

Abstract

Cognitive maps, as applied to robot navigation, usually contain only landmark information that is stable over large time periods. In order to make use of such a map, e.g. for executing route plans, the robot also needs to be able to react to unpredictable events, e.g. dynamic changes or other uncertainties.

Because of their capability to react quickly to appearing obstacles insects have often been used as examples for designing motor controls applied in autonomous robots. A view on the visual system of insects reveals that the poor spatial resolution is compensated by an excellent temporal resolution. Some insects are capable to perceive over 300 images per second whereas humans are limited to about 60-65 images per second. Therefore flying insects are able to robustly determine the spatial structure of the environment based on fast and precise flow computation.

In order to simulate the insects vision we use a catadioptric vision system (a panoramic mirror enabling a 360 degree field of view) mounted on a Khepera-Robot. In a first step the optical flow field is computed using the Lucas-Kanade-Algorithm, a standard differential technique which uses a weighted least square estimate of the brightness change constraint equation. In a second step, local flow estimates are integrated over specific regions of the robots field of view using a robust statistical estimator used to solve the problem of strong outliers. In order to build a reactive system, we directly use the output of the flow computation as an input to the motor controller instead of reconstructing the local environment.

One of the initial results is the robots capability to centre in a corridor, adjusting its speed to the width of the corridor so that it moves slower and thereby safer in narrow passages. The derived motor controller is based on the assumption that the magnitude of the integrated translational flow is indirect proportional to the distance. Therefore the field of view is divided into two hemispheres and the robot always orients towards the smaller flow until a balance is achieved. If only one wall is visible the robot tries to keep a fixed distance to the wall. This behaviour is used to bridge gaps in the corridor. Finally the robot can dock to a wall by adjusting its velocity with respect to the distance to the wall until it grinds to a halt.

Zusammenfassung

Kognitive Karten, wie sie für die Roboternavigation genutzt werden, beinhalten in der Regel nur Landmarkeninformationen, welche über lange Zeiträume stabil sind. Um solche Karten zum Beispiel bei der Ausführung von Routenplänen nutzen zu können, müssen Roboter jedoch auch in der Lage sein, auf unvorhersagbare Ereignisse wie dynamische Veränderungen zu reagieren.

Aufgrund ihrer Fähigkeit schnell auf erscheinende Hindernisse zu reagieren wurden oft Insekten als Vorbilder für das Design von Motorregelungen in autonomen Robotern herangezogen. Ein Blick auf das visuelle System von Insekten zeigt, dass die schlechte räumliche Auflösung durch eine exzellente zeitliche Auflösung kompensiert wird. Manche Insekten sind in der Lage, mehr als 300 Bilder pro Sekunde wahrzunehmen wohingegen das menschliche Auge auf ca. 60-65 Bilder pro Sekunde begrenzt ist. Dadurch sind fliegende Insekten in der Lage, anhand von schnellen und präzisen Flussberechnungen robust die räumliche Struktur der Umwelt abzuschätzen.

Um die Sicht der Insekten zu simulieren benutzen wir ein katadioptrisches System (ein kegelförmiger Spiegel, welcher eine 360 Grad Sicht erlaubt), welches auf einem Khepera-Roboter angebracht ist. In einem ersten Schritt wird das optische Flussfeld anhand des Lucas-Kanade-Algorithmus berechnet. Dies ist ein differentielles Verfahren, welches eine gewichtete minimale Fehlerquadratsschätzung der Helligkeitsveränderungsgleichung benutzt. In einem zweiten Schritt werden die optischen Flüsse über bestimmte Regionen des Sichtfeldes des Roboters integriert, wobei ein robuster statistischer Schätzer benutzt wird um starke Ausreißer zu filtern. Um ein reaktives System zu erstellen benutzen wir den berechneten Fluss direkt als Eingabe für den Motorregler anstatt eine lokale Umgebung zu rekonstruieren.

Eines der ersten Ergebnisse ist die Fähigkeit des Roboters, sich in einem Korridor zu zentrieren und dabei seine Geschwindigkeit an die Breite des Korridors anzupassen, so dass er sich in engen Passagen langsamer und damit sicherer bewegt. Der abgeleitete Motorregler basiert auf der Annahme, dass der Betrag des integrierten Translationsflusses indirekt proportional zu der Distanz der Umwelt ist. Daher ist das Blickfeld in zwei Hemisphären aufgeteilt und der Roboter orientiert sich immer in Richtung des kleineren Flusses bis eine Balance erreicht ist. Wenn nur eine Wand sichtbar ist, versucht der Roboter eine gleichbleibende Distanz zu dieser Wand aufrechtzuerhalten. Dieses Verhalten kann genutzt werden, um Aussparungen in einem Korridor zu überbrücken. Letztendlich kann er an eine Wand andocken indem er seine Geschwindigkeit an die Distanz zu der Wand anpasst bis er schließlich in unmittelbarer Nähe zum Stillstand kommt.

Inhaltsverzeichnis

1	Einführung	1
1.1	Fragestellung	1
1.2	Biologische Motivation	2
1.2.1	Aufbau des Insektenauges	2
1.2.2	Neuronale Verschaltung auf höherer Ebene	3
1.2.3	Experimente mit Insekten	4
1.2.3.1	Entfernungsmessung der Heuschrecke	4
1.2.3.2	Korridorflug	5
1.3	Omnidirektionale Kamerasysteme	6
1.4	Der optische Fluss	7
1.4.1	Das zweidimensionale Bewegungsfeld \vec{v}	7
1.4.2	Der optische Fluss als messbarer Teil des Bewegungsfeldes	9
1.4.3	Optischer Fluss durch Eigenbewegungen	9
1.4.3.1	Rotationsfluss	10
1.4.3.2	Translationsfluss	11
1.4.4	Methoden der Flussmessung	12
1.4.4.1	Korrelationsdetektoren	12
1.4.4.2	Ableitungsdetektoren	13
1.4.4.3	Bewegung zweiter Ordnung	13
1.4.5	Aperturproblem	13
1.5	Literatur	15
1.6	Aufgabenstellung	15
1.7	Struktur des Dokuments	16
2	Material	17
2.1	Khepera-Roboter	17
2.2	Testumgebung	19
2.2.1	Tracking-System	19
2.2.2	Wandtexturen	19
2.3	PC-Ausstattung	21
3	Algorithmen	21
3.1	Flussberechnung	21
3.1.1	Grundlagen des Lucas-Kanade-Algorithmus	21
3.1.2	Merkmalsauswahl und Konfidenzmaß	23
3.1.3	Entfaltung des Panoramabildes	24
3.1.4	Fluss-Histogramme	25
3.1.5	Parameter der Flussberechnung	26
3.2	Korridorzentrierung	27
3.2.1	Aufgabenbeschreibung	27
3.2.2	Genutzte Informationen des Flussfeldes	27

3.2.3	Ausrichtung des Roboters	28
3.2.4	Geschwindigkeitsregulation	30
3.2.5	Parameter der Korridorzentrierung	31
3.3	Wandverfolgung	33
3.4	Andocken	33
4	Experimente und Ergebnisse	36
4.1	Zentrierung im geraden Korridor	37
4.1.1	Korridor mit einer Breite von 30 Zentimetern	37
4.1.1.1	Versuchsaufbau	37
4.1.1.2	Ergebnisse	38
4.1.2	Korridor mit einer Breite von 15 Zentimetern	41
4.1.2.1	Versuchsaufbau	41
4.1.2.2	Ergebnisse	41
4.1.3	Kontrastempfindlichkeit	43
4.1.3.1	Versuchsaufbau	43
4.1.3.2	Ergebnisse	44
4.2	Zentrierung im sich verengenden Korridor	45
4.2.1	Versuchsaufbau	45
4.2.2	Ergebnisse	45
4.3	Zentrierung im gekrümmten Korridor	46
4.3.1	Versuchsaufbau	46
4.3.2	Ergebnisse	46
4.4	Wandverfolgung	49
4.4.1	Versuchsaufbau	49
4.4.2	Ergebnisse	49
4.5	Andocken	50
4.5.1	Versuchsaufbau	50
4.5.2	Ergebnisse	50
5	Diskussion und Schlussfolgerung	53
6	Ausblick	56

Abbildungsverzeichnis

1	Drosophila Melanogaster	2
2	Aufbau eines Ommatidiums	3
3	Optische Ganglien einer Fliege	4
4	Bewegungsantworten des visuellen Neurons VS8	5
5	Verhalten der Biene im Korridor	6
6	Katadioptrisches System mit mehreren Spiegeln	7
7	Katadioptrisches System des Khepera-Roboters	7
8	Projektion von Bewegungen auf die Bildebene	8
9	Auswirkung einer sich bewegenden Lichtquelle	9
10	Flussfelder einer Rotation	11
11	Flussfelder einer Translation	12
12	Bewegungen zweiter Ordnung	14
13	Darstellung des Aperturproblems	14
14	Beispielhafte Umgebung für eine lokale Navigationsaufgabe .	15
15	Khepera-Roboter	17
16	Panorama-Spiegel des Khepera-Roboters	18
17	Anordnung der Infrarot-Sensoren	18
18	Aufbau der Testumgebung	19
19	Testumgebung, aufgenommen von der Tracking-Kamera . . .	20
20	Texturen der Wände	20
21	Panoramabild eines Outdoor-Roboters	25
22	Histogramm mit Ausreißern	26
23	Aufteilung der Translationsbewegung	28
24	Zentrierung: Blickrichtungen des Roboters	29
25	Auswirkung der Blickwinkel	30
26	Panoramabild und entfaltetes Bild	31
27	Histogramme verschiedener Blickrichtungen	32
28	Kabelführung	33
29	Wandverfolgung: Blickrichtungen des Roboters	34
30	Andocken: Blickrichtungen des Roboters	35
31	Aufbau des 30 Zentimeter breiten Ganges	36
32	30 cm Korridor, Klassenbreite 0,4: Trajektorien	36
33	30 cm Korridor, Klassenbreite 0,4: Standardabweichungen . .	37
34	30 cm Korridor, Klassenbreite 0,4: Flussverlauf	37
35	Aufenthaltswahrscheinlichkeit des Roboters	38
36	30 cm Korridor, Klassenbreite 0,1: Trajektorien	38
37	30 cm Korridor, Klassenbreite 0,1: Standardabweichungen . .	39
38	30 cm Korridor, Klassenbreite 0,1: Flussverlauf	39
39	Aufbau des 15 Zentimeter breiten Ganges.	40
40	15 cm Korridor: Trajektorien	40
41	15 cm Korridor: Standardabweichungen	41
42	15 cm Korridor: Flussverlauf	41

43	Texturen mit unterschiedlich starken Kontrasten	42
44	Khepera-Roboter vor kontrastarmer Textur.	42
45	Schwach texturierter Korridor: Trajektorien	43
46	Schwach texturierter Korridor: Standardabweichungen	43
47	Aufbau des sich verengenden Korridors.	44
48	Verengender Korridor: Trajektorien	44
49	Verengender Korridor: Standardabweichungen	45
50	Geschwindigkeit in Abhängigkeit der Korridorbreite	46
51	Aufbau des gekrümmten Korridors	47
52	Aufbau einer Gasse mit Modellhäusern	47
53	Gekrümmter Korridor: Ergebnisse	48
54	Gasse mit Modellhäusern: Trajektorien	48
55	Experimentaufbau für die Wandverfolgung.	49
56	Wandverfolgung: Trajektorien	50
57	Wandverfolgung: Standardabweichungen	50
58	Experimentaufbau für das Andocken	51
59	Andocken: Trajektorien	52
60	Andocken: Geschwindigkeitsverlauf	52

1 Einführung

1.1 Fragestellung

Für lokale Navigationsaufgaben werden häufig reaktive dynamische Verhaltensweisen realisiert, welche ohne eine Modellierung der Umwelt auskommen und direkt auf Sensoreingaben mit einer Motorantwort reagieren können. Durch den Verzicht auf eine vorausschauende Planung (wie sie bei deliberativen Systemen durchgeführt wird) ist ein solches System in der Lage, auf unvorhergesehene Veränderungen in der Umwelt zu reagieren.

In Anlehnung an die von Braitenberg vorgestellten Verhaltensweisen, welche sich anhand einfacher Stimulus-Antwort-Regeln bewerkstelligen lassen, sollen in dieser Arbeit lokale Navigationsaufgaben entwickelt werden. Braitenberg sah seine Vehikel als ein Beispiel dafür, dass scheinbar komplexe Verhaltensweisen durch einfache Reiz-Reaktions-Schemata erzeugt werden können (siehe hierzu [Braitenberg, 1993]) In dieser Arbeit soll als einzige Eingabe das durch die Eigenbewegung induzierte Bewegungsfeld (optisches Flussfeld) der Umwelt ausgewertet werden.

Von Interesse ist die Frage, wie sensibel ein optisches System, welches aus aufeinanderfolgenden Bildern das Flussfeld der Umwelt berechnet, auf Einflüsse wie Kontrastveränderungen reagiert. Es soll geprüft werden, ob die quantitative Genauigkeit der Flusswerte, welche anhand eines differentiellen Algorithmus berechnet werden, ausreicht, um sowohl eine Geschwindigkeits- als auch eine Rotationsregulation eines Roboters in ausreichendem Maße zu bewerkstelligen.

Von weiterem Interesse ist die Anwendung eines geeigneten probabilistischen Schätzers, welcher aus einer Fülle von Flüssen den zur Berechnung heranzuziehenden Erwartungswert auswählen muss, um eine Robustheit des Verfahrens gegenüber starken Ausreißern zu gewährleisten.

1.2 Biologische Motivation

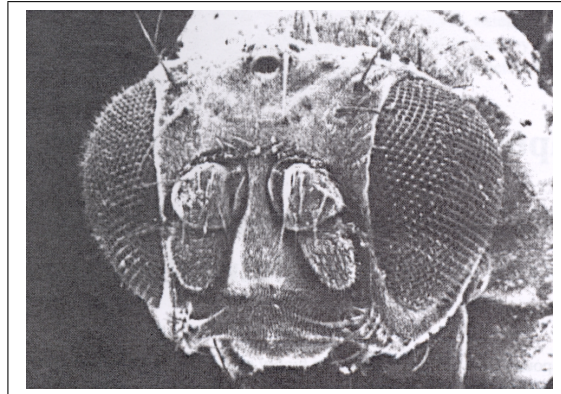


Abbildung 1: Kopf der Fliege *Drosophila Melanogaster*. (Aus [Neumann, 2004])

1.2.1 Aufbau des Insektenauges

Das Komplexauge (Abb. 1) setzt sich je nach Insektenart aus einigen tausend Einzelaugen (Ommatidien) zusammen. So besteht zum Beispiel das Auge der Libelle aus 10.000, das der *Drosophila* aus knapp eintausend Ommatidien. Bei Insekten, welche sich nicht in der Luft fortbewegen, ist diese Zahl reduziert. Die Ameise *Ponera punctatissima* besitzt als Extremfall auf jeder Seite nur ein einziges Ommatidium.

Das einzelne Ommatidium (Abb. 2) besitzt dabei eine eigene Hornhaut und Linse und ist so ausgerichtet, dass das Licht aus einer bestimmten Richtung des visuellen Feldes aufgenommen wird. Jedes Ommatidium eines Komplexauges enthält je nach Spezies sechs oder sieben photosensible Zellen, die Rhabdomere.

In Appositionsaugen sind diese Rhabdomere zu einem Rhabdom fusioniert, welches einen Lichtleiter auf der zentralen Achse des Ommatidiums bildet. Solch ein Ommatidium registriert Lichtreize aus einer charakteristischen Blickrichtung.

Ein Spezialfall ist das neurale Superpositionsauge, in welchem die Rhabdomere der einzelnen Ommatidien räumlich getrennt voneinander sind und als eigenständige Lichtleiter wirken können. Durch diesen Aufbau können Photonen aus unterschiedlichen Raumrichtungen in einem Ommatidium unterschiedliche Rhabdomere innervieren.

Damit die Rhabdomere nur über die Linse aktiviert werden, ist jedes einzelne von Pigmentzellen umhüllt, welche seitlich einfallendes Licht abhalten.

Beim Zusammensetzen des Gesamtbildes entsteht somit ein grobes *Mosaik*, welches eine Winkelauflösung von 1 bis 2 Grad aufweist (das menschliche Auge hat im Vergleich eine Auflösung von circa 1/100 Grad). Die hohe

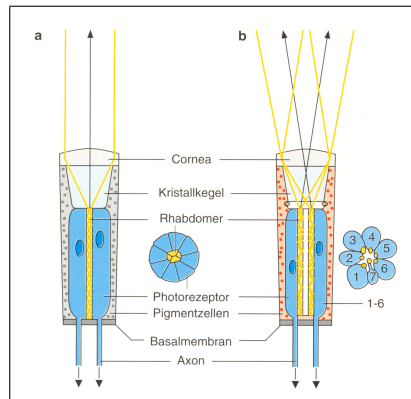


Abbildung 2: Aufbau eines Ommatidiums im Längs- und Querschnitt **(a)** Appositionsauge: Das Ommatidium registriert Licht aus einer Raumrichtung, die Rhabdomere bilden ein fusioniertes Rhabdom. **(b)** Neutrales Superpositionsauge: Durch die räumliche Trennung der Rhabdomere agiert jedes einzelne als separater Lichtleiter. Die Photorezeptoren innerhalb eines Ommatidium reagieren somit auf Lichtstrahlen unterschiedlicher Raumrichtungen. (Aus [Dudel et al., 1996])

zeitliche Auflösung kommt dadurch zustande, dass sich die einzelnen Photorezeptoren nach Lichtreizen äußerst schnell regenerieren und somit weitere Lichtreize verarbeiten können. Im Gegensatz zum Menschen, welcher bei optimalen Bedingungen 60 bis 65 Bilder pro Sekunden wahrnehmen kann, ist das Insektenauge dadurch in der Lage, bis zu 330 Reize pro Sekunde zu erkennen (Zur Biologie des Komplexauges siehe [Campbell, 2003], [Chapman, 1998] und [Dudel et al., 1996]). Die durch die Lichtreize erzeugten Erregungen der Photorezeptoren werden von deren Axonen zu den optischen Ganglien geleitet welche als Lamina, Medulla und Lobulakomplex bezeichnet werden, von denen Lamina und Medulla direkt von den Photorezeptoren erregt werden (Abb. 3).

1.2.2 Neuronale Verschaltung auf höherer Ebene

Ein besonderer Zelltyp im Lobulakomplex, welcher sich bei Dipteren aus Lobula und Lobulaplatte zusammensetzt, wurde von Krapp und Hengstenberg beschrieben. Sie entdeckten Neurone, welche spezifisch auf globale Bewegungsmuster reagieren. In ihren Experimenten mit der Fliege *Calliphora* fanden Krapp und Hengstenberg eine Zelle welche bei Translationen der Fliege maximal ansprach und 10 Zellen, die auf Rotationen reagierten.

Diese Zellen, welche sich in der Lobula-Platte befinden, erhalten weit verteilte retinotopische Signale von Nervenzellen in der Medulla, welche lokal auf bestimmte Bewegungsreize ansprechen. Anhand dieser ausgedehnten Integration können die Zellen Translationen und Rotationen unterscheiden, was bei EMDs nicht möglich ist. Anhand Deletion dieser Zellen konnte ge-

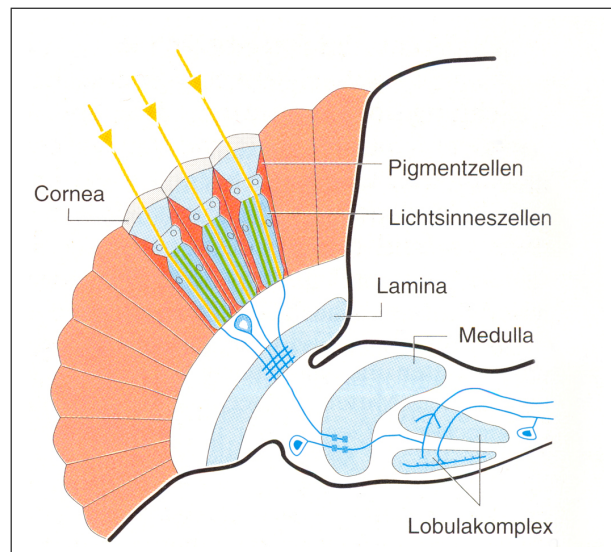


Abbildung 3: Die optischen Ganglien einer Fliege. Da es sich um ein neurales Superpositionsauge handelt, wird eine Nervenzelle in der Lamina durch benachbarte Ommatidien erregt, welche auf parallel einfallende Lichtreize reagieren. (Aus [Dudel et al., 1996])

zeigt werden, dass diese Neurone an der Flugsteuerung und Blickstabilisierung beteiligt sind (siehe hierzu [Krapp and Hengstenberg, 1996]). Sowohl die Bewegungsdetektoren für Translation als auch für Rotation sind an die Umwelt, welche von der Fliege während des Flugs wahrgenommen wird, angepasst. So reagieren die Translationsdetektoren besonders stark bei Bewegungen im ventralen Bereich, während die Rotationsdetektoren dort nur sehr schwach ansprechen und bevorzugt Bewegungen am und über dem Horizont wahrnehmen. Anhand dieser unterschiedlichen rezeptiven Felder werden Interferenzen der unterschiedlichen Detektoren vermieden und eine effektivere Klassifizierung gewährleistet.

1.2.3 Experimente mit Insekten

Es wurden in der Vergangenheit verschiedene Experimente durchgeführt, welche die Tiefenschätzung bei Insekten erforschten. Zwei Experimente sollen hier kurz genannt werden, wobei das zweite direkt in die Aufgabenstellung miteinfließt.

1.2.3.1 Entfernungsmessung der Heuschrecke

Heuschrecken messen die Entfernung zu Objekten indem sie ihren Kopf fünf bis zehn Millimeter horizontal hin- und herschwingen um eine entfernungsabhängige Bewegung auf der Retina zu erhalten. In einem Experiment von Sobel wurde ein Objekt synchron zu der Spähbewegung der Heuschrecke

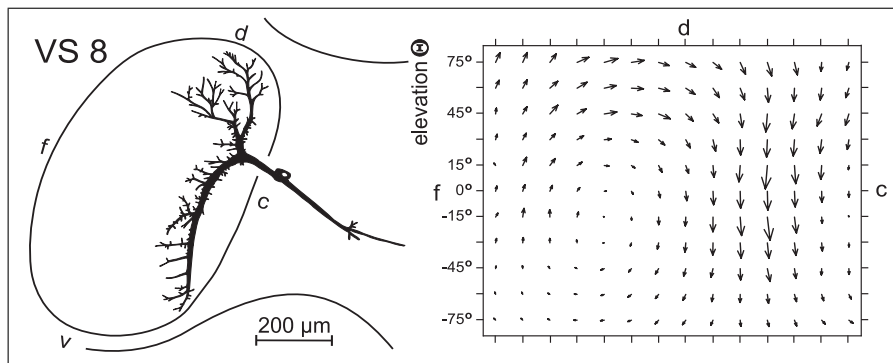


Abbildung 4: Bewegungsantworten des visuellen Neurons VS8: Die linke Seite zeigt die Anatomie der Zelle, die Karte auf der rechten Seite repräsentiert die rechte Hemisphäre des Blickfeldes. Die Ausrichtung der Pfeile spiegelt die präferierte Richtung wider, die Länge stellt die Bewegungssensibilität an der jeweiligen Stelle dar. Man erkennt, dass VS8 die stärkste Erregung aufweist, wenn eine Rotation um das Zentrum bei $\Psi = 45^\circ$ und $\Theta = -15^\circ$ stattfindet. (Aus [Krapp and Hengstenberg, 1996])

Schistocerca americana bewegt. Das Ergebnis war, dass die Entfernung zu einem in Phase oszillierenden Objekt unterschätzt wurde wohingegen Objekte, welche sich phasenverschoben bewegten, als zu weit entfernt eingeschätzt wurden. Die Einschätzung der Entfernung wurde hierbei anhand der Sprunggeschwindigkeit gemessen (siehe [Sobel, 1990] und [Wallace, 1959]).

1.2.3.2 Korridorflug

In [Srinivasan et al., 1999] wurde untersucht, anhand welcher Merkmale sich Bienen in einem Gang bewegen während sie diesen durchqueren. Srinivasan et al. gingen dabei von der Hypothese aus, dass die Biene die Bewegungen, welche sie mit ihren beiden Augen registriert, durch zentrales Ausrichten im Gang ausbalanciert. Um dies zu testen, ließen sie Bienen durch einen Korridor fliegen, dessen Wände mit vertikalen Streifen verkleidet waren (siehe Abb. 5).

Die Bienen neigten dazu bei stationären Streifenmuster in der Mitte des Korridors und somit in gleichem Abstand zu den beiden Wänden zu fliegen. Wurde jedoch eines der beiden Muster während der Durchquerung in Flugrichtung verschoben ohne damit jedoch den räumlichen Aufbau des Korridors zu beeinflussen, so änderten die Bienen ihr Verhalten insofern, als dass sie sich näher an dem sich bewegenden Muster entlangbewegten. Umgekehrt wandten sie sich von ihnen entgegenkommenden Mustern ab und bevorzugten die Wand mit stationärem Muster.

Diese Ergebnisse führten zu dem Schluss, dass die Biene tatsächlich die gemessene Bewegung auf den beiden retinalen Bildern balanciert. Variationen der Periodizität der Streifenmuster resultierten nicht in verändertem

Verhalten, welches darauf hindeutet, dass die eigentliche Struktur der Oberfläche nicht von Bedeutung ist.

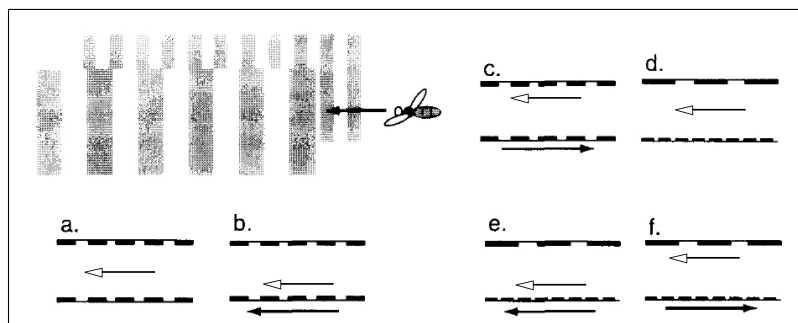


Abbildung 5: Verhalten der Biene im Korridor: Die Pfeile innerhalb des Ganges stellen die Bewegung der Biene dar, die Pfeile außerhalb die Verschiebung der Wandtextur. (a) Zentrierung bei unbewegten Mustern. (b) Annäherung an das in Flugrichtung bewegte Muster. (c) Abwenden von dem gegen Flugrichtung bewegten Muster. (d),(e),(f) Wiederholung mit Mustern unterschiedlicher Periodizität, welches keine Veränderung im Verhalten der Bienen bewirkt. (Verändert nach [Srinivasan et al., 1999])

1.3 Omnidirektionale Kamerasysteme

Um Bilder zu erhalten, welche in horizontaler Richtung ein möglichst großes Sichtfeld abdecken und somit dem Insektenmodell möglichst nahe kommen, sind unterschiedliche Ansätze möglich wie zum Beispiel Weitwinkelobjektive, rotierende Kameras und Systeme, welche sich aus Linsen und Spiegeln zusammensetzen (so genannte katadioptrische Systeme). In dieser Arbeit werden wir ein katadioptrisches System verwenden, welches in Abschnitt 2.1 näher beschrieben wird.

Abbildung 6 zeigt eine Spiegelkonstruktion, wie sie 1996 von Weber et al. entworfen wurde. Hierbei wurden über diverse Spiegel je zwei lateral und frontal aufgenommene Bilder von einer Kamera erfasst und zu einem Bild fusioniert.

Neuere Ansätze verwenden anstelle von solch aufwendigen Konstruktionen Spiegel, welche der aufnehmenden Kamera ein vollständiges 360°-Panorama-Bild ermöglichen. In Abbildung 7 sind der benutzte kegelförmige Spiegel und die darunter montierte Kamera zu sehen.

Nachteil des Spiegel-Systems sind Abbildungsfehler, da bei parallel einfallenden Strahlen nur vertikal versetzte Strahlen parallel reflektiert werden. Horizontal versetzte Strahlen werden dagegen unterschiedlich stark gestreut, wodurch ein unscharfes Bild entsteht (siehe auch [Stürzl, 2004]). Diese Abbildungsfehler erschweren diffizile Aufgaben wie die Erkennung von Objekten zwecks deren Manipulation. Vorteil im Vergleich zu rotierenden Systemen

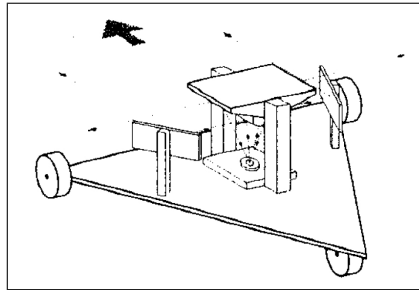


Abbildung 6: Katadioptrisches System, in welchem die CCD-Kamera ein Bild aufnimmt, welches sich aus je zwei frontalen und lateralen Bildern anhand diverser Spiegel zusammensetzt. (Aus [Weber et al., 1996])

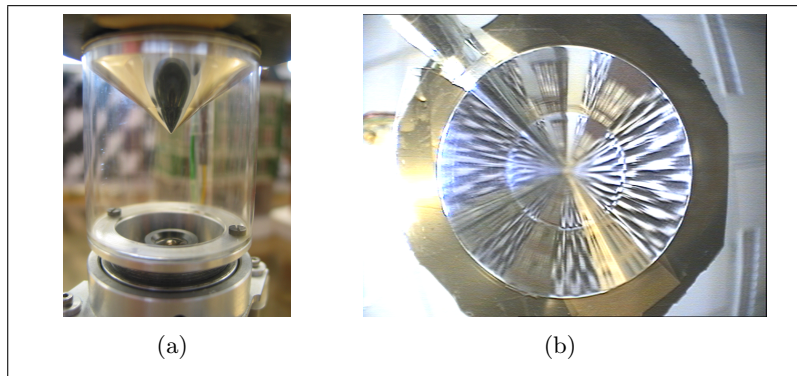


Abbildung 7: Katadioptrisches System des Khepera-Roboters: **(a)** Der kegelförmige Spiegel wird von der darunter montierten Kamera gefilmt. **(b)** Ein Panorama-Bild, wie es von der Kamera aufgenommen wird.

ist die Abwesenheit von sich bewegenden Bauteilen, sowie die Vernachlässigung einer komplexen Bildkontrolle, wie sie bei sich bewegenden Systemen notwendig ist.

1.4 Der optische Fluss

1.4.1 Das zweidimensionale Bewegungsfeld \vec{v}

Ausgangspunkt ist ein dreidimensionales Bewegungsfeld der Umwelt, welches entweder durch Eigenbewegung des Beobachters oder durch unabhängige Bewegungen innerhalb der Umwelt entsteht und somit die Relativbewegung zwischen dem Agenten und der Umwelt beschreibt. Das Bewegungsfeld wird als Funktion $\vec{w}(\vec{a})$ beschrieben, wobei $\vec{a} = (a_x, a_y, a_z)^T$ ein beliebiger Punkt im Raum ist. Wir gehen im folgenden immer von statischen Umgebungen aus, so dass $\vec{w}(\vec{a})$ ausschließlich durch Eigenbewegungen des Beobachters erzeugt wird. Die Projektion der an den sichtbaren Oberflächen der Umwelt ansetzenden Bewegungsvektoren auf die Bildebene stellt dann das

zweidimensionale Bewegungsfeld $\vec{v}(\vec{a}')$ dar, wobei $\vec{a}' = (a'_x, a'_y)^T$ die Projektion des dreidimensionalen Punktes \vec{a} auf die Bildebene ist (Abb. 8).

In den folgenden Formeln gehen wir bei der Projektion von einer Lochkamera mit einer Brennweite der Länge 1 aus, so dass sich der projizierte Punkt \vec{a}' auf der Bildebene durch folgende Formel ergibt:

$$\vec{a}' = -\frac{1}{a_z} \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad (1)$$

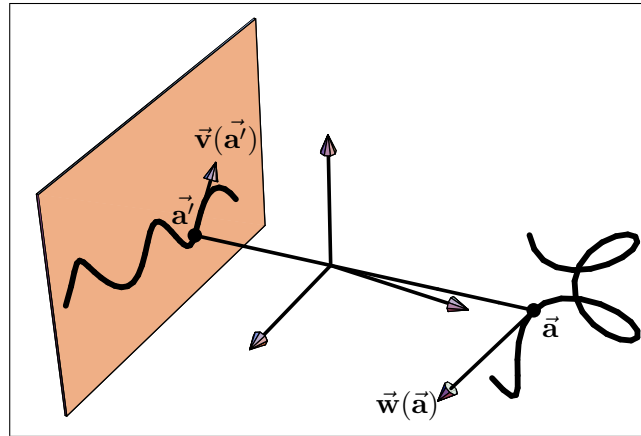


Abbildung 8: Projektion von Bewegungen auf die Bildebene. Der Punkt \vec{a}' auf der Bildebene entsteht durch Projektion des Punktes \vec{a} . Die Bewegung $\vec{v}(\vec{a}')$ stellt die Projektion der Bewegung $\vec{w}(\vec{a})$ dar. (Verändert nach [Mallot and Allen, 2000])

Die zweidimensionale Bewegung \vec{v} , welche an dem Punkt \vec{a}' ansetzt, kann durch die Ableitung der projizierten Bahnkurve:

$$\vec{a}'(t) = -\frac{1}{a_z(t)} \begin{pmatrix} a_x(t) \\ a_y(t) \end{pmatrix} \quad (2)$$

berechnet werden, welche eine Parametrisierung nach der Zeit der Gleichung 1 darstellt. Durch Anwenden der Quotientenregel auf Gleichung 2 erhalten wir für die x -Komponente:

$$\frac{d}{dt} a'_x(t) = \frac{d}{dt} \left(-\frac{a_x(t)}{a_z(t)} \right) = -\frac{1}{a_z(t)} \left(\frac{da_x(t)}{dt} + a'_x \frac{da_z(t)}{dt} \right) \quad (3)$$

und analog für die y -Komponente:

$$\frac{d}{dt} a'_y(t) = -\frac{1}{a_z(t)} \left(\frac{da_y(t)}{dt} + a'_y \frac{da_z(t)}{dt} \right) \quad (4)$$

Ersetzt man nun in diesen beiden Gleichungen $\frac{d}{dt} a_n(t)$ durch w_n , so erhält man die folgende Formel für das zweidimensionale Bewegungsfeld \vec{v} :

$$\vec{v}(a'_x, a'_y) = \left(\frac{d}{dt} a'_x(t), \frac{d}{dt} a'_y(t) \right)^T = -\frac{1}{a_z} \left(\begin{pmatrix} w_x \\ w_y \end{pmatrix} + w_z \begin{pmatrix} a'_x \\ a'_y \end{pmatrix} \right) \quad (5)$$

Man sieht an dieser Gleichung, dass Bewegungen in der Umwelt immer nur bis auf einen konstanten Faktor genau angegeben werden können, da die Multiplikation von \vec{a} und \vec{w} den 2D-Bewegungsvektor \vec{v} nicht verändert. Das heißt, ein Objekt O_1 mit Relativgeschwindigkeit \vec{w}_{O_1} erzeugt den gleichen Vektor wie ein n -mal entferntes Objekt O_2 , für dessen Relativgeschwindigkeit $\vec{w}_{O_2} = n \cdot \vec{w}_{O_1}$ gilt.

1.4.2 Der optische Fluss als messbarer Teil des Bewegungsfeldes

Das zweidimensionale Bewegungsfeld $\vec{v}(x, y)$ stellt jedoch ein rein theoretisches Konstrukt dar. Genau genommen arbeitet man in der Praxis mit dem optischen Fluss $\vec{f}(x, y)$, welcher nur die messbaren Anteile der Bewegungen berücksichtigt. Einfluss auf den optischen Fluss haben somit nur Bildbereiche, welche ausreichende Bildstruktur besitzen.

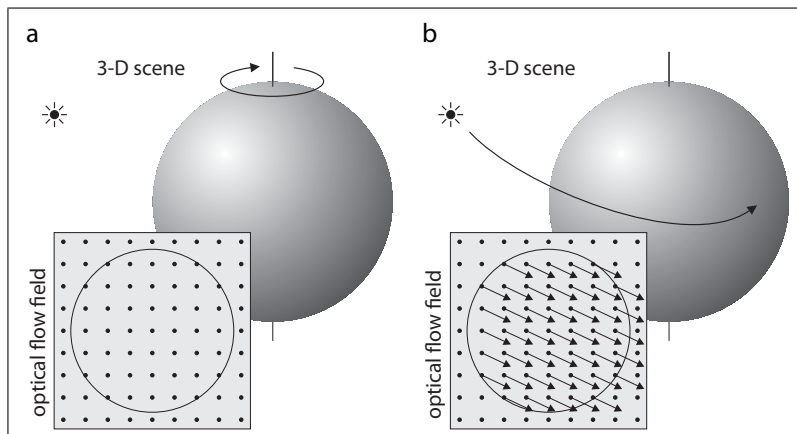


Abbildung 9: Auswirkung einer sich bewegenden Lichtquelle, welche eine ideal glatte Kugel beleuchtet: (a) Eine sich drehende Kugel mit statischer Lichtquelle führt zu keinem optischen Fluss wohingegen (b) eine statische Kugel, welche von einer sich bewegenden Lichtquelle beleuchtet wird, eine stattfindende Bewegung suggeriert. (Aus [Jähne et al., 1999])

Bildrauschen und sich verändernde Lichtquellen können scheinbare Bewegungen im Bild erzeugen, wie in Abbildung 9 zu sehen. In unseren Experimenten werden wir daher sich bewegenden gerichtete Lichtquellen ausschließen.

Im folgenden Abschnitt werden wir von idealen Sensoren ausgehen und den Begriff des optischen Flusses \vec{f} synonym zum zweidimensionalen Bewegungsfeld \vec{v} benutzen.

1.4.3 Optischer Fluss durch Eigenbewegungen

Eine bedeutende Eigenschaft des optischen Flusses ist die Tatsache, dass sich bei bekannter Eigenbewegung *Tiefeninformationen* über die Umwelt

aus dem Fluss ableiten lassen. Eigenbewegungen eines Agenten setzen sich aus zwei sich additiv überlagernden Komponenten zusammen, der Rotation um die Achse $\vec{\mathbf{r}}$ mit der Winkelgeschwindigkeit ω und der Translation in Richtung des Einheitsvektors $\vec{\mathbf{u}}$ mit der Geschwindigkeit v . Das dreidimensionale Bewegungsfeld in statischen Umgebungen lässt sich daher wie folgt beschreiben:

$$\vec{\mathbf{w}}(\vec{\mathbf{a}}) = -v\vec{\mathbf{u}} - \omega\vec{\mathbf{r}} \times \vec{\mathbf{a}} \quad (6)$$

Man sieht hier, dass das durch Translation verursachte Bewegungsfeld $-v\vec{\mathbf{u}}$ unabhängig von der Position des Punktes in der Umwelt ist wohingegen der Rotationsfluss $-\omega\vec{\mathbf{r}} \times \vec{\mathbf{a}}$ das Kreuzprodukt der Rotationsachse und dem Raumpunkt darstellt und somit von der Lage des Raumpunktes und dessen Entfernung zum Beobachter abhängt. Die Flussfelder, welche durch die beiden beschriebenen Bewegungsarten induziert werden, werden im folgenden genauer betrachtet (siehe auch [Mallot and Allen, 2000]).

1.4.3.1 Rotationsfluss

Die Bewegung eines Punktes $\vec{\mathbf{a}}$ in der Umwelt bei einer reinen Rotation des Beobachters ist nach Gleichung 6:

$$\vec{\mathbf{w}}(a_x, a_y, a_z) = -\omega\vec{\mathbf{r}} \times \vec{\mathbf{a}} = -\omega \begin{pmatrix} r_y a_z - r_z a_y \\ r_z a_x - r_x a_z \\ r_x a_y - r_y a_x \end{pmatrix} \quad (7)$$

Wird dieser Vektor nach Gleichung 5 auf die Bildebene projiziert ergibt sich der optische Fluss:

$$\vec{\mathbf{f}}(a'_x, a'_y) = \frac{\omega}{a_z} \left(\begin{pmatrix} r_y a_z - r_z a_y \\ r_z a_x - r_x a_z \end{pmatrix} + \begin{pmatrix} r_x a_y - r_y a_x \end{pmatrix} \begin{pmatrix} a'_x \\ a'_y \end{pmatrix} \right) \quad (8)$$

Die Koordinaten der Umwelt kommen hier nur als Quotienten der Form $\frac{a_x}{a_z}$ und $\frac{a_y}{a_z}$ vor. Dies bedeutet, dass bei einer Rotation sämtliche Punkte unterschiedlicher Distanz, welche in der Umwelt auf dem gleichen Sehstrahl liegen, den gleichen Fluss erzeugen. Aus Rotationsflüssen können somit keine Tiefeninformationen über die Umwelt erschlossen werden. Noch deutlicher wird dies, wenn wir anhand der Projektionsformel 1 diese Brüche durch die negierten Bildkoordinaten ersetzen:

$$\vec{\mathbf{f}}(a'_x, a'_y) = \omega \left(-r_x \begin{pmatrix} a'_x a'_y \\ 1 + a'^2_y \end{pmatrix} + r_y \begin{pmatrix} 1 + a'^2_x \\ a'_x a'_y \end{pmatrix} + r_z \begin{pmatrix} a'_y \\ -a'_x \end{pmatrix} \right) \quad (9)$$

In Abbildung 10 ist das resultierende Flussfeld dargestellt.

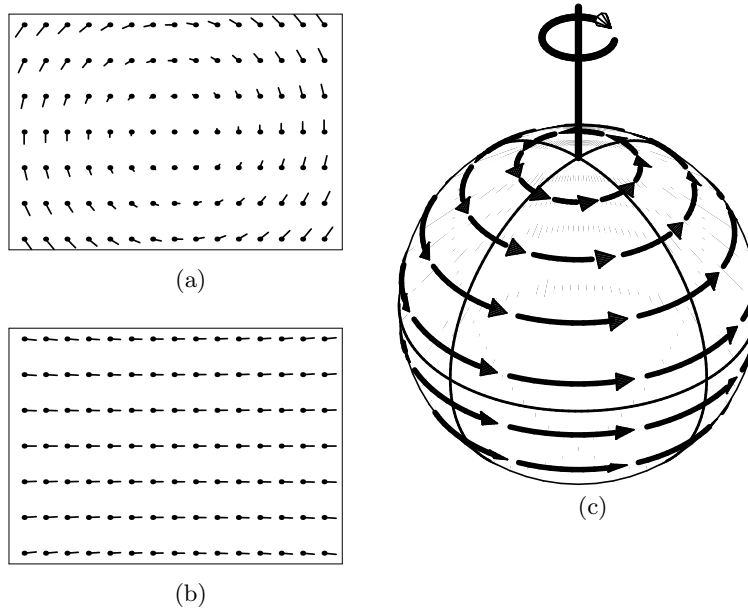


Abbildung 10: Flussfeld, welches durch eine reine Rotation verursacht wird. Die einzelnen Flüsse verlaufen in Richtung der Punkte. **(a)** Ein Ausschnitt des Flussfeldes, welches bei Rotation des Beobachters um seine Blickrichtung entsteht. **(b)** Ein Ausschnitt des Flussfeldes, welches bei Rotation um die Hochachse des Beobachters entsteht. **(c)** Die Projektion der Bewegung auf eine Kugeloberfläche erzeugt eine Reihe von Breitenkreisen, anhand derer man den Fluss in jede beliebige Blickrichtung berechnen kann. (Verändert nach [Mallot and Allen, 2000])

1.4.3.2 Translationsfluss

Das Bewegungsfeld der Umwelt bei reiner Translation \vec{u} des Beobachters berechnet sich nach:

$$\vec{w}(a_x, a_y, a_z) = -v\vec{u} = -v \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} \quad (10)$$

Der Fluss auf der Bildebene ist nach Gleichung 5:

$$\vec{f}(a'_x, a'_y) = \frac{v}{a_z} \left(\begin{pmatrix} u_x \\ u_y \end{pmatrix} + u_z \begin{pmatrix} a'_x \\ a'_y \end{pmatrix} \right) \quad (11)$$

Da der Kehrwert der z -Komponente des Punktes \vec{a} in die Projektion eingeht, ist der Fluss umso größer, je näher der Punkt dem Beobachter ist. Tiefeninformationen können somit nur aus dem Translationsteil des Flusses gezogen werden. Dies macht sich im Alltag insofern bemerkbar, als dass bei einer Translationsbewegung naheliegende Objekte schneller an dem Beobachter vorbeiziehen als weit entfernt liegende.

Dieser Tatsache bedienen sich sämtliche später vorgestellten Algorithmen zur Berechnung der Entfernung von Hindernissen.

In Abbildung 11 sieht man den durch eine Translation verursachten Fluss, welcher sich vom *Expansionspunkt* vor dem Agenten in Kurven zum *Kontraktionspunkt* hinter dem Agenten erstreckt.

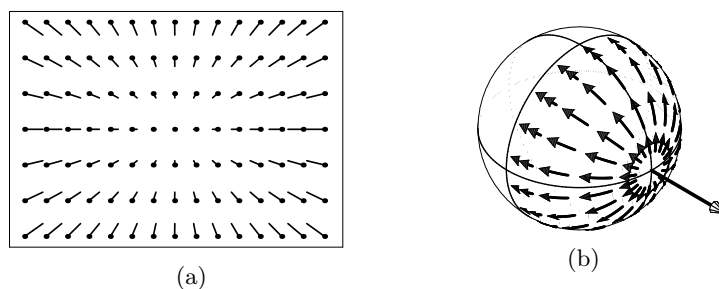


Abbildung 11: Flussfeld, welches durch reine Translation verursacht wird. Die einzelnen Flüsse verlaufen in Richtung der Punkte. (a) Translation frontal auf eine Wand zu. (b) Projektion des Flussfeldes auf eine Kugeloberfläche. Die Flusslinien sind Abschnitte von Großkreisen, welche durch den Expansions- und Kontraktionspunkt verlaufen. (Verändert nach [Mallot and Allen, 2000])

1.4.4 Methoden der Flussmessung

1.4.4.1 Korrelationsdetektoren

Diese Ansätze versuchen, die beste Übereinstimmung eines Bildmerkmals und dem korrespondierenden Merkmal im folgenden Frame zu ermitteln ([Jähne et al., 1999], [Miles and Wallman, 1992], [Mallot and Allen, 2000]). Sie können weiter in merkmals- und regionsbasierte Verfahren unterteilt werden. Der Fluss \vec{f} ermittelt sich dann durch $\vec{f} = \frac{\vec{s}(\vec{x})}{t_2 - t_1}$, wobei s die Verschiebung ist, welche die beste Übereinstimmung in den beiden Bereichen $g(\vec{x}, t_1)$ und $g(\vec{x} - \vec{s}, t_2)$ liefert. Die beste Verschiebung kann anhand einer Ähnlichkeitsmaximierung oder Distanzminimierung ermittelt werden. Als Beispiel sei der Reichardt-Detektor angeführt, einer der ersten Korrelationsfilter welcher für Erklärungen von Bewegungsdetektoren in verschiedenen Vertebraten herangezogen wird (siehe [Hassenstein and Reichardt, 1956] und [van Santen and Sperling, 1985]). Dieser Detektor besteht aus zwei spiegelsymmetrischen Untereinheiten, welche jeweils die beiden Signale multiplizieren, nachdem jeweils eines zeitverzögert wurde. Die Ausgaben dieser beiden Untereinheiten werden schließlich voneinander subtrahiert. Der Detektor springt auf Bewegungen in eine vorgegebene Raumrichtung an, wobei das Vorzeichen der Ausgabe die Orientierung der Bewegung angibt.

1.4.4.2 Ableitungsdetektoren

Diese Detektoren bringen die räumliche Helligkeitsveränderung mit der zeitlichen Veränderung an einer bestimmten Stelle im Bild in Verbindung. Eine wesentliche Annahme bei der Berechnung des optischen Flusses mit Ableitungsdetektoren ist, dass sich die Helligkeit E an einem Punkt $\vec{a}' = (a'_x, a'_y)$ nur aufgrund von Bewegungen verändert (*brightness change constraint*) [Jähne et al., 1999]. Formal bedeutet dies, dass die Ableitung des vollständigen Differentials der Funktion $E(a'_x, a'_y, t)$ nach der Zeit gleich Null sein muss:

$$\frac{dE}{dt} = \frac{\partial E}{\partial a'_x} \frac{da'_x}{dt} + \frac{\partial E}{\partial a'_y} \frac{da'_y}{dt} + \frac{\partial E}{\partial t} = 0 \quad (12)$$

Gleichung 12 lässt sich mit $E_t = \frac{\partial E}{\partial t}$ wie folgt schreiben:

$$(\nabla E)^T \vec{f} + E_t = 0 \quad (13)$$

$\vec{f} = (f_x, f_y)$ beschreibt hierbei den optischen Fluss mit $f_x = \frac{d}{dt}a'_x$ und $f_y = \frac{d}{dt}a'_y$; ∇E den Bildgradienten. Da diese Gleichung jedoch mit zwei Unbekannten unterbestimmt ist, erwächst aus ihr direkt das Aperturproblem für Bewegungsschätzungen.

Der in dieser Arbeit angewandte Lucas-Kanade-Algorithmus ist ein solcher Ableitungsdetektor. Ausschlaggebender Vorteil dieser Art von Detektoren ist für diese Arbeit deren hohe Geschwindigkeit, welche für die Online-Berechnungen notwendig ist und die hohe Robustheit gegenüber Ausreißern aufgrund geeigneter Konfidenzmaße (siehe auch Abschnitt 3.1.2). Für einen ausführlichen Vergleich verschiedener Detektoren siehe [Barron et al., 1994].

1.4.4.3 Bewegung zweiter Ordnung

Im Gegensatz zu Bewegungen, welche durch eine Verschiebung eines Bildbereichs entstehen, gibt es so genannte Bewegungen zweiter Ordnung, welche nicht durch die bisher beschriebenen Verfahren detektiert werden können (Abb. 12). Detektoren für diese Arten von Bewegungen müssen auf der Ableitung nach der Zeit arbeiten, um auf diese Bewegungen anzusprechen ([Mallot and Allen, 2000]).

1.4.5 Aperturproblem

Das Aperturproblem sagt aus, dass nur diejenige Flusskomponente berechnet werden kann, welche senkrecht zu der in Gleichung 13 definierten Linie auftritt. So können verschiedene Bewegungen, wie sie in Abbildung 13 zu sehen sind, zu dem gleichen optischen Fluss führen. Das generalisierte Aperturproblem bezeichnet den Konflikt zwischen dem einfachen Aperturproblem, welches eine größere Nachbarschaft um den fokussierten Punkt erzwingt um unterschiedliche räumliche Strukturen erfassen zu können und der Tatsache, dass ein Vergrößern der Nachbarschaft ein *Verwischen* von

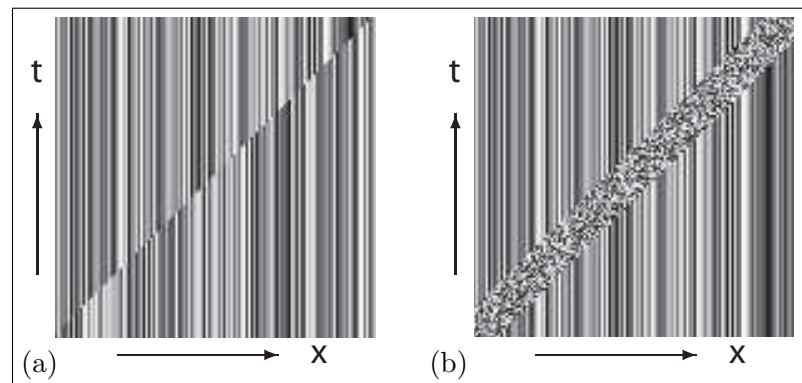


Abbildung 12: Orts-Zeitdiagramme von Bewegungen zweiter Ordnung. **(a)** Prozess, welcher jeweils ein Pixel in einem statischen Rauschen verändert während er sich in x -Richtung fortbewegt. **(b)** Ein Fenster, welches dynamisches Rauschen enthält, bewegt sich vor dem Hintergrund eines statischen Rauschens. (Aus [Mallot and Allen, 2000])

unterschiedlichen Bewegungen bewirken kann. Im Falle der hier gestellten Aufgaben beschäftigt uns jedoch nur das einfache Aperturproblem, da wir es hauptsächlich mit kontinuierlichen Korridor-Wänden zu tun haben. Die Qualität der Nachbarschaft lässt sich anschaulich anhand der Krümmung der Bildstruktur festmachen, welche direkt detektierbar ist.

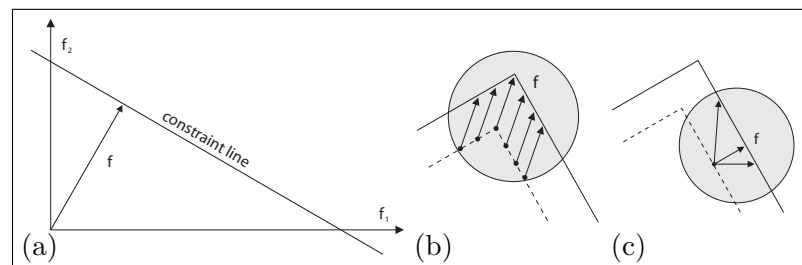


Abbildung 13: Darstellung des Aperturproblems: **(a)** Dargestellt ist die in Gleichung 13 definierte Gerade (*constraint line*). Der normale Fluss f steht senkrecht zu dieser Linie. **(b)** Es ist kein Aperturproblem vorhanden, da in der lokalen Nachbarschaft unterschiedliche Strukturen auftreten und der Fluss eindeutig bestimmt werden kann. **(c)** Da innerhalb der lokalen Nachbarschaft sämtliche Gradienten parallel sind, kann nur der normale Fluss gemessen werden, es liegt das Aperturproblem vor. (Aus [Jähne et al., 1999])

1.5 Literatur

Eine Einleitung zum optischen Fluss findet sich in [Mallot and Allen, 2000], Kap. 7. Für weitere Informationen wurde [Jähne et al., 1999] herangezogen, insbesondere Kapitel 13 des zweiten Bandes.

In [Lucas and Kanade, 1981] wird der verwendete Flussdetektor von Lucas und Kanade besprochen, welcher iterativ die Verschiebung zweier Bilder berechnet. Die angewandte pyramidale Implementierung (siehe hierzu auch [Bouguet, 2000]) der openCV arbeitet mit einer Grob-Fein-Strategie, welche eine effiziente Verarbeitung auch bei größeren Bildverschiebungen ermöglicht. Die openCV ist eine Programmbibliothek von Intel mit Algorithmen rund um die Bildbearbeitung. Ein Vergleich unterschiedlicher Techniken, anhand derer sich der optische Fluss berechnen lässt, findet sich in [Barron et al., 1994]. Verschiedene Konfidenzmaße werden unter anderem in [Barron et al., 1994], [Jähne et al., 1999] und [Horn, 2003] beschrieben.

Korridorauflagen, für welche der optische Fluss genutzt wurde, werden in [Argyros et al., 2004], [Tsakiris and Argyros, 2000] und [Weber et al., 1996] beschrieben. Sämtliche Experimente in diesen Veröffentlichungen wurden ohne Rekonstruktion der Umwelt ausgeführt. Eine Auflistung verschiedener biomimetischer Navigationsaufgaben, welche mit unterschiedlichen Ansätzen gelöst wurden, findet sich in [Franz and Mallot, 2000].

1.6 Aufgabenstellung

Es sollen lokale Navigationsaufgaben realisiert werden, die auf der Berechnung des optischen Flusses aufbauen. Im speziellen soll eine Korridorzentrierung implementiert werden, welche dem Roboter eine kollisionsfreie Fahrt in sowohl geraden, in sich verengenden als auch in gekrümmten Gängen ermöglicht. In Abbildung 14 ist beispielhaft ein solcher Gang gezeigt.

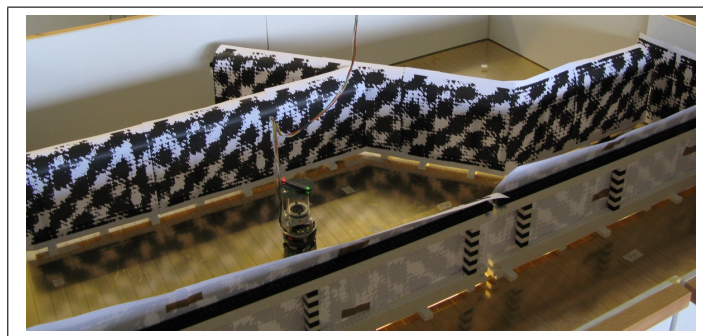


Abbildung 14: Beispielhafte Umgebung für eine lokale Navigationsaufgabe

Die Magnituden der gemessenen Flüsse sollen zusätzlich genutzt werden, um die Translationsgeschwindigkeit der Korridorbreite anzupassen. Hierbei

wird die Robustheit des Verfahrens in Abhängigkeit von unterschiedlichen Umwelteinflüssen und Parametereinstellungen untersucht, wobei vor allem die Berechnung des Flussfeldes anhand des Lucas-Kanade-Algorithmus' als auch die darauf aufbauende Schätzung von Interesse sind.

Desweiteren soll sich der Roboter in Anwesenheit von nur einer Wand entlang dieser bewegen können, um Lücken in Korridoren zu überbrücken. Letztendlich soll ein Andockverhalten implementiert werden, welches bei frontal in Fahrtrichtung liegenden Hindernissen dafür gedacht ist, den Roboter in unmittelbarer Nähe zu dem Hindernis anzuhalten.

1.7 Struktur des Dokuments

Kapitel 2 beschreibt den verwendeten Roboter, dessen Testumgebung und die sonstige verwendete Hardware. Kapitel 3 geht auf den Lucas-Kanade-Algorithmus ein, welcher für die Berechnung des optischen Flusses benutzt wird und erläutert die Algorithmen, welche für die verschiedenen Navigationsaufgaben verantwortlich sind. In Kapitel 4 werden die durchgeführten Experimente samt Ergebnissen beschrieben. Die Diskussion und die Schlussfolgerungen finden sich in Kapitel 5, der Ausblick in Kapitel 6.

2 Material

2.1 Khepera-Roboter

Benutzt wurde ein Khepera-Roboter der ersten Generation (Abb. 15) der Firma K-Team (www.k-team.com). Dieser Roboter besitzt zwei unabhängig ansteuerbare Räder an den Seiten seiner kreisförmigen Bodenplatte und zwei kleine Plastikstifte, welche jeweils vorne und hinten an der Unterseite angebracht sind um den Roboter zu stabilisieren. Der Roboter hat eine Maxi-



Abbildung 15: Khepera-Roboter

malgeschwindigkeit von 1 Meter pro Sekunde, wobei die Geschwindigkeit in Schritten von 8 Millimetern pro Sekunde geregelt werden kann. Die Kommunikation mit dem Roboter fand anhand des Kommunikationsprotokolls Version 4.04 über den seriellen Port des PCs statt. Das Einlesen der Bilddaten mit anschließender Flussberechnung und Motoransteuerung dauern auf dem verwendeten PC (siehe Abschnitt 2.3) ca. 0,07 Sekunden, wobei die genaue Zeit aufgrund der Flussberechnung um ca. 0,01 Sekunden schwanken kann.

Der Roboter besitzt einen kegelförmigen Panorama-Spiegel, welcher von einer Kamera mit einer Auflösung von 768 x 576 Pixel gefilmt wird. Dieser Spiegel wurde ursprünglich für panoramische Stereobilder konzipiert und besteht aus zwei aufeinander sitzenden Kegeln mit unterschiedlichen Öffnungswinkeln von $\gamma_1 = 97^\circ$ und $\gamma_2 = 109^\circ$ (Abb. 16). Wir machen jedoch von der speziellen Stereobauweise in dieser Arbeit keinen Gebrauch. Der Spiegel ist direkt über der Rotationsachse des Roboters angebracht, so dass eine Rotation des Roboters zu einer Rotation des aufgenommenen Kamerabildes führt.

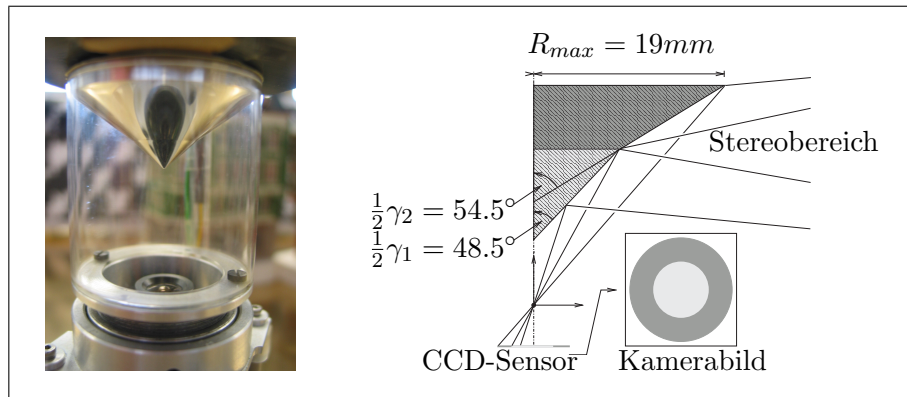


Abbildung 16: Panorama-Spiegel, welcher auf dem Khepera-Roboter montiert ist. Rechts unten erkennt man schematisch das Kamerabild, dessen innere Fläche von dem unteren, der darum liegende Ring von dem oberen Spiegel abgebildet wird. (Verändert nach [Stürzl, 2004])

Für unsere Aufgaben nutzten wir eine Auflösung von 384 x 288 Pixel, um die späteren Verarbeitungsschritte zu beschleunigen.

Der Roboter besitzt 8 Infrarotsensoren, welche wie in Abbildung 17 gezeigt angeordnet sind. Diese wurden benutzt, um den Roboter in der Testphase bei zu geringer Entfernung zu einem Hindernis anzuhalten, indem die durch die Sensoren realisierte Hinderniserkennung in einer Subsumptionsarchitektur unterhalb der zu testenden Navigationsaufgabe eingebunden wurde.

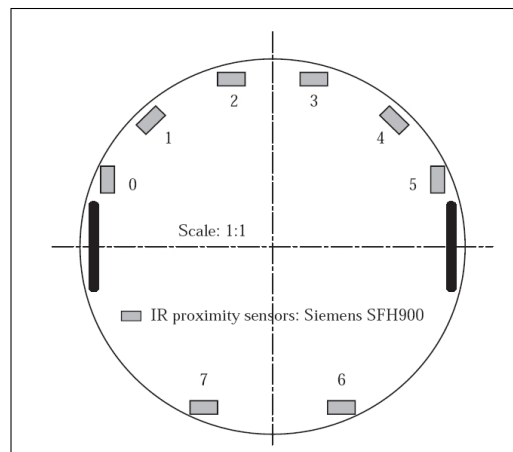


Abbildung 17: Anordnung der Infrarot-Sensoren. (Aus [K-Team, 1994])

2.2 Testumgebung

Der Roboter bewegte sich auf einer circa 140 mal 120 cm großen Ebene, wie sie in Abbildung 18 zu sehen ist. Die Kabel, welche sowohl den Roboter mit Strom und Steuersignalen versorgen, als auch die Videosignale zum Rechner leiten, wurden über die Decke zum Steuerrechner geleitet.

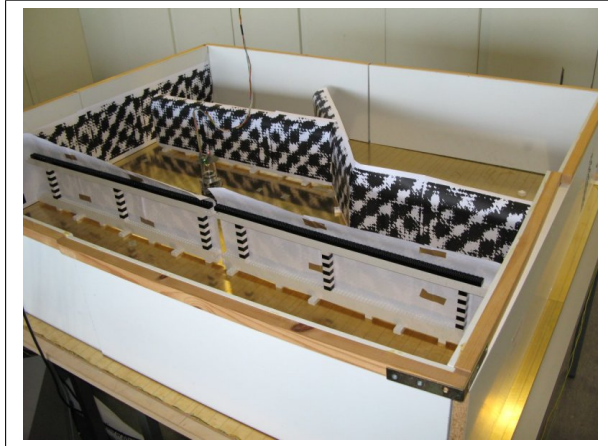


Abbildung 18: Aufbau der Testumgebung. Zu sehen sind der Khepera-Roboter, die Kabelführung und eine beispielhafte Korridorumgebung wie sie in Abschnitt 4.2 benutzt wird.

2.2.1 Tracking-System

Die Navigationsplattform, auf der die Experimente stattfinden, wird von einer Kamera mit einer Auflösung von 768 x 576 Pixel erfasst. Anhand dieser *Tracking-Kamera* kann zu jedem Zeitpunkt sowohl die Position als auch die Orientierung des Roboters berechnet werden, indem die Position der beiden Dioden, welche auf dem Roboter angebracht sind, berechnet werden. Abbildung 19 zeigt beispielhaft die Sicht der Tracking-Kamera. Die Ausleuchtung der Roboterumgebung findet anhand der normalen Deckenbeleuchtung statt. Um die beiden Dioden zu detektieren, werden die Pixel, deren Helligkeitswerte unter 200 liegt, auf 0 gesetzt. Danach wird das Bild mit einem Gaussfilter mit einer Standardabweichung von 3,5 Pixel gefaltet, um lokale Maxima im Zentrum der Dioden zu erhalten. Die Position des Roboters wird durch den Mittelwert der beiden detektierten Dioden dargestellt.

2.2.2 Wandtexturen

Um eine ausreichende Flussberechnung zu ermöglichen, wurden die Wände, zwischen denen sich der Roboter entlang bewegt, mit einer kontrastreichen Textur tapeziert. Für diese Textur wurden jeweils zehn Sinusverläufe in horizontaler und vertikaler Richtung und in Richtung der ersten und

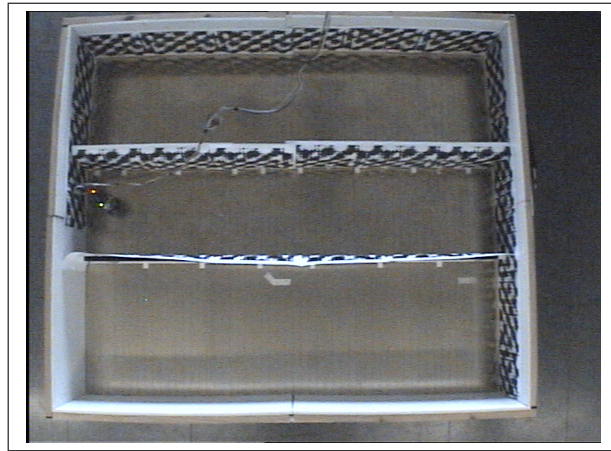


Abbildung 19: Testumgebung, aufgenommen von der Tracking-Kamera: Zu erkennen sind die beiden Dioden, welche auf dem Roboter angebracht sind.

zweiten Hauptdiagonalen erzeugt, deren Perioden zufällig aus dem Bereich $[10\pi, 200\pi]$ gewählt wurde. Nach der Durchschnittsbildung dieser resultierenden Grauverläufe unterschiedlicher Richtung und Periode wurde das Graubild in ein reines Schwarz-Weiß-Bild umgewandelt (Abb. 20). Dieses Vorgehen wurde gewählt, um Probleme mit einfach periodischen Mustern wie zum Beispiel bei Schachbrettmustern zu verhindern. Bei solchen Mustern ist die Flussberechnung aber einer bestimmten Geschwindigkeit nicht mehr eindeutig, da nicht entschieden werden kann, um wieviele Perioden die beiden aufeinanderfolgenden Bilder zueinander verschoben wurden (siehe hierzu auch [Jähne et al., 1999]).

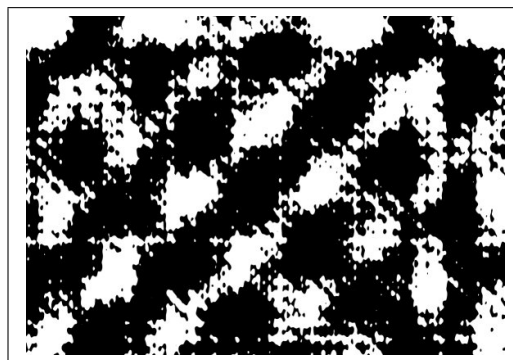


Abbildung 20: Texturen der Wände

2.3 PC-Ausstattung

Die Verarbeitung der einkommenden Videodaten lief auf einem PC mit Intel® Pentium® 4 Prozessor, welcher mit 2,6 GHz getaktet war. Der PC war mit 1024 MByte DDR Arbeitsspeicher (400 MHz) ausgestattet und kommunizierte mit Khepera- und Tracking-Kamera über jeweils eine Hauppauge WinTV mit dem Chipsatz Bt878 von Brooktree Corporation. Die Kommunikation mit der Motorkontrolle fand über den seriellen Port statt.

Der PC wurde unter Linux (Kernel 2.6.11.4) betrieben. Als Programmiersprache kam C++ zum Einsatz. Für die Kompilierung wurde der Gnu Compiler in Version 3.3.5 gewählt. Zur Visualisierung der Bedienoberfläche wurde der Qt-Designer Version 3.3.4 verwendet.

3 Algorithmen

3.1 Flussberechnung

3.1.1 Grundlagen des Lucas-Kanade-Algorithmus

Der in dieser Arbeit benutzte Algorithmus nutzt den räumlichen Bildgradienten, um die Suche nach dem besten Fluss \vec{f} an einem Punkt $\vec{u} = (u_x, u_y)^T$ zwischen zwei Bildern A und B in die richtige Richtung zu dirigieren. Es handelt sich daher um einen differentiellen Algorithmus erster Ordnung. Außerdem wird die Tatsache ausgenutzt, dass zwei hintereinander aufgenommene Bilder einer Kamera nur eine geringe Bildveränderung aufweisen. Andere Techniken sind unter anderem in [Lucas and Kanade, 1981] und [Barron et al., 1994] aufgeführt. Der Algorithmus gründet auf eine Fehlerminimierung der Funktion:

$$E(\vec{f}) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \omega(u_x - x, u_y - y) (A(x, y) - B(x + f_x, y + f_y))^2 \quad (14)$$

Diese stellt die gewichtete Summe der Quadrate der Helligkeitsdifferenzen zwischen den Pixeln des Ursprungsbildes und den durch den Fluss verschobenen Pixeln des zweiten Bildes dar. In die Summe gehen nur Pixel in einer definierten Nachbarschaft U ein, innerhalb derer man von einem konstanten Fluss ausgeht, was bei meistens kontinuierlichen Veränderungen des Flusses lokal gewährleistet ist. Diese Nachbarschaft U , welche als Integrationsfenster bezeichnet wird, ist hierbei ein Rechteck mit \vec{u} als Mittelpunkt, welches $2w_x + 1$ Pixel breit und $2w_y + 1$ Pixel hoch ist. Die Gewichtung ω besteht im Allgemeinen aus einem Gauss-Filter und einer Konfidenzmessung – in dieser Arbeit wird die Konfidenzmessung jedoch schon bei der Auswahl der Bildmerkmale (siehe Abschnitt 3.1.2) ausgeführt und auf einen Gauss-Filter wird aus Geschwindigkeitsgründen verzichtet. Gleichung 14 vereinfacht sich

somit zu:

$$E(\vec{\mathbf{f}}) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (A(x, y) - B(x + f_x, y + f_y))^2 \quad (15)$$

Da wir von geringen Bildverschiebungen ausgehen und bei Bedarf die Flussberechnung im Stile einer Newton-Raphson-Iteration wiederholen können (siehe [Bouguet, 2000]), kann die Helligkeit in Bild B an der Stelle $(x + f_x, y + f_y)$ lokal durch seine Taylor-Erweiterung erster Ordnung linear angenähert werden. Aufgrund dieser Näherung handelt es sich bei dem Lucas-Kanade-Algorithmus um einen gradientenbasierten Ansatz:

$$B(x + f_x, y + f_y) \approx B(x, y) + \left(\frac{\partial B}{\partial x} \quad \frac{\partial B}{\partial y} \right) * \vec{\mathbf{f}} \quad (16)$$

Für größere Bildverschiebungen kann ein pyramidaler Ansatz benutzt werden, welcher den Fluss in verkleinerten Bildern berechnet und diese groben Näherungen Schritt für Schritt verfeinert. Auf diese Art kommen auf jedem Skalierungslevel nur kleine Flüsse vor, für welche die Taylorerweiterung erster Ordnung anwendbar ist (siehe hierzu [Bouguet, 2000]). Minima der Fehlerfunktion 15 werden analytisch bestimmt durch die Nullstellen der ersten Ableitung:

$$\frac{\partial E(\vec{\mathbf{f}})}{\partial \vec{\mathbf{f}}} = 0$$

Die Ableitung der Fehlerfunktion nach Anwenden der Taylorerweiterung (Gleichung 16) ist:

$$\frac{\partial E(\vec{\mathbf{f}})}{\partial \vec{\mathbf{f}}} \approx -2 \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \left(A(x, y) - B(x, y) - \left[\frac{\partial B}{\partial x} \quad \frac{\partial B}{\partial y} \right] * \vec{\mathbf{f}} \right) \left[\frac{\partial B}{\partial x} \quad \frac{\partial B}{\partial y} \right] \quad (17)$$

Um eine vereinfachte Notation zu erlangen, führen wir einige neue Variablen ein. So beschreibt $\delta I(x, y)$ die zeitliche Bildableitung:

$$\delta I(x, y) \doteq A(x, y) - B(x, y)$$

und ∇I den räumlichen Bildgradienten:

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \doteq \left[\frac{\partial B}{\partial x} \quad \frac{\partial B}{\partial y} \right]^T$$

Mithilfe dieser Variablen lässt sich Gleichung 17 folgendermaßen formulieren:

$$0 = \frac{1}{2} \frac{\partial E(\vec{\mathbf{f}})}{\partial \vec{\mathbf{f}}} \approx \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (\nabla I^T \vec{\mathbf{f}} - \delta I) \nabla I^T \quad (18)$$

Ausmultipliziert und transponiert ist dies:

$$0 = \frac{1}{2} \left[\frac{\partial E(\vec{\mathbf{f}})}{\partial \vec{\mathbf{f}}} \right]^T \approx \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \vec{\mathbf{f}} - \begin{pmatrix} \delta I * I_x \\ \delta I * I_y \end{pmatrix} \quad (19)$$

Aus Gründen der Übersicht definieren wir eine Matrix G , welche die Information über die Bildstruktur enthält und einen Vektor $\vec{\mathbf{b}}$ (*image mismatch vector*):

$$G = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

$$\vec{\mathbf{b}} = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \begin{pmatrix} \delta I * I_x \\ \delta I * I_y \end{pmatrix}$$

Mit diesen Variablen lässt sich Gleichung 19 in folgender Form schreiben:

$$0 \approx G * \vec{\mathbf{f}} - \vec{\mathbf{b}}$$

so dass sich der Fluss durch folgende Gleichung berechnen lässt:

$$\vec{\mathbf{f}} \approx G^{-1} * \vec{\mathbf{b}} \quad (20)$$

Diese Gleichung ist nur lösbar, wenn die Matrix G invertierbar ist, welches bedeutet, dass $A(x, y)$ in der Nachbarschaft U hinreichend Varianz in allen Richtungen hat. Dies wird durch die vorgeschaltete Merkmalsauswahl, welche einer Konfidenzmessung (siehe Abschnitt 3.1.2) gleichkommt, garantiert.

3.1.2 Merkmalsauswahl und Konfidenzmaß

In Abschnitt 3.1.1 haben wir gesehen, dass der Algorithmus voraussetzt, dass die Matrix G invertierbar ist. In denjenigen Fällen, in denen jedoch die Determinante der Matrix G sehr klein ist, gewinnen Messfehler systematisch an Einfluss, da wir für die Bestimmung des optischen Flusses durch $\det(G)$ dividieren (siehe [Harris and Stephens, 1988]). Diese Fälle gilt es anhand einer Konfidenzmessung auszuschließen (für Übersichten verschiedener Konfidenzmaße siehe [Barron et al., 1994] und [Jähne et al., 1999]). Jedoch garantiert eine große Determinante alleine noch nicht die Minimierung von Fehlern, da beide Komponenten der Geschwindigkeitsmessung wohlbestimmt sein müssen und dies nicht anhand einer einzigen Messung garantiert werden kann. Zu diesem Zwecke betrachten wir die beiden Eigenwerte der Matrix G , welche genauer Auskunft über die Fehlerverstärkung geben können (siehe [Shi and Tomasi, 1994] und [Horn, 2003]). Je kleiner die Eigenwerte λ_1 und λ_2 der Matrix G , desto größer die Eigenwerte $1/\lambda_1$ und $1/\lambda_2$ der Inversen

G^{-1} . Die zugehörigen Eigenvektoren \vec{e}_1 und \vec{e}_2 ändern sich bei der Invertierung nicht, sind also für beide Matrizen identisch. Wir können nun jeden Vektor \vec{b} aus Gleichung 20 als Linearkombination der beiden Eigenvektoren der Matrix G schreiben:

$$\vec{b} = \alpha_1 \vec{e}_1 + \alpha_2 \vec{e}_2 \quad (21)$$

Der berechnete optische Fluss ist dann:

$$\vec{f} = G^{-1} * \vec{b} = \frac{\alpha_1}{\lambda_1} \vec{e}_1 + \frac{\alpha_2}{\lambda_2} \vec{e}_2 \quad (22)$$

Da sich eine fehlerhafte Messung auf die Faktoren α_i auswirkt, kann man sehen, dass die Fehler-Komponenten in Richtung des Eigenvektors mit dem kleineren Eigenwert einen größeren Einfluss auf den Gesamtfehler haben als Komponenten in Richtung des zweiten Eigenvektors. Als Konfidenzmaß benutzen wir daher eine untere Schranke für den vom Betrag her kleineren Eigenwert der Matrix G , wobei gleichzeitig vermieden wird, dass die Determinante von G nahe Null ist, da sich diese aus den beiden Eigenwerten λ_1 und λ_2 zusammensetzt:

$$\det(G) = \lambda_1 \lambda_2$$

Alternativ wäre es möglich, die Flüsse an sämtlichen Bildpunkten mit $\det(G) \neq 0$ zu berechnen und diese dann mit anhand eines Konfidenzmaßes wie zum Beispiel dem kleineren Eigenwert zu gewichten. Dies würde jedoch zum einen erhebliche Einbußen bezüglich der Laufzeit und zum anderen den Einfluss vieler fehlerträchtigen Flüsse mit sich bringen.

3.1.3 Entfaltung des Panoramabildes

Die aufgenommenen Panoramabilder (Abb. 21) werden entfaltet, um bei der Fehlerfunktion des Lucas-Kanade (Gleichung 15) eine Generalisierung bzgl. linearer Transformationen beiseite lassen zu können. Die Fehlerfunktion müsste bei Panoramabilder rotationsinvariant gemacht werden, da eine Rotation des Roboters eine reine Rotation des Bildes um den Mittelpunkt bewirkt und sich somit auch die lokalen Bildstrukturen in den zu vergleichenden Nachbarschaften drehen:

$$E(\vec{f}) = \min_{\phi} \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (A(x, y) - B(x, f_x, y, f_y, \phi))^2 \quad (23)$$

Im entfaltetem Bild wirkt sich dies als reine horizontale Verschiebung des Bildes aus, so dass die Minimierung über sämtliche ϕ entfallen kann.

Das Panoramabild kann durch Anwenden einer Transformation von Polarkoordinaten in kartesische Koordinaten entfaltet werden (siehe [Chahl and Srinivasan, 1997]). Da die Werte für die Entfaltung zum Programmstart berechnet und gespeichert werden, müssen dafür keine merklichen Laufzeiteinbußen in Kauf genommen werden. In dem resultierenden

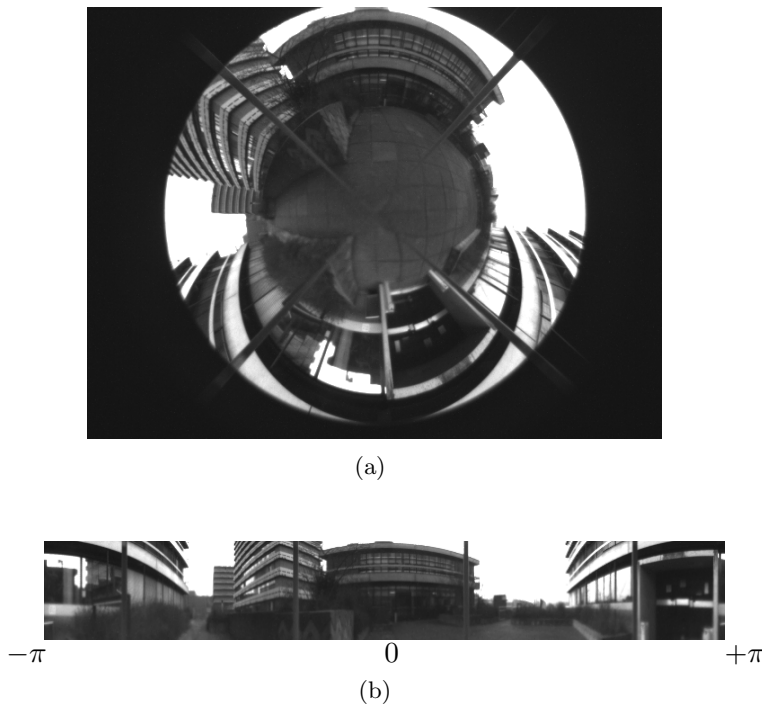


Abbildung 21: **(a)** Panoramabild und **(b)** das dazugehörige entfaltete Bild, welches zur Flussberechnung herangezogen wird. Zum besseren Verständnis ist ein Bild, welches mit einem Outdoor-Roboter gemacht wurde, zu sehen.

Bild ist die Blickrichtung in der Mitte dargestellt, die linke und rechte Seite zeigen jeweils die entsprechende Hemisphäre im Panoramabild. Eine alternative Möglichkeit, die Projektion der Geschwindigkeitsvektoren auf eine Sphäre, wird in [Gluckman and Nayar, 1998] diskutiert.

3.1.4 Fluss-Histogramme

Um einen verlässlichen Schätzer für die gemessenen Flüsse in einem Blickbereich zu erhalten, werden sämtliche horizontalen Anteile der Flüsse in ein Histogramm eingetragen. Als integrierter Fluss für diesen Blickbereich wird das Maximum des Histogramms herangezogen, welches sich sowohl als robuster als auch als schneller Schätzer erwiesen hat (siehe hierzu auch [Ballard, 1981]). Im Vergleich zu dem Durchschnittsfluss haben die Histogramme den Vorteil, dass das Ergebnis nicht von Ausreißern verfälscht wird. Ein Histogramm mit solchen Ausreißern ist beispielhaft in Abbildung 22 zu sehen.

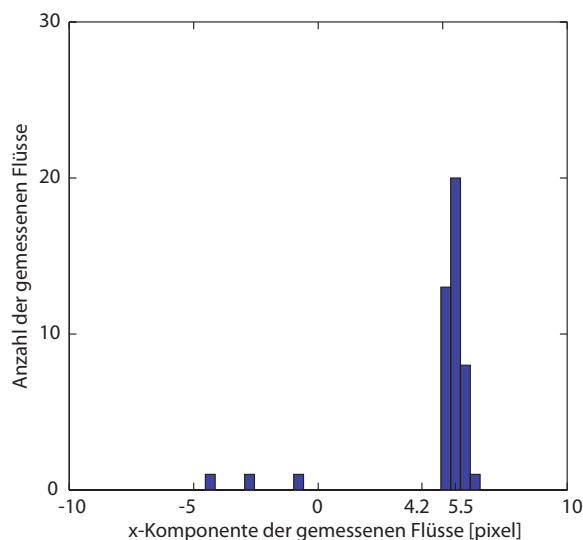


Abbildung 22: Histogramm, wie es während der Fahrt durch einen Korridor gemessen wird: Zum Vergleich ist der Durchschnitt der Flüsse von 4,2 eingezeichnet, welcher im Gegensatz zu dem Dichteschätzer von den Ausreißern beeinflusst wird.

3.1.5 Parameter der Flussberechnung

In Tabelle 1 sind die verschiedenen Parameter mitsamt der benutzten Werte aufgelistet, welche die Flussberechnung beeinflussen. Werden in einzelnen Experimenten andere Parameter benutzt, so wird dies in dem entsprechenden Abschnitt explizit angegeben.

Es wurde eine zusätzliche pyramidale Ebene eingefügt, um größere Flüsse detektieren zu können, wie sie bei Rotationen auftreten. Es hat sich außerdem gezeigt, dass 5 Iterationen auf jeder pyramidalen Ebene genügen, um eine ausreichend genaue Flussberechnung zu gewährleisten. Verbessert sich der berechnete Fluss zwischen zwei Schritten nicht um den Iterationsschwellenwert, so bricht die Iteration schon früher ab.

Die Merkmalsqualität wird für eine erste Auswahl der Bildmerkmale genutzt. In dieser Arbeit wurden alle Merkmale welche nach dem verwendeten Konfidenzmaß (siehe Abschnitt 3.1.2) mindestens eine Güte von 10 Prozent des besten Bildmerkmals aufweisen, zur weiteren Auswahl herangezogen. Dieser Standardwert wurde nach [Bouguet, 2000] gewählt.

Aus diesem Merkmalspool werden dann die 50 besten Merkmale zur Flussberechnung ausgewählt. Eine Anzahl von verfolgten 50 Merkmalen pro Blickrichtung hat sich empirisch als guter Kompromiss zwischen schneller Ausführung und ausreichender Robustheit herausgestellt. Im Gegensatz zu dem verwendeten Dichteschätzer wäre für eine robuste Schätzung anhand des Mittelwerts eine erheblich größere Anzahl an Punkten notwendig. Als Größe der Nachbarschaft wurde ein Quadrat von 13 mal 13 Pixeln als Stan-

Parameter	Beschreibung	Wert
Pyramidale Ebenen	Anzahl der herunterskalierten Bilder	1
Iterationen	Anzahl der Iterationen auf jedem Level	5
Merkmalsqualität	Schwellenwert, welcher in Prozent die Güte eines Merkmals im Vergleich zu dem optimalen Merkmal darstellt	0,1
Fenstergröße	Größe der quadratischen Nachbarschaft mit Kantenlänge n	13
Schwellenwert für die Iterationen	Bei Flussberechnungen, welche unterhalb dieses Wertes liegen, wird die Iteration auf dem gegebenen Level abgebrochen	0,1
Anzahl der Merkmale	Anzahl der Merkmale, welche maximal zur Flussberechnung herangezogen werden	50

Tabelle 1: Parameter der Flussberechnung

dardwert der Literatur entnommen ([Bouguet, 2000]).

3.2 Korridorzentrierung

3.2.1 Aufgabenbeschreibung

Während der Fahrt durch einen Korridor soll sich der Roboter immer in dessen Mitte bewegen, um die Gefahr der Kollision mit einer der Wände zu minimieren. Zusätzlich soll er dabei seine Geschwindigkeit an die Korridorbreite anpassen, so dass zu jedem Zeitpunkt eine sichere Navigation gewährleistet ist. Erreicht der Roboter das Ende des Korridors bzw. verengt sich dieser zu stark, soll der Roboter seine Fahrt beenden.

3.2.2 Genutzte Informationen des Flussfeldes

Da wir von einem ebenen Untergrund ausgehen ist nur die horizontale Komponente des optischen Flusses von Interesse. Die Bewegung des Roboters kann in eine Translations- und eine Rotationsgeschwindigkeit aufgeteilt werden. Wir gehen von einem nonholonomen Roboter aus, d.h. ein Roboter dessen Translationsbewegung sich stets nur auf seine Blickrichtung begrenzt. Ausgehend davon können wir annehmen, dass die Geschwindigkeit senkrecht zur Blickrichtung Null ist, der Geschwindigkeitsvektor in Abbildung 23 also stets in Blickrichtung zeigt. Wir teilen die auftretende Translationsgeschwindigkeit v des Roboters für eine beliebige Richtung ϕ eines Sehstrahls nochmals in zwei Bewegungskomponenten U und W auf (Abb. 23). $U = v \cdot \cos\phi$ sei hierbei die Lateralbewegung, $W = v \cdot \sin\phi$ die Vorwärtsbewegung, wobei die Bezeichnungen sich aus der Tatsache ableiten, dass von einer virtuel-

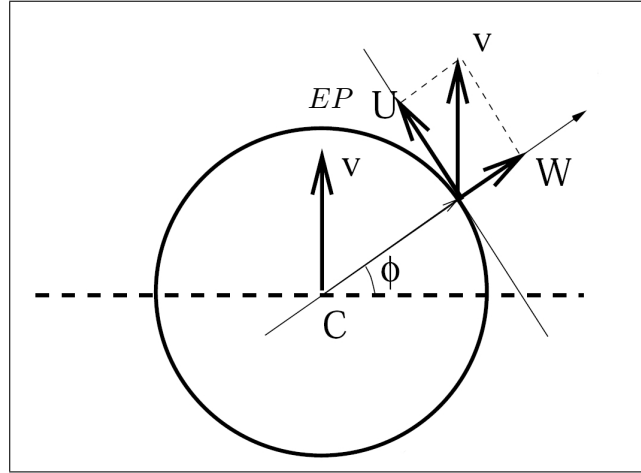


Abbildung 23: Aufteilung der Translationsbewegung in Vorwärtsbewegung W und Lateralbewegung U . Der Expansionspunkt (EP) liegt bei einer reinen Translation in Fahrtrichtung. (Verändert nach [Tsakiris and Argyros, 2000])

len Kamera in die jeweilige Blickrichtung ausgegangen wird. (Siehe auch [Argyros et al., 2004] und [Tsakiris and Argyros, 2000]) Von diesen beiden Bewegungskomponenten erzeugt nur die Lateralbewegung U einen optischen Fluss. Der gesamte Fluss, welcher sich aus einem Translations- und einem Rotationsteil zusammensetzt, ist dann:

$$u_\phi = \frac{f}{d} \cdot U + \omega \cdot f = \frac{f \cdot v \cdot \cos\phi}{d} + \omega \cdot f \quad (24)$$

d ist die Entfernung eines Punktes in der Umwelt auf dem Sehstrahl in Richtung ϕ , ω die Rotationsgeschwindigkeit des Roboters und f die Brennweite des optischen Systems ist.

Da bei einer reinen Translation der Expansionspunkt in Fahrtrichtung $\phi_h = 90^\circ$ liegt, ist dort der horizontale Anteil des optischen Flusses $u_{\phi_h} = -\omega f$ und somit unabhängig von der Translation des Roboters. Aufgrund dieser Translationsunabhängigkeit ist dieser Fluss als Maß für die Rotation geeignet.

3.2.3 Ausrichtung des Roboters

Die Zentrierung des Roboters im Korridor basiert auf der Berechnung der Flüsse zur linken und rechten Seite des Roboters und einem anschließenden Abwenden von dem größeren Fluss. Daraus entsteht ein dynamisches Einpendeln in der Mitte des Korridors. Es werden die Flüsse in den Blickrichtungen 45° links (ϕ_{l_1}) und 45° rechts (ϕ_{r_1}) der Fahrtrichtung über einen Bereich von jeweils 30° gemessen (Abb. 24) und anhand der in Abschnitt 3.1.4 beschriebenen Dichteschätzung der integrierte Fluss für die jeweilige Blickrichtung ermittelt.

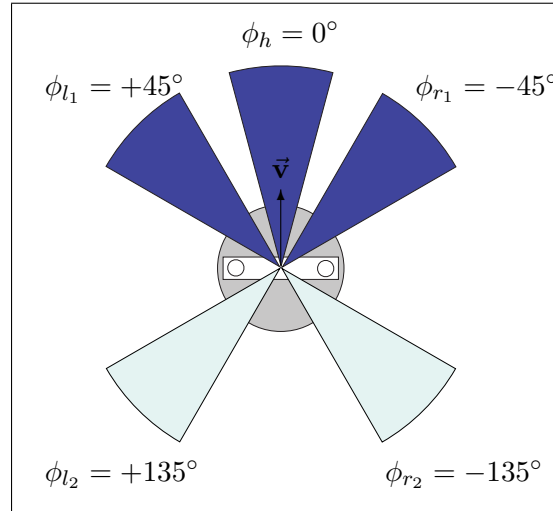


Abbildung 24: Blickrichtungen des Roboters: Die Blickrichtungen ϕ_{l_1} , ϕ_h und ϕ_{r_1} des Roboters für die Aufgabe der Korridorzentrierung sind blau eingefärbt. Die verbleibenden Blickrichtungen werden für später beschriebene Aufgaben benötigt.

Die Wahl der Blickrichtungen ϕ_i hat Einfluss auf das Verhalten des Roboters (Abb. 25). Werden die Winkel zu klein gewählt, so reagiert der Roboter auf weit entfernt liegende Hindernisse, werden die Winkel zu groß (d.h. nahe 90°) gewählt, so werden Fehler in der Orientierung des Roboters nicht erkannt und der Roboter beginnt, im Gang zu oszillieren (siehe hierzu auch [Argyros et al., 2004]). Die Wahl des Öffnungswinkels von 30° ist klein genug, um den Algorithmus nicht unnötig zu verlangsamen und gleichzeitig groß genug, um eine robuste Auswertung des Flussfeldes in diesem Bereich zu gewährleisten. In Abbildung 26 ist ein Panoramabild der Testumgebung zu sehen. Darunter ist das entfaltete Bild samt eingezeichneter Flüsse abgebildet. Wir nutzen die oben besprochene Eigenschaft des Flusses in Fahrtrichtung aus, um die Translationskomponente der Flüsse zu extrahieren. Der analog ermittelte Fluss in Fahrtrichtung u_{ϕ_h} wird somit von den lateralen Flüssen subtrahiert. Ein Maß für die Ausrichtung des Roboters ist nun die Differenz der Beträge der beiden Translationsflüsse $u_{\phi_{l_1}}$ und $u_{\phi_{r_1}}$:

$$\omega(u) = \left(\left| u_{\phi_{r_1}} - u_{\phi_h} \right| - \left| u_{\phi_{l_1}} - u_{\phi_h} \right| \right) \quad (25)$$

Überschreitet der Betrag der Winkelgeschwindigkeit einen festgelegten Schwellenwert, so wird eine Ausgleichsbewegung des Roboters ausgeführt, deren Rotationsgeschwindigkeit immer $17^\circ/s$ ist. (Dies entspricht der geringstmöglichen Rotationsgeschwindigkeit des Khepera-Roboters)

Abbildung 27 zeigt drei Histogramme, welche die gemessenen Flüsse in den drei Blickrichtungen der Korridorzentrierung darstellen (siehe Abschnitt 3.2.2). Abbildung 27(b) zeigt die Flüsse in Fahrtrichtung und lässt auf eine

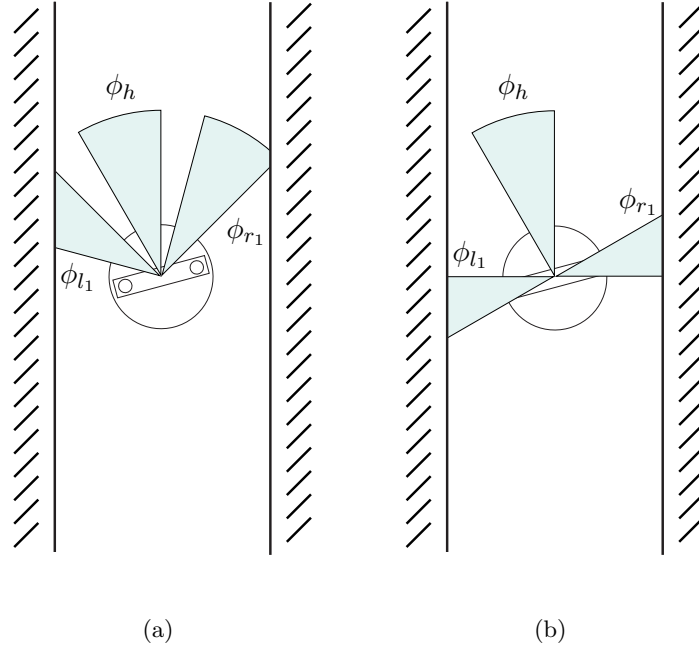


Abbildung 25: Auswirkung der Blickwinkel auf das Verhalten im Korridor:

(a) $\phi_{l_1} = \phi_{r_1} = 45^\circ$: Obwohl der Roboter sich in der Mitte des Korridors befindet, beginnt er mit einer Ausgleichsbewegung um die fehlerhafte Orientierung zu korrigieren. **(b)** $\phi_{l_1} = \phi_{r_1} = 90^\circ$: Der Roboter reagiert nicht auf die fehlerhafte Orientierung, da die Wände in beiden lateralen Blickrichtungen die gleiche Entfernung haben.

Linksrotation schließen. Die Flüsse in den Histogrammen für die linke (Abb. 27(a)) und rechte Hemisphäre (Abb. 27(c)) kommen sowohl durch diese Rotation als auch durch die Translation zustande.

3.2.4 Geschwindigkeitsregulation

Für die Anpassung der Translationsgeschwindigkeit an die Breite des Korridors berechnen wir den Durchschnitt \bar{u} der Translationsflüsse zur linken und rechten Seite $\bar{u} = \frac{1}{2} |u_{\phi_{r_1}} - u_{\phi_{l_1}}|$ und vergleichen diesen mit einem vorher festgelegten Referenzfluss u_* . Dieser Referenzfluss gibt die gewünschte Flussgeschwindigkeit in den Blickrichtungen 45 Grad lateral zur Fahrtrichtung an, welche während der Navigation auftreten soll. Die Differenz Δu des tatsächlichen durchschnittlichen Translationsflusses von dem Referenzfluss:

$$\Delta u = (\bar{u} - u_*) \quad (26)$$

führt zu einem Anpassen der Geschwindigkeit des Roboters. Überschreitet der Betrag von Δu einen festgelegten Schwellwert, so wird die Translationsge-

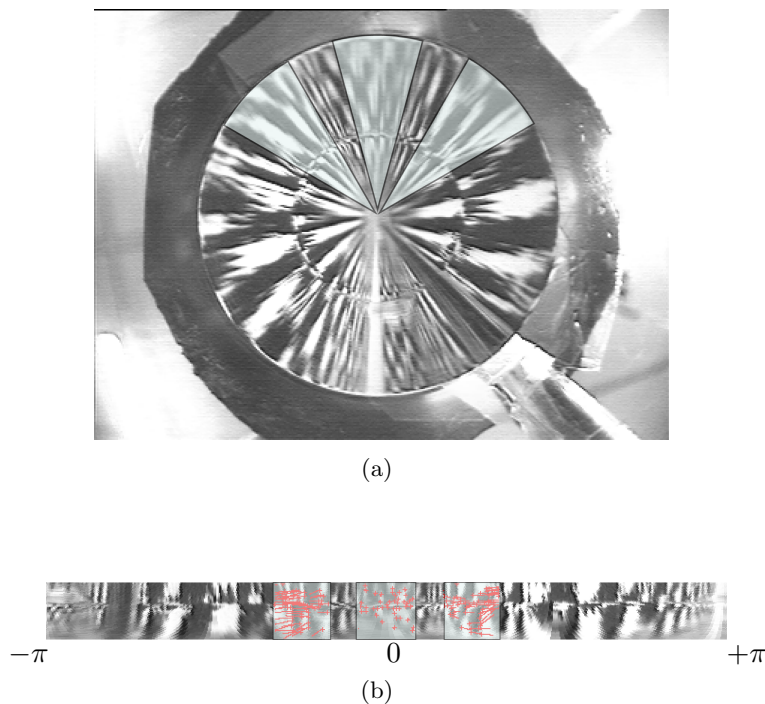


Abbildung 26: **(a)** Panoramabild der Testumgebung und **(b)** das dazugehörige entfaltete Bild, welches zur Flussberechnung herangezogen wird. In beiden Bildern sind die Sichtbereiche für die Korridorzentrierung eingezeichnet, in dem entfalten Bild sind zusätzlich die berechneten Flüsse zu erkennen (in zehnfacher Länge eingezeichnet).

schwindigkeit für $\Delta u > 0$ um 8 mm/s erhöht oder für $\Delta u < 0$ um 8 mm/s verringert.

Diese konstante Veränderung der Geschwindigkeit sowohl bezüglich der Translation wurde gewählt, da die gemessenen Flüsse in einem Blickwinkel nicht nur von der Entfernung des Roboters zur Wand abhängen, sondern auch durch die Orientierung des Roboters beeinflusst werden. Insofern misst der Roboter deutlich höhere Flüsse, wenn er diagonal im Korridor orientiert ist und somit eine der lateralen Blickrichtungen orthogonal auf eine Wand zeigen (Abb. 25).

3.2.5 Parameter der Korridorzentrierung

In Tabelle 2 sind die verschiedenen Parameter für die Zentrierungslogik mit-samt der benutzten Werte aufgelistet. Werden in einzelnen Experimenten andere Parameter benutzt, so wird dies in dem entsprechenden Abschnitt explizit angegeben.

Sämtliche Werte dieser Tabelle wurden empirisch ermittelt. Der Referenzfluss u_* von 4 wurde gewählt, so dass für die Experimente das Verhältnis

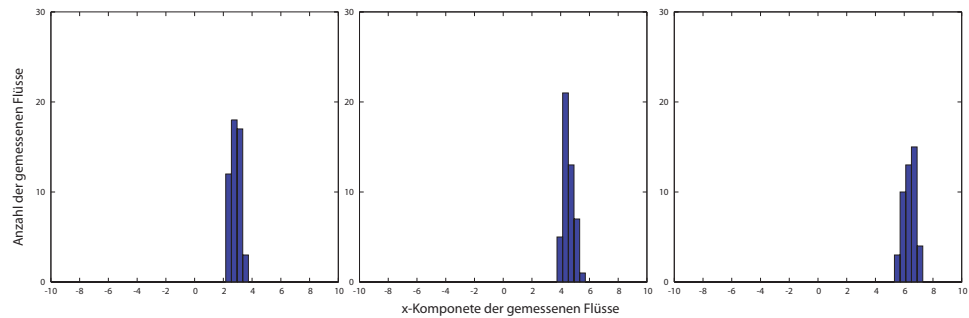


Abbildung 27: Histogramme, wie sie während einer Kombination aus Rotation und Translation für die Korridorzentrierung gemessen werden. **(b)** Das Histogramm, welches die Flüsse in Fahrtrichtung enthält, lässt auf eine Linksrotation schließen. **(a),(c)** Die Histogramme für die linke und rechte Hemisphäre sind aufgrund einer Translation relativ dazu nach links bzw. rechts verschoben.

Parameter	Beschreibung	Wert
Bin-Breite	Auflösung des Histogramms	0,4
Beschleunigungsschwelle	Wert, welcher überschritten werden muss, damit der Roboter seine Translationsgeschwindigkeit ändert	0,1
Referenz-Fluss	Zur Geschwindigkeitsregulation genutzter Translationsfluss, welcher von dem Roboter angestrebt wird	4

Tabelle 2: Parameter der Korridorzentrierung

der Geschwindigkeit des Roboters zu der Korridorbreite angemessen ist. Die Beschleunigungsschwelle verringert den Einfluss der Roboterorientierung auf die Translationsgeschwindigkeit. Die Korridorzentrierung wurde sowohl mit einer von 0,1 als auch mit einer Breite von 0,4 Pixel durchgeführt (siehe Abschnitt 4.1.1) ohne dass merkliche Unterschiede im Verhalten des Roboters zu erkennen war. Für die restlichen Experimente wurde daher eine Bin-Breite von 0,4 Pixel gewählt.

3.3 Wandverfolgung

Bei dieser Aufgabe soll sich der Roboter in einem konstanten Abstand parallel zu einer Wand bewegen. Aufgrund der Kabelführung des Khepera-Roboters (Abb. 28) ist dies in unserem Experiment nur mit rechts von dem Agenten liegenden Wänden möglich.

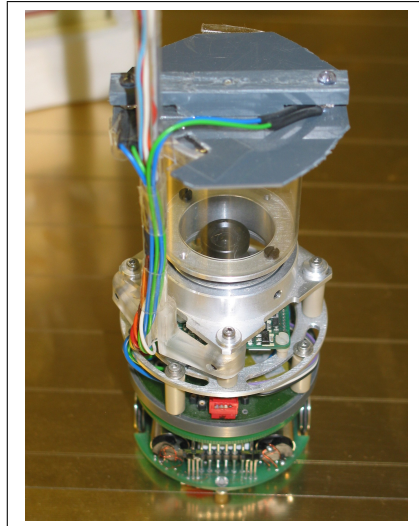


Abbildung 28: Kabelführung, welche die Wandverfolgung auf die rechte Seite einschränkt.

Hierbei wird der Fluss wie in Abbildung 29 gezeigt in die zwei Blickrichtungen ϕ_{r_1} und ϕ_{r_2} gemessen, welche diagonal auf die betreffende Wand zeigen. Der Öffnungswinkel beträgt in beide Richtungen 30° .

Die Ausrichtung an der Wand erfolgt über ein Balancieren der in diesen beiden Blickrichtungen gemessenen Flüsse $u_{\phi_{r_1}}$ und $u_{\phi_{r_2}}$:

$$\omega = u_{\phi_{r_1}} - u_{\phi_{r_2}} \quad (27)$$

Überschreitet der Betrag von ω einen Schwellenwert, so wird eine Ausgleichsbewegung des Roboters ausgeführt, deren Rotationsgeschwindigkeit immer $17^\circ/s$ ist.

3.4 Andocken

Das Andock-Verhalten ist für den Zweck gedacht, dass der Roboter an Wänden, die sich frontal in seinem Weg befinden, seine Fahrt beendet. Werden die Wände in einem spitzen Winkel angefahren, so soll der Roboter statt des Andockens die Wandverfolgung ausführen. Zu diesem Zweck werden die Flüsse in sämtlichen eingezeichneten Blickrichtungen gemessen (Abb. 30), wobei die Richtungen ϕ_{l_1} , ϕ_h und ϕ_{r_2} für das Andocken, die Richtungen ϕ_{l_2} und ϕ_{r_1} für die mögliche Wandverfolgung benötigt werden.

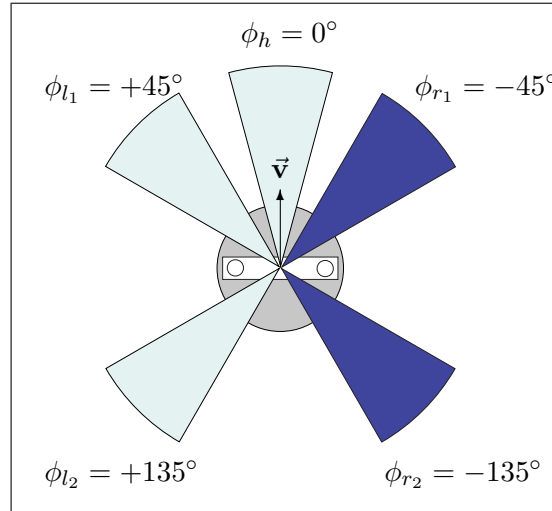


Abbildung 29: Blickrichtungen des Roboters für die Aufgabe der Wandverfolgung: Von der Fahrtrichtung \vec{v} aus ist die vordere Blickrichtung um 45° , die hintere Blickrichtung um 135° geneigt. Mit dem benutzten Roboter sind nur die eingefärbten Blickbereiche ϕ_{r_1} und ϕ_{r_2} zur rechten nutzbar.

In Anfahrt auf ein frontal in Fahrtrichtung liegende Wand misst der Roboter die Translationsflüsse in den Richtungen ϕ_{l_1} und ϕ_{r_1} (die anhand der Flüsse in Fahrtrichtung ϕ_h von der Rotationskomponente bereinigt wurden) und berechnet den Quotienten der Differenz und der Summe der Beträge dieser Flüsse in den beiden lateralen Blickrichtungen:

$$\frac{|u_{\phi_{r_1}} - u_{\phi_h}| - |u_{\phi_{l_1}} - u_{\phi_h}|}{|u_{\phi_{r_1}} - u_{\phi_h}| + |u_{\phi_{l_1}} - u_{\phi_h}|}$$

Überschreitet dieser Term einen festgelegten Schwellenwert k_w , so deutet dies darauf hin, dass die Wand in einem spitzen Winkel angefahren wird und die Wandverfolgung wird eingeleitet. Andererseits kann davon ausgegangen werden, dass die Wand annähernd orthogonal in Fahrtrichtung liegt, so dass der Roboter das Andockverhalten ausführt. Zu diesem Zweck vergleicht er in jedem Berechnungsschritt den maximalen Betrag des Translationsflusses in ϕ_{l_1} und ϕ_{r_1} mit einem Referenzwert k_d und bremst bei Überschreiten dieses Referenzwertes schrittweise ab (d.h. er verringert seine Geschwindigkeit um $8 \frac{mm}{s}$).

Die gewählten Werte des Schwellenwertes und des Referenzwertes sind in Tabelle 3 aufgelistet.

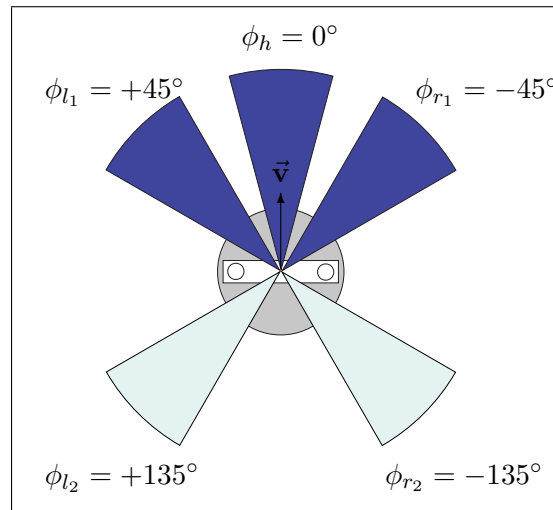


Abbildung 30: Blickrichtungen des Roboters für die Aufgabe des Andockens.

Parameter	Beschreibung	Wert
k_d	Schwellenwert, welcher für das Abbremsen genutzt wird	1,2
k_w	Schwellenwert, welcher für den Wechsel zur Wandverfolgung genutzt wird	0,85

Tabelle 3: Parameter des Andockens

4 Experimente und Ergebnisse

In der in Abschnitt 2.2 dargestellten Test-Umgebung wurden mehrere Experimente mit dem Khepera-Roboter durchgeführt, deren Ziel es war, die Leistungsfähigkeit der Flussberechnung, die Funktionalität der Motorregulation und die Sensibilität bezüglich kontrastarmer Umwelten zu untersuchen. Die Trajektorien wurden anhand der in Abschnitt 2.2.1 beschriebenen Methode ausgewertet, wobei auf eine Kalibrierung des Kamerasystems verzichtet wurde. Die anfängliche Orientierung des Roboters bei den Experimenten ist in den Abbildungen an den beiden Dioden zu erkennen, deren Verbindungslinie orthogonal zu der Fahrtrichtung verläuft.



Abbildung 31: Aufbau des 30 Zentimeter breiten Ganges. Der Roboter befindet sich in der Startposition.

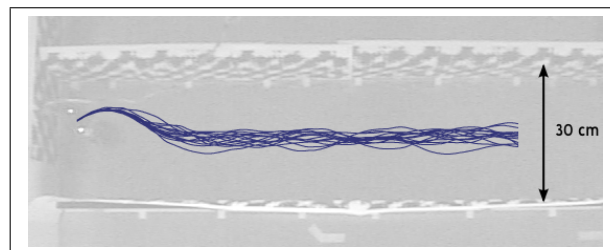


Abbildung 32: Trajektorienschar von 15 Fahrten durch einen 30 Zentimeter breiten Korridor mit einer Bin-Breite von 0,4 Pixel. Der Roboter ist in der Startposition abgebildet.

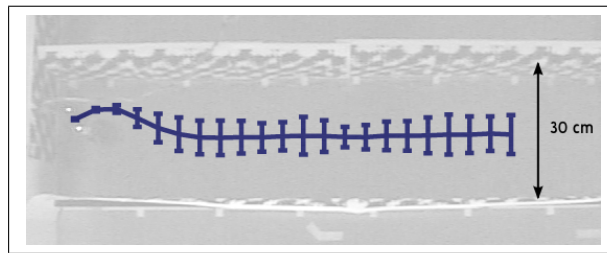


Abbildung 33: Durchschnittliche Trajektorie mit dreifachen Standardabweichungen (0,4 Pixel Bin-Breite). Zugrunde liegen die Trajektorien aus Abbildung 32. Der Roboter ist in der Startposition abgebildet.

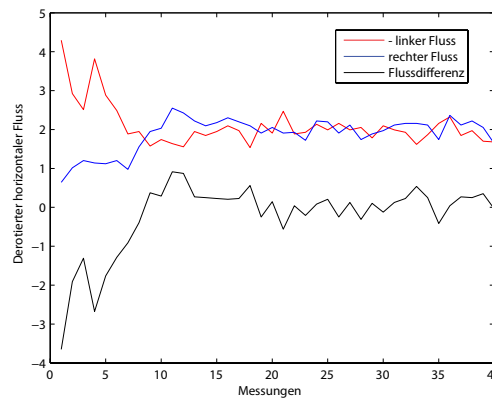


Abbildung 34: Beträge der derotierten Flüsse: Die rote und die blaue Kurve zeigen den Verlauf der Flüsse im linken bzw. rechten Blickfeld (0,4 Pixel Bin-Breite). Die schwarze Linie stellt die Differenz dieser beiden Kurven dar.

4.1 Zentrierung im geraden Korridor

Es wurden Experimente in zwei geraden Korridoren mit unterschiedlicher aber gleichbleibender Breite von 30 respektive 15 Zentimetern durchgeführt.

4.1.1 Korridor mit einer Breite von 30 Zentimetern

4.1.1.1 Versuchsaufbau

In dem Korridor mit einer Breite von 30 Zentimetern wurden Experimente mit unterschiedlichen Einstellungen für die Histogramm-Auswertung durchgeführt (siehe hierzu auch Abschnitt 3.1.4), um die Einflüsse der Diskretisierung zu untersuchen. Die restlichen Parameter waren für beide Experimente identisch. In Abbildung 31 ist der Roboter in Startposition zu sehen.

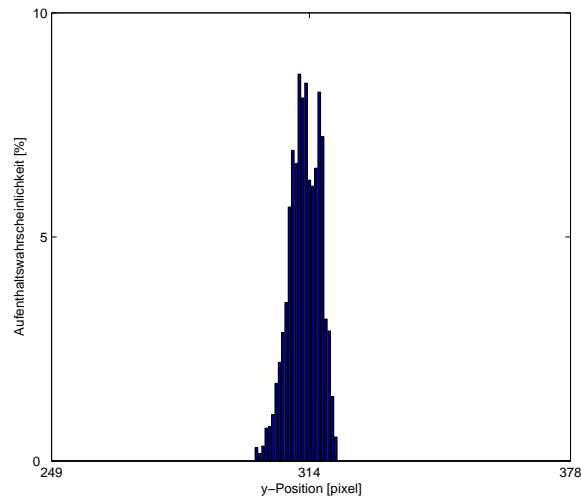


Abbildung 35: Aufenthaltswahrscheinlichkeit im Korridor: Es ist die Wahrscheinlichkeit aufgetragen, dass sich der Roboter in einem bestimmten Bereich der y-Achse befindet. Es liegen die Positionen der 15 Trajektorien aus dem Bereich zugrunde, in dem sich der Roboter in der Mitte des Korridors eingependelt hat.

4.1.1.2 Ergebnisse

4.1.1.2.1 Bin-Breite von 0,4 Pixel

In diesem Experiment betrug die Klassenbreite in den für die Auswertung benutzten Histogrammen 0,4 Pixel. Es wurden insgesamt 15 Fahrten durchgeführt.

In Abbildung 32 ist die Trajektorien­schar für 15 durchgeführte Läufe zu sehen. Abbildung 33 zeigt die über diese Schar gemittelte Trajektorie mit den dreifachen Standardabweichungen, welche einer Aufenthaltswahrschein-

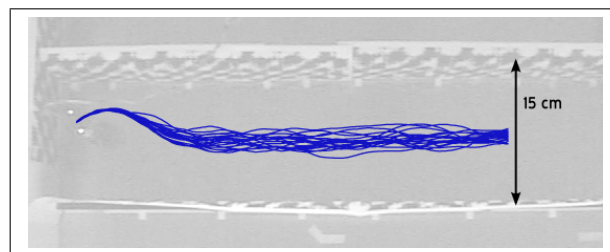


Abbildung 36: Trajektorien­schar von 20 Fahrten durch einen 30 Zentimeter breiten Korridor mit einer Bin-Breite von 0,1 Pixel. Der Roboter ist in der Startposition abgebildet.

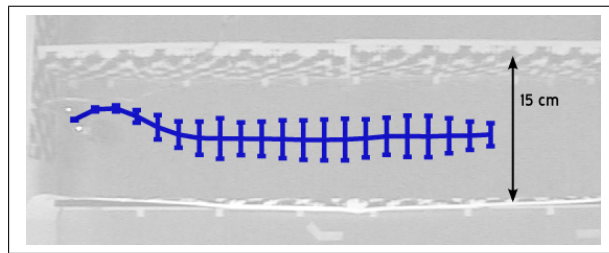


Abbildung 37: Durchschnittliche Trajektorie mit dreifachen Standardabweichungen (0,1 Pixel Bin-Breite). Zugrunde liegen die Trajektorien aus Abbildung 36. Der Roboter ist in der Startposition abgebildet.

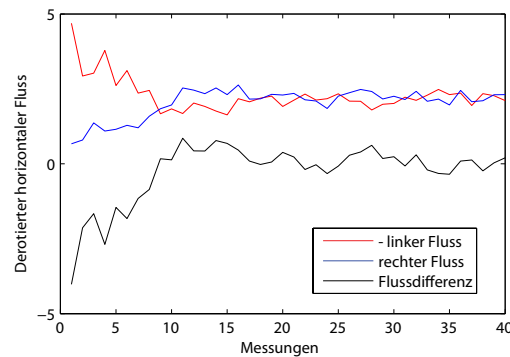


Abbildung 38: Beträge der derotierten Flüsse: Die rote und die blaue Kurve zeigen den Verlauf der Flüsse im linken bzw. rechten Blickfeld (0,1 Pixel Bin-Breite). Die schwarze Linie stellt die Differenz dieser beiden Kurven dar.

lichkeit von über 99 Prozent entsprechen.

Abbildung 34 zeigt die durchschnittlichen Beträge der derotierten Flüsse im linken und rechten Blickbereich über die 15 Läufe. Zusätzlich ist die Differenz dieser Beträge eingezeichnet, welche ein Maß für die Zentrierung des Roboters ist.

Da der Korridor in Richtung der x-Achse ausgerichtet ist, kann die y-Komponente der Roboterposition Auskunft über dessen Zentrierung geben. Die Verteilung dieser y-Komponente nachdem sich der Roboter in dem Korridor eingependelt hat ist in Abbildung 35 zu sehen.

4.1.1.2.2 Bin-Breite von 0,1 Pixel

In diesem Experiment wurde die Klassenbreite in den für die Auswertung benutzten Histogrammen auf 0,1 Pixel verringert, um mögliche Einflüsse der Diskretisierung zu untersuchen. Es wurden insgesamt 20 Fahrten durchgeführt.

In Abbildung 36 ist die Trajektorienschar für 20 durchgeführte Läufe zu sehen. Abbildung 37 zeigt die über diese Schar gemittelte Trajektorie mit

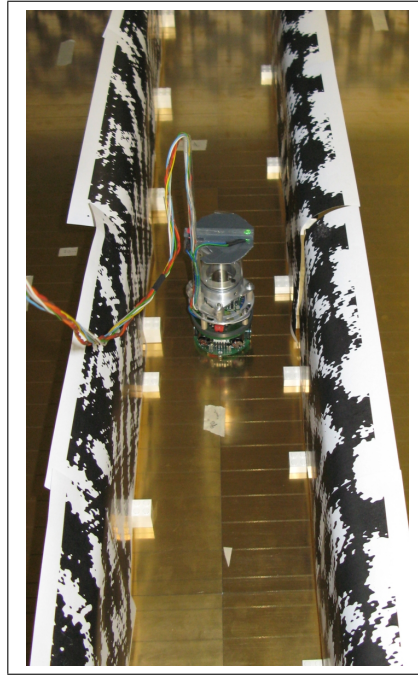


Abbildung 39: Aufbau des 15 Zentimeter breiten Ganges.

den dreifachen Standardabweichungen, welche einer Aufenthaltswahrscheinlichkeit von über 99 Prozent entsprechen.

Abbildung 38 zeigt die durchschnittlichen Beträge der derotierten Flüsse im linken und rechten Blickbereich über die 20 Läufe. Zusätzlich ist die Differenz dieser Beträge eingezeichnet, welche ein Maß für die Zentrierung des Roboters ist.

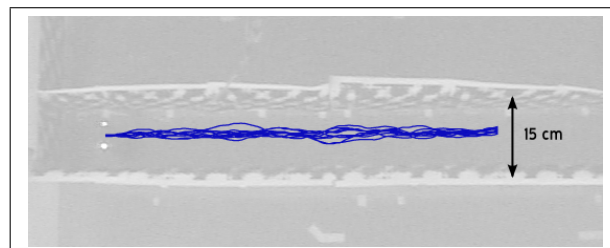


Abbildung 40: Trajektorienschar von 10 Fahrten durch einen 15 Zentimeter breiten Korridor. Der Roboter ist in der Startposition abgebildet.

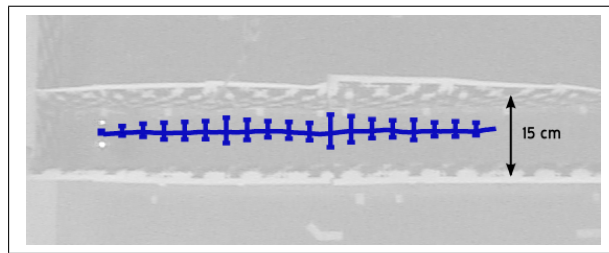


Abbildung 41: Durchschnittliche Trajektorie mit dreifachen Standardabweichungen. Zugrunde liegen die Trajektorien aus Abbildung 40. Der Roboter ist in der Startposition abgebildet.

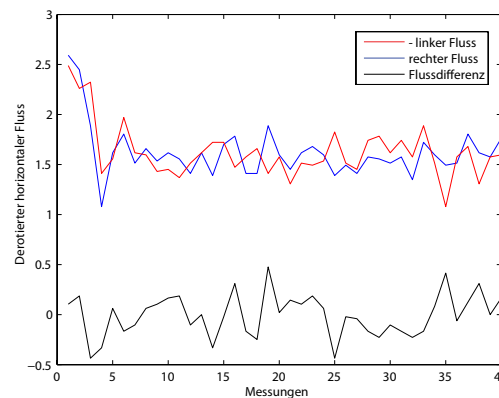


Abbildung 42: Beträge der derotierten Flüsse: Die rote und die blaue Kurve zeigen den Verlauf der Flüsse im linken bzw. rechten Blickfeld. Die schwarze Linie stellt die Differenz dieser beiden Kurven dar.

4.1.2 Korridor mit einer Breite von 15 Zentimetern

4.1.2.1 Versuchsaufbau

In einem schmalen Korridor mit einer gleichbleibenden Breite von 15 Zentimetern wurde das Verhalten des Roboters auf mögliche Oszillationen bei kurzen Distanzen zu Hindernissen hin untersucht. Abbildung 39 zeigt den Korridor mit dem Roboter.

4.1.2.2 Ergebnisse

In Abbildung 40 ist die Trajektorien­schar für 10 durchgeführte Fahrten zu sehen. Abbildung 41 zeigt die über diese Schar gemittelte Trajektorie mit den dreifachen Standardabweichungen, welche einer Aufenthaltswahrscheinlichkeit von über 99 Prozent entsprechen.

Abbildung 42 zeigt die durchschnittlichen Beträge der derotierten Flüsse

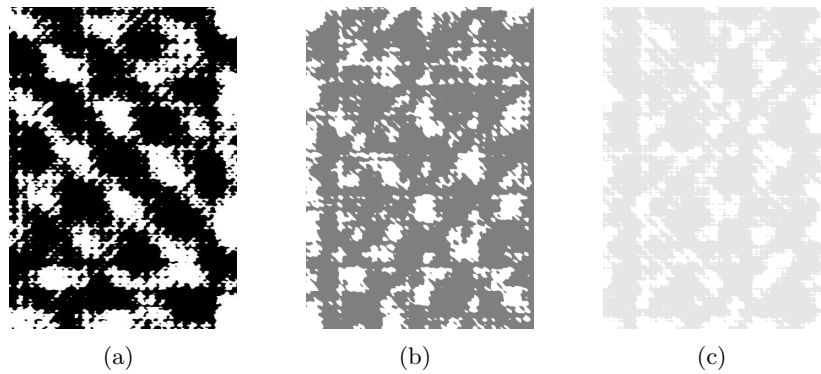


Abbildung 43: Texturen mit unterschiedlich starken Kontrasten. In (a) ist eine Modulation von 1 zu sehen, in (b) eine Modulation von ca. 0,33 und in (c) eine Modulation von ca. 0,05.

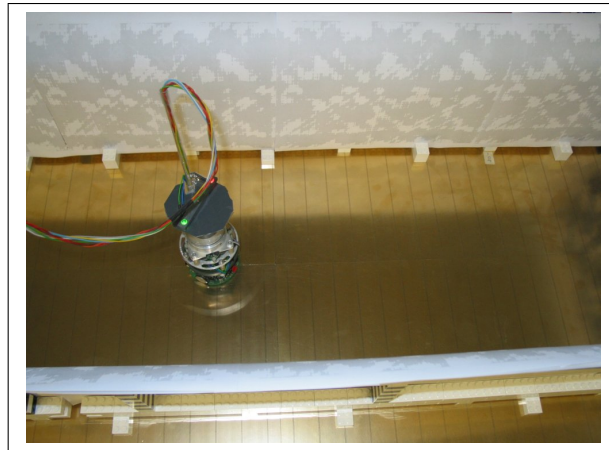


Abbildung 44: Khepera-Roboter vor kontrastarmer Textur.

im linken und rechten Blickbereich über die zehn Läufe. Zusätzlich ist die Differenz dieser Beträge eingezeichnet, welche ein Maß für die Zentrierung des Roboters ist.

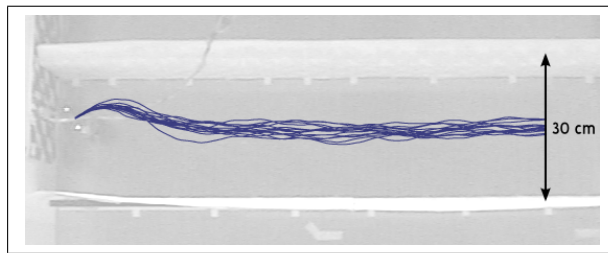


Abbildung 45: Trajektorienschar von 15 Fahrten durch einen 30 Zentimeter breiten Korridor. Der Roboter ist in der Startposition abgebildet.

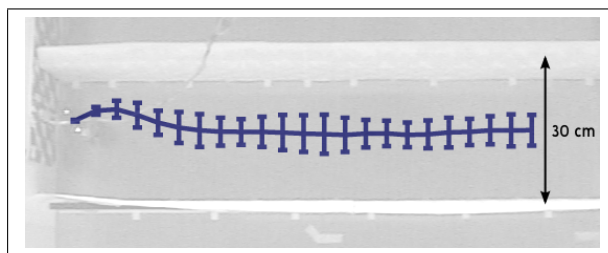


Abbildung 46: Durchschnittliche Trajektorie mit dreifachen Standardabweichungen. Zugrunde liegen die Trajektorien aus Abbildung 45. Der Roboter ist in der Startposition abgebildet.

4.1.3 Kontrastempfindlichkeit

4.1.3.1 Versuchsaufbau

Um die Robustheit des Algorithmus bezüglich des Kontrastes der Umwelt zu untersuchen, wurde der Helligkeitswert der ursprünglich schwarzen Bereiche der Wand-Textur (Abschnitt 2.2.2) sukzessive erhöht (Abb. 43) und das Verhalten des Roboters in einem geraden Korridor mit der Breite von 30 Zentimetern untersucht.

Die Modulation (auch Michelson-Kontrast genannt) bezeichnet das Verhältnis zwischen der Differenz und der Summe der beiden Leuchtdichten und wird anhand folgender Formel berechnet:

$$Modulation = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \quad (28)$$

Nach mehreren Testfahrten mit unterschiedlichen Kontrasten wurde eine Modulation von 0,05 wie sie in Abbildung 43(c) zu sehen ist für die Untersuchung bezüglich der Navigationsfähigkeit bei kontrastarmen Hindernissen benutzt, da bei einem noch geringeren Kontrast keine kontrollierte Fahrt mehr möglich war.

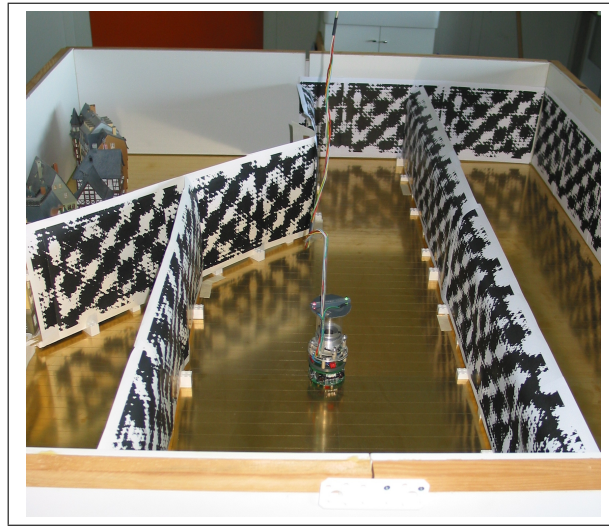


Abbildung 47: Aufbau des sich verengenden Korridors.

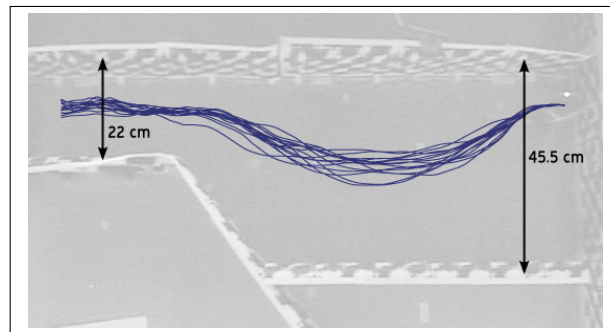


Abbildung 48: Trajektorienschar von 15 Fahrten durch den sich verengenden Korridor. Der Roboter ist rechts oben im Bild in der Startposition parallel zu den Wänden orientiert abgebildet.

4.1.3.2 Ergebnisse

In Abbildung 45 ist die Trajektorienschar für 15 durchgeführte Fahrten zu sehen. Abbildung 46 zeigt die über diese Schar gemittelte Trajektorie mit den dreifachen Standardabweichungen, welche einer Aufenthaltswahrscheinlichkeit von über 99 Prozent entsprechen.

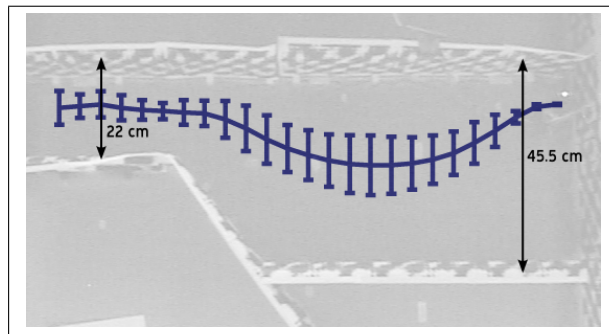


Abbildung 49: Durchschnittliche Trajektorie mit dreifachen Standardabweichungen. Zugrunde liegen die Trajektorien aus Abbildung 48. Der Roboter ist rechts oben im Bild in der Startposition parallel zu den Wänden orientiert abgebildet.

4.2 Zentrierung im sich verengenden Korridor

4.2.1 Versuchsaufbau

In einem sich verengenden Korridor wurde die Fähigkeit des Roboters untersucht, seine Geschwindigkeit dynamisch auf die Entfernung zu seiner Umwelt (d.h. zu den Korridorwänden) anzupassen. Der Korridor besaß eine Anfangsbreite von 45,5 Zentimetern und verengte sich auf 22,0 Zentimeter. Abbildung 47 zeigt ein Photo des Korridors mit dem Roboter während der Zentrierung.

4.2.2 Ergebnisse

In Abbildung 48 ist die Trajektorienchar für 15 durchgeführte Fahrten zu sehen. Abbildung 49 zeigt die über diese Schar gemittelte Trajektorie mit den dreifachen Standardabweichungen, welche einer Aufenthaltswahrscheinlichkeit von über 99 Prozent entsprechen.

In Abbildung 50 ist die Entwicklung der Translationsgeschwindigkeit während der Fahrt durch den sich verengenden Korridor dargestellt. Der Roboter bewegt sich dabei im Diagramm von rechts nach links. Die Beschleunigung zu Beginn des Laufs kommt durch eine festgelegte Startgeschwindigkeit zustande, welche für die hier gewählte Korridorbreite zu gering ist.

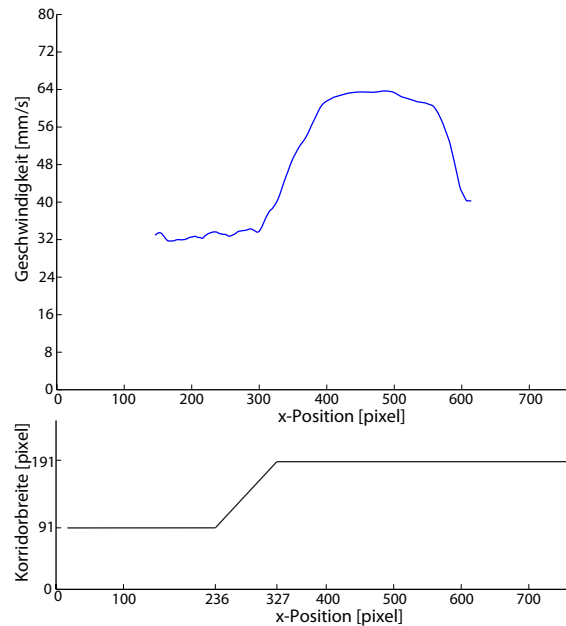


Abbildung 50: Geschwindigkeit in Abhängigkeit der Korridorbreite. Der Roboter bewegt sich von rechts nach links.

4.3 Zentrierung im gekrümmten Korridor

4.3.1 Versuchsaufbau

In einem Korridor mit einer 90° -Kurve wurde die Fähigkeit des Roboters, sich auch in gekrümmten Korridoren fortzubewegen, getestet. Abbildung 51 zeigt ein Bild des gekrümmten Korridors mit dem Roboter in Startposition.

Zusätzlich zu diesem Experiment wurde eine Fahrt durch einen gekrümmten Korridor durchgeführt, welcher aus Modellhäusern besteht welche eine realistischere Umwelt ohne homogenen Kontrast darstellen. Der Aufbau ist in Abbildung 52 zu sehen.

4.3.2 Ergebnisse

In Abbildung 53(a) ist die Trajektorienschar für 15 durchgeführte Läufe zu sehen. Abbildung 53(b) zeigt die über diese Schar gemittelte Trajektorie mit den dreifachen Standardabweichungen, welche einer Aufenthaltswahrscheinlichkeit von über 99 Prozent entsprechen.

Die Trajektorien der Fahrten zwischen den Modellhäusern sind in Abbildung 54 zu sehen.

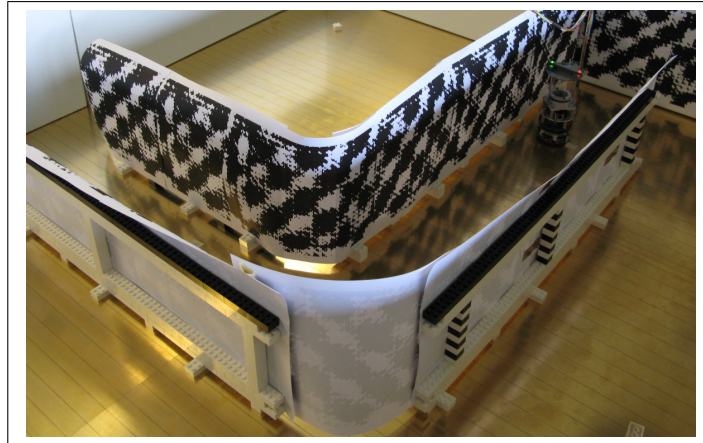


Abbildung 51: Aufbau des gekrümmten Korridors: Der Roboter ist rechts oben in der Startposition zu sehen.



Abbildung 52: Aufbau einer gekrümmten Gasse mit Modellhäusern.

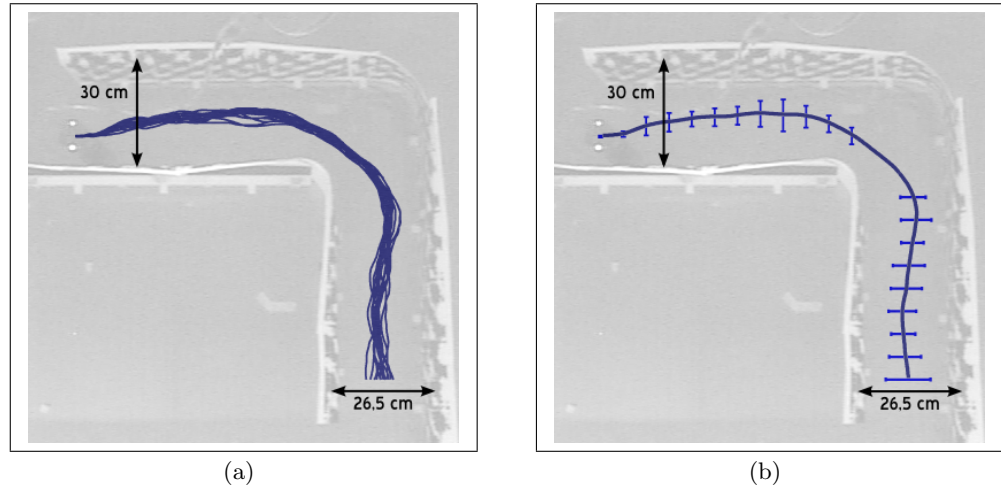


Abbildung 53: Gekrümmter Korridor: **(a)** Trajektorienchar von 15 Fahrten durch den gekrümmten Korridor. Der Roboter ist in der Startposition abgebildet. **(b)** Durchschnittliche Trajektorie mit dreifachen Standardabweichungen. Zugrunde liegen die Trajektorien aus Abbildung 53(a). Der Roboter ist in der Startposition abgebildet

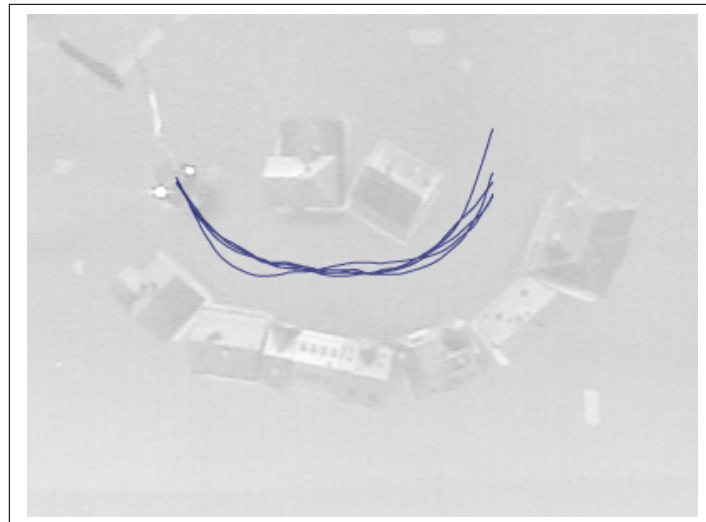


Abbildung 54: Fünf Trajektorien der Fahrten durch die Modellhausgasse.

4.4 Wandverfolgung

4.4.1 Versuchsaufbau

In diesem Experiment wurde die Fähigkeit des Roboters getestet, sich an einer Wand entlangzubewegen. Zu diesem Zweck wurde der Roboter wie in Abbildung 55 zu sehen in einer Entfernung von 15 Zentimetern zu einer geraden Wand positioniert und die Wandverfolgungs-Logik aus Abschnitt 3.3 verwendet.

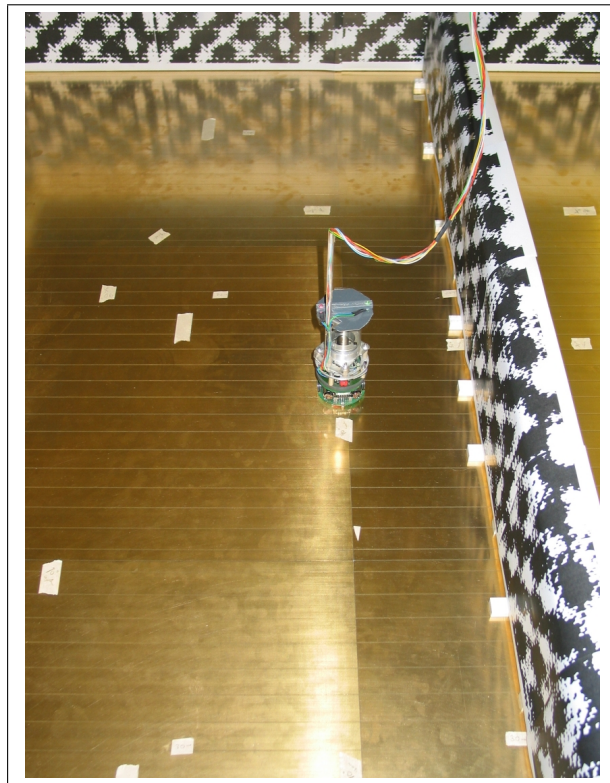


Abbildung 55: Experimentaufbau für die Wandverfolgung.

4.4.2 Ergebnisse

In Abbildung 56 ist die Trajektorienschar für 14 durchgeführte Läufe zu sehen. Abbildung 57 zeigt die über diese Schar gemittelte Trajektorie mit den dreifachen Standardabweichungen, welche einer Aufenthaltswahrscheinlichkeit von über 99 Prozent entsprechen.

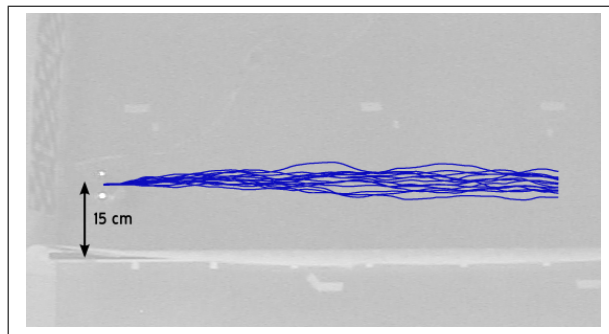


Abbildung 56: Trajektorienschärze von 14 Fahrten während der Wandverfolgung. Der Roboter ist in der Startposition abgebildet.

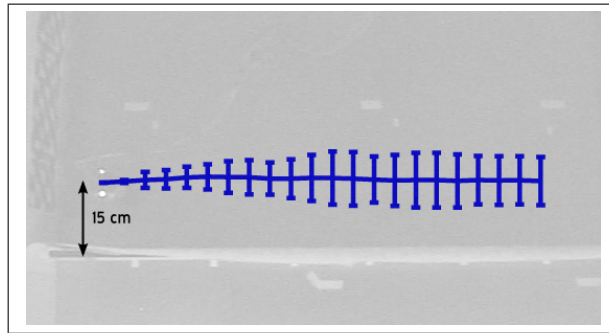


Abbildung 57: Durchschnittliche Trajektorie mit dreifachen Standardabweichungen. Zugrunde liegen die Trajektorien aus Abbildung 56. Der Roboter ist in der Startposition abgebildet.

4.5 Andocken

4.5.1 Versuchsaufbau

In diesem Experiment wurde das Andocken an eine Wand getestet. Dieses Verhalten ist für das Beenden der Navigation vorgesehen, wenn durch anderweitige Navigationsstrategien keine Richtungsänderung vorgegeben wird, welche die Kollision mit einem Hindernis abwenden würden. Abbildung 58 zeigt den Roboter in der Startposition mit einer Orientierung von 75° zur Wand. Der Roboter wurde in vier verschiedenen Winkeln zur Wand gestartet (90° , 75° , 60° , 45°).

4.5.2 Ergebnisse

Der Roboter wurde mehrmals in unterschiedlichen Winkel zur Wand gestartet. Jeweils drei Trajektorien aus vier verschiedenen Startpositionen sind in Abbildung 59 zu sehen. Wechselt der Roboter vom Andocken in die Wandverfolgung, so versucht er den zu diesem Zeitpunkt vorhandenen Abstand

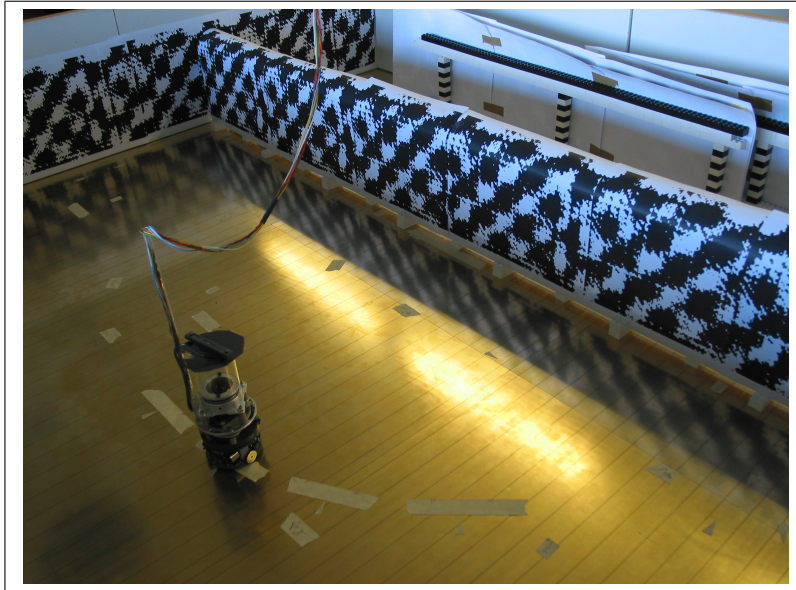


Abbildung 58: Experimentaufbau für das Andocken. Der Roboter ist in einem Winkel von 75° zur Wand orientiert.

zur Wand zu halten. Je nach Zeitpunkt der Aktivierung kann die Wandverfolgung somit in unterschiedlichen Abständen erfolgen.

Abbildung 60 zeigt die Geschwindigkeitsanpassung für die Fahrten mit einem Winkel von ungefähr 90 und 75 Grad zwischen Fahrtrichtung der Roboters und Ausrichtung der Wand.

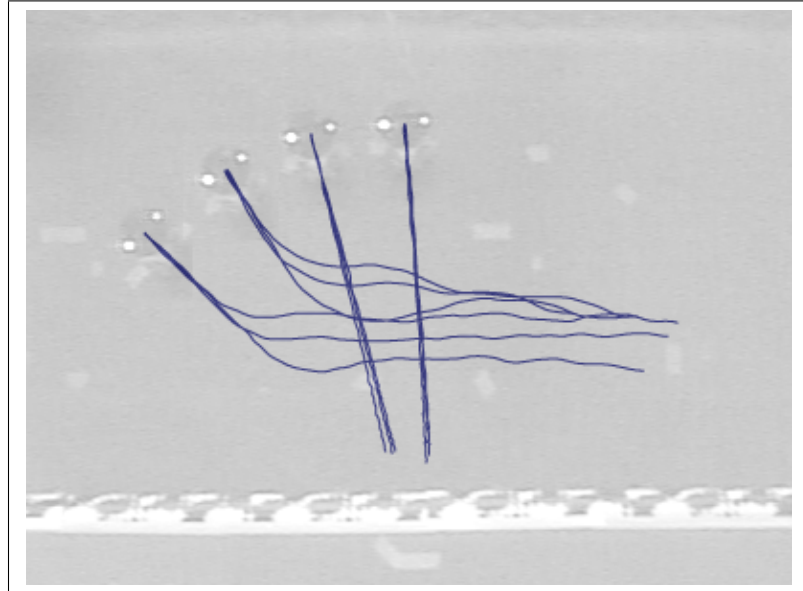


Abbildung 59: Trajektorienschar von 12 Fahrten während des Andockens. Der Roboter ist in den vier verschiedenen Startpositionen abgebildet.

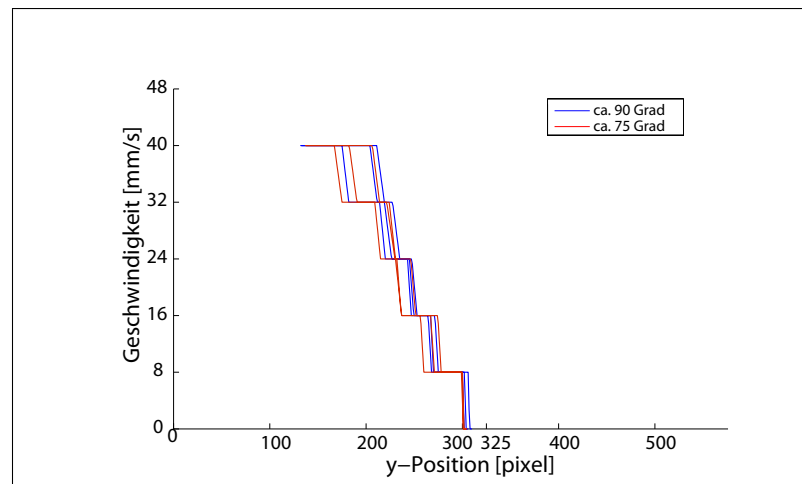


Abbildung 60: Entwicklung der Geschwindigkeit während des Andockens: Die Wand befindet sich auf einer Höhe von 325 Pixel im Bild. Die sprunghafte Veränderung stellt die minimale mögliche Geschwindigkeitsänderung des Roboters dar.

5 Diskussion und Schlussfolgerung

Durch die überragende Navigationsfähigkeit vieler Insekten inspiriert wurden in dieser Arbeit verschiedene auf die Auswertung des optischen Flussfeldes basierende lokale Navigationsstrategien entwickelt und auf unterschiedliche Einflussfaktoren hin getestet. So war zum einen von Interesse, in welchem Kontrastbereich der verwendete differentielle Algorithmus mit nachgeschaltetem Dichteschätzer einsetzbar ist. Zum anderen wurde der Dichteschätzer auf mögliche Diskretisierungsartefakte hin untersucht.

Die Ergebnisse zeigen, dass sowohl die Genauigkeit als auch die Geschwindigkeit des benutzten differentiellen Flussalgorithmus für die gestellten Aufgaben ausreichen. Im Verlaufe sämtlicher hier aufgeführten Experimente fand keine Kollision mit einer der Wände statt.

Die Zentrierung des Roboters war sowohl in geraden als auch in gekrümmten Korridoren möglich. Die Ergebnisse in Kapitel 4 zeigen, dass unter der konservativen Annahme einer Normalverteilung der Robotertrajektorien in über 99 Prozent aller Fahrten keine Kollision mit den Wänden stattfindet.

Die Ergebnisse der Experimente in dem Korridor mit einer Breite von 15 Zentimetern zeigen, dass auch in geringer Distanz zu den Wänden keine merklichen Oszillationen stattfinden (Abschnitt 4.1.2). Die Zentrierung in dem anhand von Modellhäusern (Abschnitt 4.3) erstellten Korridor zeigt, dass auch unter realistischeren Bedingungen die Informationen in der Umwelt ausreichen, um eine sichere Navigation zu ermöglichen.

Ebenfalls positive Ergebnisse im Bezug auf die Korridorzentrierung anhand des optischen Flusses wurden von [Argyros et al., 2004] erbracht, welche in ihren Experimenten ebenfalls ein katadioptrisches System benutzten.

Wie in Abschnitt 4.4 zu sehen, ist die simple Logik für eine Wandverfolgung ausreichend, welche zu jedem Zeitpunkt einen konstanten Abstand zur Wand einhält. Sollte der Roboter jedoch eine definierte Entfernung zu der Wand einhalten, so wäre es nötig, den gemessenen Fluss in diesem Abstand als Referenzfluss beizubehalten.

Sowohl die Korridorzentrierung als auch die Wandverfolgung wurde von [Serres et al., 2006] mit Hilfe eines Luftkissenfahrzeuges entwickelt, auf welchem ein bewegliches optisches System installiert war, welches der Rotation des Fahrzeugs entgegenwirken konnte um die Korridorwände jeweils in einem rechten Winkel anzuvisieren. Ebenfalls eine Kombination aus Korridorzentrierung und Wandverfolgung mit starker biologischer Motivation wurde von [Weber et al., 1996] vorgestellt.

[Zufferey and Floreano, 2005] haben ebenfalls Experimente mit einem Khepera-Roboter durchgeführt, um die Tauglichkeit von Flussalgorithmen für fliegende Roboter zu testen. Im Gegensatz zu dem hier verwendeten katadioptrischen System wurde jedoch eine Fischaugenkamera verwendet. Sowohl Hindernisvermeidung als auch Wandverfolgung wurden in dieser Ar-

beit erfolgreich untersucht.

Bei den Tests des Andockens (Abschnitt 4.5) hielt der Roboter in unmittelbarer Nähe zu der in Fahrtrichtung gelegenen Wand an, wenn er mit einer Orientierung von 90° und 75° auf diese auffuhr. Bei den Fahrten mit einer Orientierung von 60° und 45° ging der Roboter in jeder Fahrt zur Wandverfolgung über.

Die Dichteschätzung der Flussvektoren zeigte sich weitgehend robust gegenüber der gewählten Diskretisierung (Bin-Breiten von 0,1 bis 0,4 Pixel). Testfahrten mit einer Rotationsschwelle ähnlich der bei der Translation benutzten Schwelle brachten keine Verbesserung der Navigation mit sich, da der Roboter zwar weniger Ausgleichsbewegungen ausführte, dafür jedoch stärker von der Mitte des Korridors abwich.

Experimente durch einen 30 Zentimeter breiten Korridor mit verschiedenen Kontrasten wurden in Abschnitt 4.1.1 und Abschnitt 4.1.3 durchgeführt. Im ersten Versuch wurde eine Textur mit einem Kontrast von 1 benutzt (siehe auch Abschnitt 2.2.2), im zweiten Versuch eine Textur mit geringerem Kontrast von 0.05, welcher durch eine Abschwächung der dunklen Texturbereiche von dem Helligkeitswert 0 auf 230 (255 stellt reines Weiß dar) zustande kam. In den beiden Versuchen fand in keinem der Experimente eine Kollision mit einer der Wände statt und auch die Trajektorienscharen zeigten keine erkennbaren Unterschiede. Dies zeigt, dass sich das verwendete Verfahren in einem großen Kontrastbereich äußerst robust verhält.

Wie schon in den Abschnitten 3.2.3 und 3.2.4 erläutert, wird in dieser Arbeit eine konstante Motorregulation benutzt, da nur die kleinstmögliche Winkelgeschwindigkeit gefahren wird. Falls größere Winkelgeschwindigkeiten benötigt werden, wäre eine sigmoidale Aktivierungsfunktion statt eines diskreten Schwellenwertes denkbar.

Durch die 45° -Blickrichtungen reagiert die Geschwindigkeitsregulation sehr sensibel auf Rotationen, da durch diese die Entfernungen zu den Wänden in den genutzten Blickrichtungen stark variieren. Aus diesem Grund wurde nur eine konstante Veränderung der Translationsgeschwindigkeit (siehe Abschnitt 3.2.4) zugrunde gelegt, welche anhand eines Schwellenwertes rotationsbedingte Geschwindigkeitsänderung abschwächt. Die Ergebnisse in Abschnitt 4.2.2 zeigen, dass dies für eine ausreichend konstante Geschwindigkeit in den gewählten künstlichen Umgebungen ausreicht.

Für eine agilere Geschwindigkeitsregulation wäre es besser, für die Anpassung der Geschwindigkeit das gesamte 360° -Blickfeld auszunutzen und die Richtung mit den größten Flüssen zur Regulation heranzuziehen. Auf diese Weise würde der Einfluss der Orientierung des Roboters in seiner Umwelt keinen Einfluss auf seine Translationsgeschwindigkeit haben und die Translation könnte proportional zum Fluss verändert werden.

Die Dauer eines Steuerzyklus für die lokalen Navigationsaufgaben ist mit einer Dauer von ca. 0,07 Sekunden ausreichend, kann jedoch bei Bedarf anhand einiger Optimierungen weiter verbessert werden. So wurde in dieser

Arbeit das gesamte Bild entfaltet obwohl im Nachhinein nur wohl definierte Blickrichtungen für die Flussberechnungen genutzt wurden. Außerdem wurden von Seiten der Software die farbigen Kamerabilder in Schwarzweiß-Bilder konvertiert, was ebenfalls zu Laufzeiteinbußen führte. Es ist jedoch zu bedenken, dass für diffizilere Aufgaben eventuell die Auflösung des entfalteten Bildes nicht ausreicht und dass durch eine Erhöhung der Auflösung der Rechenbedarf ansteigt. Eine weitere Möglichkeit der Geschwindigkeitsoptimierung wäre es, weite Teile des Verfahrens als Hardwarelösung bereitzustellen.

Obwohl der Roboter mit den hier entwickelten Regelkreisen unterschiedliche Situationen aufgrund fehlender globaler Flussauswertungen (ähnlich dem VS8-Neuron aus [Krapp and Hengstenberg, 1996]) nicht immer unterscheiden kann, haben sich die Algorithmen für die vorgegebenen Aufgaben als sehr robust erwiesen.

6 Ausblick

Um die Stabilität der verschiedenen Navigationsstrategien weiter zu erhöhen, könnte das Blickfeld des Roboters sukzessive vergrößert werden. So könnte für die Messung des Rotationsflusses auch die Blickrichtung entgegen der Fahrtrichtung genutzt werden d.h. der Bereich um den Kontraktionspunkt des Flussfeldes.

Ein kontinuierlicher Regler, welcher die einzelnen hier implementierten Aufgaben den Umständen entsprechend gewichtet, würde eine deutliche Komplexitätssteigerung des möglichen Verhaltens des Roboters mit sich bringen. Der Übergang während des Andockens zur Wandverfolgung ist ein Beispiel für die sinnvolle Kombination lokaler Navigationsstrategien.

Als Einsatzgebiet der implementierten Regelkreise sei die Integration in globale Aufgaben wie Pfadintegration zu nennen. Für Zielfindungsaufgaben z.B. wäre der Einsatz von Potentialfeld-Methoden denkbar, deren Felder sich je nach Position des Roboters in einer zerklüfteten Umgebung verändern und die gewichteten Einflüsse des Ziels und der einzelnen Hindernisse berücksichtigen (siehe auch [Huang et al., 2006]).

Es bleibt anzumerken, dass aufgrund der fest gewählten Parameter der Roboter speziell auf seine Testumgebung eingestellt wurde. Interessant wäre daher, wie ein dynamisches Anpassen anhand verschiedener Lernalgorithmen den Roboter zur Navigation in unterschiedlichen Umgebungen befähigen würde.

Da es sich bei dem Khepera-Roboter um einen Roboter für den Einsatz in künstlichen Umgebungen handelt, wäre es von Interesse, die hier vorgestellten Verfahren auch in der natürlichen Umgebung anhand größerer Roboter zu testen. So könnte wie in [Zufferey and Floreano, 2005] beschrieben die Auswertung des optischen Flusses auch auf fliegende Roboter übertragen werden um z.B. das Einhalten einer konstanter Flughöhe anhand der Auswertung des optischen Flusses ähnlich der Wandverfolgung zu gewährleisten.

Literatur

- [Argyros et al., 2004] Argyros, A., Tsakiris, D., and Groyer, C. (2004). Biomimetic centering behavior [mobile robots with panoramic sensors]. *Robotics & Automation Magazine, IEEE*, 11(4):21–30,68.
- [Ballard, 1981] Ballard, D. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2).
- [Barron et al., 1994] Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77.
- [Bouguet, 2000] Bouguet, J.-Y. (2000). Pyramidal implementation of the lucas kanade feature tracker.
- [Braitenberg, 1993] Braitenberg, V. (1993). *Vehikel. Experimente mit kybernetischen Wesen*. Rowohlt Taschenbuch, Deutschland. Neudruck im LIT-Verlag, Münster (2004), ISBN 3-8258-7160-6.
- [Campbell, 2003] Campbell, N. A. (2003). *Biologie (Spektrum Lehrbuch)*. Spektrum Akademischer Verlag.
- [Chahl and Srinivasan, 1997] Chahl, J. S. and Srinivasan, M. V. (1997). Reflective surfaces for panoramic imaging. *Applied Optics*, 36(31):8275–8285.
- [Chapman, 1998] Chapman, R. F. (1998). *The Insects: Structure and Function*. Cambridge University Press, 4 edition.
- [Dudel et al., 1996] Dudel, J., Menzel, R., and Schmidt, R. F. (1996). *Neurowissenschaft. Vom Molekül zur Kognition (Springer-Lehrbuch)*. Springer-Verlag GmbH.
- [Franz and Mallot, 2000] Franz, M. O. and Mallot, H. A. (2000). Biomimetic robot navigation. *Robotics and Autonomous Systems*, 30(1-2):133–153.
- [Gluckman and Nayar, 1998] Gluckman, J. and Nayar, S. (1998). Egomotion and omnidirectional cameras. In *Computer Vision, 1998. Sixth International Conference on*, pages 999–1005.
- [Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151.
- [Hassenstein and Reichardt, 1956] Hassenstein, B. and Reichardt, W. (1956). Systemtheoretische Analyse der Zeit-, Reihenfolgen- und Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers *Chlorophanus*. *Z. Naturforsch.*, 11b:513–524.

- [Horn, 2003] Horn, B. (2003). Determining constant optical flow.
- [Huang et al., 2006] Huang, W. H., Fajen, B. R., Fink, J. R., and Warren, W. H. (2006). Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems*, pages 288–299.
- [Jähne et al., 1999] Jähne, B., Haußecker, H., and Geißler, P., editors (1999). *Handbook of Computer Vision and Applications*, volume I–III. Academic Press.
- [K-Team, 1994] K-Team (1994). *Khepera User Manual*, version 3.0 edition.
- [Krapp and Hengstenberg, 1996] Krapp, H. G. and Hengstenberg, R. (1996). Estimation of self-motion by optic flow processing in single visual interneurons. *Nature*, 384(6608):463–466.
- [Lucas and Kanade, 1981] Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proc. DARPA Image Understanding Workshop*, pages 121–130.
- [Mallot and Allen, 2000] Mallot, H. A. and Allen, J. S. (2000). *Computational Vision: Information Processing in Perception and Visual Behavior (Computational Neuroscience)*. The MIT Press.
- [Miles and Wallman, 1992] Miles, F. A. and Wallman, J. (1992). *Visual Motion and its Role in the Stabilization of Gaze*. Elsevier Science Pub Co.
- [Neumann, 2004] Neumann, T. R. (2004). *Biomimetic Spherical Vision. Biomimetische Algorithmen zum sphärischen Sehen*. Logos Berlin.
- [Serres et al., 2006] Serres, J., Ruffier, F., Viollet, S., and Franceschini, N. (2006). Toward optic flow regulation for wall-following and centring behaviours. *International journal of Advanced Robotic Systems*, 3(2):147–154.
- [Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600.
- [Sobel, 1990] Sobel, E. C. (1990). The locust’s use of motion parallax to measure distance. *J Comp Physiol [A]*, 167(5):579–588.
- [Srinivasan et al., 1999] Srinivasan, M. V., Chahl, J. S., Weber, K., Venkatesh, S., Nagle, M. G., and Zhang, S.-W. (1999). Robot navigation inspired by principles of insect vision. *Robotics and Autonomous Systems*, 26(2-3):203–216.
- [Stürzl, 2004] Stürzl, W. (2004). *Sensorik und Bildverarbeitung für Landmarken-basierte Navigation*. PhD thesis, Universität Tübingen.

- [Tsakiris and Argyros, 2000] Tsakiris, D. and Argyros, A. (2000). Corridor following by mobile robots equipped with panoramic cameras.
- [van Santen and Sperling, 1985] van Santen, J. P. and Sperling, G. (1985). Elaborated reichardt detectors. *J Opt Soc Am A*, 2(2):300–321.
- [Wallace, 1959] Wallace, G. (1959). Visual scanning in the desert locust *schistocerca gregaria*. *Journal of Experimental Biology*, 36:512–525. zitiert aus Srinivasan [4].
- [Weber et al., 1996] Weber, K., Venkatesh, S., and Srinivasan, M. (1996). Insect inspired behaviours for the autonomous control of mobile robots. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 1, pages 156–160vol.1.
- [Zufferey and Floreano, 2005] Zufferey, J.-C. and Floreano, D. (2005). Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2594–2599.