Oberseminar Mobile Robotik WS 2005/2006
Seminarthema: Visuelle Navigation
Technische Universität Dresden – Fakultät für Elektrotechnik und Informationstechnik – Institut für Automatisierungstechnik

NAVIGATION MITTELS OPTISCHEM FLUSS - ERKENNUNG DER DOMINANTEN EBENE -

Müller René, Veigel Thomas

Zusammenfassung: Ziel dieses Papers ist, es einen Algorithmus vorzustellen und zu implementieren, der mit Hilfe des optischen Flusses die dominante Ebene erkennen und auswerten kann. Bei der selbständigen Navigation von Robotern ist es wichtig, Regionen zu finden, in denen sich der Roboter frei, d.h. ohne auf Hindernisse zu treffen, bewegen kann. Die dabei zu erkennende dominante Ebene ist meist die Projektion einer ebenen Fläche wie z.B einer Straße oder eines Fußbodens. Wenn der Roboter anhand der Daten erkennt, dass die ebene Fläche aufgrund von Hindernissen in seiner Bewegungsrichtung nicht weiterführt, muss er sich einen neuen Weg suchen. Der hier vorgestellte Algorithmus arbeitet mit nur einer Kamera und geht davon aus, dass die Bewegungsebene flach ist und dass Objekte ausreichend erkennbar sind.

Keywords: Visuelle Navigation, Dominante Ebene, Optischer Fluss

1 EINLEITUNG

Bei der autonomen Navigation von Robotern ist es wichtig, dass der Roboter selbständig Regionen findet, in denen er sich frei, d.h. ohne auf Hindernisse zu treffen, bewegen kann. Die Grundlage des hier vorgestellten Algorithmus ist eine Kamera die auf dem Roboter montiert ist. Der geringere Hardwareaufwand wird mit einem erhöhten Rechenaufwand erkauft. Der Vorteil besteht darin, dass die Kamerabilder vielseitig einsetzbar sind. Aus den Kamerabildern wird der optische Fluss, d.h. die relative Bewegung eines Bildpunktes zwischen zwei Bildern, ermittelt. Daraus kann man die Eigenbewegung des autonomen Systems bestimmen bzw. Hindernisse erkennen. Alternativ zu dem hier vorgestellten Algorithmus zur Hinderniserkennung gibt es u.a. noch die Methode der Purposive Active Vision [3]. Bei der Erkennung der dominanten Bodenebene wird der errechnete optische Fluss mit einer Vorlage verglichen. Die Vorlage ist im Idealfall der optische Fluss zwischen zwei Kamerabildern ohne Hindernisse. In der Realität wird die Vorlage während der Lernphase ermittelt. D.h. der Roboter bewegt sich translatorisch bzw rotatorisch im Raum ohne Hindernisse im Sichtfeld der Kamera. Die Kenntnisse über Hindernisse im Bewegungsbereich des Roboters bilden wiederum die Grundlage für die visuelle Navigation und somit für die Steuerung der Aktoren.

1.1 Dominante Ebene

Die dominante Ebene ist die Projektion einer ebenen Fläche auf die Bildebene der Kamera. Bei der Roboternavigation ist die dominante Ebene meist die Bodenebene, d.h. die Projektion einer Straße oder wie in unserem Fall die Projektion eines Fußbodens auf die Bildebene. Der hier vorgestellte Algorithmus ermöglicht die Auswertung der dominanten Ebene. Er kann anhand der Bilder erkennen, ob die dominante Ebene in der Bewegungsrichtung des Roboters weiterführt. Ist dies nicht der Fall, befindet sich ein Hindernis vor dem Roboter und er kann seine Bewegungsrichtung den gegebenen Umständen anpassen.

1.2 Randbedingungen

Um die Möglichkeiten der Roboterbewegung etwas einzuschränken wurden folgende Bedingungen festgelegt:

- der Roboter stoppt, wenn er seine Bewegungsrichtung ändert
- der Roboter bewegt sich entweder translatorisch oder rotatorisch

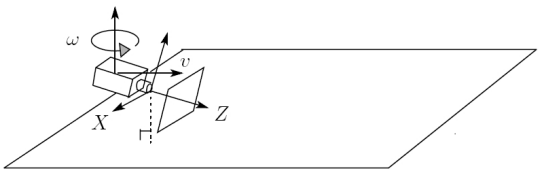

Abbildung 1: Kamera Koordinatensystem und Kamera Bewegung [1]

- der Roboter bewegt sich mit konstanter Geschwindigkeit bzw. konstanter Winkelgeschwindigkeit
- die Bewegung erfolgt parallel zur Bodenebene

Aus der letzten Bedingung ergibt sich ein Koordinatensystem, wie in Abbildung 1 gezeigt.

2 OBJEKTERKENNUNG

Die Kamera des Roboters liefert Bilder mit der dominanten Ebene und Hindernissen. Da die Kamera fest auf dem mobilen Roboter befestigt ist, bewegt sich die Kamera zusammen mit dem Roboter parallel zur dominanten Ebene. Die Bewegung der Hindernisse, wie auch der dominanten Ebene, wird durch den optischen Fluss in der Bildebene dargestellt. Dabei ist der optische Fluss eines Hindernisses anders als der der dominanten Ebene (siehe auch Abbildung 2). Vergleicht man nun den aktuellen optischen Fluss mit einem optischen Fluss derselben Situation ohne Hindernisse (statisches lokales Verschiebungsfeld, Abbildung 4), so kann man durch die Unterschiede die Hindernisse erkennen. Dargestellt ist dies in Abbildung 3.

2.1 Bewegungsgleichung

Gleichung (1) beschreibt die Bewegung des Roboters.

$$w = w_t + w_r = \frac{1}{z_p} A(u_p, v_p) * v + B(u_p, v_p) * \omega \quad (1)$$

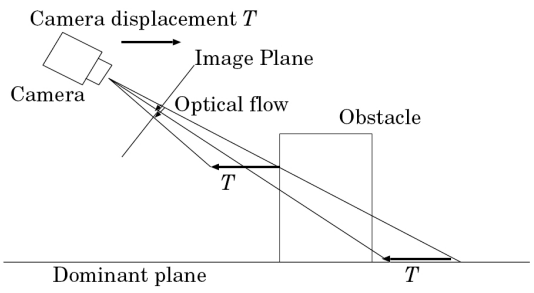

Abbildung 2: Unterschied zwischen dem optischen Fluss mit und ohne Hindernis [2]

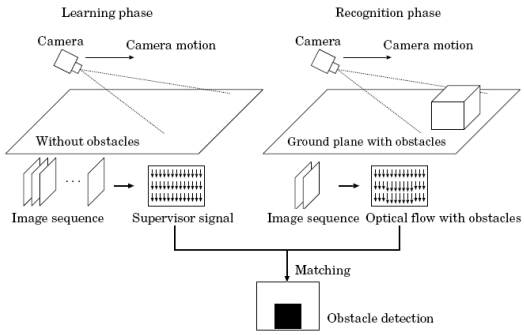

Abbildung 3: Realisierung der Objekterkennung [2]

z_p ist der Abstand eines Punktes im Blickfeld der Kamera vom Roboter. w_t ist der translatorische Teil und w_r ist der rotatorische Teil der Bewegung des Roboters.
Aus Gleichung (1) folgt, dass w_r nicht von z_p abhängt, deshalb ist der optische Fluss für den rotatorischen Teil der Bewegung unabhängig vom Abstand des Roboters zum Hindernis. Daraus folgt, dass sich bei einer rein rotatorischen Bewegung aus dem optischen Fluss keine Information über evtl. vorhandene Hindernisse gewinnen läßt. Im Folgenden betrachten wir daher nur noch die translatorische Bewegung.

2.2 Statisches lokales Verschiebungsfeld

Als Referenz wird das statische lokale Verschiebungsfeld für die Translation des Raumes benötigt, in dem sich der Roboter später autonom bewegen soll. Dies kann man prinzipiell auf zwei Arten gewinnen:

- Mathematisch berechnen
- Lernphase

Bei der ersten Methode berechnet man aus der Kamerageschwindigkeit, der Kamerahöhe und dem Kamerawinkel einen optischen Fluss, der sich ergeben müsste, wenn sich kein Hindernis vor dem Roboter befindet. Für die zweite Methode wird das Sichtfeld der Kamera eingeschränkt und man lässt den Roboter durch einen kleinen Teil des Raumes ohne Hindernisse fahren. Dabei wird eine translatorische Bewegung ausgeführt und die Bilder werden aufgezeichnet. Aus diesen Bildern wird nun der optische Fluss berechnet. Für die Translation wird das statische lokale Verschiebungsfeld ermittelt.

Wir haben den zweiten Weg gewählt und aus den vorliegenden Bildsequenzen solche ausgesucht, die

keine Hindernisse in unserem Kamerabereich enthalten. Als nächstes haben wir den Mittelwert aus den zugehörigen optischen Flüssen gebildet. Daraus haben wir das statische lokale Verschiebungsfelder für die Translation erhalten.

3 MATHEMATISCHE GRUNDLAGEN

3.1 Bewegung auf der Ebene

Wir definieren uns ein dreidimensionales kartesisches Koordinatensystem mit dem Ursprung in der Kameralinse. Außerdem einen Punkt $\mathbf{X} = (X, Y, Z)^T$, einen Normalenvektor $\mathbf{n} = (a, b, c)^T$ und d als Abstand zwischen Kameralinse und der ebenen Fläche. Die ebene Fläche wird durch folgende Gleichung beschrieben

$$\mathbf{n}^T\mathbf{X} = d \tag{2}$$

Weiter definieren wir uns ein x-y Bildkoordinatensystem so, dass der Nullpunkt auf der Z-Achse liegt und die x- und y-Achse parallel zur X-, Y-Achse liegen. Somit ist $\mathbf{x} = (x, y, f)^T$ eine perspektivische Projektion von $\mathbf{X}$ auf die Bildebene.

ω ist die rotatorische und υ die translatorische Geschwindigkeit der Kamera. Nehmen wir an, dass sich das Kamerakoordinatensystem in einer statischen Szene mit unendlich kleiner gerichteter Bewegung (ω, υ) bewegt, ergibt sich folgender mathematischer Zusammenhang

$$\dot{n} = -\omega \times \mathbf{n}, \qquad \dot{d} = -\mathbf{n}^T\upsilon \tag{3}$$

Laut Definition bewegt sich die Kamera entweder translatorisch oder rotatorisch um eine feste Achse. Bewegt sich die Kamera translatorisch in einer stationären Szene, ergeben sich für die ebene Oberflächenbewegung folgende Gleichungen

$$\dot{n} = 0 \qquad \dot{d} = -\mathbf{n}^T\upsilon \tag{4}$$

Bewegt sich die Kamera rotatorisch in einer stationären Szene, ergeben sich für die ebene Oberflächen Bewegung folgende Gleichungen

$$\dot{n} = -\omega \times \mathbf{n} \qquad \dot{d} = -\mathbf{n}^T(\omega \times \mathbf{C}) \tag{5}$$

$\mathbf{C}$ ist das Zentrum der Kreisförmigen Bahnbewegung. Wir gehen davon aus, dass die Kamera um einen fixen Punkt $\mathbf{C}$ rotiert, welcher nicht mit dem ursprünglich im Kamerakoordinatensystem definierten Punkt übereinstimmt.

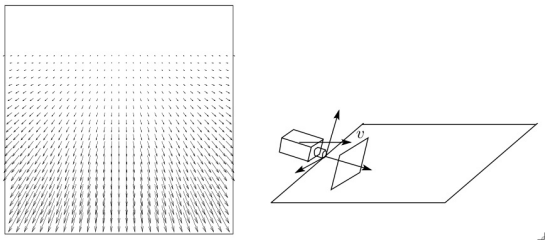

Abbildung 4: Translatorische Bewegung [1]

3.2 Statisches lokales Verschiebungsfeld

Die unendlich kleine gerichtete Bewegung (ω, υ) erzeugt ein lokales Verschiebungsfeld $\dot{x}$ auf der Bildebene, welches unveränderlich mit der Zeit ist.

These: *Wir nehmen wie zuvor definiert an, dass* $\boldsymbol{n}$ *der Oberflächen-Normalenvektor und d der Abstand zur Bodenebene ist. Wenn sich jetzt das Kamerasystem mit einer konstanten translatorischen Geschwindigkeit* υ *parallel zur Bodenebene bewegt, und die Brennweite sich nicht ändert, dann ist die lokale Verschiebung* $\dot{x}$ *der Bodenebene unabhängig von der Zeit.*

Wenn sich das Kamerakoordinatensystem parallel zur ebenen Oberfläche bewegt, dann stehen der Vektor der translatorischen Geschwindigkeit υ und der Oberflächen-Normalenvektor senkrecht aufeinander. Daraus ergibt sich folgender mathematischer Zusammenhang

$$\mathbf{n}^T\upsilon = 0 \tag{6}$$

in Verbindung mit Gleichung (4) erhalten wir

$$\dot{n} = 0, \qquad \dot{d} = 0 \tag{7}$$

Wie man erkennt, ergibt sich aus der translatorischen Bewegung zu jeder Zeit die gleiche lokale Verschiebung $\dot{x}$ der Bodenebene.

3.3 Algorithmus

Wir nehmen an, dass $\dot{x}$(x,y) die lokale Verschiebung der Bodenebene an der Stelle (x,y) der Bildebene und $\dot{x}_t$ die lokale Verschiebung an der gleichen Stelle zum Zeitpunkt t ist. Wenn sich das Kamerasystem parallel zur Bodenebene translatorisch bewegt und sich kein Hindernis an der Stelle (x,y) befindet, dann ist $\dot{x}$ gleich $\dot{x}_t$. In diesem Fall gelten die folgenden Beziehungen

$$\frac{\dot{x}_t(x,y)^T\dot{x}(x,y)}{\|\dot{x}_t(x,y)\|\|\dot{x}(x,y)\|} = 1, \quad \frac{\|\dot{x}_t(x,y)\|}{\|\dot{x}(x,y)\|} = 1 \tag{8}$$

Es ergibt sich eine Grenze G von 1. Wenn jedoch $\dot{x}_t$(x,y) die lokale Verschiebung eines Hindernisses

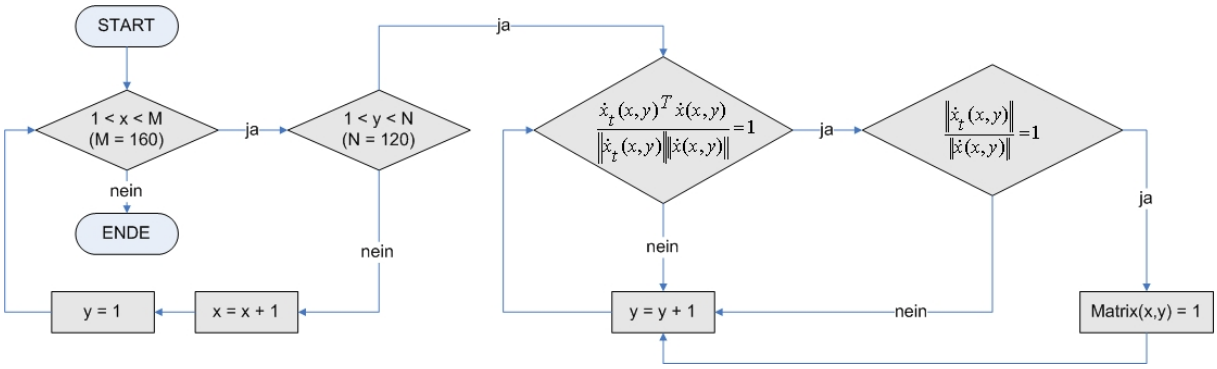

Abbildung 5: zu realisierender Algorithmus

ist, dann ist $\dot{x}_t$(x,y) im Normalfall ungleich $\dot{x}$(x,y). In diesem Fall gelten die folgenden Beziehungen

$$\frac{\dot{x}_t(x,y)^T \dot{x}(x,y)}{\|\dot{x}_t(x,y)\|\|\dot{x}(x,y)\|} \neq 1, \quad \frac{\|\dot{x}_t(x,y)\|}{\|\dot{x}(x,y)\|} \neq 1 \qquad (9)$$

Gleichung (9) gilt jedoch nicht, wenn x_t(x,y) eine lokale Verschiebung der Grenzen zwischen der Bodenebene und einer anderen ebenen Oberfläche ist. Solange folgende Beziehung gilt

$$\left(\frac{\mathbf{n}}{d} - \frac{\mathbf{n}'}{d'}\right)^T \mathbf{x} = 0 \qquad (10)$$

hat Gleichung (9) keine Gültigkeit. Gleichung (8) ermöglicht es uns zu überprüfen, ob $\dot{x}_t$(x,y), sofern wir $\dot{x}$(x,y) kennen, eine lokale Verschiebung der Bodenebene ist oder nicht. Die lokale Verschiebung $\dot{x}$(x,y) erhalten wir, indem wir im Voraus die Bodenebene ohne Hindernisse auswerten. Wenn wir das vorausbestimmte $\dot{x}$(x,y) als Vorlage benutzen, können wir die Bodenebene in einer Videosequenz ermitteln, indem wir die Vorlage und die Verschiebung zum Zeitpunkt t in Gleichung (8) einsetzen. Dadurch erhalten wir einen einfachen Algorithmus zur Erkennung der Bodenebene zum Zeitpunkt t. Dieser Algorithmus geht von idealen Bedingungen aus. In der Realität ist die Länge bzw. die Richtung der Verschiebungsvektoren nur zufällig identisch, daher werden die Bedingungen so gut wie nie erfüllt sein. Experimentell haben wir geeignete Grenzen ermittelt. Als sinnvollen Wertebereich für G ergibt sich G $\in$ [0.85, 1.15].

4 DEMONSTRATIONS-PROGRAMM

4.1 Allgemeines

Zur Demonstration des oben vorgestellten Algorithmus haben wir ein Demonstrationsprogramm entwickelt, welches die Ergebnisse anschaulich darstellt. Es ist in Matlab geschrieben und benutzt als Eingangsdaten die vorgegebenen Matrizen mit dem optischen Fluss. Zusätzlich wird das zugehörige Bild dargestellt, um eine höhere Anschaulichkeit zu erreichen. Die Bedienung ist recht simpel, Hinweise dazu finden sich im Abschnitt Bedienhinweise.

4.2 Bedienhinweise

Nach dem Start des Programmes sind zunächst das Startbild und die Anzahl der Bilder, die man sich anschauen möchte, auszuwählen. Dabei ist zu beachten, dass die Zahl für das Startbild zwischen 2 und 557 liegen muss. Die Anzahl der Bilder kann maximal 555 betragen. Bevor man die Berechnung startet, kann man noch wählen, ob der ideale optische Fluss oder der gestörte optische Fluss als Datenquelle benutzt werden soll. Während der Berechnung erhält man wie in Abbildung 6 gezeigt folgende Angaben:

- das aktuelle Bild,
- den aktuellen optischen Fluss,
- das Berechnungsergebnis für die translatorische Bewegung und
- das Berechnungsergebnis für die rotatorische Bewegung.

Zu Beachten ist, dass immer nur ein Berechnungsergebnis angezeigt wird, entweder für die Translation oder für die Rotation. Dabei ist nur das Ergebnis für die Translation sinnvoll, wie in Abschnitt 2.1 gezeigt.

5 ERGEBNISSE

Bei unseren Experimenten mit dem Demonstrationsprogramm ist es uns gelungen, gute Ergebnisse für die Translation zu erzielen. Durch die Einschränkung des Sichtfeldes und die Mittelung über

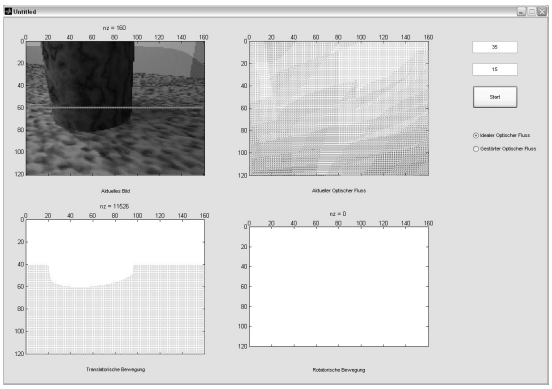

Abbildung 6: Demonstrationsprogramm

mehrere Bildsequenzen konnten wir ein gutes statisches lokales Verschiebungsfeld für die Translation erstellen. Die Fehlerrate ist bei der Auswertung recht gering.

Für die eigentliche Navigation des Roboters ist es wichtig, dass er Hindernisse, die in Richtung seiner Bewegung liegen, erkennt. Da er sich bei rotatorischer Bewegung auf der Stelle bewegt und so nicht auf Hindernisse treffen kann, ist in dem Fall eine Erkennung nicht notwendig. Befindet sich nach einer Drehung des Roboters erneut ein Hindernis in Fahrtrichtung, erkennt dies unser Algorithmus umgehend, sobald der Roboter anfängt sich erneut translatorisch zu bewegen.

Literatur

[1] Kawamoto, Yamada, Imiya, Klette (2002). *Navigation Using Optical Flow Fields: An Application of Dominant Plane Detection*

[2] Ohnishi, Imiya (2005). *Dominant Plane Detection Using Optical Flow and Independent Component Analysis*

[3] Young, Hong, Herman, Yang (1992). *Obstacle Detection and Terrain Characterization Using Optical Flow Without 3-D Reconstruction*