereich der autonomen Robotik insbesondere mobiler Systeme gehört zu den Triebfedern der Entwicklung im Bereich der Stereorekonstruktion. Dabei werden die Daten jedoch in den meisten Fällen zur Selbstlokalisation des Roboters, zum Kartenaufbau oder zur Hindernisvermeidung genutzt. Damit gelten vollkommen andere Zielsetzungen für die Modellbildung. Zusätzlich wird der Darstellung der Modelldaten meist nur eine geringe Bedeutung beigemessen. Ein derartiges System wird zum Beispiel von **Burschka und Eberst** beschrieben [14, 8].

Deutlich näher verwandt mit der vorliegenden Arbeit ist das System von **Hirschmüller** [37]. Hier wird mit Hilfe eines frei beweglichen Stereo-Kamerasystems ein Szenenmodell aufgebaut, auf dessen Basis bereits während des Modellaufbaus virtuelle Ansichten der Szene erzeugt werden können. Die Applikation ist ebenfalls im Bereich der Telepräsenz angesiedelt. Obwohl das System viele Parallelen zur vorliegenden Arbeit aufweist, unterscheidet es sich gerade in der Art der Modellierung. Ein Ergebnis der Modellbildung sind bei Hirschmüller zweidimensionale Karten der Szene. Ein weiterer Schwerpunkt ist die Erzeugung virtueller Ansichten *direkt aus der Disparitätenkarte*. Das heißt, die in dieser Arbeit beschriebene Umwandlung der Daten in ein polygonales Netz spielt keine Rolle. Das System überzeugt jedoch durch hohe Geschwindigkeit.

Mit den Arbeiten von **Vitor Sequeira et al.** [80] sei eine weitere Arbeit im Grenzgebiet zwischen Robotik und virtueller Realität vorgestellt. Ein mobiler Roboter als Teleoperator erstellt auf der Basis eines Laserscanners und einer Kamera ein Modell eines unbekannten Raumes. Abgesehen vom Einsatz des Laserscanners zur Geometrierekonstruktion weist die Arbeit eine Fülle von Ähnlichkeiten zum vorliegenden System auf. Die 3D-Punkte des Laserscanners werden zunächst, soweit möglich, in ebene Oberflächen segmentiert. Innerhalb dieser Oberflächen können die 3D-Punkte somit von störendem Rauschen befreit werden. Nun wird auf die 3D-Punkte eine verfeinernde Triangulierung durch hierarchische Unterteilung in rechtwinklige Dreiecke angewendet. Die Modellinformation von benachbarten Aufnahmepositionen des Roboters wird verknüpft, indem die Einzelnetze jeweils in Paaren verglichen und vernäht werden. Obwohl die Arbeit große Ähnlichkeit mit dem vorliegenden System aufweist, unterscheidet sie sich doch in wesentlichen Komponenten. So wird in der vorliegenden Arbeit eine überlegene Methode zur Triangulierung verwendet, die mit weniger Dreiecken auskommt. Die von Sequeira vorgenommene Segmentierung der 3D-Daten in Ebenen ist jedoch hervorragend geeignet, die Dreiecksanzahl zur Verminderung des Rauschens zu reduzieren. Sie ist jedoch aufgrund des größeren Rauschens nur schlecht auf 3D-Daten auf Stereokamera-Basis anwendbar. Die in dieser Arbeit vor-

geschlagene 2D-Methode zum Vernähen von Dreiecken ist einfacher als die von Sequeira vorgestellte.

2.1.3 Szenenmodelle aus Videosequenzen

Die Extraktion der Szenengeometrie aus Videosequenzen stellt die logische Fortführung der Arbeiten zur Videokompression dar. Nach der einfachen Einzelbildkompression wurde das zweidimensionale Blockmatching zum Kompressionsstandard, da damit auch die zeitliche Redundanz der Bilder genutzt werden kann. Eine mögliche Fortsetzung dieser Methode besteht in der expliziten Verwendung der Szenengeometrie zur Kompression. So liegt in verschiedenen Arbeiten der Schwerpunkt nun auf der Segmentierung der Szene in einzelne Tiefenschichten, um diese dann weitgehend unabhängig voneinander komprimieren zu können. Die Arbeiten von **Steinbach et al.** [83] sind hierfür ein Beispiel. Aufgrund von Merkmalskorrespondenzen in konsekutiven Aufnahmen wird die Bewegung der Kamera durch *„structure from motion“*-Algorithmen bestimmt. Damit sind Struktur und Bewegungsmodelle einzelner Starrkörper in der Szene bekannt und können in Folgebildern zur Erweiterung des Szenenmodells genutzt werden. Auch in dieser Arbeit erfolgt keine polygonale Modellierung der resultierenden Szenengeometrie, sondern eine der Applikation angepasste Darstellung als segmentierte Punktewolken.

2.1.4 Virtuelle Realität

Marc Pollefeys hat mit seinem System [70] zur Szenenmodellierung auf der Basis unkalibrierter monokularer Kamerabilder in den vergangenen Jahren einen hohen Bekanntheitsgrad erreicht. Das eingesetzte Verfahren ähnelt dem im letzten Abschnitt vorgestellten, hat jedoch mit der Szenenrekonstruktion für die Anwendung in der virtuellen Realität eine unterschiedliche Zielsetzung. Auf der Grundlage vergleichsweise weniger Merkmalskorrespondenzen in den ersten beiden Bildern einer Sequenz wird eine projektive Rekonstruktion der Szene erzeugt. Auf der Basis dieser initialen Szenengeometrie werden im Anschluss die Kamerapositionen aller Aufnahmen bestimmt und das Szenenmodell verfeinert. Mit Hilfe weniger grundlegender Annahmen über die intrinsischen Parameter der Kamera kann nun der Schritt von einer projektiven zu einer metrischen Rekonstruktion erfolgen. Dies wiederum ermöglicht es, dichte Tiefenkarten für benachbarte Bildpaare zu berechnen. Die resultierende Tiefenkarte aus allen Aufnahmen in Form einer Punktewolke wird nun basierend auf einem *„thin-plate“*-Modell und Splines approximiert und in eine polygonale Beschreibung umgewandelt. Texturextraktion aus mehreren Bildern und darauffolgende Fusion ermöglicht nun die Darstellung durch Standard-Computergrafik. Das System ist im Prinzip zwar auf eine iterative Erweiterung durch neue Aufnahmen der Szene ausgelegt, dies ist jedoch gerade beim letzten Schritt, der Umwandlung in eine polygonale Beschreibung nicht berücksichtigt, da es nicht zu den Projektzielen gehört. Nichtsdestotrotz hat Pollefeys ein beeindruckendes System vorgestellt, das allerdings keine Ansprüche auf schnelle Modellierung erhebt.

Große Bekanntheit hat **Takeo Kanade** mit seinem *„virtualized Reality“*-Projekt erlangt. Hier wurde eine große Anzahl synchronisierter Kameras in Form einer Kugelschale um die Szene herum angeordnet, um durch eine Kombination von silhouettenbasierten Methoden und trinokularem Stereo, Modelle bewegter Szenen zu erstellen. Das Projekt erhebt dabei keinen Anspruch auf schritthaltende Verarbeitung. In einer aktuellen Weiterentwicklung (*EyeVision* [46]) wurde ein Sportereignis durch dreißig Kameras auf Schwenk-Neige-Plattformen aufgezeichnet. Nach kurzer Berechnungszeit konnten so virtuelle Kameraschwenks in dynamischen Szenen dargestellt werden. Doch basiert auch dieses System auf einer voxelbasierten Repräsentation der Szene, wodurch sich die Komplexität erheblich verringert. Nichtsdestotrotz ist allein die Modellerstellung aus dreißig Kameras eine beeindruckende Leistung.

2.2 Fotorealistische Szenenprädiktion

Die Arbeiten von Tim Burkert [7] stellen innerhalb des Projektes „Übertragungszeitkompensation durch Szenenprädiktion“ die Ergänzung der vorliegenden Arbeit dar. Ihr Schwerpunkt liegt in der fotorealistischen Darstellung der Szene. Wie im linken Bereich der Abbildung 1.1 dargestellt, wird dem Bediener eine aktuelle, aber prädizierte Ansicht der Szene basierend auf einem laufend aktualisierten lokalen Modell dargestellt. Er ist durch die sofortige visuelle Rückmeldung auf seine Bewegungen (Hand und Kopf) in der Lage, intuitiv und immersiv mit dem Roboter zu agieren, auch wenn – bedingt durch Übertragungszeiten – nur verzögerte Bilder als Rückmeldung eintreffen. Um bei komplett synthetisch erzeugten Bildern, wie sie hier verwendet werden, fotorealistische Qualität zu erreichen, werden Texturen für jedes Polygon aus den verzögert eintreffenden Kamerabildern erzeugt. Die dafür erforderlichen Berechnungen werden fast ausschließlich auf der Grafikkarte des PC-Systems ausgeführt. Damit kann die dort verfügbare hohe Leistung für Grafik-Operationen auch für die Texturerzeugung genutzt werden. Gleichzeitig ist kein aufwändiger Transport der speicherintensiven Texturen erforderlich. Bei der Texturextraktion werden Verdeckungen in der Szene berücksichtigt, die Texturen normalisiert, Texturen aus unterschiedlichen Kamerabildern fusioniert, Löcher in Texturen, soweit möglich, gefüllt und die Texturen schließlich zur Darstellung abgelegt. Gleichzeitig erfolgt die schritthaltende Darstellung der texturierten polygonalen Szene für den Operator.

Sämtliche texturierte polygonale Ansichten in Kapitel 4.4 und Kapitel 6 wurden mit Hilfe dieses Systems auf der Basis von polygonalen Netzen und dazugehörigen Kamerabildern (und ihrer Position) erstellt.

3 3D-Rekonstruktion aus Stereobildern

Dieses Kapitel bildet mit dem Thema Stereosicht einen der beiden Schwerpunkte dieser Arbeit. Zunächst werden die Grundlagen der Kameratechnik und Abbildungsgeometrie vorgestellt, um dann zu Stereo-Kamerasystemen und den damit verbundenen Fragestellungen der Korrespondenzfindung, Rektifizierung und Rekonstruktion vorzudringen. Um einen Überblick über die mannigfaltigen Algorithmen im Bereich der Tiefenrekonstruktion zu ermöglichen, werden im Abschnitt 3.3 die Komponenten erklärt, die den meisten Algorithmen zugrunde liegen. Drei Verfahren mit niedriger, mittlerer und hoher Komplexität werden im Anschluss mittels Testaufnahmen verglichen und ein Verfahren ausgewählt. Implementierungs- und Komplexitätsfragen schließen das Kapitel ab.

3.1 Grundlagen der Bildaufnahme

3.1.1 Technische Grundlagen

Funktionsprinzip

Videokameras bilden dreidimensionale Szenen durch Projektion auf eine lichtempfindliche Ebene ab und erlauben die Digitalisierung der Bildinformation. Dabei erfolgt eine räumliche, zeitliche und messtechnische Quantisierung. So führt die endliche Ausdehnung der lichtempfindlichen Elemente (Pixel) zu einer Mittelung der Helligkeitsinformation über die Fläche. Zu diskreten Zeitpunkten wird der (elektronische) Verschluss der Kamera geöffnet, und es erfolgt eine Messung der Helligkeit am Ort des Pixels. Dies geschieht je nach Sensortyp entweder durch Integration der Helligkeit über die Verschlusszeit oder durch Messung der Helligkeit zum Zeitpunkt der Abtastung. Dabei gelten spezifische obere und untere Helligkeitsgrenzen durch Sättigungseffekte bzw. Rauschen. Die gemessene Helligkeit wird analog-digital gewandelt und damit diskretisiert.

Bildgebende Sensoren

Die Entwicklung bildgebender Sensoren auf Röhrenbasis zu Beginn des 20.Jahrhunderts war ein Meilenstein der elektronischen Kommunikation. Der rasante technische Fortschritt ermöglichte den regulären Betrieb von Fernsehstudios bereits in den Dreißigerjahren. Schon mit dem nächsten Technologiesprung, der Entwicklung des CCD-Sensors im Jahr 1969 durch die Bell-Laboratories, wurde auf eine Technik umgestellt, die bis heute den

Markt dominiert. Diese Technologie hat zu einer rasanten Verbreitung der Kameras in verschiedensten Bereichen geführt. Der Sensor basiert auf der materialimmanenten Lichtempfindlichkeit des Siliziums. Eintreffende Photonen werden in Ladung umgewandelt und durch im Silizium implantierte Potentialwände räumlich fixiert. Beim Auslesevorgang werden die Ladungen innerhalb einer Zeile spaltenweise durch den Sensor bis zu einem Ladungsverstärker transportiert. Durch die geringen Ladungsmengen und die hohe Auslesefrequenz werden höchste Anforderungen an die Bandbreite und das Rauschverhalten des Verstärkers gestellt. Da die Ladungen nur durch Gate-Spannungen und Ladungsbarrieren am Ort fixiert werden, kann überschüssige Ladung leicht in benachbarte Spalten abfließen, was zu einem „*blooming*“-Effekt bei starken Lichtquellen führt. Vorteilhaft ist die Verwendung nur eines Verstärkers für alle Pixel, der im Multiplex benutzt wird, da die Verstärkungsparameter für alle Pixel identisch sind. Die Eigenschaften der Pixel eines Sensors sind somit weitgehend identisch, was sich in geringem ortsfestem Rauschen äußert.

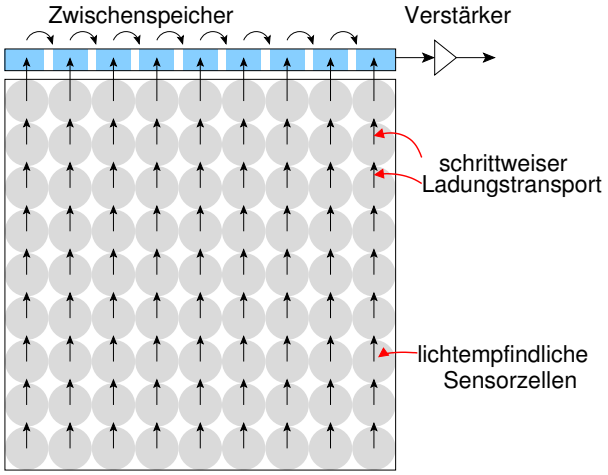


Abbildung 3.1: Funktionsprinzip des CCD-Sensors.

Der CCD-Sensor wird in jüngerer Zeit schrittweise von CMOS-Sensoren abgelöst. Diese Sensoren basieren ebenfalls auf der Ladungsintegration in den Pixeln, verstärken diese jedoch durch je einen Verstärker pro Pixel, der die Ladung vor Ort in eine Spannung wandelt. Diese Spannung wird im Multiplex nacheinander von einem im Sensor integrierten Analog-Digitalwandler umgewandelt und ausgegeben. Da kein analoger Ladungstransport von Pixel zu Pixel mehr nötig ist, können höhere Ladungsbarrieren zwischen den Pixeln eingesetzt werden, wodurch kein *blooming*-Effekt mehr auftritt. Durch die direkte Integration von lichtempfindlicher Fläche und Verstärker können pixelweise zusätzliche Funktionen, wie zum Beispiel ein elektronischer Verschluss, veränderte Kennlinien und Ähnliches implementiert werden. Gleichzeitig reduziert dies natürlich den für die Photodiode nutzbaren Flächenanteil des Siliziums. Durch die lokale Verstärkung ergibt sich ein stärkeres pixelabhängiges Rauschen [42], da die fertigungstechnisch schwieriger zu beherrschenden Verstärkerparameter jedes einzelnen Pixelverstärkers das Bild beeinflussen. Neuere Weiterentwicklungen der CMOS-Technik widmen sich insbesondere einer Erhöhung des Dynamikbereichs des Sensors. Tastet man die akkumulierte Ladung mehrmals während der

Integrationszeit ab, so kann auch bei Sättigung des Pixels am Ende der Integrationszeit aus den vorangegangenen Werten ein sinnvoller Pegel extrapoliert werden. Auf ähnliche Weise wurde durch die Kombination kurzer und langer Verschlusszeiten schon früher bei statischer Szene eine Erhöhung der Dynamik erreicht [12]. Als weiterer Nebeneffekt der mehrfachen Abtastung lässt sich durch sehr frühes Abtasten das pixelabhängige Rauschen deutlich reduzieren.

Eine weitere, ebenfalls auf CMOS-Technik basierende Sensorgeneration steht derzeit unter dem Namen HDRC (High Dynamic Range Control) in den Startlöchern. Hier wird die Ladung im Sensor nicht aufintegriert, sondern die Helligkeit im Pixel ähnlich einer Fotodiode zum Abtastzeitpunkt gemessen. Erste Produkte zeigen eine vielversprechende hohe Dynamik bei einer logarithmischen Kennlinie. Dabei scheint aber die Fertigungstechnik noch problematisch zu sein, da die Sensoren noch großes ortsfestes Rauschen und Pixelausfälle zeigen.

Farbe

Farbkameras werden entweder durch den Einsatz von Filtermasken vor den Pixeln realisiert, oder durch einen Strahlteiler im Strahlengang der Kamera und drei Sensorchips für die Grundfarben. Das erste Verfahren ist günstiger und dominiert somit den Markt für einfachere Systeme. Abbildung 3.2a veranschaulicht das so genannte Bayer-Pattern. Eine Filtermaske in den Grundfarben Rot, Grün und Blau filtert das einfallende Licht. Die Überrepräsentation der Farbe Grün hängt dabei mit der höheren spektralen Empfindlichkeit des Menschen für diese Farbe zusammen. Die tatsächliche Auflösung reduziert sich auf ein Viertel – trotzdem wird üblicherweise durch Interpolation ein Bild in voller Auflösung rekonstruiert.

Abbildung 3.2b zeigt eine neuere Alternative hierzu. Die Zeilen- und Spalten-Struktur herkömmlicher Sensoren wird um 45° gekippt. Damit reduziert sich der Pixelabstand auf $\sqrt{2}$ in horizontaler und vertikaler Richtung. Damit steigt die nutzbare Auflösung entlang dieser Richtungen entsprechend.

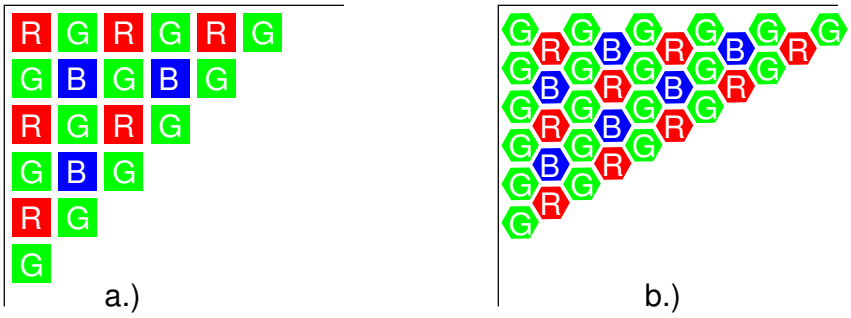


Abbildung 3.2: a.) Bayerpattern, b.) Um 45° gekippte Struktur erhöht die Auflösung in horizontaler und vertikaler Richtung.

Höherwertige Kameras setzen drei Sensoren ein, um die Auflösungsreduktion zu vermeiden und um zu jedem Pixel eine Farbaussage treffen zu können. Bei diesem aufwändigen Verfahren verkompliziert sich die Optik entsprechend, da die Abbildung auf die einzelnen Sensoren möglichst deckungsgleich sein muss.

Durch die Fortschritte bei den CMOS-Sensoren wurde es in jüngster Zeit möglich, Zonen mit unterschiedlicher spektraler Empfindlichkeit im selben Pixel zu integrieren. Im Halbleiter werden durch spezielle Dotierungen verschiedene photoempfindliche Schichten übereinander realisiert. Dabei wird die unterschiedliche Eindringtiefe der verschiedenen Wellenlängen ausgenutzt. Unter dem Namen *Foveon X3* vertreibt die Firma *Foveon*[1)] bereits Kamerasysteme, die auf diesem Prinzip beruhen. Sie ermöglichen, in jedem Pixel die Anteile der Grundfarben zu messen und damit die volle Auflösung bei gleichzeitig einfachem Aufbau der Kamera zu erreichen.

Schnittstellen

Durch den Übergang von der Analog- zur Digitaltechnik, der sich derzeit im Bereich der Kameras vollzieht, sind ihre Schnittstellen einem deutlichen Wandel unterworfen. Waren bis vor kurzem noch Kameras mit analogen Schnittstellen in Form von S-Video- oder Composite-Ausgängen der Standard in industriellen Anwendungen, so steigt seit einigen Jahren der Anteil digitaler Schnittstellen stetig. Dabei liegt der zentrale Unterschied im Ort der Wandlung von Analog nach Digital. Findet die Wandlung erst im Computer statt, wie dies bei analogen Systemen der Fall ist, ergeben sich verschiedene Nachteile:

- Lange Übertragungswege für das analoge Signal und damit Qualitätsverlust und Störempfindlichkeit
- Komplexe Wandlerkarten im Computer (Framegrabber) durch die Vielzahl an möglichen Kameravarianten (Auflösung, Bildwiederholfrequenz, Halbbild/Vollbild[2)])
- Parameter, die die Bildaufnahme beeinflussen (Empfindlichkeit, Verschlusszeit, Helligkeit, etc.), werden zum Teil am Framegrabber und zum Teil an der Kamera eingestellt. Dies verkompliziert standardisierte Softwareschnittstellen.
- Bildsignale lassen sich nicht ohne Zeitverlust in identischer Form auf verschiedenen Rechnern empfangen, wie dies für parallelisierte Bildverarbeitung wünschenswert ist.

Bei der **Camera-Link**-Schnittstelle handelt es sich um eine schnelle digitale Schnittstelle geringer Komplexität. Bilddaten werden parallel und synchron auf einer oder mehreren unidirektionalen Verbindungen in digitaler Form übertragen. Ein Rückkanal zur Kamera ist in Form einer einfachen asynchronen Schnittstelle mit geringer Geschwindigkeit

[1] www.foveon.com

[2] Bedingt durch das angestammte Einsatzgebiet der Kameras werden, um bei der Darstellung Flimmereffekte zu reduzieren, Bilder beim Halbbild-Verfahren in Halbbildern aufgenommen und übertragen (*interlaced/progessive scan*).

spezifiziert. Durch die geringe Komplexität und die Verwendung eines weit verbreiteten Standards (*channel link LVDS* gemäß ANSI/TIA/EIA-644) für die physikalische Umsetzung ist eine günstige Umsetzung in Hardware möglich, wodurch sich wiederum eine große Verbreitung im industriellen Umfeld ergibt.

Im Gegensatz dazu entstammt die **Firewire**- oder **IEEE1394**- Schnittstelle der PC-Welt. Es handelt sich dabei um einen synchronen seriellen Bus mit physikalischer Punkt-zu-Punkt-Topologie und logischer Baumstruktur. Die maximale Bandbreite beträgt 400 MBit (bzw. 800 MBit IEEE1394b) bei einem Maximum von 63 Geräten pro Bus. Die *Digital Camera Specification* standardisiert das Interface zu digitalen Kameras. Der Markt bietet zum gegenwärtigen Zeitpunkt rund 300 Kameratypen, die dieser Spezifikation entsprechen.

Die **USB**-Schnittstelle ähnelt Firewire und entstammt ebenfalls der PC-Welt. Sie besitzt mit etwa 480 MBit seit Version 2.0 ausreichende Bandbreite für die Videoübertragung, wird aber aufgrund ihrer Historie (zu geringe Bandbreite in Version 1.1, keine garantierte Bandbreite) nicht im professionellen Umfeld eingesetzt. Für das Anwendungsumfeld der Videokonferenz für den PC ist eine Vielzahl von Kameras verfügbar. Sie sind jedoch nicht in der Lage, unkomprimierte Videoströme schritthaltend zu übertragen.

3.1.2 Mathematisches Modell

Die geometrischen Verhältnisse in der Anordnung einzelner Kamerakomponenten und der Kameras zueinander sowie die optischen Eigenschaften der eingesetzten Linsensysteme führen zu einer sehr komplexen mathematischen Beschreibung des Systems. Da aber bestimmte Effekte wie zum Beispiel die wellenlängenabhängigen Eigenschaften der Linsen häufig vernachlässigbar sind, wird üblicherweise ein vereinfachendes mathematisches Modell angenommen.

Lage der Kamera im Raum: die externen Kameraparameter

Ein Punkt ${}^{W}\mathbf{s}$ im dreidimensionalen Raum muss vor der Projektion zunächst in das Kamerakoordinatensystem transformiert werden. Die Lage der Kamera im Raum, also die Transformation aus dem Kamerakoordinatensystem in das Weltkoordinatensystem wird durch einen Translationvektor und eine 3x3-Rotationmatrix ausgedrückt.

$$ {}^{K}\mathbf{s} = \mathbf{R}({}^{W}\mathbf{s} + \mathbf{t}) \tag{3.1} $$

Gleichung 3.1 stellt die Abbildung des Punktes ${}^{W}\mathbf{s}$ in das Kamerasystem dar. Dabei besitzt die orthonormale 3x3-Rotationsmatrix $\mathbf{R}$ natürlich nur drei Freiheitsgrade. Die Darstellung in Form dreier Winkel ist jedoch nicht eindeutig. Auch wenn eine klare Spezifikation der Reihenfolge der elementaren Drehungen gegeben ist, gibt es mehrdeutige Formulierungen einer eindeutigen Orientierung. Im Rahmen dieser Arbeit wird ausschließlich die Darstellung in Eulerwinkeln mit Drehreihenfolge um die Z-Achse (Schwenken/*yaw*), Y-Achse (Neigen/*pitch*) und X-Achse (Rollen/*roll*) verwendet.

Häufig findet sich in der Literatur auch die Darstellung in homogenen Koordinaten gemäß Gleichung 3.2, da diese Darstellung die einheitliche Behandlung von Rotation und Translation ermöglicht. Die oben genannte Transformation lässt sich wie folgt in Form einer Transformationsmatrix ${}^{K}\mathbf{T}_{W}$ vom System $\mathcal{F}_W$ in das System $\mathcal{F}_K$ ausdrücken.

$$ {}^{K}\mathbf{T}_{W} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2} $$

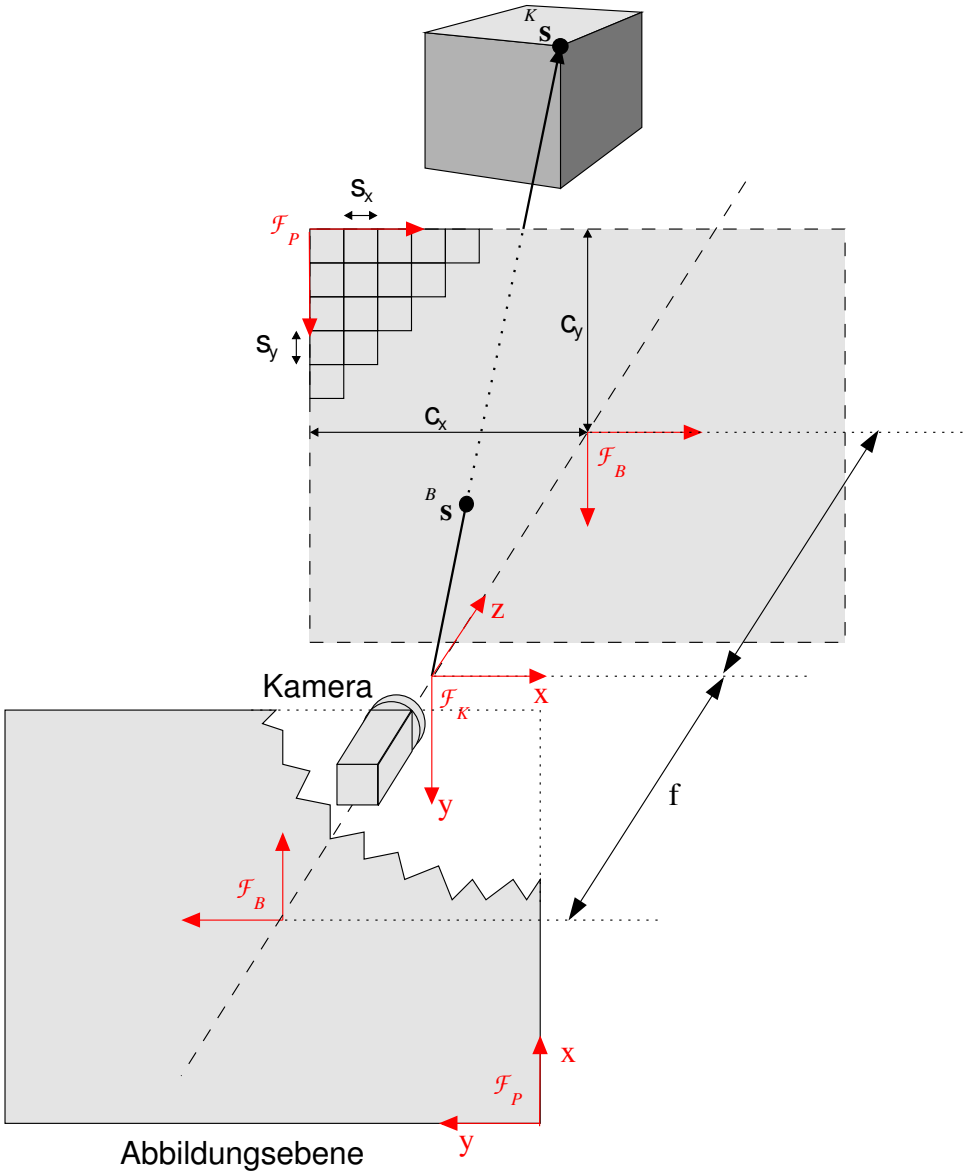


Abbildung 3.3: Monokulares Kameramodell.

Von 3D nach 2D: die Projektion

Beim verwendeten Kameramodell handelt es sich um ein einfaches Lochkameramodell mit radialer Verzerrung. Abbildung 3.3 veranschaulicht die Verhältnisse. Ein Punkt ${}^{K}\mathbf{s}$ wird durch das Linsensystem mit dem Brennpunkt im Nullpunkt des Kamerakoordinatensystems und der Brennweite f auf den Punkt ${}^{B}\mathbf{s}$ auf der lichtempfindlichen Sensoroberfläche abgebildet. Aus Gründen der Anschaulichkeit wird die physikalische Bildumkehrung häufig vernachlässigt und die Bildebene an der in Abbildung 3.3 gestrichelten Stelle *vor* dem Brennpunkt der Kamera angenommen.

Linsenverzerrungen

Zusätzlich zu dieser affinen Abbildung wird die Verzerrung der Linse in radialer Richtung modelliert. Dies ist insbesondere bei kleinen Brennweiten erforderlich, da die Verzerrungen in diesem Fall besonders groß werden. Abbildung 6.17 veranschaulicht den Effekt dieser Verzerrung. Die Verzerrung wird im System $\mathcal{F}_B$ abhängig vom radialen Abstand von der Mittelachse des Linsensystems der Kamera, also dem Nullpunkt des Systems $\mathcal{F}_B$ in radialer Richtung modelliert. Die Verzerrung wird gemäß Gleichung 3.3 als Reihenentwicklung modelliert, die nach dem ersten oder zweiten Glied abgebrochen wird.

$$v_x = u_x \cdot (1 + \kappa_1 r^2 + \kappa_2 r^4 + ...) \tag{3.3}$$

$$r^2 = v_x^2 \cdot v_y^2 \tag{3.4}$$

Nur bei einem Abbruch nach dem ersten Glied existiert eine analytische Umkehrung wie in Gleichung 3.5 angegeben.

$$u_x = \frac{v_x}{1 + \kappa_1 (v_x^2 + v_y^2)} \tag{3.5}$$

$$v_x = \frac{2u_x}{1 + \sqrt{1 - 4\kappa_1 (u_x^2 + u_y^2)}} \tag{3.6}$$

Quantisierung

Auf der Sensorebene erfolgt die schon genannte räumliche Quantisierung in Sensorelemente oder Pixel. Die Pixelbreite und -höhe werden als s_x und s_y bezeichnet. Der Eintrittspunkt des Lotes durch den Brennpunkt auf der Sensorebene, also der Punkt des Durchtritts der z-Achse durch die Bildebene, wird als Hauptpunkt bezeichnet. Die Koordinaten eines Pixels werden jedoch nicht vom Hauptpunkt, sondern ausgehend von der linken oberen Sensorecke gezählt. Der sich ergebende Offset zwischen diesen *technischen Pixelkoordinaten* ($\mathcal{F}_P$) und der Lage des Hauptpunktes ($\mathcal{F}_B$) wird durch die Parameter

c_x und c_y angegeben. Die Projektion eines dreidimensionalen Punktes auf die Bildebene lässt sich also wie folgt formulieren:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cdot s = \begin{bmatrix} \frac{f}{s_x} & 0 & c_x \\ 0 & \frac{f}{s_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot {}^{K}\mathbf{s} \tag{3.7}$$

3.1.3 Kalibrierung

Unter dem Begriff Kalibrierung versteht man im Kontext von Kamerasystemen die Bestimmung der Parameter eines zugrundegelegten mathematischen Modells des Systems aus messtechnisch erzeugten Daten. Da die Kalibrierung eines Kamerasystems zwar mittlerweile ein Standardvorgang ist, die Praxis jedoch häufig viel Kontextwissen über Konzepte und Alternativen verlangt, soll das Thema Kalibrierung hier kurz und eher praxisorientiert behandelt werden.

Intrinsische Kalibrierung

Die internen Parameter einer Kamera (f,κ_1, κ_2, c_x, c_y, s_x, s_y, b, h) werden bei diesem Prozess aus einer Serie von Aufnahmen bekannter Objekte bestimmt. Ein zwei- oder dreidimensionaler *Kalibrierkörper* oder *Eichkörper* mit bekannter Geometrie wird mit einer Kamera mit identischen Einstellungen (insbesondere Fokus und Zoom) mehrmals abgebildet. In den entstandenen Bildern lassen sich manuell oder durch automatisierte Bildverarbeitung die Merkmale des Kalibrierkörpers bestimmen und zuordnen. Die resultierenden 2D-3D-Korrespondenzen ermöglichen zusammen mit dem Kameramodell und den Abbildungsgleichungen die Aufstellung eines überbestimmtes Gleichungssystems. Mit Hilfe einer initialen Parameterschätzung (zum Beispiel aus dem Datenblatt der Kamera) lassen sich die Parameter durch ein numerisches Lösungsverfahren bestimmen. Die intrinsischen Parameter lassen sich nach dieser Methode bequem mit Hilfe des Softwarepaketes *HALCON*[1)] bestimmen.

Extrinsische Kalibrierung, Kamera zu Drehpunkt

Soll ein Raum mit Hilfe einer auf einer Schwenk-Neige-Plattform angebrachten Kamera exploriert werden, so ist die Lage der Kamera relativ zum Drehpunkt zu bestimmen. Dies wird wieder mit Hilfe von HALCON und der *hand-eye calibration* umgesetzt. Die zu kalibrierende Transformationskette soll hier anhand von Abbildung 3.4 dargestellt werden.

Der Kalibrierkörper wird bei dieser Methode nicht bewegt und dient als messtechnische Referenz. Er bildet das Weltkoordinatensystem, dessen Achsen ähnlich denen des Kamerakoordinatensystems liegen, wie in Abbildung 3.4 dargestellt. Die feste Transformation

[1] Ein kommerzielles Produkt der Firma MVTec Software GmbH (www.mvtec.com).

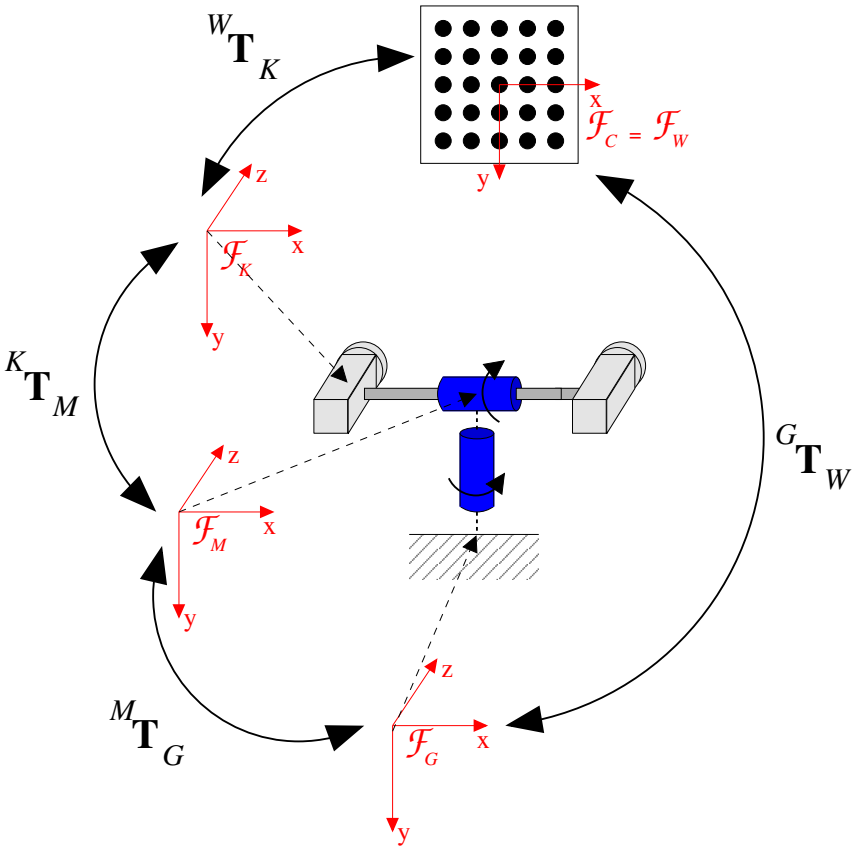


Abbildung 3.4: Kalibrierung des Schwenk-Neige-Kopfes.

${}^{G}\mathbf{T}_{W}$ führt in das ebenfalls ortsfeste Basiskoordinatensystem des Schwenk-Neige-Kopfes. Das Schwenk- und das Neigegelenk werden für jede Aufnahme bewegt und die Transformation ${}^{M}\mathbf{T}_{G}$ aus den Winkeln berechnet. Die zu kalibrierende Transformation ${}^{K}\mathbf{T}_{M}$ führt in das Kamerakoordinatensystem. Per Bildverarbeitung wird die Transformation ${}^{K}\mathbf{T}_{W}$ gemessen. Diese geschlossene Transformationskette ermöglicht es, die Transformationen ${}^{G}\mathbf{T}_{W}$ und ${}^{K}\mathbf{T}_{M}$ durch Lösen eines überbestimmten Gleichungssystems zu bestimmen.

3.1.4 Registrierung der Kamera

Ein zur extrinsischen Kalibrierung der Kamera verwandtes Problem ist die Registrierung der Kamera gegen Modelldaten. Sie ermöglicht im Idealfall einen Modellaufbau bei bewegter Kamera und die gleichzeitige fortschreibende Positionsmessung der Kamera relativ zum bereits bestehenden Modell. Obwohl in dieser Arbeit die Kameraposition durch die Schwenk-Neige-Einheit meist als bekannt vorausgesetzt wird, wurden auch Experimente mit frei beweglicher Kamera durchgeführt. Der folgende Abschnitt erläutert ein Standardverfahren zur Registrierung von rekonstruierten Punktewolken gegeneinander, das in dieser Arbeit für derartige Experimente eingesetzt wurde.

Der ICP-Algorithmus

Der *„Iterative Closest Point“*-Algorithmus (ICP) wurde 1992 von Besl und McKay, in einer Arbeit über die Registrierung von Punktewolken im dreidimensionalen Raum [4], beschrieben. Dabei handelt es sich um einen Algorithmus, der zum Lösen eines schon länger bekannten Problems des Robotersehens verwendet werden kann:
Bestimme zu einem gegebenen 3D-Datensatz in einem Sensorkoordinatensystem und einem Datensatz in einem Modellkoordinatensystem die Transformation, die Modelldaten und Sensordaten am besten aufeinander abbildet, so dass der euklidische Abstand zwischen den Punktewolken im Sinne des mittleren Fehlerquadrats minimiert wird.
Der Algorithmus kann für die Registrierung von Punktewolken, Liniensegmenten, parametrisierten und impliziten Kurven, Dreiecken, parametrisierten und impliziten Flächen verwendet werden [4]. In dieser Arbeit soll jedoch nur auf die Verwendung im Zusammenhang mit Punktewolken eingegangen werden.

Gegeben sei eine Daten-Punktewolke $\mathcal{P} = \{\vec{p}_i\}$, mit N_p Punkten, und eine Modell-Punktewolke $\mathcal{X} = \{\vec{x}_i\}$, mit N_x Punkten.

Mathematisch kann die Funktion, die zur Lösung des Problems minimiert werden muss, folgendermaßen beschrieben werden:

$$f(T) = \frac{1}{N_p} \sum_{i=0}^{N_p} (\vec{x}_i - T * \vec{p}_i)^2 \tag{3.8}$$

Der ICP-Algorithmus beruht darauf, die Transformationsmatrix T iterativ zu bestimmen. Bei diesem Annäherungsverfahren werden drei grundlegende Arbeitsschritte mehrmals wiederholt, bis ein Abbruchkriterium erfüllt ist.

1. **Nächste Nachbarn finden** Als Korrespondenzpunkte werden beim ICP-Algorithmus Punkte einer Wolke und ihre nächsten Nachbarn in der anderen Wolke angenommen. In der Arbeit von Besl und McKay wird der Beweis erbracht, dass der Gesamtabstand zweier Punktewolken über alle Punkte in ein Minimum konvergiert, wenn man in jedem Iterationsschritt eine Registrierung der nächsten Nachbarn durchführt [4]. Für jeden Punkt $\vec{p}_i$ wird hierfür die euklidische Distanz zu jedem Punkt der Menge $\mathcal{X}$ berechnet. Der Punkt $\vec{x}_i$ mit der minimalen Distanz zu $\vec{p}_i$ wird gemerkt. Das heißt, es werden Punktpaare ($\vec{p}_i$,$\vec{x}_i$) mit der minimalen euklidischen Distanz aus der Sicht der Daten-Punktewolke $\mathcal{P}$ erzeugt.

   Aus den entstandenen Punktkorrespondenzen wird ein Gleichungssystem der Form

   $$X = T_n * P = R_n * P + t_n \tag{3.9}$$

   erstellt. Die Matrix P trägt in jeder Spalte den Positionsvektor eines Punktes $\vec{p}_i$ in homogenen Koordinaten und die Matrix X die jeweiligen nächsten Nachbarn. Beide haben die Dimension $4 \times N_P$.

2. **Transformation berechnen** In diesem Schritt wird das Gleichungssystem nach der 4x4 Transformationsmatrix T_n aufgelöst. Diese soll die optimale Rotation und Translation der homogenen Koordinaten im n-ten Iterationsschritt beschreiben. Bei Gleichung 3.9 handelt es sich um ein nichtlineares und überbestimmtes Gleichungssystem. Beim Lösen dieses Gleichungssystems muss darauf geachtet werden, dass es sich nur um eine homogene Rotationsmatrix mit translatorischem Anteil handeln darf. Das heißt, der rotatorische Anteil muss orthonormal sein. Dazu müssen beim Lösen von Gleichung 3.9 Bedingungen für T_n erfüllt werden.
   Dies wird mit Hilfe von Quaternionen in einer geschlossenen Lösung nach Horn [40] erreicht. Dabei wird die Transformation in eine Rotationsmatrix und einen Translationsvektor aufgespalten. Deren optimale Lösungen werden getrennt voneinander berechnet. Diese werden dann in die Lösungsmatrix T_n eingetragen.

3. **Transformation anwenden** Die berechnete Transformation wird in jedem Iterationsschritt auf die Punkte $\vec{p}_i$ angewandt. Zudem wird die Gesamttransformation jedes Mal aktualisiert:
$$T_{ges_n} = T_n * T_{ges_{n-1}}$$

Besl und McKay verwenden in ihrer Arbeit als Abbruchkriterium einen minimalen Grenzwert für die Änderung des mittleren quadratischen Fehlers von einer Iteration zur nächsten als Abbruchkriterium [4]. Dies kann aber hier, aus später angebrachten Gründen, nicht verwendet werden.

Verbesserungen des ICP-Algorithmus Verbesserungen des ICP-Algorithmus zielen darauf ab, die Ausführungszeit, bis das Abbruchkriterium erfüllt ist, zu verkürzen. Der Rechenaufwand für die Suche nach dem nächsten Nachbarn beträgt über 90% des gesamten Rechenaufwands [44]. Deshalb werden große Anstrengungen aufgebracht, diese Zeit zu verkürzen. Dies kann auf zwei Arten geschehen. So können Zusatzkriterien bei der Suche nach dem nächsten Nachbarn helfen und damit die Konvergenzgeschwindigkeit erhöhen, oder es kann die Suche nach dem nächsten Nachbarn beschleunigt werden. Einige gängige Methoden sollen hier genannt werden.

- **Trimmed ICP**
  Eine große Einschränkung des ursprünglichen ICP-Algorithmus ist es, dass die zu registrierende Datenpunktewolke $\mathcal{P}$ eine Untermenge der Modellpunktewolke $\mathcal{X}$ sein muss. Da im vorliegenden Fall jedoch nur eine gewisse Überlappung zwischen Bildern vorliegt, ist dies nicht gegeben. Der Trimmed-ICP-Algorithmus führt hier eine Überlappungskonstante *Overlap* ein, um die Robustheit des Algorithmus zu steigern. Diese Konstante legt den Anteil der Punkte fest, die in jedem Iterationsschritt für die Berechnung der Transformation berücksichtigt werden.
  Um Ausreißer und falsche Korrespondenzen zu vernachlässigen, werden die gefundenen Punktpaare nach ihrem jeweiligen Abstand zueinander in aufsteigender Reihenfolge sortiert. Nur der festgelegte Anteil der Punkte mit kleinerem Abstand wird berücksichtigt. So können laut [48] Punktewolken mit einem Überlappungsgrad

von unter 50% registriert werden.

- **Iterative Closest Reciprocal Point (ICRP)**
  Eine weitere Möglichkeit, den ICP-Algorithmus zu verbessern, ist, alle gefundenen Punktpaare aus der Sicht der anderen Punktewolke zu prüfen. Pajadla und Van Gool [68] schlagen vor, nach der Suche des nächsten Nachbarns x_i eines Datenpunktes p_i, für diesen Modellpunkt x_i ebenfalls den nächsten Nachbarn p'_i in der Datenpunktewolke zu suchen. Falls der euklidische Abstand $\|p_i - p'_i\| > \epsilon$ wird, wird eine Fehlkorrespondenz angenommen und das Punktepaar verworfen.

- **Verwendung von kD-Trees**
  Die Berechnung des Abstands aller Punkte zueinander wird in jeder Iteration ausgeführt. Sie erfordert $N_p * N_x$ Vergleichsoperationen. Eine oft verwendete Methode, den Rechenaufwand zu dezimieren, ist die Verwendung von Suchbäumen zur Suche des nächsten Nachbarn. Dabei stellen kD-Trees die Standardlösung im Zusammenhang mit dem ICP-Algorithmus dar. Der Rechenaufwand wird mit dieser Methode auf $N_p \log N_x$ Vergleiche gesenkt.

- **Zusatzkriterien für die Nachbarnsuche**
  Bei der Korrespondenzsuche der nächsten Nachbarn ist es in den ersten Iterationsschritten sehr unwahrscheinlich, dass der nächste Nachbar auch wirklich der richtige korrespondierende Punkt ist. Es gibt verschiedene Varianten, diese Wahrscheinlichkeit zu erhöhen. Gängige Methode ist es, die Farbkomponente oder die Oberflächennormale der Punkte mit einzubeziehen [44].

- **Beschleunigung durch 2D-Korrespondenzen**
  Durch die Verwendung von bekannten 2D-Korrespondenzen kann die Registrierung erheblich beschleunigt werden. Dafür müssen Punktkorrespondenzen zwischen den beiden Aufnahmen (derselben Kamera) berechnet werden. Zu diesen Korrespondenzpunkten in Bildkoordinaten können für beide Aufnahmen die entsprechenden 3D-Punkte der zugehörigen Punktewolke gefunden werden. Stellt man hohe Anforderungen an die Korrespondenzsuche, so wird die Punktewolke auf wenige Punkte beschränkt. Aus dieser initialen Registrierung kann eine gute erste Schätzung der Kameralage berechnet werden.

3.2 Stereopsis

3.2.1 Problemstellung

Beobachten mehrere Kameras dieselbe Szene zum selben Zeitpunkt aus unterschiedlichen Positionen, so kann aus dem Unterschied der Bilder die dreidimensionale Struktur der beobachteten Szene berechnet werden, wenn diese geeignet strukturiert ist. Hierzu müssen zunächst korrespondierende Punkte in beiden Aufnahmen gefunden werden. Dies

wird als das **Korrespondenzproblem** bezeichnet. Vorgegeben durch die Geometrie der Stereoanordnung, also ihrer **externen Kalibrierung**, können Einschränkungen für die Lagebeziehung korrespondierender Punkte festgelegt werden. Die Ausnutzung dieser *constraints*[1)] erlaubt eine Vereinfachung des Problems.

Die Lagebeziehung korrespondierender Punkte kann nun bei bekannten externen Kameraparametern zur Bestimmung der dreidimensionalen Lage des zugrundeliegenden Punktes herangezogen werden, also zur **Rekonstruktion** der Szene.

In diesem Kapitel werden die Grundlagen der Stereorekonstruktion vorgestellt, um im darauf folgenden Abschnitt 3.3 einige Ansätze im Detail vorzustellen.

3.2.2 Korrespondenzfindung

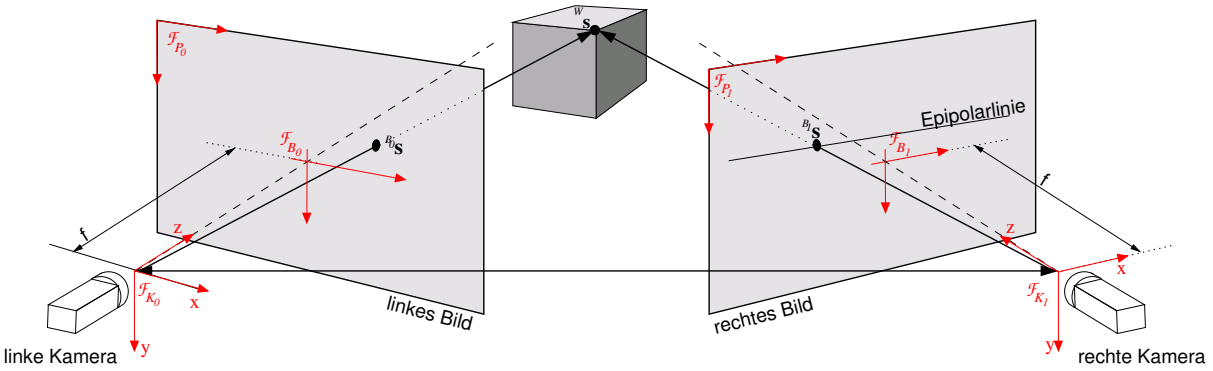


Abbildung 3.5: Modell eines Stereokamerasystems.

Das Stereosystem sei auf zwei Kameras beschränkt angenommen, wie sie in Abbildung 3.5 dargestellt sind. Sie beobachten den selben Punkt ${}^{W}\mathbf{s}$, der durch die Kameras auf den Punkt ${}^{B_0}\mathbf{s}$ bzw. ${}^{B_1}\mathbf{s}$ abgebildet wird. Die Umkehrung dieser Projektion ist die zentrale Fragestellung.

Zur Lösung dieses **Korrespondenzproblems** existieren grundlegend unterschiedliche Ansätze. **Merkmalsbasierte Verfahren** extrahieren zunächst in beiden Bildern Merkmale, um diese dann zu vergleichen und einander entsprechende Merkmale zu bestimmen. **Korrelationsbasierte Verfahren** korrelieren Bildbereiche durch ein Ähnlichkeitsmaß mit dem Ziel, Entsprechungen zu finden.

3.2.3 Epipolargeometrie

Die nähere Betrachtung der Geometrie eines Stereoaufbaus offenbart einige nützliche Eigenschaften. Betrachtet man in Abbildung 3.5 im linken Kamerabild nur den Punkt ${}^{B_0}\mathbf{s}$,

[1] Der in der englischsprachigen Fachliteratur etablierte Begriff *constraint* lässt sich in deutscher Sprache nur unzureichend mit *Einschränkung* oder *(Zwangs-)Bedingung* übersetzen. Aus diesem Grund wird er im Rahmen dieser Arbeit als Fachbegriff im Original verwendet.

so kann bereits eine Aussage über die Position des zugrundeliegenden Punktes in 3D getroffen werden: Er muss auf einem Strahl durch den Ursprung der linken Kamera und ${}^{B_0}\mathbf{s}$ liegen. Die Abbildung dieses Strahls auf die Bildebene der rechten Kamera ergibt eine zu ${}^{B_0}\mathbf{s}$ gehörende *Epipolarlinie*. Auf dieser Linie muss die Abbildung von ${}^{W}\mathbf{s}$ in die rechte Bildebene, also der Punkt ${}^{B_1}\mathbf{s}$ liegen.

Eine Epipolarlinie schränkt somit den Suchraum bei der Korrespondenzsuche auf ein eindimensionales Problem ein.

Die Abbildung des Brennpunktes einer Kamera in die Bildebene der anderen Kamera heißt *Epipol*. Alle Epipolarlinien passieren diesen Punkt, da die dazugehörigen Strahlen im dreidimensionalen Raum alle den Brennpunkt der Kamera passieren. Sind die Bildebenen beider Kameras planparallel, so ergeben sich Epipole in unendlicher Entfernung und parallele Epipolarlinien.

Die **essentielle Matrix E** stellt diesen Zusammenhang zwischen den Projektionen ${}^{B_0}\mathbf{s}$ und ${}^{B_1}\mathbf{s}$ des Punktes $\mathbf{s}$ dar.

$$ {}^{B_1}\mathbf{s}^{\mathsf{T}}\mathbf{E}{}^{B_0}\mathbf{s} = 0 \tag{3.10} $$

Sie ist jedoch in Bildkoordinaten $\mathcal{F}_{B_0}$ bzw. $\mathcal{F}_{B_1}$ definiert. Die Darstellung dieser Epipolarbeziehung in Pixelkoordinaten im System $\mathcal{F}_P$ heißt **Fundamentalmatrix F**. Gleichungen 3.11 bis 3.13 stellen die Umformungen dar.

$$ {}^{P_1}\mathbf{s}^{\mathsf{T}}\mathbf{F}{}^{P_0}\mathbf{s} = 0 \tag{3.11} $$

$$ {}^{P}\mathbf{s} = \mathbf{P} \cdot \begin{bmatrix} {}^{B}\mathbf{x} \\ 1 \end{bmatrix} \quad bzw. \quad {}^{P}\mathbf{s} = \mathbf{P}^{-1} \cdot \begin{bmatrix} {}^{B}\mathbf{x} \\ 1 \end{bmatrix} \tag{3.12} $$

$$ \mathbf{F} = {}^{P_1}\mathbf{P}_{B_1}{}^{-\mathsf{T}} \quad \mathbf{E} \quad {}^{P_0}\mathbf{P}_{B_0}{}^{-1} \tag{3.13} $$

3.2.4 Rektifizierung

In vielen Stereoanwendungen sind parallele und horizontale Epipolarlinien und damit ein perfekt paralleler Aufbau des Kamerasystems wünschenswert, um Algorithmen zur Auswertung effektiv implementieren zu können. Doch lässt sich dies selbst durch höchste Präzision im mechanischen Aufbau nicht gewährleisten. Die Gründe dafür sind vielfältig:

- Linsenverzerrung
- Montage des Sensors nicht im Lot der Linse und verdreht gegen die Kamerahorizontale
- Optischer Mittelpunkt der Optik stimmt nicht mit Sensormittelpunkt überein

Durch die im Normalfall nicht parallelen und durch die Linsenverzerrung gekrümmten Epipolarlinien muss in einer Stereo-Anwendung also entweder eine eindimensionale Suche entlang der Epipolarlinien oder eine zweidimensionale Suche in der Nähe der Epipolarlinen

erfolgen. Um den damit verbundenen Aufwand zu reduzieren, können die Bilder dergestalt entzerrt werden, dass die Epipolarlinien parallel und an den Zeilen oder Spalten des Bildes ausgerichtet sind. Diese zweidimensionale Entzerrung der Bilder wird als **Rektifizierung** bezeichnet.

Sind die Zielparameter des rektifizierten Systems bekannt, kann eine Abbildungsmatrix vom verzerrten System in das entzerrte System bestimmt werden. Es muss jedoch wie oben bereits angesprochen auch die radiale Linsenverzerrung berücksichtigt werden, die sich natürlich nicht in einer Entzerrungsmatrix ausdrücken lässt.

Im folgenden Abschnitt werden die rektifizierten Koordinatensysteme mit Kleinbuchstaben bezeichnet. So handelt es sich beispielsweise beim System $\mathcal{F}_{k_0}$ um das entzerrte Kamerakoordinatensystem der linken Kamera, deren ursprüngliches Koordinatensystem mit $\mathcal{F}_{K_0}$ bezeichnet wird. Abbildung 3.6 zeigt die Lage der neuen (virtuellen) Kameras mit ihren Koordinatensystemen.

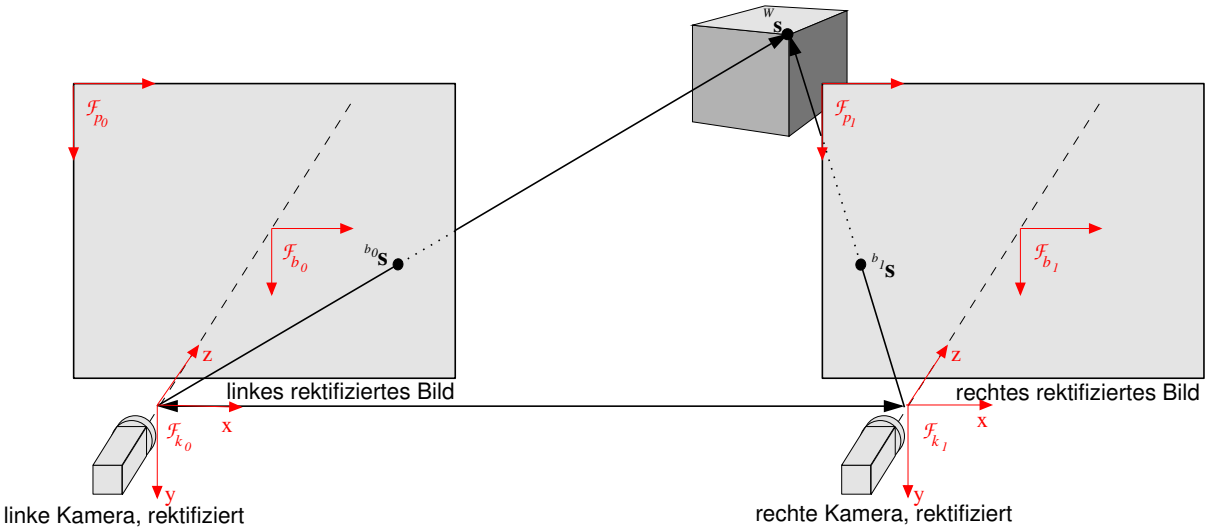


Abbildung 3.6: Modell eines rektifizierten Stereokamerasystems.

In einem rektifizierten Kamerasystem müssen beide Bildebenen in derselben Ebene liegen und die Kamerakoordinatensysteme $\mathcal{F}_{k_0}$ und $\mathcal{F}_{k_1}$ dürfen nur entlang ihrer x- oder y-Achse verschoben sein. Dies kann durch Rotation der Kamerasysteme um ihren Brennpunkt und Anpassung der Brennweite erreicht werden. Im direkten Vergleich der Abbildungen 3.5 und 3.6 wird die erforderliche Rotation verdeutlicht. Als Resultat sind die Bildebenen der beiden rektifizierten Kameras koplanar.

Um die rektifizierenden Projektionsmatrizen eines idealen Kamerasystems ${}^{p_0}\mathbf{P}_W$ und ${}^{p_1}\mathbf{P}_W$ aus den Projektionsmatrizen ${}^{P_0}\mathbf{P}_W$ und ${}^{P_1}\mathbf{P}_W$ zu berechnen, wird ein lineares und homogenes Gleichungssystem aus den Bedingungen für rektifizierte Systeme aufgestellt. Eine detaillierte Darstellung dieser Einzelbedingungen wird von Fusiello et al. [28] for-

muliert. Hier sollen die resultierenden Gleichungen jeweils kurz aufgeführt werden. Sie basieren auf einer Zerlegung der Projektionsmatrix gemäß Gleichung 3.14.

$$ {}^{p_0}\mathbf{P}_W = \left(\begin{array}{c|c} \mathbf{a}_1^\mathsf{T} & a_{14} \\ \mathbf{a}_2^\mathsf{T} & a_{24} \\ \mathbf{a}_3^\mathsf{T} & a_{34} \end{array}\right) \quad {}^{p_1}\mathbf{P}_W = \left(\begin{array}{c|c} \mathbf{b}_1^\mathsf{T} & b_{14} \\ \mathbf{b}_2^\mathsf{T} & b_{24} \\ \mathbf{b}_3^\mathsf{T} & b_{34} \end{array}\right) \tag{3.14} $$

- gleiche Skalierung

$$ \| \mathbf{a}_1 \| = 1 \quad \| \mathbf{b}_3 \| = 1 \tag{3.15} $$

- unveränderte Brennpunkte

$$ \begin{cases} \mathbf{a}_1^\mathsf{T}\,\mathbf{c}_1 + a_{14} = 0 \\ \mathbf{a}_2^\mathsf{T}\,\mathbf{c}_1 + a_{24} = 0 \\ \mathbf{a}_3^\mathsf{T}\,\mathbf{c}_1 + a_{34} = 0 \\ \mathbf{b}_1^\mathsf{T}\,\mathbf{c}_1 + b_{14} = 0 \\ \mathbf{b}_2^\mathsf{T}\,\mathbf{c}_1 + b_{24} = 0 \\ \mathbf{b}_3^\mathsf{T}\,\mathbf{c}_1 + b_{34} = 0 \end{cases} \tag{3.16} $$

- gemeinsame Abbildungsebene

$$ \mathbf{a}_3 = \mathbf{b}_3 \quad a_{34} = b_{34} \tag{3.17} $$

- gleiche Epipolarlinien

$$ \mathbf{a}_2 = \mathbf{b}_2 \quad a_{24} = b_{24} \tag{3.18} $$

Diese notwendigen Bedingungen reichen für eine Rektifizierung aus, sie definieren jedoch zusammen noch keine eindeutige Lösung. So könnten die Bildebenen zum Beispiel stark um ihre x-Achse gekippt sein, was zu unnötig verzerrten Bildern führt. Es müssen also weitere Bedingungen festgelegt werden, die zu einer sinnvollen Lösung führen.

- Die Orientierung der gemeinsamen Abbildungsebene soll parallel zum Schnitt der unrektifizierten Ebenen sein.

$$ \mathbf{a}_3^\mathsf{T}(\mathbf{f}_0 \wedge \mathbf{f}_1) = 0 \qquad (\mathbf{f}_0 = [\mathbf{a}_3^\mathsf{T}\, a_{34}] \quad \mathbf{f}_1 = [\mathbf{b}_3^\mathsf{T}\, b_{34}]) \tag{3.19} $$

- Die rektifizierten Systeme $\mathcal{F}_{p_0}$ und $\mathcal{F}_{p_1}$ müssen orthogonale Systeme sein.

$$ \mathbf{a}_1^\mathsf{T}\,\mathbf{a}_2 = 0 \quad \mathbf{b}_1^\mathsf{T}\,\mathbf{b}_2 = 0 \tag{3.20} $$

- Die neuen Hauptpunkte seien am Punkt (0,0)

$$ \begin{cases} \mathbf{a}_1^\mathsf{T}\,\mathbf{a}_3 = 0 \\ \mathbf{a}_2^\mathsf{T}\,\mathbf{a}_3 = 0 \\ \mathbf{b}_1^\mathsf{T}\,\mathbf{a}_3 = 0 \end{cases} \tag{3.21} $$

- Die Brennweite der rektifizierten Projektionsmatrizen sei gleich und festgelegt (z. B. auf die Brennweiten der linken Kamera α_x, α_y).

$$\begin{cases} \| \mathbf{a}_1 \wedge \mathbf{a}_3 \|^2 = \alpha_x^2 \\ \| \mathbf{a}_2 \wedge \mathbf{a}_3 \|^2 = \alpha_y^2 \\ \| \mathbf{b}_1 \wedge \mathbf{a}_3 \|^2 = \alpha_x^2 \end{cases} \tag{3.22}$$

Das resultierende Gleichungssystem lässt sich als generalisiertes Eigenvektorproblem lösen [28]. Vorausgesetzt, $\mathcal{F}_{K_0}$ ist identisch mit $\mathcal{F}_W$, besitzen die beiden rektifizierenden Projektionsmatrizen ${}^{p_0}\mathbf{P}_W$ und ${}^{p_1}\mathbf{P}_W$ dann folgende Form:

$${}^{p_0}\mathbf{P}_W = \left(\begin{array}{c|c} \mathbf{a}_1^\mathsf{T} & 0 \\ \mathbf{a}_2^\mathsf{T} & 0 \\ \mathbf{a}_3^\mathsf{T} & 0 \end{array}\right) \quad {}^{p_1}\mathbf{P}_W = \left(\begin{array}{c|c} \mathbf{a}_1^\mathsf{T} & d \\ \mathbf{a}_2^\mathsf{T} & 0 \\ \mathbf{a}_3^\mathsf{T} & 0 \end{array}\right) \text{ mit } \quad d < 0 \tag{3.23}$$

Das heißt, die extrinsischen Parameter beider rektifizierter Kameras besitzen denselben rotatorischen Anteil, und ihr translatorischer Anteil ist beschränkt auf nur eine Achse.

3.2.5 Rekonstruktion

Allgemeiner Fall

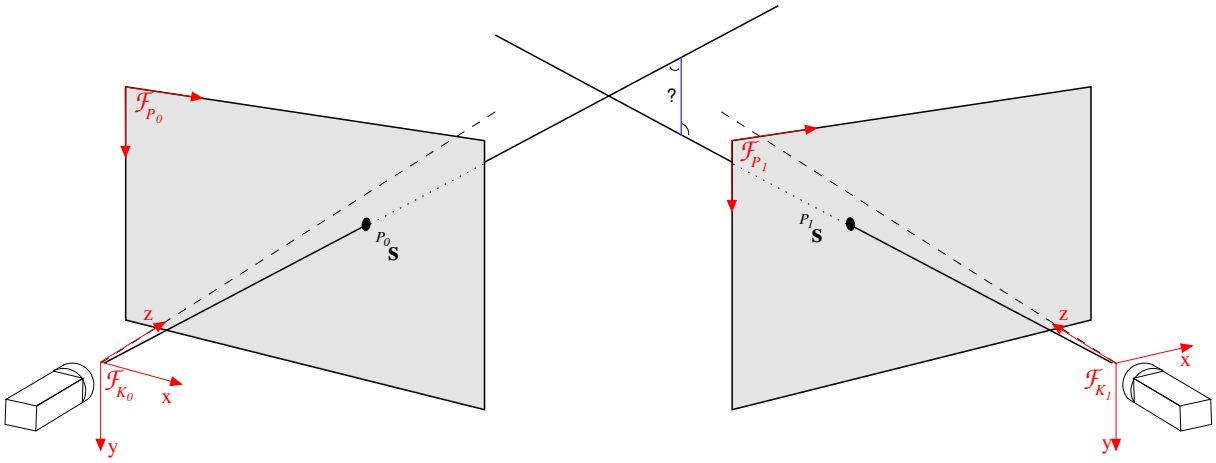


Abbildung 3.7: Problem der windschiefen Strahlen bei der Rekonstruktion.

Die Rekonstruktion geht von bereits bestimmten Korrespondenzen aus und hat die Aufgabe, die dreidimensionale Position des entsprechenden Punktes zu bestimmen. Dabei handelt es sich um ein überbestimmtes Problem, da aus den 2D-Koordinaten ${}^{P_0}\mathbf{s}$ und ${}^{P_1}\mathbf{s}$ die 3D-Position ${}^{W}\mathbf{s}$ bestimmt wird. Veranschaulicht man sich (vgl. Abb. 3.7), dass die Strahlen $\overline{{}^{P_0}\mathbf{s}^{K_0}\mathbf{o}}$ und $\overline{{}^{P_1}\mathbf{s}^{K_1}\mathbf{o}}$ in den meisten Fällen windschief sein werden, wird dies

klar. Hat die Projektionsmatrix die Struktur wie in Gleichung 3.14, führt die Lösung des überbestimmten Gleichungssystems

$$\mathbf{A}\mathbf{s} = \mathbf{y} \tag{3.24}$$

auf die 3D-Position des Punktes $\mathbf{s}$. Dabei gelten folgende Ersetzungen:

$$\mathbf{A} = \begin{pmatrix} (\mathbf{a}_1 - {}^{p_0}s_x\,\mathbf{a}_3)^\mathsf{T} \\ (\mathbf{a}_2 - {}^{p_0}s_y\,\mathbf{a}_3)^\mathsf{T} \\ (\mathbf{b}_1 - {}^{p_1}s_x\,\mathbf{b}_3)^\mathsf{T} \\ (\mathbf{b}_2 - {}^{p_1}s_y\,\mathbf{b}_3)^\mathsf{T} \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} -a_{14} + {}^{p_0}s_x\,a_{34} \\ -a_{24} + {}^{p_0}s_y\,a_{34} \\ -b_{14} + {}^{p_1}s_x\,a_{34} \\ -b_{24} + {}^{p_1}s_y\,a_{34} \end{pmatrix} \tag{3.25}$$

$${}^{P_0}\mathbf{s} = \begin{pmatrix} {}^{P_0}s_x \\ {}^{P_0}s_y \end{pmatrix} \qquad {}^{P_1}\mathbf{s} = \begin{pmatrix} {}^{P_1}s_x \\ {}^{P_1}s_y \end{pmatrix} \tag{3.26}$$

Bei rektifizierten Systemen lässt sich diese Rekonstruktion jedoch auf den Strahlensatz zurückführen und damit deutlich vereinfachen. Natürlich ist das Grundproblem der windschiefen Geraden auch im Falle rektifizierter Systeme vorhanden, da aber die y-Koordinate korrespondierender Punkte per Definition identisch ist, entsteht kein überbestimmtes Gleichungssystem.

Disparität

Zunächst muss an dieser Stelle der nützliche Begriff der Disparität eingeführt werden. Wie oben dargestellt, befinden sich die Abbildungen ${}^{p_0}\mathbf{s}$ und ${}^{p_1}\mathbf{s}$ des Punktes ${}^{W}\mathbf{s}$ in den rektifizierten Bildern auf derselben Bildzeile. Das bedeutet, dass die Entfernung des Punktes ${}^{W}\mathbf{s}$ vom Kamerasystem indirekt proportional zu dem Unterschied der x-Komponenten der Vektoren ${}^{p_0}\mathbf{s}$ und ${}^{p_1}\mathbf{s}$ sein muss. Dieser Unterschied wird Disparität genannt und meist in Pixeln im Koordinatensystem $\mathcal{F}_p$ angegeben. Für die Disparität findet sich je nach Quelle eine positive oder negative Definition. Im Rahmen dieser Arbeit sei sie mit

$$d = {}^{p_0}s_x - {}^{p_1}s_x \tag{3.27}$$

als positiv definiert.

Die Definition der Disparität im allgemeinen, also nicht rektifizierten Fall ist sehr viel komplexer. Da sie jedoch in dieser Arbeit keine Rolle spielt, sei hier auf das Grundlagenwerk von Faugeras verwiesen [19].

Rekonstruktion mit Hilfe des Strahlensatzes

Betrachtet man zunächst nur die Komponenten in Richtung der rektifizierten Basis, so lässt sich mit einer Basisbreite von g und einer Disparität von d die Entfernung des Objektes entlang der optischen Achse der rektifizierten Systeme basierend auf dem Strahlensatz zu

$${}^{k_0}s_z = f\frac{g}{d} \tag{3.28}$$

bestimmen. Dabei gilt

$$\begin{pmatrix} g \\ 0 \\ 0 \end{pmatrix} = {}^{k_0}\mathbf{o} - {}^{k_0}\mathbf{T}_{k_1}\,{}^{k_1}\mathbf{o} \tag{3.29}$$

$$\text{und} \quad d = {}^{p_0}s_x - {}^{p_1}s_x \tag{3.30}$$

Damit lässt sich nun ${}^{k_0}\mathbf{s}$ bestimmen:

$${}^{k_0}\mathbf{s} = {}^{k_0}s_z \cdot \begin{pmatrix} {}^{b_0}s_x \\ {}^{b_0}s_y \\ 1 \end{pmatrix} = {}^{k_0}s_z \cdot \begin{pmatrix} \frac{{}^{p_0}s_x - cx}{fx} \\ \frac{{}^{p_0}s_y - cy}{fy} \\ 1 \end{pmatrix} \tag{3.31}$$

Um diese sehr einfache Rekonstruktion nutzen zu können, müssen die Parameter der rektifizierten Kamera bekannt sein.

Kameraparameter der rektifizierten Kamera

Die Rektifizierung bringt grundsätzlich eine Drehung der rektifizierten Kamera ($\mathcal{F}_k$) gegenüber der realen ($\mathcal{F}_K$) mit sich. Daher spielt zwar für die Rekonstruktion nur der intrinsische Anteil der rektifizierenden Projektionsmatrix eine wichtige Rolle, aber da die rekonstruierten Punkte im System $\mathcal{F}_{k_0}$ vorliegen, ist der extrinsische Anteil wichtig, um die resultierenden 3D-Daten von $\mathcal{F}_{k_0}$ in das System $\mathcal{F}_{K_0}$ umzurechnen.

Aus einer Projektionsmatrix lassen sich die intrinsischen und extrinsischen Parameter der Kamera auf einfache Weise berechnen [85]. Gleichung 3.32 zeigt die Bestandteile der Projektionsmatrix.

$$\mathbf{P} = \begin{bmatrix} -f_x r_{11} + c_x r_{31} & -f_x r_{12} + c_x r_{32} & -f_x r_{13} + c_x r_{33} & -f_x t_x + c_x t_z \\ -f_y r_{21} + c_y r_{31} & -f_y r_{22} + c_y r_{32} & -f_y r_{23} + c_y r_{33} & -f_y t_y + c_y t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \tag{3.32}$$

$$\text{mit} \qquad f_x = f/s_x \qquad f_y = f/s_y \tag{3.33}$$

Die intrinsischen Parameter c_x, c_y, f_x, f_y korrespondieren mit den in Kapitel 3.1.2 verwendeten wie in Gleichung 3.33 dargestellt.

3.2.6 Extrinsische Kalibrierung, Kamera zu Kamera

Die externe Kalibrierung des Stereosystems ist Voraussetzung für die metrische Rekonstruktion der Szene. Da bei Beginn dieser Arbeit keine Softwarelösung für dieses Problem zur Verfügung stand, musste aus existierenden Komponenten ein solches System entwickelt werden. Obwohl heute diverse Softwarepakete für diesen Zweck zur Verfügung stehen, soll das entwickelte Verfahren hier kurz vorgestellt werden, da es für bestimmte Anwendungen Vorteile bietet.

Wie im Abschnitt 3.2.3 erläutert, kann die reine 2D-Beziehung zweier Kameras in Form der Fundamentalmatrix oder Essentiellen Matrix dargestellt werden. Gelingt es, diese Matrix aus Punktkorrespondenzen zu schätzen, kann mit Hilfe der internen Kameraparameter eine externe Transformationsmatrix zwischen den Kameras bis auf einen Skalierungsfaktor bestimmt werden. Die skalierten Projektionsmatrizen lassen sich bei Kenntnis der 3D-Position eines einzigen Punktes im Bild zurück in den euklidischen Raum skalieren. Dieses Verfahren benötigt also keinen Kalibrierkörper, sondern nur eine Szene mit vielen und möglichst großen Tiefensprüngen und deutlicher Struktur.

Die praktische Vorgehensweise gestaltet sich wie folgt:

1. Kalibrierung der internen Parameter beider Kameras mit herkömmlichen Methoden
2. Aufnahme von beliebigen Stereobildern mit möglichst großer Entfernungsdynamik in der Blickrichtung und viel Struktur.
3. Extraktion von Punktkorrespondenzen aus einem oder mehreren Bildpaaren
4. Schätzung der Fundamentalmatrix
5. Berechnung der normierten externen Parameter ${}^{K_0}\mathbf{T'}_{K_1}$ aus $\mathbf{F}$, ${}^{P_0}\mathbf{P}_{K_0}$ und ${}^{P_1}\mathbf{P}_{K_1}$
6. Aufnahme eines Bildes eines einfachen Kalibrierkörpers (z. B. Meterstab)
7. Skalierung von ${}^{K_0}\mathbf{T'}_{K_1}$ nach ${}^{K_0}\mathbf{T}_{K_1}$

Schritte 1 und 6 wurden mit dem Softwarepaket *HALCON* implementiert und der Kernalgorithmus wurde von C. Sun [84] beschrieben und öffentlich bereitgestellt. Das numerische Mathematiksystem *Matlab* dient als Programmierumgebung für die Schnittstellen dieser Einzelschritte.

Wie oben kurz angesprochen, sind mittlerweile geschlossene Systeme zur externen Kalibrierung verfügbar. So kann *HALCON* mit Hilfe des Kalibrierkörpers seit Version 7.0 auch Stereosysteme extern kalibrieren und das *Small Vision System* verfügt ebenfalls über eine externe Kalibrierung. Da beide Systeme jedoch auf dem Einsatz eines komplexen Kalibrierkörpers beruhen, sind sie für spezielle Anwendungen wie zum Beispiel sehr große Basisweiten nicht nutzbar.

3.3 Verfahren der Tiefenrekonstruktion aus Kamerabildern

Die große Anzahl verschiedener Ansätze zur Tiefenrekonstruktion erschwert in diesem Bereich eine vollständige Darstellung aller Verfahren. Dies resultiert sicher aus dem großen Interesse, das dem Forschungsgebiet bereits seit langer Zeit entgegengebracht wird [41]. Dabei inspiriert sowohl das Vorbild Natur als auch das große technische Potential, das in der Erfassung der Raumgeometrie durch Sensoren liegt. Gleichzeitig sind Anwendungen so unterschiedlich in ihren Anforderungen, dass fast zwangsläufig für jedes Problem eine

neue Variante eines Algorithmus existiert, die besser funktioniert als bisherige Verfahren. So kann beispielsweise ein Algorithmus zur Rekonstruktion von Höhenprofilen aus Luftbildern von Kontinuität, Texturierung und Reflexionsarmut ausgehen, wogegen bei der Vermessung von Innenräumen vollkommen gegensätzliche Bedingungen herrschen.

Eine grundlegende Unterteilung passiver Verfahren lässt sich nach **merkmalsbasierten Verfahren**, **flächenbasierten Verfahren** und **phasenbasierten Verfahren** vornehmen. Dem stehen **aktive Stereoverfahren** gegenüber, bei denen kontrollierte Beleuchtung der Szene zu ihrer Rekonstruktion genutzt wird.

Diese Verfahren sollen nun jeweils kurz erläutert werden, wobei der Schwerpunkt klar im Bereich der korrelationsbasierten Verfahren liegt, da sie für diese Arbeit von größerem Interesse sind.

3.3.1 Aktive Verfahren

Aktive Verfahren nutzen kontrollierte Beleuchtung, um für eine statische Szene aus mehreren Bildern mit systematisch veränderter Beleuchtung die Entfernungen zu bestimmen. Das einfachste zugrundeliegende Prinzip ist das der Triangulation: Kamera, Objekt und Lichtquelle bilden ein Dreieck. Ist die Position des Lichtpunktes und die Lage der Kamera und der Beleuchtung bekannt, kann die dreidimensionale Objektposition an der Stelle des Lichtpunktes bestimmt werden. Dieses Prinzip kann auf eine Beleuchtungslinie erweitert werden, wenn die Linie senkrecht zur optischen Achse der Kamera und zur Basis des Systems ist. Dieses Verfahren ist seit langem bekannt [67] und wird wegen seiner einfachen Anwendbarkeit und Robustheit auch im industriellen Umfeld eingesetzt [1]. Wird das Lichtmuster auf mehrere Linien erweitert, sinkt die Anzahl der notwendigen Aufnahmen für eine komplette Rekonstruktion [77]. In einer Weiterentwicklung dieses Prinzips wird die Szene durch Hell-Dunkel-Muster im *gray code* in immer feineren Auflösungsstufen beleuchtet. So lässt sich die Anzahl der erforderlichen Aufnahmen weiter reduzieren [77]. Wird die Szenenbeleuchtung um Grauwerte erweitert, lassen sich damit Quantisierungsfehler reduzieren und die Anzahl erforderlicher Aufnahmen noch weiter einschränken. Dieses *Phasenshiftverfahren* nutzt eine Lichtquelle mit sinusförmigem Helligkeitsverlauf und eine kontrollierte Phasen- und Frequenzverschiebung zwischen Aufnahmen zur Rekonstruktion.

Im Gegensatz zu diesen Verfahren, die explizit mit der Struktur der Beleuchtung arbeiten, gibt es auch den Ansatz, die Unterscheidbarkeit ansonsten gleichförmiger Bereiche zu verbessern, indem zufällige Lichtmuster projiziert werden. So verwendet das System von Kang [47] vier Kameras in Verbindung mit korrelationsbasierten Stereoverfahren und strukturierter Beleuchtung.

Derzeit entstehende Lösungen für integrierte 3D-Kameras – also Kameras, die für jedes Pixel bereits eine Entfernung messen – basieren ebenfalls auf aktiver Beleuchtung der Szene. Dabei handelt es sich um Laufzeitmessungen durch spezielle Sensoren bei modulierter Beleuchtung [6]. Diese Methode ist eher vergleichbar mit Entfernungsbestimmung

durch einen modulierten Laserstrahl, der die Szene durch eine Ablenkeinheit abtastet. Die Entfernungsmessung durch einen Laserstrahl spielt derzeit wohl die wichtigste Rolle unter allen Verfahren zur optischen Rekonstruktion.

Ein weiterer monokularer Ansatz besteht darin, die Reflexionsrichtung des Lichtes an Oberflächen als Indikator für die Szenengeometrie zu nutzen. Dabei wird eine Szene nacheinander mit natürlicher und gerichteter Beleuchtung aufgenommen, um daraus die Oberflächengeometrie zu bestimmen [39]. Eine ähnliche Vorgehensweise macht sich die Schatten in einer Szene als Hinweise auf die Oberflächengeometrie zu Nutze [38].

Da aktive Verfahren durch die notwendige Beleuchtung nur bei geringen bis mittleren Distanzen einsetzbar sind, wurden sie im Kontext der Szenenrekonstruktion für die Telepräsenz nicht weiter verfolgt. Nichtsdestotrotz wurden Versuche mit zufälliger Beleuchtung unternommen, um ähnlich wie Kang et al. [47] die Unterscheidbarkeit schwach texturierter Bereiche zu verbessern. Ein weiterer Grund gegen den Einsatz von aktiver Beleuchtung ist die schlechte Skalierbarkeit der Systeme. So ist die Rekonstruktion durch strukturiertes Licht für mikroskopische Szenen durch die beengten Platzverhältnisse kaum durchführbar.

3.3.2 Merkmalsbasierte Verfahren

Die Klasse der merkmalsbasierten Verfahren geht von einer Abstraktion der Bilder zu Merkmalen aus, die von unterschiedlichster Art sein können. Ihre einzig notwendige Eigenschaft besteht in ihrer Robustheit gegenüber den zu erwartenden Unterschieden in den beiden Stereoaufnahmen. Zwei Schritte müssen unterschieden werden: Die Merkmalsextraktion und die Suche nach korrespondierenden Merkmalen.

Merkmalsextraktion Nutzbare Merkmale ergeben sich normalerweise unmittelbar aus der geometrischen Struktur der Szene [50]. Als symbolische Beschreibung kommen im einfachsten Fall Kanten [3, 8] zum Einsatz. Sie werden durch ihren Winkel, ihren Schwerpunkt und ihre Länge charakterisiert. Dabei sind Kanten, die parallel zur Kamerabasis verlaufen, nicht nutzbar. Um die Korrespondenzsuche zu vereinfachen, werden häufig weitere Eigenschaften der Kanten hinzugenommen. So nennt Faugeras [19] hier den Intensitätsverlauf senkrecht zu einer Kante und den mittleren Helligkeitsgradienten einer Kante.

Weitet man die Beschreibung eines Bildes durch Kanten auch auf Konturen aus, müssen diese durch passende Beschreibungen approximiert werden. Hier kommen Polygone höherer Ordnung und freie Kurven wie BSplines oder Kettencodes zum Einsatz.

Ein weiteres häufig genutztes Merkmal sind Ecken, die jedoch unterschiedlich definiert sein können. So definiert Moravec [65] eine Ecke über bestimmte Eigenschaften der lokalen Autokorrelation [64]. Der viel genutzte *Harris corner detector* [35] nutzt hier den lokalen zweidimensionalen Bildgradienten. Im Gegensatz dazu werden bei Freeman et al. [24] oder Medioni et al. [62] Ecken als Punkte hoher Krümmung einer Kontur interpretiert.

Korrespondenzsuche Die Suche nach korrespondierenden Merkmalen wird in vielen Fällen durch Zwangsbedingungen *(constraints)* eingeschränkt. So kann jedem Merkmal nur maximal ein korrespondierendes Merkmal zugeordnet werden *(uniqueness constraint)*. Weiterhin muss die relative Reihenfolge der Merkmale in beiden Bildern identisch sein *(ordering constraint)*. Dies gilt jedoch nur für bestimmte Szenen (z. B. terrestrische Luftaufnahmen) und bestimmte Geometrie des Stereosystems (Basisweite, Brennweite, Vergenzwinkel). Natürlich wird auch die Epipolarbedingung zur Einschränkung des Suchraumes genutzt. Merkmalsbasierte Ansätze führen prinzipbedingt zu 3D-Information nur am Ort eines Merkmals. Es ist also nicht möglich, dichte Tiefenkarten zu erzeugen. Allerdings werden zum Beispiel von Baker [3] ausgehend von den gefundenen Kantenkorrespondenzen in einem zweiten Schritt die Bildintensitäten für eine Korrespondenzsuche herangezogen, um so dichte Tiefenkarten zu erzeugen.

3.3.3 Phasenbasierte Verfahren

Bei phasenbasierten Stereoverfahren werden für beide Aufnahmen zunächst die Ortsfrequenzen durch Faltung mit lokalen Bandpassfiltern bestimmt. Ohne Suche nach korrespondierenden Punkten kann nun unter Ausnutzung der Eigenschaften der Fouriertransformation direkt die Phasenverschiebung im Frequenzraum als Maß für die Disparität im Bild genutzt werden [74]. Für die Filterung wird meist ein Gaborfilter eingesetzt. Diese Wahl wird durch Untersuchungen am visuellen Kortex bestätigt, da die dortigen Zellen ähnliche Strukturen wie zweidimensionale Gaborfilter aufweisen [43].

Seit diese Methode 1988 von Sanger beschrieben wurde [74], hat sie viel Beachtung in verschiedenen Anwendungsbereichen gefunden (z. B. [25]). Sie ist sehr robust gegenüber Kontrast- und Helligkeitsunterschieden zwischen den Bildern, zeigt aber eine gewissse Empfindlichkeit für Bildrauschen [11] und insbesondere für große Disparitätssprünge [26]. Zuverlässige Entfernungsmaße sind mit diesem Verfahren nur in gut strukturierten Bereichen des Bildes möglich, da nur hier höhere Ortsfrequenzen eine präzise Bestimmung der Phase ermöglichen. Zusätzlich können bei der initialen Faltung Singularitäten auftreten und damit eine weitere Berechnung verhindern.

3.3.4 Flächenbasierte Verfahren

Flächenbasierte Verfahren gehen von der Beobachtung aus, dass sich die Bilder der beiden Kameras grundsätzlich ähneln. Je geringer dabei die Auflösung der Bilder ist, desto größer ist ihre Ähnlichkeit, da eine gegebene Disparität bei geringerer Auflösung einen proportional geringeren Effekt auf das Ergebnis hat. Werden die Ansichten in immer kleinere lokale Regionen unterteilt, werden korrespondierende Regionen sich immer ähnlicher. Das bedeutet, dass durch räumliche Quantisierung in kleine Blöcke unter Berücksichtigung eines photometrischen **Ähnlichkeitsmaßes** korrespondierende Regionen gefunden werden können. Unterschiedliche Verfahren unterscheiden sich hauptsächlich in der Wahl eines Ähnlichkeitsmaßes, in der Strategie bei der **Suche nach den ähnlichsten Regionen**

und der **Form der Regionen**. Das Ergebnis ist üblicherweise eine dichte Disparitätskarte mit einem Disparitätswert pro Pixel. Normalerweise wird dabei von unendlich entfernter Beleuchtung und lambertinischen Oberflächen ausgegangen. Gleichzeitig sollten die Ansichten möglichst ähnlich sein, damit die unbekannten perspektivischen Verzerrungen innerhalb einzelner Regionen vernachlässigbar sind [50].

Der gelungene Überblick und Vergleich verschiedener flächenbasierter Stereoverfahren von Scharstein, Szeliski und Zabih aus dem Jahr 2001 [76] unterteilt die übliche Vorgehensweise zur Berechnung einer Disparitätskarte in

1. Ähnlichkeitsmaß und Kostenfunktion *(matching cost computation)*
2. Kostenaggregation *(aggregation of cost)*
3. Berechnung der Disparität *(disparity computation and optimization)*
4. Verfeinerung der Disparität *(refinement of disparities)*

Da diese Einteilung eine Taxonomie der flächenbasierten Verfahren ermöglicht, soll sie im weiteren Verlauf dieses Abschnitts als Gliederung dienen.

Ähnlichkeitsmaß und Kostenfunktion *(matching cost computation)*

Ziel einer Korrespondenzsuche ist die Identifikation korrespondierender Punkte in beiden Aufnahmen, also Punkte, die die Projektion desselben 3D-Punktes darstellen. Für den Vergleich von Regionen muss daher ein Ähnlichkeitsmaß definiert werden. Hier existieren eine Reihe von Standardmaßen, die vielfach verwendet werden. Im einfachsten Fall wird die Summe absoluter Differenzen (SAD) oder die Summe quadratischer Differenzen eingesetzt. Höherer Berechnungsaufwand ergibt sich beim Einsatz der normalisierten Kreuzkorrelation (NCC), die den flächenbasierten Verfahren auch den den etwas irreführenden Namen *Korrelationsbasierte Verfahren* eingebracht hat. Sind I_L und I_R die beiden Aufnahmen eines Stereopaares und d die Disparität, so lassen sich die gängigen Ähnlichkeitsmaße wie folgt definieren:

$$SSD(d) = \sum_i \sum_j I_L(x+d+i, y+j) - I_R(x+i, y+j)^2 \tag{3.34}$$

$$SAD(d) = \sum_i \sum_j |I_L(x+d+i, y+j) - I_R(x+i, y+j)| \tag{3.35}$$

$$NCC(d) = \frac{C(I_L, I_R) - \sum_i \sum_j \mu_L \mu_R}{\sum_i \sum_j \sigma_L \sigma_R} \tag{3.36}$$

Dabei repräsentieren μ_L und μ_R den Intensitätsmittelwert der Fenster in den beiden Bildern (vgl. 3.37) und σ_L und σ_R die Standardabweichung gemäß Gleichung 3.38. Gleichung 3.39 ist die Kreuzkorrelation zwischen den beiden Fenstern.

$$\mu = \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} I(x+i, y+j) \tag{3.37}$$

$$\sigma = \sqrt{\frac{1}{n \cdot m - 1} \left[\sum_{i=1}^{n} \sum_{j=1}^{m} (I(x+i, y+j) - \mu)^2 \right]} \tag{3.38}$$

$$C(I_L, I_R) = \sum_{i} \sum_{j} I_L(x+d+i, y+j) \cdot I_R(x+i, y+j) \tag{3.39}$$

Jenseits dieser Standardmaße existiert eine Reihe speziellerer Metriken. Nichtparametrische Verfahren wie *Rank* und *Census* [89] eigenen sich gut für eine Implementierung in Hardware [9] und sind robust gegenüber Unterschieden bei der Bildaufnahme in beiden Kameras. In dieselbe Richtung zielen gradientenbasierte Maße oder die Vorfilterung der Bilder durch Histogrammausgleich oder den *„laplacian-of-gaussian“*-Filter.

Kostenaggregation *(aggregation of cost)*

Würden die Kosten bzw. das Ähnlichkeitsmaß lokal aus einzelnen Pixeln bestimmt, so wäre zwar theoretisch höchste Ortsauflösung möglich, dabei würde der Merkmalsraum, in dem die Pixel verglichen werden, jedoch typischerweise nur 256 Quantisierungsstufen umfassen. Da damit eine Identifikation von korrespondierenden Punkten nur sehr schlecht gelingt, wird in allen Algorithmen eine Aggregation der Kosten bzw. des Ähnlichkeitsmaßes entweder implizit oder explizit vorgenommen.

Die oben genannten Metriken nehmen diese Aggregation bereits implizit über ihre Fenstergröße (i,j) vor. Sie gehen damit aber von frontoparallelen Disparitätsebenen aus, also von der Annahme, dass für jeden Punkt in einem Fenster dieselbe Disparität gilt. Alternative Ansätze nutzen statt einem festen Fenster ein gaußsches Fenster mit ortsabhängig unterschiedlichen Multiplikatoren. Um dem Problem der gleichbleibenden Disparität innerhalb des Fensters Rechnung zu tragen, kann der Ankerpunkt der Fenster, das heißt die Lage des interessierenden Pixels, relativ zum restlichen Fenster verschoben werden. Tiefensprünge lassen sich mit diesen *asymmetrischen Korrelationsfenstern* besser auflösen [5]. Kapitel 3.4.5 geht auf diese Verfahren genauer ein. Ebenfalls mit dem Ziel, die Ortsauflösung der Disparitätskarten zu steigern, wird häufig mit variablen Fenstergrößen gearbeitet [45, 86, 37].

Die Fensterung kann natürlich auch im x-y-d Raum erfolgen und dabei die Annahme frontoparalleler Ebenen aufgeben. Damit sind Tiefenveränderungen auch innerhalb eines Fensters zulässig [30, 71].

Berechnung der Dispartität *(disparity computation and optimization)*

Aufgrund der Tatsache, dass es sich bei der Korrespondenzsuche um eine inverse perspektivische Abbildung und damit um ein schlecht gestelltes Problem *(ill-posed problem)* handelt, sind mit einer einfachen Suche nach dem Maximum der Korrelation bzw. den minimalen Kosten keine optimalen Ergebnisse erreichbar.

Dies zeigt sich beispielsweise bei Aufnahmen repetitiver Strukturen, bei denen es leicht zu Fehlkorrespondenzen kommt. Der übliche Lösungsansatz ist es, weitere Einschränkungen (*constraints*)[1)] hinzuzunehmen um die Lösung einzugrenzen. Die folgenden Einschränkungen werden häufig implizit oder explizit verwendet (z. B. [17, 18, 19]):

- **Epipolarbedingung** *(epipolar constraint)*
  Die Epipolargeometrie in einem Stereosystem kann auch als Bedingung interpretiert werden, da sie den Lösungsraum auf ein eindimensionales Problem einschränkt. Sie erlaubt jedoch keine weitere Suchraumreduktion und ist daher eher als Voraussetzung zu sehen, denn als Zusatzbedingung für die Einschränkung der Lösung. Obwohl sie häufig zusammen mit den folgenden Einschränkungen genannt wird, kommt ihr genau genommen eine Sonderrolle zu.

- **Eindeutigkeitsbedingung** *(uniqueness constraint)*
  Jedes Pixel in einer Aufnahme besitzt maximal ein korrespondierendes Pixel in der Aufnahme der zweiten Kamera. Diese Bedingung drückt eigentlich die weiter oben bereits gemachte Voraussetzung aus, dass die Szene keine transparenten Objekte enthält und dass keine Reflexionen existieren [58, 59, 90].

- **Disparitätsgradientenbeschränkung** *(disparity gradient limit)*
  Bedingt durch den Aufbau eines Stereosystems darf der Gradient der Disparität für die linke Aufnahme -1 nicht unterschreiten und für die rechte Aufnahme +1 nicht überschreiten. Dies ergibt sich aus der Tatsache, dass jeder Tiefensprung in der Szene in der Disparitätskarte entweder als Tiefensprung oder als Verdeckung in Erscheinung tritt. Eine genauere Erklärung dieses Phänomens wird in Abschnitt 3.4.3 gegeben. Die Disparitätsgradientenlimits lassen sich wie folgt darstellen:

$$\begin{aligned} -1 < \frac{\partial d_h}{\partial d} \leq +\infty \quad & - \text{ linke Disparitätskarte} \\ -\infty < \frac{\partial d_h}{\partial d} \leq +1 \quad & - \text{ rechte Disparitätskarte} \end{aligned} \tag{3.40}$$

[1] In der englischsprachigen Fachliteratur haben sich feste Namen für die einzelnen Bedingungen etabliert. Um Unklarheiten durch die Übersetzung zu vermeiden, sollen sie hier jeweils genannt werden

- **Reihenfolgebedingung** *(monotonic ordering constraint)*
  Die Reihenfolge der Korrespondenzen muss in beiden Aufnahmen gleich sein. Diese Einschränkung ist allerdings nicht für beliebige Szenen gültig, wie Abbildung 3.8b veranschaulicht. Wenn diese Bedingung durchgesetzt wird, können kleine Objekte im Vordergrund entweder nicht aufgelöst werden, oder erzeugen Fehlkorrespondenzen.
  Diese Einschränkung entspricht der vorangegangenen, indem sie den maximalen Disparitätsgradienten begrenzt.

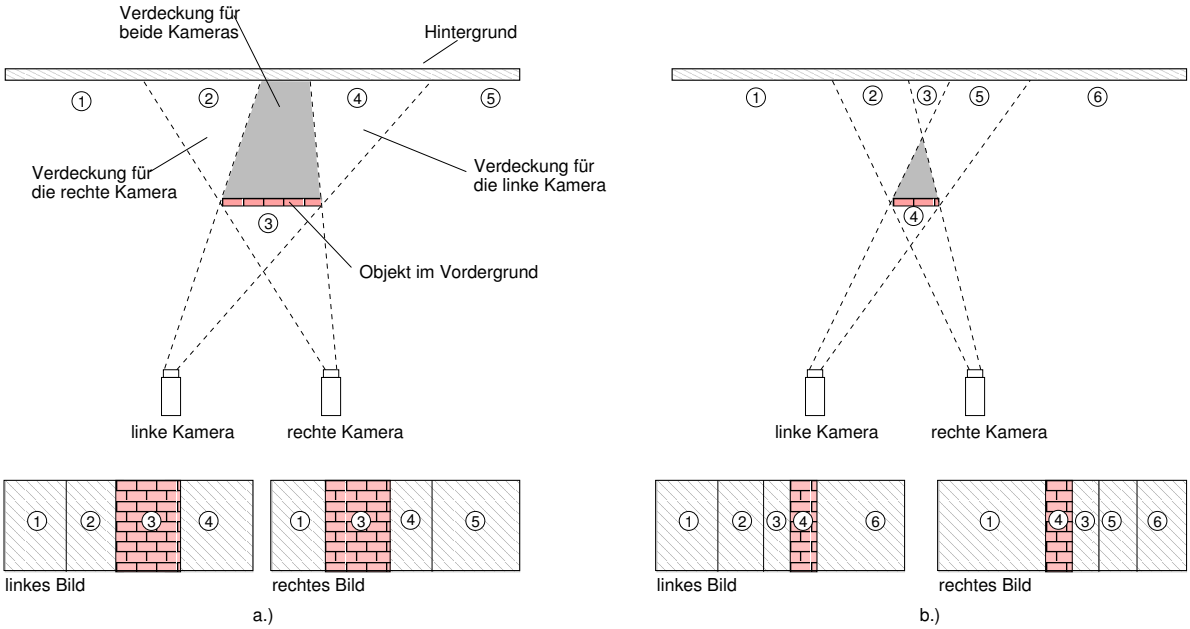


Abbildung 3.8: Effekt von Verdeckung in beiden Abbildungen bei großen (a) und kleinen (b) Objekten im Vordergrund.

- **Disparitäts-Ähnlichkeitsfunktion** *(disparity similarity function)*
  Prazdny formuliert in seiner Arbeit [71] eine Bedingung, die ebenfalls die „Gleichmäßigkeit“ der Szene betrifft. Er stellt eine lokale Unterstützungs-Funktion *(local support criterion)* auf, die einen möglichen Korrespondenzkandidaten bestätigt, wenn benachbarte Punkte eine ähnliche Disparität besitzen, aber keinen Einfluss hat, wenn ein Disparitätssprung vorliegt.
- **Luminanzbedingung** *(luminance constraint)*
  Die Korrespondenzsuche in den beiden Bildern wird durch das Kamerarauschen und den Helligkeitsabfall an den Objektivrändern beeinträchtigt. Abgesehen von Spiegelungen auf Oberflächen muss die lokale Helligkeitsvarianz σ^2 in korrespondierenden Blöcken größer als die durch Rauschen und Helligkeitsabfälle erzeugte Varianz σ^2_{cam} sein. Daher kann die Korrespondenzsuche auf Pixel mit hoher Helligkeitsvarianz σ^2 eingeschränkt werden.

$$\sigma^2 \gg \sigma^2_{cam} \tag{3.41}$$

- **Erweiterte Kontinuitätseinschränkung** *(extended continuity constraint)* Die Tatsache, dass sich viele Szenen aus stetigen Oberflächen zusammensetzen, die Tiefensprünge nur an ihren Begrenzungen aufweisen, wird in dieser Einschränkung ausgedrückt. Das bedeutet, dass eine szenenabhängige Wahrscheinlichkeit für Tiefensprünge festgelegt werden kann [17].

Die der Korrespondenzsuche zugrundeliegende Methode lässt sich als lokal oder global klassifizieren:

Lokale Korrespondenzsuche Hier reduziert sich die Korrespondenzsuche auf eine einfache Suche nach dem lokalen Maximum bzw. Minimum der Kosten für alle potentiellen Korrespondenzkandidaten eines Pixels. Demzufolge müssen potentielle Fehlkorrespondenzen bereits in den vorangehenden Schritten unterdrückt werden. Bei lokalen Verfahren wird folgerichtig mehr Aufwand in die Wahl des Ähnlichkeitsmaßes und der Kostenaggregation (Fenstergröße, und -form) investiert. Weiterhin kann Information über die lokale Nachbarschaft genutzt werden, da diese höchstwahrscheinlich zum selben Objekt gehört.

Globale Korrespondenzsuche Im Gegensatz dazu liegt bei globalen Methoden der Schwerpunkt bei der Berechnung der Disparität. Lokale Mehrdeutigkeiten werden durch die Berücksichtigung der Gesamtinformation aufgelöst. Dies wird häufig auf die Minimierung einer Gesamtenergie zurückgeführt [76]. Vorteilhaft ist die klare Formulierung des Korrespondenzproblems mit seinen Einschränkungen. Dies wird durch die durchweg hohe Komplexität globaler Verfahren erkauft.

Eine mögliche Realisierung stammt zum Beispiel von Cox et al., der das Problem als Bayes'sches Netzwerk definiert und Annahmen und Bedingungen in Form von Übergangswahrscheinlichkeiten formuliert [10].

Bei der viel genutzten Dynamischen Programmierung handelt es sich ebenfalls um eine globale Lösung, obwohl es sich hier nur um 'zeilenweise Globalität' handelt. Es werden zeilenweise (auf rektifizierten Bildern) alle Kandidaten für eine mögliche Korrespondenz durch ein Ähnlichkeitsmaß verglichen, um dann einen Pfad der geringsten Kosten durch diese Matrix zu suchen. Dies gelingt auf der Basis der Reihenfolgebedingung (*ordering constraint*) die (für geeignete Szenen) garantiert, dass die Punktkorrespondenzen in beiden Bildern dieselbe Reihenfolge besitzen. Das Verfahren arbeitet zeilenweise unabhängig, was die Berücksichtigung von Kriterien wie der Disparitäts-Ähnlichkeitsfunktion oder der erweiterten Kontinuitätseinschränkung zwischen den Zeilen verhindert. Das Verfahren wird im Abschnitt 3.4.5 detailliert vorgestellt, da es in dieser Arbeit eingesetzt wurde.

Weitere globale Verfahren basieren auf der Darstellung des Problems als Graphen, dessen Knoten oder Kanten Werte zugeordnet werden, die iterativ verändert werden. So werden bei der Formulierung als Maximal-Fluss-Problem (*maximum flow*) die Kosten für eine Korrelation an der Position x-y-d als lokaler Fluss am Knoten x-y-d des Graphen dargestellt. Iterative Aktualisierung des Flusses im Graphen führt zur Stabilisierung einer Lösung [73, 72].

Kooperative Methoden orientieren sich am menschlichen Vorbild und gehören zu den ältesten Stereoverfahren [58]. Lokale nichtlineare Berechnungen werden iterativ ausgeführt und führen so zu einem insgesamt globalen Verhalten. Für manche Verfahren kann die globale Minimierungsfunktion dargestellt werden, für viele jedoch nicht. Ein jüngeres kooperatives Verfahren von Zitnick und Kanade [90] wird in dieser Arbeit evaluiert und zu diesem Zweck im Abschnitt 3.4.6 genauer beschrieben.

Verfeinerung der Disparität *(refinement of disparities)*

Flächenbasierte Verfahren stellen mittels eines Ähnlichkeitsmaßes die Ähnlichkeit einer Region mit einer anderen Region fest. Da dies natürlich nicht nur für das letztendlich bestimmte Regionenpaar bestimmt wird, sondern auch in seiner unmittelbaren Nachbarschaft, kann diese Information gezielt benutzt werden, um den *genauen* Ort der Korrespondenz *subpixelgenau* zu bestimmen. Dazu wird eine Parabel in die Kosten am Ort der bestimmten Korrespondenz und seiner direkten Nachbarschaft eingepasst und ihr Minimum bestimmt. Der Ort des Minimums ist der genaue Punkt der Korrespondenz der beiden Regionen.

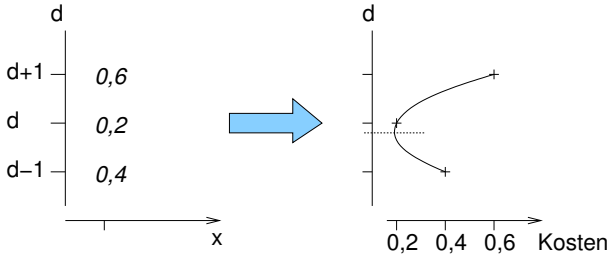


Abbildung 3.9: Kosten für eine Punktkorrespondenz an den Positionen $(x, d-1), (x, d)$ und $(x, d+1)$ und die eingepasste Parabel zur subpixelgenauen Bestimmung der Disparität.

3.4 Auswahl eines geeigneten Stereoverfahrens

Wie für alle Anwendungen der Szenenrekonstruktion gilt für die Telepräsenz das Ziel, die Szene mit bestmöglicher Qualität in einer kürzestmöglichen Zeit zu erfassen. Da die Beantwortung der Frage nach dem *besten* Stereoverfahren aber immer eine Abwägung zwischen Rechenzeit und Qualität bedeutet, sind eigene Tests verschiedener Verfahren unumgänglich. Dies bedeutet jedoch unter Umständen erheblichen und möglicherweise auch vergeblichen Implementierungsaufwand. Trotzdem ermöglicht erst eine eigene Implementierung die Abschätzung der Stärken und Schwächen verschiedener Verfahren. In jüngerer Zeit hat sich mit den *Middlebury Stereo Pages* ein gesamtes Forschungsprojekt dieser Problematik angenommen. Ein Web-basiertes Interface ermöglicht dort den Vergleich der Ergebnisse verschiedener Standardverfahren mit eigenem Datenmaterial. Diese Möglichkeit besteht jedoch erst seit relativ kurzer Zeit und so wurden im Rahmen dieser Arbeit ohne diesen praktischen Zugriff auf Standardverfahren drei repräsentative Verfahren gegeneinander getestet, um eines für die Anwendung weiterzuentwickeln.

Als Vertreter komplexerer Verfahren wurde das global optimierende in [90] beschriebene Verfahren von Zitnick und Kanade „A Cooperative Algorithm for Stereo Matching and Occlusion Detection“ implementiert. Ein durch den Einsatz dynamischer Programmierung zeilenweise optimierendes Verfahren wurde basierend auf [84] und [17, 18] aufgebaut. Als Vertreter einfacher Methoden mit lokaler Optimierung wurde schließlich mit dem **Small Vision System** ein kommerzielles System beschafft, das sogar schritthaltende Verarbeitung bei typischen Bildraten erreicht.

Dieser Abschnitt beschäftigt sich zunächst mit einigen Grundsatzentscheidungen, die im Rahmen dieser Arbeit getroffen werden. Um einen Vergleich der drei genannten Verfahren zu ermöglichen, werden Testaufnahmen mit ihren Eigenschaften vorgestellt und mit der Kostenmatrix eine grundlegende Datenstruktur eingeführt. Schließlich werden die drei Verfahren im Detail vorgestellt und an Testdaten gemessen, um in einer abschließenden Diskussion eine fundierte Entscheidung vorstellen zu können. Einige Überlegung zur effektiven Implementierung des ausgewählten Verfahrens schließen das Kapitel ab.

3.4.1 Grundsatzentscheidungen

Aktiv oder passiv? Aktive Rekonstruktionsverfahren erlauben meist eine präzise Rekonstruktion und sind weitgehend unabhängig vom Szeneninhalt. Gleichzeitig verfügen sie über begrenzte Reichweite, da sie immer auf Reflexion der aktiven Lichtquelle beruhen. Damit sind sie nicht frei skalierbar. Diese eher philosophische Überlegung hat die Arbeit der Robot-Vision-Gruppe des Lehrstuhls für Realzeit-Computersysteme der Technischen Universität München seit ihrem Entstehen geleitet. So wurde auch im Projekt „Übertragungszeitkompensation durch Szenenprädiktion“ im Rahmen des Sonderforschungsbereichs 453 auf den Einsatz aktiver Verfahren verzichtet. Diese Vorgehensweise hat sich rückblickend bewährt. So konnte das Telepräsenzsystem in einem Szenario der endoskopischen Chirurgie eingesetzt werden. Aktive Verfahren hätten sich nicht auf den Einsatz mit einem herkömmliche Laparoskop mit integrierter Beleuchtung durch Faserbündel anwenden lassen. Außerhalb des Einsatzes in der Telepräsenz konnten die Stereoalgorithmen auch auf der mobilen Roboterplattform Marvin im Rahmen des Projektes „Exploration von Innenräumen“ eingesetzt werden. Auch hier wäre ein aktives System schwieriger zu integrieren gewesen.

Merkmalsbasiert oder flächenbasiert? Diese Grundsatzentscheidung kann nicht eindeutig beantwortet werden. Vieles spricht bei dem anvisierten Einsatz in der Telepräsenz in der Tat für merkmalsbasierte Ansätze. So bringt die Reduktion des komplexen Bildinhalts auf Merkmale eine große Datenreduktion bereits auf niedriger Verarbeitungsebene mit sich. Eben diese Datenreduktion macht jedoch die Interpretation sehr viel komplexer. Innenräume als typisches Anwendungsszenario führen zusätzlich zu hohem Informationsgehalt auch einfacher Merkmale. So dominieren in menschengemachter Umgebung rechte Winkel, gerade Kanten, Ecken und Kreuzungspunkte. Damit können zwar einzelne Objekte, Polygone oder Flächen mit hoher Präzision rekonstruiert werden, jedoch keine *dichten*

Szenenmodelle. Dies resultiert aus der per definitionem nur lokal geschlossenen Szenenbeschreibung.

Im Gegensatz dazu kann auf der Basis eines flächenbasierten Verfahrens eine dichte Szenenrekonstruktion erstellt werden, wenn ausreichend Textur vorhanden ist. Da aber auf die frühe Datenreduktion verzichtet wird, entsteht sehr viel größeres Datenaufkommen und damit eine generell langsamere Lösung.

Trotzdem wurde in dieser Arbeit bewusst der flächenbasierte Ansatz verfolgt. Ein Grund dafür ist die laufend steigende Rechenleistung verfügbarer Systeme, die schritthaltende Verarbeitung auch bei flächenbasierten Systemen ermöglicht. Ein weiterer Grund ist die Tatsache, dass der Schritt von dichten Tiefenkarten zu einer dichten polygonalen Beschreibung kürzer und durch den Verzicht auf frühe Interpretation weniger fehleranfällig erscheint, als von Merkmalen zu Polygonen.

3.4.2 Testaufnahmen für Experimente

In der Fachliteratur für Stereorekonstruktion konnten sich einige rektifizierte Bildpaare aus unterschiedlichen Quellen [78] als Standard für den Vergleich von Algorithmen etablieren. Abbildung 3.10 zeigt die verwendeten Testbilder und gibt ihre relevanten Daten an. Dabei fällt auf, dass *Pentagon* und *Tsukuba* einen sehr geringen Disparitätsbereich und starke Texturierung besitzen. Dies kommt zwar vielen Algorithmen sehr entgegen, führt aber insbesondere bei besseren Verfahren zu kaum differenzierbaren Ergebnissen. Diese Problematik haben Scharstein und Szeliski erkannt und neue Testaufnahmen erzeugt und bereitgestellt. Aus dieser Serie stammt die Aufnahme *Cones*, die komplexere Geometrie, hohe Auflösung, einen großen Disparitätsbereich und eine subpixelgenaue und hochpräzise Disparitätenkarte bietet [1)].

Die Testbilder von Scharstein und Szeliski sind jedoch ebenfalls intensiv texturiert und daher keine geeigneten Testfälle für Algorithmen, die auch in typischen Innenräumen gute Leistung erbringen sollen. Daher werden im Rahmen dieser Arbeit Aufnahmen vom Laborraum des Lehrstuhls ebenfalls für Tests verwendet. Die Aufnahmen *Telekiste* und *Teletisch* stammen aus dieser Serie. Beide Aufnahmen umfassen Disparitäten von ca. 50 bis ca. 100. *Telekiste* ist fast überall schwach texturiert. *Teletisch* enthält diffuse Reflexionen (v. a. auf dem Rollcontainer) und ist kaum bis gar nicht texturiert. Im Kapitel 6 werden drei weitere Bildserien bestehend aus vielen Stereopaaren mit Kamerapositionen zum Test der Algorithmen genutzt.

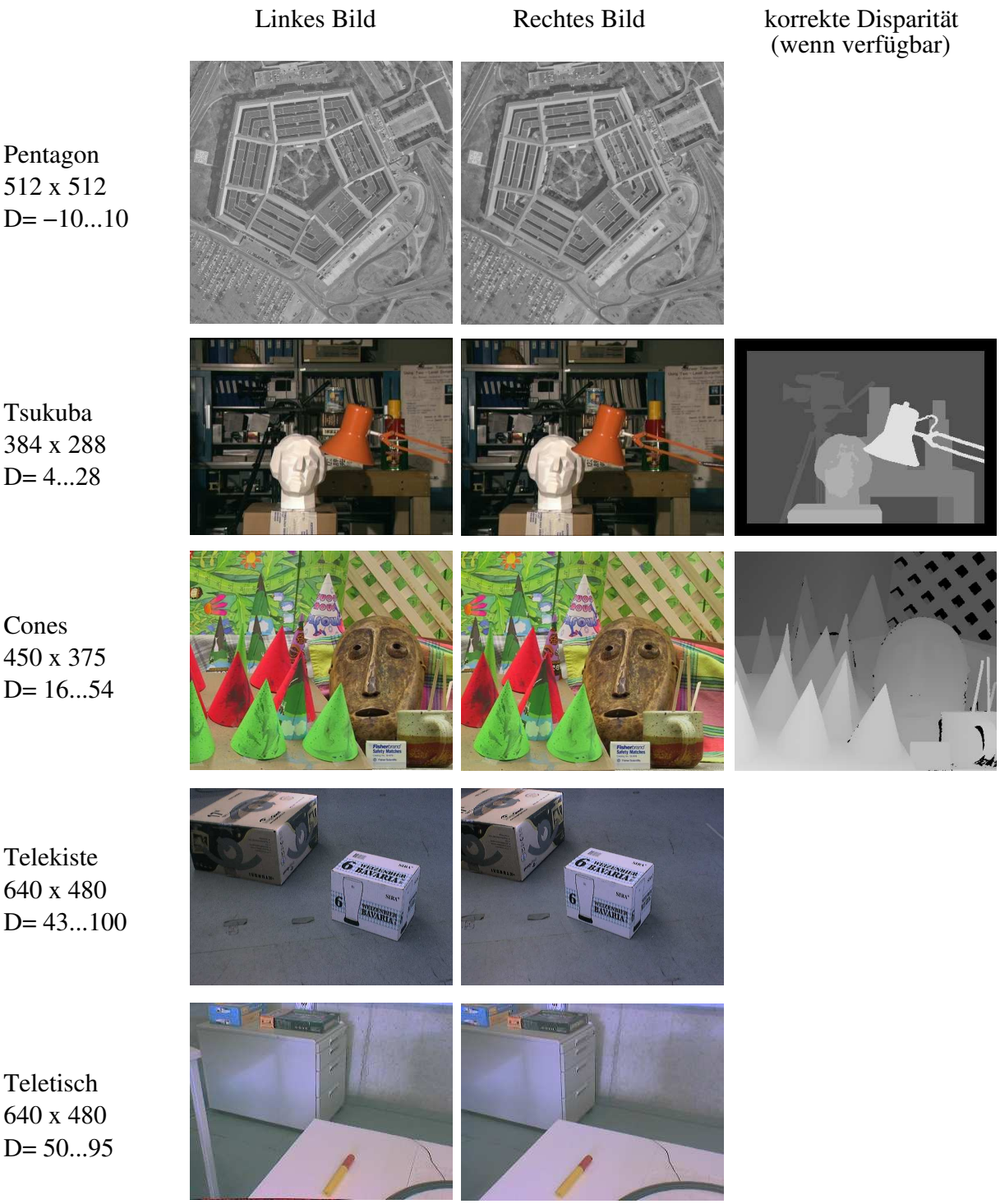


Abbildung 3.10: Testdaten für die Tiefenrekonstruktion.

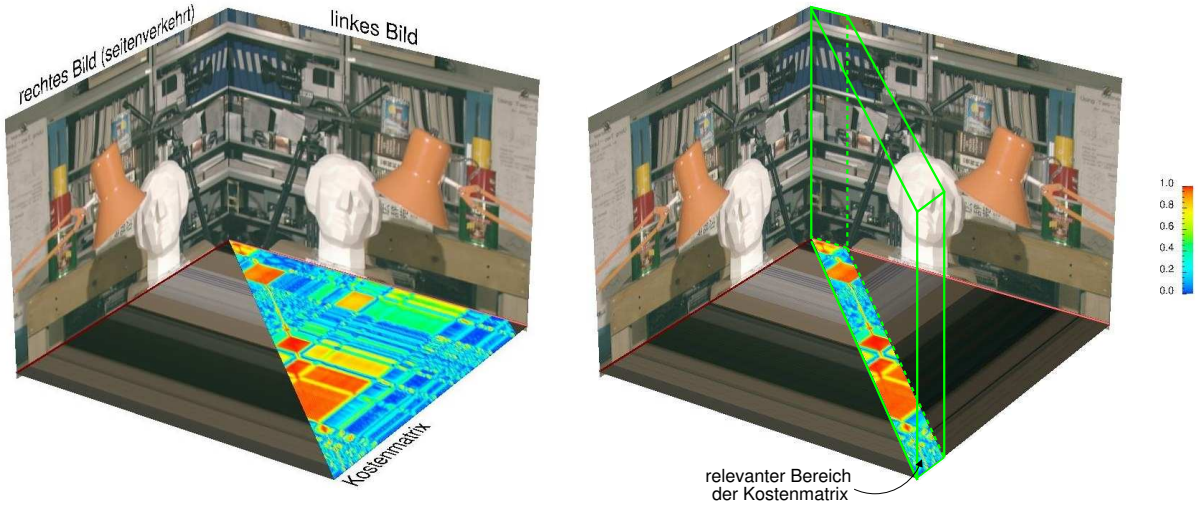


Abbildung 3.11: Die Entstehung der Kostenmatrix einer Bildzeile (links) und ihre Reduktion auf die relevanten Einträge (rechts).

3.4.3 Die Kostenmatrix

Für die im Kontext dieser Arbeit eingesetzten flächenbasierten Verfahren gibt es eine günstige Datenstruktur für die Ablage der Korrelationsergebnisse. In diesem Kapitel wird diese Struktur vorgestellt, da sich viele Abbildungen und Erklärungen auf diese Datenstruktur stützen.

Wie in Kapitel 3.3.4 bereits vorgestellt, werden die Aufnahmen nach einer eventuell notwendigen Vorfilterung durch Fensterung auf der Basis eines Kostenmaßes verglichen. Diese Zwischenergebnisse können in einer Kostenmatrix wie in Abbildung 3.11 links dargestellt, abgelegt werden. Diese Matrix besitzt zunächst die Dimensionen b x b x h. Da der Bereich gültiger Disparitäten durch die Rektifizierung und durch die Definition einer minimalen Objektdistanz jedoch immer im Intervall $[0, d_{max}]$ liegt, lässt sich die Information in der Kostenmatrix, wie in Abbildung 3.11 rechts dargestellt, auf die relevanten Einträge beschränken. Durch den Kollaps der Struktur auf die Dimensionen

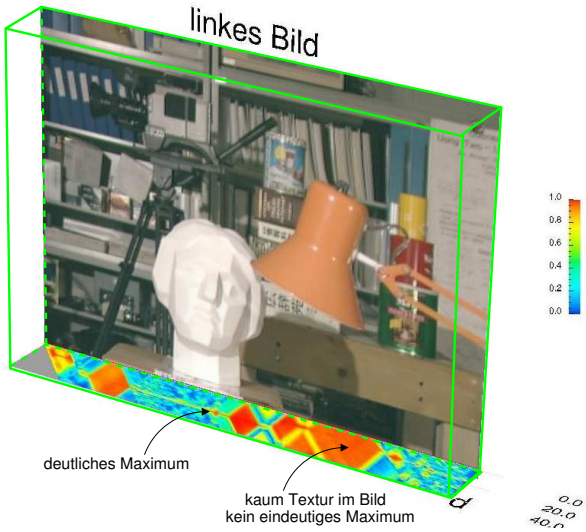


Abbildung 3.12: Reduzierte Version der Datenstruktur.

[1] Diese Vergleichsdaten werden im englischen Sprachraum als *ground truth* bezeichnet. Da es in deutscher Sprache keinen passenden Begriff gibt, soll dieser Fachausdruck im Rahmen dieser Arbeit verwendet werden.

b x h x d_{max} ergibt sich die Ansicht wie in Abbildung 3.12. Deutlich erkennbar sind bereits lokale Maxima der Kreuzkorrelation sowie Bereiche ohne Texturierung, die zu gleichförmigen Bereichen in der Kostenmatrix führen.

Trägt man nun in der Kostenmatrix den für diese Zeile gültigen Verlauf der Disparität ein, so ergibt sich eine Ansicht wie in Abbildung 3.13.

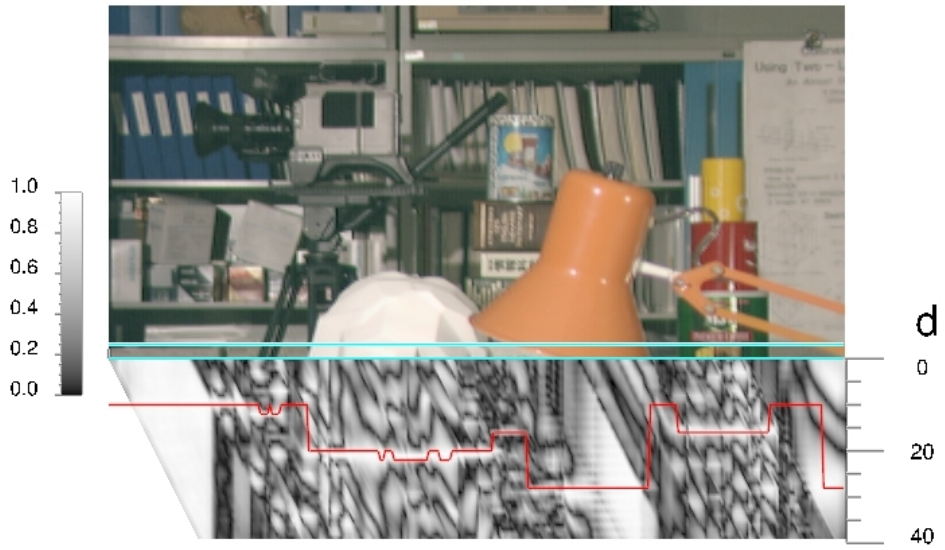


Abbildung 3.13: Die Kostenmatrix für die Zeile 151 des Tsukuba-Testbildes mit dem Disparitätsverlauf, wie er aus der dazugehörigen Disparitätenkarte hervorgeht. Die blauen Linien kennzeichnen die verwendete Fenstergröße von 7 x 7.

Unter Verzicht auf perspektivische Darstellung und Falschfarben für die Korrelationsergebnisse ist hier wieder die Testaufnahme *Tsukuba* dargestellt. Die blauen Linien kennzeichnen die verwendete Fenstergröße von 7x7 Pixeln. Der rot dargestellte Pfad repräsentiert die Disparität, wie sie in der zum Bild gehörigen Disparitätenkarte angegeben ist (vgl. Abschnitt 3.4.2). Deutlich ist erkennbar, dass die Tiefenkarte keine Subpixelgenauigkeit aufweist, da der Disparitätspfad im Bereich des Gipskopfes im linken Teil des Bildes zwischen den ganzzahligen Disparitäten 20 und 21 pendelt und das Maximum der Korrelation deutlich sichtbar dazwischen liegt.

Eine weitere interessante Tatsache erschließt sich, wenn die unkollabierte Kostenmatrix zusammen mit der Disparität für dieselbe Bildzeile dargestellt wird (Abb. 3.14): Verdeckungen des Hintergrunds aufgrund von Objekten im Vordergrund führen dazu, dass mehrere Pixel in einem Bild einem einzigen Pixel im korrespondierenden Bild zugeordnet werden. Für diese Bereiche kann keine Disparität berechnet werden, da die dafür nötige Information nicht in den Ansichten enthalten ist. Dementsprechend ist die Disparitätenkarte zu den Testaufnahmen mit Vorsicht zu genießen: sie enthält Information, die in den beiden Aufnahmen nicht enthalten ist. Dies muss bei der Evaluierung von Algorithmen berücksichtigt werden.

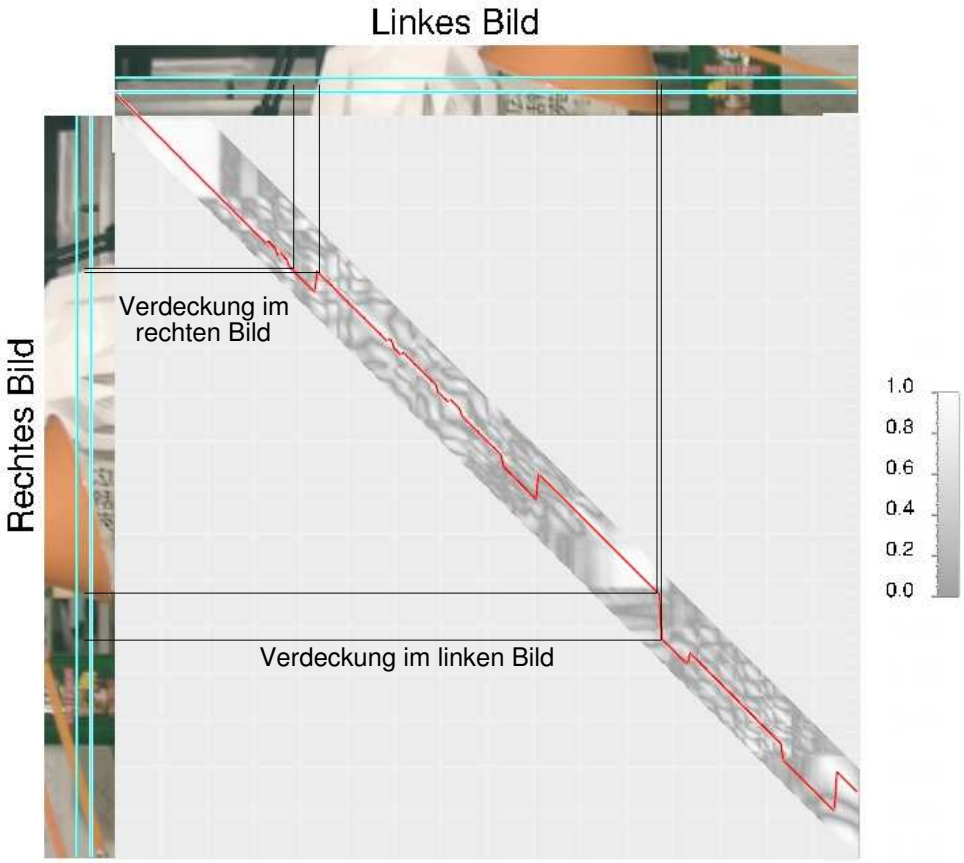


Abbildung 3.14: Effekt von Verdeckung auf den Verlauf der Disparität.

3.4.4 Korrelationsbasiertes Stereo mit einfacher Maximumsuche

Als einfaches Stereosystem soll in diesem Abschnitt das *Small Vision System* vorgestellt werden, da es einen typischen Vertreter dieser Kategorie darstellt. Bei diesem System handelt es sich um ein Stereosystem, das im Forschungsinstitut *SRI International,* USA entwickelt wurde und seit einigen Jahren von der Firma *videredesign*[1)] vertrieben wird. Es basiert auf rektifizierten Bildern und führt zunächst eine Filterung mit dem LoG-Filter durch. Dabei wird die zweidimensionale zweite Ableitung durch den Laplacefilter berechnet. Damit sind die Bilder auf ihre höherfrequente Struktur reduziert und somit frei von Gleichanteil. Da der Laplacefilter jedoch sehr empfindlich auf Bildrauschen reagiert, werden die Bilder in einem vorhergehenden Schritt durch einen Gauß-Filter geglättet. Um den Berechnungsaufwand zu reduzieren, werden beide Faltungskerne kombiniert. Abbildung 3.15 zeigt den Effekt der LoG-Filterung für das gesamte Bild und zur Verdeutlichung als Helligkeitsprofil einer Bildzeile.

Auf diese Filterung folgt eine Berechnung der Kostenmatrix mit der Summe absoluter Differenzen (SAD) als Kostenfunktion. Ohne vorangegangene Filterung würde durch

[1] www.videredesign.com

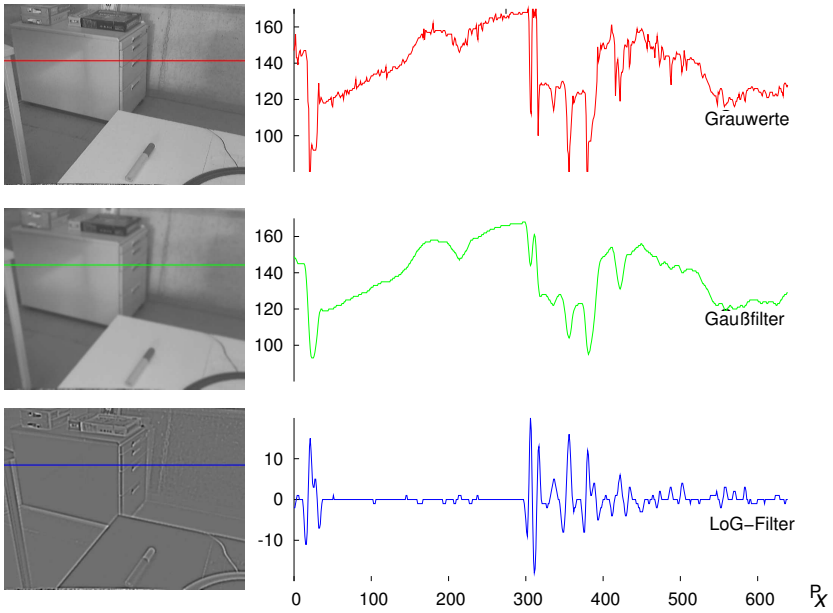


Abbildung 3.15: Effekt der LoG-Filterung auf eine Testaufnahme: Originalbild (oben), Gaußfilterung (mitte) und LoG-Filterung (unten), zur Verdeutlichung jeweils mit Helligkeitsverlauf entlang einer Bildzeile.

dieses einfache Maß eine große Abhängigkeit vom Helligkeitsbias der Bilder entstehen. Die Bestimmung der Disparität basiert auf einer einfachen Suche nach dem lokalen Minimum unter Berücksichtigung maximaler Disparitätsgradienten. Es folgt ein Rechts-Links-Konfidenztest und die subpixelgenaue Verfeinerung der Disparität. Durch die einfachen Algorithmen, effektive Implementierung und die Verwendung von SIMD-Befehlen[1)] zur Parallelisierung erreicht das Small Vision System schritthaltende Verarbeitung auf Standard-PCs. Die folgende Tabelle zeigt die Systemleistung für einen Pentium-III-basierten PC bei 1 GHz:

Auflösung	Disparitäten	Bildwiederholrate
320 x 240	32	42 Hz
640 x 480	16	17 Hz
640 x 480	32	9 Hz
640 x 480	64	5 Hz

Eine Schwäche dieses Systems liegt in der Verwendung einer einfachen Kostenfunktion (SAD). Um die hohe Abhängigkeit von absoluten Helligkeitswerten und damit von der

[1] SIMD: Single Instruction Multiple Data

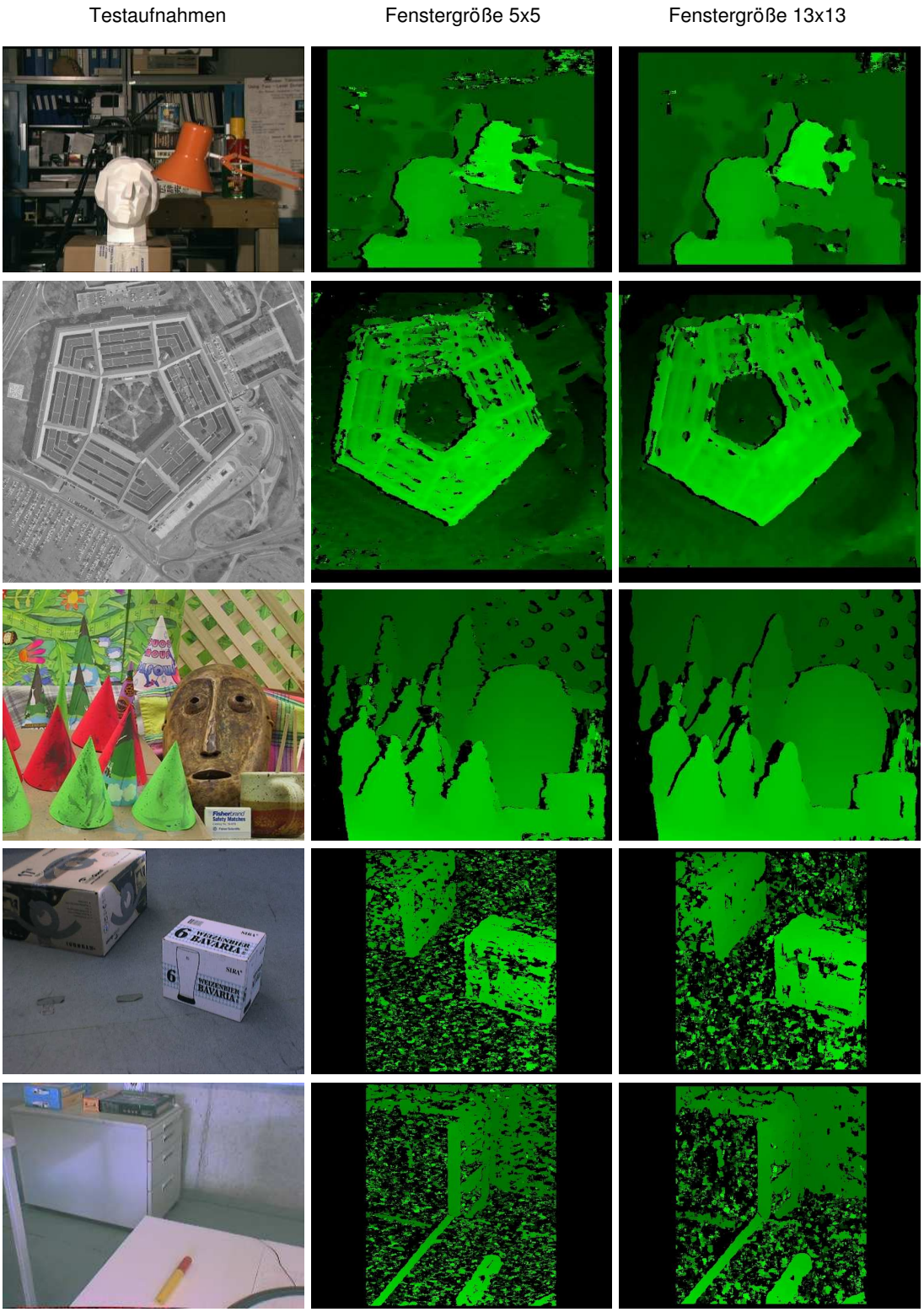


Abbildung 3.16: Ergebnisse einfacher Maximumsuche: Disparitätenkarten für die Fenstergrößen 5x5 und 13x13.

präzisen Helligkeitsabstimmung der Kameras zueinander zu reduzieren, muss ein LoG-Filter für die Vorfilterung zum Einsatz kommen. Dadurch ist das System nicht mehr in der Lage, in schwach texturierten Bildbereichen Tiefeninformation zu extrahieren. Abbildung 3.16 zeigt mit dem Small Vision System erstellte Disparitätenkarten für fünf Testaufnahmen.

Sind bei den stark texturierten Aufnahmen mit geringem Disparitätenbereich *Pentagon* und *Tsukuba* noch nahezu perfekte Ergebnisse erreichbar, so lässt die Leistung aufgrund des großen Disparitätsbereich von *Cones* an Objektkanten bereits nach. Bei den Innenraumaufnahmen mit geringer Texturierung *Telekiste* und *Teletisch* ist eine Berechnung nur noch an Kanten und in stark strukturierten Bereichen möglich. Das starke Rauschen der Disparitäten lässt sich auf einfache Weise durch einen Texturfilter im Bild entfernen. Dies wurde aus Gründen der Vergleichbarkeit hier unterlassen.

3.4.5 Dynamische Programmierung

Wie schon im Abschnitt 3.3.4 erläutert, handelt es sich bei der Dynamischen Programmierung um ein Standardverfahren, das in vielen Bereichen zur Anwendung kommt. Es kann in seiner einfachsten Ausprägung dazu verwendet werden, die Ähnlichkeit zweier Signale zu bestimmen und dabei die beste Zuordnung der Signalabschnitte zueinander zu berechnen. Mit diesem einfachen Beispiel soll das Verfahren zunächst vorgestellt werden.

Nebenstehende Abbildung 3.17 zeigt beispielhaft den Vergleich eines Referenzsignals mit einem Messsignal. Mittels der absoluten Differenzen der Signalwerte als Kostenfunktion werden die beiden zeitdiskreten Signale an allen Abtastpunkten miteinander verglichen. Die Ergebnisse werden in einer Matrix aufgetragen. Deutlich ist zu sehen, dass eine einfache Suche nach dem Minimum pro Spalte kein eindeutiges Ergebnis ergibt und damit keine Aussage, welcher Referenzwert mit welchem Signalwert korrespondiert. Daher wird der Pfad durch die Matrix gesucht, dessen Einträge in ihrer Summe die geringsten Kosten besitzen. Schrittweise von links unten ausgehend werden jeweils die Kosten an einer bestimmten Position (i, j) zu denen des *Vorgängers* mit den geringsten Kosten addiert. Dabei bezeichnet *Vorgänger* die potentiellen Punkte, durch die der Pfad führen könnte.

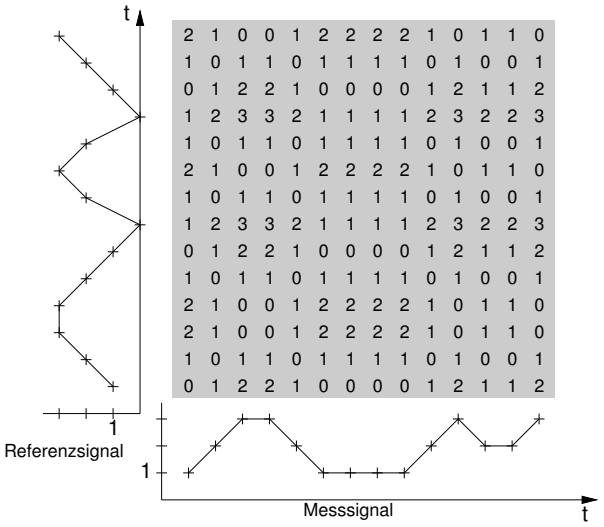


Abbildung 3.17: Vergleich zweier Signale mit Hilfe dynamischer Programmierung: Aufbau der Kostenmatrix mit der absoluten Differenz als Kostenfunktion.

Diese Situation stellt Abbildung 3.18 links dar: Der Punkt (4,9) soll als nächster berechnet werden. Die Punkte auf weißem Hintergrund wurden bereits berechnet, die auf grauem Grund noch nicht. Der neue Wert ergibt sich durch die Addition des Vorgängers mit den geringsten Kosten zum Wert der neuen Zelle. Im dargestellten Fall ist die neue Summe $0 + min(5, 6, 7) = 5$.

Nach der Summation aller Werte ergibt sich ein Bild wie in Abbildung 3.18 rechts dargestellt. Am rechten und oberen Rand ist mit der Rückwärtssuche am Punkt mit den geringsten Kosten zu beginnen. Dann wird zurückverfolgt, welche Vorgängerpunkte die geringsten Kosten besitzen. Das Ergebnis, eine Zuordnung der Kurven zueinander, stellt Abbildung 3.19 dar.

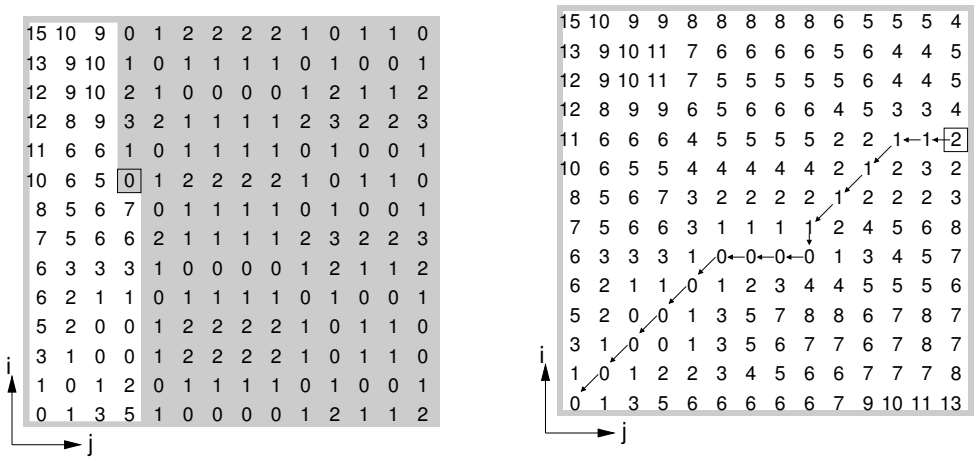


Abbildung 3.18: Schrittweise Summation der Einträge (links) und bester Pfad durch die Matrix als Ergebnis (rechts).

Für die Anwendung auf Bilddaten ergeben sich kaum Änderungen im Ablauf. Die Erzeugung der Kostenmatrix kann durch verschiedene Korrelationsverfahren erfolgen. Die Berechnung der Einträge der Kostenmatrix erfolgt nur für gültige Disparitäten $(0 \ldots d_{max})$. Damit wird die Matrix auch nur in ihrer kollabierten Form mit den Abmessungen b x h x d_{max} aufgebaut um Berechnungszeit und Speicherplatz zu sparen (vgl. Abschnitt 3.4.3). Neben der Kostenfunktion muss in diesem Schritt auch eine Entscheidung über die Fenstergröße gefällt werden. Die Fenstergröße beeinflusst die Anzahl der Fehlkorrespondenzen und die Ortsauflösung an Tiefensprüngen. Auf der Basis einer Kostenmatrix erfolgt nun für jede Zeile unabhängig die **Suche nach dem besten Pfad durch die Matrix**.

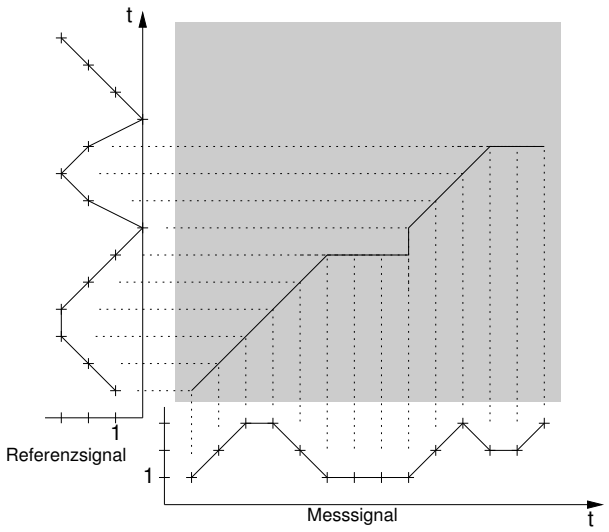


Abbildung 3.19: Abschnittweise Zuordnung der Kurven.

$\mathcal{P}_{ideal}$ repräsentiert den idealen Pfad in der Menge aller gültigen Disparitätspfade $\mathcal{P}_{valid}$, bei dem die Summen aller Kosten entlang des Pfades ein Maximum ergeben.

$$\mathcal{P}_{ideal} = \max_{\forall j : \mathcal{P}_j \in \mathcal{P}_{valid}} \left(\sum_{i=0}^{B-1} I_{Corr,y}(i, \mathcal{P}_j(i)) \right) \tag{3.42}$$

Da die meisten Kostenfunktionen (SAD, SSD, ...) am Ort der besten Übereinstimmung minimal (0) werden, wird die Gleichung (3.42) folgenderweise umgeschrieben:

$$\mathcal{P}_{ideal} = \min_{\forall j : \mathcal{P}_j \in \mathcal{P}_{valid}} \left(\sum_{i=0}^{B-1} C(i, \mathcal{P}_j(i)) \right) \tag{3.43}$$

$C(i, \mathcal{P}_j(i))$ wird dabei direkt aus der Kostenfunktion mit normierten Korrelationswahrscheinlichkeiten $I_{Corr,y} \in [0:1]$ berechnet.

$$C(i, \mathcal{P}_j(i)) = C\left(1 - I_{Corr,y}(i, \mathcal{P}_j(i))\right) \tag{3.44}$$

Wie schon kurz im Kapitel 3.4.3 angedeutet, kommt dem Verlauf des Pfades und dabei insbesondere der Einschränkung des möglichen Pfadverlaufs eine wichtige Rolle zu. So führen Verdeckungen zu Sprüngen im Pfad, entlang derer keine gültige Tiefeninformation existiert. Gleichermaßen führen manche der im Kapitel 3.3.4 vorgestellten *constraints* zu einer Einschränkung möglicher Pfade. Dies wird bei der Dynamischen Programmierung berücksichtigt, indem die Gruppe potentiellen Vorgänger entsprechend angepasst wird. Die zu Beginn dieses Kapitels vorgestellte einfachste Version der Dynamischen Programmierung setzt hier nur die **D**irekten **N**achbarn als Vorgänger ein. Dies wird als DN-Kostenfunktion bezeichnet [17].

DN-Kostenfunktion Bei der Summation entlang des möglichen Pfades $\mathcal{P}$ setzen sich die Kosten für einen aktuellen Kandidaten aus der lokalen Kosten am Punkt (x, d) und den Kosten für den wahrscheinlichsten Vorgänger zusammen. Abbildung 3.20 zeigt zur Verdeutlichung noch einmal eine Kostenmatrix einer beliebigen Bildzeile mit einem angenommenen Disparitätenverlauf. Gleichzeitig sind die kollabierten Versionen der Matrix für die Pfadsuche dargestellt. Um nun den besten Pfad durch die Matrix zu finden, werden für jeden Kandidaten seine drei möglichen Vorgänger mit ihren assoziierten Subpfadkosten $C_{0,1,2}$ unter Berücksichtigung der DN-Kostenfunktion evaluiert:

$$C(x, d) = \min \begin{cases} C_1 + C_{change} \\ C_0 + C_{match} \\ C_2 + C_{change} \end{cases} \tag{3.45}$$

Dabei gilt eine Einschränkung der maximalen Disparitätsdifferenz von $\Delta d = \pm 1$. Abbildung 3.21 stellt die Verhältnisse dar, nachdem die Subpfadkosten der dunkelgrau markierten Felder bereits bestimmt wurden. Um die Kosten des mit „?“ gekennzeichneten Eintrages zu bestimmen, wird unter den drei mit einem Kreis gekennzeichneten Vorgängern unter Berücksichtigung der DN-Kostenfunktion der Kandidat mit den geringsten Subpfadkosten ausgewählt. Mit dem Parameter C_{change} werden Sprünge des Disparitätsverlaufes gehemmt. Ergebnisse für verschiedene Testaufnahmen stellt Abbildung 3.22 dar.

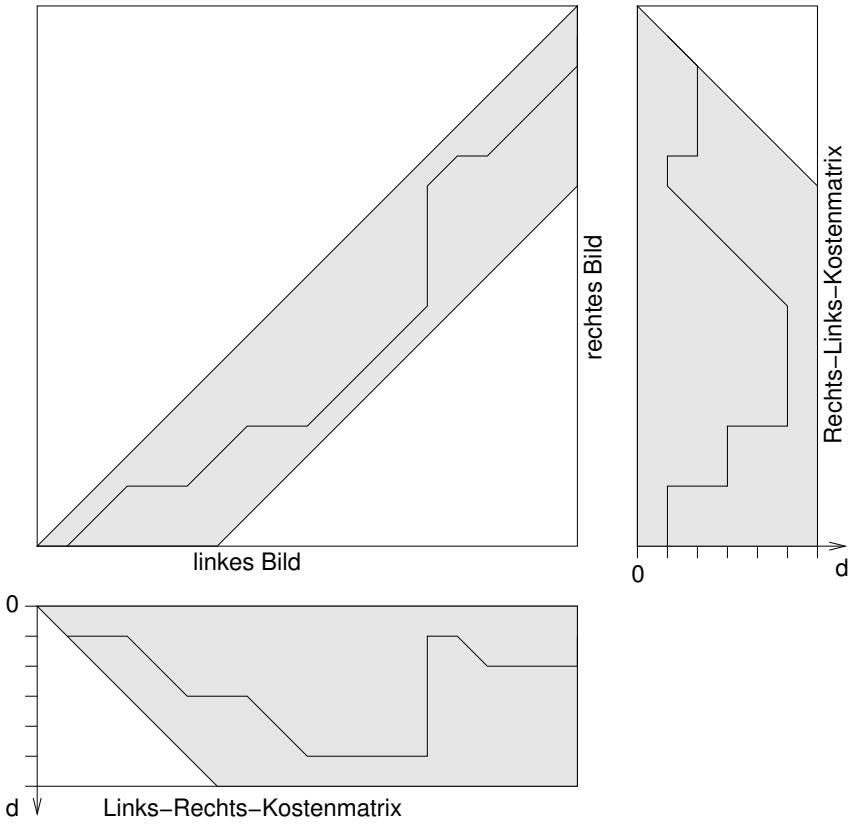


Abbildung 3.20: Kostenmatrix mit angenommenem Pfad und kollabierte Versionen der selben Matrix.

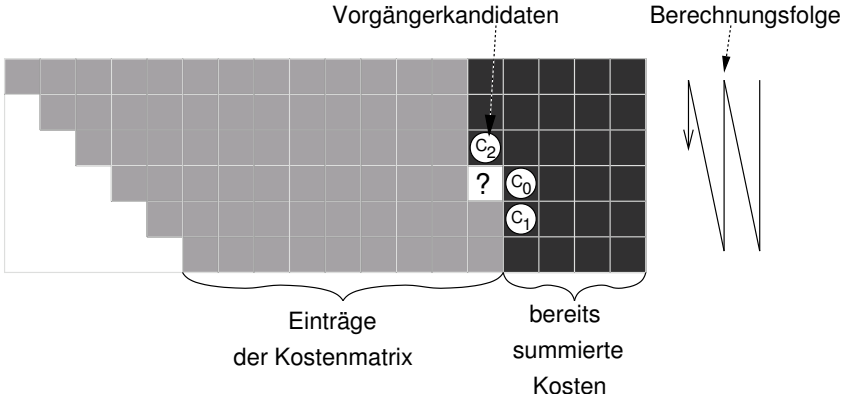


Abbildung 3.21: DN-Kostenfunktion.

EN-Kostenfunktion Mit der EN-Kostenfunktion stellt Falkenhagen in seiner Arbeit [17] eine erweiterte Variante der DN-Kostenfunktion vor. Sie berücksichtigt zusätzlich die erweiterte Kontinuitätsbedingung (*extended continuity constraint*), indem mehr als drei Vorgänger bei der Pfadsuche berücksichtigt werden und damit auch Sprünge im Pfad vorkommen können.

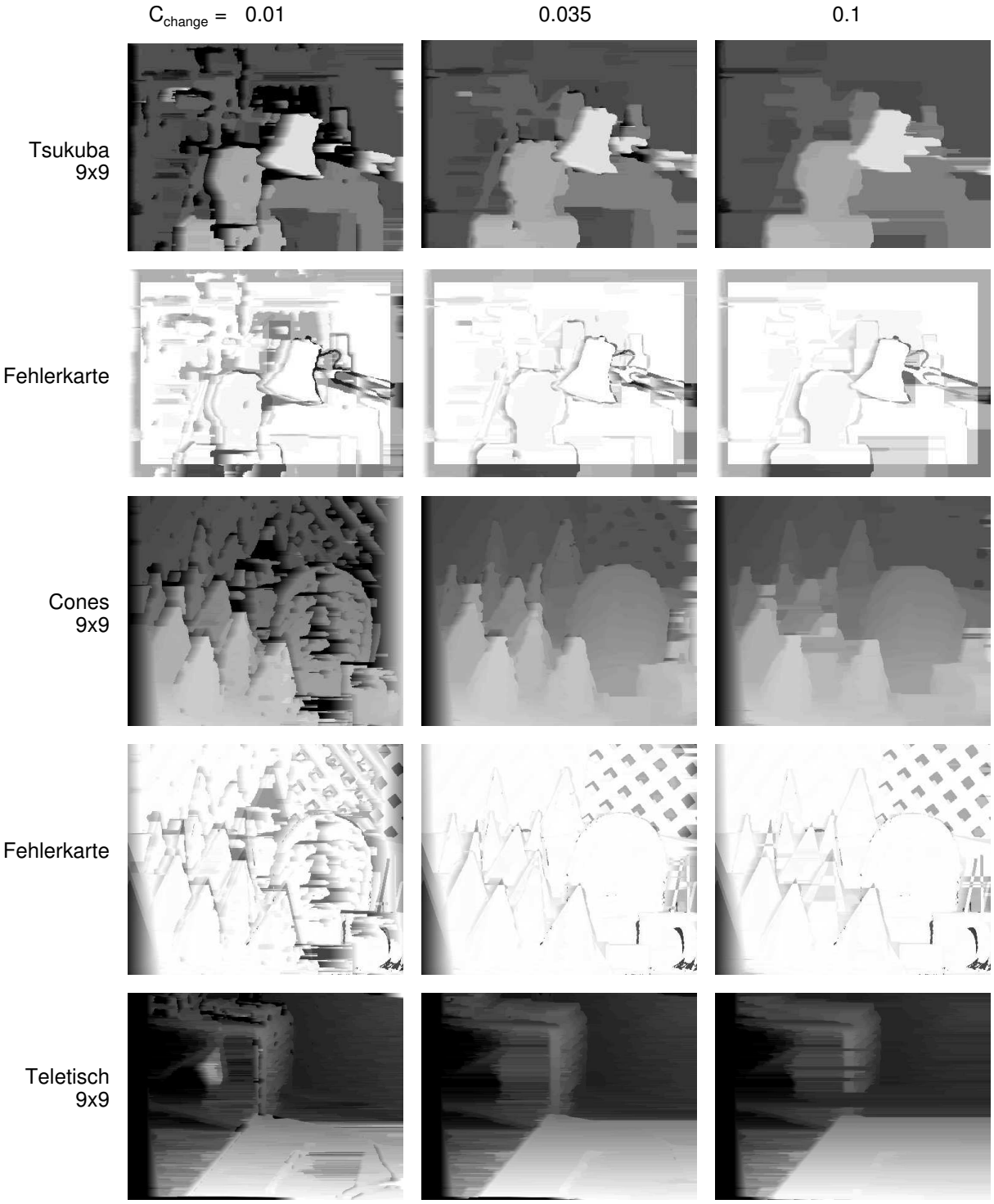


Abbildung 3.22: Disparitätskarten und Fehlerkarten für verschiedene Testaufnahmen und unterschiedliche Strafkosten C_{change}.

Die EN-Kostenfunktion setzt sich gemäß Falkenhagen [17] aus den folgenden drei Komponenten zusammen:

- $C_{\mathcal{P}_{\Delta d}}$, die Kosten des Vorgängers
- C_{match}, die lokalen Kosten
- $C_{penalty_{\Delta d}}$, die Kosten für eine Disparitätsveränderung

Aus den Vorgängerkosten wird – wie bei der DN-Kostenfunktion – der Vorgänger mit den geringsten Kosten ausgewählt. Dabei kann, wie aus Abbildung 3.23 ersichtlich, der maximale Disparitätssprung zum aktuellen Kandidaten größer eins sein: $|\Delta d| \in [0; d_{max} - 1]$.

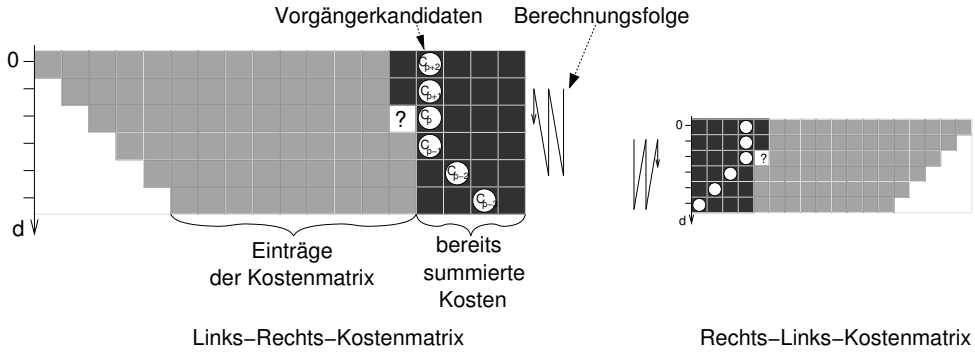


Abbildung 3.23: EN-Kostenfunktion für die Links-Rechts-Kostenmatrix.

Tiefensprünge im Pfad mit $\Delta d = d_{current} - d_{predecessor}$ werden mit Strafkosten $C_{penalty_{\Delta d}}$ belegt, um die Wahrscheinlichkeit für Sprünge über mehrere Disparitätsstufen kontrollieren zu können. Dabei wird die Strafe wie folgt berechnet:

- **Fall 1:** Kontinuierliche Disparität aufgrund einer frontoparallelen Oberfläche in der Szene.

$$\Delta d = 0 \quad \Rightarrow \quad C_{penalty_{\Delta d}} = 0$$

- **Fall 2:** Kontinuierliche Veränderung der Disparität aufgrund einer zur Abbildungsebene geneigten Oberfläche.

$$\Delta d = \pm 1 \quad \Rightarrow \quad C_{penalty_{\Delta d}} = C_{inclination}$$

- **Fall 3:** Diskontinuität der Disparität aufgrund eines Entfernungssprungs vom Hintergrund zum Vordergrund und die damit evtl. verbundene Verdeckung.

$$-\infty < \Delta d < -1 \quad \Rightarrow \quad C_{penalty_{\Delta d}} = C_{discontinouity} + \Delta d \cdot C_{occlusion}$$

- **Fall 4:** Diskontinuität der Disparität aufgrund eines Entfernungssprungs vom Vordergrund zum Hintergrund.

$$+1 < \Delta d < +\infty \quad \Rightarrow \quad C_{penalty_{\Delta d}} = C_{discontinouity}$$

Wie bereits am Ende von Abschnitt 3.4.3 dargestellt, müssen in der kollabierten Kostenmatrix zwei Fälle von Verdeckung unterschieden werden. Die Fälle 3 und 4 spiegeln dies

wieder: Ein fallender[1)] Disparitätsgradient an einer Diskontinuität korrespondiert in einer Links-Rechts-Korrelationsmatrix immer mit Verdeckung im rechten Bild, also Pixeln im linken Bild, die im rechten nicht sichtbar sind. Daraus resultieren Pixel mit unbestimmter Disparität in der Tiefenkarte des linken Bildes. Die Kostenfunktion berücksichtigt dies, indem der erste Vorgänger verwendet wird, dessen korrespondierendes Pixel in der rechten Aufnahme nicht verdeckt ist ($C(x - \Delta d, d - \Delta d)$). Ein zusätzlicher Strafterm berücksichtigt die Anzahl übersprungener Disparitätsstufen. Im Falle eines steigenden Disparitätsgradienten fällt eine Diskontinuität nicht mit einer Verdeckung zusammem.

Im Gegensatz zur DN-Kostenfunktion, bei der nur drei Vorgängerkandidaten zu evaluieren waren, müssen für die EN-Kostenfunktion also d_{max} Vorgängerkandidaten evaluiert werden.

$$C(x,d) = \min_{\forall \Delta d}\{C_{P_{\Delta d}} + C_{penalty_{\Delta d}} + C_{match}\} \tag{3.46}$$

Falkenhagen schlägt auf der Basis eigener Experimente folgende Strafkosten vor:

$$\begin{aligned} C_{inclination} &= C_{change} \\ C_{discontinuity} &= 4 \cdot C_{change} \\ C_{occlusion} &= 4 \cdot C_{change} \end{aligned}$$

Da diese Werte jedoch vom Bildmaterial abhängen, konnte die Wahl Falkenhagens im Rahmen dieser Arbeit nicht bestätigt werden.

Abbildung 3.24 stellt Ergebnisse für dieselben Testaufnahmen wie oben dar. Deutlich ersichtlich sind die Sprünge über mehrere Disparitätsstufen im Fall von Verdeckung. Die Strafterme $C_{inclination}, C_{discontinuity}$ und $C_{occlusion}$ ermöglichen die Anpassung des Systems an das Datenmaterial. Es werden jedoch in Abbildung 3.24 aus Platzgründen nur relevante Parameterkombinationen dargestellt.

Die EN-Kostenfunktion ermöglicht durch die Berücksichtigung der erweiterten Kontinuitätsbedingung zwar theoretisch bessere Ergebnisse, doch steigt auch die Komplexität der erforderlichen Berechnungen.

Rechts-Links-Konfidenztest Bei der Dynamischen Programmierung wird die Berechnung immer relativ zu einem Referenzbild, also der rechten oder der linken Aufnahme durchgeführt. Die resultierende Disparitätenkarte gibt die Disparität für jedes Pixel der Referenzaufnahme an. Das bedeutet, dass durch einen Vergleich der Disparitätskarten der rechten und linken Aufnahme Pixel mit unterschiedlicher Disparität detektiert werden können. Dies ist in der Praxis insbesondere bei Verdeckungen und in Flächen mit geringer Textur der Fall. Als Ausgangspunkt für diesen Test sind Tiefenkarten relativ zur

[1] Im Gegensatz zu den Arbeiten von Falkenhagen [17], sind die Fälle 3 und 4 hier vertauscht. Falkenhagen definiert die Disparität bei paralleler Kamera als grundsätzlich negativ. Er geht also von einem Wertebereich von $d \in [-\infty; 0]$ aus. Im Gegensatz dazu stellen seine Korrelationskarten die Disparität invertiert in einem Wertebereich von $d \in [0; \infty]$ dar.

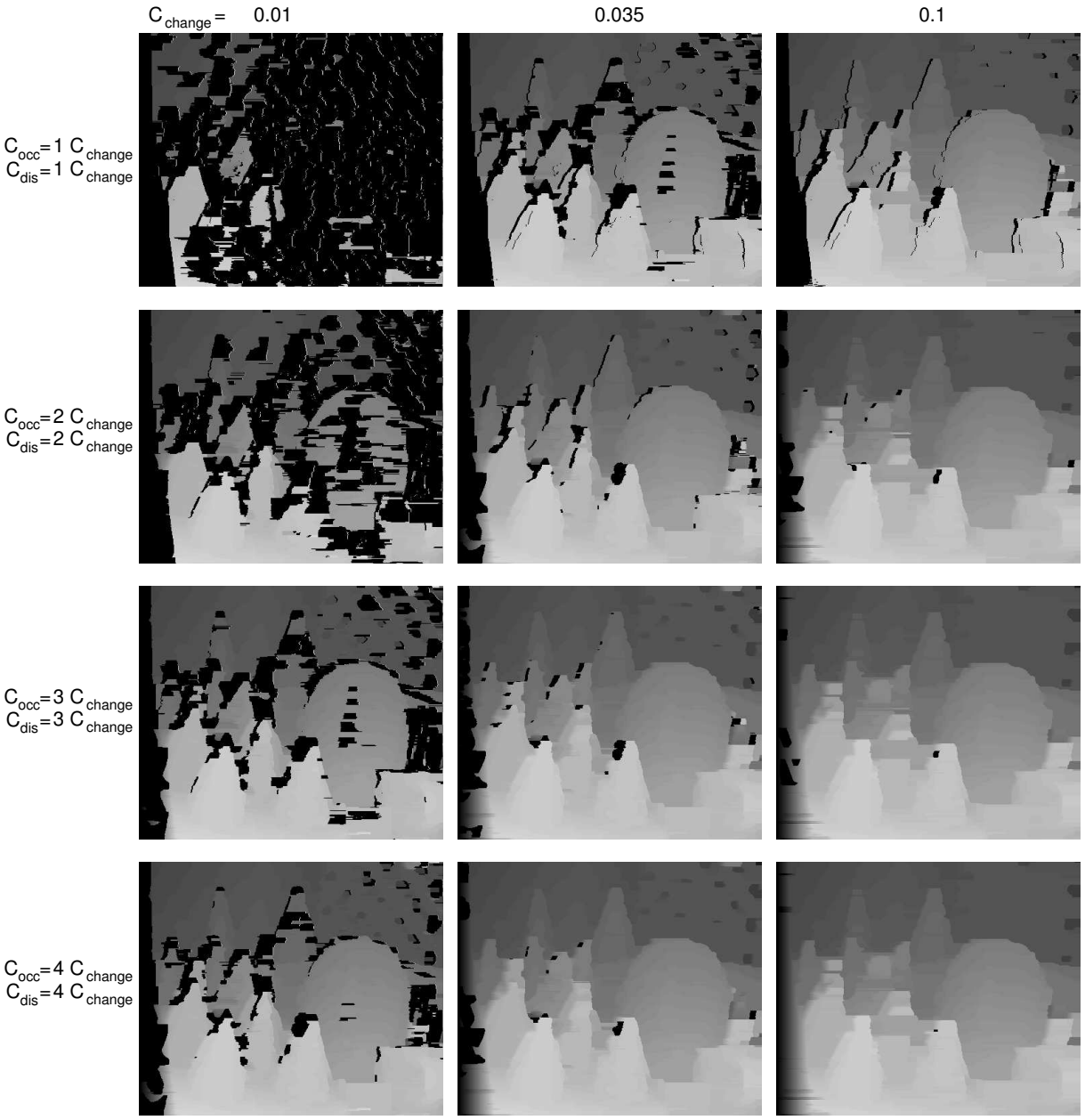


Abbildung 3.24: Variation der Parameter $C_{change}, C_{occlusion}$ und $C_{discontinuity}$ bei der EN-Kostenfunktion für die Links-Rechts-Kostenmatrix der Testaufnahme *Cones*.

linken und rechten Kamera erforderlich, wie sie in Abbildung 3.25 für *Cones* dargestellt sind. Jedes Pixel, für das $|d_L(x,y) - d_R(x - d_L(x,y), y)| \leq 1$ nicht erfüllt ist, wird in der Disparitätskarte für ungültig erklärt.

Insbesondere für *Teletisch* können durch den Konfidenztest massive Pixelfehler detektiert werden. Gleichzeitig zeigt sich aber an der Vorderseite des Rollwagens, dass reflektierende Flächen in beiden Bildern gleiche Disparitätswerte ergeben können und somit nicht durch den Konfidenztest detektiert werden können.

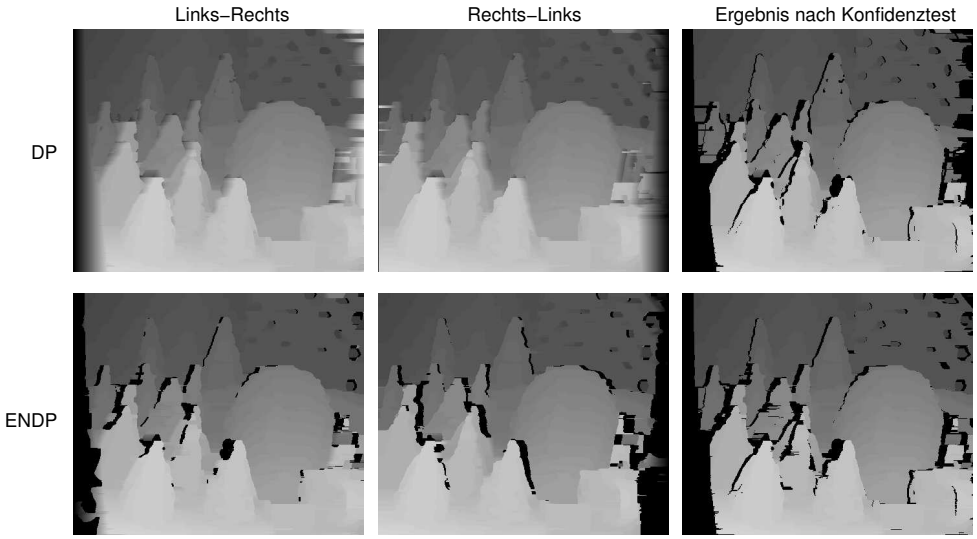


Abbildung 3.25: Rechts-Links-Konfidenztest für die Aufnahme *Cones* für die DN-Kostenfunktion und die EN-Kostenfunktion (SAD, 9x9, C=0.035,2,2).

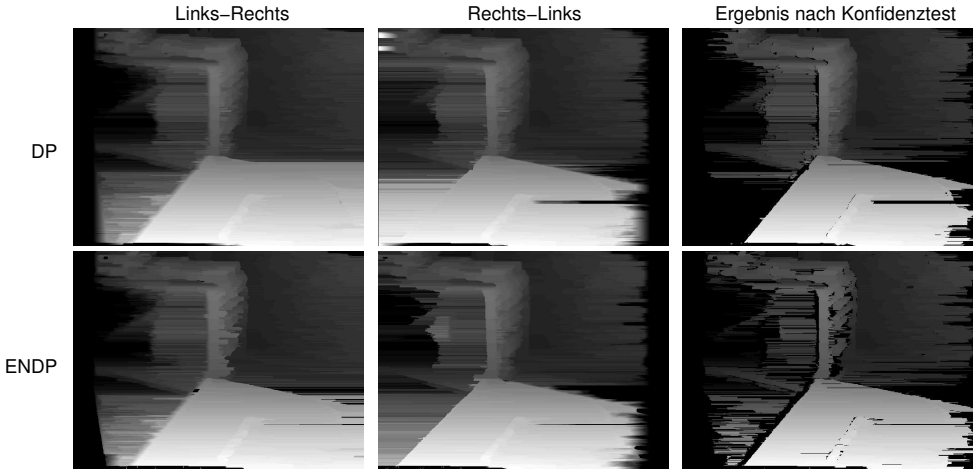


Abbildung 3.26: Rechts-Links-Konfidenztest für die Aufnahme *Teletisch* für die DN-Kostenfunktion und die EN-Kostenfunktion (SAD, 9x9, C=0.035,2,2, Kontrast für die Darstellung erhöht).

Asymmetrische Fenster In allen bisher genannten Ansätzen wird vom Einsatz symmetrischer Fenster fester Größe für die Kostenberechnung ausgegangen. Dabei ergeben sich Probleme an Objektkanten mit Sprüngen in der Disparität, da dort je nach Textur des Vorder- und Hintergrunds die stärker texturierte Hälfte des Fensters dominiert. Dieses Problem ist in Abbildung 3.27 dargestellt.

In den bisher dargestellten Disparitätskarten äußert sich dieser Effekt in unsauberen Konturen und einer Expansion der Objekte. Dieser Effekt wird auch als Korona-Effekt bezeichnet. Abbildung 3.29 zeigt in der oberen Reihe (Mitte und rechts) diesen Effekt an realen Disparitätskarten. Eine häufig verwendete Lösung dieses Problems besteht in der

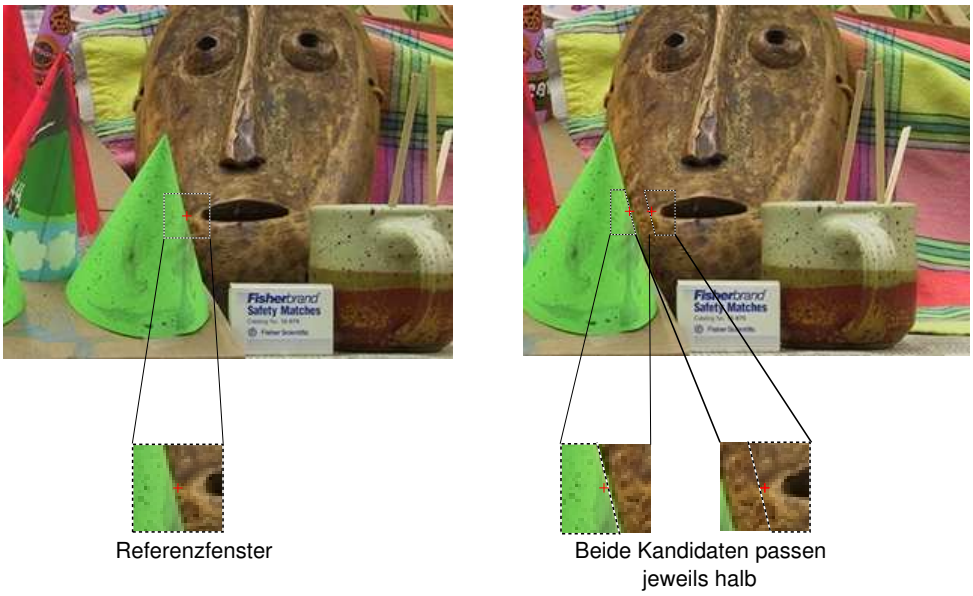


Abbildung 3.27: Zuordnungsproblem an Objektkanten durch Fensterung: die Textur des Vorder- und Hintergrunds entscheidet, welche Disparität an der Kante gilt.

Variation der Fenstergröße – es wird nur die für die lokale Textur minimal nötige Fenstergröße verwendet. Dies zieht jedoch eine mehrfache Berechnung der Kostenmatrix mit unterschiedlichen Fenstergrößen nach sich. Häufig bietet sich daher eine Integration mit einer Bildpyramide zur Beschleunigung der Berechnung an.

Die alternative Lösung durch asymmetrische Fenser wurde in dieser Arbeit eingesetzt. Da eine gefundene Korrespondenz zweier rechteckiger Blöcke prinzipiell die Disparität für alle Pixel innerhalb dieses Blocks festlegt, kann der Blockmittelpunkt beliebig innerhalb eines Blocks gewählt werden. Fusiello schlägt in [27] daher vor, die Mittelpunkte asymmetrisch innerhalb der Blöcke zu verteilen und somit für jedes Pixel und jede potentielle Disparität mehrere Konstellationen der beteiligten Blöcke gemäß Abbildung 3.28 zu berechnen. Ausgewählt wird die Konstellation mit geringsten Kosten für eine Korrespondenz.

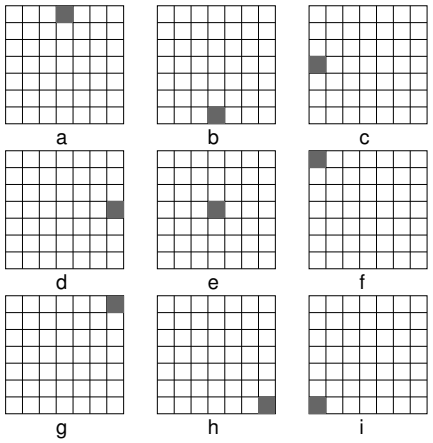


Abbildung 3.28: Asymmetrische Fenster nach Fusiello [27].

Bedingt durch die horizontale Stereobasis sollten vor allem die asymmetrischen Fenster c und d in Abbildung 3.28 deutliche Verbesserung der Konturen ergeben. Abbildung 3.29 zeigt den Effekt auf reale Daten – die Konturen verbessern sich deutlich. Da die Kosten für jede mögliche Fensterkombinationen ohnehin in der Kostenmatrix berechnet werden, beschränkt sich der zusätzliche Aufwand auf eine Minimumsuche in mehreren Einträgen der Kostenmatrix.

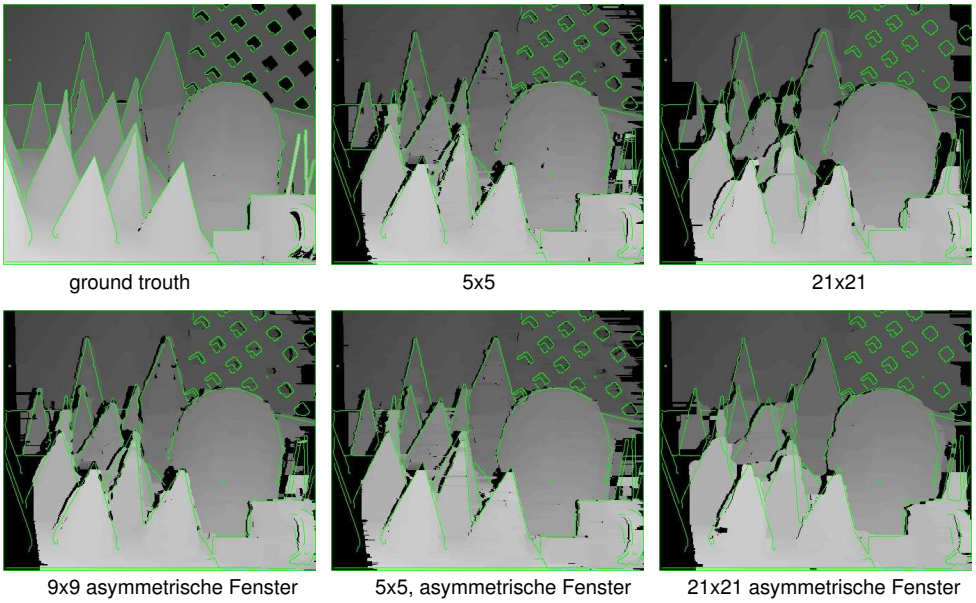


Abbildung 3.29: Unsauber rekonstruierte Objektkanten: Die obere Reihe zeigt die *ground trouth* und Ergebnisse für 5x5 und 21x21 Pixel große Fenster. Die untere Reihe zeigt das Resultat bei der Anwendung asymmetrischer Fenster (DN, C=0.035).

3.4.6 Ein kooperativer Ansatz

Der im folgenden vorgestellte Ansatz von Zitnick und Kanade stellt eine logische Ergänzung zu den bisher in dieser Arbeit präsentierten Methoden dar: Wurden in der einfachen Minimumsuche Entscheidungen über Korrespondenzen noch aufgrund lokaler Bedingungen gefällt, so optimiert die Dynamische Programmierung bereits zeilenweise mit deutlich besseren Ergebnissen. Natürlich gilt die Kontinuitätsbedingung nicht nur in horizontaler, sondern auch in vertikaler Richtung. Zitnick und Kanade stellen in ihrer Arbeit „A Cooperative Approach for Stereo Matching“ [90] einen Ansatz vor, der global operiert und über die gesamten Bilddaten optimiert. Damit wird die Tatsache, dass eine bestätigte Punktkorrespondenz zwischen den beiden Aufnahmen gleichzeitig Einfluss auf ihre benachbarten Punkte hat, besser genutzt.

Der kooperative Ansatz basiert auf zwei grundlegenden Annahmen (vgl. Abschnitt 3.3.4):

1. Der Eindeutigkeitsbedingung

2. und der Kontinuitätsbedingung

Die im Abschnitt 3.4.3 vorgestellte Kostenmatrix der Abmessungen b x h x d_{max} dient auch für diesen Ansatz als grundlegende Datenstruktur. Unter der Annahme rektifizierter Bilder kann jeder ihrer Einträge (x, y, d) als Projektion des Pixels $p_l(x, y)$ im linken Bild und $p_r(x + d, y)$ im rechten Bild betrachtet werden.

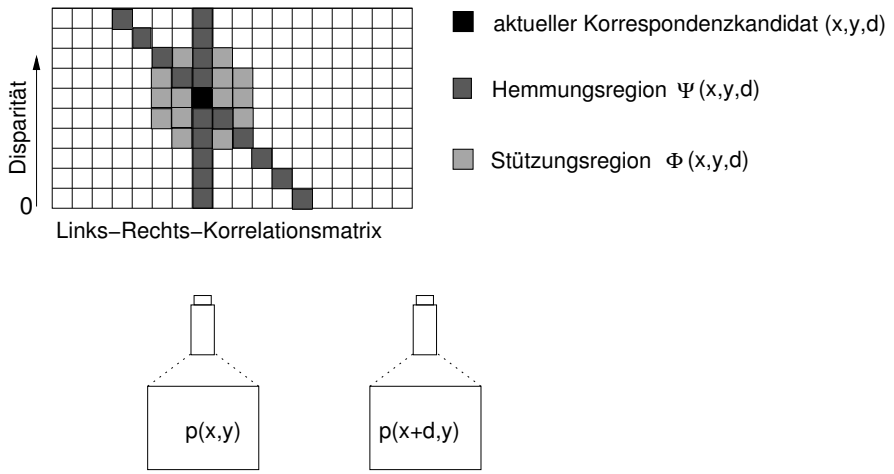


Abbildung 3.30: Kooperativer Ansatz: Stützungs- und Hemmungsregion für einen Eintrag der Kostenmatrix.

Durch iterative Manipulation dieser Matrix werden nun die beiden oben genannten Bedingungen schrittweise durchgesetzt, bis sich eindeutige Maxima[1)] herausbilden. Im folgenden Abschnitt werden die erforderlichen Funktionen entwickelt, die sich aus den beiden Bedingungen ergeben.

Die Kontinuitätsbedingung besagt, dass benachbarte Einträge der Kostenmatrix konsistente Korrelationswahrscheinlichkeiten enthalten. Dies wird durch eine iterative Mittelwertbildung innerhalb eines lokalen, dreidimensionalen Volumens (*local support area*) realisiert. Idealerweise sollten natürlich nur die Punkte in einer Nachbarschaft zu einer Korrespondenzwahrscheinlichkeit beitragen, die zum selben Objekt gehören. Da die Abmessungen des Objekts im Voraus nicht bekannt sind, wird stattdessen nur ein dreidimensionales, rechteckiges Volumen als Stützungsregion angenommen.

Die Eindeutigkeitsbedingung besagt, dass es zu jedem Pixel nur maximal einen Korrespondenzpartner geben darf. Dies wird durch Hemmung benachbarter Einträge in der Kostenmatrix umgesetzt. Ein Eintrag mit großer Korrespondenzwahrscheinlichkeit hemmt somit sämtliche Einträge in seiner Nachbarschaft, die keine Korrespondenzen enthalten dürfen, wenn dieser Punkt ausgewählt werden würde. Für ein bestimmtes Pixelpaar $(p_l(x,y), p_r(x+d,y))$ müssen in der Kostenmatrix somit sämtliche Einträge gehemmt werden, die eine eine Projektion von $p_l(x,y)$ oder $p_r(x+d,y)$ darstellen oder auf $p_l(x,y)$ oder $p_r(x+d,y)$ projezieren. Innerhalb dieser Hemmungsregion *(inhibition area)* $\Psi(x,y,d)$ darf aufgrund der Eindeutigkeitsbedingung nur ein Eintrag als Korrespondenzpaar ausgewählt

[1] Auch hier gilt die im Kapitel 3.4.5 angesprochene Problematik: Die Begriffe Minima und Maxima bzw. Kosten und Ähnlichkeitsmaß sind natürlich dual und von der verwendeten Kostenfunktion abhängig.

werden. Also wird jeder Eintrag in $\Psi(x, y, d)$ durch die Summe aller Korrespondenzwahrscheinlichkeiten in $\Psi(x, y, d)$ gehemmt.

$$R_n(x, y, d) = \left(\frac{S_n(x, y, d)}{\sum_{(x'', y'', d'') \in \Psi(x,y,d)} S_n(x'', y'', d'')} \right)^{\alpha} \tag{3.47}$$

Der Exponent α reguliert die Hemmungsintensität pro Iteration. Damit genau ein Element innerhalb der Hemmungsregion zu 1 konvergiert, muss $\alpha > 1$ gelten. Wird es zu niedrig gewählt, so ist sein Einfluss gering, wird es zu hoch gewählt, kommt es zu starker Glättung und langsamer Konvergenz.

Die von Zitnick und Kanade vorgeschlagene Update-Funktion ist

$$L_{n+1}(x, y, d) = L_0(x, y, d) \cdot \left(\frac{S_n(x, y, d)}{\sum_{(x'', y'', d'') \in \Psi(x,y,d)} S_n(x'', y'', d'')} \right)^{\alpha} \tag{3.48}$$

Nach mehreren Iterationen konvergiert die Kostenmatrix zu eindeutigen Maxima für jedes Pixel. Genau ein Maximum bildet sich am als korrekt identifizierten Disparitätswert pro Pixel. Abbildung 3.31 stellt diesen Prozess für *Cones* dar. Aufgrund der starken Strukturierung dieser Szene bietet die einfache Maximumsuche (0^{te} Iteration) bereits gute Ergebnisse. Wendet man dasselbe Verfahren jedoch auf die wesentlich schwierigere Testaufnahme *Telekiste* an (vgl. Abb. 3.32), zeigt sich ein deutlicher Unterschied.

Problematisch ist die Wahl einer einfachen, quaderförmigen Stützungsregion, da dadurch Objektkonturen unpräzise rekonstruiert werden. Die Komplexität des Verfahrens ist deutlich höher als die Komplexität der Dynamischen Programmierung. Der Algorithmus läuft über jeden Eintrag der Kostenmatrix, wobei für jedes Element die gesamte Hemmungsregion berücksichtigt wird. Zusätzlich ist die Anzahl erforderlicher Iterationen bis zur Konvergenz vom Bildinhalt abhängig. Die Komplexität beträgt $\mathcal{O}(n \cdot (si \cdot bhd_{max})) = \mathcal{O}(n \cdot (bhd_{max}^2))$ mit der Anzahl si von Einträgen in der Hemmungsregion proportional zum Disparitätsbereich d_{max}. Der Speicherbedarf ist ebenfalls wesentlich größer als der der Dynamischen Programmierung, da die initiale Kostenmatrix während der Iterationen verfügbar bleiben muss. Da für die Iterationen die Unterstützungsregion und die Hemmungsregion benötigt werden, muss die gesamte Matrix verfügbar sein. Damit werden zwei komplette Speicherstrukturen von der Größe der Kostenmatrix benötigt. Damit liegt der Speicherbedarf in der Größenordnung von $2 \cdot S_{double} \cdot bhd_{max}$ Bytes (mit S_{double} als Speicherbedarf einer Zahl in der Speicherform *double*).

Verdeckungen werden bei diesem Verfahren durch einen Schwellwert bei der abschließenden Bestimmung der Disparität realisiert. Alle maximalen Einträge der Kostenmatrix nach den Iterationen, die unterhalb eines bestimmten Schwellwertes liegen, werden als

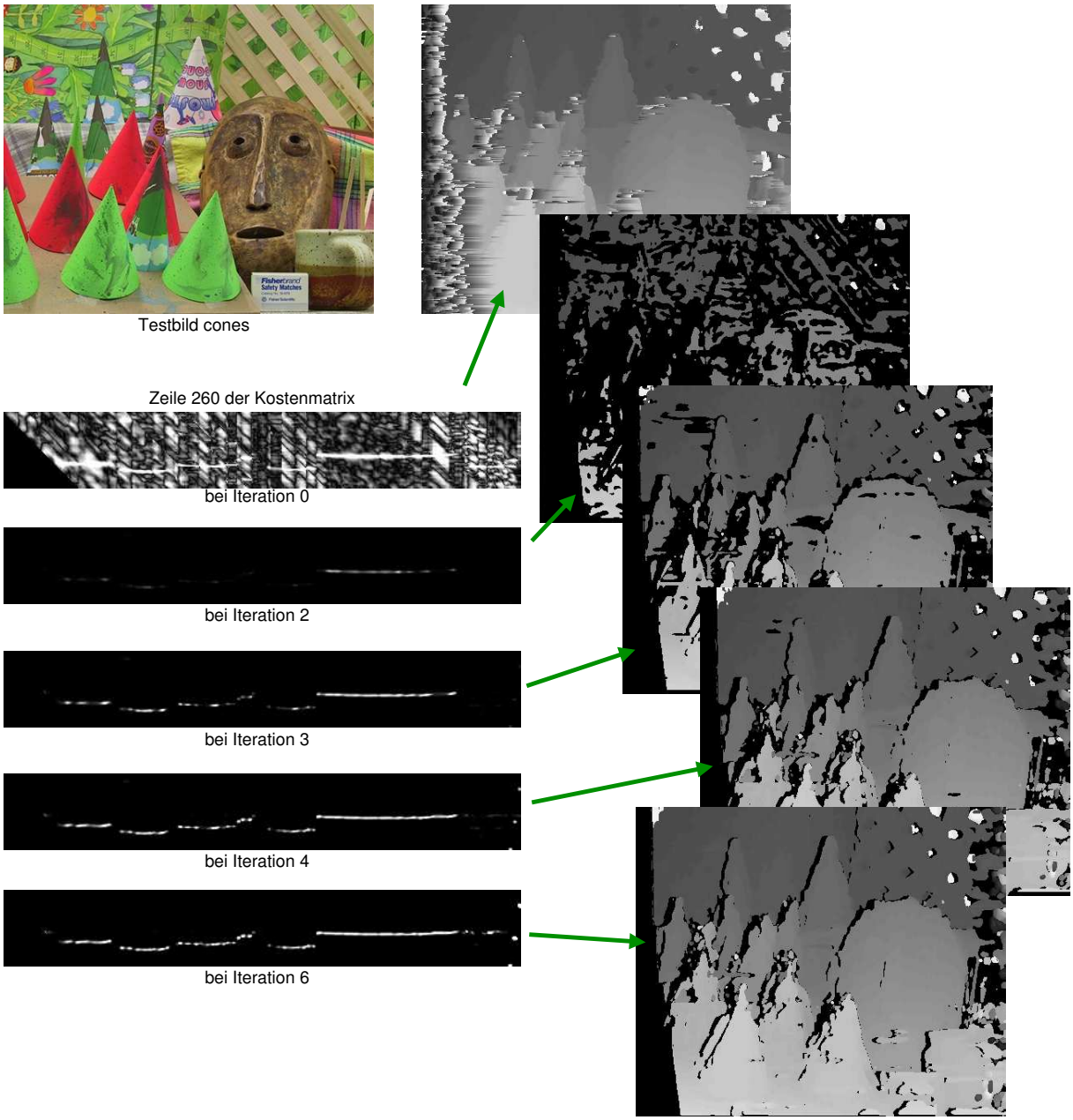


Abbildung 3.31: Testbild *Cones*: Zustand einer Zeile der Kostenmatrix nach den einzelnen Iterationsschritten und damit korrespondierende Disparitätskarte (ZNCC, 9x9, $\alpha = 2$).

ungültig markiert. Abbildung 3.32 zeigt das Ergebnis für die schwach texturierte Aufnahme *Telekiste* mit einem Schwellwert von 0. So werden keinerlei Fehlkorrespondenzen ausmaskiert. Anhand der Entwicklung der dargestellten Zeile der Kostenmatrix zeigt sich der Effekt der Iterationen deutlich: Es bilden sich auch in schwach texturierten Bereichen eindeutige Maxima heraus.

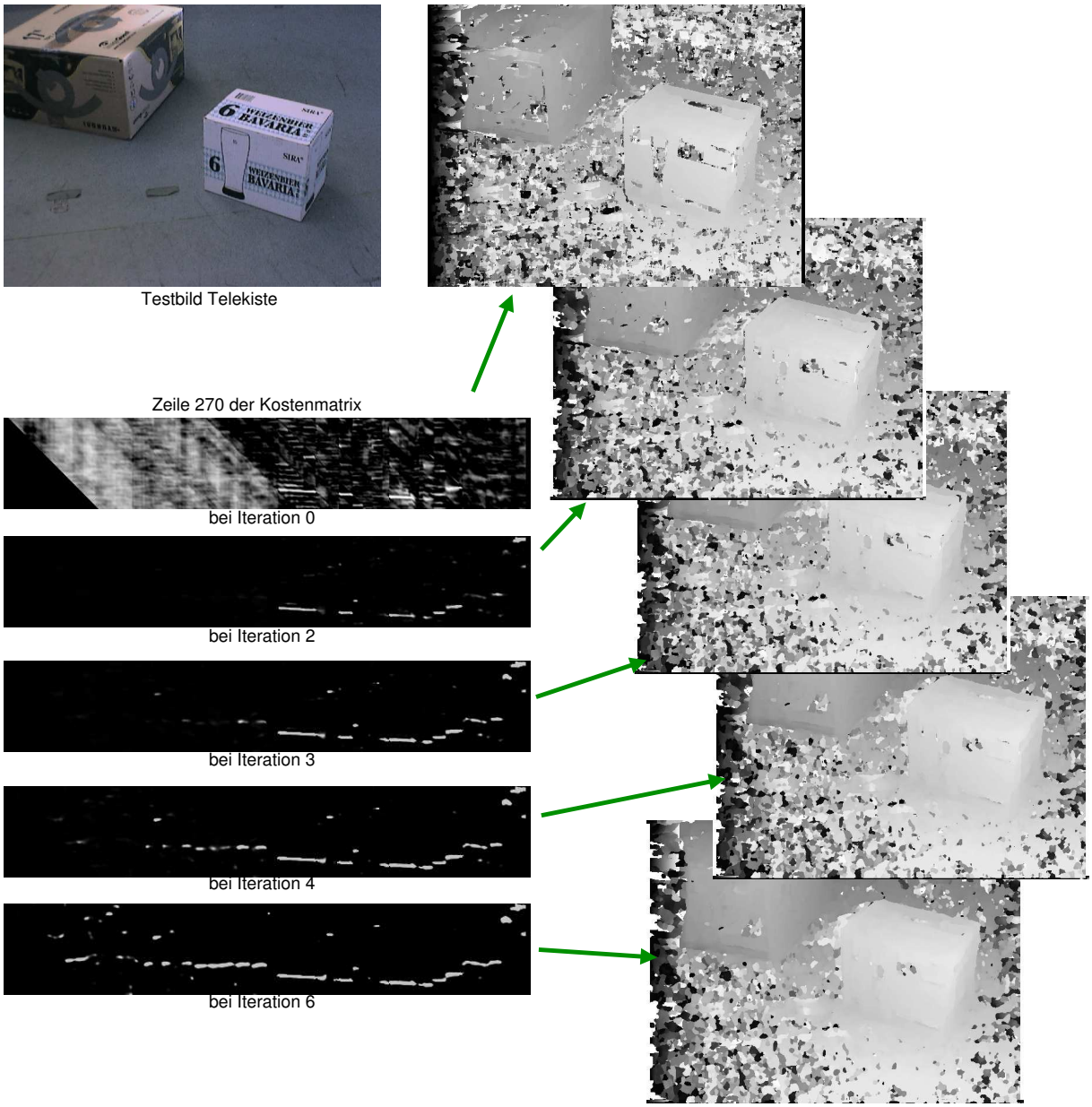
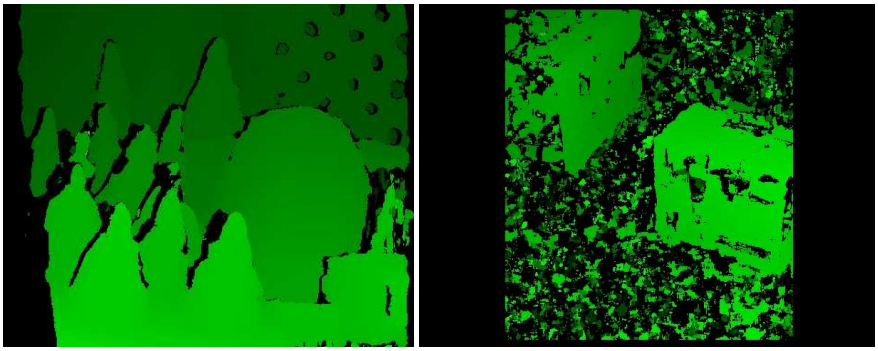


Abbildung 3.32: Testbild *Telekiste:* Zustand einer Zeile der Kostenmatrix nach den einzelnen Iterationsschritten und damit korrespondierende Disparitätskarte (ZNCC, 13x13, $\alpha = 2$).

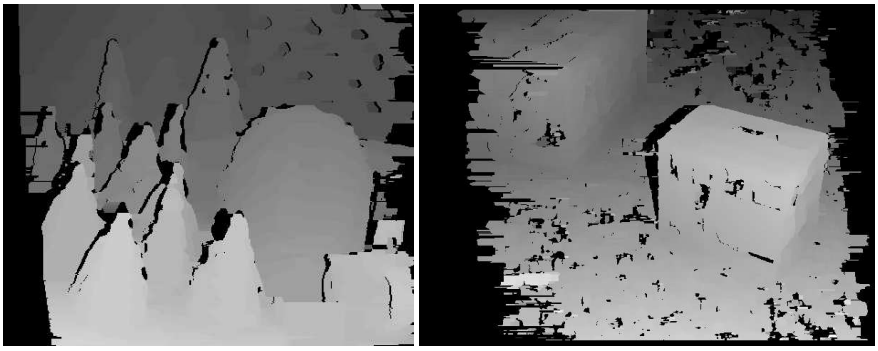
3.4.7 Diskussion der Ergebnisse

Die *einfache Maximumsuche* erscheint durch ihre geringe Komplexität zunächst bei gleichzeitig sehr guter Leistung zunächst als Favorit (vgl. Abb. 3.33). Tatsächlich sind die Resultate für die stark texturierten Testaufnahmen *Cones, Pentagon* und *Tsukuba* nahezu optimal. Einzelne ungültige Disparitätswerte, wie sie bei diesem Verfahren manchmal innerhalb von Regionen mit gleichförmiger Disparität auftreten, lassen sich durch Filterung



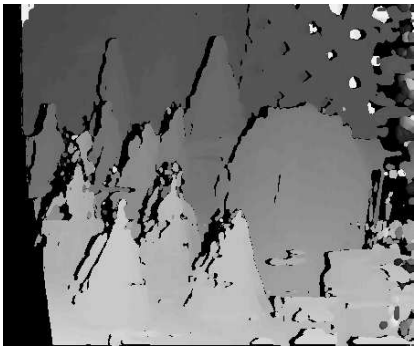
SVS, LoG–Filter, 13x13, t = ~0.08s

SVS, LoG–Filter, 13x13, t = 0.1s



DP–DN, SAD, C= 0.035, 9x9, t=3.0s

DP–DN, ZNCC, C = 0.33, 15x15, t = 11.0s



Cooperative, ZNCC, 9x9, 4 Iterationen, t = 32.5s

Abbildung 3.33: Bestmögliche Ergebnisse der getesteten Verfahren für zwei Testaufnahmen mit jeweiligen Parametern und Ausführungszeiten (Athlon, 1GHz).

maskieren. In deutlichem Gegensatz dazu steht die schlechte Leistung in schwach texturierten Bereichen. Da aber texturarme Oberflächen in Innenräumen dominieren, muss das gewählte Verfahren auch für diese Bereichen gute Leistung erzielen.

Das reduziert die Auswahl unter den getesteten Verfahren auf Methoden, die durch Nutzung der Einschränkungen (*constraints* vgl. Abschnitt 3.3.4) und durch größere Empfindlichkeit auch in schwach texturierten Bereichen Ergebnisse erzielen. Dabei disqualifiziert sich das zuletzt vorgestellte Verfahren durch seine hohe Komplexität und damit durch hohe Ausführungszeiten.

So stellt die Dynamische Programmierung den günstigsten Kompromiss zwischen Komplexität und Leistung dar. Ihr Hauptnachteil liegt in dem fehlenden Bezug benachbarter Bildzeilen. Daraus resultieren typische Kammstrukturen an Objektkanten. Filterung der Disparitätenkarte kann, wie im folgenden Kapitel ausgeführt wird, diesen Nachteil etwas ausgleichen. Ein weiterer Nachteil ist die für eine schritthaltende Verarbeitung zu große Komplexität des Verfahrens.

Für die Pfadsuche stehen bei der Dynamischen Programmierung die direkte Nachbarschaft (DN) und die erweiterte Nachbarschaftssuche zur Auswahl. Obwohl sich der Einsatz der ENDP-Kostenfunktion günstig auf die Ergebnisse auswirkt, muss aus Leistungsgründen darauf verzichtet werden. Pro Eintrag in der Kostenmatrix verursacht der Einsatz der ENDP-Funktion im Vergleich zur einfachen Nachbarschaft $d_{max} - 3$ zusätzliche Vergleiche für die Suche nach dem minimalen Vorgänger. Dies äußert sich in erheblich größeren Ausführungszeiten. Vergleicht man die Ergebnisse der DP und der ENDP-Nachbarschaft (vgl. Abb. 3.25), so zeigt sich, dass der Rechts-Links-Konfidenztest ähnliche Ergebnisse bei einer deutlich geringeren Komplexität bringt.

3.4.8 Implementierung

Bildentzerrung Bei der zur Rektifizierung nötigen Bildentzerrung handelt es sich um eine Kombination aus radialer Entzerrung und affiner Abbildung. Um die notwendigen Berechnungen nicht für jedes Pixel eines jeden Bildes durchführen zu müssen, wird für jede Kamera eine *look-up-table* angelegt, der die erforderlichen Pixelkorrespondenzen enthält. Darin sind für jedes Pixel im entzerrten Zielbild die Koordinaten im verzerrten Ursprungsbild angegeben. Auf dieser Basis wird eine lokale Interpolation der vier Nachbarpixel durchgeführt.

Korrelationsverfahren Bei allen eingesetzten Kostenfunktionen hat die Implementierung entscheidenden Einfluss auf die Komplexität. Herausragendes Beispiel ist die Kreuzkorrelation, daher soll sie hier als Beispiel dienen, doch sind die Prinzipien auf alle genannten Kostenfunktionen anwendbar.

$$c_{ij,d} = \frac{cov_{ij,d}(I_L, I_R)}{var_{ij}(I_L) \cdot var_{ij,d}(I_R)} \tag{3.49}$$

$$cov_{ij,d}(I_L, I_R) = \sum_{m=i-K}^{i+k} \sum_{n=j-L}^{j+L} (I_L(m,n) - \overline{I_L})(I_R(m,n) - \overline{I_R}) \tag{3.50}$$

$$var^2_{ij}(I_L) = \sum_{m=i-K}^{i+k} \sum_{n=j-L}^{j+L} (I_L(m,n) - \overline{I_L})^2 \tag{3.51}$$

$$var^2_{ij,d}(I_R) = \sum_{m=i-K}^{i+k} \sum_{n=j-L}^{j+L} (I_R(m+d,n) - \overline{I_R})^2 \tag{3.52}$$

Für die Berechnung der Kreuzkorrelation gemäß Gleichung 3.49 werden im einfachsten Fall fünf geschachtelte Schleifen verwendet:

```
Mittelwert links berechnen
Mittelwert rechts berechnen
Varianz links berechnen
Varianz rechts berechnen
for h = 4...Bildhöhe-3
   for b = 4...Bildbreite-3
      for d = 1...Dmax
      fenstersummeL = fenstersummeR = 0
        for x = b-x...b+x
           for y = h-x...h+x
              fenstersummeL += PixelwertL(x,y)- Mittelwert links
              fenstersummeR += PixelwertR(x,y)- Mittelwert rechts
           end
        end
        Kostenmatrix(h,b,d) = fenstersummeL * fenstersummeR  /
                              (sqrt(VarianzL * VarianzR))
      end
   end
end
```

Damit ergibt sich die Komplexität von $\mathcal{O}(h \cdot b \cdot d_{max} \cdot W^2)$ alleine für die Berechnung der Kreuzkorrelation *bei bereits vorliegenden Mittelwerten und Varianzen*. Sun [84] beschreibt basierend auf den Arbeiten von McDonnel [61] eine auf laufenden Summen basierende Technik, mit der sich die Komplexität aller Berechnungen deutlich reduzieren lässt. Dabei wird ein Berechnungsfenster über die beiden Ansichten geschoben und dabei in jedem Schritt die neu hinzugekommenen Werte hinzuaddiert und die aus dem Fenster

hinausgeschobenenen Werte subtrahiert. Dies gelingt durch die Umformung der beteiligten Gleichungen wie folgt:

$$\begin{aligned} var_{ij}^2(I_L) &= \sum_{m=i-K}^{i+k} \sum_{n=j-L}^{j+L} (I_L(m,n) - \overline{I_L})^2 \\ &= \sum_{m=i-K}^{i+k} \sum_{n=j-L}^{j+L} (I_L(m,n))^2 - (2K+1)(2L+1)\overline{I_L}^2 \qquad (3.53) \\ cov_{ij,d}(I_L, I_R) &= \sum_{m=i-K}^{i+k} \sum_{n=j-L}^{j+L} (I_L(m,n) - \overline{I_L})(I_R(m,n) - \overline{I_R}) \\ &= \sum_{m=i-K}^{i+k} \sum_{n=j-L}^{j+L} I_L(m,n) I_R(m+d,n) - (2K+1)(2L+1)\overline{I_L} \cdot \overline{I_R} \qquad (3.54) \end{aligned}$$

In dieser Form lassen sich die Gleichungen nun auf die beschriebene Weise beschleunigt implementieren.

Rechts-Links-Konfidenztest Für den Rechts-Links-Konfidenztest ist die Berechnung der Kostenmatrix relativ zu beiden Aufnahmen nötig. Betrachtet man jedoch die Herkunft der kollabierten Kostenmatrizen aus einer einzigen Datenstruktur, die jeweils unterschiedlich umgeformt wird (vgl. Abschnitt 3.4.3 und Abbildung 3.11), zeigt sich, dass die doppelte Berechnung nicht notwendig ist. Statt dessen wird durch eine einfache Umformung die jeweils korrespondierende Darstellung der Kostenmatrix gewonnen.

3.4.9 Zusammenfassung

Im vorangegangenen Kapitel „3D-Rekonstruktion aus Stereobildern“ wurden zunächst die Grundlagen der Bildaufnahme und Stereopsis besprochen. Aufbauend auf diesem Wissen konnte ein gegliederter Überblick über die in der Literatur beschriebenen Methoden zur 3D-Rekonstruktion gegeben werden.

Den Schwerpunkt des Kapitels bilden schließlich die detaillierte Darstellung dreier typischer Verfahren und ihr Vergleich. Dies ist im Rahmen dieser Arbeit erforderlich, da nur so die Leistung der einzelnen Verfahren anhand zu erwartender Szenarien beurteilt werden kann. Die Dynamische Programmierung bildet dabei den besten Kompromiss zwischen Ergebnisqualität und Komplexität. Dies wird in einer Diskussion der drei Verfahren begründet. Ein Abschnitt zu einigen Implementierungsdetails mit denen die Komplexität weiter reduziert werden kann schließt das Kapitel ab.

4 Modellaufbau

Der Aufbau polygonaler Netze aus Tiefenkarten bildet das zentrale Thema dieses Kapitels. Es stellt nach der 3D-Rekonstruktion den zweiten Schwerpunkt dieser Arbeit dar. Nach einer Beschreibung der Problemstellung werden die Grundlagen polygonaler Netze eingeführt, um dann im mittleren Teil dieses Kapitels zu den Details der Realisierung vorzudringen. Begonnen wird mit der Filterung der Disparitätenkarten, um danach verschiedene Varianten der initialen Triangulierung vorzustellen. Nach einem Abschnitt zur Netzdezimierung bildet das Thema Netzfusion den Abschluss des Themas Netzaufbau. Unter dem Begriff Netzfusion ist im Rahmen dieser Arbeit sowohl die räumliche Fusion der Daten aus unterschiedlichen Kamerapositionen als auch die zeitliche Fusion der Modelldaten zu unterschiedlichen Zeitpunkten zusammengefasst. Im Kontrast zur bis hierher vorgestellten Methode zum Netzaufbau wird im letzten Abschnitt ein auf einer neuartigen Datenstruktur basierendes Verfahren vorgestellt, das es ermöglicht, die Szeneninformation im Form von Punktewolken und Polygonen parallel zu modellieren. Obwohl diese Methode viele Vorteile bringt, konnte sie sich aus Komplexitätsgründen nicht gegen die einfachere Methode durchsetzen.

4.1 Problemstellung

Abbildung 4.1 fasst das Problem des polygonalen Modellaufbaus zusammen. Aus einer Stereoaufnahme einer Szene zum Zeitpunkt t_0 wird eine Disparitätenkarte berechnet. Das Ergebnis der Triangulierung wird in ein 3D-Szenenmodell eingetragen. Zu einem darauffolgenden Zeitpunkt t_1 muss die resultierende Disparitätenkarte in das polygonale Szenenmodell integriert werden. Zu beachten ist dabei, dass es sich um eine andere Problemstellung handelt, als sie zum Beispiel in den Arbeiten von Pollefeys [70] oder Eisert et al. [16] bearbeitet wird, da der Modellaufbau im Kontext der Telepräsenz nie als abgeschlossen betrachtet werden kann. Es wird also das Modell nicht auf der Basis eines geschlossenen Datensatzes aus einer begrenzten Anzahl von Aufnahmen erstellt, sondern idealerweise mit jeder neuen Aufnahme der Szene erweitert und aktualisiert. Ebenfalls unterscheidet sich die Problemstellung von typischen 3D-Videokonferenz-Systemen, wie sie zum Beispiel von Daniilidis [66] entwickelt wurden, da dort kein stabiles polygonales Netz zur Modellierung erforderlich ist.

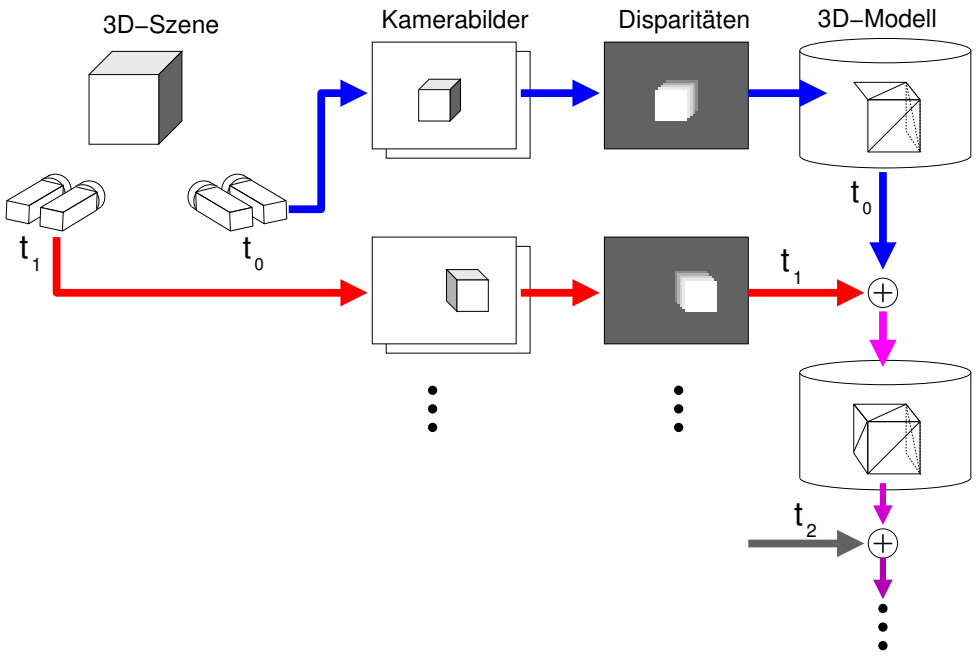


Abbildung 4.1: Übersicht zum Modellaufbau: Zu unterschiedlichen Zeiten aufgenommene Stereoaufnahmen werden in ein polygonales Szenenmodell integriert.

4.2 Grundlagen polygonaler Netze

4.2.1 Begriffe

Netz

Dreiecksnetze bestehen aus Eckpunkten, Kanten, und Dreiecken. Abbildung 4.2 veranschaulicht die Terminologie und die Tatsache, dass Netzelemente als eigenständige Primitiva existieren. So definiert eine geschlossene Folge dreier Kanten kein Dreieck. Dreiecke bestehen grundsätzlich aus drei Kanten. Kanten bestehen aus zwei Eckpunkten, zwei Eckpunkte definieren jedoch keine Kante.

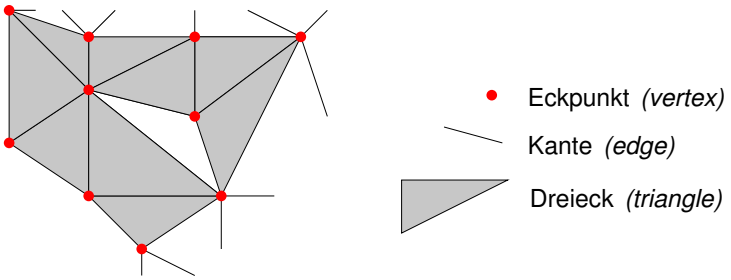


Abbildung 4.2: Terminologie einer Dreiecksnetz-Struktur.

2D-, $2\frac{1}{2}$D- und 3D-Netze

Abbildung 4.2 zeigt ein einfaches **2D-Netz**. Alle Elemente befinden sich im zweidimensionalen Raum. Somit ist dieser Netztyp zum Beispiel geeignet, zweidimensionale Regionen beliebiger Form zu approximieren.

$2\frac{1}{2}$D-Netze haben eines ihrer Hauptanwendungsbereiche in der digitalen Geländemodellierung. Mathematisch stellen sie einen Graphen einer bivariaten Funktion $z = \sigma(x, y)$ definiert auf einer Untermenge $\mathcal{D}$ der x-y-Ebene dar.

3D-Netze werden üblicherweise für die Modellierung einzelner Objekte verwendet. Typische Beispiele zeigt Abbildung 4.3 a, b und c. Die Netze a.) und b.) stellen dabei einfache geschlossene Objekte dar. Abbildung 4.3c zeigt dagegen ein komplexeres Modell: An den mit einem Pfeil gekennzeichneten Stellen schneiden sich Flächen. Obwohl dies nicht notwendigerweise problematisch ist, führen derartige Strukturen zu unnötig komplexen Algorithmen. Daher sollen im Rahmen dieser Arbeit ausschließlich *mannigfaltige* Netze eingesetzt werden.

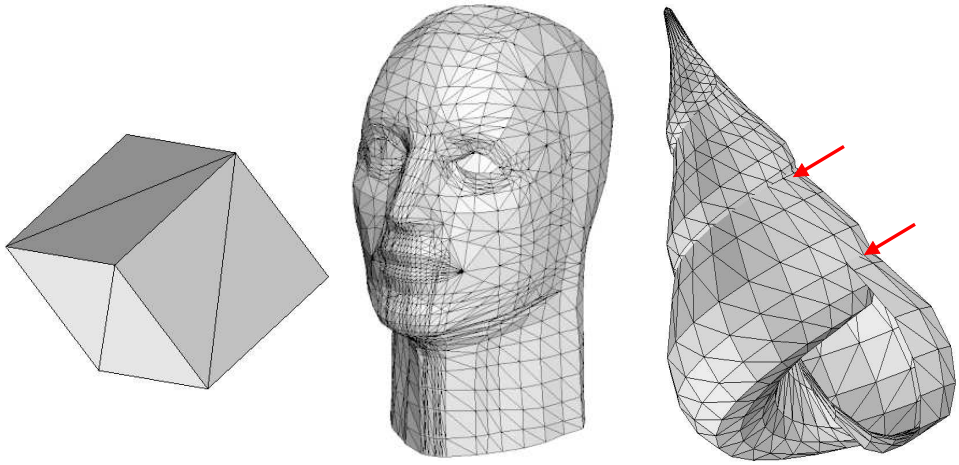


Abbildung 4.3: Beispiele für 3D-Netze. a.) und b.) sind mannigfaltige Netze, in c.) schneiden sich Flächen ohne gemeinsame Kante, somit handelt es sich nicht um ein mannigfaltiges Netz.

Obwohl die Definition des Begriffs der Mannigfaltigkeit mathematisch gesehen über den hier dargestellten Ausschnitt hinausgeht, ist im Rahmen dieser Arbeit nur eine vereinfachte Definition erforderlich. Ein mannigfaltiges Netz besitzt folgende vier Eigenschaften:

1. An jede Kante stoßen genau zwei Flächen
2. Um jeden Eckpunkt existiert ein einziger Ring von Flächen
3. Flächen können sich nur an einer gemeinsamen Kante/Ecke schneiden.
4. Die Euler-Poincaré-Gleichung muss erfüllt sein: $E - K + F - (L - F) - 2(S - G) = 0$
   E - Anzahl der Knoten *(vertices V)*

K - Anzahl der Kanten *(edges E)*
F - Anzahl der Facetten *(faces F)*
L - Anzahl der Schleifen *(loops L)*
S - Anzahl der Schalen *(shells S)*
G - Anzahl der Henkel *(genus G)*
(Anzahl der Löcher durch die Schalen)

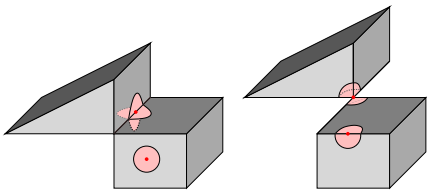


Abbildung 4.4: Lokale Äquivalenz zu einer Kreisscheibe als Hinweis auf Mannigfaltigkeit

Abbildung 4.4 zeigt zwei Beispiele nicht mannigfaltiger Körper. Sie verletzen Bedingung 1 bzw. 2. Der in 4.3c dargestellte Körper verletzt Punkt 3, da sich Flächen ohne gemeinsame Kante schneiden. Bei Punkt 4 handelt es sich um eine einfache Grundbedingung, die die Anzahl von Eckpunkten, Kanten und Flächen für beliebige Objekte zueinander in Beziehung setzt. Sie wurde um Löcher und Durchstiche erweitert.

Eine weitere anschauliche Definition der Mannigfaltigkeit lautet: *Jeder Punkt auf der Oberfläche (dem Rand des Körpers) hat eine Umgebung, die zu einer Kreisscheibe topologisch äquivalent ist.* Diese Definition veranschaulicht Abbildung 4.4. Deutlich zeigt sich, dass die kritischen Stellen nicht homeomorph auf eine Kreisscheibe abgebildet werden können.

4.2.2 Triangulierung

Der Begriff Triangulierung bezeichnet die Verknüpfung von irregulär verteilten Eckpunkten zu Dreiecksnetzen. Dies gilt jedoch zunächst nur für den zweidimensionalen Fall, da sich hier die verknüpfenden Polygone immer auf Dreiecke zurückführen lassen. Im dreidimensionalen Fall wird der Vorgang als *Tetrahedrization* bezeichnet, da die minimale volumenfüllende Form die Form eines Tetraeders besitzt. Der Begriff Triangulierung wird jedoch im dreidimensionalen Fall auch benutzt, um Grenzschichten in Volumendaten oder parametrische Oberflächen als Dreiecksnetze darzustellen. So könnten die Objekte in Abbildung 4.3 a und b als Volumendaten vorliegen, über deren Oberfläche ein Dreiecksnetz gelegt wurde. Mit dem *Marching-Cubes*-Algorithmus [57] gibt es seit einigen Jahren eine Standardlösung für dieses Problem, das zum Beispiel bei der silhouettenbasierten Objektrekonstruktion auftritt.

Da im vorliegenden Kontext der Umwandlung von Tiefenkarten in Dreiecksnetze immer $2\frac{1}{2}$-D Daten zugrundeliegen, lässt sich das Problem der Triangulierung immer zweidimensional behandeln. Obwohl ein Satz von Eckpunkten prinzipiell durch eine beliebige Triangulierung verbunden werden kann, spielt die *Delaunay-Triangulierung* mit Abstand die wichtigste Rolle. Dies resultiert aus den besonderen Eigenschaften, die sie besitzt. Aus diesem Grund beschäftigt sich der folgende Abschnitt etwas ausführlicher mit den Grundlagen der Delaunay-Triangulierung. Abgesehen von dieser häufig verwendeten Triangulierung sind die *Greedy-Triangulierung* und die *Optimale Triangulierung* noch von gewisser Bedeutung. Da sie im Rahmen dieser Arbeit aber keine Rolle spielen, wird auf eine nähere Betrachtung verzichtet und auf die Übersicht von Kumar [49] verwiesen.

Delaunay-Triangulierung

Die Delaunay-Triangulierung basiert auf den Arbeiten von M.G. Voronoi, der mit den Voronoi-Polygonen eine der fundamentalsten und nützlichsten durch irreguläre Gitter definierten Konstruktionen, entdeckt hat [87]. Die Mittelsenkrechten benachbarter Punkte bilden die Voronoi-Polygone eines Satzes irregulär verteilter Punkte. Abbildung 4.5 zeigt, dass die gesamte Fläche durch Voronoi-Polygone bedeckt ist.

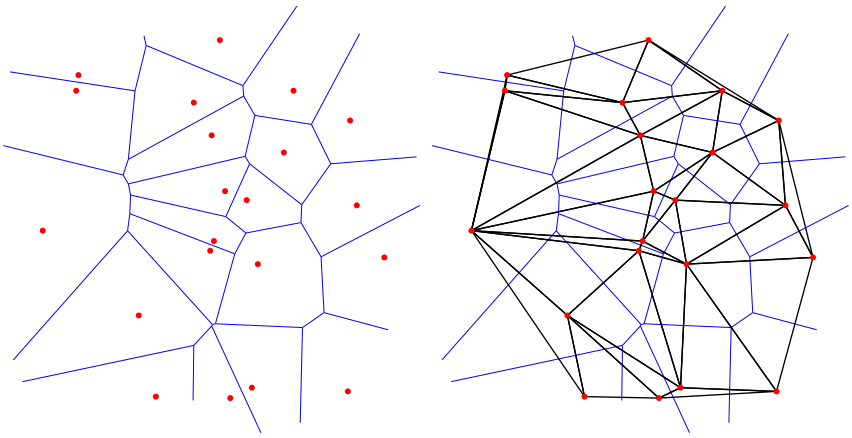


Abbildung 4.5: Voronoi-Polygone einer irregulär verteilten Punktemenge (links) und ihre dazugehörigen Delaunay-Dreiecke.

Die Delaunay-Triangulierung steht in engem Zusammenhang mit den Voronoi-Polygonen. Sie ist nach B. Delaunay benannt, der diesen dualen Zusammenhang als erster nutzte [13]. Verbindet man in einem Voronoi-Diagramm die Mittelpunkte zweier benachbarter Punkte, so erhält man die Delaunay-Triangulierung. Abbildung 4.5 stellt rechts die Delaunay-Triangulerung der Punktewolke zusammen mit den Voronoi-Polygonen dar.

Natürlich lässt sich eine Delaunay-Triangulierung auch direkt bestimmen. Das Umkreis-Kriterium ermöglicht eine Definition ohne Rückgriff auf die Voronoi-Polygone: *Ein Delaunay-Netzwerk in zwei Dimensionen besteht aus sich nicht überlappenden Dreiecken, wobei in keinem der Umkreise dieser Dreiecke ein weiterer Punkt liegt.* Abbildung 4.6 veranschaulicht diese Definition anhand einiger Umkreise.

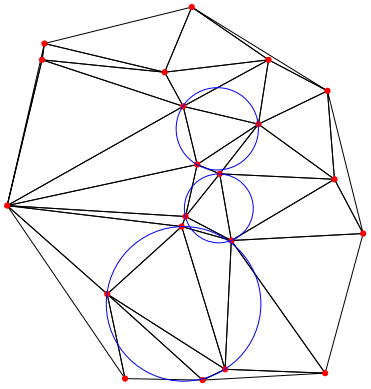


Abbildung 4.6: Veranschaulichung der Umkreisbedingung für eine Delaunay-Triangulierung

Für die Erzeugung von Delaunay-Triangulierungen existieren etliche Methoden [63, 69, 55, 53, 60, 22, 52, 53, 32, 88] in [2], die sich gemäß [20] in fünf Kategorien klassifizieren lassen:

- *two-step-methods*

- Inkrementelle Algorithmen
- *divide and conquer methods*
- *sweep line methods*
- Dreidimensionale Algorithmen

Eine kurze Zusammenfassung bietet hierzu Apel [2] – ausführlicher werden die einzelnen Verfahren von DeFloriani [20] vorgestellt.

In vielen Anwendungen kommt es vor, dass irregulär verteilte Punkte trianguliert werden müssen, dabei jedoch fest vorgegebene Kanten als Randbedingungen eingehalten werden müssen. In der Umgebung der Zwangskanten kann es lokal zu Abweichungen vom Delaunay-Kriterium kommen. Man spricht von einer *constrained Delaunay triangulation*. Durch Einfügen zusätzlicher Punkte, so genannter *Steiner-Punkte*, lässt sich diese in eine echte Delaunay-Triangulierung umwandeln. Dabei wird jede Zwangskante in einzelne kollineare Kantensegmente zerlegt, um die Delaunay-Bedingung einhalten zu können. Steiner-Punkte können auch dazu verwendet werden, zusätzliche Bedingungen an das Netz durchzusetzen. So können dadurch minimale Winkel oder eine maximale Dreiecksfläche innerhalb des Netzes festgelegt werden.

4.2.3 Netzmanipulation

Die Manipulation an Dreiecksnetzen kann aus unterschiedlichen Gründen erforderlich sein. Dabei gilt es die Topologie zu verändern, ohne die Netzstruktur zu verletzen, also ohne Fehltriangulierungen im Netz zu erzeugen. Es lassen sich zwei grundlegende Klassen von Operationen unterscheiden:

- **Netzdezimierung** reduziert die Dreiecksanzahl
- **Netzverfeinerung** erhöht die Dreiecksanzahl

Netzdezimierung

Die Netzdezimierung hat das Ziel, die Dreiecksanzahl in einem Netz zu reduzieren, ohne dabei die Form des Netzes zu verändern. Dabei wird zunächst ein Loch im Netz erzeugt, um dieses dann mit weniger Dreiecken zu füllen. Entsteht das Loch durch Löschen eines Eckpunktes, so beeinflusst die Neutriangulation alle Dreiecke, die den Eckpunkt zum Bestandteil haben, wie Abbildung 4.7 veranschaulicht. Deutlich sichtbar ist, dass es sich um eine lokale Operation handelt.

Häufiger eingesetzt wird die in Abbildung 4.8 skizzierte Kantenkollaps-Operation. Dabei wird eine Kante im Netz durch einen neuen Eckpunkt ersetzt. Die beiden die Kante begrenzenden Eckpunkte werden mit all ihren benachbarten Dreiecken zu einem neuen Eckpunkt zusammengezogen. Dabei kollabieren die beiden an der Kante anliegenden Dreiecke zu Kanten.

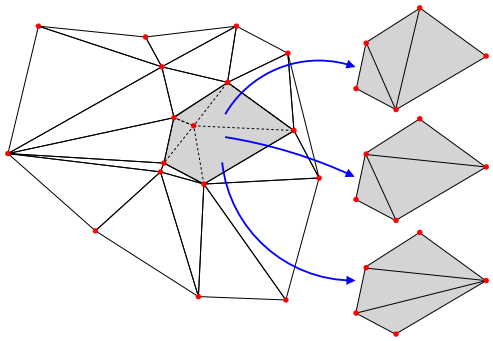


Abbildung 4.7: Netzdezimierung: verschiedene Varianten, ein erzeugtes Loch zu füllen.

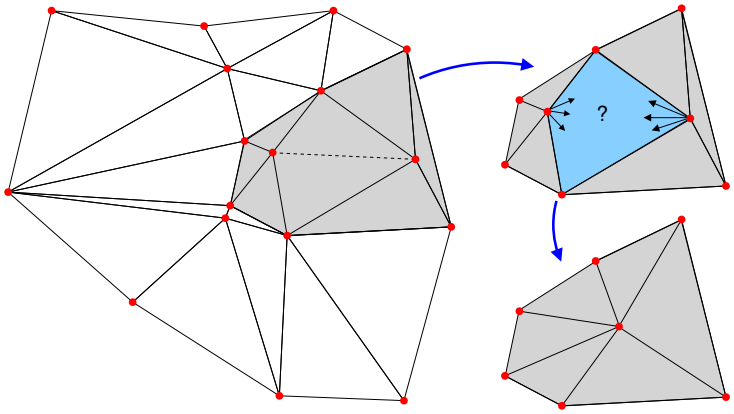


Abbildung 4.8: Netzdezimierung durch Kantenkollaps: Die Position des neuen Eckpunktes muss in der hellblau gekennzeichneten Region liegen.

Dieses Verfahren wird bevorzugt, obwohl es mehr Dreiecke pro Operation betrifft, da die freie Platzierung des neuen Eckpunktes leichter handhabbar ist als die freie Wahl neuer Kanten wie im vorher vorgestellten Verfahren.

Weiterhin wird *Dreieckskollaps* als Erweiterung des Kantenkollaps zum Beispiel von Hamann [36, 33] eingesetzt. Die Schwierigkeit der lokalen Operation steigt jedoch mit der Anzahl beteiligter Dreiecke, da zusätzliche Tests erforderlich werden, um Netzdegenerierung zu verhindern. Eine eher generelle Erweiterung des Konzeptes ist die Neutriangulierung zusammenhängender Subnetze aus mehreren Dreiecken. Dieser Ansatz wird von Heckbert und Garland in ihrer Übersicht [36] als *patch decimation* bezeichnet und ebenfalls in vielen Arbeiten verwendet.

Drei unabhängige Fragestellungen lassen sich bei den meisten Dezimierungsverfahren identifizieren und und können auf unterschiedlichste Weise gelöst werden:

- Lokale Dezimierungsmethode (z. B. Kantenkollaps)
- Fehlerüberwachung (z. B. wie wird sichergestellt, dass das Netz dem Ausgangsnetz ähnlich bleibt)
- Qualitätsmaß für Dreiecke (z. B. keine ungünstig geformten Dreiecke)

Netzverfeinerung

Diese Operation fügt neue Eckpunkte und damit neue Dreiecke in das Netz ein. Dabei kommt als Triangulierungsmethode meist die inkrementelle Delaunay-Triangulierung zum Einsatz. Die Verfeinerung von Netzen ohne Bezug zu Basisdaten kommt einer zweidimensionalen Interpolation gleich. Sie kann nötig werden, wenn eine Region in einem Netz gelöscht werden muss, deren Grenze nicht mit Netzkanten zusammenfällt.

Sehr viel wichtiger ist die Verfeinerung von Netzen für den initialen Netzaufbau. Eine Triangulierung basiert häufig auf Basisdaten, die mit hoher Auflösung vorliegen. Oft ist es unnötig, die gesamten Basisdaten zu berücksichtigen, wenn eine gute Triangulierung bereits durch geschickte Auswahl von Eckpunkten innerhalb der Basisdaten erreicht werden kann. Daher werden Netze oft inkrementell erzeugt. Ausgehend von einer groben Approximation durch wenige Dreiecke werden schrittweise neue Eckpunkte eingefügt, um den Approximationsfehler zu reduzieren. Mit dieser Problematik beschäftigt sich Kapitel 4.3 ausführlich.

Sonstige Operationen

Eine Vielzahl von Verfahren dienen der Veränderung von Dreiecksnetzen mit dem Ziel, ihre Qualität zu verbessern. Dabei kann 'Qualität' in diesem Kontext sowohl für netzimmanente Qualitätsmaße wie minimale Winkel, als auch für die Qualität der Approximation stehen, die ein Netz zu seinen Basisdaten erreicht. Zwei Operationen sollen hier kurz erläutert werden:

Kantenflip Diese Operation verändert die gemeinsame Kante zweier benachbarter Dreiecke. Die Kante wird zwischen den beiden Diagonalen des einschließenden Vierecks 'umgeklappt'. So kann auf einfache Weise die Qualität eines Netzes verändert werden, wie nebenstehende Abbildung verdeutlicht.

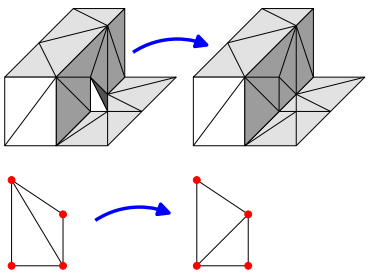


Abbildung 4.9: Kantenflip

Verschieben von Eckpunkten innerhalb der Netzebene Hier werden Eckpunkte in einem Netz entlang der Netztangentialen verschoben, um damit eine günstigere Triangulierung für Folgeschritte zu erhalten. Diese Operation wird zum Beispiel von Scarlatos und Pavlidis [75] verwendet, um Dreiecke in Bereichen hoher Krümmung zu verkleinern und danach koplanare Dreiecke zu fusionieren.

4.2.4 Oberflächenrekonstruktion durch Dreiecksnetze

Das Problem der Oberflächenrekonstruktion stellt sich als Berechnung eines digitalen Oberflächenmodells *(DSM Digital Surface Model)* als diskrete Approximation einer Oberfläche auf der Grundlage einer finiten Menge abgetasteter Datenpunkte $\mathcal{S}$ dar. Auf Grund der 1-zu-1 Beziehung von Datenpunkten und Punkten der zweidimensionalen Untermenge $\mathcal{D}$ lässt sich die Triangulierung auf eine Unterteilung der Untermenge $\mathcal{D}$ in Dreiecke und eine Aufstellung einer analytischen Beziehung der Dreiecke zu den Funktionswerten reduzieren.

Die Datenpunkte $\mathcal{S}$, auf deren Grundlage ein DSM aufgebaut wird, können entweder auf einem äquidistanten Gitternetz oder an irregulär verteilten Positionen in $\mathcal{D}$ liegen. Es gibt pro Stützpunkt nur einen Funktionswert. Abbildung 4.10 stellt die Disparitätenkarte der Testaufnahme *Tsukuba* als $2\frac{1}{2}$D-Netz trianguliert dar. Dabei wird der Disparitätenwert, der für ein Pixel vorliegt, als Funktionswert der entsprechenden Position auf dem äquidistanten 2D-Gitternetz der Bildkoordinaten interpretiert.

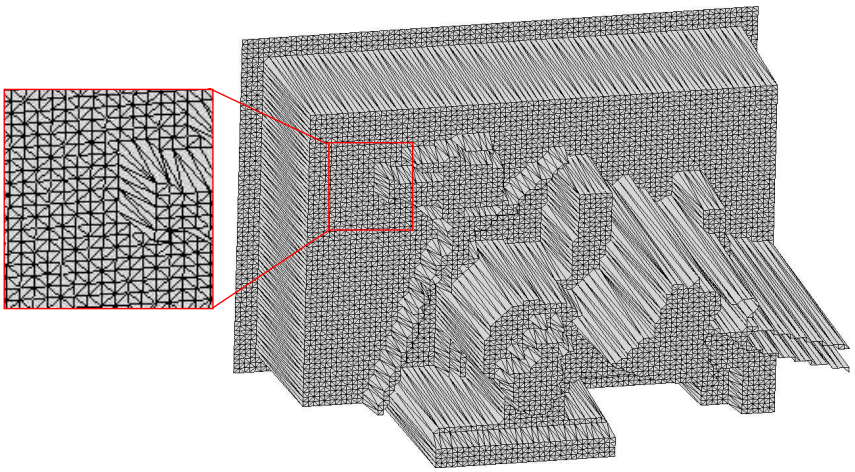


Abbildung 4.10: Disparitätenkarte von *Tsukuba* als $2\frac{1}{2}$D-Netz mit äquidistanten Basisdaten.

Für die hier vorliegende Anwendung ist dies der wichtigste Netztyp.

4.3 Von Tiefenkarten zu Dreiecksnetzen

Nach der allgemeinen Klärung verschiedener Begriffe und Konzepte im vorangegangenen Abschnitt soll in diesem Kapitel die spezielle Anwendung im Vordergrund stehen. Zu diesem Zweck werden zunächst die besonderen Randbedingungen innerhalb des Projektes und die sich daraus ergebenden Konsequenzen vorgestellt. Im nächsten Abschnitt werden die Eigenschaften der Eingangsdaten etwas genauer betrachtet und die eingesetzten Filter beschrieben.

4.3.1 Randbedingungen aus dem Projektrahmen

Folgende Randbedingungen haben prägenden Einfluß auf die eingesetzten Konzepte:

- Ein Stereo-Kamerasystem dient als einziger Sensor im System. Die daraus resultierenden Geometriedaten besitzen einige Eigenarten, die für die Triangulierung und Fusion wichtig sind. Dies unterscheidet das System von vielen ähnlichen Projekten, in denen ein Laserscanner für die Datenerfassung der Szenengeometrie zum Einsatz kommt.
- Die Anwendung Telepräsenz bedingt eine inkrementelle und fortschreitende Modellbildung, -verifikation und -veränderung.
- Der Einsatz von Standard-Algorithmen der Computergrafik basierend auf texturierten Polygonen und die Verwendung von Original-Bildinformation zur Texturierung erfordert (zeitlich) möglichst stabile Polygone, damit Texturen aus älteren Aufnahmen weiterverwendet werden können. Das heißt, dass Veränderungen der Netztopologie zu einem Verlust von Texturinformation führen würden und somit zu vermeiden sind.

4.3.2 Struktur der Eingangsdaten

Abbildung 4.11 zeigt eine typische Disparitätskarte einer einfachen Szene. Ihre besonderen Eigenschaften sollen nun einer genauen Untersuchung unterzogen werden.

- **Verdeckungen**
  Wie am Ende des Abschnitts 3.4.3 erläutert, führen Verdeckungen in einem Stereosystem zu Bildbereichen in der Disparitätenkarte, für die keine Information verfügbar ist. Dies ist bei der Triangulierung zu berücksichtigen.
- **Löcher**
  Der Rechts-Links-Konfidenztest (vgl. Abschnitt 3.4.5) erzeugt Löcher in der Disparitätenkarte, für die keine Information vorliegt. Diese Löcher lassen sich ohne eine Analyse der sie umgebenden Disparitäteninformation nicht von Verdeckungen unterscheiden. Im Gegensatz zu Verdeckungen lassen sie sich aber in den meisten Fällen durch Interpolation aus benachbarter Disparitätsinformation korrekt rekonstruieren.
- **Ausreißer**
  Fehlkorrespondenzen, die beim Rechts-Links-Konfidenztest in beiden Aufnahmen auftreten, werden nicht ausgefiltert. Sie können durch repetitive Strukturen oder Glanzlichter in den Aufnahmen auftreten.
- **Entfernungsabhängige Genauigkeit**
  Betrachtet man den Prozess zur Gewinnung der Subpixel-Genauigkeit (vgl. Abschnitt 3.3.4), so zeigt sich, dass Kamerarauschen die Einträge der Kostenmatrix und damit die korrekte Bestimmung der Disparität durch Parabelapproximation

beeinflusst. Die Genauigkeit der Disparitätenkarte ist also nicht abhängig vom Disparitätswert, sehr wohl aber von der Bildinformation und den Sensoreigenschaften. Für die Aufnahme *Cones* beträgt die Standardabweichung der Messergebnisse von der *ground truth* etwa 0.167 Pixel Disparität, wenn Ausreisser ausgefiltert werden. Durch die Rekonstruktion der Tiefe aus der Disparität wird diese Genauigkeit jedoch entfernungsabhängig, da für die Rekonstruktion der Zusammenhang $d \propto 1/z$ gilt.

4.3.3 Filterung der Eingangsdaten

Dieser Abschnitt stellt verschiedene implementierte Filtermodule für Disparitätenkarten und ihr jeweiliges Anwendungsgebiet vor. Die Filter können, abhängig vom Szenario und den Randbedingungen, in unterschiedlicher Kombination genutzt werden. So kann bei vorwiegend stetigen Oberflächenverläufen, wie sie zum Beispiel in der endoskopischen Herzchirurgie vorkommen, wesentlich agressiver gefiltert werden als bei typischen Innenraumaufnahmen mit großen Tiefensprüngen innerhalb der Szene. Nebenstehende Disparitätenkarte (Abb. 4.11) dient dabei als Ausgangslage, anhand derer die Effekte einzelner Filter dargestellt werden. Zur Verdeutlichung der Filterwirkung ist zusätzlich der Inhalt zweier Zeilen der Disparitätenkarte dargestellt.

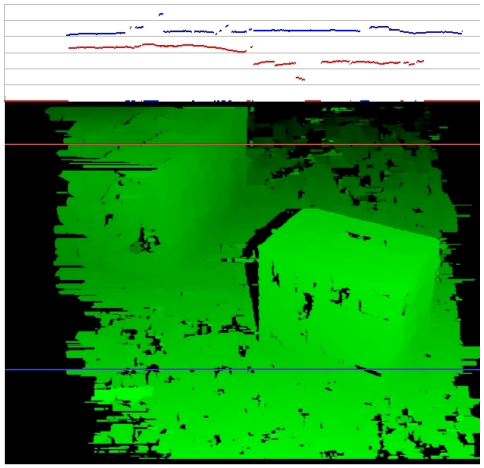


Abbildung 4.11: Ungefilterte Disparitätenkarte der Szene *Telekiste*

Beschränkung der Szenenausdehung Disparitäten kleiner als eine Mindestschranke und größer als ein Maximalwert werden entfernt. Natürlich sollte diese Einschränkung bereits in der Wahl des Suchbereichs der Korrespondenzfindung getroffen werden, um unnötige Berechnungen zu vermeiden. Doch gibt es auch Gründe, diese Einschränkung erst im Disparitätsraum, in Kamerakoordinaten oder Weltkoordinaten zu treffen.

Ausreißerfilterung Ausreißer sind häufig auf kleine Regionen oder Abschnitte einzelner Bildzeilen begrenzt. Sie lassen sich erkennen, indem man eine Mindestausdehnung zusammenhängender Regionen in der Tiefenkarte und einen Maximalabstand zu benachbarten Regionen definiert. Da zusammenhängende Regionen identifiziert werden müssen, wird dafür ein Clustering-Algorithmus gewählt.

Eine zusammenhängende Region R_k ist dadurch definiert, dass für jedes ihrer Pixel p_i gilt

$$\forall p_i \in R_k \exists\ p_j \in R_k : |p_i - p_j| < d \qquad i \neq j \quad (4.1)$$

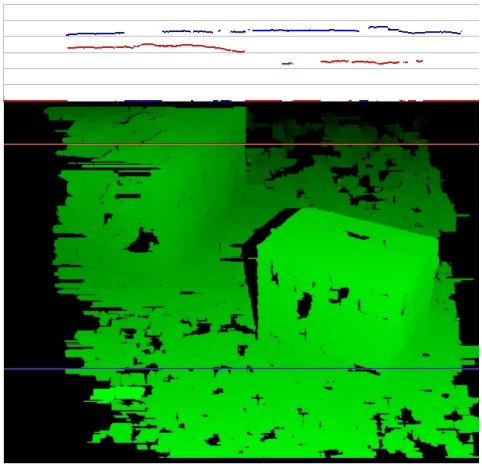


Abbildung 4.12: Effekt der Ausreißerfilterung

Die Berechnung dieser Regionen nutzt eine Liste noch zu untersuchender Pixel L und ein Array r der Maße Bildbreite x Bildhöhe, in dem die Zugehörigkeit zu einer Region R_k abgelegt wird. Der Algorithmus lautet wie folgt:

```
solange bis alle gültigen Punkte zu einer Region zugeordnet sind
   suche einen gültigen Punkt p_i mit r[i] = 0
   füge p_i zu L hinzu
   erhöhe die aktuelle Regionennummer k

   solange bis L leer ist
      Hole nächstes Listenelement p_i aus L
      für jeden gültigen Nachbarn p_j von p_i mit r[j] = 0
         wenn |p_i − p_j| < d dann
            r[j] = k
            füge p_j zu L hinzu
```

Dabei bezeichnet d den maximalen Abstand, für den zwei Punkte zu derselben Region zählen und *Nachbarn* einen definierbaren Suchbereich um einen Punkt herum.

Das Array r enthält nach der Regionenzuordnung für jedes Pixel eine Regionennummer. Im nächsten Schritt werden die Pixel pro Region gezählt, was einer Histogrammbildung auf r gleichkommt. Regionen unterhalb einer definierbaren Größe werden aus der Tiefenkarte entfernt. Abbildung 4.12 zeigt den Effekt für einen minimalen Abstand von 1 cm, einen Nachbarschaftssuchbereich von ±5 Pixeln und eine minimalen Regionengröße von 200 Pixeln.

Konturenglättung Feinste Strukturen der Kontur der Disparitätenkarte führen zu einer großen Anzahl von Polygonen entlang ihres Randes, da die Kontur präzise durch Polygone modelliert werden muß. Es ist daher sinnvoll, die Konturen durch eine einfache *Opening*-Operation zu beschneiden und damit zu glätten. In Abbildung 4.13 wurde ein Radius von 2,5 Pixel für diese Operation gewählt.

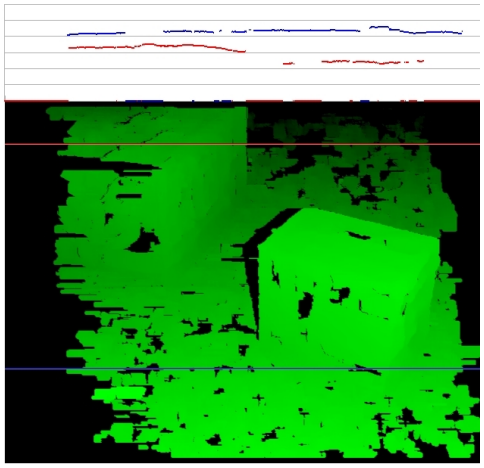


Abbildung 4.13: Konturenglättung

Verdeckungen und Löcher Kleine durch Ausreißerfilterung oder Rechts-Links-Konfidenztest entstandene Löcher lassen sich durch einen einfachen Medianfilter (angewendet auf die Umgebung der Löcher) füllen. Dabei ist der Medianfilter dem Mittelwertfilter vorzuziehen, da sonst in Bereichen von Disparitätensprüngen in der Umgebung eines Lochs neue Disparitätswerte interpoliert werden, die in der Umgebung nicht auftreten. Der Medianfilter dagegen erzeugt ausschließlich Werte, die in der Nachbarschaft ohnehin vorkommen. Abbildung 4.14 stellt das Ergebnis dieser Filterung dar.

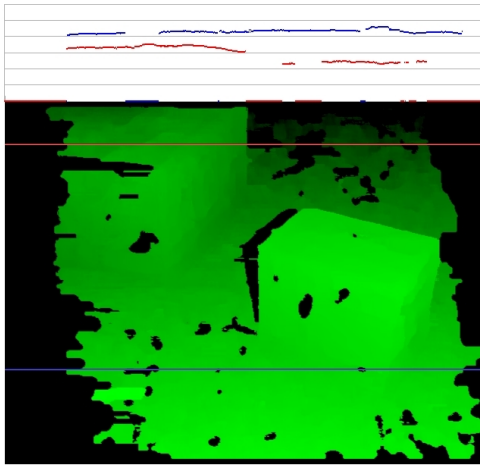


Abbildung 4.14: Medianfilter

Große Löcher in der Szene können vor einer Triangulierung auf unterschiedliche Weise behandelt werden:

1. Das Loch wird durch eine Triangulierung seines Randes geschlossen.
2. Das Loch wird nicht trianguliert – es bleibt als Loch im resultierenden Netz.
3. Das Loch wird durch morphologische Operationen in der Disparitätenkarte geschlossen.
4. Eine Extrapolation der Nachbarpolygone schließt das Loch

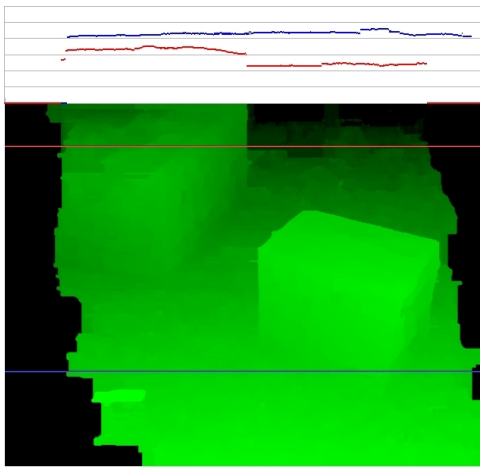


Abbildung 4.15: Dual-Rank-Filter

Gerade größere Löcher müssen abhängig vom Szenario unterschiedlich behandelt werden. Gilt es, aus wenigen Aufnahmen bereits eine überzeugende texturierte Darstellung zu erzeugen, müssen die Löcher gefüllt werden, da sonst die Darstellungsqualität sehr leidet. Besteht bereits ein Szenenmodell, sollte der Vergleich zwischen neuer Aufnahme und bekanntem Modell (vgl. Abschnitt 4.5) vor dem Füllen

der Löcher durchgeführt werden, um keine extrapolierte Information innerhalb der Löcher zu einem Vergleich heranziehen zu müssen. Unter den obengenannten Varianten führt Methode 1 zu deutlichen Artefakten an Tiefensprüngen. Variante 4 ist zwar prinzipiell überlegen, jedoch stark von der erreichten Netzqualität und der Netzgeometrie in der Umgebung des Loches abhängig. Betrachtet man die Disparitätenkarte als Bild, sind Grauwert-basierte morphologische Operationen gut geeignet, Bildbereiche zu schließen. Der *Dual-Rank*-Operator [15] stellt einen geeigneten Operator für diesen Zweck dar.

Der *Dual-Rank*-Operator führt eine zweimalige *Rank*-Operation mit unterschiedlichen Parametern aus. Die *Rank*-Operation führt eine wählbare Maske über jedes Pixel und sortiert die Pixel ähnlich einem Median-Operator in aufsteigender Reihenfolge ihrer Grauwerte. Ein wählbarer Parameter legt fest, welcher Eintrag aus dieser Liste ausgewählt wird. Beim *Dual-Rank*-Operator wird der *Rank*-Operator im ersten Durchlauf mit n% und in einem zweiten Durchlauf mit 1-n% angewendet. Der *Dual-Rank*-Operator entspricht damit für $n = 0\%$ einem *Opening*-Operator, für $n = 50\%$ einem Median-Operator und für $n = 100\%$ einem *Closing*-Operator [34].

Wird der *Dual-Rank*-Operator mit einem Wert von 5% und einer ausreichend großen Maske auf eine Disparitätenkarte angewendet, so werden Löcher, die nicht größer als die Maske sind, mit dem kleinsten Disparitätswert in ihrer Nachbarschaft gefüllt. Da Löcher in der Tiefenkarte mit einer Disparität von -1 gekennzeichnet werden, muss $n > 0$ gewählt werden, da Löcher sonst mit dem kleinsten vorkommenden Wert, mithin also -1, gefüllt werden würden. Dabei wird das Ergebnis der Filterung nur für die Löcher verwendet. Die übrige Disparitätenkarte bleibt also unangetastet. Beispiele für den Effekt des Operators zeigt Abbildung 4.15.

Kantenerhaltende Mittelwertsfilterung Um den Effekt der entfernungsabhängigen Genauigkeit der Tiefenkarte zu reduzieren, wird ein Gaußfilter auf die Disparitätenkarte angewendet. Um dabei Bereiche mit hohen Disparitätsgradienten aus der Filterung auszuschließen, wird eine Kombination aus einem Filter zur Kantendetektion mit mehreren Gaußfiltern mit unterschiedlicher Maskengröße eingesetzt.

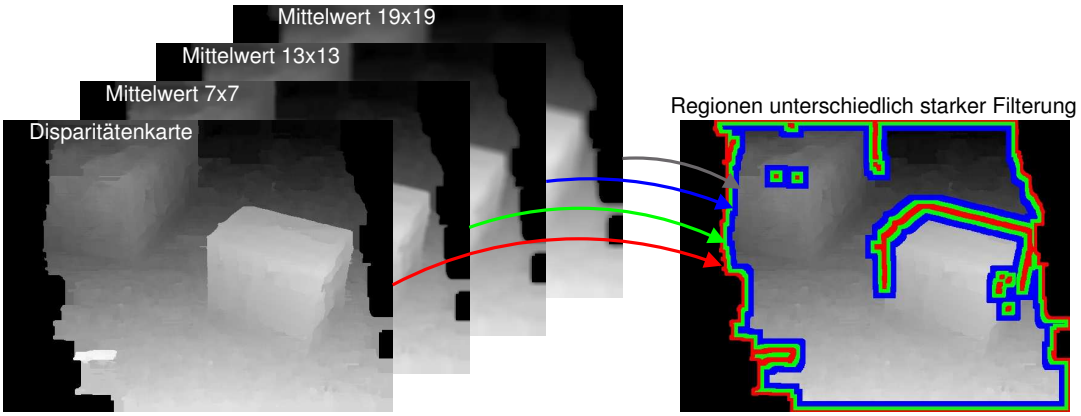


Abbildung 4.16: Funktionsweise der kantenerhaltenden Mittelwertfilterung: Abhängig vom Abstand von Diskontinuitäten im Disparitätsbild wird die Fenstergröße des Mittelwertfilters gewählt.

In der Nähe von Löchern und Orten mit großem Disparitätsgradienten sollte keine Filterung durchgeführt werden. Mit steigendem Abstand sollte immer stärker geglättet werden. Zu diesem Zweck wird ein Mittelwertfilter mit quadratischer Maske mit drei unterschiedlichen Maskengrößen von 7,13 und 19 Pixeln auf das Disparitätenbild angewendet. Dadurch entstehen drei geglättete Disparitätenbilder $\mathcal{M}_1, \mathcal{M}_2$ und $\mathcal{M}_3$, wie sie in Abbildung 4.16 links dargestellt sind.

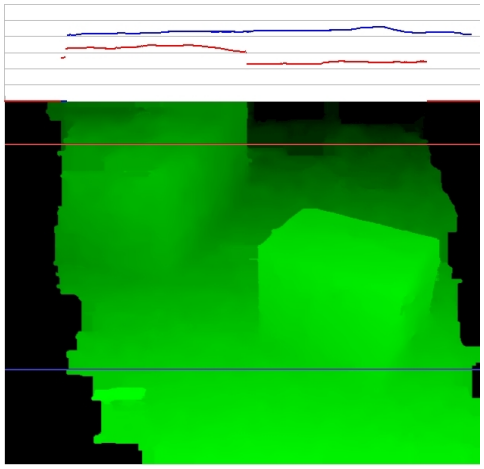


Abbildung 4.17: Ergebnis der kantenerhaltenden Mittelwertsfilterung

Nun wird eine Maske erzeugt, die alle gültigen Bildbereiche abzüglich Disparitätensprüngen enthält. Sie wird durch zweidimensionale Sobelfilterung mit nachfolgender Schwellwertbildung erzeugt. Inkrementelle Erosion dieser Maske mit einer den Mittelwertfiltern entsprechenden Fenstergröße ermöglicht es, den Abstand von Diskontinuitäten zu bestimmen. Um nun die Disparitätenkarte selektiv zu filtern, werden abhängig vom Abstand zu den Diskontinuitäten die Ergebnisse der Mittelwertfilter verwendet. Parametrierbar sind bei diesem Prozess der Schwellwert, ab dem eine Kante detektiert wird, und die Maskengröße m.

4.3.4 Vollständige Triangulierung

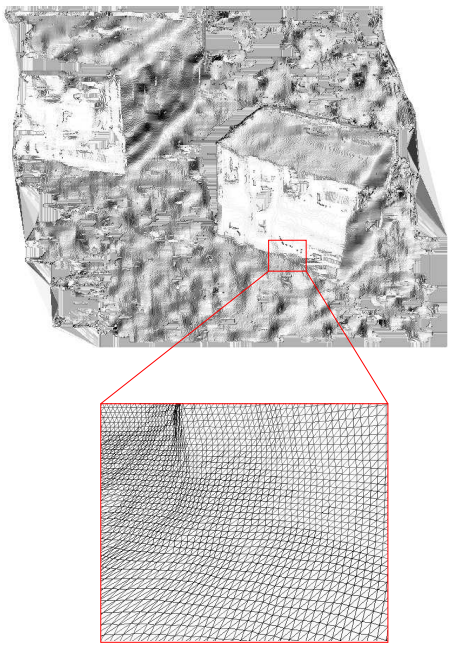


Abbildung 4.18: Maximal feine Triangulierung von *Telekiste*

Die vollständige Triangulierung ist die denkbar einfachste Art, $2\frac{1}{2}$D-Daten in ein Dreiecksnetz umzuformen. Da die Daten in einer Disparitätenkarte auf einem zweidimensionalen Gitternetz definiert sind, können jeweils drei benachbarte Pixel zu einem Dreieck verbunden werden. Enthält die Disparitätenkarte Löcher, so existieren in diesen Bereichen im Raster keine direkten Nachbarn. Die Triangulierung muss dort dann auf andere Weise (z. B. Delaunay-Triangulierung) erfolgen. Damit ist die Topologie des Netzes festgelegt. Durch eine Transformation jedes Eckpunktes aus dem Disparitätenraum in den dreidimensionalen Raum mittels 3D-Rekonstruktion entsteht das dreidimensionale Netz (vgl. Abschnitt 3.2.5). Da zur Triangulierung ausschließlich zweidimensionale Information aus dem Raster genutzt wird, kann es zu ungünstigen Triangulierungen kommen. Die Qualität des Netzes, also das Maß, wie gut das Netz die zugrundeliegenden Disparitätsdaten repräsentiert, kann durch *Edge-Flip*-Operationen deutlich verbessert werden. Abbildung 4.18 zeigt die feinstmögliche Triangulierung der Szene *Telekiste*. Sie wurde als inkrementelle Delaunay-Triangulierung auf dem x-y-Raster erstellt und enthält rund 489.000 Dreiecke und 245.000 Eckpunkte. Natürlich ist es aus Laufzeitgründen nicht sinnvoll, regulär auf einem Gitternetz angeordnete Punkte durch Delaunay-Triangulierung zu triangulieren, die Triangulierungsmethode spielt jedoch für das hier vorgestellte Prinzip keine Rolle.

Bedingt durch die Triangulierungsmethode werden konkave Bereiche der Kontur durch Dreiecke geschlossen. Da dieser Effekt bei verschiedenen Triangulierungsmethoden auftritt, wird er im Anschluss an die Vorstellung der Triangulierungsmethoden im Abschnitt 4.3.7 besprochen.

Die vollständige Triangulierung führt zwar schnell zu einen Dreiecksnetz, das resultierende Netz besteht jedoch aus einer sehr großen Anzahl von Dreiecken, was für nachfolgende Operationen ungünstig ist. So führen kleine Dreiecke zu großem Speicherbedarf, Netzmanipulationen sind sehr aufwändig und Texturierung sehr kleiner Dreiecke ist ungünstig. Vorteilhaft dagegen ist die Tatsache, dass die vollständige Triangulierung die komplette Information der Disparitätenkarte nutzt und somit der geringstmögliche Fehler erreicht wird.

Dieses Verfahren wird auf Grund der genannten Nachteile eher selten eingesetzt. Lee beschreibt in seiner Arbeit [54] eine darauf aufbauende Triangulierungsmethode. Nach der initialen Feintriangulierung wird in jedem Durchgang der Eckpunkt entfernt, der den geringsten Fehler im Netz erzeugt. Das resultierende Loch benachbarter Dreiecke wird

durch Delaunay-Triangulierung neu verknüpft. Das Verfahren ist, wie zu erwarten, von hoher Komplexität und von hohem Speicherbedarf [36].

4.3.5 Adaptive Triangulierung

Die adaptive Triangulierung basiert auf dem Prinzip, die Disparitätenkarte auf ihre Homogenität zu untersuchen und abhängig von ihren lokalen Eigenschaften gröber oder feiner zu triangulieren. Im Rahmen dieses Projekts wurde eine mehrstufige diskrete Variante implementiert. Es wird zunächst der lokale Disparitätsgradient bestimmt und jedes Pixel durch Schwellwertbildung in eine von vier Kategorien eingeteilt. Nun werden vier verschiedene generische Eckpunkteraster gebildet. Abhängig von der Kategorie, in die ein Pixel fällt (d. h. von seinem quantisierten Gradienten), wird ein Eckpunkt aus dem entsprechenden Raster hinzugefügt. Dies führt an Tiefensprüngen zu einer maximal feinen Rasterung und abhängig vom Disparitätsgradienten zu einem entsprechend gröberen Raster. Ein typisches Ergebnis stellt Abbildung 4.19 dar.

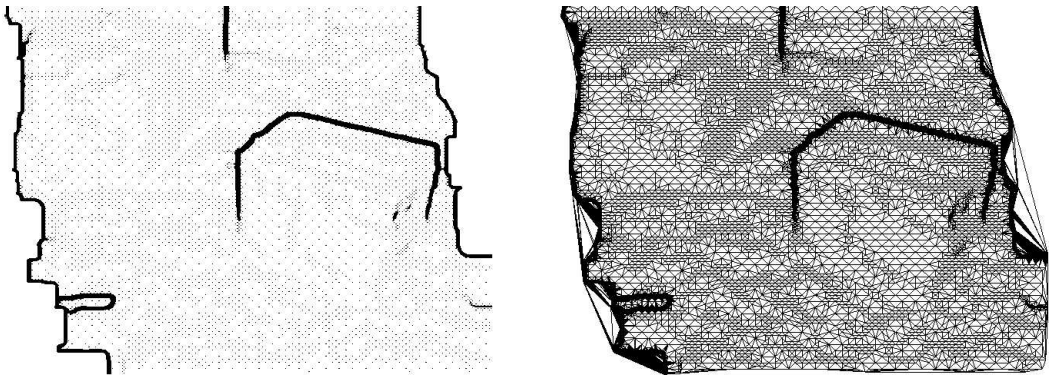


Abbildung 4.19: Resultierende Eckpunkte der adaptiven Triangulierung und das dazugehörige Dreiecksnetz.

4.3.6 Verfeinernde Triangulierung

Die verfeinernde Triangulierung ist ein iteratives Verfahren, das mit einer minimalen Triangulierung beginnt und in jedem Durchgang einen oder mehrere Eckpunkte hinzufügt, bis ein definierter Fehler oder eine definierte Anzahl von Dreiecken erreicht ist. Normalerweise werden inkrementelle Triangulierungsmethoden wie die inkrementelle Delaunay-Triangulierung verwendet, um die Triangulierung während des Prozesses zu erhalten. Bei dieser Triangulierung lauet die zentrale Frage:

An welchem Punkt der Disparitätenkarte wird ein neuer Eckpunkt eingefügt, um den Approximationsfehler weiter zu reduzieren?

Im Falle $2\frac{1}{2}$-dimensionaler Eingangsdaten wird hierfür meist der Punkt des maximalen vertikalen Fehlers verwendet.

Nebenstehende Abbildung 4.20 veranschaulicht die Vorgehensweise anhand eines eindimensionalen Problems. Um in jedem Durchgang den Punkt des größten Fehlers zu finden, muss der vertikale Abstand zwischen Netz und Basisdaten in jedem Durchgang bestimmt werden. Aus der großen Anzahl verschiedener verfeinernder Triangulierungsmethoden (siehe Überscht in [36]) wurde für die vorliegende Arbeit die Methode von Garland und Heckbert [29] ausgewählt. Die Methode stellt eine recht klare Umsetzung der Problematik dar, basiert auf der Grundlagenarbeit von De Floriani [23, 21] und ist als Programmcode verfügbar. Sie basiert auf einer sortierten Liste von Punktkandidaten mit jeweils assoziiertem Fehler. Dadurch lässt sich der Punkt des höchsten Fehlers schnell auffinden. Für jeden hinzugefügten Eckpunkt werden nur die Einträge der sortierten Liste aktualisiert, die in der Nachbarschaft des neuen Eckpunktes liegen. Dabei erfolgt nach einer Fehlerberechnung sofort eine neue Einsortierung des Eintrags in der Liste.

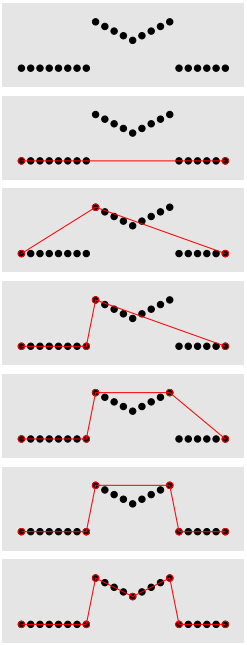


Abbildung 4.20: Verfeinernde Triangulierung

Es wird mit einem, die x-y-Ebene der Disparitätenkarte umschliessenden Dreieck begonnen. Alle gültigen Pixel der Disparitätenkarte werden in die sortierte Liste eingetragen. In jedem Durchgang wird nun am Punkt des größten Fehlers ein Eckpunkt hinzugefügt, das Netz durch inkrementelle Delaunay-Triangulierung aktualisiert und der Fehler aller betroffenen Pixel neu berechnet. Bei erreichen einer maximalen Dreiecksanzahl oder eines gegebenen Approximationsfehlers ist die Triangulierung beendet und das initiale Hilfsdreieck wird aus der Triangulierung entfernt.

Abbildung 4.21 zeigt das Resultat für die Aufnahme *Telekiste*. Das Netz enthält 10.000 Dreiecke.

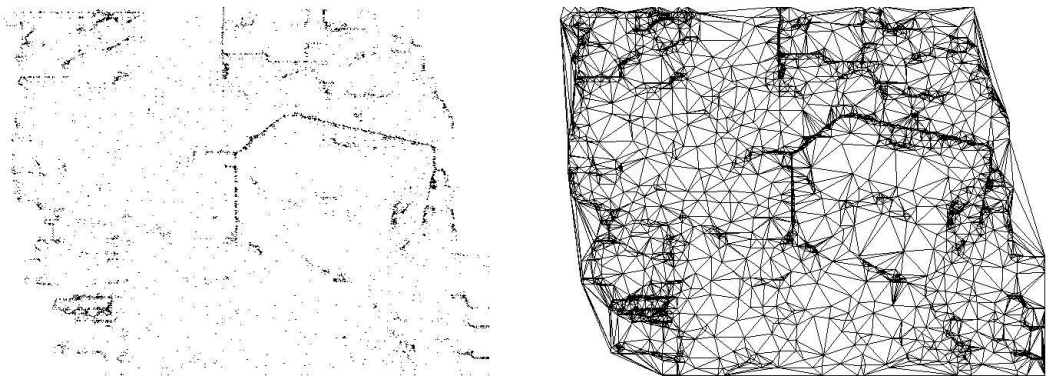


Abbildung 4.21: Resultierende Eckpunkte der verfeinernden Triangulierung und das dazugehörige Dreiecksnetz.

4.3.7 Bereinigung des Dreiecksnetzes

Wie an den verschiedenen Illustrationen zu den oben aufgeführten Triangulierungsmethoden ersichtlich ist, werden konkave Bereiche der Kontur der Disparitätenkarte durch Dreiecke geschlossen. Dasselbe gilt natürlich auch für eventuelle Löcher innerhalb der Disparitätenkarte. Unter dem Begriff *Bereinigung* werden in diesem Abschnitt verschiedene Verfahren vorgestellt, mit denen derartige Probleme nach der Triangulierung behoben werden können. Da bei allen drei vorgestellten Triangulierungsmethoden die Triangulierung zunächst im $2\frac{1}{2}$-dimensionalen Disparitätenraum erfolgt, lässt sich die Bereinigung ebenfalls in diesem Raum durchführen. Dadurch ist die Topologie des Netzes auf dem zugrundeliegenden x-y-Gitter definiert, und die Überlegungen lassen sich auf zwei Dimensionen beschränken.

Konkavitäten in der Kontur Der Bereich gültiger Disparitätenwerte lässt sich durch einfache Schwellwertbildung im Disparitätsbild in Form einer zweidimensionalen Region gewinnen. Die Umrandung dieser Region, also die zukünftige Grenze des Netzes wird nun pixelweise vollständig in Eckpunkte und dazwischenliegende Kanten umgewandelt. Diese Kanten werden als Zyklus gerichteter Zwangskanten zum Netz hinzugefügt und durch Delaunay-Triangulierung (*constraint Delaunay triangulation*) mit dem Netz verknüpft. Nach Abschluss der Triangulierung kann nun ausgehend von den äußeren (in der verfeinernden Triangulierung als Startdreieck hinzugefügten) Kanten das Netz beschnitten werden, bis die gerichtete Zwangskante als äußere Begrenzung übrig bleibt.

Löcher innerhalb des Netzes Löcher werden durch morphologische Operationen auf der Region gültiger Pixel der Disparitätenkarte identifiziert. Ihre Umrandung wird ebenfalls als gerichtete Folge von Zwangskanten zum Netz hinzugefügt, wobei Eckpunkte speziell markiert werden. Nach Abschluss der Triangulierung können nun sämtliche Dreiecke gelöscht werden, deren drei Eckpunkte entsprechend markiert sind.

Tiefensprünge Ein naheliegendes Verfahren identifiziert Tiefensprünge in der Disparitätenkarte mit Hilfe eines Kantenfilters und entfernt alle Dreiecke, die auf einer so detektierten Kante liegen. Diese Vorgehensweise hat aber den Nachteil, dass Dreiecke die mit einer Fläche < 1 Pixel auf einer Kante liegen, unzureichend detektierbar sind. Da aber gerade an Tiefensprüngen lange dünne Dreiecke auftreten, erlaubt die Methode keine zuverlässige Öffnung der Tiefensprünge.

So wurde eine weitere Methode entwickelt, um das Netz an Tiefensprüngen zu öffnen, die im Folgenden erklärt werden soll.

Da das Dreiecksnetz im Disparitätenraum erzeugt wird, kann hier bereits eine Öffnung der Tiefensprünge, die eine bestimmte Disparität d überschreiten, vorgenommen werden. Dabei ist es nützlich, sich in Erinnerung zu rufen, dass das Dreiecksnetz im zweidimensionalen Unterraum $\mathcal{D}$, also der x-y-Ebene definiert ist, und die z-Achse dem Funktionswert,

also der Disparität entspricht. Die zu entfernenden Dreiecke zeichnen sich durch folgende beiden Eigenschaften aus:

- Ihre Abmessungen im zweidimensionalen Unterraum $\mathcal{D}$, also der x-y-Ebene sind klein (wenige Pixel Kantenlänge).
- Sie besitzen zwei Kanten, die eine Disparität $> d$ überspannen.

Zur Visualisierung dieser Tatsache kann Abbildung 4.10 dienen, die ein $2\frac{1}{2}$-D-Netz darstellt. Deutlich sichtbar sind die langen, schmalen Dreiecke an Tiefensprüngen. Abbildung 4.22 stellt das Ergebnis für die Testszene dar.

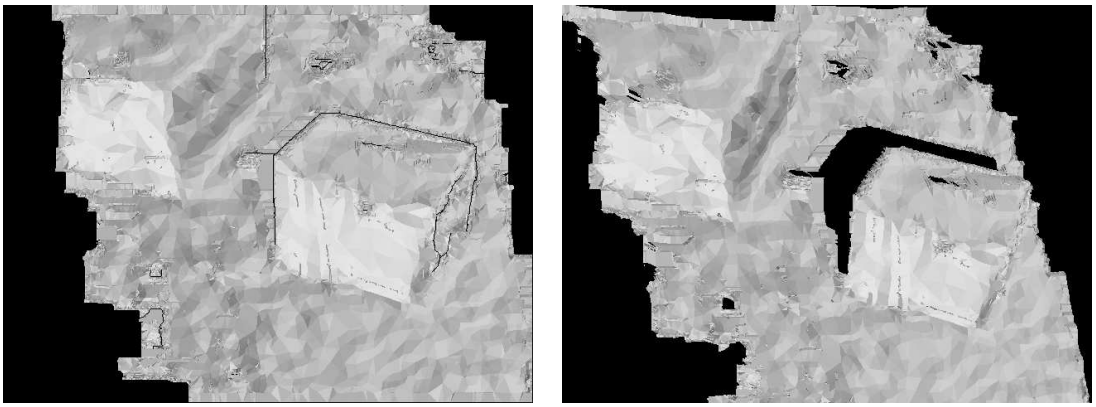


Abbildung 4.22: Resultat der verfeinernden Triangulierung mit geöffneten Tiefensprüngen aus verschiedenen Ansichten.

4.4 Dezimierung

Automatische Dezimierung von Dreiecksnetzen erzeugt aus einer Triangulierung einer Szene eine neue Triangulierung, die der ursprünglichen durch das Netz repräsentierten Form ähnlich ist, aber aus wesentlich weniger Dreiecken besteht und somit auch eine andere Topologie besitzt.

4.4.1 Dezimierung nach Lindstrom und Turk

Garland und Heckbert stellen in ihrer Übersicht zur Vereinfachung polygonaler Netze [29] eine Vielzahl verschiedener Verfahren vor. Darunter wurde für diese Arbeit das Verfahren von Lindstrom und Turk [56] ausgewählt, da es durch die Beschränkung auf lokale Operationen im Dreiecksnetz geringe Komplexität und geringen Speicherbedarf aufweist. Weiterhin ist es im Programmcode verfügbar und gehört auch sechs Jahre nach seinem Entstehen zu den häufig eingesetzten Verfahren.

Wie in Abschnitt 4.2.3 dargestellt, lassen sich Dezimierungsverfahren klassifizieren, indem die folgenden Einzelprobleme genauer betrachtet werden:

- Lokale Dezimierungsmethode
- Fehlerüberwachung
- Qualitätsmaß für Dreiecke

Die vorliegende Methode nutzt als lokale **Dezimierungsmethode** die Kantenkollaps-Operation. Wie bereits erläutert, müssen dabei zwei Kernfragen beantwortet werden:

- Welche Kosten verursacht ein Kantenkollaps?
- Wo soll der neue Eckpunkt platziert werden?

Beide Fragen hängen eng mit der Art der Fehlerüberwachung zusammen. Das Verfahren von Lindstrom und Turk **verzichtet auf eine globale Fehlerüberwachung**, womit es sich aus der Masse der konkurrierenden Methoden heraushebt. Im Gegensatz zur bis zur genannten Veröffentlichung vorherrschenden Meinung in diesem Fachgebiet, führt der Verzicht auf eine globale Fehlerüberwachung gegen die Ausgangsdaten nicht zu verminderter Qualität des resultierenden Netzes. Im Zentrum der lokalen Fehlerüberwachung steht die durch eine lokale Operation induzierte Volumenänderung des Netzes.

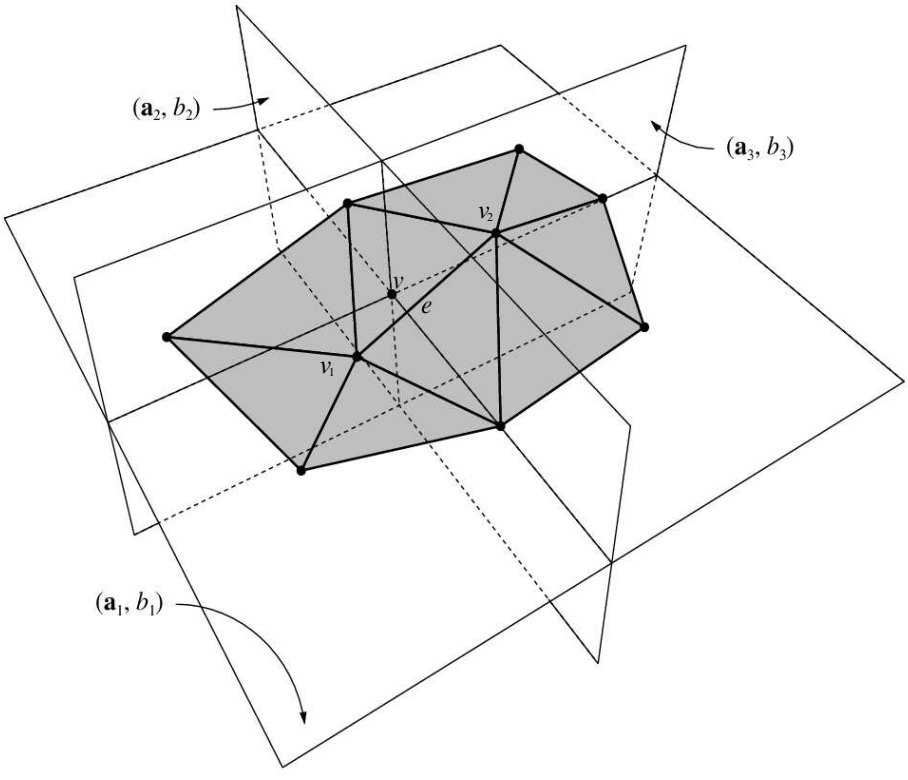


Abbildung 4.23: Die Position des neuen Eckpunktes **v** wird ausgedrückt als Schnitt dreier Ebenen; (a_1, b_1) gewährleistet den Volumenerhalt und (a_2, b_2) und (a_3, b_3) die Volumenoptimierung (Abbildung entnommen aus [56]).

- **Volumenerhalt**
  Betrachtet man bei einer Kantenkollaps-Operation die Bewegung der zwei Eckpunkte v_1 und v_2 an die neue Position v, so überstreichen sämtliche benachbarte Dreiecke ein Volumen in der Form eines Tetraeders. Fixiert man das bei einer Kantenkollaps-Operation veränderte Volumen zu Null, müssen sich positive und negative Volumina aufheben. Aus dieser Bedingung resultiert eine Ebene, auf der der neue Eckpunkt liegen muss (vgl. Abb. 4.23).

- **Randerhalt**
  Bei einem Kantenkollaps einer Randkante kann analog zum Volumenerhalt die durch den Kollaps veränderte Fläche eingeschränkt werden.

- **Volumenoptimierung**
  Beim Volumenerhalt wurde die Summe der vorzeichenbehafteten Volumenänderungen zu Null gesetzt. Es ist jedoch zusätzlich sinnvoll, auch die absolute Volumenänderung zu minimieren. Daraus resultieren zwei weitere Flächen, die die Lage des neuen Eckpunktes einschränken (vgl. Abb. 4.23).

- **Randoptimierung**
  Analog zur Volumenoptimierung wird die absolute Flächenänderung durch den Kollaps von Randkanten minimiert.

- **Optimierung der Dreiecksform**
  Die letzte oben genannte Fragestellung betrifft die Kontrolle der Dreiecksqualität. In bestimmten Fällen ergeben die genannten Bedingungen noch keine einfache Lösung. Dies ist zum Beispiel der Fall, wenn die limitierenden Ebenen nahezu parallel sind. In diesen Fällen wird zusätzlich die Dreiecksform als Kriterium herangezogen. Dabei wird ein Qualitätsmaß verwendet, das gleichseitigere Dreiecke bevorzugt.

Mit den genannten Einschränkungen wird nun für jeden potentiellen Kollaps eine gewichtete Kombination der beiden Hauptkriterien Volumen und Rand, nach denen oben minimiert wurde, zur Kostenberechnung genutzt werden. Lindstrom und Turk geben hier eine 50:50-Gewichtung der beiden Kriterien als optimal an. Damit lassen sich nun für jeden Kollaps die Kosten berechnen und in einer sortierten Liste speichern. Diese Liste wird beginnend mit den geringsten Kosten abgearbeitet, wobei nach jeder Operation die Kosten aller daran beteiligten Kanten in der Liste aktualisiert werden müssen. Als Abbruchkriterium dient eine Schranke für den akkumulierten Fehler oder eine maximale Anzahl von verbleibenden Dreiecken im Netz.

4.4.2 Anwendung auf Testdaten

Die schon für die Disparitätsberechnung genutzten Testaufnahmen eignen sich im Prinzip auch für Tests der Triangulierung und Dezimierung. Problematisch ist dabei aber, dass keine Kameraparameter verfügbar sind. Da die Bilder *Cones* und *Tsukuba* aber rektifiziert vorliegen, können beliebige Parameter angenommen werden. Dies führt bei der

Betrachtung des Modells aus unterschiedlicher Perspektive zu Verzerrungen, die sich jedoch durch Variation der angenommenen Kameraparameter nach Augenmaß minimieren lassen. Vorteilhaft ist, dass damit optimale Testdaten zur Verfügung stehen, mit denen die Dezimierung unabhängig von der Qualität der Stereorekonstruktion beurteilt werden kann.

Abbildungen 4.24 und 4.25 zeigen für unterschiedliche Dezimierungsstufen das Dreiecksnetz und texturierte Ansichten für die Testaufnahmen *Cones* und *Telekiste*. Dabei wurden Tiefensprünge wie in Abschnitt 4.3.7 beschrieben geöffnet. Um die Qualität besser beurteilen zu können, wurde auf Verschleierung von Modellfehlern (Löcher in Texturen, Rückseite von Dreiecken o.Ä.) verzichtet.

4.4.3 Diskussion

Probleme Auffällig an Abbildung 4.24 ist, dass bei fortschreitender Dezimierung einzelne Dreiecksnormalen vom Betrachter wegkippen und die Dreiecke damit von ihrer Rückseite sichtbar werden. Dies resultiert aus der Tatsache, dass die Dezimierungsmethode auf einem 3D-Netz arbeitet und damit kein 'vorne' und 'hinten' kennt. Der alternative Einsatz eines Verfahrens zur $2\frac{1}{2}$-D Dezimierung (z. B. [54]) fällt aus, wenn, wie hier der Fall, auch kombinierte Netze aus unterschiedlichen Kamerapositionen dezimiert werden sollen. Einzig ein zusätzliches, auf der Dreiecksnormalen basierendes Kriterium wäre denkbar, um das Kippen von Dreiecken zu verhindern.

Weiterhin arbeitet die Dezimierung im Nahbereich der Szene besser als in Bereichen, die weiter von der Kamera entfernt sind. Daraus ergibt sich eine insgesamt unnötig große Anzahl Dreiecke. Die Ursache liegt in der im Abschnitt 4.3.2 beschriebenen entfernungsabhängigen Genauigkeit. Kleine Variationen der Disparität in weit entfernten Bereichen ergeben nach der Rekonstruktion sehr viel größere Variationen in der Entfernung, als dies im Vordergrund der Aufnahme der Fall wäre. Da die Dezimierung im dreidimensionalen Netz durchgeführt wird, wird die Netzgeometrie unabhängig von ihrer absoluten Position in Kamerakoordinaten dezimiert.

Abhilfe Die naheliegende Idee, die Dezimierung im Disparitätenraum durchzuführen, ist aus zwei Gründen nicht praktikabel:

- Ein Netz kann auch aus mehreren Einzelaufnahmen aus unterschiedlicher Kameraposition zusammengesetzt sein. Dadurch existiert kein gemeinsamer Disparitätenraum.
- Schräge Ebenen über mehrere Entfernungsstufen im dreidimensionalen Raum sind im Disparitätenraum nicht eben. Dadurch können sie nicht durch wenige Dreiecke angenähert werden. Die Umkehrung des Problems, also die Näherung einer Ebene durch wenige Dreiecke im Disparitätenraum, ist nach der Rekonstruktion im dreidimensionalen Raum nicht eben und somit fehlerhaft dezimiert.

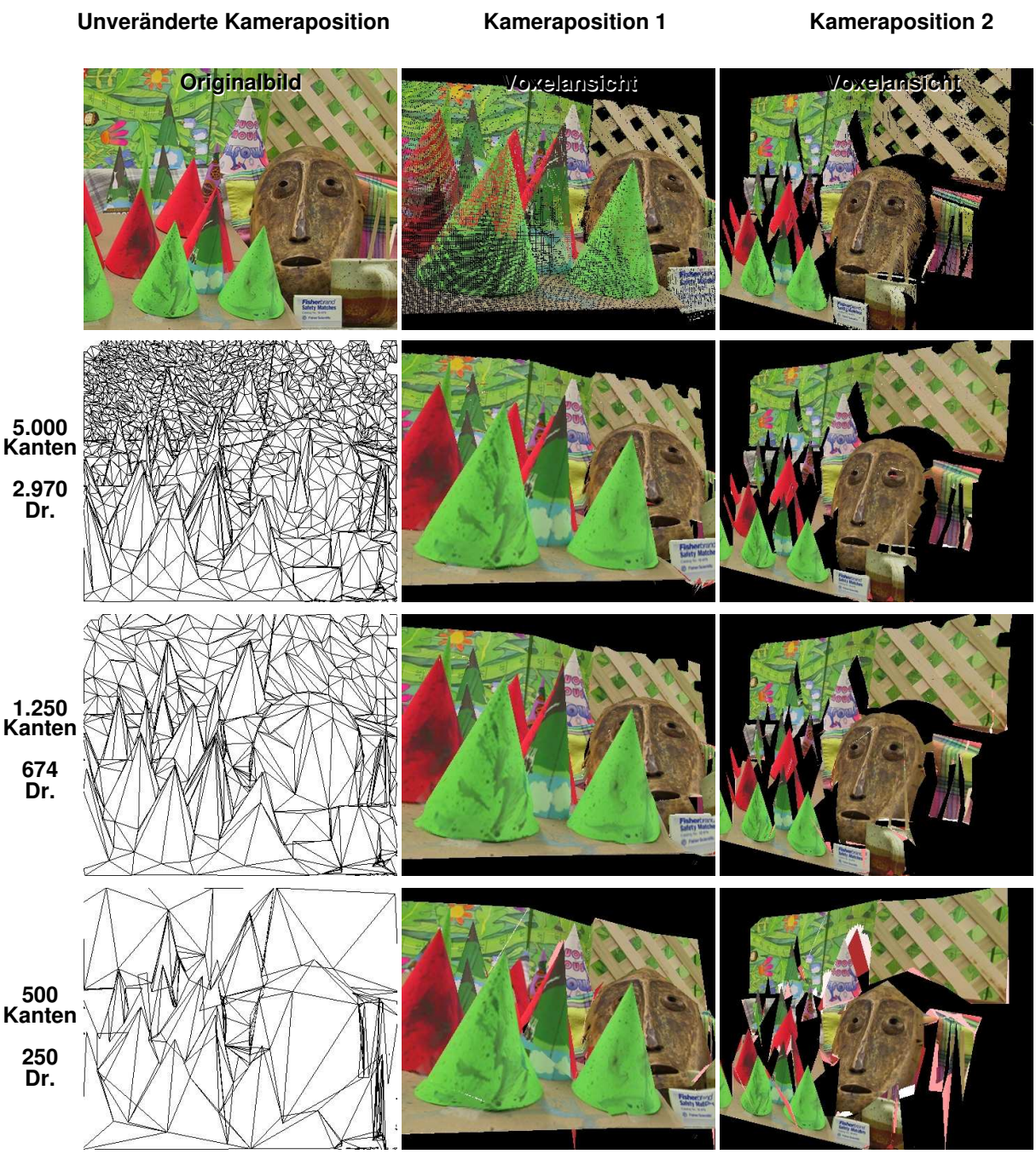


Abbildung 4.24: Dezimierung des Dreiecksnetzes *Cones*. Die initiale Triangulierung umfasst etwa 60.000 Kanten. Tiefensprünge wurden geöffnet. Von oben nach unten sind die Dezimierungsstufen von 5.000 bis 500 Kanten jeweils als Dreiecksnetz und in texturierter Form dargestellt.

Eine denkbare Abhilfe besteht in der Erweiterung der Kostenberechung bei der Dezimierung um einen disparitätsabhängigen Faktor. Das bedeutet aber, dass zu jedem Eckpunkt eine Disparität abgespeichert werden müsste. Die typische Ortsunsicherheit der Messwerte

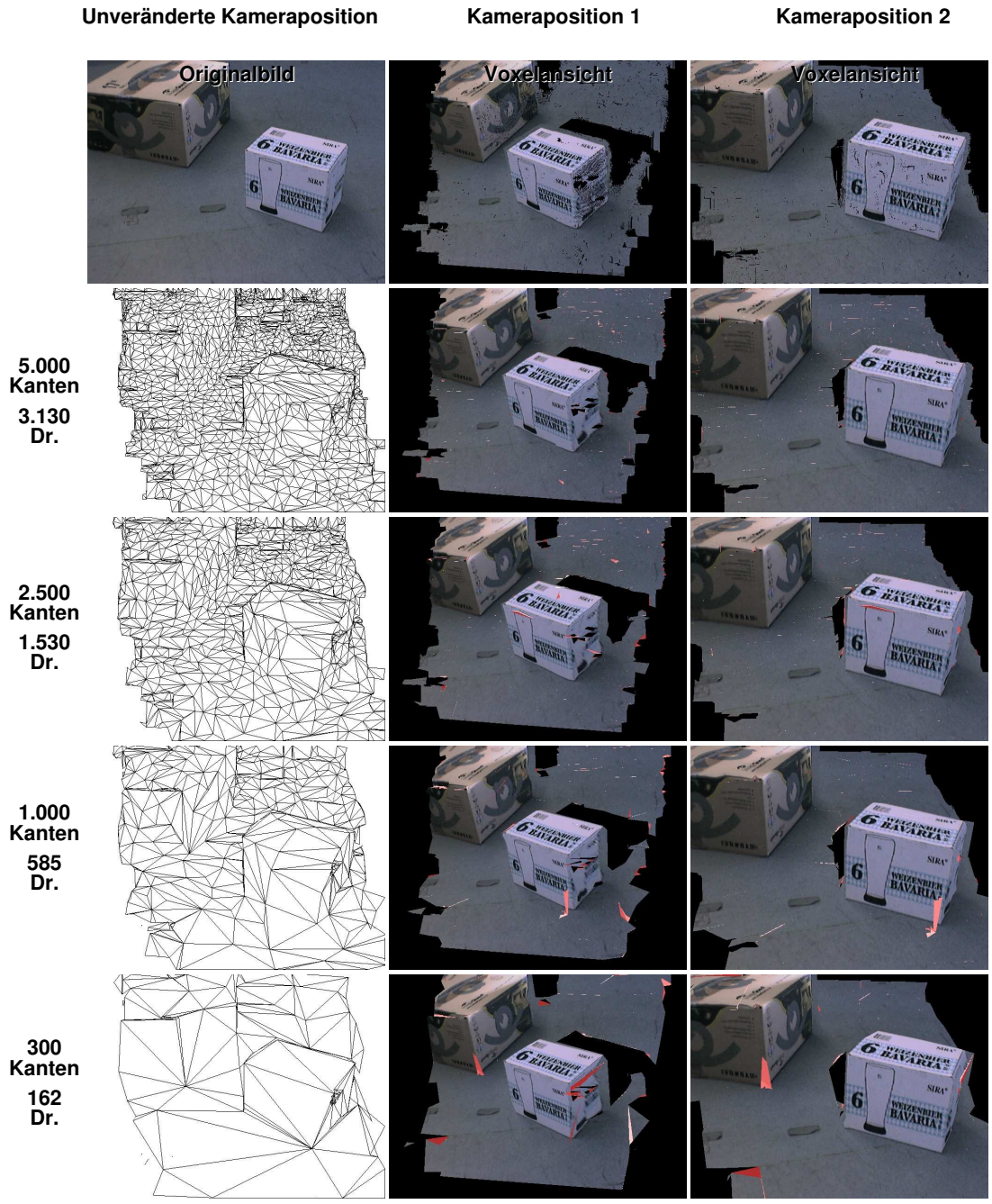


Abbildung 4.25: Dezimierung des Dreiecksnetzes *Telekiste*. Die initiale Triangulierung umfasst etwa 60.000 Kanten. Tiefensprünge wurden geöffnet. Von oben nach unten sind die Dezimierungsstufen von 5.000 bis 300 Kanten jeweils als Dreiecksnetz und in texturierter Form dargestellt.

in einem Stereosystem ist aber nicht nur von der Disparität abhängig, sondern vor allem von der Kameraposition zum Zeitpunkt der Messung. Obwohl eine konsequente Behandlung der Ortsunsicherheit der Messwerte sicher Vorteile besitzt, bringt sie so doch einen erheblichen Zuwachs an Komplexität und Speicherbedarf mit sich.

Verzichtet man auf diesen Aufwand, gibt es eine relativ einfache Lösung, zu einer ausgewogenen Triangulierung zu kommen: Die Erzeugung der Dreiecke findet im Disparitätenbild statt. Dadurch werden im Hintergrund der Szene eher wenige Dreiecke erzeugt, da hier nur geringe Unterschiede benachbarter Disparitäten vorherrschen[1)]. Die Dezimierung dagegen entfeinert aus den genannten Gründen eher im Vordergrund der Szene. Dadurch ergänzen sich beide Methoden bei geeigneter Wahl der initialen Triangulierungsdichte.

4.5 Netzfusion

Dieser Abschnitt beschäftigt sich mit der Frage, wie die Information aus mehreren Aufnahmen von unterschiedlichen Aufnahmezeitpunkten oder -positionen im Modell integriert und fusioniert werden kann. Dabei sind die Fragestellungen der Veränderung der Kameraposition bei statischer Szene und die Änderung der Szene bei statischer Kamera zu unterscheiden. Da beide Probleme in dieser Arbeit mit einer ähnlichen Methode behandelt werden, wird die Vorgehensweise zunächst anhand der räumlichen Fusion vorgestellt, um dann kurz die Besonderheiten zeitlicher Fusion zu besprechen. Wesentliche Teilprobleme bei der Kombination von Dreiecksnetzen sind die Erhaltung der Konnektivität des Netzes an den Bildrändern und das Löschen veralteter Teile des Netzes. Daher wird diesen Teilproblemen jeweils ein Abschnitt gewidmet. Passend zum Kontext der räumlichen Fusion endet das Kapitel noch mit einem kurzen Exkurs zur Registrierung der Kamera gegen das Modell der Szene.

4.5.1 Positionsänderung der Kamera

Ändert die Stereokamera ihre Position von einer Aufnahme zur nächsten, so müssen die aus der neuen Position sichtbaren Daten in das bereits aus vorhergehenden Aufnahmen existierenden Szenenmodell integriert werden. Dabei erfolgt die Fusion zwischen dem polygonalen Szenenmodell, das als Ergebnis sämtlicher vorangegangener Aufnahmen vorliegt, und einer 3D-Punktewolke, wie sie aus der Disparitätenkarte erzeugt werden kann.

Beispiel
Als Szenario für die folgenden Erklärungen soll die simulierte Szene, wie sie in Abbildung 4.26 dargestellt ist, dienen. Ein Stereokamerasystem beobachtet aus zwei unterschiedlichen Kamerapositionen eine einfache Szene bestehend aus einer rechteckigen Fläche im Vordergrund vor einem ebenen Hintergrund.

Ausgangspunkt
Zunächst wird aus dem Kamerabild I_{P1} eine Tiefenkarte D_{P1} berechnet. Durch adaptive Triangulierung wird daraus ein Dreiecksnetz, wie es Abbildung 4.27 darstellt. Wird nun die Kamera an Position $P2$ bewegt und dort das Bild I_{P2} aufgenommen, kann daraus die Disparitätenkarte D_{P2} berechnet werden.

[1] Dies gilt natürlich nur bei der dezimierenden Triangulierung

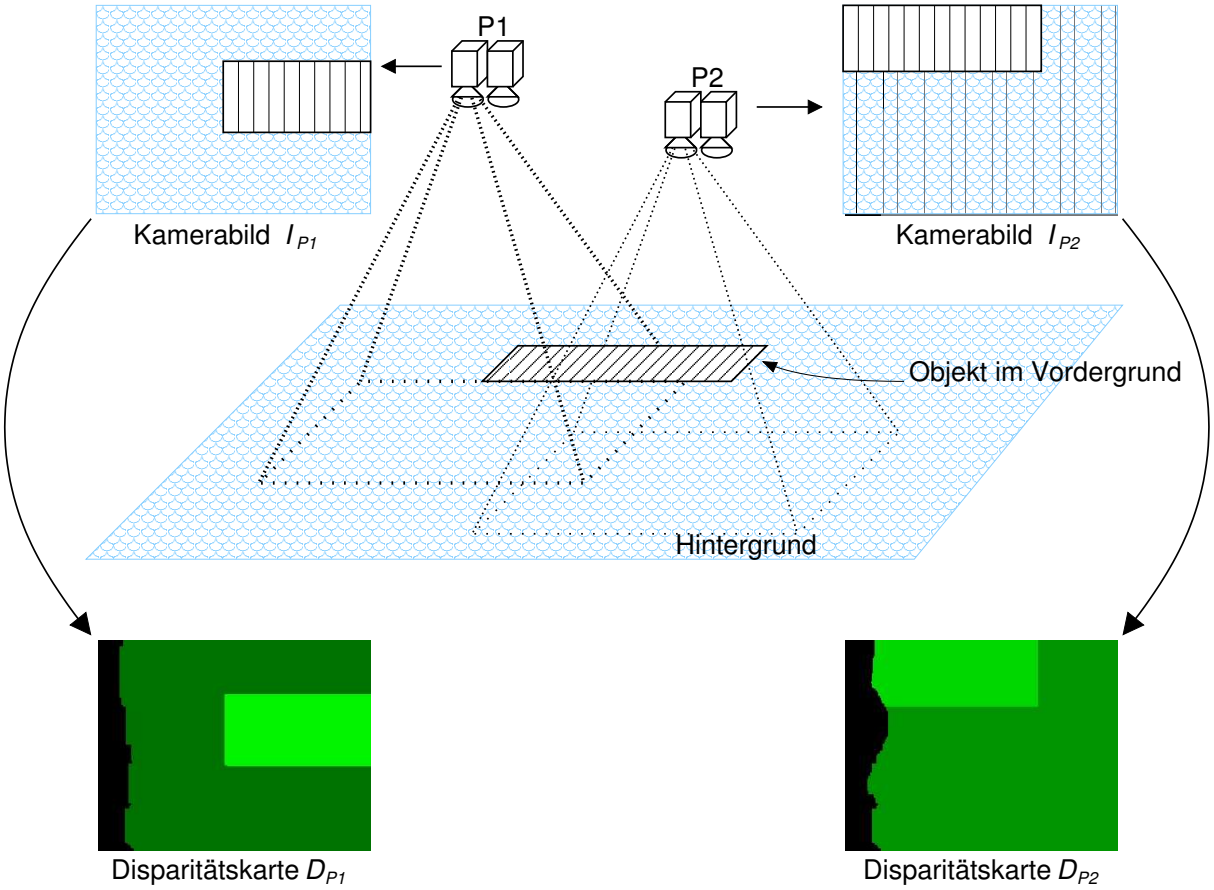


Abbildung 4.26: Simulierte Szene für die räumliche Fusion.

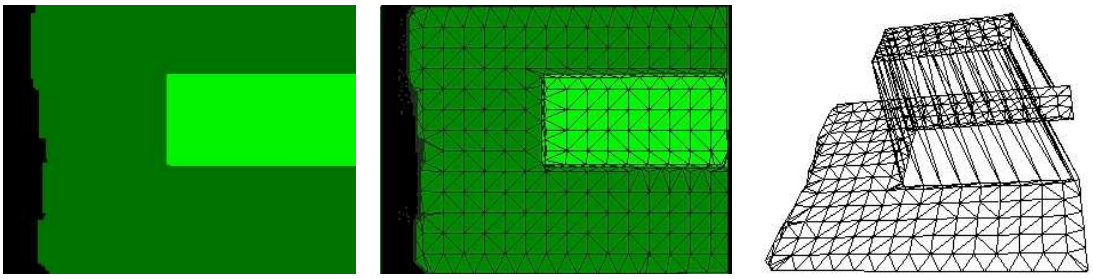


Abbildung 4.27: Disparitätenkarte D_{P0} und ihre Triangulierung (Mitte). Visualisierung des 3D-Netzes aus einer anderen Kameraperspektive (rechts).

Vergleich zwischen altem Modell und neuer Disparitätenkarte

Als gemeinsame Basis für den Vergleich der Modellinformation mit der neuen Tiefenkarte wird die neue Kameraposition $P2$ verwendet. Das 'alte' Szenenmodell wird aus Kameraposition P2 mit den Parametern des Kamerasystems abgebildet (gerendert). Die dabei entstehende virtuelle Disparitätenkarte D'_{P2} ist in Abbildung 4.28a dargestellt. Nun können die Disparitätenkarten D_{P2} und D'_{P2} durch Subtraktion verglichen werden.

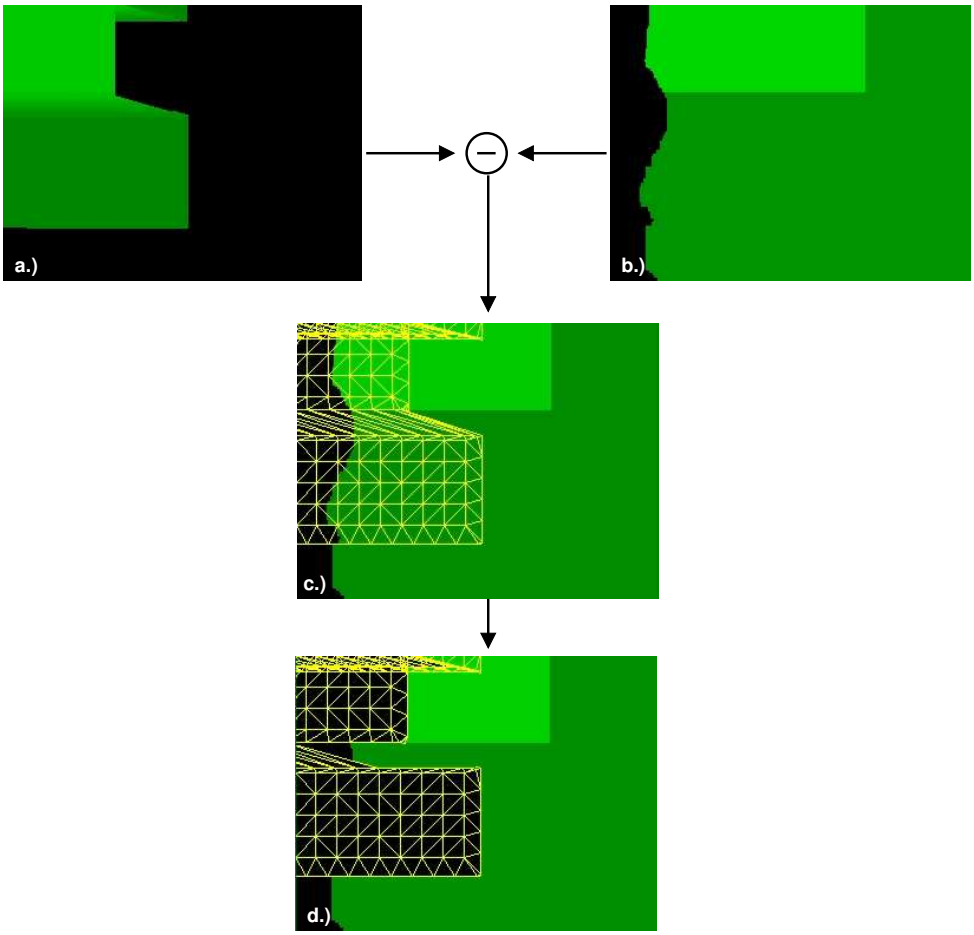


Abbildung 4.28: Vergleich der virtuellen Tiefenkarte (D'_{P2}) des Modells von Kameraposition P2 aus betrachtet (a) und der realen Tiefenkarte D_{P2} (b). Überlagerung des Modells mit der neuen Disparitätenkarte D_{P2} (c) und das Ergebnis des Vergleichs der Tiefenkarten: Bereits modellierte Bereiche von D_{P2} wurden entfernt, Falsche Dreiecke im Modell wurden entfernt (d).

Ergebnis des Vergleichs

Für jeden Bildpunkt in D_{P2} tritt einer von folgenden vier Fällen ein:

$D'_{P2}(x,y) = D_{P2}(x,y)$	Der Bildpunkt ist bereits modelliert	Er wird verworfen
$D'_{P2}(x,y) < D_{P2}(x,y)$	Der Bildpunkt liegt vor dem Dreiecksnetz	Er wird als neues Polygon zum Netz hinzugefügt
$D'_{P2}(x,y) > D_{P2}(x,y)$	Der Bildpunkt liegt hinter dem Dreiecksnetz	Das Netz muss an dieser Stelle geöffnet werden
$D'_{P2}(x,y) = \emptyset$	An dieser Stelle gibt es noch kein Dreiecksnetz	Es muss ein neues Polygon erzeugt werden

Im vorliegenden Beispiel müssen somit einige falsche Polygone aus dem Dreiecksnetz entfernt werden und die Bereiche in D_{P2}, die zum Modell passen, aus D_{P2} entfernt werden.

So entsteht die reduzierte Disparitätenkarte D^*_{P2} (vgl. Abb. 4.28d). Die Entscheidung über die Zugehörigkeit zu einer der vier Klassen erfolgt jedoch nicht disparitätengenau sondern durch Definition eines Fusionsbereichs wie in Abbildung 4.29 dargestellt. Er definiert eine maximal erlaubte Abweichung eines Pixels aus D von einem Pixel aus D'.

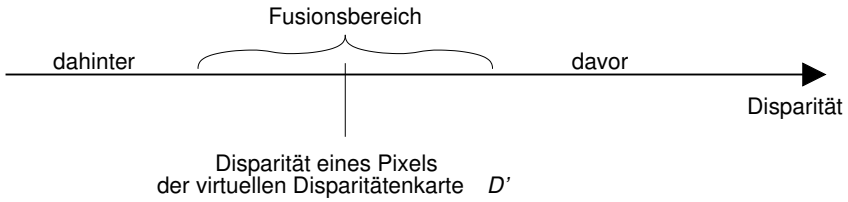


Abbildung 4.29: Der Fusionsbereich.

Abbildung 4.28d stellt das Resultat dar. Die verbleibenden Reste der reduzierten Tiefenkarte D^*_{P2} können nun ebenfalls trianguliert, und zum Modell hinzugefügt werden. Das resultierende Netz ist zunächst nicht verbunden, da die Triangulierung nur für die neue Tiefenkarte durchgeführt wird.

Vernähen von Dreiecksnetzen

Um die Ränder benachbarter Netze zu vebinden, müssen die Netzkanten sowohl in ihrer zweidimensionalen Projektion als auch im dreidimensionalen Raum einen bestimmten Maximalabstand besitzen. Ist dies der Fall, werden neue Dreiecke zwischen den Eckpunkten der beiden Netzränder eingefügt. Die erforderlichen Berechnungen lassen sich im dreidimensionalen Raum oder in der zweidimensionalen Projektion aus der neuen Kameraposition durchführen. Da im vorliegenden System die Fusion durch Projektion im zweidimensionalen Raum erfolgt, wird in dieser Arbeit das Vernähen der Netze ebenfalls im zweidimensionalen Raum durchgeführt. Für die weiteren Erklärungen dient wieder das im vorangegangenen Abschnitt eingeführte Beispiel. Folgende Einzelschritte werden durchgeführt:

1. Berechnung des Randes B des 'alten' Dreiecksnetzes M
2. Projektion der Eckpunkte von B in das Kamerasystem $P2$
3. Auswahl der Eckpunkte, deren Projektion innerhalb des Kamerabildes I_{P2} liegt und die *nah* (2D-Distanz typ. 5 Pixel) zum Rand der reduzierten Disparitätenkarte D^*_{P2} liegen.
4. Weitere Einschränkung der Eckpunkte nach ihrer Disparität: Es werden nur Eckpunkte verwendet, deren Disparität innerhalb des Fusionsbereichs um ihre nächsten Nachbarn in D_{P2} liegt.
5. Sämtliche Kanten des 'alten' Netzes M, deren beide Eckpunkte den aufgeführten Kriterien entsprechen, werden nach der Triangulierung von D_{P_2} in 2D als Zwangsbedingung hinzugefügt.

6. Durch Delaunay-Triangulierung werden die nötigen Kanten und Dreiecke erzeugt

Abbildung 4.30 zeigt die erkannten Eckpunkte und das Resultat.

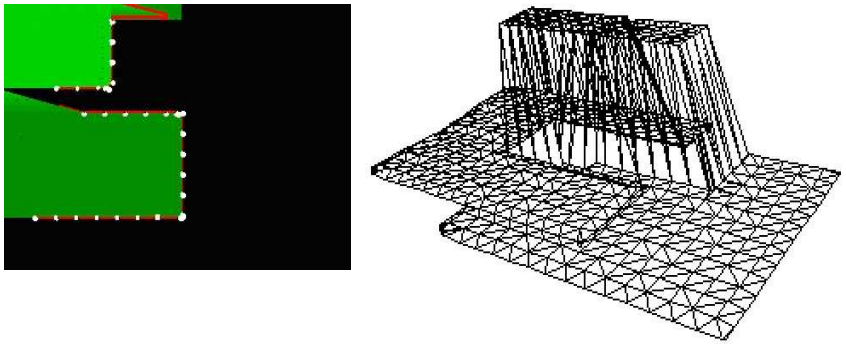


Abbildung 4.30: Eckpunkte, die bei der Triangulierung von I_{P2} mit berücksichtigt werden müssen, und das resultierende Netz aus unterschiedlicher Perspektive betrachtet.

Löschen von Modellinformation

Beim Vergleich der Modellinformation mit der neuen Disparitätenkarte müssen Dreiecke aus dem 'alten' Modell entfernt werden, wenn neue Disparitäteninformation Objekte hinter dem Netz impliziert. Da das alte Netz aber normalerweise bereits dezimiert vorliegt, führt das zu unnötigem Informationsverlust, da große Dreiecke gelöscht werden, auch wenn sie nur zum Teil fehlerhaft sind.

Hier wurde eine Methode entwickelt, das Modell zunächst so zu verfeinern, dass nur der fehlerhafte Teil großer Dreiecke gelöscht wird. Nach der Differenzbildung zwischen der aus dem Modell erzeugten Disparitätenkarte (D'_{P2} im Beispiel oben) und der neuen Disparitätenkarte D_{P2} liegen eventuell zu löschende Bereiche als 2D-Region vor. Diese Region lässt sich in ein rohrförmiges 3D-Objekt umwandeln, wenn für ihre Umrandung sehr kleine und sehr große Werte für die Disparität angenommen werden. Diese 3D-Form beschreibt den Sichtkegel dieser Region aus der neuen Kameraperspektive. Durch Bildung der Booleschen Differenz zwischen dem 3D-Modell und dem erzeugten Objekt kann durch Standardmethoden der Geometrie der Bereich des Netzes entfernt werden, der die 2D-Region verdecken würde.

Beispiel

Ein komplexeres Beispiel für die räumliche Fusion bilden die beiden in Abbildung 4.31 dargestellten Aufnahmen aus Testserie 3. Nach der Modellbildung aus der ersten Aufnahme, wie sie in Abbildung 4.32 dargestellt ist, liegt als Ergebnis das polygonale Modell der Szenengeometrie aus der ersten Aufnahme vor. Nach einer Bewegung des Kamerakopfes wird der zweite Stereo-Bildersatz aufgenommen und in eine Disparitätenkarte umgerechnet.

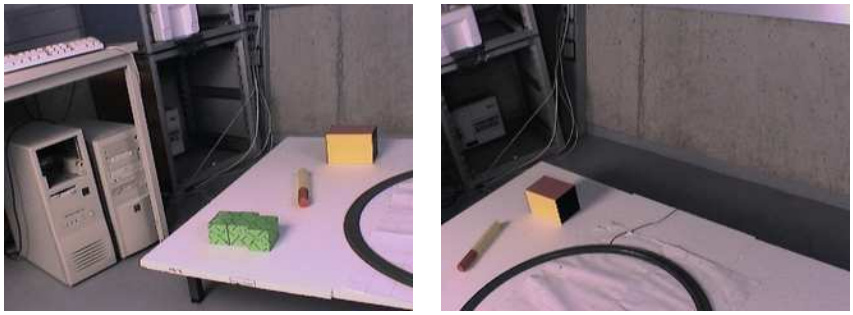


Abbildung 4.31: Die linken Bilder zweier Stereoaufnahmen mit großen translatorischen Versatz. Insbesondere an der hinteren Tischkante ergibt sich starke Aufdeckung verdeckter Bereiche.

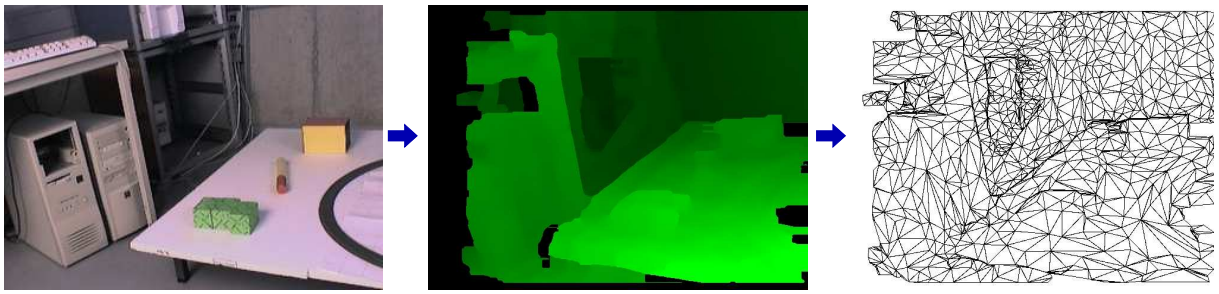


Abbildung 4.32: Berechnung der Disparitätenkarte und anschließende Modellbildung aus Aufnahme 1.

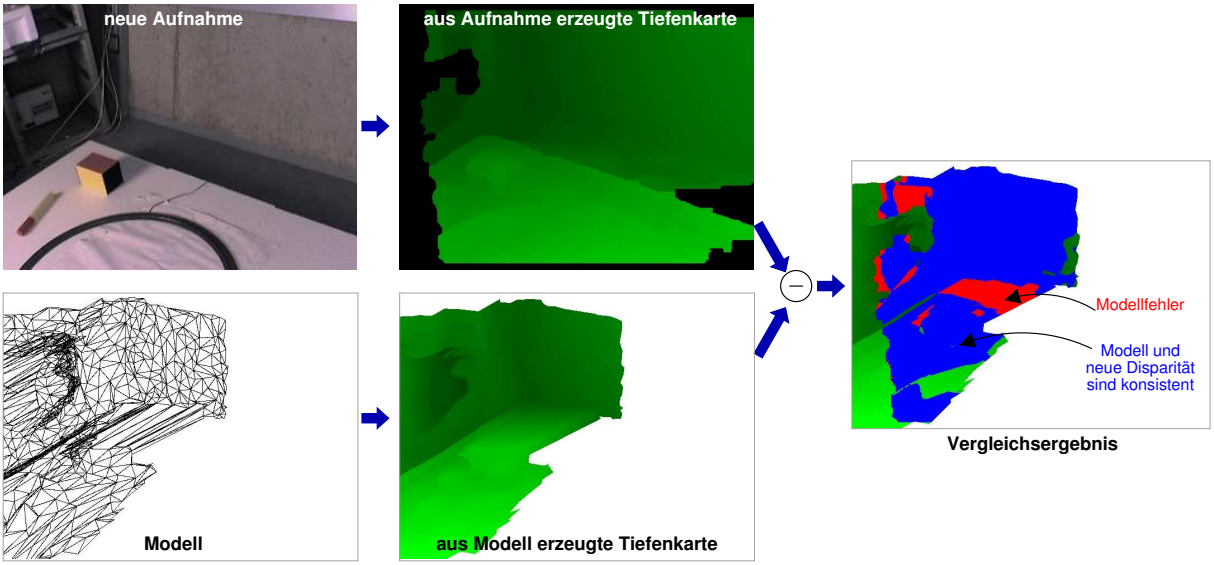


Abbildung 4.33: Vergleich der neuen realen Disparitätenkarte (oben) mit der aus dem Modell erzeugten virtuellen Disparitätenkarte (unten). Als Resultat ergeben sich Bereiche der Übereinstimmung und Bereiche, in denen Fehler im Netz vorliegen (rechts).

Abbildung 4.33 stellt den Vergleich der aus der zweiten Aufnahme gewonnenen Disparitätenkarte und der aus dem bisherigen Modell gewonnenen virtuellen Disparitätenkarte dar. Durch eine Schwellwertentscheidung im Differenzbild können passende und falsche Regionen detektiert werden.

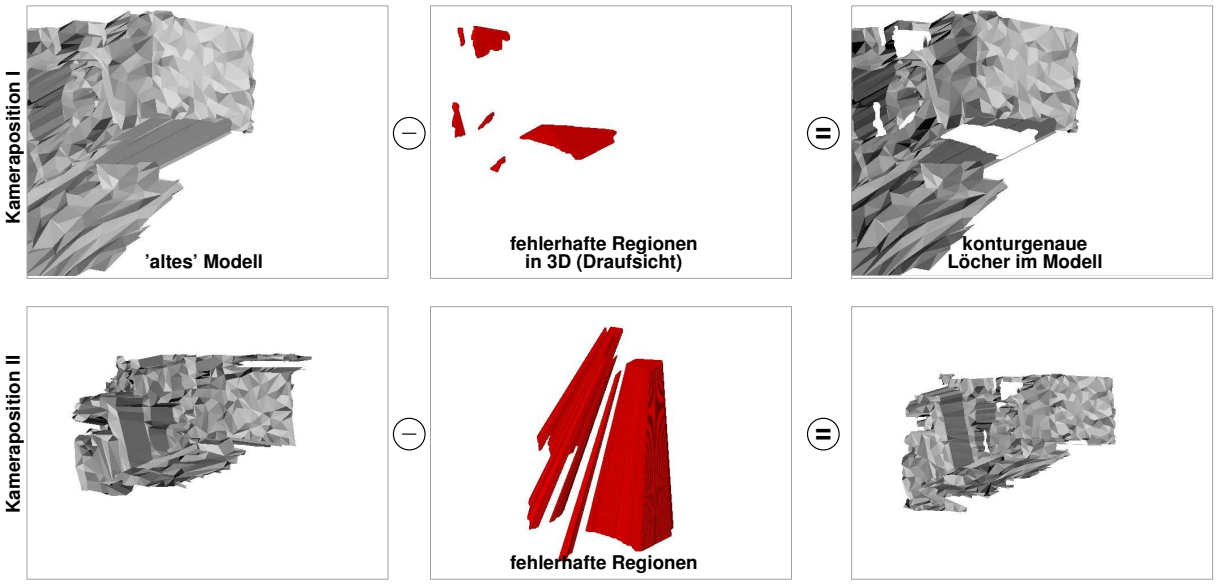


Abbildung 4.34: Bildung der Booleschen Differenz zwischen Modell und den fehlerhaften Regionen. Dazu werden die fehlerhaften Regionen geglättet und entlang ihres Sichtkegels in der neuen Aufnahme extrudiert. Die Abbildung stellt den Prozess aus zwei unterschiedlichen Perspektiven dar (obere und untere Zeile) um den dreidimensionalen Charakter der Operation zu veranschaulichen.

In einem Folgeschritt werden nun die fehlerhaften Bereiche aus dem Modell entfernt, wie Abbildung 4.34 zeigt. Dazu werden die fehlerhaften zweidimensionalen Regionen entlang des Sichtkegels der Kamera extrudiert und in dreidimensionale geschlossene Objekte umgewandelt. Mit Standardmethoden der Geometrie kann nun die Boolesche Differenz der Netze gebildet werden. Als Resultat werden die Regionen konturgenau freigeschnitten und die erforderlichen Kanten und Dreiecke zum Netz hinzugefügt. Die weiteren Schritte gestalten sich wieder wie oben beschrieben: Nach einer Randextraktion des nun reduzierten 'alten' Netzes kann die neue, ebenfalls reduzierte Disparitätenkarte trianguliert und dabei mit dem Netz vernäht werden. Das Ergebnis zeigt Abbildung 4.35 in verschiedenen Varianten und Dezimierungsstufen.

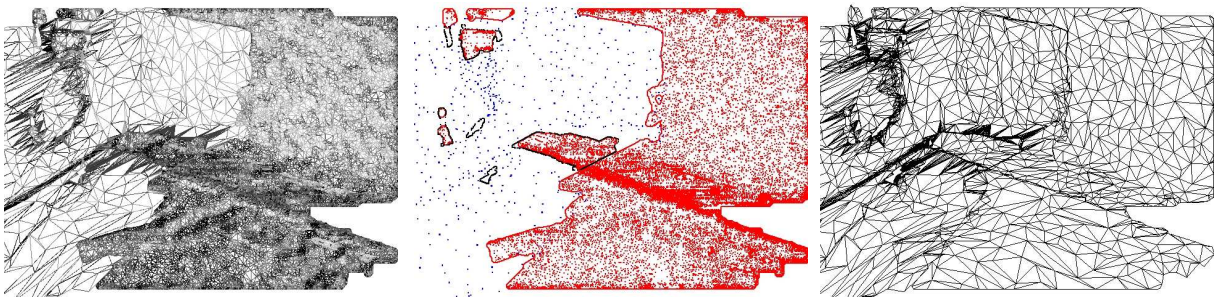


Abbildung 4.35: Ergebnis der Netzfusion: Das alte Netz mit dem noch nicht dezimierten neuen Netz direkt nach der Triangulierung (links). Die Netze sind vernäht, und fehlerhafte Bereiche wurden durch das neue Netz ersetzt. Die mittlere Ansicht zeigt die Eckpunkte des neuen Netzes koloriert nach ihrer Herkunft. Die rechte Ansicht zeigt die dezimierte Version des Netzes.

4.5.2 Veränderungen der Szene und Modellfehler

Dynamische Szenen führen zu einer Veränderung des Szeneninhalts zwischen zwei Aufnahmen des Kamerasystems. Dabei wird, um die Komplexität des Problems zu reduzieren, im Kontext dieser Arbeit von dem relativ einfachen gedanklichen Szenario, wie es im einführenden Kapitel (1.2) beschrieben ist, ausgegangen. Zunächst ist der einzige dynamische Teilnehmer der Szene der Manipulator. Da dessen Position aber im Detail als bekannt vorausgesetzt werden kann, sollte er aus sämtlichen Berechnungen des Szenenhintergrundes ausgenommen werden. Objekte, mit denen der Operator interagiert, werden kantenbasiert geometrisch modelliert. Somit bleibt ein normalerweise statischer Hintergrund. Trotzdem sollen Veränderungen des Hintergrundes bei der Modellaktualisierung berücksichtigt werden können, um die prinzipielle Möglichkeit von Veränderung der Szene nicht gänzlich auszuschließen und das System nicht unnötigerweise einzuschränken.

Auf der Basis dieser Einschränkungen kann die im Abschnitt 4.5.1 vorgestellte Methode zur Fusion von Daten bei bewegter Kamera auch für dynamische Szenen eingesetzt werden. Beim für die Fusion erforderlichen Vergleich der neuen Tiefenkarte mit der bereits vorliegenden Modellinformation fällt, wie oben beschrieben, folgende Information über Veränderungen in der Szene an:

- Liegen 3D-Punkte *in der Nähe* von Polygonen, werden sie dem Polygon zugeordnet.
- Liegen 3D-Punkte *hinter* einem Polygon, muss das Polygon fehlerhaft sein – es muss entfernt werden.
- Liegen 3D-Punkte *vor* einem Polygon, gehören sie zu einem unbekannten Objekt und müssen somit in das Modell aufgenommen werden.

Diese Vorgehensweise eignet sich auch, um das Verschwinden oder Auftauchen von Objekten in der Szene zu behandeln. Wenig geeignet ist die Methode für wirklich bewegte Objekte, die ihre Position zwischen Einzelaufnahmen verändern. Sie würden sozusagen scheibchenweise zum Modell hinzugefügt und aus dem Modell entfernt werden. Das bedeutet, dass an der Vorderkante eines bewegten Objektes neue Polygone entstehen und an seiner Hinterkante Polygone entfernt werden. Für die korrekte Behandlung dieser Objekte müssten die anfangs (Abschnitt 1.2) vorgestellten Objektklassen *Vordergrund* und *Hintergrund* mit ihren zugeordneten Aktualisierungsverfahren um eine weitere Klasse erweitert werden, die sich speziell der Modellierung unbekannter bewegter Objekte widmet.

Trotzdem wird auch bei bewegten Objekten eine korrekte Bildsynthese erreicht, auch wenn die Vorgehensweise insbesondere aus der Sicht der Texturierung nicht optimal ist.

4.5.3 Registrierung der Kamera

Diese Arbeit entstand im Umfeld des Teilprojektes C2 „Übertragungszeitkompensation durch Szenenprädiktion“ im Sonderforschungsbereich 453. Im Rahmen dieses Projektes war die Kameraposition durch eine Schwenk-Neige-Einheit festgelegt. Damit steht (nach

Kalibrierung) die genaue Position des Kamerasystems im Raum fest, es sind jedoch nur rotatorische Bewegungen möglich. Da sich damit keine Modellinformation im Bereich von Verdeckungen akquirieren lässt, wird das resultierende Modell unvollständig und 'löchrig' sein.

Die eingesetzten Verfahren in dieser Arbeit sind jedoch sehr wohl tauglich, auch translatorische Kamerabewegung zu verarbeiten. Das bedingt eine andere Quelle für die Kameraposition als die Schwenk-Neige-Einheit. Dies kann zum Beispiel ein mobiler teleoperierter Roboter sein oder – wie in der vorliegenden Arbeit eingesetzt – ein einfaches Stativ zur freien Positionierung des Kamerasystems. Das bedingt aber ein Registrierungsverfahren zur Messung der Position des Kamerasystems. Es wurden zwei Verfahren evaluiert, die eine Positionsmessung auf der Basis von Bild-bzw. 3D-Daten ermöglichen. Der *Iterative-Closest-Point*-Algorithmus – das hierfür übliche Standardverfahren – wurde bereits in Abschnitt 3.1.4 vorgestellt und die Registrierung auf der Basis von 2D-3D-Korrespondenzen, wie sie im folgenden Abschnitt vorgestellt werden.

Registrierung auf der Basis von 2D-3D-Korrespondenzen

Bei der Vorstellung des ICP-Algorithmus wurden verschiedene Möglichkeiten, dieses komplexe Verfahren zu beschleunigen vorgestellt. Eine dieser Möglichkeiten, die Beschleunigung durch 2D-Korrespondenzen, führt auf eine einfachere Lösung des Registrierungsproblems: Werden merkmalsbasiert 2D-2D-Korrespondenzen zwischen Referenzbild und neuem Bild (jeweils linke Kamera) bestimmt, können diese in 3D-2D-Korrespondenzen umgewandelt werden, da durch die fortschreibende Modellierung für das Referenzbild bereits rekonstruierte 3D-Daten vorliegen. Dies reduziert das Problem auf den selben Fall, der bei der intrinsischen Kamerakalibrierung auftritt – bei bekannten 2D-3D-Korrespondenzen wird die Lage der Kamera durch Lösung eines überbestimmten Gleichungssystems berechnet. Sind ausreichend Merkmale bei ausreichend komplexer Geometrie (nicht ausschließlich koplanare oder kollineare Punkte) ohne Fehlkorrespondenzen vorhanden, lässt sich so die Relativposition berechnen. Ausreißer lassen sich durch mehrere Durchläufe der Korrespondenzsuche mit Berechnung der Fundamentalmatrix ausfiltern.

Zusammenfassung zur Kameraregistrierung

Obwohl der ICP-Algorithmus gut geeignet ist, das Registrierungsproblem zu lösen, spricht seine hohe Komplexität gegen seinen Einsatz. Nur unter Nutzung der verschiedenen Verbesserungen kann er praktisch eingesetzt werden. Die einfache Registrierung auf der Basis von 2D-3D-Korrespondenzen besitzt eine deutlich geringere Genauigkeit, da nur wenige ausgewählte Merkmale verwendet werden. Nur die Kombination der vorgestellten Verfahren erlaubt eine schnelle relative Schätzung der Kameraposition zwischen Folgebildern. Durch die Akkumulation des Fehlers ist das Verfahren jedoch auch nicht geeignet, über längere Zeit die absolute Kameraposition bereitzustellen. Im Rahmen dieses Projektes wurde für freie Kamerabewegung die Kombination aus 2D-3D-Korrespondenzen und ICP

implementiert und evaluiert. Im folgenden Kapitel wurde für die Testserie eins und zwei die Position des Schwenk-Neige-Kopfes verwendet und die Testserie drei unter Verwendung von 2D-3D-Korrespondenzen registriert.

4.6 Eine neue Datenstruktur zur Modellierung

Dieser Abschnitt beschäftigt sich mit Fragen zur Datenhaltung und Stabilisierung. Nach einer Vorstellung des Problems wird ein neuartiger Ansatz vorgestellt und diskutiert.

4.6.1 Probleme bisheriger Lösungen

Wie im Abschnitt 4.3.1 kurz angesprochen, bedingt die Anwendung Telepräsenz eine inkrementelle und fortschreitende Modellbildung, -verifikation und -veränderung. Dadurch muss im Modell eine zeitliche Stabilisierung der Daten erfolgen. Gleichzeitig besteht die Forderung nach stabilen Polygonen, um Texturen weiterverwenden zu können.

Das Modell besteht, soweit es bisher vorgestellt wurde, aus zwei Verarbeitungsebenen – 3D-Punkte mit Nachbarschaftsbeziehungen (rekonstruierte Disparitätenkarte) und Polygone. Damit sind drei Varianten zur Stabilisierung denkbar:

- **3D-Punktewolken** – wie sie aus den Disparitätenkarten mehrerer Kamerabilder entstehen – eignen sich gut für eine Stabilisierung, da sie ein niedriges Abstraktionsniveau aufweisen. Verschiedenste Konzepte wie zum Beispiel die Verschmelzung benachbarter Punkte oder die Assoziation einzelner Punkte mit einem Vertrauensmaß sind naheliegend und werden häufig verwendet. Wird aber bei jeder Aktualisierung der Punktewolke neu trianguliert, ändert sich die Topologie des Netzes mit jeder neuen Aufnahme, auch wenn keine tatsächliche Veränderung der Szene vorliegt, da die Topologie des Netzes vom Rauschen der Disparitätenkarte abhängig ist. Hinzu kommt, dass Punktewolken als Datenbasis keine Nachbarschaftsinformation kodieren, wie sie in Disparitätenkarten noch vorliegt. Dadurch ist die Triangulierung einer Szene aufwändiger.
- **3D-Punktewolken mit kodierten Nachbarschaftsbeziehungen** sind nur bei rein rotatorischer Kamerabewegung denkbar, da hier die Topologie der 3D-Punkte, also ihre Nachbarschaftsbeziehungen in verschiedenen Aufnahmen konsistent sind. Wird die Kamera auch translatorisch bewegt, müssen Nachbarschaftsbeziehungen der 3D-Punkte aus unterschiedlichen Aufnahmen miteinander verwoben werden. Weder für die erforderliche Datenstruktur noch für ihre Triangulierung sind Verfahren verfügbar.
- **Dezimierte Polygone** als höchstes Verarbeitungsniveau stellen bereits eine verlustbehaftete starke Komprimierung der Modellinformation dar. Keinerlei Detailinformation über die zugrundeliegenden 3D-Punkte ist mehr enthalten. Polygonale

Netze aus unterschiedlichen Kamerapositionen oder Zeitpunkten können zwar fusioniert werden, doch erzeugt der Verzicht auf die Basisdaten bei dem Vergleich oder der Verschmelzung von Netzen Qualitätseinbußen.

Keine dieser drei Varianten kann das Problem wirklich lösen. So können 3D-Punktewolken zwar leicht aktualisiert und fusioniert werden, besitzen aber keine stabile Triangulierung. Die Fusion auf polygonaler Ebene, wie sie im vorangegangenen Kapitel beschrieben wurde, bringt komplexe Netzoperationen mit sich und führt durch die starke Datenreduktion zu Qualitätseinbußen.

4.6.2 Hybride Datenstruktur aus Polygonen und 3D-Punkten

Prinzip

Wird mit jedem Dreieck im Netz eine eigene Tiefenkarte assoziiert, die jeden 3D-Punkt enthält, der zu diesem Dreieck beiträgt, so ist sichergestellt, dass die tatsächliche Qualität eines Dreiecks jederzeit berechnet werden kann. Neue 3D-Punkte können auf einfache Weise mit dem Netz verschmolzen werden, da die Entscheidung über eine eventuell nötige Änderung der Triangulierung korrekt gefällt werden kann. Nebenstehende Abbildung verdeutlicht das Prinzip (vgl. 4.36). Die Punkte einer Aufnahme (rot) werden durch ein Polygon modelliert. Durch eine weitere Aufnahme (blau) kommen Punkte hinzu und führen stellenweise zu einer Änderung und stellenweise zu einer Stabilisierung der Triangulierung. Die Assoziation von Punkten zu Dreiecken wurde von Soucy und Laurendeau [82] für die Netzdezimierung verwendet, um zu jederzeit eine Kontrolle des globalen Fehlers garantieren zu können (vgl. Abschnitt 4.2.3). Sie nennen die assoziierten 3D-Punkte die *Vorfahren* (*Ancestors*) des Polygons, die in ihrem Fall jedoch ehemalige Eckpunkte des Netzes darstellen.

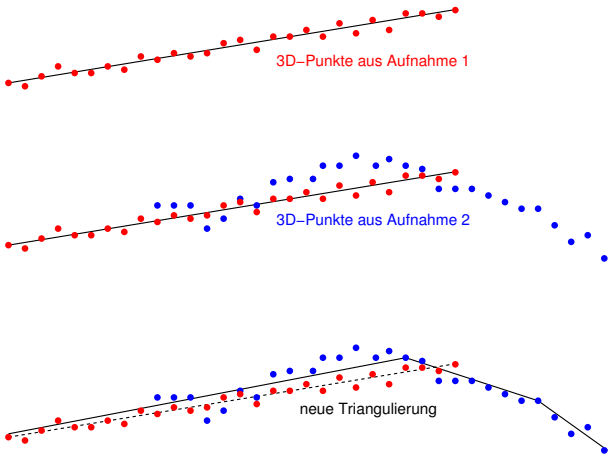


Abbildung 4.36: Jedes Polygon kennt 'seine' 3D-Punkte. Punkte aus mehreren Aufnahmen werden akkumuliert und führen gegebenenfalls zu einer Neutriangulation

Anstatt die 3D-Punkte der Dreiecke einzeln zu speichern, ist es im vorliegenden Kontext günstiger, an dieser Stelle bereits eine Fusion von sehr ähnlichen Punkten vorzusehen. Eine geeignete Datenstruktur ist der Oct-Tree. Bei dieser Datenstruktur wird ein würfelförmiges Volumen rekursiv in je 8 Teilwürfel unterteilt. Um Speicher zu sparen, wird ein Würfel nur dann weiter unterteilt, wenn es Volumenelemente (Voxel) im nächstkleineren Teilvolumen gibt. Abbildung 4.37 verdeutlicht das Prinzip. Jedes Polygon

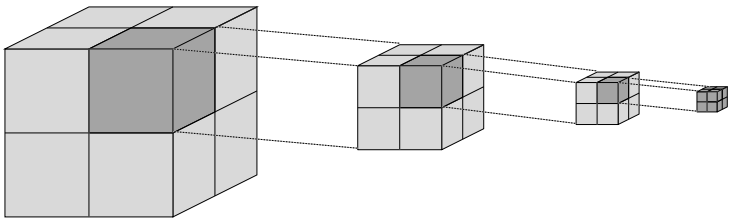


Abbildung 4.37: Prinzip des Oct-trees: Die rekursive Aufteilung eines quaderförmigen Volumens.

besitzt einen assoziierten Oct-Tree seiner konstituierenden Voxel. Die Fehlerberechnung basiert auf der Summation der Abstände der Voxel (in Normalenrichtung) von ihrem Polygon. Bei der Netzdezimierung wird ein *„one move lookahead "* [36] angewendet, das heißt jeder potentielle Kantenkollaps wird ausprobiert, der resultierende Fehler berechnet und dann die Operation mit dem geringsten Fehler ausgeführt.

Vorteile Mit dieser Struktur kann eine Obergrenze für die maximale Modellgröße effektiv beschränkt werden: Die Größe eines Polygons definiert die Maximalgröße seines Oct-Trees, und der Oct-Tree kann nur eine begrenzte Menge Voxel aufnehmen. Wird die Information einer neuen Aufnahme zum Modell hinzugefügt, wird jeder 3D-Punkt, für den bereits ein passendes Polygon existiert, zum Oct-Tree dieses Polygons hinzugefügt. Ist im Oct-Tree die passende Voxelposition bereits besetzt, ändert sich nichts. Ist die Voxelposition noch leer, wird gegebenenfalls der Oct-Tree Ast erzeugt und die Position besetzt. So kann jederzeit überprüft werden, ob ein Polygon noch zu seinen Voxeln passt oder es neu trianguliert werden muss.

Mit dieser Struktur gelingt eine implizite Fusion der Modelldaten aus mehreren Ansichten bei Bewahrung des Bezugs zur Triangulierung. 3D-Punkte an nahegelegenen Positionen werden in dasselbe Voxel fusioniert.

Nachteile Netzoperationen wie Dezimierung oder Verfeinerung werden äußerst komplex. Im Falle der Verfeinerung muss der Voxelbestand eines Dreiecks auf mehrere neue Dreiecke aufgeteilt werden. Im Fall der Dezimierung werden die Voxelbestände mehrerer Dreiecke zu einem neuen hinzugefügt. Dabei wird die Distanz in Normalenrichtung zur Entscheidung herangezogen.

Der Vorteil der präzisen Bestimmung des globalen Fehlers des Netzes ist gleichzeitig ein Nachteil: Für jede potentielle Netzmanipulation muss der lokale Gesamtfehler aller betroffenen Dreiecke vor und nach der Operation berechnet werden. Dies wird auch zur Bestimmung der Reihenfolge der Kantenkollaps-Operationen herangezogen. Die enormen Kosten der Fehlerberechnung treten also für jede Kante bei jeder Veränderung ihrer Nachbarschaft auf.

Ein weiteres Problem ist die fehlende Berücksichtigung der Positionsunsicherheit eines Voxels. Diese Positionsunsicherheit ist in Richtung der Kameraachse um ein Vielfaches

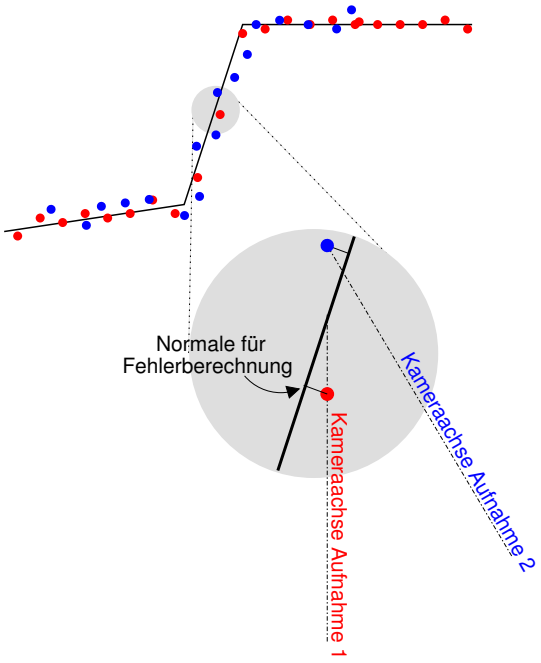


Abbildung 4.38: Die Richtung der Fehlerberechnung stimmt nicht mit der Assoziationsrichtung bei der Dreieckserstellung überein.

größer als in Normalenrichtung. Bei der Zuordnung zu einem Polygon wird daher zunächst die Blickrichtung der Kamera verwendet, bei der späteren Fehlerberechnung eines Polygons jedoch der Normalenabstand des Voxels vom Polygon. Abbildung 4.38 skizziert das Problem in zwei Dimensionen.

Dies führt bei Polygonen, die nicht senkrecht zur Kamera sind, zu Problemen. Da jedoch die Kamera auch als translatorisch beweglich angenommen wird, ändert sich ihre Blickrichtung. Das bedeutet, dass zu jedem Voxel die Richtung, aus der es gesehen wurde, von Bedeutung ist. Mit dieser Zusatzinformation würde sich die Komplexität des Systems jedoch noch weiter vergrößern, da die Fehlerberechnung noch aufwändiger wird.

Experimente

Da das Verfahren aus Komplexitätsgründen nicht weiterverfolgt wurde, sei an dieser Stelle auf eine Veröffentlichung des Autors verwiesen [100], in der auf diesem Verfahren basierende Experimente dargestellt werden.

Fazit

Das Prinzip der Assoziation jedes Polygons mit seinen Basisdaten ist eine vielversprechende Lösung für das Problem der parallelen Entwicklung eines Voxel- und eines polygonalen Modells. Trotz der Einschränkung auf eine Speicherung der Voxeldaten im Form eines

Oct-trees sind die Kosten für die Verwaltung der Voxeldaten bei Netzmanipulationen zu hoch. Laufzeiten von mehreren Minuten für die Dezimierung eines Netzes mit etwa 50.000 Dreiecken sind im Kontext der Telepräsenz nicht vertretbar.

Heckbert und Garland ordnen das mit der beschriebenen Methode verwandte Verfahren von Soucy und Laurendeau ähnlich ein:

„Their method appears to yield higher quality results than the method of Schroeder et al. but it is slower and it uses more memory since it maintains lists of all deleted points“ [36]

Die hier zitierte Methode von Schroeder et al. [79] nutzt zur Fehlerüberwachung nur ein relatives und netzimmanentes Maß, verzichtet also auf die Speicherung der Voxel.

5 Szenenrekonstruktion im Telepräsenzkontext

In diesem Kapitel wird das bisher vorgestellte System zur Umgebungsmodellierung auf der Basis von Stereo-Kamerabildern in den Kontext des Teilprojektes „Übertragungszeitkompensation durch Szenenprädiktion“ des Sonderforschungsbereichs gestellt. Dabei steht hier die Implementierung und weitere Untergliederung in Softwarekomponenten im Vordergrund. So dient dieses Kapitel einerseits der Darstellung des Gesamtzusammenhangs und andererseits der Dokumentation weiterer Komponenten, die in dieser Arbeit erstellt wurden.

Abbildung 5.1 stellt die Grobgliederung des technischen Systems dar. Sie wird im folgenden Abschnitt in einzelne Funktionsblöcke unterteilt, deren Funktion jeweils detaillierter dargestellt wird.

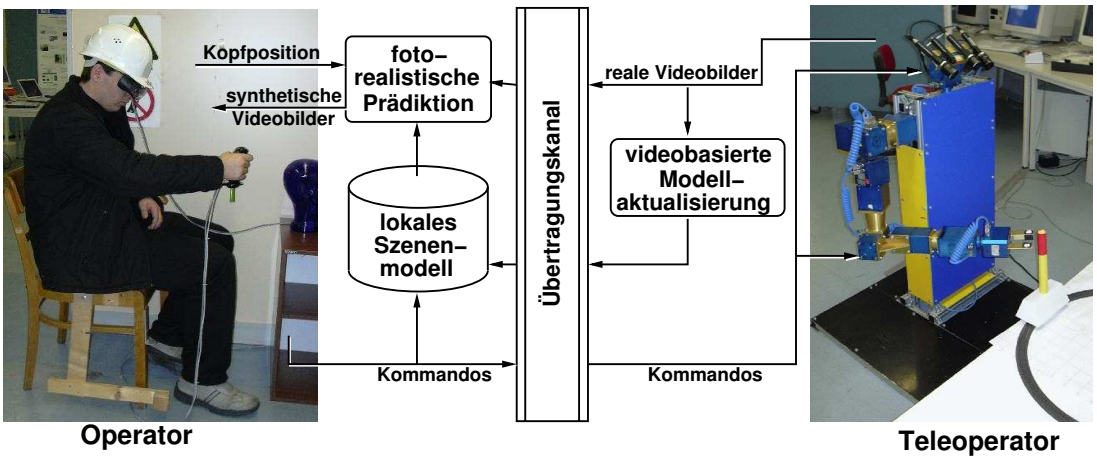


Abbildung 5.1: Übertragungszeitkompensation durch Szenenprädiktion – Übersicht.

5.1 Ergänzende Softwaremodule und ihre Aufgaben

Abbildung 5.2 zeigt alle beteiligten Komponenten. Dabei wurden alle weiss hinterlegten Komponenten vom Autor dieser Arbeit erstellt. Die Funktionalität grau hinterlegter Module wird jeweils nur kurz zusammengefasst, um den Gesamtzusammenhang darstellen zu können. Besonders herauszuheben sind die Roboterkinematik und Bahnplanung, da

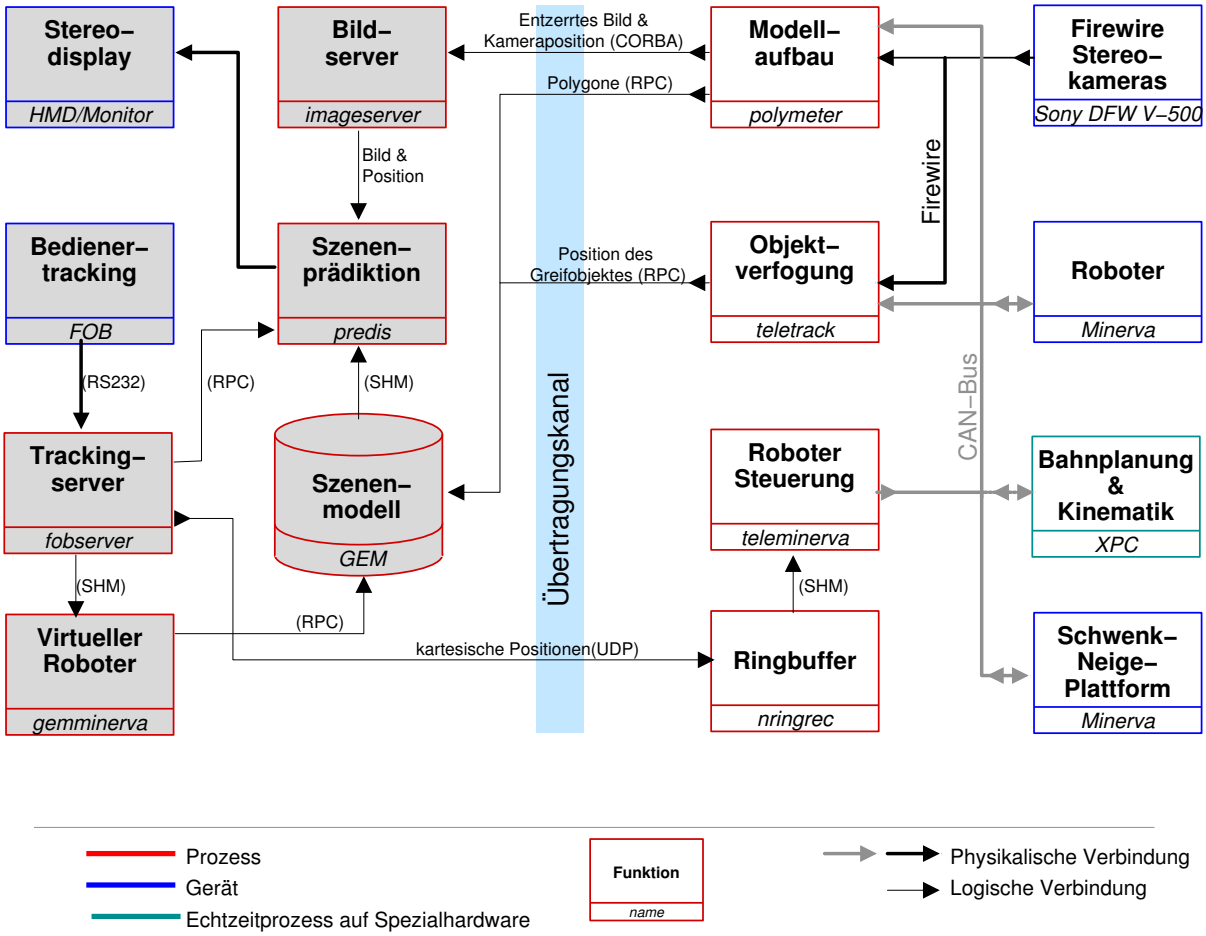


Abbildung 5.2: Übertragungszeitkompensation durch Szenenprädiktion – Funktionale Komponenten.

es sich hierbei um komplexe Eigenentwicklungen handelt, die speziell auf den Einsatz in der Telepräsenz zugeschnitten sind. Nicht minder wichtig ist ihre Implementierung, die in Form eines automatisierten modellbasierten Entwurfs durchgeführt werden konnte.

5.1.1 Firewire-Stereo-Kameras

Eingesetzt werden zwei Firewire-Kameras des Typs Sony DFW V-500 mit einfachen Objektiven mit 6 mm Brennweite. Sie besitzen eine Auflösung von 640 x 480 Pixel und sind extern synchronisierbar. Zirkular polarisierende Filter reduzieren unter Ausnutzung der Tatsache, dass durch den Brewster-Effekt viele Glanzlichter polarisiert sind, Reflexionen in den Bildern. Die Kameras werden im Modus YUV422 betrieben, um die maximal mögliche Farbinformation zu erhalten. Dadurch sinkt aufgrund der verfügbaren Busbandbreite ihre Bildwiederholrate auf nur 15 Bilder/s. Zur Stereorekonstruktion werden nur die Grauwerte (Y-Kanal) genutzt, zur Objektverfolgung jedoch die unterabgetasteten Farbkanäle. Der in Abschnitt 3.1.1 genannte Vorteil der digitalen Firewire-Schnittstelle kann hier praktisch

ausgenutzt werden, indem mehrere PCs unabhängig auf den selben Bilddaten operieren können.

Im minimalinvasiven Chirurgieszenario wurde an Stelle der Kameras die SFB-weite Kommunikationsbibliothek *sfbcomm* eingesetzt um die Bilder der analogen Endoskop-Kameras zu empfangen.

5.1.2 Roboter

Bei dem Roboter handelt es sich um ein modulares System der Firma Amtec. Jedes Gelenk ist via CAN-Bus als adressierbare intelligente Einheit ansprechbar und kann so kommandiert werden, feste Positionen, Geschwindigkeiten oder Beschleunigungen zu fahren. Durch die Bandbreite des CAN-Busses und die Anzahl an Gelenken ist die mögliche Schrittzeit auf etwa 3 ms begrenzt. Der Arm ist in einer redundanten, antropomorphen Anordnung mit sieben Gelenken konfiguriert und besitzt einen einfachen Zweibackengreifer, der ebenfalls über CAN-Bus steuerbar ist.

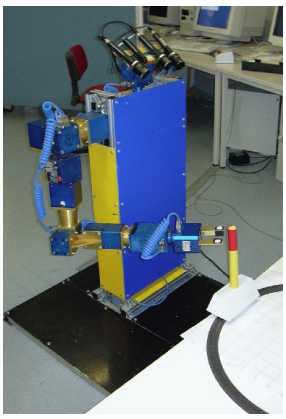


Abbildung 5.3: Roboter Minerva

5.1.3 Bahnplanung und Kinematik

Aufgabe Da das modulare Robotersystem nur gelenkweise angesprochen werden kann, übernimmt dieses Modul die Umsetzung vom kartesischen Raum in den Gelenkwinkelraum. Da der Roboter mit sieben Gelenken und sechs Freiheitsgraden Redundanz besitzt, wird diese in der Rückwärtskinematik durch die Verwendung der Hand- und Ellbogenposition des Bedieners gebunden. Das Konzept ist in Abbildung 5.4 skizziert.

- Beim Start des Systems ist die Handposition des Bedieners sowie die Handposition des Roboters per Definition bekannt und festgelegt.
- Der Roboter-'Ellbogen' muss auf einer Kugel mit einem Radius der Länge des 'Oberarms' des Roboters um die 'Schulter' des Roboters liegen.
- Gleiches gilt für eine Kugel um die Handposition mit dem Radius des 'Unterarms'.
- Der Ellbogen des Roboters liegt damit auf dem Schnittkreis dieser zwei Kugeln.
- Der 'Ellbogen' des Roboters wird an den Punkt gelegt, der den kürzesten Abstand zur Position des Ellbogens des Bedieners aufweist (Sie ist durch das magnetische Tracking bekannt).

Durch diese Methode lässt sich zunächst die Ellbogenposition des Roboters festlegen. Dadurch kann die Rückwärtskinematik geschlossen gelöst werden. Somit ist die Rückwärtskinematik in der Lage, bei bekannter Hand- und Ellbogenposition des Bedieners gültige Gelenkwinkel für den Roboter zu berechnen.

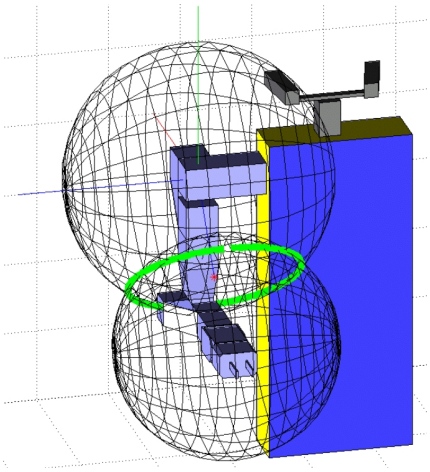


Abbildung 5.4: Bindung der Redundanz der Manipulators: Der Ellbogen liegt immer auf dem Schnitt zweier Kugelschalen. Die Position auf dem Schnittkreis wird durch die Lage des Bedienerellbogens festgelegt.

Die Bahnplanung bildet verschiedene Zeitbasen aufeinander ab: Neue Positionen des magnetischen Trackingsystems werden mit einem Takt von etwa 100 ms übertragen. Die Robotersteuerung läuft jedoch mit einem festgelegten Takt von 10 ms. Mit Hilfe einer Bahnplanung (einfache Extrapolation) werden diese beiden Takte aneinander angepasst. Die Bahnplanung bildet somit die Schnittstelle zwischen ereignisgesteuertem und zeitgesteuertem System. Weiterhin ist dieses Modul für ein geregeltes Start-Stopp-Verhalten des Roboters, (Selbst-) Kollisionsvermeidung, Fehlerabschaltung und Arbeitsraumüberwachung zuständig. Die gesamte Kommunikation mit dem Roboter und der Steuerung erfolgt über CAN-Bus.

Implementierung Das Modul wurde durch modellbasiertes Rapid-Prototyping unter Matlab/Simulink erstellt. So dient dasselbe Modell unter Simulink zur Simulation und Implementierung. Dabei kommt das XPC-Target als Zielsystem zum Einsatz. Dabei handelt es sich um einen Standard-PC geringer Leistung als Echtzeitplattform unter einem proprietären Minimal-Betriebssystem. Die gesamte Kommunikation mit der Außenwelt erfolgt durch den CAN-Bus.

Vorteilhaft bei dieser Lösung ist der geringe Implementierungsaufwand und die klare Trennung schritthaltender Systemteile von nicht realzeitfähigen Teilen. Außerdem konnte die Rückwärtskinematik unter Simulink als Linux-Bibliothek exportiert werden und zur Animation der kinematischen Kette des virtuellen Roboters eingesetzt werden.

5.1.4 Schwenk-Neige-Plattform

Die Schwenk-Neige-Plattform besteht ebenfalls aus Robotermodulen der Firma Amtec. Sie wird jedoch nicht von der Bahnplanung und Kinematik gesteuert, sondern erhält

ihre Kommandos direkt von der Robotersteuerung als Reaktion auf Kopfbewegungen des Bedieners.

5.1.5 Modellaufbau

Dieses Modul erzeugt eine fortschreitende polygonale Rekonstruktion der betrachteten Szene, wie in dieser Arbeit beschrieben. Dabei werden Einzelbildpaare des Stereosystems mit bekannter Position des Kamerakopfes (Schwenk-Neige-Plattform) und bekannter Position des Roboters (Ausmaskierung uninteressanter Bereiche) zu einer Modellrekonstruktion verrechnet. Die Haltung der akkumulierten Modelldaten erfolgt lokal. Veränderungen werden zusammen mit Bilddaten an den Bildserver bzw. die Modelldatenbank übertragen.

5.1.6 Objektverfolgung

Hierbeit handelt es sich um ein einfaches farbbasiertes Objekttrackung, das in der Lage ist, ein festgelegtes Greifobjekt schritthaltend mit dem Kameratakt zu verfolgen. Dabei wird auf einem dedizierten PC im U- und V-Farbanteil der Bilder beider Kameras nach roten Flächen bestimmter Abmessungen gesucht. Nach der zweidimensionalen Bildung der Symmetrieachse kann eine Rekonstruktion in 3D erfolgen. Durch einen Konsistenztest (3D-Daten vs. bekannte Objektgeometrie) kann das System auch geringe Verdeckungen des Objektes detektieren.

Bei diesem Trackingsystem handelt sich lediglich um einen funktionierenden Prototypen für die Aktualisierung des Szenenvordergrundes. Durch Filterung der Daten konnte bereits eine stabile Rekonstruktion mit geringer Latenz erreicht werden. Latenzzeiten zwischen der Messung der Kameraposition (Schwenk-Neige-Einheit) und der Bildaufnahme führten jedoch zu unerwünschten Objektbewegungen bei Kopfbewegungen des Bedieners.

5.1.7 Robotersteuerung

Die Robotersteuerung empfängt im laufenden Betrieb die Rohdaten des Bedienertrackings und versendet Kommandos an die Bahnplanung und die Schwenk-Neige-Plattform. Wie bereits weiter oben erläutert, ist die Handposition des Bedieners und die Roboterposition bei Systemstart bekannt. Die an dieser Stelle notwendige Koordinatentransformation wird bei Systemstart berechnet und im laufenden Betrieb jeweils auf die eingehenden Trackingdaten angewendet. Weiterhin ist das Modul zuständig für die Roboterinitialisierung und Start- und Stoppverhalten, für den Start und Stopp der XPC-Plattform und die Ansteuerung des Kamerakopfes. Da der Kamerakopf nur rotatorisch in zwei Achsen bewegt werden kann, wird die absolute Kopfposition (nur Schwenken und Neigen) des Bedieners nach einer Bereichsüberprüfung direkt an die Schwenk-Neige-Plattform gesendet. Das Modul gibt ebenfalls den Zeittakt vor, in dem der Ringbuffer gelesen wird und somit neue Zielpositionen an den Roboter geschickt werden.

5.1.8 Ringbuffer

Der Ringbuffer dient der Einstellung einer simulierten Zeitverzögerung des Kommunikationskanals. Da es in einer derartigen Telepräsenzverbindung unerheblich ist, ob Zeitverzögerung im Hin- und Rückkanal auftritt oder nur in einer Richtung, kann hier durch Auslesen der veralteten Einträge des Ringbuffers eine Zeitverzögerung simuliert werden.

5.1.9 Virtueller Roboter

Der virtuelle Roboter ist das Spiegelbild des echten Roboters im Szenenmodell. Als freilaufender Prozess setzt er die Bedienerkommandos in Positionen einzelner Objekte im Modell um. Hierfür nutzt er dieselbe Rückwärtskinematik, die auch der reale Roboter verwendet. Wie oben beschrieben kommt für die Erzeugung dieser Kinematik ebenfalls der modellbasierte Entwurf unter Matlab/Simulink zum Einsatz. Dynamische Aspekte des Systems sind jedoch nicht modelliert, da das Modell keine Zeitbasis besitzt.

5.1.10 Sonstige Module

Bildserver

Der Bildserver empfängt entzerrte Bilder mit ihrer Kameraposition und stellt sie auf Anfrage der Szenenprädiktion zur Verfügung. Der Empfang eines Bildes stellt dabei den Trigger für einen Aktualisierungsschritt auf Darstellungsseite dar.

Szenenprädiktion

Die Szenenprädiktion berechnet aus polygonalen Modelldaten und passenden Kamerabildern fotorealistische Ansichten der Szene aus beliebigen Blickwinkeln. Die Extraktion der Texturen aus Kamerabildern wird dabei aufgrund der hohen Leistung direkt auf der Grafikkarte vorgenommen. Dabei werden Texturen aus unterschiedlichen Ansichten fusioniert, Löcher in den Texturen gefüllt und Verdeckungen berücksichtigt. Die Darstellung erfolgt schritthaltend mit den Bewegungen des Bedieners.

Szenenmodell

Die Modelldatenbank GEM ist gewissermaßen der Dreh- und Angelpunkt des Systems. Sie speichert polygonale Modelle in objektorientierter Form und ist durch RPC und Shared-Memory-Schnittstellen an die Außenwelt angebunden. Sie enthält das Modell des Szenenhintergrundes, wie es vom Modellaufbau erzeugt wird, die vorab bekannten Objekte im Vordergrund der Szene (z. B. Tisch, Greifobjekt) und die kinematische Kette des Roboters.

Bedingt durch verschiedene Latenz- und Übertragungszeiten gibt es keinen einzelnen aktuellen Systemzustand. Vielmehr laufen unterschiedliche Modelle parallel. Dies ist durch ein Inselkonzept gelöst:

- **Die Modellinsel** empfängt die berechneten Modelldaten vom Modellaufbau. Sie speichert den Zustand des Systems bei der Bildaufnahme ab.
- **Die Texturinsel** dient der Texturextraktion. Da dieser Prozess Zeit benötigt, muss sichergestellt sein, dass sich der Szeneninhalt währenddessen nicht verändert. Die Insel wird durch Kopie aus der Modellinsel befüllt.
- **Die aktuelle Insel** spiegelt den Zustand des Modells, den der Bediener präsentiert bekommt. Bewegt er seine Hand, so wird in dieser Insel die virtuelle Roboter bewegt. Ihr Hintergrund wird aus der Texturinsel befüllt. Positionen von Objekten werden laufend aktualisiert.

Stereodisplay

Die Darstellung kann in Stereo oder Mono auf Monitor (Shutterbrille) oder Head-Mounted-Display erfolgen. Echte Immersion ist dabei nur mit HMD möglich.

Tracking-Server

Der Trackingserver bindet das System zum magnetischen Tracking des Bedieners softwaretechnisch ein. Die Ergebnisse stehen in Form von *shared-memory* mehreren Prozessen zur Verfügung. Gleichzeitig werden sie via UDP-Schnittstelle an die robotische Seite übertragen.

Bedienertracking

Eingesetzt wird mit dem „flock-of-birds“ ein magnetisches Trackingsystem der Firma Ascension. Ein magnetischer Transmitter ermöglicht es, die Position und Orientierung dreier Sensoren in einem Radius von etwa 3 m um den Sender zu messen. Sensor eins befindet sich am Head-Mounted-Display, Sensor zwei in einem Handgriff, mit dem der Roboter gesteuert wird und Sensor drei am Ellbogen des Bedieners.

6 Ergebnisse

Dieses Kapitel stellt Resultate der Szenenrekonstruktion, die komponentenweise in den vorangegangenen Kapiteln beschrieben wurde dar. Bei der Anwendung des Systems im Umfeld der Telepräsenz ist der zentrale Parameter das Präsenzempfinden für den Bediener. Wie bereits im Kapitel 1.3 aufgeführt, haben drei Faktoren wesentlichen Einfluss darauf: Die Modellierungsgeschwindigkeit, die Qualität der Modellierung und die Darstellungsgeschwindigkeit.

Dieses Kapitel zeigt anhand von Szenen geringer mittlerer und hoher Komplexität die Leistung des Systems. Es werden die beiden zueinander dualen Faktoren Dreiecksanzahl (bestimmt die Darstellungsgeschwindigkeit) und subjektive Qualität der Modellierung dargestellt, indem Modelle jeweils in verschiedenen Dezimierungsstufen bis hin zum Punkt der Netzdegenerierung aus verschiedenen Kameraperspektiven gezeigt werden. Als Abschluss der Testszenen aus Innenräumen bezieht eine Diskussion zu jedem der oben genannten Bewertungsfaktoren noch einmal abschließend Stellung.

Am Ende des Kapitels wird an einem Anwendungsszenario aus der minimalinvasiven Chirurgie gezeigt, dass die Verfahren auch in anderen Anwendungsdomänen einsatzfähig und nutzbringend sind.

6.1 Einfache Szene

Erzeugung Eine einfache Szene aus einigen ausgedehnten und gut texturierten Objekten dient als erste Testsequenz. Die Szene besteht aus zehn Einzelbildern in der Auflösung 640x480, wie sie in Abbildung 6.1 dargestellt sind. Die Kamera befindet sich auf einer kalibrierten Schwenk-Neige-Plattform, wodurch Positionsdaten für alle Aufnahmen verfügbar sind, jedoch nahezu keine translatorische Bewegung der Kamera möglich ist. Die Aufnahmen umfassen eine Entfernungsbereich von etwa 0,5 - 3m, was einer Disparität von etwa 140 bis 20 entspricht. Durch die gut texturierten Oberflächen können weitgehend vollständige Disparitätenkarten für jede Einzelaufnahme erstellt werden, wie Abbildung 6.1 unten zeigt. Die Korrelation erfolgt durch normierte Kreuzkorrelation auf quadratischen Fenstern der Abmessungen 15x15 Pixel. Aus Gründen der Laufzeit kommt die dynamische Programmierung mit direkter Nachbarschaftssuche zum Einsatz. Ein Rechts-Links-Konfidenztest reduziert Fehlkorrespondenzen. Eine subpixelgenaue Verfeinerung der Disparitätenkarten führt zu einer Genauigkeit von etwa $\frac{1}{4}$Disparitätsschritt. Ausreißer in den Disparitätenkarten werden entfernt und kleine Löcher durch Medianfilte-

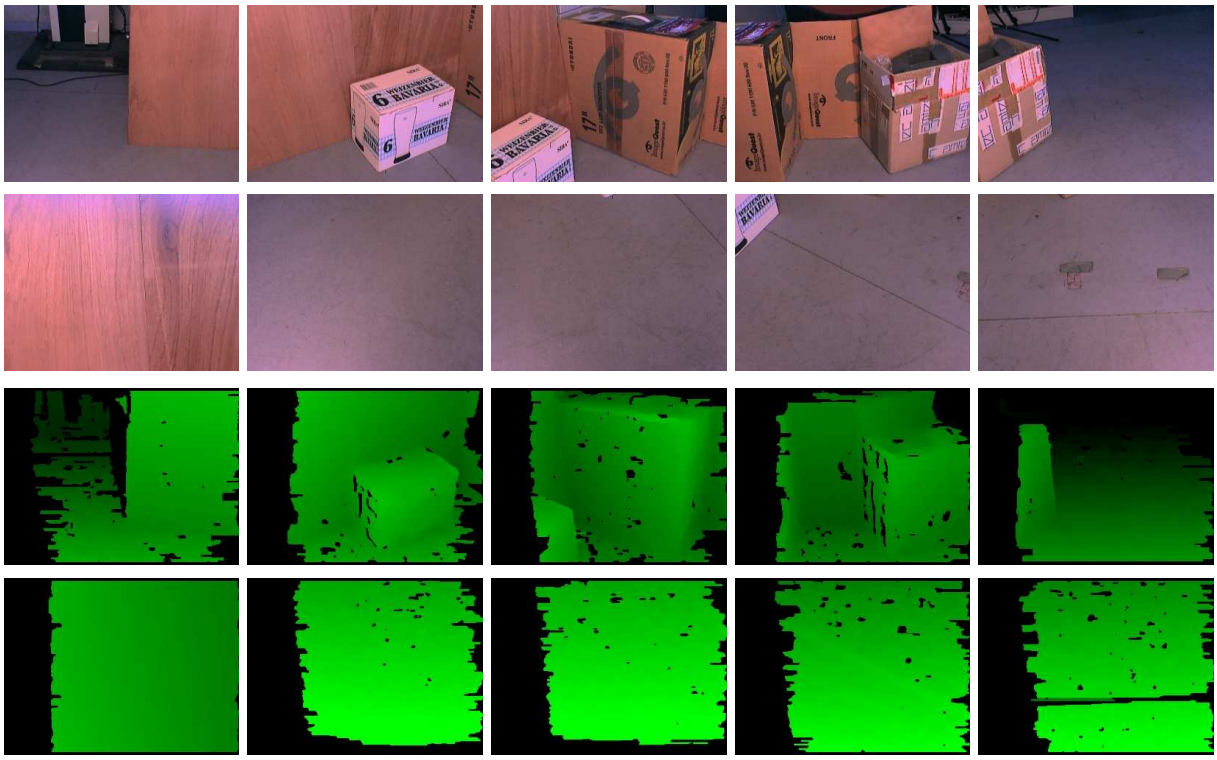


Abbildung 6.1: Zehn Einzelaufnahmen der Serie 1 mit ihren jeweiligen Disparitätenkarten (Kontrast zur Darstellung erhöht).

rung (r=3) gefüllt. Größere Löcher werden nicht gefüllt. Eine adaptive Mittelwertfilterung glättet Oberflächen in der Szene, um die Netzerzeugung zu vereinfachen.

Ergebnisse Das aus allen zehn Aufnahmen resultierende kolorierte Voxelmodell ist in Abbildung 6.2 aus vier unterschiedlichen Kamerapositionen dargestellt. Die große Genauigkeit der Rekonstruktion bei dieser einfachen Szene erlaubt eine Darstellung mit sehr guter Qualität auch bei großer translatorischer Abweichung der Betrachterposition von der Aufnahmeposition. Die Bildqualität ist hervorragend und fast frei von Störungen.

Die Triangulierung erfolgt durch verfeinernde Triangulierung, bis eine Dreieckszahl von etwa 10.000 pro Ansicht erreicht ist. Die Fusion der Einzelaufnahmen wird wie in Abschnitt 4.5 beschrieben nach dem Hinzufügen jeder Einzelaufnahme durchgeführt. Dabei wird der Fusionsbereich, also der Bereich in dem Disparitätenkarte und Modell als gleich erkannt werden, sehr groß gewählt (±25 Disparitäten) und Fehler im Netz nicht entfernt.

Löcher in der Disparitätenkarte werden durch Triangulierung geschlossen, und Tiefensprünge werden nicht geöffnet. Die nachfolgende Dezimierung reduziert jedes Einzelnetz bis auf 220 Dreiecke.

Abbildung 6.3 zeigt das texturierte Netz für eine Gesamtzahl von 2.200 Dreiecken aus vier Ansichten. Kleine Löcher im Netz entstehen beim Vernähen der Ansichten. Die Ursa-

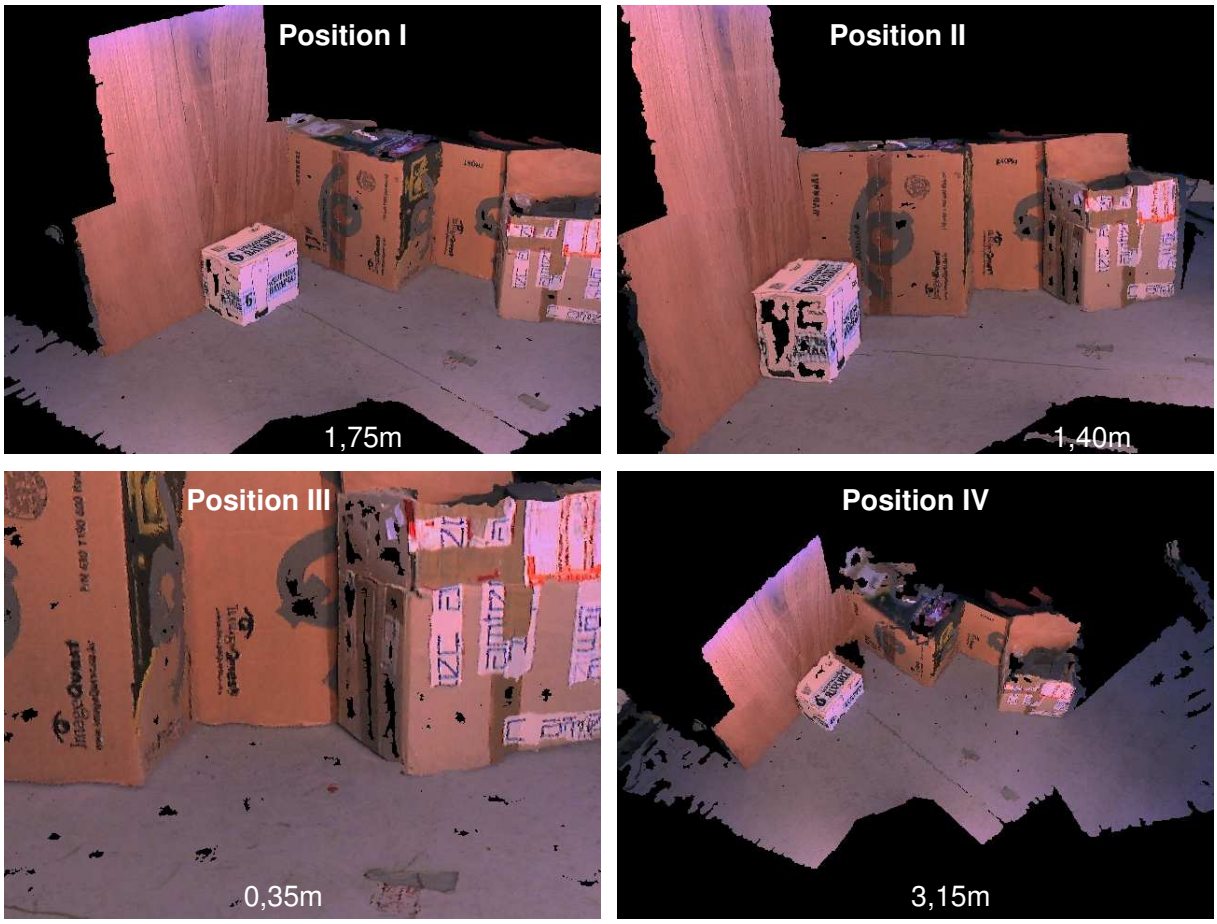


Abbildung 6.2: Das Voxelmodell der Serie 1 aus vier unterschiedlichen Positionen betrachtet. Angegeben ist die translatorische Entfernung zur Aufnahmeposition.

che liegt in der Anwendung eines zweidimensionalen Verfahrens. Ohne eine nachfolgende Analyse der entstandenen Löcher in 3D und gegebenenfalls einer Reparatur kommt es zu unnötigen Öffnungen im Netz. Ähnliches gilt für das Schließen von Löchern in der Disparitätenkarte: Innerhalb der Karte werden Löcher geschlossen. Befinden sie sich jedoch am Rand der Aufnahme, so werden sie erst nach der Fusion mehrerer Aufnahmen zu Löchern im Netz. Diese werden nicht geschlossen und bleiben bestehen.

Das Modell enthält an der Komplexität der Szene gemessen zu viele Dreiecke. Aufgrund der Voraussetzung, dass ein erstelltes Teilnetz nicht mehr verändert werden sollte, wenn sich die Szene nicht verändert, folgt automatisch eine Beschränkung der Dezimierung auf das gerade neu hinzugefügte Teilnetz. Dadurch muss die Triangulierung an der Nahtstelle zweier Teilnetze unverändert bleiben. Die daraus resultierende Einschränkung der Netztopologie hat eine unnötig große Anzahl von Dreiecken im Gesamtnetz zur Folge.

Abbildung 6.4 zeigt dasselbe Modell nach globaler Dezimierung. Hier wurde die Voraussetzung der lokalen Dezimierung aufgegeben. Somit sind deutlich geringere Dreieckszahlen bei Erhalt der Geometrie möglich. Einer weiteren Dezimierung stehen die kleinen Löcher

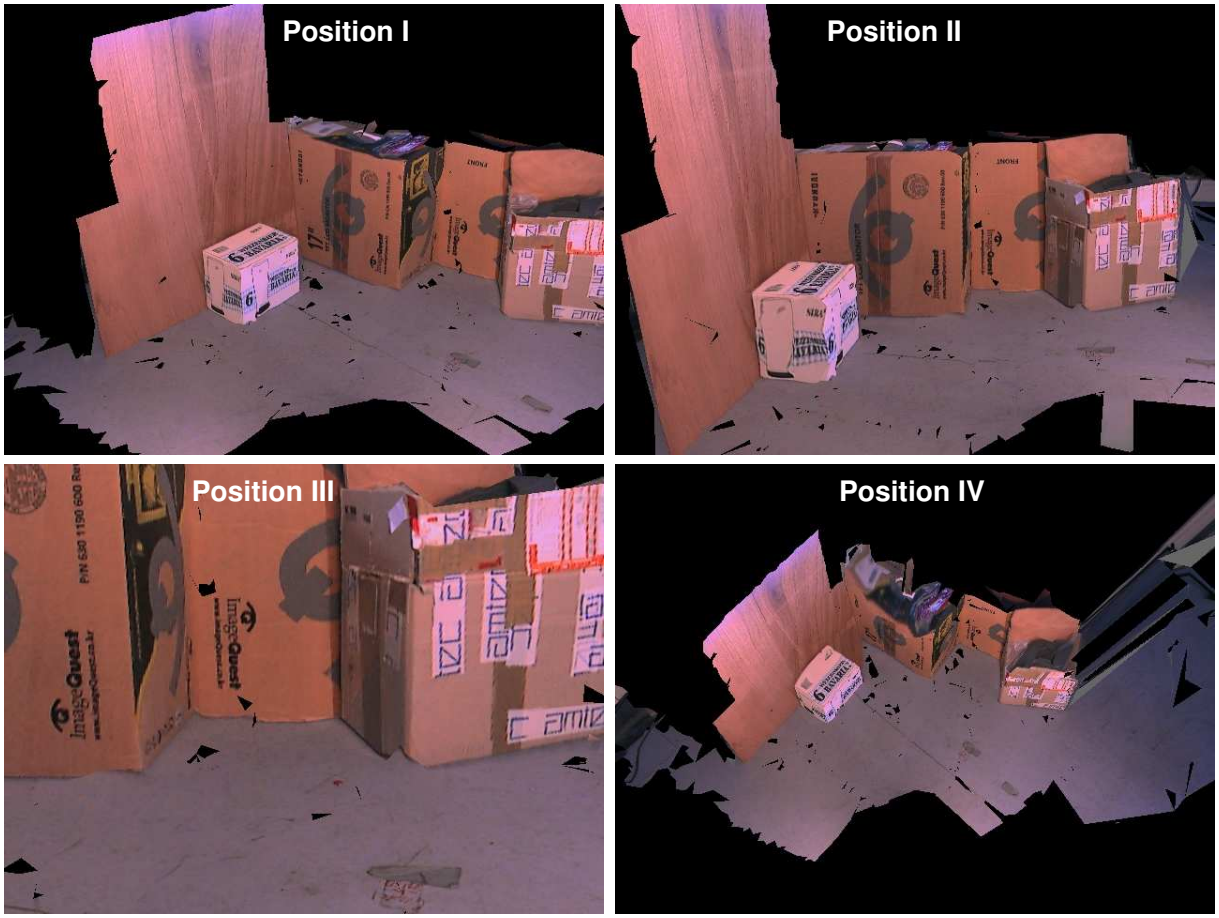


Abbildung 6.3: Das texturierte Dreiecksnetz der Testserie 1 mit ca. 2.200 Dreiecken und geschlossenen Tiefensprüngen. Bei weiterer Dezimierung kommt es zur Netzdegeneration durch die ungünstige Topologie insbesondere am rechten Szenenrand.

an den Nähten der Netze im Weg. Bei globaler Dezimierung müsste jedoch ein Konzept zur Weiterverwendung der Texturen bei Veränderung der dazugehörigen Polygone gefunden werden.

Abbildung 6.5 zeigt dieselbe Szene bei geöffneten Tiefensprüngen. Die Frage, ob Tiefensprünge im Netz geöffnet werden sollten, kann jedoch nicht eindeutig geklärt werden. So sind die sehr langgezogenen Polygone mit fehlerhaften Texturen in Abbildung 6.3 IV und 6.4 IV deutliche Störungen, die in Abbildung 6.5 nicht mehr enthalten sind. Bei dem von der weißen Kiste verdeckten Bereich kehren sich diese Verhältnisse jedoch um. So ist sowohl der Fall geöffneter Diskontinuitäten als auch der Fall geschlossener Tiefensprünge mit Bildstörungen verbunden die die Darstellungsqualität, stören.

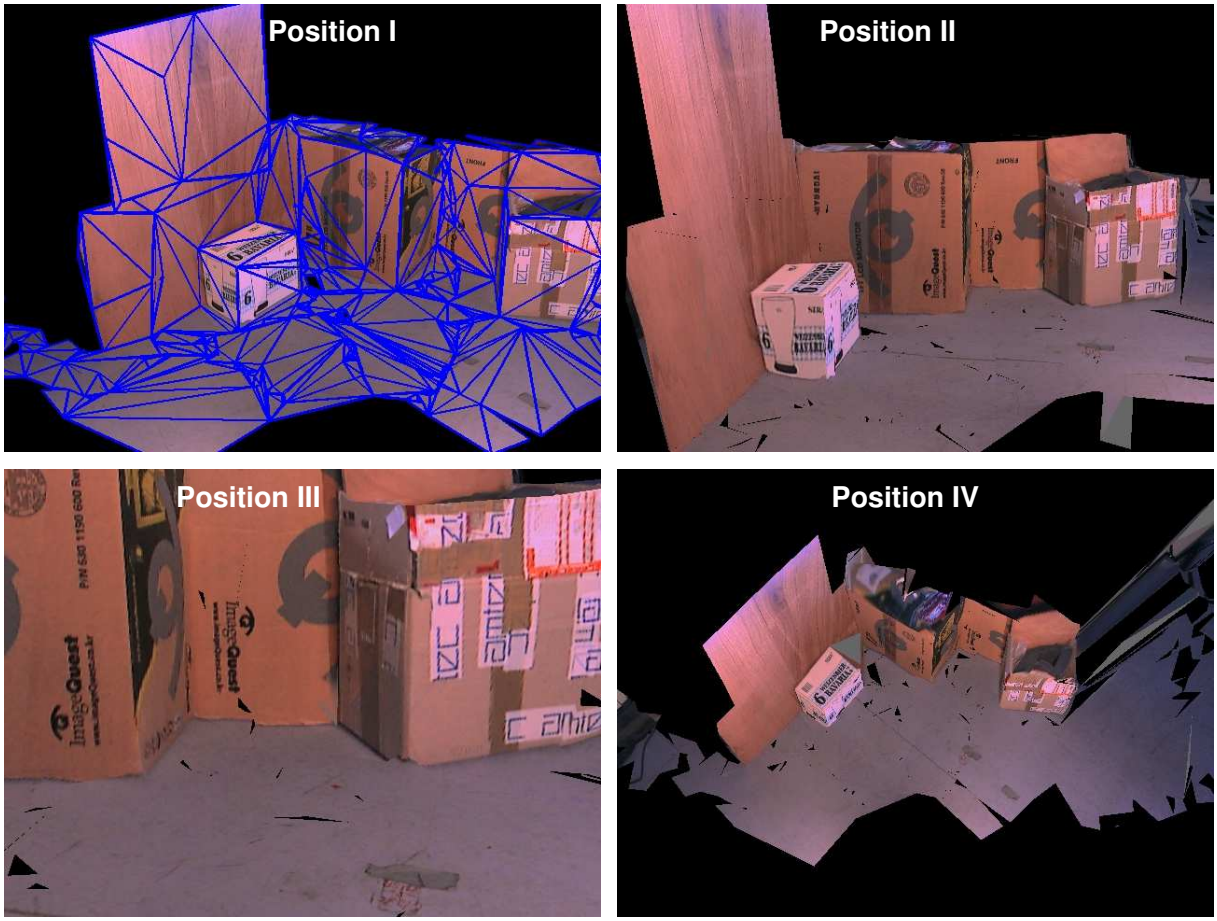


Abbildung 6.4: Das texturierte Dreiecksnetz der Testserie 1 nach globaler Dezimierung. Hier wurde die Beschränkung aufgegeben, nur das neu hinzugefügte Teilnetz lokal zu dezimieren. Das Netz ist mit ca. 1.100 Dreiecken nahezu halb so groß wie das in Abbildung 6.3 dargestellte und zeigt noch keine Degeneration.

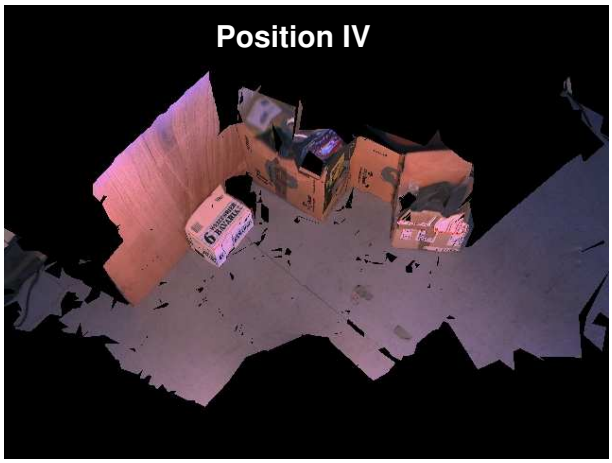


Abbildung 6.5: Das texturierte Dreiecksnetz der Serie 1 nach globaler Dezimierung bei geöffneten Tiefensprüngen.

Disparitätenkarte	9,2 s	9,2 s
Filterung der Disparitätenkarte	1,1 - 1,7 s	0,1 s
Vergleich zw. Modell und Disparitätenkarte	0,2 s	0,2 s
Triangulierung inkl. Vernähen	3,1 s	2,1 - 2,8 s
Netzbereinigung	0,2 - 0,6 s	0,2 - 1,0 s
Netzdezimierung	6,0 - 8,3 s	2,7 - 3,5 s
$\sum$ (gemessen)	21,7 - 29,6 s	16,6 - 17,7 s

Tabelle 6.1: Die Tabelle stellt die gemessenen Zeiten für einzelne Verarbeitungsschritte dar. Die linke Spalte korrespondiert mit der Parametrierung des Systems, wie sie oben beschrieben ist, die rechte Spalte gilt für eine Parametrierung für möglichst schnelle Ausführung bei reduzierter Qualität.

Laufzeit In allen bisher dargestellten Versuchen wurde die Parametrierung des Systems ohne Rücksicht auf Rechenzeiten vorgenommen. Insbesondere durch die Art der Filterung der Disparitätenkarten und die Netzerzeugung und Dezimierung lassen sich die Rechenzeitanforderungen und die Qualität der Resultate beeinflussen.

Tabelle 6.1 stellt die Rechenzeiten des Systems aufgeschlüsselt nach Einzelschritten dar. Dabei korrespondiert die linke Spalte mit dem Modell aus Abbildung 6.3. Bei Verzicht auf Filterung der Disparitätenkarte und reduzierter Dreieckanzahl (Triangulierung: 2.000 Dreiecke/Teilbild – Dezimierung 220 Dreiecke/Teilbild) sind Werte wie rechts in der Tabelle angegeben erreichbar. Dabei leidet natürlich die Qualität des Modells. Das resultierende Netz stellt Abbildung 6.6 links im Vergleich zu einem Netz guter Qualität (rechts) dar. Die Berechnungen wurden auf einem Intel Pentium mit 3,2 GHz und 512 MB RAM durchgeführt. Die angegebenen Zeiten verstehen sich nicht als präzise Angaben - sie enthalten Systemausgaben (Konsole und Grafik) und wurden in einer üblichen Multitasking-Umgebung berechnet, der Prozess wurde also durch Systemprozesse unterbrochen. Insbesondere die Filterung der Disparitätenkarte und das Vernähen der Netze sind nicht optimal implementiert. Bei der Stereokorrelation wurden laufende Summen zur Beschleunigung eingesetzt. Alle anderen Funktionen wurden ohne spezielle Maßnahmen zur Beschleunigung implementiert.

Abbildung 6.6: Resultierende Qualität für schnelle und normale Berechnung. Für die Berechnung des linken Bildes wurde die Tiefenkarte ungefiltert verwendet und die Anzahl der Dreiecke für die Triangulierung deutlich reduziert. Die Ansicht wurde so gewählt, dass die entstehenden Fehler deutlich sichtbar werden.

6.2 Komplexe Szene

Eine komplexe Szene mit schwach texturierten Objekten, Reflexionen, komplizierter Geometrie und schmalen Objekten stellt einen schwierigeren Testfall für das System dar. Dabei wird dieselbe Szene in zwei Varianten mit unterschiedlicher Aufnahmetechnik betrachtet. Zunächst ist das Kamerasystem noch auf einer Schwenk-Neige-Plattform montiert, wodurch kaum translatorische Kamerabewegung möglich ist und vor allem die genaue Position der Kamera gemessen werden kann. Dann wird auch diese Einschränkung aufgegeben und die Kamera frei beweglich auf einem Stativ befestigt.

6.2.1 Vorwiegend rotatorische Kamerabewegung

Die Sequenz besteht aus 20 Aufnahmen mit geringer Überdeckung. Die Kombination der Teilverfahren und ihre Parametrierung wurde wie für die einfache Szene gewählt. Größere Löcher in den Disparitätenkarten ($\varnothing < 6\,\text{Pixel}$) wurden durch die Anwendung des morphologischen Dual-Rank-Filters geschlossen. Das vereinfacht die Topologie des entstehenden Netzes und ermöglicht damit eine problemlosere Dezimierung. Abbildung 6.8 zeigt die Rekonstruktionen als Voxelmodelle. Deutlich zeigt sich hier bereits der Effekt der ausgedehnten Löcher in den Disparitätenkarten. Da derartig große Löcher nicht einfach gefüllt werden können, bleiben sie auch in den texturierten Netzen in Abbildung 6.9 sichtbar. Diese Abbildung zeigt das Netz in unterschiedlichen Dezimierungsstufen und mit geschlossenen bzw. geöffneten Tiefensprüngen.

Im Kontext der Telepräsenz sind jedoch Übersichtsansichten wie die hier vorgestellten von geringer Bedeutung. Abbildung 6.10 zeigt wesentlich relevantere Ansichten. Sie befinden sich innerhalb eines translatorischen Radius von 1 m um die ursprüngliche Kameraposition.

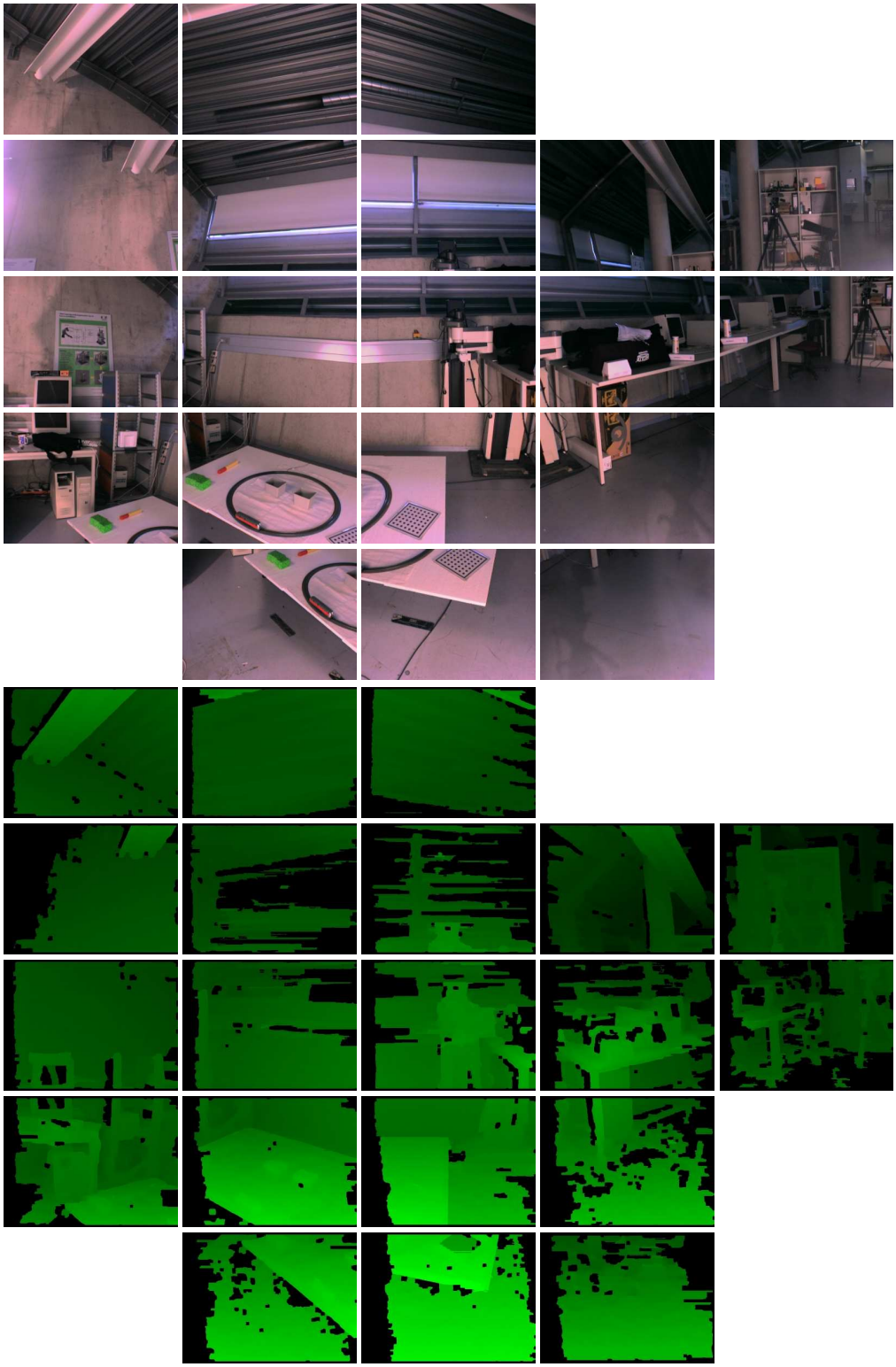


Abbildung 6.7: Zwanzig Einzelaufnahmen der Testserie 2 mit ihren jeweiligen Disparitätenbildern.

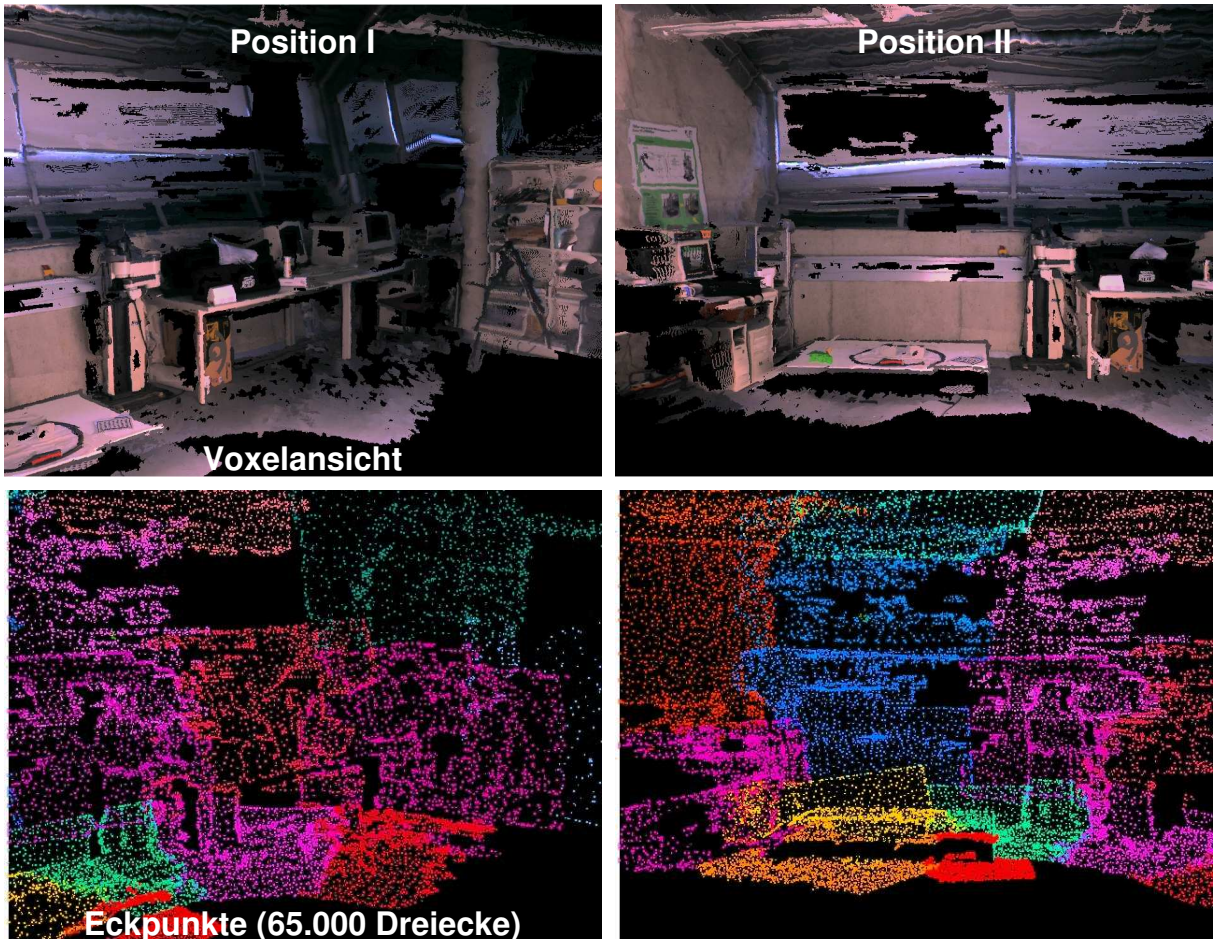


Abbildung 6.8: Voxelansicht der Testserie 2 aus zwei verschiedenen Positionen (oben) und Eckpunkte eines Netzes mit ca. 65.000 Dreiecken nach Zugehörigkeit zu einer Aufnahme koloriert.

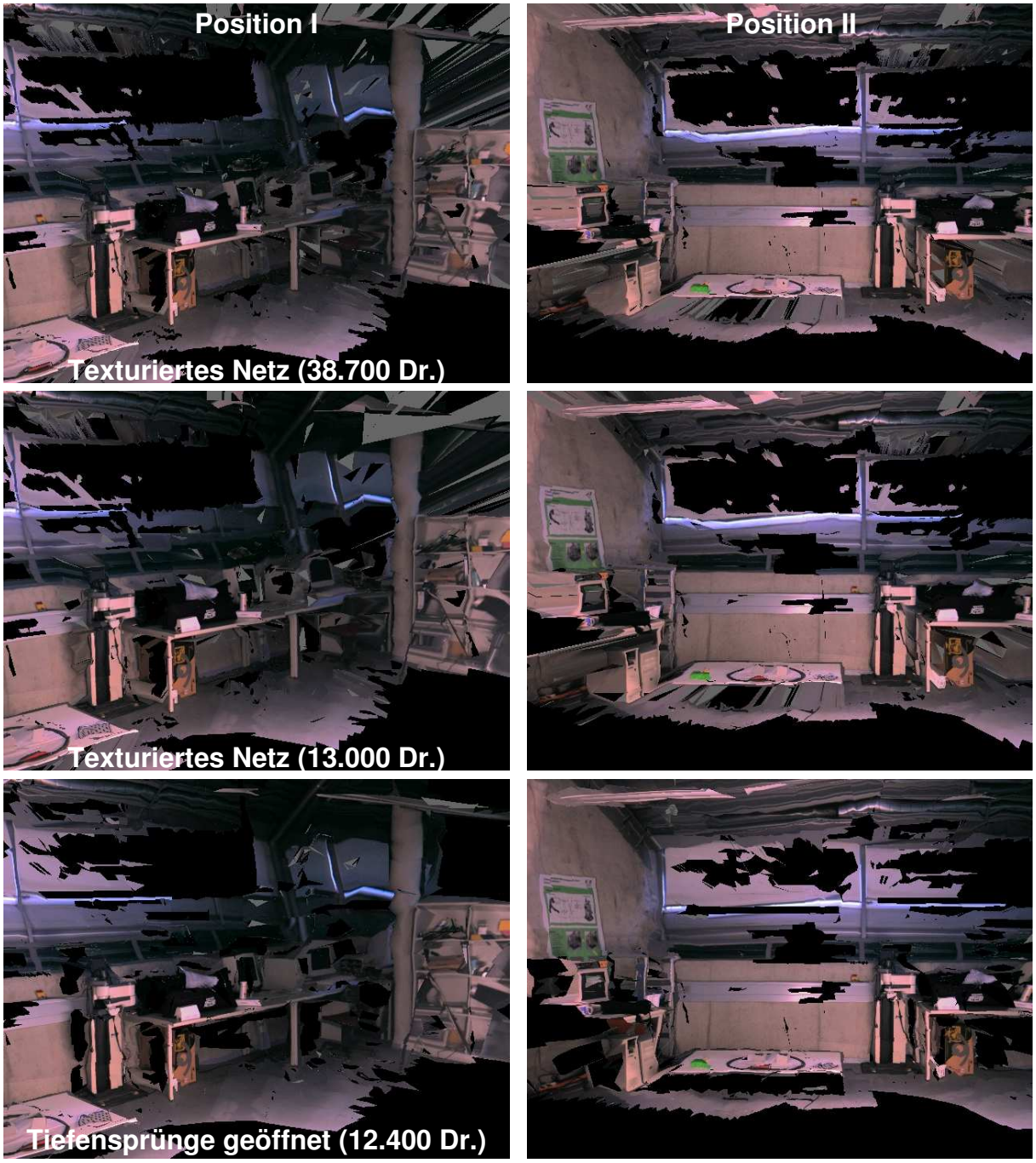


Abbildung 6.9: Texturiertes Modell der Testserie 2 aus zwei unterschiedlichen Positionen betrachtet. Die obere Reihe zeigt das Modell bei 38.700 Dreiecken, die mittlere Reihe bei 13.000 Dreiecken, der unteren Grenze, unterhalb derer Degeneration auftritt. Die untere Reihe zeigt das Netz bei geöffneten Tiefensprüngen.

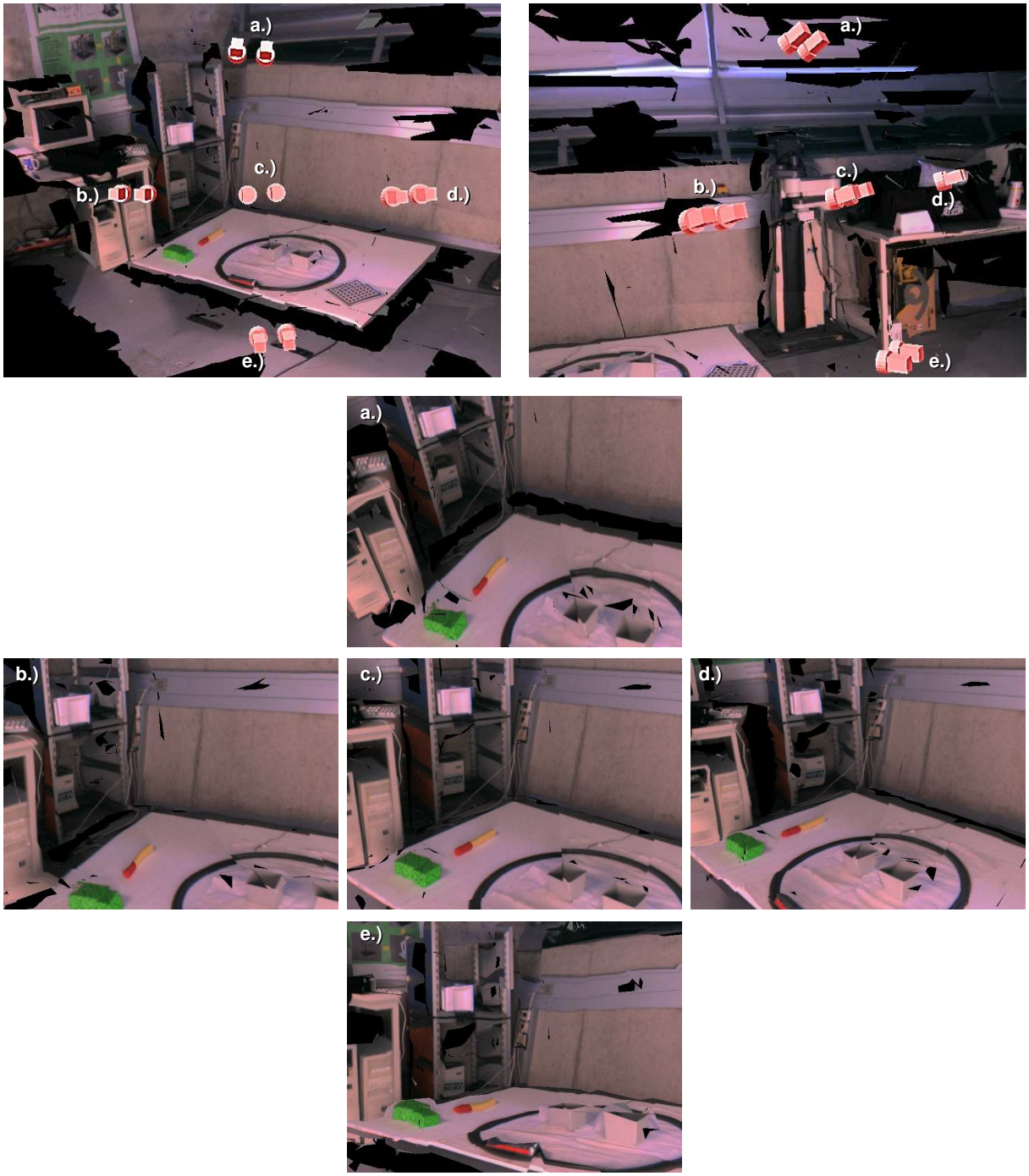


Abbildung 6.10: Texturiertes Modell der Testserie 2 aus fünf für die Telepräsenz realistischen Positionen betrachtet. Die obere Reihe zeigt die Kamerapositionen – sie befinden sich jeweils in einem Abstand von 1 m rund um die mittlere Position.

6.2.2 Rotatorische und translatorische Kamerabewegung

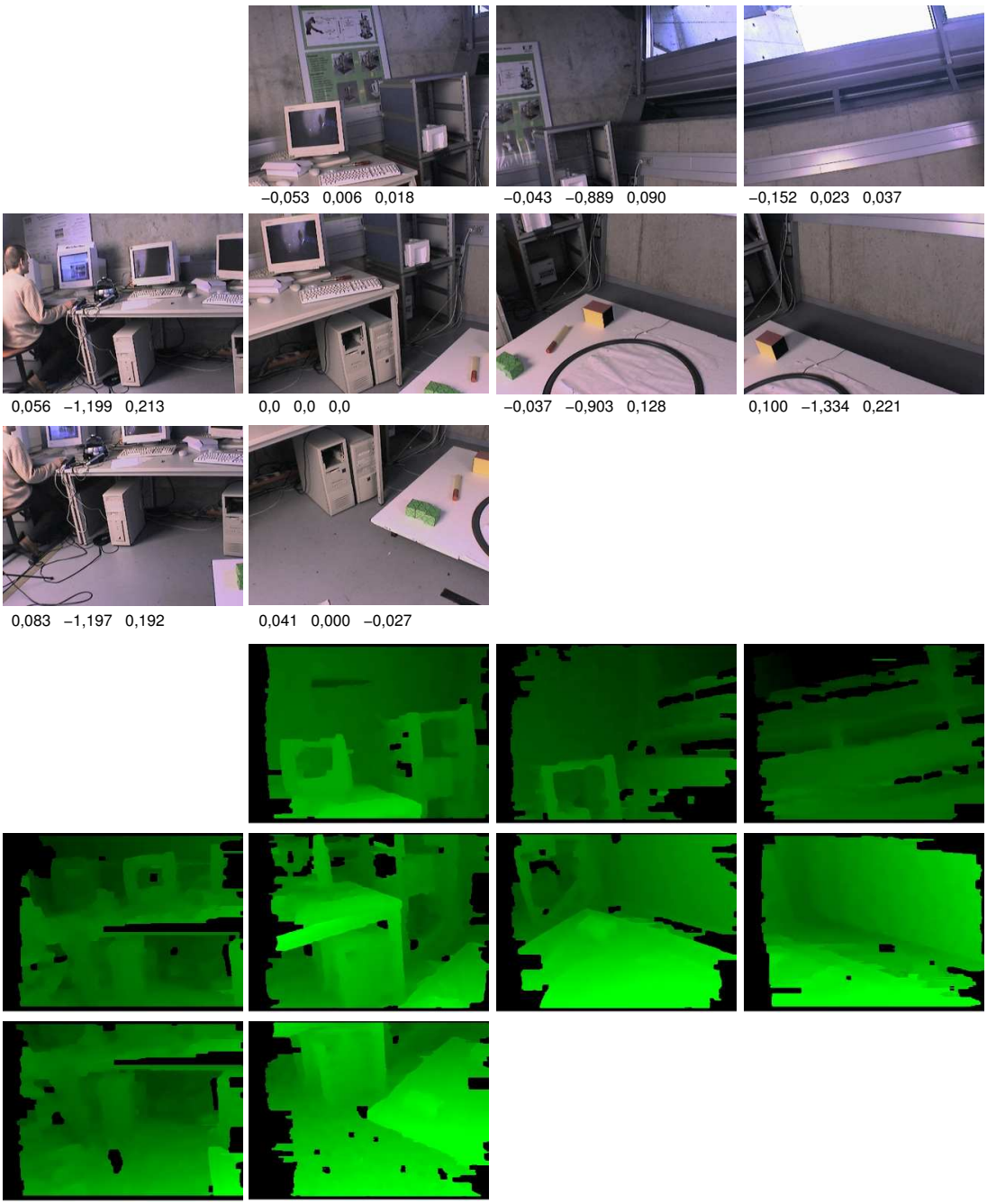


Abbildung 6.11: Neun Einzelaufnahmen (jeweils linke Aufnahme) der ursprünglich 33 Aufnahmen der Testserie 3 mit ihrer jeweiligen translatorischen Position in Metern in x, y und z und den dazugehörigen Disparitätenkarten.

Diese Sequenz besteht aus 33 Einzelaufnahmen, deren Position in sechs Freiheitsgraden aus der Bildinformation geschätzt wurde. Dabei kam die in Abschnitt 4.5.3 beschriebene Methode zum Einsatz. So wurden zunächst 2D-Bildkorrespondenzen zwischen zwei be-

nachbarten Bildern bestimmt. Mit Hilfe der bereits berechneten Tiefenkarte im Referenzbild können diese in 3D-2D-Korrespondenzen umgewandelt werden. Unter Verwendung der bekannten Projektionsmatrix der Kamera kann nun die Lage der Kamera bei der Aufnahme des neuen Bildes bestimmt werden.

Die große Anzahl an Aufnahmen ist notwendig, um eine starke Überlappung der Einzelbilder für die Positionsschätzung gewährleisten zu können. Trotzdem kommt es zu deutlichen Registrierungsfehlern, da die Position teilweise über eine Kette von Relativregistrierungen über bis zu acht Bildern gewonnen wird. Um zu vermeiden, dass sich durch diese Fehler und durch die starke Überlappung der Bilder der Modellaufbau verkompliziert, wurden nur neun Aufnahmen zur Modellierung verwendet. So ist eine geringere Überlappung der Bilder sichergestellt. Dies geschieht jedoch ausschließlich auf Grund der eingesetzten Registrierung die im Rahmen dieser Arbeit nur ersatzweise verwendet wurde, da in einem Telepräsenzsystem normalerweise davon ausgegangen werden kann, dass die Position des Telemanipulators bekannt ist. Abbildung 6.11 zeigt die neun Aufnahmen in räumlich konsistenter Anordnung zueinander. Die translatorische Kameraposition zum Zeitpunkt der Aufnahme und die Disparitätenkarte sind ebenfalls dargestellt.

Die Parameter für die Berechnung der Disparitätenkarten sind wie im oben beschriebenen ersten Beispiel gewählt. Abbildung 6.11 zeigt unten die resultierenden Disparitätenkarten.

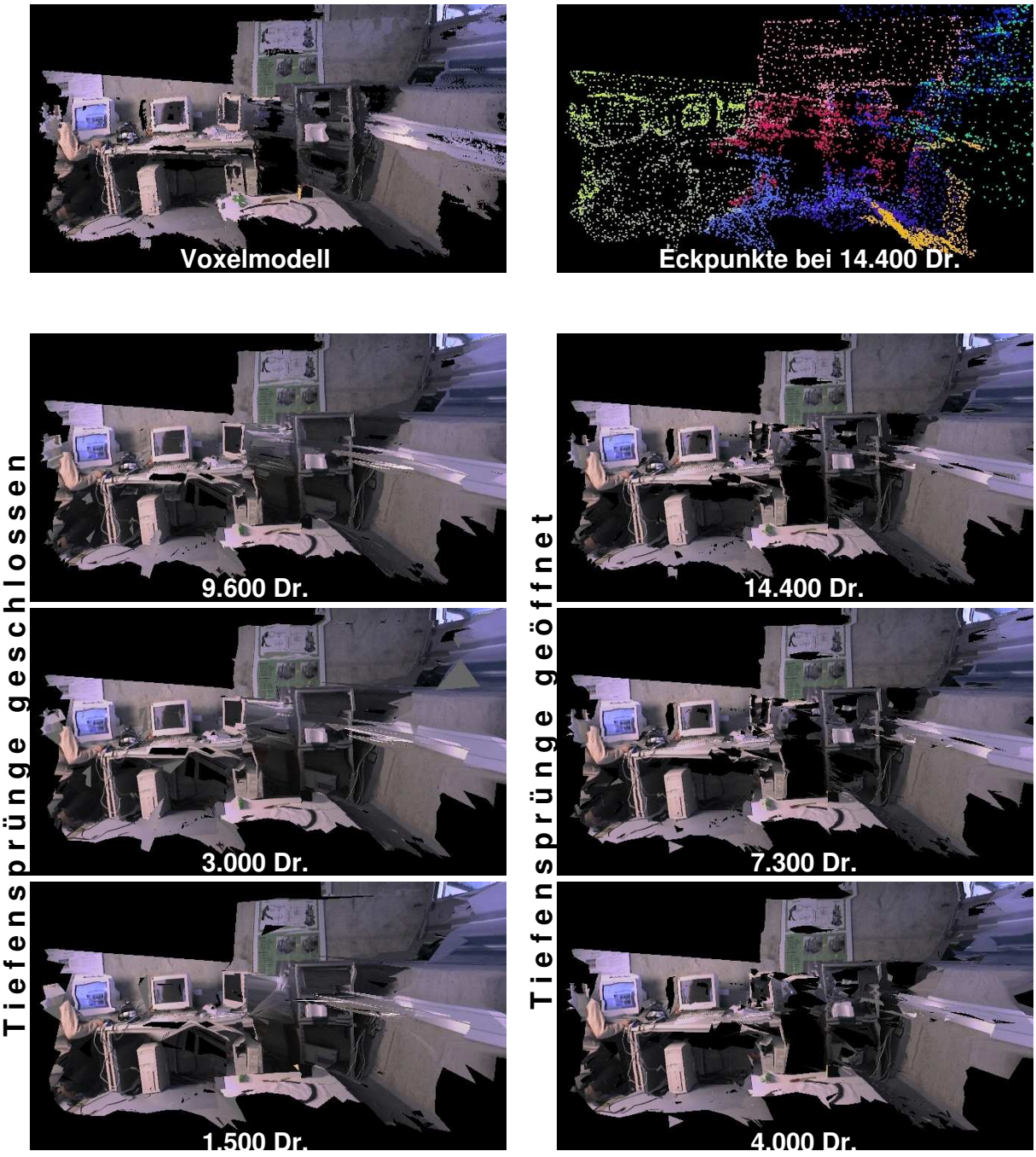


Abbildung 6.12: Ergebnisse der Testserie 3 dargestellt aus Kameraposition A: Die obere Reihe zeigt die Voxelansicht und Eckpunkte eines Netzes mit 14.400 Dreiecken. Die linke Spalte zeigt Ansichten bei geschlossenen Tiefensprüngen, die rechte Spalte bei geöffneten Tiefensprüngen. Dabei sind jeweils drei Dezimierungsstufen dargestellt.

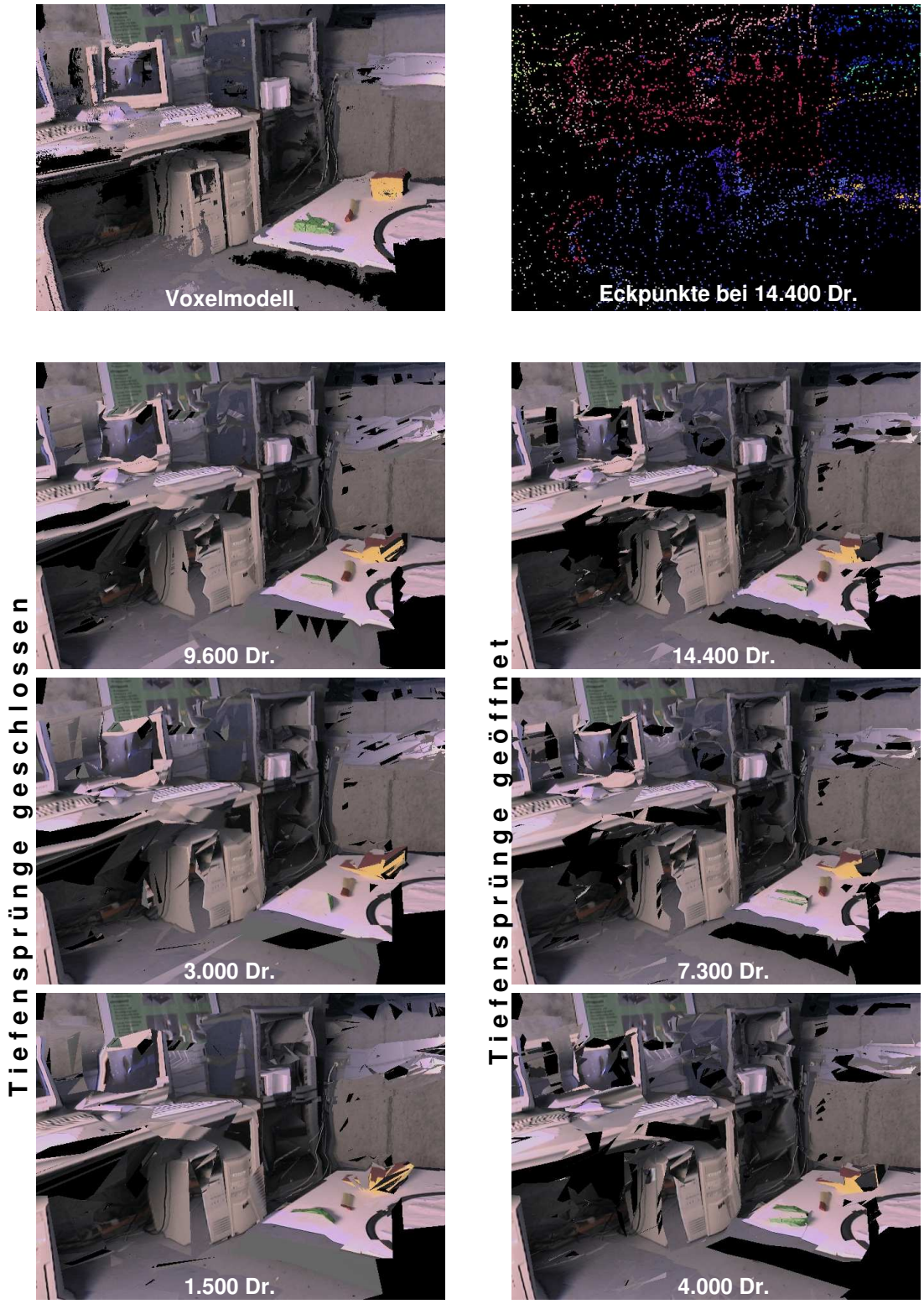


Abbildung 6.13: Ergebnisse der Testserie 3 dargestellt aus Kameraposition B: Die obere Reihe zeigt die Voxelansicht und Eckpunkte eines Netzes mit 14.400 Dreiecken. Die linke Spalte zeigt Ansichten bei geschlossenen Tiefensprüngen, die rechte Spalte bei geöffneten Tiefensprüngen. Dabei sind jeweils drei Dezimierungsstufen dargestellt.

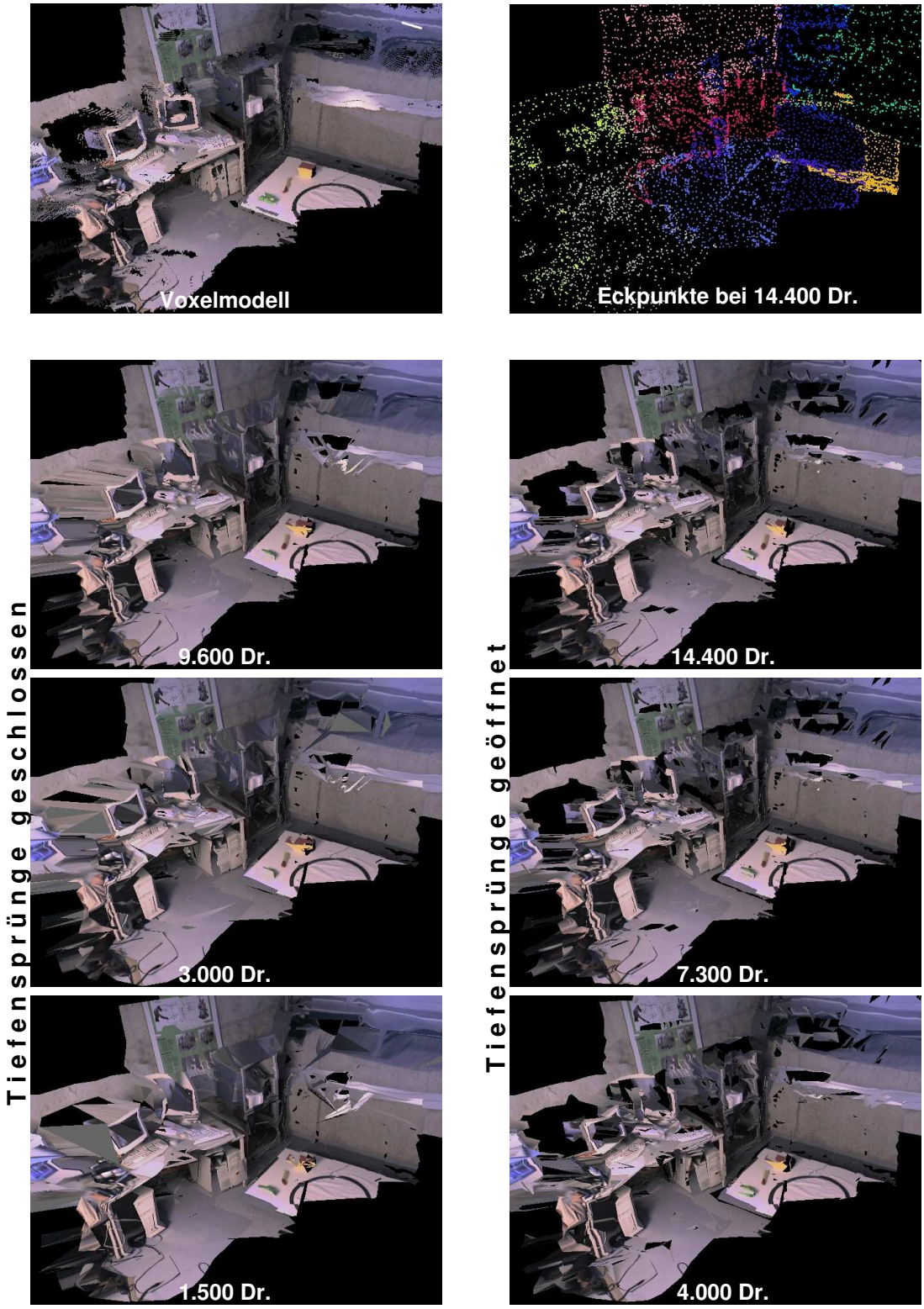


Abbildung 6.14: Ergebnisse der Testserie 3 dargestellt aus Kameraposition C: Die obere Reihe zeigt die Voxelansicht und Eckpunkte eines Netzes mit 14.400 Dreiecken. Die linke Spalte zeigt Ansichten bei geschlossenen Tiefensprüngen, die rechte Spalte bei geöffneten Tiefensprüngen. Dabei sind jeweils drei Dezimierungsstufen dargestellt.

6.3 Bewertung der Ergebnisse

Wie bereits in der Einleitung dargestellt, gibt es drei Kriterien, an denen sich diese Arbeit messen lassen muss: Die Modellierungsgeschwindigkeit, die Qualität der Modellierung und die Darstellungsgeschwindigkeit.

6.3.1 Modellierungsgeschwindigkeit

Die Modellierungsgeschwindigkeit ist mit ca. 25s pro Ansicht auf einem aktuellen PC ausreichend schnell, um eine Szene in vertretbarer Zeit zu erfassen, wenn die Szene durch koordinierte Kamerabewegung abgetastet wird. Dies entspricht in der Praxis einer Modellierungsphase, die vor der eigentlichen Telepräsenzaufgabe stattfinden muss. Der komplexere Fall einer Telepräsenzaufgabe, in der der Bediener sich in unbekannter Umgebung telepräsent zurechtfinden muss, um zum Beispiel einen mobilen Teleoperator zu steuern, kann nur mit Einschränkungen durchgeführt werden. Dynamische Szeneninhalte (des Hintergrunds) sind zwar möglich und können vom Bediener mit Verzögerung erkannt werden, können aber nicht fotorealistisch wiedergegeben werden.

Verbesserungen: Die komplexesten Operationen sind der Aufbau der Disparitätenkarte, die Triangulierung und die Dezimierung des Netzes. Die Berechnung der Disparitäten lässt sich zeilenweise unabhängig durchführen und kann damit durch Parallelisierung auf einfache Weise beschleunigt werden. Dies ist im Falle der Netzoperationen nicht ohne weiteres möglich. Hier ist eine Beschleunigung durch konzeptionelle Änderungen erreichbar: Das eingesetzte Triangulierungsverfahren erzeugt entlang von Tiefensprüngen grundsätzlich eine sehr feine Triangulierung. Würden zu Beginn der Triangulierung einzelne Eckpunkte explizit entlang der Tiefensprünge gesetzt, könnte dies die erforderliche Rechenzeit für die Triangulierung deutlich reduzieren. Ansätze eines derartigen hybriden Triangulierungsverfahrens wurden bereits getestet und sind in einer Veröffentlichung des Autors [100] dargestellt.

Mit den beschriebenen Veränderungen müssten auf einfachen Mehrprozessormaschinen Berechnungszeiten von etwa 7s pro Ansicht mit geringem Aufwand machbar sein. Natürlich steht jederzeit auch der Weg offen, ein einfacheres Verfahren zur Disparitätenberechnung einzusetzen und damit die Rechenzeit drastisch zu reduzieren. Abhängig von den Eigenschaften der Szene (Texturierung, Beleuchtung) ist damit eine geringe bis deutliche Verschlechterung der Bildqualität verbunden.

6.3.2 Qualität der Modellierung

Bei translatorischer Kamerabewegung führt die nur ungenau bekannte Kameraposition (Testserie 3) zu zusätzlichen Störungen in der Rekonstruktion, da die Fusion zwischen Tiefenkarte und Modell dadurch gestört wird. So ist in diesem Fall eine Rekonstruktion komplexer Szenen nur eingeschränkt möglich.

Wenn die Kamera sich auf einer Schwenk-Neige-Plattform befindet und ihre Position somit präzise bekannt ist, sind die texturierten Szenenmodelle von guter Qualität. In einfacheren Szenen werden alle Oberflächen genau genug rekonstruiert, um fotorealistische Ansichten von sehr guter Qualität zu erzeugen. Die komplexe Testszene 2 enthält die gröbsten Fehler nur in Bereichen, in denen Anfangs gemachte Voraussetzungen wie Reflexionsfreiheit oder die Einhaltung des *ordering constraints* verletzt wurden. Ansonsten befinden sich fehlerhafte Dreiecke häufig in Bereichen, die bei der Darstellung nicht als störend empfunden werden. So entstehen falsche Dreiecke oft in schwach texturierten Regionen oder in entfernten Bereichen der Szene, wodurch die Fehler kaum wahrgenommen werden, *wenn sich der Bediener virtuell in der Nähe der Kamera aufhält*. Entfernt er sich deutlich ($>$ 1m) vom Standort der Kamera, werden derartige Fehler durch die Parallaxe immer deutlicher. Löcher im Netz finden sich oft ebenfalls in schwach texturierten Bereichen oder in ursprünglich verdeckten Bereichen. Sie werden als störend empfunden, da sie den synthetischen Charakter der Szene unterstreichen.

Im Fall translatorischer Kamerabewegung müsste eine deutliche Verbesserung mit Hilfe besserer Registrierung erreichbar sein. Trotzdem sind aus konzeptionellen Gründen keine translatorischen Bewegungen über große Entfernungen möglich, da im vorgestellten Modell keine Information über die Konfidenz und Genauigkeit eines Polygons abgelegt werden. Einen möglichen Weg zu Lösung dieser Einschränkung stellt die in Abschnitt 4.6 vorgestellte Datenstruktur dar.

Die erreichte Bildqualität muss im Kontext der eingesetzten Technik gesehen werden: Ausschließlich aufgrund der Daten aus einem einfachen Stereosystem, können fotorealistische Bilder der Szene erzeugt werden. Dies gilt sogar auch, wenn sich die Betrachterposition deutlich von der Aufnahmeposition unterscheidet.

6.3.3 Darstellungsgeschwindigkeit

Die Darstellungsgeschwindigkeit ist von der Anzahl der darzustellenden Polygone abhängig. Abbildung 6.15 stellt die gemessene Bildwiederholrate auf einem typischen System bei der Darstellung der Testszene *Telekiste* in Abhängigkeit von der Anzahl darzustellender Dreiecke dar.

Wird von einer für immersives Arbeiten angemessenen Bildwiederholrate von 25 Bildern/s ausgegangen, so sind Netze mit ca. 7.000 Dreiecken erforderlich. Diese Grenze wird nur in einfachen Szenen (wie Testserie 1, 10 Stereoaufnahmen) deutlich unterschritten. Testszene 2 umfasst dagegen 20 Einzelbilder und besitzt gleichzeitig eine größere Komplexität. Hier beginnt die Degeneration durch Netzdezimierung bereits bei etwa 13.000 Dreiecken. Trotzdem sind bei dieser Anzahl bereits Bildwiederholraten von ca. 15 Bildern/s möglich. So ist die erreichbare Darstellungsgeschwindigkeit generell ausreichend, um beschränkte Szenen darzustellen. Bei sehr großen Modellen muss auf Darstellungsseite eine Vorauswahl darzustellender Polygone getroffen werden, um interaktive Bildwiederholraten zu ermöglichen. Gerade der Markt der Grafikkarten entwickelt sich jedoch derzeit schnell genug um diese Grenze weiter zu erhöhen.

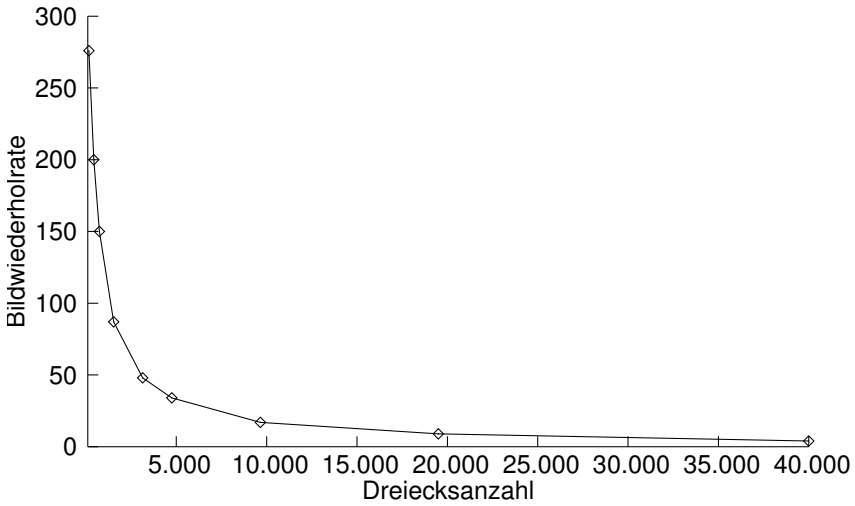


Abbildung 6.15: Bildwiederholrate in Abhängigkeit von der Anzahl darzustellender Dreiecke.

6.4 Minimalinvasive Chirurgie

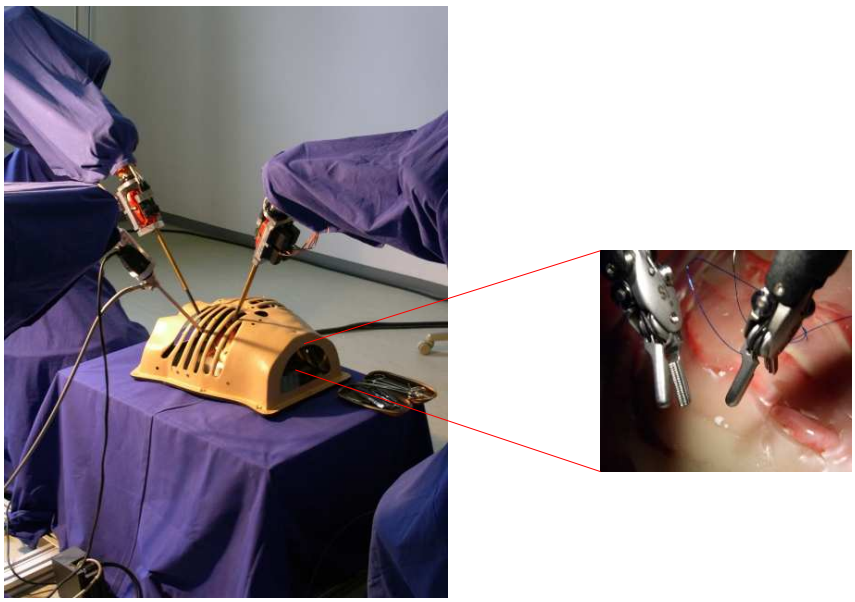


Abbildung 6.16: Demonstrationssystem für die endoskopische Herzchirurgie im SFB 453

Im Rahmen des Sonderforschungsbereichs *Wirklichkeitsnahe Telepräsenz und Teleaktion* wurde im Arbeitsbereich Medizintechnik ein Demonstrationsszenario für die endoskopische Herzchirurgie aufgebaut (siehe Abb. 6.16). In diesem Demonstrationssystem wurde das in dieser Arbeit beschriebene System integriert, um das folgende Problem der minimalinvasiven Chirurgie zu lösen:

Das Blickfeld des Arztes durch das Endoskop umfasst etwa 25x25 mm. In diesem Arbeitsbereich stehen ihm zwei in mehreren Freiheitsgraden bewegliche Instrumente zur Verfügung. Im Verlauf einer Operation kommt es bei den Instrumenten und dem Endoskop häufig zu Indexierungsbewegungen durch den Chirurgen. Dies ist durch die Skalierung der Handbewegung des Chirurgen und den vergleichsweise geringen Arbeitsbereich der Eingabegeräte bedingt. Durch dieses 'Umgreifen' besteht die Gefahr, dass der Chirurg seine Instrumente 'verliert'. Dies geschieht insbesondere, wenn sich die Instrumente außerhalb des Sichtbereichs des Endoskops befinden. Verliert der Chirurg in diesen Momenten die intuitive Koordination zwischen Handbewegungen und wahrgenomener Szene muss er die Instrumente durch Absuchen des Arbeitsraums mit dem Endoskop finden und dann schrittweise wieder in die Nähe des Operationsgebietes bewegen.

Wird dem Chirurgen ein auf einem texturierten Umgebungsmodell basierendes Übersichtsbild des Operationsgebietes gezeigt, so kann er sich in dieser virtuellen Übersicht orientieren und die Instrumente wieder zielstrebig in das Operationsgebiet führen. Für diese Anwendung wurde das vorgestellte System angepasst.

Stereorekonstruktion für das Endoskop Betrachtet man die Aufnahme einer (künstlichen) Herzoberfläche in Abbildung 6.16, fallen zwei Störfaktoren sofort ins Auge.

- Die Szene enthält viele schwach texturierte Oberflächen.
- Die Instrumente verdecken die Sicht.

Zwei weitere Probleme fallen erst bei genauer Untersuchung auf.

- Die Bilder zeigen einen starken Helligkeitsabfall am Bildrand.
- Die Bilder weisen deutliche radiale Verzerrung bedingt durch die Weitwinkeloptik auf.

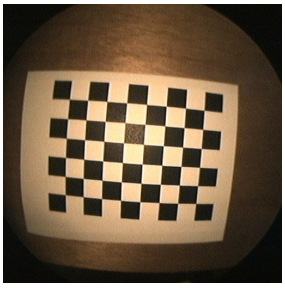


Abbildung 6.17: Radiale Verzerrung bei endoskopischen Aufnahmen

Die starke radiale Verzerrung erfordert eine sorgfältigere Modellierung des Kamerasystems, um durch Rektifizierung Zeilenkorrespondenz zu erreichen. Durch Modellierung der radialen Verzerrung erster und zweiter Ordnung (vgl. Abschnitt 3.1.2) können die Eigenschaften der Optik ausreichend genau dargestellt werden. Da die zugrundeliegende Gleichung 3.3 keine analytische Umkehrung besitzt, muss diese Umkehrung numerisch gelöst werden. Da dies zeitlich aber nur beim Start des Systems für die Erstellung der *look-up-table* (vgl. Abschnitt 3.4.8) relevant ist, hat die komplexere Berechnung keinen nachteiligen Einfluss auf die Systemgeschwindigkeit.

Abbildung 6.19 zeigt einen Satz ungefilterter Disparitätenkarten der Aufnahmen in Abbildung 6.18, die mit Dynamischer Programmierung (ZNCC, DN-Kostenfunktion) bei verschiedenen Fenstergrößen und Strafkosten berechnet wurden. Deutlich sichtbar ist, dass starke Glättung durch große Fenster und hohe Strafkosten bei einer derartigen Szene notwendig, aber durch die Struktur der Szene auch vertretbar ist.

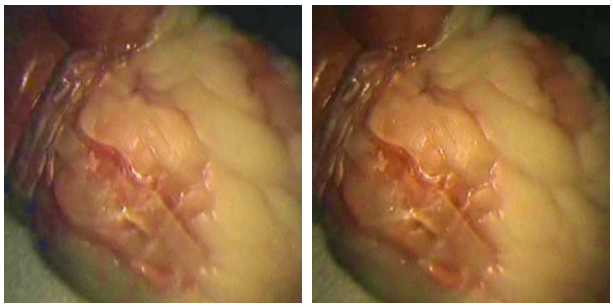


Abbildung 6.18: Rektifizierte Aufnahmen einer künstlichen Herzoberfläche.

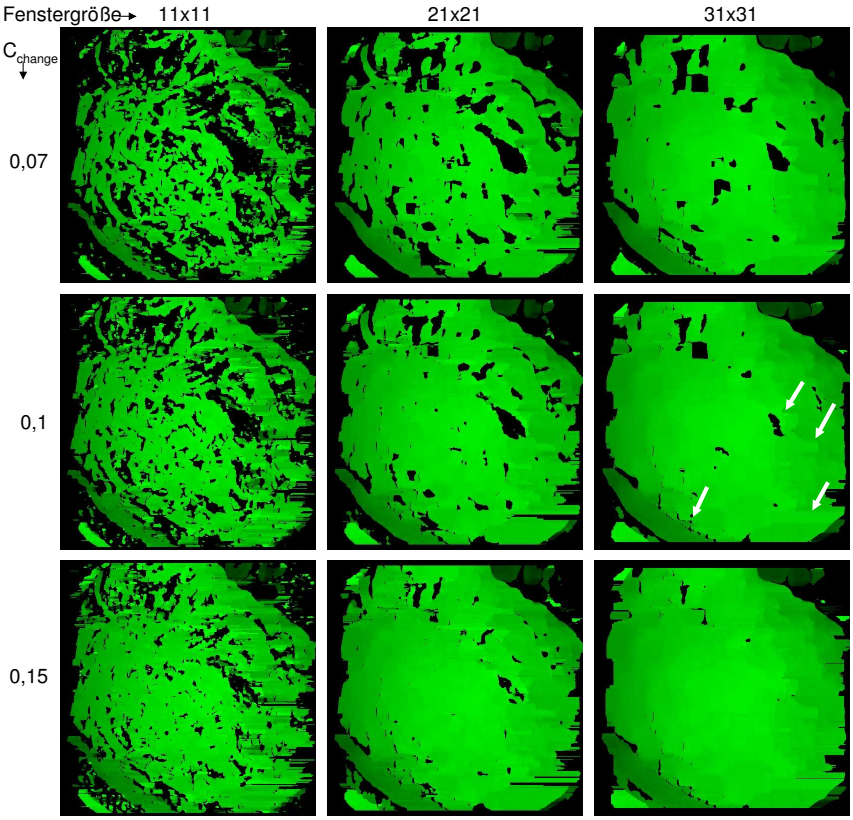


Abbildung 6.19: Disparitätenkarten bei unterschiedlicher Wahl relevanter Parameter. Einige problematische Fehler in der Disparitätenkarte sind durch Pfeile gekennzeichnet.

Leider lassen sich die Fehlkorrespondenzen, die in Abbildung 6.19 durch Pfeile gekennzeichnet sind, nicht entfernen. Sie treten in sehr texturarmen Regionen des Bildes auf und resultieren aus der unterschiedlichen Grundhelligkeit bei gleichzeitig kontinuierlichem Helligkeitsverlauf innerhalb der texturarmen Region. Die Anwendung der in Abschnitt 4.3.3 vorgestellten Filter ermöglicht eine weitere Reduktion von Fehlkorrespondenzen und Artefakten. Hier spielen insbesondere der Distanzfilter, der Ausreißerfilter und die kantenerhaltende Mittelwertfilterung eine wichtige Rolle. Abbildung 6.20 stellt die gefilterte Variante der rechten mittleren Karte in Abbildung 6.19 ($C_{change} = 0.1, 31 \cdot 31$) dar.

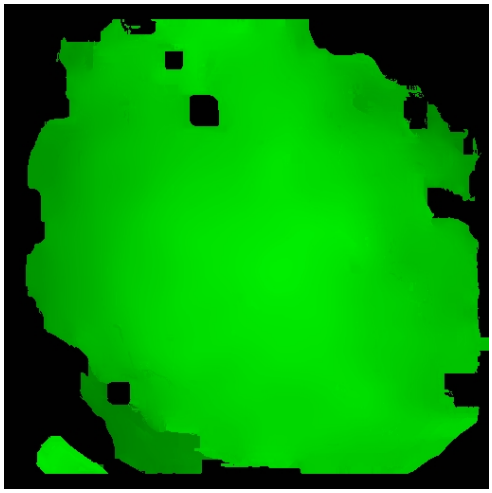


Abbildung 6.20: Gefilterte Disparitätenkarte.

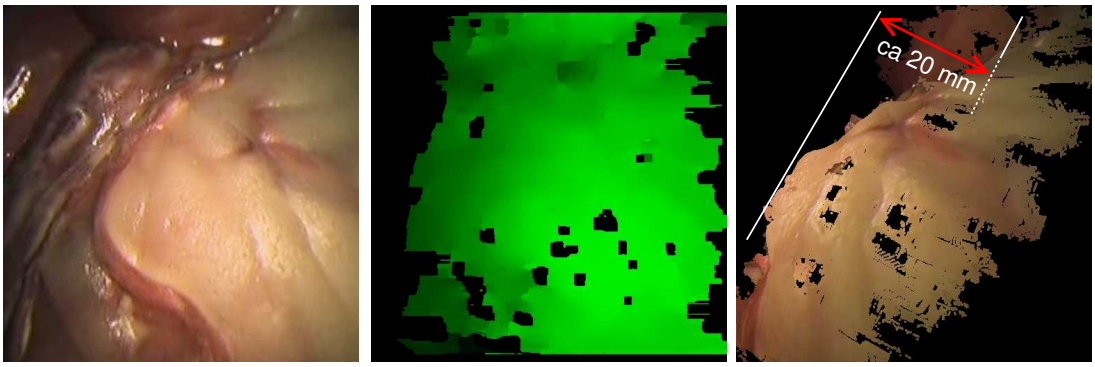


Abbildung 6.21: Nahaufnahme aus ca. 30 mm Entfernung. Disparitätenkarte (Kontrast zu Visualisierungszwecken erhöht) und Ansicht der kolorierten 3D-Punkte aus unterschiedlicher Perspektive.

Abbildung 6.21 zeigt eine Aufnahme aus einer für Operationen üblichen Distanz von 30 mm zusammen mit ihrer Disparitätenkarte und einer kolorierten Voxel-Darstellung. Die gesamte Szene umfasst nur etwa 30 mm Tiefe bei Disparitäten zwischen 35 und 65.

Zusammengesetzte Szenen Da das Endoskop durch einen Roboter geführt wird, kann die Kameraposition bestimmt werden, indem dieses System kalibriert wird. Es handelt sich dabei um eine gedankliche Erweiterung des Kalibrierszenarios, wie es in Abschnitt 3.1.3 dargestellt wurde. Statt eines Schwenk-Neige-Kopfes ist die Kamera nun auf einem robotischen System mit sechs Freiheitsgraden befestigt. Der Kalibriervorgang ändert sich dadurch nicht: Ein ortsfester Kalibrierkörper wird aus unterschiedlichen Positionen betrachtet. Aus den anfallenden Daten kann die relative Position der Kamera zum Handwurzelpunkt des Roboters bestimmt werden.

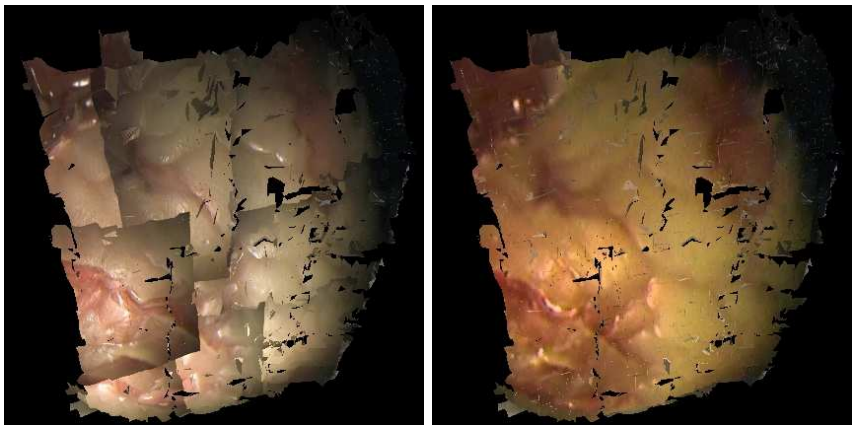


Abbildung 6.22: Texturiertes Modell aus 10 Ansichten (unzusammenhängende Einzelnetze). Deutlich ist der überaus störende Effekt des starken Helligkeitsabfalls zum Bildrand hin sichtbar. Die rechte Ansicht zeigt dasselbe Modell, bei der Verwendung nur eines einzigen Bildes zur Texturierung des Netzes.

Abbildung 6.22 zeigt ein texturiertes Modell, das aus 10 Einzelaufnahmen aus unterschiedlichen Positionen zusammengesetzt wurde. Die Einzelnetze sind hier nicht vernäht. Hier zeigt sich ein Beleuchtungsproblem bei einem endoskopischen System: Die Beleuchtung wird mit der Kamera mitgeführt. Das Konzept der Texturextraktion aus Kamerabildern basiert jedoch auf der Annahme ortsfester Beleuchtungsquellen. Wird die Lichtquelle – wie im vorliegenden Fall – bewegt, so sind deutliche Helligkeitssprünge zwischen benachbarten Aufnahmen die Folge. Aus diesem Grund wurde darauf verzichtet, die Szene aus Einzelaufnahmen zusamenzusetzen. Abbildung 6.22 zeigt rechts dasselbe Modell. In dieser Aufnahme wurde jedoch als letztes Bild eine Übersichtsaufnahme des Herzens zur Texturierung verwendet. Diese Vorgehensweise funktioniert – sie hat jedoch den Nachteil, dass das Endoskop für jede Aktualisierung in eine Übersichtsposition bewegt werden muss. Gleichzeitig nimmt die Beleuchtung bei größerem Abstand ab, wodurch die Verstärkung bei der Bilddigitalisierung erhöht werden muss. Dies resultiert in starkem Bildrauschen und schlechter Bildqualität. Bei dieser Vorgehensweise kann jedoch auch das Modell aus der Übersichtsansicht erstellt werden. Nachteilig sind die damit verbundene geringere Tiefenauflösung und das starke Bildrauschen. Vorteilhaft ist der Verzicht auf das Vernähen benachbarter Netze, wodurch die Netztopologie einfacher wird.

6.4.1 Bewertung

Die Szenenrekonstruktion ist auch auf endoskopische Bilder anwendbar. Trotz des nichtidealen Kamerasystems lassen sich Disparitätenkarten von guter Qualität berechnen. Der Aufbau eines Szenenmodells aus mehreren Aufnahmen ist ebenfalls machbar, auch wenn Ungenauigkeiten in der Kamerasteuerung hier zu Positionsfehlern einzelner Netzabschnitte führen (vgl. Sprünge in der Textur in Abb. 6.22 links), die wiederum das Vernähen von Netzen deutlich verkomplizieren. Die Texturierung des Modells aus Einzelaufnah-

men scheitert an den Beleuchtungsbedingungen, da die Beleuchtung nicht ortsfest ist. Beschränkt man das System auf nur eine Aufnahme für Modellerstellung und Texturierung, sind gute Ergebnisse erzielbar. Das Problem des eingeschränkten Gesichtsfeldes des Chirurgen ist mit dem vorliegenden System mit geringem Aufwand lösbar.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Die vorliegende Arbeit stellt ein komplettes System zur kamerabasierten Umgebungsmodellierung im Kontext der Telepräsenz vor. Alleine auf der Basis von Stereo-Bildpaaren eines kalibrierten Kamerasystems und deren bekannter Lage im Raum, wird ein polygonales Modell der Szene aufgebaut und aktualisiert.

Die Arbeit ist im Kontext der Telepräsenz angesiedelt und wurde im Rahmen des Sonderforschungsbereichs 453 „Wirklichkeitsnahe Telepräsenz und Teleaktion“ gefördert. Um Übertragungszeiten in einem Telepräsenzsystem vor dem Bediener zu verbergen, werden statt realer Kamerabilder mit Zeitverzögerung, prädizierte fotorealistische Ansichten der Szene dargestellt. So sieht der Bediener sofort, wie seine Bewegungen am entfernten Ort von einem Roboter ausgeführt werden, obwohl dies durch die Übertragungszeit tatsächlich erst mit einer Verzögerung stattfindet. Das hierzu nötige Modell der Szene umfasst den Roboter, einzelne Objekte im Vordergrund und den Szenenhintergrund.

Diese Arbeit stellt die Rekonstruktion des Szenenhintergrunds aus Kamerabildern im Detail vor. Weiterhin werden auch die Einbettung in das Gesamtsystem und ergänzende Komponenten wie die Roboterkinematik kurz dargestellt. Die Arbeit besitzt zwei Schwerpunkte: Die 3D-Rekonstruktion aus Stereo-Bildpaaren und die Erzeugung, Dezimierung und Weiterverarbeitung des polygonalen Szenenmodells.

Im ersten Schwerpunkt werden drei typische **Stereoverfahren** aus der Literatur detailliert vorgestellt und getestet. Dies ist erforderlich, da verschiedene Verfahren in der Literatur zwar oft an standardisierten Testaufnahmen gemessen und verglichen werden, diese Testbedingungen aber wenig über die Leistung der Verfahren in einem Telepräsenzszenario in Innenräumen aussagen.

Der zweite Schwerpunkt liegt in der **Netzerzeugung, Dezimierung und Weiterverarbeitung**. Hier werden zunächst verschiedene Filter dargestellt, mit denen die Disparitätenkarte einer Szene in Hinblick auf die Umwandlung in ein Dreiecksnetz verbessert werden kann. Nach der Vorstellung verschiedener Verfahren zur Netzerzeugung werden Fragen der Nachbearbeitung des Netzes behandelt. So kann es erforderlich sein, Tiefensprünge im Netz zu öffnen oder Löcher explizit zu schließen.

Aus der Verwendung des erzeugten Modells für die fotorealistische Darstellung der Szene ergeben sich spezielle Bedingungen für den Modellaufbau, die direkten Einfluss auf die eingesetzten Konzepte haben. So muss die Information, die durch ein neues Stereo-Bildpaar

gewonnen wird möglichst schnell und ohne bereits bekannte Polygone der Szene unnötig zu verändern, in das Modell eingearbeitet werden. Die vorliegende Arbeit löst dieses Problem durch den Vergleich der realen Disparitätenkarte mit einer synthetisch aus dem Modell erzeugten Disparitätenkarte und modifiziert das polygonale Modell soweit erforderlich. Das Vernähen angrenzender Teilnetze wird ebenfalls im $2\frac{1}{2}$D-Raum durchgeführt.

Ebenfalls im Bereich der Modellstruktur leistet die Arbeit einen Beitrag mit der Vorstellung einer hybriden Datenstruktur, die es erlaubt, 3D-Punkte und Polygone parallel in einem Modell zu speichern. Dies erlaubt bei Veränderungen im Modell die genaue Aussage darüber, ob Polygone durch neu eintreffende Information aktualisiert werden müssen oder noch unverändert bleiben können. Dies ist für die Wiederverwendbarkeit der Texturen entscheidend.

Das in dieser Arbeit entwickelte System erlaubt es alleine auf der Basis von kalibrierten Stereo-Kamerabildern polygonale Modelle zu erzeugen, die eine fotorealistische Darstellung der Szene auf handelsüblichen Grafikkarten aus beliebigen Perspektiven ermöglichen. Die Qualität der synthetischen Ansichten ist gemessen an dem bescheidenen Einsatz von nur einem bewegten Kamerapaar sehr gut. Allerdings gerät das System bei komplexen Szenen mit diffusen Reflexionen und texturarmen Flächen an seine Grenzen. Die Modellerzeugung benötigt mit etwa 25s pro Bild verhältnismäßig wenig Zeit. Da die aktuelle Implementierung auf einem handelsüblichen PC mit nur einem Prozessor läuft und Spielraum zur Parallelisierung bietet, kann diese Zeit noch deutlich reduziert werden. Durch das Konzept der Szenenprädiktion ist die Aktualisierungsrate von der Bildwiederholrate bei der Darstellung entkoppelt. Durch die Dezimierung des Dreiecksnetzes wird eine Netzkomplexität erreicht, die interaktive Darstellung auf handelsüblichen Grafikkarten ermöglicht.

7.2 Ausblick

Das vorgestellte System leistet bereits bei geringem Einsatz an Sensoren und Hardware Beachtliches, doch ist die Aktualisierungsrate für den Telepräsenzeinsatz nicht optimal. Wie an verschiedenen Stellen erläutert, ist eine Beschleunigung sowohl durch implementierungstechnische Maßnahmen als auch durch konzeptionelle Änderungen möglich. So sollte die Rekonstruktion durch dynamische Programmierung parallel implementiert werden, um von den Tendenzen zu Mehrprozessorsystemen in der PC-Technik zu profitieren. Da Parallelisierung im Bereich des Netzaufbaus und der Netzdezimierung zu komplexen Fragestellungen der Parallelisierbarkeit von Manipulationen in Dreiecksnetzen führt, sind hier zunächst konzeptionelle Änderungen vielversprechender. Durch eine weitergehende Analyse der Disparitätenkarten mit bildverarbeitungstechnischen Mitteln können Dreiecks-Eckpunkte entlang von Tiefensprüngen in der Szene explizit gesetzt werden. Dadurch lässt sich die Anzahl von Dreiecken zur initialen Modellierung und damit auch die Rechenzeit zur Dezimierung deutlich reduzieren.

Die Konzepte zur zeitlichen Stabilisierung des Modells sind nicht auf hochdynamische Szenen hin ausgelegt. Die in Kapitel 4.6 vorgestellte Datenstruktur zur parallelen Modellierung von Polygonen und 3D-Punkten könnte dieses Problem zwar beheben, ist aufgrund ihrer hohen Komplexität und der damit verbundenen Rechenzeiten aber nicht geeignet. So müsste eine zeitliche Stabilisierung schon auf der Ebene der 3D-Punkte stattfinden, auf der sie unkompliziert zu handhaben ist. Dies erfordert neue Konzepte zur stabilen polygonalen Modellierung dynamischer Punktewolken.

Das Folgeprojekt im Sonderforschungsbereich 453 „Virtuelle Weiträumigkeit mit einem hybriden Display“ setzt an diesen Punkten an: So soll die Problematik der großen Komplexität in der Szenenrekonstruktion mit einem neuen Ansatz untersucht werden. Die auftretenden Daten sollen auf algebraisch ausbeutbare Strukturen hin untersucht werden, um diese dann als zeitvariantes Zustandsmodell mit niedrigerer Zustandsdimension zu approximieren. Die zeitliche Stabilisierung soll dabei bereits auf dem Niveau von 3D-Punkten erfolgen.

Literatur

[1] *ABW GmbH*. www.abw-3d.de. 31

[2] Apel, Jens: *Rekonstruktion von Oberflächen aus Punktewolken/Delaunay Netzwerke*. Arbeit im Hauptseminar Grafische Datenverarbeitung, TU Ilmenau, 1997. 72, 73

[3] Baker, Henry Harlyn: *Depth from Edge and Intensity Based Stereo*. Doktorarbeit, Stanford University CA, Departement of Computer Science, 1982. 32, 33

[4] Besl, Paul J. und Neil D. McKay: *A Method for Registration of 3-D Shapes*. IEEE Transaction on Pattern Analysis and Machine Intelligence, 14, Februar 1992. 20, 21

[5] Bobick, A. und S. Intille: *Large Occlusion Stereo*. International Journal of Computervision, 33(3):181–200, 1999. 35

[6] Bundesministerium für Bildung und Forschung: *Forschungsprojekt „3D-Sensorik für vorausschauende Sicherheitssysteme im Automobil - 3D-SIAM “*. Förderkennzeichen: 16SV1334/0, 2001 – 2005. 31

[7] Burkert, Tim: *Hardware-beschleunigte Textur-Extraktion für ein fotorealistisches Prädiktives Display*. Doktorarbeit, TU München, Fakultät für Elektro- und Informationstechnik, 2005. 7, 10

[8] Burschka, Darius: *Videobasierte Umgebungsexploration am Beispiel eines binokularen Stereo-Kamerasystems*. Doktorarbeit, TU München, Fakultät für Elektro- und Informationstechnik, 1998. 8, 32

[9] Corke, Peter, Paul Dunn und Jasmine Banks: *Frame-rate Stereopsis using Non-parametric Transforms and Programmable Logic*. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, Mai 1999. 35

[10] Cox, I. J.: *Stereo without Disparity Gradient Smoothing: a Bayesian Sensor Fusion Solution*. In: *British Machine Vision Conference*, Seiten 337–346, Leeds, GB, 1992. 38

[11] Cozzi, A., B. Crespi, F. Valentinotti und F. Wörgötter: *Performance of Phase-based Algorithms for Disparity Estimation*. Journal of Machine Vision and Applications, 9:334–340, 1997. 33

[12] Debevec, Paul E. und Jitendra Malik: *Recovering High Dynamic Range Radiance Maps from Photographs*. In: Whitted, Turner (Herausgeber): *Proceedings of SIGGRAPH 97*, Seiten 369–378, Los Angeles, California, 1997. Addison Wesley. 13

[13] Delaunay, B.: *Sur la Sphère Vide*. Bulletin of Academy of Sciences of the USSR, Seiten 793–800, 1934. 72

[14] Eberst, Christof: *Incorporation of Recognition Strategies in Sensory Exploration*. Doktorarbeit, TU München, Fakultät für Elektro- und Informationstechnik, 2000. 8

[15] Eckstein, W.: *Unified Gray Morphology: The Dual Rank*. Pattern Recognition and Image Analysis, 7(1):29–37, 1997. 81

[16] Eisert, Peter, Eckehard Steinbach und Bernd Girod: *Automatic Reconstruction of 3-D Stationary Objects from Multiple Uncalibrated Camera Views*. IEEE Transactions on Circuits and Systems for Video Technology, 10(2):261–277, März 2000. 68

[17] Falkenhagen, Lutz: *Depth estimation from stereoscopic image pairs assuming piecewise continuous surfaces*. In: *Proc. of European Workshop on combined Real and Synthetic Image Processing for Broadcast and Video Production*, Hamburg, Deutschland, 1994. 36, 38, 40, 51, 52, 53, 55

[18] Falkenhagen, Lutz: *Hierarchical Block-Based Disparity Estimation Considering Neighbourhood Constraints*. In: *Proceedings of the International Workshop on SNHC and 3D Imaging, Rhodes, Greece*, September 1997. 36, 40

[19] Faugeras, Olivier: *Three-Dimensional Computer Vision*. MIT Press, 1993. 28, 32, 36

[20] Floriani, L. De, S. Bussi und P. Magillo: *Intelligent Systems and Robotics*, Kapitel Triangle-Based Surface Models, Seiten 340–373. Gordon and Breach Science Publishers, 2000. 72, 73

[21] Floriani, L. De, B. Falcidieno und C. Pienovi: *A Delaunay-Based Representation of Surfaces Defined over Arbitrarily-shaped Domains*. Computer Vision, Graphics and Image Processing, 32:127–140, 1985. 85

[22] Floriani, Leila De, Bianca Falcidieno, George Nagy und Caterina Pienovi: *A Hierarchical Structure for Surface Approximation*. Computer and Graphics, 8(2):183–193, 1984. 72

[23] Floriani, Leila De, Bianca Falcidieno und Caterina Pienovi: *A Delaunay-Based Method for Surface Approximation*. In: *Eurographics '83*, Band 24, Seiten 333–350. Elsevier Science, 1983. 85

[24] Freeman, H. und L. S. Davis: *A Corner Finding Algorithm for Chain Code Curves*. IEEE Trans. on Computers, (26):297–303, 1977. 32

[25] Fröhlinghaus, T. und J. Buhmann: *Real-time Phase-based Stereo for a Mobile Robot*. In: *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996. 33

[26] Fröhlinghaus, T. und J.M. Buhmann: *Regularizing Phase-Based Stereo*. In: *Proceedings of the International conference on Pattern Recognition*, Seiten 451–455, Wien, AU, 1996. 33

[27] Fusiello, A., V. Roberto und E. Trucco: *Efficient Stereo with Multiple Windowing*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seiten 858–863. IEEE Computer Society Press, Juni 1997. 58

[28] Fusiello, Andrea, Emanuele Trucco und Alessandro Verri: *Rectification with unconstrained stereo geometry*. In: Clark, A. F. (Herausgeber): *Proceedings of the British Machine Vision Conference*, Seiten 400–409, September 1997. 25, 27

[29] Garland, Michael und Paul S. Heckbert: *Fast Polygonal Approximation of Terrains and Height Fields*. Technischer Bericht CMU-CS-95-181, September 1995. 85, 87

[30] Grimson, W. E. L.: *Computational Experiments with a Feature Based Stereo Algorithm*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 7:17–34, 1985. 35

[31] Gross, M., S. Würmlin, M. Naef, E.Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. V. Gool, S. Lang, K. Strehlke und A. V. Moereand O. Staadt: *blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence*. In: *SIGGRAPH*. ACM, 2003. 8

[32] Guibas, Leonidas und Jorge Stolfi: *Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams*. ACM Transactions on Graphics, 4(2):74–123, 1985. 72

[33] Hamann, B.: *A Data Reduction Scheme for Triangulated Surfaces*. Computer Aided Geometric Design, 11:197–214, 1994. 74

[34] Haralick, R. und L. Shapiro: *Computer and Robot Vision*. Addison-Wesley Publishing Company, 1992. 81

[35] Harris, C. und M. Stephens: *A Combined Corner and Edge Detector*. In: *Proc of the Alvey Vision Conference*, Seiten 147–151, Univ. Manchester, 1988. 32

[36] Heckbert, Paul S. und Michael Garland: *Survey of Polygonal Surface Simplification Algorithms*. SIGGRAPH 1997, Multiresolution Surface Modeling Course Notes, 1997. 74, 84, 85, 105, 107

[37] Hirschmüller, Heiko: *Stereo Vision Based Mapping and Immediate Virtual Walkthroughs*. Doktorarbeit, De Montfort University, Leicester, UK, Juni 2003. 8, 35

[38] Horn, B.: *Robot Vision*. MIT Press, 1986. 32

[39] Horn, B. und M. Brooks (Herausgeber): *Shape from Shading*. MIT Press, 1989. 32

[40] Horn, B. K. P.: *Closed-form Solution of Absolute Orientation using Unit Quaternions*. Journal of the Optical Society of America, 1987. 21

[41] Ingleby, C.M.: *The Stereoscope Considered in Relation to the Philosophy of Binocular Vision*. London: Walton, 1853. 30

[42] Jakl, Edward A.: *Why CMOS Image Sensors are Poised to Surpass*. In: *Proceedings of the International Integrated Circuits'99*, 1999. 12

[43] Jones, JP und LA Palmer: *An Evaluation of the Two-dimensional Gabor Filter Model of Simple Receptive Fields in the Cat Striate Cortex*. Journal of Neurophysiology, 58(6):1233–1258, Dezember 1987. 33

[44] Jost, Timothee: *Fast Geometric Matching for Shape Registration*. Doktorarbeit, Université de Neuchatel, 2002. 21, 22

[45] Kanade, T. und M. Okutomi: *A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(9):920–932, September 1994. 35

[46] Kanade, Takeo und CBS Broadcasting Inc.: *Eye Vision*, 2001. www.ri.cmu.edu/events/sb35/tksuperbowl.html. 10

[47] Kang, S.B., J. Webb, C.L. Zitnick und T. Kanade: *A Multibaseline Stereo System with Active Illumination and Real-Time Image Acquisition*. In: *ICCV95*, Seiten 88–93, 1995. 31, 32

[48] Krsek, Pavel: *The Trimmed Iterative Closest Point Algorithm*. Technischer Bericht, Center for Applied Cybernetics, 2002. 21

[49] Kumar, S.: *Surface Triangulation: A Survey*. Technischer Bericht, Department of Computer Science, University of North Carolina, Januar 1996. 71

[50] Lane, R.A. und N.A. Thacker: *Stereo Vision Research: An Algorithm Survey*. 32, 34

[51] Lang, S., M. Naef, M. Gross und L. Hovestadt: *IN:SHOP – Using Telepresence and Immersive VR for a New Shopping Experience*. In: Ertl, T., B. Girod, H. Niemann, H.-P. Seidel, E. Steinbach und R. Westermann (Herausgeber): *Vision, Modeling, and Visualization 2003*, Seiten 3–10, Technische Universität München, November 2003. 8

[52] Lawson, Charles L.: *Transforming Triangulations*. Discrete Mathematics, 3:365–372, 1972. 72

[53] Lee, D.T. und B.J. Schachter: *Two Algorithms for Constructing a Delaunay Triangulation*. International Journal of Computer and Information Sciences, 9(3):219–242, 1980. 72

[54] Lee, J.: *A Drop Heuristic Conversion Method for Extracting Irregular Networks from Digital Elevation Models*. In: *Proceedings GIS/LIS'89*, Seiten 30–39, Orlando, FL, USA, 1989. 83, 90

[55] Lewis, B.A. und J.S. Robinson.: *Triangulation of Planar Regions with Applications*. The Computer Journal, 21(4):324–332, 1978. 72

[56] Lindstrom, P. und G. Turk: *Fast and Memory Efficient Polygonal Simplification*. In: *Proceedings of IEEE Visualization '98*, Seiten 279–286. IEEE, Oktober 1998. 87, 88

[57] Lorensen, W. und H. Cline: *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*. Computer Graphics, 21(4):163–169, 1987. 71

[58] Marr, D. und T. Poggio: *Cooperative Computation of Stereo Disparity*. Science, 194:283–287, 1976. 36, 39

[59] Marr, D. und T. Poggio: *A Computational Theory of Human Stereo Vision*. Proceedings of the Royal Society of London B, 204:301–328, 1979. 36

[60] McCullagh, Michael J. und Charles G. Ross: *Delaunay Triangulation of a Random Data Set for Isarithmic Mapping*. The Cartographic Journal, 17(2):93–99, 1980. 72

[61] McDonnell, M. J.: *Box-Filtering Techniques*. Computer Graphics and Image Processing, 17:65–70, 1981. 66

[62] Medioni, G. und Y. Yasumoto: *Corner Detection and Curve Representation Using Cubic B-Splines*. Computer Vision, Graphics and Image Processing, (39):267–278, 1987. 32

[63] Mirante, A. und N. Weingarten: *The Radial Sweep Algorithm for Constructing Triangulated Irregular Networks*. IEEE Computer Graphics and Applications, 2(3):11–21, 1982. 72

[64] Moravec, H.P.: *Towards Automatic Visual Obstacle Avoidance*. In: *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Seite 584, MIT, Cambridge, Mass. USA, 1977. 32

[65] Moravec, H. P.: *Visual Mapping by a Robot Rover*. In: *Proc. of the 6th International Joint Conference on Artificial Intelligence*, Seiten 598–600, 1979. 32

[66] Mulligan, J., N. Kelshikar, X. Zabulis und K. Daniilidis: *Stereo-based Environment Scanning for Immersive Tele-presence*. IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Immersive Telepresence, 14(3):304–320, 2004. 7, 68

[67] Ozeki, O., T. Nakano und S. Yamamoto: *Real-Time Range Measurement Device for Three-Dimensional Object Recognition*. PAMI, 8(4):550–554, Juli 1986. 31

[68] Pajdla, Tomas und Luc J. Van Gool: *Matching of 3-D Curves using Semi-differential Invariants*. In: *5th International Conference on Computer Vision*. T.Pajdla, L.V.Gool, 1995. 22

[69] Petrie, G. und T.J.M Kennie (Herausgeber): *Terrain Modelling in Surveying and Civil Engineering*. Whittles Publishing in association with Thomas Telford Ltd, 1990. 72

[70] Pollefeys, M., R. Koch, M. Vergauwen und L. Van Gool: *Automated reconstruction of 3D scenes from sequences of images*. ISPRS Journal of Photogrammetry and Remote Sensing, 55(4):251–267, 2000. 9, 68

[71] Prazdny, K.: *Detection of Binocular Disparities*. Biological Cybernetics, 52:93–99, 1985. 35, 37

[72] Roy, S.: *Stereo without Epipolar Lines: A Maximum Flow Formulation*. International Journal of Computer Vision, 1(2), 1999. 38

[73] Roy, Sébastien und Ingemar J. Cox: *A Maximum-Flow Formulation of the N-Camera Stereo Correspondence Problem*. In: *Proceedings of the International Conference of Computer Vision 1998*, Seiten 492–502, 1998. 38

[74] Sanger, T.D.: *Stereo Disparity Computations Using Gabor Filter*. Biol. Cybern., 59:405–418, 1988. 33

[75] Scarlatos, Lori L. und Theo Pavlidis: *Optimizing Triangulations by Curvature Equalization*. In: *Proc. Visualization '92*, Seiten 333–339. IEEE Comput. Soc. Press, 1992. 75

[76] Scharstein, Daniel und Richard Szeliski: *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*. Technischer Bericht, Microsoft Research Technical Report MSR-TR-2001-81, 2001. 34, 38

[77] Scharstein, Daniel und Richard Szeliski: *High-Accuracy Stereo Depth Maps Using Structured Light*. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, Seiten 195–202, Madison, WI, USA, Juni 2003. 31

[78] Scharstein, Daniel und Richard Szeliski: *Middlebury College, Stereo Vision Research Page*. cat.middlebury.edu/stereo/data.html, 2005. 41

[79] Schroeder, William J., Jonathan A. Zarge und William E. Lorensen: *Decimation of Triangle Meshes*. SIGGRAPH Comput. Graph., 26(2):65–70, 1992. 107

[80] Sequeira, Vitor, K. Ng, E. Wolfart, J.G.M. Gonçalves und D. Hogg: *Automated Reconstruction of 3D Models from Real Environments*. ISPRS J. of Photogrammetry and Remote Sensing, 54(1):1–22, Februar 1999. 8

[81] Sonderforschungsbereich 453: *Wirklichkeitsnahe Telepräsenz und Teleaktion*. Finanzierungsantrag 1999-2001, 1998. 2

[82] Soucy, M. und D. Laurendeau: *Multi-Resolution Surface Modeling Based on Hierarchical Triangulation.* In: *Proceedings of the CVGIP: Image Understanding*, 1993. 104

[83] Steinbach, E., P. Eisert und B. Girod: *Motion-Based Analysis and Segmentation of Image Sequences using 3-D Scene Models.* Signal Processing, Special Issue: Video Sequence Segmentation for Content-based Processing and Manipulation, 66(2), 1998. 9

[84] Sun, Changming: *A Fast Stereo Matching Method.* In: *Digital Image Computing: Techniques and Applications*, Seiten 95–100, Massey University, Auckland, New Zealand, Dezember 1997. 30, 40, 66

[85] Trucco, E. und A. Verri: *Introductory Techniques for 3-D Computer Vision.* Prentice-Hall, 1998. 29

[86] Veksler, Olga: *Stereo Matching by Compact Windows via Minimum Ratio Cycle.* In: *International Conference on Computer Vision*, Seiten 540–547, 2001. 35

[87] Voronoi, M. G.: *Nouvelles Applications des Paramètres Continus à la Théorie des Formes Quadratiques.* Journal für Reine und Angewandte Mathematik, 134:198–287, 1908. 72

[88] Watson, D.F.: *Computing the N-dimensional Delaunay Tessellation with Application to Voronoi Polytopes.* The Computer Journal, 24(2):167–172, 1981. 72

[89] Zabih, Ramin und John Woodfill: *Non-parametric Local Transforms for Computing Visual Correspondence.* In: *European Conference on Computer Vision*, Seiten 151–158, 1994. 35

[90] Zitnick, C. Lawrence und Takeo Kanade: *A Cooperative Algorithm for Stereo Matching and Occlusion Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(7):675–684, 2000. 36, 39, 40, 59

Eigene Veröffentlichungen und Diplomarbeiten zum Thema

[91] Blasczyk, Martin: *Bildverarbeitung mit Pentium-III SIMD Befehlen.* TU München, Lehrstuhl für Realzeit-Computersysteme, Studienarbeit, Dezember 2000.

[92] Burkert, Tim, Jan Leupold und Georg Passig: *Hardware Accelerated Texture Extraction for a Photo-Realistic Predictive Display.* In: Ertl, T., B. Girod, H. Niemann, H.-P. Seidel, E. Steinbach und R. Westermann (Herausgeber): *Vision, Modeling, and Visualization 2003*, Seiten 11–18, Technische Universität München, November 2003.

[93] Diemer, Robert: *Entwicklung einer DSP-Firmware für Bildverarbeitung und Kameralokalisation auf Firewire-Basis.* Diplomarbeit, TU München, Lehrstuhl für Realzeit-Computersysteme, Februar 2002.

[94] Fiala, Daniel: *Monokulares Kamera-System mit Prismen-Optik zur Erstellung von Tiefenkarten.* Diplomarbeit, TU München, Lehrstuhl für Realzeit-Computersysteme, Juli 2003.

[95] Gröll, Klaus: *Triangulierung von Bildsequenzen.* Diplomarbeit, TU München, Lehrstuhl für Realzeit-Computersysteme, Oktober 2001.

[96] Hauck, Alexa, Michael Sorg, Georg Passig Thomas Schenk und Georg Färber: *On the Performance of a Biologically Motivated Visual Control Strategie for Robotic Hand-Eye Coordination.* In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'00)*, Oktober 2000.

[97] Passig, G., T. Burkert und J. Leupold: *Scene Model Acquisition from Stereo Vision for Photo-Realistic Scene Prediction.* In: Knoll, A. (Herausgeber): *Third IEEE International Conference on Humanoid Robots, Workshop on Telepresence*, Technische Universität München, Oktober 2003.

[98] Passig, Georg und Tim Burkert: *A Photo-Realistic Predictive Display for Telepresence Applications.* In: Färber, Georg und Jens Hoogen (Herausgeber): *Advances in Interactive Multimodal Telepresence Systems*, Technische Universität München, März 2001. DFG.

[99] Passig, Georg und Tim Burkert: *Ein Prädiktives Display für Telepräsenzanwendungen.* In: Dillmann, R. (Herausgeber): *Human Centered Robotic Systems*, Band 1, Seiten 75–82, 2002.

[100] Passig, Georg, Tim Burkert und Jan Leupold: *A Photo-Realistic Predictive Display.* Presence, Seiten 22–43, 2004. 106, 132

[101] Passig, Georg, Tim Burkert und Jan Leupold: *Scene Model Acquisition for a Photo-realistic Predictive Display and its Application to Endoscopic Surgery.* In: *MIRAGE 2005: International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, Seiten 89–97. INRIA Rocquencourt, France, März 2005.

[102] Rauschert, Ingmar: *Window-based Stereo-correlation in a Teleoperation Task.* Mastersthesis, TU München, Lehrstuhl für Realzeit-Computersysteme, Juni 2001.

[103] Smoravek, Martin: *Telepräsente Steuerung eines antropomorphen redundanten Manipulators.* Diplomarbeit, TU München, Lehrstuhl für Realzeit-Computersysteme, Januar 2004.