

Verfahren zur hierarchischen Stereobildverarbeitung zur Umgebungsmodellierung unterstützt durch Bewegungsinformationen

Method for Hierarchical Stereo Vision for Spatial Environment Modelling Supported by Motion Information

T. Ihme, Mannheim, **R. Bade**, Magdeburg

Kurzfassung

Der Einsatz mobiler Roboter, die sich autonom in einer fremden Umgebung bewegen können, ist für eine Vielzahl von Anwendungsgebieten interessant. Diese Arbeit beschäftigt sich mit der Adaption eines Stereobildverfahrens, um dieses in einer Echtzeitumgebung, wie sie auf einem Roboter gegeben ist, anwenden zu können. Neben dem eigentlichen Verfahren werden Ansätze zur Effizienz- und Qualitätssteigerung durch die Verwendung redundanter Informationen diskutiert.

Abstract

The use of mobile robots that move autonomously in an unknown environment is interesting for a number of applications. This paper deals with the adaptation of a stereo vision method to use it in a real time environment like present on robots. Besides the actual method, approaches for increasing efficiency and quality by using redundant information are discussed.

1 Einleitung

Der Bedarf an mobilen Robotern für automatischen Transport und Ausführung von Diensten fördert das Interesse an radgetriebenen und laufenden Robotern. Die Einsatzgebiete zur Nutzung mobiler Roboter reichen von Transport, Inspektion und Teleoperation bis zur autonomen Exploration und Operation. Mobile Roboter müssen in einer unbekanntem Umwelt agieren. Dazu müssen sie Informationen über ihre Umgebung gewinnen können.

Kamerasysteme bieten hier den Vorteil, eine große räumliche Auflösung im Vergleich zu anderen Sensoren zu haben. Räumliche Informationen können mittels Stereobildverarbeitung gewonnen werden. Sie basieren auf der Analyse von Verschiebungen von Bildinformationen zwischen den beiden Stereobildern sowie einer darauf aufbauenden Triangulation.

Bei mobilen Robotern muss diese Informationsgewinnung zusätzlich in Echtzeit erfolgen, um beispielsweise ein rechtzeitiges Reagieren auf Hindernisse zu ermöglichen. Bildverarbeitung ist in der Regel jedoch ein sehr zeitintensives Verfahren. Trotzdem ist es anwendbar, da zu Gunsten einer kürzeren Laufzeit auf Genauigkeit verzichtet werden kann. Außerdem kann die Verarbeitung redundanter Information vermieden werden, indem zeitliche Redundanzen und Informationen anderer Sensoren hinzugezogen werden.

Echtzeitumgebungen erfordern aber auch vorhersagbare Ausführungszeiten. Im klassischen Echtzeitkonzept wird dies durch die WCET (worst case execution time) definiert. Für viele Algorithmen, so auch die Bildverarbeitung, ist diese jedoch zu pessimistisch. Neue Schedulingalgorithmen wie der TAFT Scheduler [2] verwenden stattdessen die ECET (expected case execution time), welche auf statistischen Informationen aufbaut [3]. Die Laufzeit wird hier garantiert, indem eine Task (die Main Task) abgebrochen wird, wenn sie ihre vorgegebene Laufzeit überschreitet. Dann sorgt eine zweite Task (die Exception Task) für einen sicheren Systemzustand. Diese Task kann auch dafür verwendet werden, Zwischenergebnisse der Main Task einzusammeln [4]. Kann die Exception Task schon nach sehr kurzer Laufzeit der Main Task immer ein brauchbares Ergebnis liefern, handelt es sich um einen Anytime Algorithmus [5].

2 Verwandte Arbeiten

Es gibt eine Reihe Implementierungen für Stereobildverarbeitung, die für den Einsatz unter Echtzeitbedingungen geeignet sind [6],[7]. Allerdings stellen diese Ansätze entweder spezielle Hardwareanforderungen (z.B. die Benutzung von Signalprozessoren wie in [8] und [9]) oder stellen Bedingungen an die Einsatzumgebung [10]. Unser Ziel ist jedoch ein Algorithmus, der auf Standardhardware (in unserem Falle ein PC104+ System) läuft und keine Einschränkungen bezüglich der Zielumgebung hat.

Das Betriebssystem unseres mobilen Roboters verwendet einen TAFT Scheduler. Die zu implementierende Stereobildverarbeitungsmethode muss daher die Anforderungen des TAFT Schedulers erfüllen (z.B. ein Anytime Algorithmus sein) oder sich leicht um diese Eigenschaften erweitern lassen, im Gegensatz zu anderen Arbeiten, die nicht auf Anytime Algorithmen basieren. Dafür wählten wir das Block Matching Verfahren [15]. Es ist ein Vertreter aus der korrelationsbasierten Stereobildverarbeitung oder auch *area based stereo* [11].

Zur Entscheidung beigetragen hat auch die Tatsache, dass *area based stereo* entweder für alle Pixel eines Bildes oder für ausgewählte Bereiche erfolgen kann [9]. Damit ist die Möglichkeit gegeben, Merkmale im Bild zu extrahieren, um nur auf diesen die Berechnungen durchzuführen. Das erlaubt die Betrachtung von einzelnen Bildbereichen, ohne das Gesamtbild verarbeiten zu müssen, was sich wiederum auf die Berechnungsdauer und damit auf die Ergebnisqualität positiv auswirken kann.

3 Stereobildverarbeitung mit Block Matching

Die grundlegende Idee des Block Matching Verfahrens ist ein Ähnlichkeitsvergleich zwischen zwei gleich großen Blöcken ($n \times m$ -Matrix) im Stereobildpaar. D.h. ein Block im linken Bild wird mit allen möglichen Blöcken des rechten Bildes verglichen. Meist wird der mittlere quadratische Fehler (*MSE*) zwischen den Pixelwerten in den Blöcken als Ähnlichkeitsmaß definiert.

Das linke Bild I_L und das rechte Bild I_R sind mit Grauwerten G definiert als

$$I_L(i, j) = G_L(i, j) \text{ und } I_R(i, j) = G_R(i, j). \quad (1),(2)$$

Der *MSE* ist dann definiert mit $n=m=2k+1$ als

$$MSE(x, y, \Delta) = \frac{1}{n \cdot m} \sum_{i=-k}^k \sum_{j=-k}^k (|G_L(x+i, y+j) - G_R(x+i+\Delta, y+j)|^2). \quad (3)$$

Δ ist ein Offset in Pixeln, der die Differenz ($x_L - x_R$) zwischen den Spalten im linken und rechten Bild beschreibt, um den der Block (der Größe $n \times m$) innerhalb des Suchraumes Pixel für Pixel verschoben wird. Bei Benutzung der Standardstereogeometrie, d.h. die Epipolarlinien entsprechen den Bildzeilen (*epipolar constraint*), kann das Problem so weit vereinfacht werden, dass korrespondierende Pixel nur in einer Zeile gefunden werden können, der vertikale Suchraum praktisch nicht betrachtet wird. Die Suche nach gleichen Pixeln wird auch als Korrespondenzproblem bezeichnet.

Die Disparität D von zwei Blöcken in beiden Bildern ist definiert als die horizontale Distanz an dem der *MSE* minimal ist. Des Weiteren wird der Suchraum in horizontaler Richtung im rechten Bild durch einen vordefinierten maximalen Disparitätswert d_{max} begrenzt. D ergibt sich somit aus

$$D = \min_{|\Delta| \leq d_{max}} \{MSE(x, y, \Delta)\}. \quad (4)$$

Die berechneten Blockdisparitäten werden danach mit einem Medianfilter bearbeitet, um Ausreißer zu eliminieren. Die abschließend berechnete Disparität für jeden Pixel des Bildes zur Erzeugung der Disparitätsmatrix wird mit Hilfe einer Pixelselektion durchgeführt (siehe [12] für genauere Information).

Das Block Matching Verfahren liefert in dieser Form Ergebnisse aber erst nach Beendigung des Algorithmus. Ihn in einen Anytime Algorithmus zu transformieren, erfordert die möglichst schnelle Bereitstellung eines ersten Ergebnisses, welches sukzessive verbessert werden muss.

Gelöst haben wird das Problem mit Hilfe eines Pyramidenmodells.

4 Hierarchische Stereobildverarbeitung

Die Idee Pyramidenmodelle für die Bildverarbeitung zu benutzen, stammt von Tanimoto und Pavlidis [14]. Wir erweitern das Block Matching Verfahren, indem wir eine vierstufige Pyramide benutzen.

Das Korrespondenzproblem wird angefangen bei der größten Auflösung auf jeder Ebene der Pyramide Schritt für Schritt gelöst. Die Ergebnisse einer Stufe werden dann als Initialisierungsvektor für das Block Matching Verfahren für die nächste Stufe verwendet. Dadurch wird der Startbereich für die Suche nach Korrespondenzen schon an die in der größeren Auflösung berechneten Stellen verschoben. Das erlaubt wiederum, dass der maximale Disparitätswert d_{max} verkleinert und ein minimaler MSE , also ein hohes Ähnlichkeitsmaß trotzdem schnell gefunden werden kann.

Der Grundgedanke ist die Berechnung in jeder Ebene abbrechen zu können, sollte der Algorithmus wegen zeitlicher Randbedingungen vorzeitig abgebrochen werden. Erfolgt der Abbruch während der Berechnung einer Ebene, was meist immer der Fall ist, werden die fehlenden Ergebnisse dieser Stufe durch die der letzten ergänzt. Dadurch entsteht eine kontinuierliche Verbesserung der Daten.

Ein erstes Ergebnis, sicherlich ein sehr ungenaues, liegt somit sehr schnell vor und kann verwendet werden. Trotzdem sind weniger detaillierte Ergebnisse keinen Ergebnissen vorzuziehen, da sie, um Kollisionen zu vermeiden, schon durchaus eingesetzt werden können.

Es ist nun zu überlegen, wie das Verfahren weiter beschleunigt werden kann, um möglichst häufig hohen Detailreichtum zu erreichen.

Statischer Ansatz

Ein großes Problem im Block Matching Verfahren ist die Größe des Suchraumes. Wenn ein anderes Aufnahmegerät als der Stereo-Kamerakopf eingesetzt wird, kann das *epipolar constraint* nicht mehr angenommen werden. In diesem Fall müsste auch über verschiedene Zeilen gesucht werden. Ein weiterer Punkt ist die Annahme über die Reihenfolge der Pixel auf beiden Bildern. Es wird oft davon ausgegangen, dass Pixel des einen Stereobildes, die

auf einer Epipolarlinie liegen, in genau derselben Reihenfolge auf der korrespondierenden Epipolarlinie abgebildet sind (*ordering constraint*). Diese Annahme trifft aber genau dann nicht mehr zu, wenn zwischen den Objekten in der beobachteten Szene eine große Tiefendifferenz vorliegt.

Aufgrund dieser Überlegungen ist das Block Matching Verfahren dahingehend geändert worden, dass ausgehend von einem Disparitätsstartwert korrespondierende Blöcke im rechten Bild in beiden Richtungen sowohl horizontal als auch vertikal gesucht werden können.

Des Weiteren kann der Parameter d_{max} und damit der horizontale Suchraum auf Grund der Sichtparameter der Kameras eingeschränkt werden. Je dichter sich ein Objekt an den Kameras befindet, desto größer wird die Disparität für die entsprechenden Pixel. Wenn man von einer Mindestentfernung ausgeht, z.B. durch den Aufbau der Stereokamera auf dem Roboter, kann der maximale horizontale Suchraum genau definiert werden.

Aber nicht nur der Suchraum entscheidet über Geschwindigkeit und Qualität der Ergebnisse. Ein wichtiger Parameter ist die Blockgröße. Wir haben verschiedene Blockgrößen getestet und sind zu dem Schluss gekommen, dass, wenn das epipolar constraint zugelassen wird, die Blockhöhe ca. 2/3 der Blockbreite entsprechen sollte. In unseren Tests erzielte die Blockgröße 12x8 Pixel die besten Ergebnisse.

Durch den hierarchischen Aufbau des Algorithmus und den dadurch verfügbaren Zwischenergebnissen haben wir einen Anytime Algorithmus zur Verfügung, der geeignet für das TAFT Scheduling ist. Damit ist eine Basis geschaffen, die aus einem Stereobildpaar Tiefeninformationen in Echtzeit liefern kann. Der nächste Schritt besteht in der Einbeziehung der Informationen über die Eigenbewegung des Roboters unter Verwendung zeitlich aufeinanderfolgender Bilder.

Dynamischer Ansatz

Mit einer Folge von Bildsequenzen unter Berücksichtigung der Eigenbewegung des Roboters kann der Algorithmus in einer feineren Pyramidenebene starten. Um das zu erreichen ist eine Klassifizierung der Bewegungen, die auftreten können notwendig. Bei einem Schreitroboter sind das 6 Freiheitsgrade (Rotationen: Rollen, Gieren, Neigen; Translationen entlang der x-, y- oder z-Achse).

Ausgehend von der Annahme, dass sich keine dynamischen Objekte in der beobachteten Szene befinden, führt die Bewegung des Roboters und damit des Kamerasystems zu einer relativen Positionsänderung zwischen Kamera und Umgebung. Damit hängen die Bildveränderung direkt von der Roboterbewegung ab. Alle Pixel in den Bildern bewegen sich

invers entweder horizontal (Gieren) oder vertikal (Neigen) zur Roboterbewegung oder werden gedreht (Rollen). Bewegt sich der Roboter translatorisch, werden Fluchtlinien für die Vorhersage der Bildveränderungen wie in [13] verwendet.

Durch die Kenntnis Roboterbewegung, kann zusätzlich abgeschätzt werden, wie groß der Parameter d_{max} sein muss, um den Suchraum möglichst klein zu halten. Dadurch wird das Block Matching Verfahren beschleunigt und die Möglichkeit von Fehlzuordnungen wird gesenkt.

Fusionsansatz

Problematisch wird es, wenn perspektivische Änderungen wie Vergrößerungen und Verkleinerungen, hohe Geschwindigkeitsunterschiede der betrachteten Objektbewegungen oder große Änderungen im Betrachtungswinkel im Vergleich zum letzten Bild auftreten. Durch Integration weiterer Sensoren kann diesem Problem entgegengewirkt und das Wissen über die Umgebung verbessert werden.

Während beim dynamischen Ansatz nur bereits gewonnene Ergebnisse aus vorangegangenen Block Matching Durchläufen zur Verbesserung des aktuellen verwendet werden, werden beim Fusionsansatz diese Ergebnisse vorher mit den Informationen weiterer Sensoren verknüpft. Dadurch werden die gewonnenen Umgebungsdaten detaillierter, was zur Folge hat, dass die Initialisierung des darauf folgenden Block Matching Durchlaufes bereits genauer geschehen kann.

5 Ergebnisse

Das beschriebene hierarchische Block Matching Verfahren wurde von uns erfolgreich als Anytime Algorithmus umgesetzt. Dabei stand in erster Linie die möglichst schnelle Bereitstellung eines ersten verwendbaren Ergebnisses im Vordergrund. Dieses Ergebnis wird in jeder Hierarchiestufe sukzessive verbessert. Sollte der Algorithmus auf Grund von zeitlichen Bedingungen abgebrochen werden, können die bisherigen Resultate verwendet werden.

Bild 1 zeigt die relativen Größenverhältnisse der Laufzeiten in den unterschiedlichen Phasen des Algorithmus. Nach der Vorverarbeitung (VV) und der ersten Pyramidenstufe (a-b) ist das erste Ergebnis vorhanden. Die Phasen b-c, c-d und d-e stellen die nachfolgenden Pyramidenstufen dar. Abschließend erfolgt die Nachbearbeitung (NB). Aus dem Blickwinkel des TAFT Schedulers ist die Main Task durch die Bereiche VV bis e definiert, die nach einer kurzen Startphase (VV-b) zu jeder Zeit unterbrochen werden kann. Der Exception Part ist der Nachbearbeitungsteil, in dem die Ergebnisse durch einen Medianfilter aufbereitet und Tiefeninformationen berechnet werden.

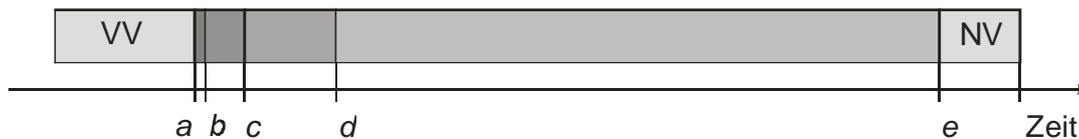


Bild 1: Relative Laufzeiten des Algorithmus in den verschiedenen Phasen

6 Zusammenfassung

In der vorliegenden Arbeit wurde eine hierarchische Stereobildverarbeitungsmethode als Anytime Algorithmus umgesetzt. Ein Block Matching Algorithmus diente als Basis. Er wurde in drei Schritten so modifiziert, dass Ergebnisse auf verschiedenen Auflösungsstufen einer Bildpyramide berechnet werden. Ein schnelles Vorliegen eines ersten Ergebnisses, wenn auch verhältnismäßig ungenau, wurde ermöglicht. Damit ist der Einsatz auf einem Echtzeitsystem mit einem TAFT Scheduler möglich.

Dieser Algorithmus kann nun mit verschiedenen Ansätzen in ein Gesamtsystem eingebettet werden. Beim statischen Ansatz wurde das hierarchische Block Matching direkt zur Gewinnung von Informationen über die Umgebung verwendet. Nachfolgend wurden im dynamischen Ansatz Redundanzen, die sich aus der Eigenbewegung des Roboters ergeben, ausgenutzt, um den Algorithmus gegebenenfalls auf einer höheren Auflösungsstufe vorzuinitialisieren. Im Fusionsansatz schließlich wurde zusätzlich die räumliche Redundanz der Informationen über die Positionen externer Objekte verwendet. Dies geschah unter anderem durch die Anwendung weiterer Sensoren.

Die praktische Umsetzung des statischen Ansatzes zeigt, dass erste verwendbare Ergebnisse über Tiefeninformationen sehr schnell gewonnen und sukzessive verbessert werden konnten, so dass eine echtzeitgerechte Stereobildverarbeitung für mobile Robotikanwendungen ermöglicht wurde.

7 Literatur

- [1] Nett, E.; Gergeleit, M.; Mock, M.: An Adaptive Approach to Object-Oriented Real-Time Computing. In: Proceedings of the IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '98), Kyoto, Japan, 20-22 April 1998.
- [2] Gergeleit, M.; Buss Becker, L.; Nett, E.: Robust Scheduling in Team-Robotics. In Proceedings of Eleventh International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS 2003), April 22-24, 2003 Nice, France.
- [3] Gergeleit, M.; Streich, H.: TaskPair-Scheduling with Optimistic Case Execution Times – An Example for an Adaptive Real-Time System. In: Second International Workshop on

- Object-oriented Real-time Dependable Systems, Laguna Beach, CA., IEEE Computer Society Press, 1996
- [4] Nett, E.; Gergeleit, M.; Mock, M.: An Adaptive Approach to Object-Oriented Real-Time Computing. In: Proceedings of the IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '98), Kyoto, Japan, 20-22 April 1998.
 - [5] Dean, T. L.: Intracibility and Time-Dependent Planning, Workshop on Reasoning about Actions and Plans, 1987
 - [6] Konolige, K.; Small vision systems: Hardware and implementation. In: Eighth International Symposium on Robotics Research, (Hayama, Japan), pp. 203-212, London, Springer, Oktober 1997.
 - [7] Volpe, R.; Balaram, J.; Ohm, T. und Ivlev, R.: The rocky 7 mars rover prototype. In: International Conference on Intelligent Robots and Systems, Vol. 3, (Osaka, Japan), pp. 1558-1564, November 1996
 - [8] Faugeras et al.: Real time correlation-based stereo: algorithm, implementations and applications. Rapport de recherche, no. 2013, Institut national de recherche en informatique et en automatique, 1993
 - [9] Dickmanns, E. D.; Graefe, V.: Dynamic monocular machine vision, Machine Vision and Applications, Springer International, Vol. 1, 1988
 - [10] Siedersberger, K.-H.; Dickmanns, E. D.: EMS-Vision: Enhanced Abilities for Locomotion, Intelligent Vehicles Conference 2000, Dearborn, MI, USA, October 4-5, 2000
 - [11] Zhang, C.: A Survey on Stereo Vision for Mobile Robots. Dept. Of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA
 - [12] A. Koschan, "Chromatic block matching for dense stereo correspondence", Proc. 7th Int. Conf. On Image Analyses and Processing 7ICIAP, Capitolo, Monopoly, Italy, pp. 641-648, 1993
 - [13] Raquel, F.; Vassallo, Schneebeli, Hans J.; Santos-Victor, José: Visual servoing and appearance for navigation. Robotics and Autonomous Systems 31 (2000) 87-97
 - [14] S. Tanimoto, T. Pavlidis, "A Hierarchical Data Structure for Picture Processing", Computer Graphics and Image Processing, vol. 4, pp. 104-119, 1975
 - [15] Klette, R.; Koschan A.; Schlüns, K.: Computer Vision – Räumliche Information aus digitalen Bildern. Vieweg Technik Verlag; 1996