Dreidimensionale Umgebungsmodellierung durch monokulare Exploration mit einem mobilen Roboter

Stefan Feyrer, Oliver Schimmel, Andreas Zell

Wilhelm-Schickard-Institut für Informatik, Abt. Rechnerarchitektur
Universität Tübingen
Köstlinstraße 6, D-72074 Tübingen
{feyrer, schimmel, zell}@informatik.uni-tuebingen.de

Der vorliegende Beitrag stellt ein System vor, das basierend auf der Analyse von monokularen Bildsequenzen, die im Laufe von Explorationsfahrten mit einem mobilen Roboter aufgenommen werden, eine Rekonstruktion der dreidimensionalen Umgebung vornimmt. Zur Berechnung der Tiefendaten wird die über Odometrie- und Pan-Tilt-Daten gewonnene Bewegungsinformation der Kamera herangezogen. Das Verfahren arbeitet punktorientiert, d.h. es werden mit Hilfe eines Merkmalsdetektors einzelne markante Bildpunkte extrahiert und über die Bildsequenz hinweg getrackt. Flächeninformation wird über eine spezielle Triangulierungsmethode gewonnen, die in den Bildern vorhandene Objektkanten mit einbezieht und somit eine realistische Modellierung ermöglicht. Zur Visualisierung werden wahlweise grauwertcodierte Tiefenkarten oder texturierte 3D-Grafiken im VRML-Format generiert.

1 Einleitung

Eine wichtige Fähigkeit von mobilen Robotern ist die Erfassung der räumlichen Struktur der Umgebung, in der sie sich bewegen. Zwar lassen sich Navigationsaufgaben unter entsprechenden Umgebungsvoraussetzungen sehr gut mittels herkömmlicher Abstandssensorik (Infrarot, Ultraschall, Laser) bewältigen, Basis für das Verrichten komplexerer Aufgaben wie beispielsweise das flexible Greifen, Manipulieren und Plazieren von Objekten ist jedoch die Kenntnis der geometrischen Beschaffenheit der Szene. Daneben eröffnen sich durch die Möglichkeit einer automatischen Generierung einer dreidimensionalen geometrischen Repräsentation der Umgebung eine Reihe von Anwendungen auf dem Gebiet der virtuellen Realität, da auf eine aufwendige manuelle Szenenmodellierung verzichtet werden kann.

In der Literatur wurde eine Reihe von Ansätzen beschrieben, die sich in den Problemkreis der automatisierten Rekonstruktion von 3D-Szenen aus Bildfolgen einordnen lassen. Eines der ersten Systeme in diesem Bereich stammt von Hans Moravec [6]. Er benutzte eine ferngesteuerte mobile Plattform, auf der eine orthogonal zur Blickrichtung translatierbare Kamera montiert war. Tiefeninformationen wurden aus Bildern berechnet, die bei temporär stationärem Roboter an unterschiedlichen Kamerapositionen aufgenommen wurden. Das FINALE-System von Kosaka und Kak [4] versucht, einfache geometrische Merkmale, wie z.B. Ecken von Quadern zu erkennen. Durch vorge-

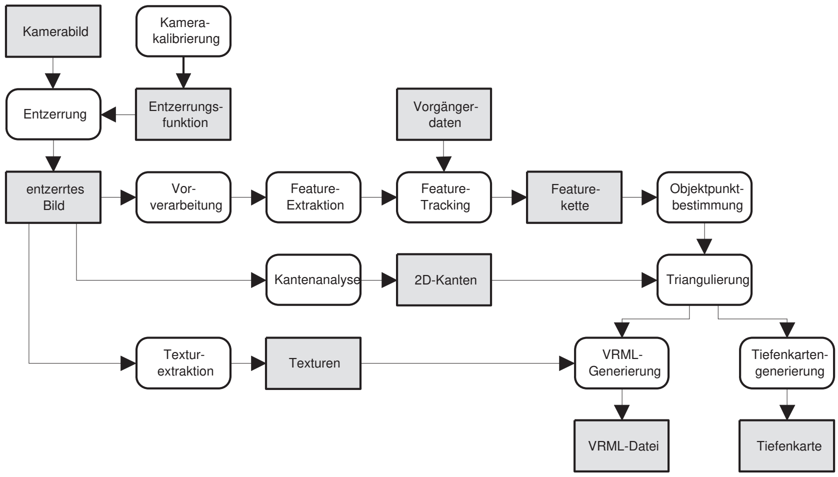

Abbildung 1. Struktureller Aufbau des Gesamtsystems

gebenes modellbasiertes Wissen werden diese Eckpunkte zu geometrischen Objekten verknüpft. Shah und Aggarwal [9] verfolgen einen ähnlichen Ansatz. Hier werden jedoch lediglich Linien entlang der drei Raumrichtungen ausgewertet, vertikale Linien werden dabei als Kanten von Wänden, Türen oder Fenstern interpretiert. Poelman und Kanade [7] sowie Pollefeys et al. [8] erzeugen texturierte dreidimensionale Modelle aus zuvor gefilmten Szenen. Dabei ist kein Wissen über die Kameraposition während der Bildaufnahme notwendig. Vielmehr wird diese aus den Bildern selbst berechnet. Während in [8] ein epipolar-geometrischer Ansatz verwendet wird, kommt in [7] ein algebraischer Ansatz mittels Eigenvektorbestimmung der in Matrizenform verwalteten Merkmalspunkte zum Einsatz. Auch das DROID-System von Harris [3] extrahiert Bewegungsinformation aus den Bilddaten. Featurepunkte werden darin unter Verwendung eines Kalman-Filters getrackt und durch einen Triangulierungsalgorithmus zu einem dreidimensionalen Raummodell zusammengefügt.

Das hier vorgestellte System ist speziell für den Einsatz auf einem mobilen Roboter konzipiert. Es verwendet dessen Odometrie- und Pan-Tilt-Daten, um daraus die für die Tiefenberechnung nötige Bewegungsinformation zu berechnen. Im Gegensatz zu [4] und [9] wird jedoch – abgesehen von der Beschränkung auf den Einsatz im Indoor-Bereich – kein modellbasiertes Wissen über die Umgebung verwendet.

Die Analyse einer Bildsequenz gliedert sich in sechs Hauptkomponenten. Erster Schritt ist die Vorverarbeitung, die eine Bildentzerrung, eine Kontrastverstärkung und eine Rauschminderung beinhaltet. Anschließend wird für jedes Bild eine Detektion von markanten Punkten vorgenommen. Die detektierten Features werden über die Bildsequenz hinweg getrackt und zu sogenannten Featureketten verknüpft. Über die Berechnung von Strahlengängen werden dreidimensionale Raumpositionen ermittelt. Flächeninformation wird über eine Triangulierungsmethode gewonnen, welche einzelne Raumpositionen unter Berücksichtigung von Objektkanten verbindet. Letzter Schritt ist die Generierung von grauwertcodierten Tiefenkarten bzw. texturierten 3D-Grafiken im VRML-Format. Abb. 1 gibt eine Übersicht über den Verarbeitungsablauf.

2 Vorverarbeitung

Um eine präzise 3D-Rekonstruktion einer Szene zu erzielen, müssen hohe Ansprüche an die Genauigkeit der Ausgangsdaten gestellt werden. Dazu gehört zum einen eine gute Bildvorverarbeitung, die Rauschen und Kontrastschwächen weitestgehend eliminiert, zum anderen eine genaue Kamerakalibrierung, um durch Linsensysteme hervorgerufene Abbildungsverzerrungen ausgleichen zu können.

2.1 Kamerakalibrierung

Die Kalibrierung der verwendeten CCD-Farbkamera erfolgt automatisiert mit Hilfe einer planaren Kalibrierungsvorlage mit quadratischem Gitter. Der Kalibrierung liegt ein Kameramodell mit perspektivischer Projektion und radialer Linsenverzerrung zugrunde, bei dem die Verzeichnugskorrektur durch ein nichtlineares Gleichungssystem beschrieben werden kann und wie folgt iterativ zu berechnen ist:

$$\begin{aligned} x_{ui} &= \frac{x_v}{1 + \kappa_1 r_{i-1}^2 + \kappa_2 r_{i-1}^4 + \ldots + \kappa_n r_{i-1}^{2n}} \\ y_{ui} &= \frac{y_v}{1 + \kappa_1 r_{i-1}^2 + \kappa_2 r_{i-1}^4 + \ldots + \kappa_n r_{i-1}^{2n}} \end{aligned} \quad \text{mit} \quad r_i = \begin{cases} \sqrt{x_{ui}^2 + y_{ui}^2} & \text{für} \quad i > 0 \\ \sqrt{x_v^2 + y_v^2} & \text{für} \quad i = 0 \end{cases} \tag{1}$$

Dabei sind die verzerrten Bildkoordinaten mit (x_v, y_v), die entzerrten mit (x_u, y_u) bezeichnet. Die Transformation von Bildspeicherkoordinaten in Koordinaten der Kalibrierungsebene (x_b, y_b) ist durch eine Bildhauptpunktverschiebung (c_x, c_y), eine Skalierung s und eine Bildrotation um den Winkel α modelliert:

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = R(\alpha) \cdot \begin{bmatrix} s \cdot x_b + c_x \\ y_b + c_y \end{bmatrix} \tag{2}$$

Während die Entzerrungkoeffizienten $\kappa_1, \ldots, \kappa_n$ und der Skalierungsfaktor s unter Verwendung eines Gradientenabstiegsverfahrens gemäß dem Kalibrierungsalgorithmus von Tsai [10] bestimmt werden, erfolgt die Berechnung der rotatorischen Korrektur und der Bildhauptpunktverschiebung nach jedem Kalibrierungsschritt durch Mittelwertbildung der entsprechenden Abweichungen über alle Kalibrierungspunkte. Bei der Wahl der Anzahl der Entzerrungskoeffizienten hat sich ein Wert von $n = 6$ als optimal herausgestellt; für $n = 8$ ergaben sich keine weiteren Verbesserungen. Abb. 2 zeigt exemplarisch das Ergebnis eines Kalibrierungsvorgangs.

2.2 Rauschunterdrückung

Mit dem speziell im Hinblick auf die Erfordernisse der sich im nächsten Schritt anschließenden Featureextraktion entwickelten Rauschfilter gelingt es, Rauschen in Farbbildern zu mindern, gleichzeitig aber die Kontrasteigenschaften markanter Punkte nicht zu beeinträchtigen. Dieses Verhalten wird dadurch erzielt, daß der verwendete Faltungskern der Größe $n \times n$ in einen inneren $k \times k$-Kern und einen äußeren Ringbereich

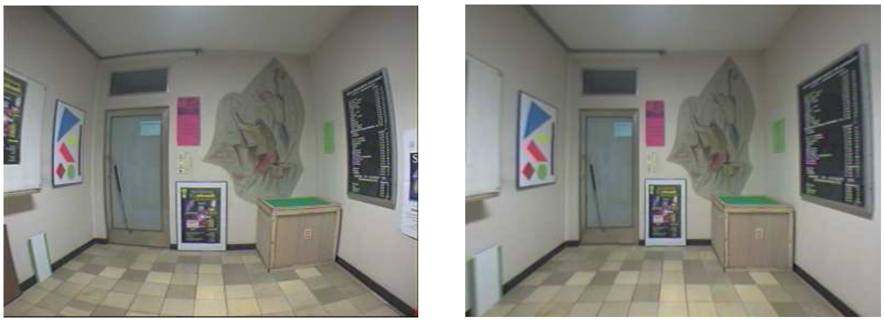

Abbildung 2. Entzerrung eines mit einer 3.8-mm-Linse aufgenommenen Bildes

unterteilt wird. Für jeden Farbkanal *i* werden innerhalb des Rings der Maximalfarbwert $p_{\max,\, i}$ und der Minimalfarbwert $p_{\min,\, i}$ bestimmt, und es wird abhängig davon eine Korrektur des momentan betrachteten Bildpunktes *P* bezüglich jedes Farbkanals vorgenommen:

$$P'_i = \begin{cases} p_{\max,\, i} & \text{für} \quad P_i > p_{\max,\, i} \\ p_{\min,\, i} & \text{für} \quad P_i < p_{\min,\, i} \\ P_i & \text{sonst} \end{cases} \tag{3}$$

Diese Vorgehensweise bewirkt, daß lediglich Strukturelemente, die nicht größer als der innere Kern sind, eliminiert werden. Das Filter ist somit linienerhaltend, ohne dabei kontrastschwächend zu wirken.

2.3 Kontrastverstärkung

Zur Kontrastverstärkung kommen häufig Histogammausgleichsverfahren zum Einsatz. Diese wirken jedoch global auf die gesamte Bildfläche und können somit lokale kontrastschwache Bereiche nur bedingt anheben. Ferner bleiben bei derartigen Verfahren Farbkontraste unberücksichtigt. Um diese unerwünschten Eigenschaften zu umgehen, wurde ein Kontrastverstärkungsfilter entworfen, das gleichermaßen für Grauwert- und für Farbbilder anwendbar ist und zudem kontrastarme Bereiche ohne Einfluß auf die restlichen Bildregionen besonders stark anhebt. Für einen Bildpunkt $P(x, y)$ wird dazu zunächst in dem umgebenden $n \times n$ -Fenster das Maximum über alle Farbkanäle *i* der auf das Intervall [0, 1] normierten Farbwerte bestimmt:

$$p_{\max} = \max_{i = 1, \ldots, N} \left(\max_{-\frac{n}{2} < r,\, s < \frac{n}{2}} (P_i(x + r, y + s)) \right) \tag{4}$$

Dieses Maximum wird für die Bestimmung der neuen Farbwerte $P'_i(x, y)$ herangezogen:

$$P'_i(x, y) = P_i(x, y) \cdot \left(\frac{k}{p_{\max}} + (1 - k) \right) \tag{5}$$

Mit dem Parameter $k \in [0, 1]$ kann der Grad der Kontrastanhebung gewählt werden. Das Ergebnis für unterschiedliche Verstärkungsgrade ist in Abb. 3 dargestellt.

Abbildung 3. Kontrastverbesserung mit unterschiedlichen Verstärkungsgraden.
Links: Originalbild, Mitte: Ergebnis für $k = 0.5$, rechts: Ergebnis für $k = 1.0$

3 Extraktion und Tracking markanter Punkte

Um die Korrelation von Bildpunkten nicht flächig durchführen zu müssen, beschränken wir uns auf die Betrachtung einzelner markanter Punkte. Diese müssen zunächst detektiert und anschließend über die Bildsequenz hinweg verfolgt werden.

3.1 Extraktion von Features

Ziel dieses Verarbeitungsschritts ist es, Bildpunkte zu detektieren, die in gewisser Weise markant sind, d.h. sich entweder durch Kontrast oder Farbe von ihrer Umgebung besonders absetzen oder Anfangs- und Endpunkte von Linien bzw. Eckpunkte von Flächen darstellen. Von solchen Punkten kann angenommen werden, daß sie auch in Folgebildern von Bildsequenzen mit hoher Wahrscheinlichkeit wieder erkannt werden können, was für ein erfolgreiches Verfolgen der Punktpositionen eine notwendige Voraussetzung darstellt.

Der zur Featureextraktion verwendete Operator basiert auf dem Merkmalsdetektor von Moravec [5], welcher für jeden Bildpunkt zunächst innerhalb des Operatorfensters die Summen von Grauwertdifferenzen direkt nebeneinanderliegender Bildpunkte in den 4 Hauptrichtungen (horizontal, vertikal und zweimal diagonal) berechnet und anschließend darüber das Minimum bildet. Eine Maximumsuche über das so entstehende Ergebnisbild liefert die Positionen markanter Punkte.

In seiner ursprünglichen Form besitzt der Moravec-Operator einige Unzulänglichkeiten. So arbeitet er beispielsweise lediglich auf Grauwertbildern, was die Erkennung kontrastreicher Farbübergänge gleicher Helligkeit ausschließt. Unser erweiterter Operator verwendet deshalb anstelle der Grauwertdifferenzen eine Abstandsnorm bezüglich der Farbwerte der jeweiligen Pixel im verwendeten Farbraum (in unserem Fall im RGB-Raum). Die Wahl der Abstandsnorm ist nicht beliebig, da sich in ungünstigen Fällen eine unterschiedliche quantitative Bewertung verschiedener Farbübergänge ergeben kann. Umgangen wird dieses Problem durch die Verwendung der Maximum-Norm, die nur die stärkste Differenz aller Farbkanäle berücksichtigt und somit keine nominellen Unterschiede zwischen Grund- und Mischfarben sowie schwarz und weiß existieren.

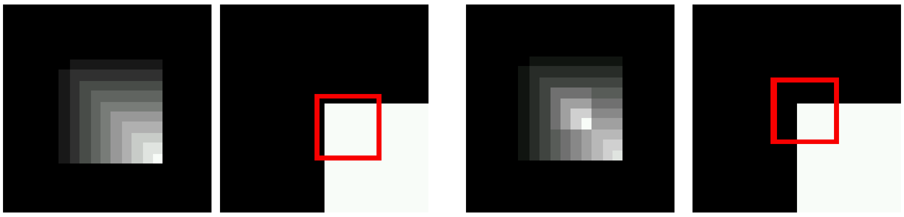

Abbildung 4. Operatorergebnis und resultierende Merkmalsposition, links ohne, rechts mit gewichtetem Operatorkern

Eine weitere unerwünschte Eigenschaft des Moravec-Operators hängt mit der diskreten Natur digitalisierter Bilder zusammen. Linien, die nicht direkt horizontal oder vertikal verlaufen, bilden treppenartige Artefakte aus, die der Operator, der nur die 4 Hauptrichtungen berücksichtigt, oftmals als Feature erkennt. Da der Operator aus Effizienzgründen nicht um alle theoretisch möglichen Richtungen erweitert werden kann, wurde ein Rotationstest eingeführt, der nur auf die bereits detektierten Featurepositionen angewendet wird. Dazu wird die graphische Umgebung um ein Feature in 15°-Schritten gedreht. In jedem dieser Rotationszustände wird erneut der Moravec-Test mit Farberweiterung durchgeführt. Anhand der sich anschließenden Minimumbildung über die Ergebnisse alle Rotationsschritte kann entschieden werden, ob es sich tatsächlich um einen Merkmalspunkt oder nur um eine treppenartige Kante handelt.

Je größer das verwendete Operatorfenster für den Moravec-Detektor gewählt wird, desto ungenauer wird die Übereinstimmung zwischen detektierten und tatsächlichen Objektpunkten. An Eckpunkten beispielsweise sind die erkannten Features Richtung Objektmitte nach innen versetzt, was eine genaue sichtstrahlbasierte Rekonstruktion erschwert. Durch einen zusätzlichen, stärker gewichteten inneren Kern des Operators kann diesem Phänomen entgegengewirkt werden (siehe Abb. 4). Tests haben ergeben, daß sich gute Ergebnisse erzielen lassen, wenn der innere Kern doppelt gewichtet und als dessen Seitenlänge die Hälfte des äußeren Kerns gewählt wird.

Abb. 5 stellt die Ergebnisse des ursprünglichen und des erweiterten Operators für eine reale Szene gegenüber.

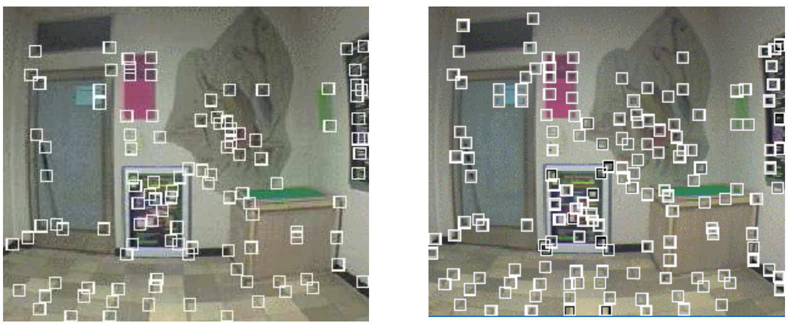

Abbildung 5. Ergebnisse des Moravec-Operators (links) und des erweiterten Operators (rechts)

3.2 Tracking von Features

Bei monokularen Bildaufnahmen ist es nur dann möglich, Tiefeninformation zu gewinnen, wenn die Feature-Information mehrerer aufeinanderfolgender und aus unterschiedlichen Perspektiven aufgenommener Bilder logisch miteinander verknüpft wird und dadurch sogenannte Featureketten erzeugt werden. Ziel der Featuretracking-Komponente ist es, die im aktuellen Bild detektierten Merkmalspunkte den aus der vorangehenden Bearbeitung der Bildsequenz bereits existierenden Featureketten zuzuordnen. Dies erfolgt zum einen über eine Positionskorrelation, zum anderen über die Korrelation der Grauwertstruktur in der lokalen Umgebung des Merkmalspunktes.

Der Positionskorrelation liegt ein Bewegungsmodell zugrunde, das die Geschwindigkeitswerte aller zur Featurekette gehörenden Punkte in negativ-exponentieller Abhängigkeit von deren Alter einbezieht und daraus eine Schätzung für die Position zum nächsten Zeitpunkt errechnet. Die euklidische Distanz zwischen geschätzter und tatsächlicher Position wird einer Bewertungsfunktion zugeführt, die schließlich den Positionskorrelationswert liefert. Über einen Gewichtungsparameter kann das Einflußverhältnis zwischen Positions- und Strukturkorrelation eingestellt werden.

Im Entscheidungsprozeß bezüglich der Zuordnung von Merkmalspunkten zu Featureketten treten einige Spezialfälle auf, die gesondert berücksichtigt werden müssen. Wünschenswert ist der Fall einer Fortsetzung der Featurekette. Diese wird nur dann vorgenommen, wenn eine bidirektionale Korrelationsübereinstimmung festgestellt werden kann. Ist dies nicht der Fall, gibt es mehrere Möglichkeiten. Wird für eine existierende Featurekette nur eine unidirektionale oder gar keine Zuordnung zu einem Featurepunkt gefunden, entsteht ein sogenannter *Ghost*. Ursachen der Entstehung von Ghosts sind häufig Verdeckungen durch im Vordergrund liegende Objekte oder Rauschen. Ghosts werden gemäß dem zuvor erwähnten Bewegungsmodell von Features über eine bestimmte Anzahl von Bildern weitergeführt. Ist diese Anzahl überschritten, wird die als Ghost geführte Featurekette endgültig beendet. Neue Featureketten entstehen, wenn Featurepunkte nur unidirektional einer Featurekette zugeordnet werden können oder gar keine passende Featurekette gefunden wird. Abb. 6 gibt einen Überblick über den Entscheidungsablauf bei der Ketten-Feature-Zuordnung.

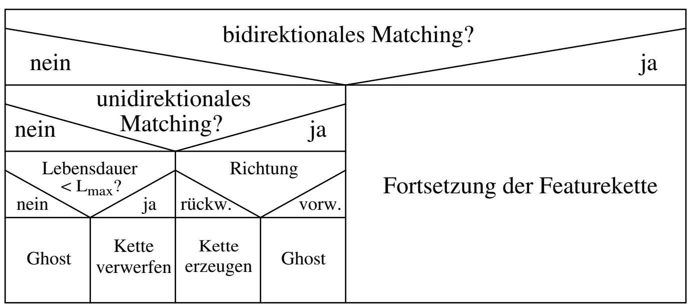

Abbildung 6. Ablauf der Kategorisierung detektierter Featurepunkte

4 Generierung des Szenenmodells

Zur Rekonstruktion von Raumpositionen einzelner Punkte wurde eine Kombination aus geometrischem und stochastischem Ansatz gewählt. Verknüpft werden die so ermittelten Objektpunkte mit Hilfe eines Triangulierungsalgorithmus, der sich an realen Objektkanten orientiert.

4.1 Bestimmung von Objektpunktpositionen

Für jede Featurekette werden anhand der Featureposition, der Kameraposition- und Richtung sowie der Position und Ausrichtung des Roboters die Sichtstrahlen rekonstruiert. In den meisten Fällen liegen die Geraden paarweise windschief, so daß die Schnittpunkte durch die Mittelpunkte der kürzesten Verbindung je zweier Sichtstrahlen angenähert werden müssen. Als Wahrscheinlichkeitsmaß für die Korrektheit einer solchen Annäherung dient eine Funktion, die zum einen vom minimalen Abstand d der beiden Strahlen abhängt und zudem deren Winkel α in Richtung der gemeinsamen Normalen berücksichtigt:

$$\upsilon = \upsilon_d \cdot \upsilon_\alpha \text{, wobei } \upsilon_d = \frac{1}{1 + \beta d} \text{ und } \upsilon_\alpha = \sin^2\alpha \tag{6}$$

β ist dabei ein konstanter Skalierungsfaktor. Die endgültige dreidimensionale Position P eines Objektpunkts wird über eine gewichtete geometrische Mittelwertsbildung über alle errechneten paarweisen Schnittpunktnäherungen S_{ij} bestimmt:

$$P = \frac{\sum_{i,j} \upsilon_{ij} S_{ij}}{\sum_{i,j} \upsilon_{ij}} \tag{7}$$

4.2 Kantenanalyse und Triangulierung

Für eine korrekte Szenendarstellung reicht es nicht aus, die Tiefeninformation einzelner Bildpunkte zu kennen. Erst durch eine Kanten- und Flächenanalyse können räumliche Zusammenhänge zwischen Punkten hergestellt und somit ein dreidimensionales Modell erstellt werden. In diesem System kommt ein Linientest zur Anwendung, der auf der Grundlage des Bresenham-Algorithmus [1] auf einem Binärbild operiert, das durch Schwellwertfilterung eines Gradientenbildes hervorgeht. Das Gradientenbild wird aus den drei Farbkanälen R, G und B des Originalbildes generiert, indem für jeden Kanal der Canny-Kantendetektor [2] angewandt und anschließend eine Maximumbildung durchgeführt wird. Der Linientest dient dazu, die Konnektivität aller Objektpunktkombinationen zu prüfen. Spezialfälle wie beispielsweise dicht beieinanderliegende parallele Kanten oder unterbrochene Kanten werden mit Hilfe von Sonderbehandlungsroutinen berücksichtigt.

Die extrahierten Verbindungen zwischen Featurepunkten bilden die Basis für die Flipping-Option des sich anschließenden Triangulierungsalgorithmus. Ausgehend von einer Delaunay-Triangulierung der ermittelten Merkmalspunkte wird versucht, eine neue Triangulierung zu finden, die mit den extrahierten Kanten zur Deckung kommt und somit ein realistisches Szenenmodell liefert. Dazu wird ein Schnittest zwischen Objektkanten und Dreieckskanten durchgeführt. Wird ein Schnittpunkt gefunden, entfernt der Algorithmus die entsprechende Dreieckskante und ersetzt sie durch die Gegendiagonale des entstehenden Vierecks. Der Flipping-Vorgang wird so lange wiederholt, bis keine weiteren Schnittpunkte gefunden werden.

4.3 Visualisierung

Zur Visualisierung ist zum einen die Darstellung über transparente, grauwertcodierte Tiefenkarten implementiert, zum anderen besteht die Möglichkeit, die 3D-Modelle im standardisierten Grafikformat VRML 2.0 abzulegen. Die für eine realistische Darstellung nötige Texturinformation wird vom Programm aus den Originalbildern extrahiert.

5 Ergebnisse und Ausblick

Das beschriebene System kommt auf mobilen Robotern vom Typ RWI B21 zum Einsatz und wurde zunächst für künstliche, aus farblich markanten geometrischen Objekten zusammengesetzte Szenen getestet. Bei diesen Versuchen konnten qualitativ sehr gute Ergebnisse erzielt werden. In den Bildern sichtbare Eckpunkte von Objekten wurden ausnahmslos korrekt detektiert und stabil getrackt. Lediglich das verfahrensimmanente Problem der perspektivischen Verdeckung von Objekten und Objektkanten führte an den betreffenden Merkmalspunkten zu fehlerhaften Tiefenwerten. Real-World-Szenen beinhalten naturgemäß weniger stark ausgeprägte Merkmalspunkte, die nicht immer eindeutig identifiziert werden können. Versuche in Real-World-Umgebungen haben gezeigt, daß abgesehen von einzelnen Ausreißerpunkten dennoch auch hier gute Resultate erzielt werden können. Die Ergebnisse für eine künstliche Szene sind in Abb. 7, verschiedene generierte Ansichten einer Real-World-Szene in Abb. 8 dargestellt. Die fehlenden Randbereiche in den Bildern der VRML-Modelle sind darauf zurückzuführen, daß nur innerhalb der konvexen Hülle der Featurepunkte eine Modellierung erfolgen kann.

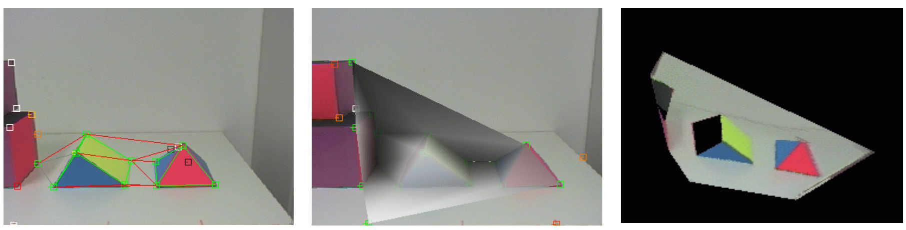

Abbildung 7. Ergebnisse für eine künstliche Szene. Von links nach rechts: Triangulierung, grauwertcodierte Tiefenkarte und VRML-Modell aus anderer Perspektive

Abbildung 8. Drei verschiedene Ansichten einer Real-World-Szene

Bisher arbeitet die 3D-Modellierungskomponente lediglich bezüglich eines einzelnen zugrunde liegenden Bildes. Dadurch entstehen Modelle von Teilszenen, die in einem zusätzlichen Integrationsschritt zu einem Gesamtszenenmodell verschmolzen werden könnten. Eine dahingehende Erweiterung ist ebenso geplant wie die Erhöhung der Verarbeitungsgeschwindigkeit durch eine Parallelisierung des Gesamtablaufs und Geschwindigkeitsoptimierungen innerhalb der einzelnen Systemkomponenten.

Literatur

1. J. E. Bresenham: *Algorithm for Computer Control of a Digital Plotter*, IBM Systems Journal, Bd. 4, Nr. 1, S. 25-30, 1965
2. J. Canny: *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Bd. 8, Nr. 6, S. 679-698, 1986
3. C. Harris: *Geometry from Visual Motion*, in A. Blake, A. Yuille: Active Vision, MIT Press, 1992
4. A. Kosaka, A. C. Kak: *Fast Vision-Guided Mobile Robot Navigation Using Model-Based Reasoning and Prediction of Uncertainties*, Computer Vision, Graphics and Image Processing – Image Understanding, Bd. 56, Nr. 3, S. 271-329, 1992
5. H. P. Moravec: *Towards Automatic Visual Obstacle Avoidance*, Proceedings of the 5th International Joint Conference on Artificial Intelligence, Vision-I, S. 584, 1977
6. H. P. Moravec: *The Stanford Cart and the CMU Rover*, Proceedings of the IEEE, Bd. 71, Nr. 7, S. 872-884, 1983
7. C. J. Poelman, T. Kanade: *A Paraperspective Factorization Method for Shape and Motion Recovery*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Bd. 19, Nr. 3, S. 206-218, 1997
8. M. Pollefeys, R. Koch, M. Vergauwen, L. van Gool: *Flexible 3D Acquisition with a Monocular Camera*, Proceedings of the IEEE International Conference on Robotics and Automation, S. 2771-2776, 1998
9. S. Shah, J. K. Aggarwal: *Mobile Robot Navigation and Scene Modeling Using Stereo Fish-Eye Lens System*, Machine Vision and Applications, Bd. 10, Nr. 4, S. 159-173, 1997
10. R. Y. Tsai: *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-The-Shelf TV Cameras and Lenses*, IEEE Transactions on Robotics and Automation, Bd. 3, Nr. 4, S. 323-344, 1987