

Realzeitfähige Auswertung des optischen Flusses zur Navigation mit einem mobilen Roboter

Norbert O. Stöffler
Lehrstuhl für Realzeit-Computersysteme
Technische Universität München, D-80333 München
stoffler@rcs.ei.tum.de

Zusammenfassung

Im sogenannten optischen Fluß (*OF*) sind prinzipiell alle zur Fahrzeugnavigation erforderlichen Informationen (Eigen- und Fremdbewegung, Hindernisse) enthalten. Dieser Beitrag stellt eine realzeitfähige Methode zur Berechnung des OF und eine problemorientierte Technik zu seiner Interpretation vor. Im Gegensatz zu ähnlichen Arbeiten sind realistische Fahrzeugbewegungen und Einsatzumgebungen erlaubt. Außerdem liefert das Verfahren zur Rekonstruktion der statischen Hindernisse nicht nur qualitative Aussagen, sondern kartesische Koordinaten.

1. Motivation

Die in diesem Beitrag präsentierten Arbeiten sind Teil eines Forschungsprojektes zur Entwicklung mobiler Roboter, die videobasiert in Innenräumen navigieren. Ein Ansatz zur Interpretation der Videobilder, der nicht durch die Eigenbewegung des Roboters behindert wird, sondern sogar von ihr profitiert, ist die Auswertung des optischen Flusses (*OF*). Vereinfacht dargestellt entsteht dieses Geschwindigkeitsfeld durch Projektion der dreidimensionalen Bewegung in die Bildebene (x, y) laut folgender, wohlbekannter Gleichung:

$$\mathbf{f} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \frac{1}{Z(x, y)} + \begin{bmatrix} xy & -x^2 - 1 & y \\ y^2 + 1 & -xy & -x \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \mathbf{E} \frac{\mathbf{T}}{Z} + \mathbf{H} \boldsymbol{\omega} \quad (1)$$

Da das Feld $\mathbf{f}(x, y)$ nicht nur von der Bewegung, sondern auch von der Tiefe Z der jeweiligen Szenenpunkte abhängt, enthält es sämtliche für die Navigation relevanten Informationen: Ist die Umgebung starr, so entsprechen \mathbf{T} und $\boldsymbol{\omega}$ der Eigenbewegung des Roboters, durch Integration kann die Eigenposition fortgeschrieben werden (*Passive Navigation*). Aus der Tiefenfunktion $Z(x, y)$ kann die Szenenstruktur rekonstruiert und damit z. B. eine Hinderniskarte erstellt werden (*Structure from Motion, SfM*). Enthält die beobachtete Szene auch noch unabhängig bewegte Objekte, so gilt Gl. (1) stückweise innerhalb der zu einem starren (Teil-) Objekt gehörenden Bildregion. Durch eine Segmentierung von $\mathbf{f}(x, y)$ ist damit auch eine Detektion von anderen Fahrzeugen oder von durchs Bild laufenden Personen möglich.

2. Realzeitfähige Bestimmung des optischen Flusses

Die bei realistischen Kamera- und Bewegungsparametern auftretenden Vektorlängen liegen in der Größenordnung von einigen Dutzend Pixeln, so daß die Berechnung von Geschwindigkeiten $\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$ mithilfe der

klassischen Gradientenverfahren nicht in Frage kommt. Stattdessen werden *Verschiebungen* von Bildpunkten bestimmt, und zwar durch Korrelation kleiner Referenzbereiche (RBs) mit entsprechenden Suchfenstern (SWs):

$$\mathbf{f} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \underset{s}{\operatorname{argmin}} \sum_r |SW(\mathbf{s} + \mathbf{r}) - RB(\mathbf{r})| \quad (2)$$

Zur Implementierung dieser Operation in Realzeit kommt in unserem System ein MPEG-Blockmatcher zum Einsatz [9]. Im Gegensatz zu verwandten Arbeiten [5] wird jedoch durch zusätzliche Hardware auch noch ein Konfidenzmaß für jeden Vektor berechnet, das die Signifikanz des bei der Korrelation gefundenen Minimums angibt. Die bei der Korrelation von texturarmen Bildbereichen oft entstehenden Fehlkorrespondenzen können so effektiv ausgesiebt werden. Im besten Fall werden 525 Vektoren pro Videotakt (40 ms) bestimmt, allerdings wird keine Subpixelgenauigkeit erreicht, was zu Quantisierungsrauschen in der Größenordnung von ± 0.5 Pixel führt. Auch verbleibt trotz der Siebung ein kleiner Anteil an falschen Vektoren (z. B. durch zyklische Muster im Bild), die in der folgenden Interpretation als statistische Ausreißer betrachtet werden müssen.

3. Bewegung und Tiefe

Die Inversion von Gl. (1) ist aufgrund ihrer Nichtlinearität nicht ohne weiteres möglich, daher wurde eine Vielzahl von Ansätzen zur Lösung des Problems vorgeschlagen (siehe z. B. [1] für einen Überblick). Unglücklicherweise reagieren all diese Verfahren sehr empfindlich auf Rauschen, so daß sie für die Interpretation der Vektorfelder $\mathbf{f}(x, y)$ laut Kap. 2 zu instabil sind. Ist die Kamera jedoch auf einem mobilen Roboter montiert, reduzieren sich die Freiheitsgrade der Kamerabewegung beträchtlich (siehe Abb. 1). Der Rotationsvektor ω hat nur noch eine vertikale Komponente; für eine große Klasse von Fahrzeugkinematiken ist auch die Richtung des Translationsvektors relativ zum Fahrzeug fix. Ist die Montage der Kamera bekannt und geht die Rotationsachse näherungsweise durch das optische Zentrum, so vereinfacht sich Gl. (1) zu

$$\mathbf{f}(x, y) = \mathbf{e}(x, y) \frac{T}{Z} + \mathbf{h}(x, y) \omega \quad (3)$$

Dieses Gleichungssystem in den beiden skalaren Unbekannten $\frac{T}{Z}$ und ω ist für jeden Vektor \mathbf{f} separat lösbar. Außerdem ist die Tiefenfunktion $Z(x, y)$ zumindest im hindernisfreien Fall über die Gleichung der Bodenebene gegeben.

Ähnliche Vereinfachungen wurden (explizit oder implizit) bereits von einigen Autoren [2, 3, 4, 6, 8] getroffen, die sich ebenfalls mit Fahrzeugnavigation beschäftigen. Diese Arbeiten unterscheiden sich u. a. in den rekonstruierten Parametern und in den erlaubten Werten für ω . Da bei realistischen Geschwindigkeiten

$$\mathbf{e} \frac{T}{Z} \ll \mathbf{h} \omega$$

ist, wird oft von reiner Translation ($\omega \equiv 0$) ausgegangen, oder die Kamera entgegengesetzt zur vorgegebenen Fahrzeugrotation nachgeführt ($\omega \approx 0$). Weitere Unterschiede liegen in der Verwendung des Verhältnisses $\frac{T}{Z}$. Ist die Geschwindigkeit T aus anderen Quellen bekannt, so kann für jeden Vektor \mathbf{f}_i die korrespondierende Tiefe Z_i berechnet werden. Umgekehrt kann im hindernisfreien Fall die Gleichung der Bodenebene

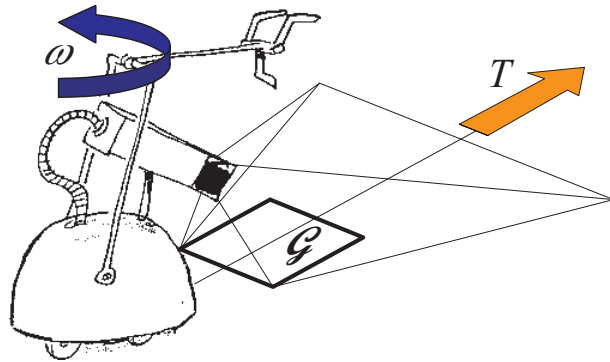


Abbildung 1: Planare Bewegung

eingesetzt, und damit der Absolutwert von T geschätzt werden. Ohne diese Vorkenntnisse ist $\frac{T}{Z}$ auch direkt für verhaltensbasierte Navigationsverfahren verwendbar, da der Kehrwert die verbleibende Zeit bis zum Erreichen des Punktes angibt (*Time To Collision, TTC*).

Diese direkten Lösungen von Gl. (3) nach den beiden Unbekannten nützen jedoch nicht eine strukturelle Besonderheit der Gleichung aus: Alle Vektoren e schneiden sich im selben Punkt, dem sogenannten *Epipol* oder *Focus of Expansion*. Daher wurde von uns ein modifizierter, mehrstufiger Lösungsweg entwickelt.

Schritt 1: Bestimmung von ω

Da ω nicht von Z abhängt, sollten die Lösungen ω_i von Gl. (3) identisch zur Kamerarotation sein. Durch die verschiedenen Rauscheinflüsse entsteht jedoch eine Verteilung, aus der ein Schätzwert für ω mit unterschiedlichen statistischen Ansätzen gewonnen werden kann. *Least-Square*-Verfahren sind zwar bei gaußschem Rauschen ideal, können aber bereits durch wenige Ausreißer völlig verfälscht werden. Robuste Methoden wie *Least-Median* lassen einen Ausreißeranteil von bis zu 50% zu, dieser Anteil kann aber in Gegenwart von *unabhängig bewegten Objekten* schnell überschritten werden. Daher wurde von uns eine Clustertechnik entwickelt, die nicht nur die Verteilung im ω -Raum, sondern auch in der Bildebene berücksichtigt, und selbst dann noch eine verlässliche Schätzung für die Eigenrotation liefert, wenn die zum statischen Hintergrund gehörende Bildregion nur noch wenige¹ Vektoren enthält [10]. Das Komplement der Hintergrundregion liefert die Objektregionen, diese können mit Standardmethoden (z. B. *Region Growing*) weiter segmentiert werden (siehe Abb. 2, links oben). Eine dreidimensionale Rekonstruktion der bewegten Objekte ist allerdings nicht möglich, da Gl. (3) nur in der Hintergrundregion gilt.

Schritt 2: Lösung nach $\frac{T}{Z}$

Der Schätzwert $\hat{\omega}$ wird nun weiter verwendet, um das Flußfeld zu *derotieren*, also den zweiten Term in Gl. (3) zu kompensieren. Der resultierende Vektor

$$d = f - h \hat{\omega} \quad (4)$$

entspricht jedoch aufgrund des Rauschens nicht exakt dem ersten Term $e \frac{T}{Z}$. Im Gegensatz zur direkten Lösung verbleiben jedoch *zwei* Gleichungen zur Bestimmung von $\frac{T}{Z}$, so daß der Wert mit dem Least-Square Ansatz

$$r = d - e \frac{T}{Z}, \quad \|r\|^2 \rightarrow \min \quad (5)$$

bestimmt werden kann. Die so gefundene Gleichung

$$\frac{T}{Z} = \frac{e^T d}{\|e\|^2} \quad (6)$$

kann geometrisch als Projektion von d auf eine Epipolarlinie interpretiert werden. Effektiv wird dadurch die senkrechte Komponente des Quantisierungsrauschens eliminiert. Desweiteren kann über die Länge des verbleibenden Residuums ein weiteres Kriterium zur Erkennung von Ausreißern definiert werden.

Schritt 3: Schätzung von T

Um T aus $\frac{T}{Z}$ berechnen zu können wird im folgenden die Annahme getroffen, daß ein kleiner Bereich \mathcal{G} unmittelbar vor dem mobilen Roboter (siehe Abb. 1) frei von Hindernissen ist. Diese Annahme ist bei der Fahrzeugnavigation ohnehin sinnvoll, da Hindernisse in einem durch die Dynamik bestimmten *Schutzraum* zwangsläufig zu Kollisionen führen würden. Innerhalb dieses Bereiches gilt damit die Ebenengleichung und T kann zu

$$\hat{T} = \text{shorth}_{\mathcal{G}}\left(\frac{e^T d}{\|e\|^2} \cdot Z(x, y)\right) \quad (7)$$

geschätzt werden. Der robuste *Shortest-Half*-Estimator erlaubt wieder einen Ausreißeranteil von 50%, so daß \mathcal{G} auch etwas größer als der dynamisch erforderliche Schutzraum gewählt werden kann. Einsetzen von \hat{T} in Gl. (6) liefert die noch fehlenden Tiefen Z .

¹je nach Schwelle für die minimale Clustergröße, in der momentanen Realisierung 5

Schritt 4: Detektion von Hindernissen

Die verbleibende Rauschkomponente wirkt sich in Abhängigkeit von (x, y) unterschiedlich stark auf die Genauigkeit der Bestimmung von Z aus, so daß eine Erkennung von Hindernissen anhand der Höhe des rückprojizierten Punktes $P = Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ über der Bodenebene nicht praktikabel ist. Besser ist es, die Abweichung

$$\frac{e^\top d}{\|e\|} - \frac{T}{Z} \|e\| \quad (8)$$

in Bildkoordinaten zu berechnen, da diese mit einer festen Schwelle in der Größenordnung des Quantisierungsrauschens verglichen werden kann. Abb. 2 zeigt links unten die berechneten Flußvektoren und detektierte Hindernispunkte. Rechts sind die Punkte bereits in Roboterkoordinaten transformiert, zur Bahnplanung können dann Standardmethoden eingesetzt werden.

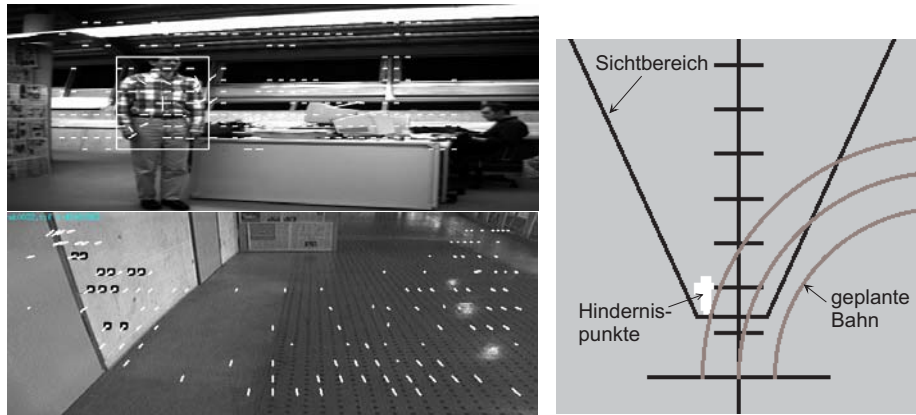


Abbildung 2: Detektion bewegter Objekte und statischer Hindernisse

4. Zusammenfassung und Bewertung

Bereits bekannte Techniken zur Bestimmung und Auswertung des optischen Flusses wurden soweit verbessert, daß damit die Realisierung eines Realzeitsystems zur Schätzung der Eigenbewegung und Detektion von statischen und bewegten Objekten möglich wurde. Ein wesentlicher Fortschritt dabei ist die kartesische Bestimmung der Hindernispositionen. Dies erlaubt nicht nur die Verwendung von Standardverfahren zur Bahnplanung, auch die Fusion mit den Daten anderer Sensoren wird vereinfacht. Der zweite Punkt ist in der Praxis sehr wichtig, da ein realer mobiler Roboter zur Durchführung sinnvoller Aufgaben weitere (logische) Sensoren, z. B. zur Erkennung missionsrelevanter Objekte, benötigt. Das OF-Subsystem funktioniert jedoch für sich alleine schon so robust, daß es unseren Versuchsroboter *MARVIN* [7] in die Lage versetzt, bei typischen Geschwindigkeiten von 0.4 m/s und Rotationsgeschwindigkeiten bis zu 30°/s in Innenräumen kollisionsfrei zu navigieren und die Eigenposition fortzuschreiben. Solange die Annahme der planaren Bewegung erhalten bleibt, ist auch ein Einsatz in Aussenumgebungen denkbar.

Literatur

- [1] Åström, K.: Multiple View Vision. In: *Proc. 15th Int. Conf. on Pattern Recognition* Bd. 1. Barcelona, E : IEEE Computer Society Press, September 2000, S. 59–66.
- [2] Coombs, D.; Herman, M.; Hong, T.-H.; Nashman, M.: Real-Time Obstacle Avoidance using Central Flow Divergence, and Peripheral Flow. In: *IEEE Trans. on Robotics and Automation* 14 (1998), Februar, Nr. 1, S. 49–59.

- [3] Enkelmann, W.: Ein System zur Hindernisdetektion durch Real-Zeit-Auswertung von optischen Fluß-Vektoren. In: *FhG/IITB Mitteilungen* (1995), S. 5–11.
- [4] Giachetti, A.; Campani, M.; Torre, V.: The Use of Optical Flow for Road Navigation. In: *IEEE Trans. on Robotics and Automation* 14 (1998), Februar, Nr. 1.
- [5] Inoue, H.; Inaba, M.; Mori, T.; Tachikawa, T.: Real-Time Robot Vision System based on Correlation Technology. In: *Proc. Int. Symposium on Industrial Robots (ISIR)*, 1993, S. 675–680.
- [6] Kröse, B.; Dev, A.; Benavent, X.; Groen, F.: Vehicle navigation on optic flow. In: *RWC Symposium*. Tokyo, J, 1997.
- [7] MARVIN - Mobile Autonomous Robot with VISION-based Navigation. <http://www.rcs.ei.tum.de/~stoffler/marvin.html>.
- [8] Santos-Victor, J.; Sandini, G.: Visual-Based Obstacle Detection: A purposive approach using the normal flow. In: et al., U. R. (Hrsg.): *Intelligent Autonomous Systems*, IOS Press, 1995.
- [9] Stöffler, N. O.; Färber, G.: An Image Processing Board with an MPEG Processor and Additional Confidence Calculation for Fast and Robust Optic Flow Generation in Real Environments. In: *Proc. Int. Conf. on Advanced Robotics (ICAR'97)*. Monterey, CA, USA, Juli 1997, S. 845–850.
- [10] Stöffler, N. O.; Schnepf, Z.: An MPEG-Processor-based Robot Vision System for Real-Time Detection of Moving Objects by a Moving Observer. In: *Proc. 14th Int. Conf. on Pattern Recognition*. Brisbane, QLD, AU, August 1998, S. 477–481.