MOVIS

Über die Entwicklung eines ersten Prototypen einer Blindenbrille

A. Luo, W. Tao, S. Utcke, H. Burkhardt

27. April 1998

MOVIS

Über die Entwicklung eines ersten Prototypen einer Blindenbrille

A. Luo, W. Tao, S. Utcke*, H. Burkhardt*

Technische Universität Hamburg-Harburg
Technische Informatik I
21079 Hamburg

27. April 1998

*Jetzige Anschrift: Albert-Ludwigs-Universität
Institut für Informatik
Lehrstuhl für Mustererkennung und Bildverarbeitung
D-79085 Freiburg i. Br.
`http://www.informatik.uni-freiburg.de/~lmb/`

Zusammenfassung

Dieser Bericht beschreibt die Ergebnisse, die im Rahmen des Projektes MOVIS— **M**obiles **O**ptoelektronisches **V**isuell **I**nterpretatives **S**ystem für Blinde und Sehbehinderte — am Arbeitsbereich Technische Informatik I der Technischen Universität Hamburg Harburg in den Jahren 1994–1997 erarbeitet wurden.

Hierbei handelt es sich zum einen um ein neues Verfahren zur Erkennung von 3-D-Landmarken durch Integration von Tiefendaten und Farbinformation. Der pyramidale Ansatz, die Verwendung von Fuzzy-Regeln und geometrischem Hashing machen diesen Ansatz sowohl schnell als auch robust gegenüber teilweiser Verdeckung.

Zum anderen wird ein Verfahren zur Erkennung von Zebrastreifen vorgestellt. Dieses Verfahren basiert auf der Kombination von projektiver Geometrie und Statistik, insbesondere der konsequenten Berücksichtigung von möglichen Fehlerquellen und deren Auswirkung auf das Gesamt-Segmentierungsergebnis. Dadurch ist es gelungen, ein extrem robustes Verfahren zu entwickeln, das unempfindlich gegenüber starken Verdeckungen und Störungen im Objekt ist.

Ferner werden Voruntersuchungen zur Verkehrszeichenerkennung basierend auf Farbsegmentierung und geometrischen Eigenschaften präsentiert.

Alle drei Verfahren werden anhand realer Aufnahmen vorgestellt. Zusätzlich wird ein Demonstrator beschrieben, der im Rahmen einer Begehung durch Gutachter des DLR am 2. Oktober 1997 die Machbarkeit einer quasi Echtzeit-Erkennung von Briefkästen, Telefonzellen und Zebrastreifen sowie die sprachliche Ausgabe der Erkennungs-Resultate eindrucksvoll demonstriert hat.

Inhaltsverzeichnis

1 Problemstellung

Seit langem wird die Leistungsfähigkeit elektronischer Systeme als Navigationshilfe für Blinde und Ersatz für den traditionellen Langstock untersucht [51]. Klassische Beispiele sind z. B. diverse auf Ultraschall basierende Geräte, die jedoch ähnlich einem Langstock nur beschränkt, im wesentlichen zur Hinderniserkennung, eingesetzt werden können. Dagegen versprechen Fortschritte auf dem Gebiet des Rechnersehens Verbesserungen bis hin zur globalen und direkten Interpretation von Szenen und könnten es Blinden ermöglichen, z. B. Orte und Gegenstände des täglichen Gebrauchs (sogenannte generische oder spezifische Landmarken) zu erkennen, Wege zu finden oder Schilder zu lesen.

Die Realisierung eines Landmarkenfinders war das wesentliche Ziel in der ersten Arbeitsphase des Projektes MOVIS, kurz MOVIS I. In diesem Bericht werden die Ergebnisse dieser ersten Phase vorgestellt, soweit sie die an der Technischen Universität Hamburg-Harburg durchgeführten Arbeiten betreffen. Abschnitt 1.1 gibt einen Überblick über die Gliederung dieses Berichts.

Ein allgemeiner Lösungsansatz zur Erkennung beliebiger Objekte ist zur Zeit kaum oder nicht realisierbar, sicherlich nicht unter den vorgegebenen Rahmenbedingungen einer Quasi-Echtzeitverarbeitung. Dagegen deuten Fortschritte in der Bildverarbeitung der letzten Jahre darauf hin, daß ein zielorientiertes Vorgehen häufig sehr effiziente Lösungen ermöglicht. Aus diesem Grund wurden entsprechend den spezifischen Eigenschaften der zu erkennenden generischen Landmarken verschiedene Lösungsansätze entwickelt. Insbesondere haben wir die Landmarken Telefonzellen, Briefkästen, Zebrastreifen und Schilder näher untersucht, da eine Umfrage des Projekt-Partner empirica Delasasse unter Blinden-Verbänden [15] ergeben hatte, daß die Erkennung dieser Landmarken auf der Prioritätenliste der Blinden weit oben steht und die Sicherheit und Orientierungsfähigkeit von Blinden erheblich verbessern kann.

Bei allen Landmarken mit hoher Priorität in [15] handelt es sich um künstliche, von Menschenhand geschaffene Objekte. Die Struktur dieser Objekte ist typischerweise durch Begriffe wie Kollinearität, Parallelität und Periodizität geprägt, die unter perspektivischer Ansicht mit invarianten Merkmalen beschrieben werden können. Dadurch wird die Erkennung von Landmarken, unabhängig von der Blickrichtung, teilweise bereits aus einem monokularen Bild, ermöglicht. Ein typisches Beispiel sind die Streifen eines Fußgängerüberwegs. Die entsprechenden invarianten Ansätze zur Erkennung solcher Landmarken werden in diesem Bericht vorgestellt.

1.1 Gliederung

Im Folgenden werden in den Abschnitten 2 bis 4 die Untersuchungen und Ergebnisse vorgestellt, die im Rahmen des Projektes MOVIS an der Technischen Universität Hamburg Harburg, Arbeitsbereich Technische Informatik I, unter Leitung von Prof. Dr.-Ing. H. Burkhardt erzielt wurden.

Abschnitt 2 stellt ein Verfahren zur Erkennung von Zebrastreifen vor. Dieses Verfahren basiert auf einer Kombination aus projektiver Geometrie und Statistik, insbesondere der konsequenten Berücksichtigung möglicher Fehlerquellen und deren Auswirkung auf das Gesamt-Segmentierungsergebnis. Dadurch ist es gelungen, ein extrem robustes Verfahren zu entwickeln, das unempfindlich gegenüber starken Verdeckungen und Störungen im Objekt ist.

In Abschnitt 3 wird ein neues Verfahren zur Erkennung von 3-D-Landmarken durch Integration von Tiefendaten und Farbinformation vorgestellt. Der pyramidale Ansatz, die Verwendung von Fuzzy-Regeln und geometrischem Hashing machen diesen Ansatz sowohl schnell als auch robust gegenüber teilweiser Verdeckung.

Schließlich werden in Abschnitt 4 Voruntersuchungen zur Verkehrszeichenerkennung basierend auf Farbsegmentierung und geometrischen Eigenschaften vorgestellt. Im Gegensatz zu den in den Abschnitten 2 und 3 vorgestellten Verfahren handelt es sich hierbei noch nicht um ein im wesentlichen ausgereiftes Verfahren, bildet aber eine wertvolle Grundlage für die ursprünglich in einer zweiten Phase geplante Erkennung von Zeichen und Piktogrammen.

Alle drei Verfahren werden anhand von realen Aufnahmen vorgestellt. Darüber hinaus beschreibt Abschnitt 5 die Online-Demonstration vom 2. Oktober 1997, in der vor Gutachtern der DLR eindrucksvoll die Wirkungsweise und volle Funktionalität der in den Abschnitten 2 und 3 beschriebenen Verfahren inklusive einer Sprachausgabe der Detektionsergebnisse demonstriert wurde.

Abschnitt 6 faßt noch einmal die Ergebnisse zusammen und gibt einen Ausblick auf mögliche Richtungen zur Fortsetzung des Projektes.

2 Erkennung von 2-D-Landmarken

2.1 Einleitung

In diesem Abschnitt geht es um die Erkennung zweidimensionaler Landmarken am Beispiel der Erkennung von Zebrastreifen. Wie eine Umfrage des Projekt-Partners empirica Delasasse unter Blinden-Verbänden [15] ergeben hatte, spielt die Möglichkeit der Erkennung von Zebrastreifen eine bedeutende Rolle für die Fähigkeit Blinder, sich in nicht- oder nur geringvertrauter Umgebung zurechtzufinden.

Aufgrund ihrer geringen Farbigkeit und der damit verbundenen geringen Signalwirkung, sowie der rein planaren Ausdehnung unterscheiden sich Zebrastreifen in ihren Eigenschaften signifikant sowohl von den in Abschnitt 3 untersuchten dreidimensionalen Landmarken, als auch von den in Abschnitt 4 untersuchten Verkehrszeichen. Aus diesem Grund, und um die für diese Anwendung erforderliche hohe Zuverlässigkeit zu erreichen, wurde bei der Erkennung von Zebrastreifen ein eigener Weg gegangen. Dabei werden Zebrastreifen in monokularen Bildern einzig aufgrund ihrer geometrischen Eigenschaften erkannt (vgl. Abschn. 2.2). Die erforderliche Zuverlässigkeit und Robustheit liefert die konsequente Anwendung der Fehlerfortpflanzung (siehe Abschn. 2.3). Die nötigen Bearbeitungsschritte lassen sich in Low-Level Routinen wie Kantenextraktion, Kantenverfolgung und Linienfindung (Abschn. 2.4) sowie High-Level Routinen, welche die eigentliche Logik enthalten, unterteilen (Abschn. 2.5). Besonders soll hier auf eine neue Methode zur schnellen Berechnung des Doppelverhältnisses in Abschnitt 2.5.4 hingewiesen werden. Abschnitt 2.6 zeigt Ergebnisse auf realen Straßenszenen.

2.2 Projektive Geometrie

Ein Zebrastreifen läßt sich vereinfacht als eine Anordnung von aufeinanderfolgenden, parallelen hellen Streifen auf dunklem Untergrund beschreiben, wobei alle Streifen eine konstante Breite und einen konstanten Abstand untereinander haben. Diese Aussage gilt für nahezu alle Länder, und tatsächlich regelt das Internationale Übereinkommen über Straßenverkehrszeichen [47] von 1968: „Um Fußgängerüberwege zu markieren, sind vorzugsweise ziemlich breite Streifen parallel zur Fahrbahnachse anzubringen“. Darüber hinaus regelt in der Bundesrepublik Deutschland die ZTV-M 84 [9] das mögliche Aussehen von Zebrastreifen bis hin zur Breite der Streifen und Abstände (jeweils $500\,\mathrm{mm} \pm 5\,\mathrm{mm}$ bei aufgelegten und $500\,\mathrm{mm} \pm 10\,\mathrm{mm}$ bei eingelegten Streifen) und deren Minimallänge (mindestens $3\,\mathrm{m}$, außer bei angeschnittenen Streifen). Identische oder zumindest ähnliche Regelungen existieren für fast alle übrigen Länder.

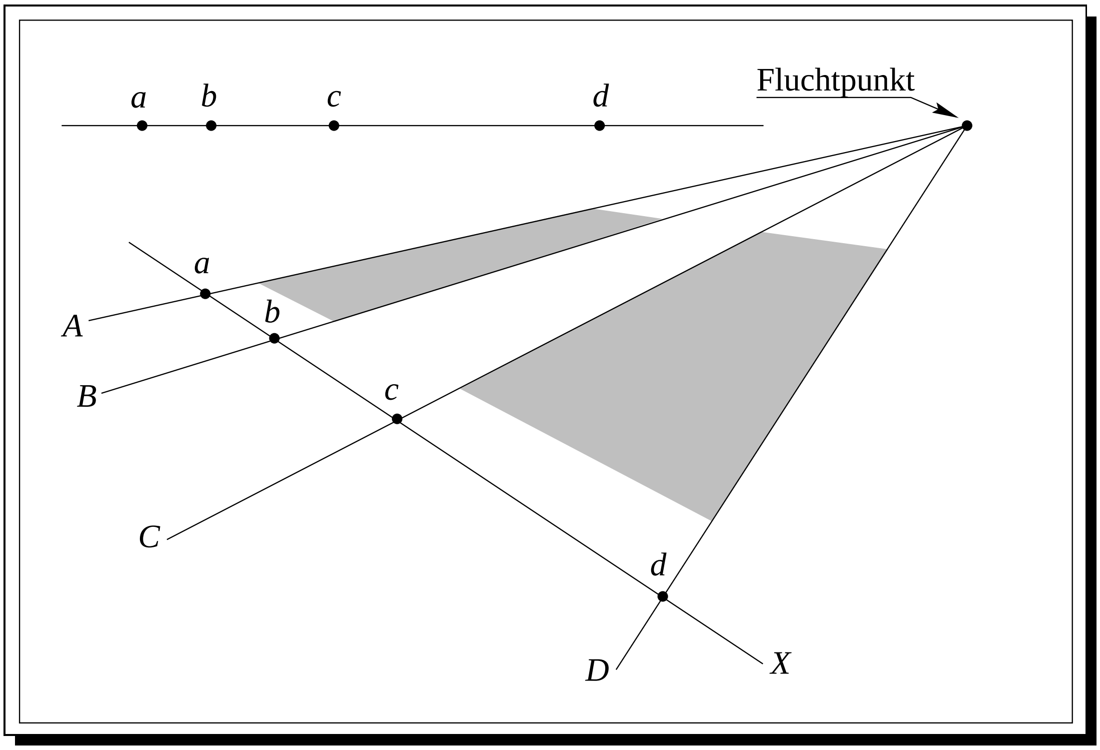


Abbildung 1: *Fluchtpunkt und Doppelverhältnis*

Was kann man aus dieser Beschreibung über das generelle Erscheinungsbild von Zebrastreifen in projektiven Aufnahmen ableiten? Zum einen ist aus der projektiven Geometrie bekannt, daß sich im Original parallele Linien in Projektionen in einem Punkt schneiden [40]. Dieser Punkt wird Fluchtpunkt genannt und kann unter Umständen im Unendlichen liegen. Weiterhin kann man aus den bekannten Breiten der Streifen und der Lücken zwischen den Streifen ein sogenanntes Doppelverhältnis bestimmen (siehe Abb. 1), das invariant unter beliebigen projektiven Transformationen ist [40]. Das Doppelverhältnis ist definiert als

$$[abcd] = \frac{\overline{ac}}{\overline{bc}} \frac{\overline{bd}}{\overline{ad}}$$

Hierbei ist $\overline{ac}$ der euklidische Abstand zwischen den Punkten a und c. Alternativ wird das Doppelverhältnis oft in der Form

$$[abcd] = [ABCD] = \frac{|ACX|}{|ABX|} \frac{|BDX|}{|ADX|} \tag{1}$$

geschrieben. Hierbei sind A, B, C, D und X Linien in homogenen Koordinaten und $|ACX|$ die Determinante einer 3×3 Matrix, deren Spalten von den 3 Vektoren A, C und X gebildet werden. Die Linie X kann theoretisch beliebig gewählt werden, solange sie nicht durch den gemeinsamen Fluchtpunkt der 4 anderen

Linien geht. Wir werden jedoch im Abschnitt 2.5 sehen, daß die geschickte Wahl des Vektors X signifikant das Ergebnis beeinflussen kann.

Da die Geometrie von Zebrastreifen bekannt ist[1], ist es somit leicht möglich, eine aus 4 aufeinanderfolgenden, sich in einem Punkt schneidenden Linien bestehende Zebrastreifen-Hypothese zu testen. Darüber hinaus ist es bei bekanntem Doppelverhältnis $[abcd]$ möglich, die Gleichung (1) nach einem der Vektoren A, B, C oder D aufzulösen. Auf diese Weise kann die Lage weiterer Streifen prognostiziert und dann im Bild verifiziert werden. Insbesondere ist es auf diese Weise möglich, die Bildposition einer Linie im Unendlichen vorauszusagen. Diese Linie entspricht dem Horizont und kann benutzt werden, um eine Reihe weiterer Verifikationsschritte zu implementieren (siehe Abschn. 2.5).

Es hat sich gezeigt, daß es möglich ist, Zebrastreifen zuverlässig nur anhand ihrer Geometrie zu erkennen. Obwohl es theoretisch möglich und zu erwarten ist, daß in seltenen Einzelfällen False-Positives beobachtet werden, ist dieses in über 1000 Tests nicht einmal aufgetreten.

2.3 Fehlerfortpflanzung

In Abschnitt 2.2 wurde das der Erkennung zugrundeliegende Modell eines Zebrastreifens vorgestellt. Leider zeigt sich bei Untersuchungen an realen Aufnahmen relativ schnell, daß die vereinfachten und idealisierten Annahmen des Abschnitts 2.2 nur in den seltensten Fällen zutreffen. Dabei lassen sich Fehler vor allem auf zwei mögliche Ursachen zurückführen.

Eine wichtige Fehlerursache ist im vereinfachten Modell selber zu suchen. Straßen sind in aller Regel nicht planar, sondern zu den Straßenrändern hin abgesenkt, um den ungehinderten Abfluß von Regenwasser zu erlauben. Die Streifen selber sind in den seltensten Fällen wirklich parallel, trotz der strengen Auflagen der ZTV-M84 [9]. Dazu kommt, daß Zebrastreifen nach einigen Jahren der Benutzung oft erhebliche Schäden sowohl in der Fläche als auch am Rand aufweisen.

Eine zweite Fehlerursache liegt im Abbildungsprozeß, und hier vor allem in der Apertur-Funktion der Kamera und der mit dem Abbildungsvorgang untrennbar verknüpften Abtastung des Bildes. Beide zusammen bewirken, daß sich die Position eines Kantenpunktes nur innerhalb eines gewissen Fehlerintervalls bestimmen läßt, das selbst bei subpixelgenauen Methoden je nach Kamera in der Größenordnung eines Drittel-Pixels liegen kann. Dies gilt insbesondere für eine 1-Chip Farb-Kamera, wie sie im MOVIS Online-Demonstrator Verwendung findet (vgl. auch Abschn. 4.5). Zu zusätzlichen Fehlern kommt es durch die falsche Gruppierung von Kantenpunkten zu Liniensegmenten, sogenannten Outliern.

[1] Das Doppelverhältnis ist 4/3.

Die oben genannten Gründe führen dazu, daß sich 4 Linien in aller Regel niemals in einem Punkt schneiden (selbst wenn die Theorie sagt, daß sie es sollten), und daß das Doppelverhältnis von 4 Linien nahezu nie 4/3 ist (obwohl die Theorie sagt, daß es so sein müßte). Es ist von daher erforderlich, gewisse Abweichungen der berechneten von den gewünschten Werten zuzulassen. In der Bildverarbeitung und Mustererkennung ist es zu diesem Zweck generell üblich, Schwankungen bis zu einem festen Wert, dem sogenannten Threshold oder Schwellwert, zu erlauben. Dabei liegt die Kunst in der Bildverarbeitung oftmals mehr in der problemangepaßten Wahl dieses Schwellwertes als im Entwurf komplexer Algorithmen; wird der Wert zu klein gewählt, kommt es zu einer Häufung von sogenannten False Negatives, Objekte werden übersehen, während eine zu großzügige Wahl zu einer Häufung von False Positives führt, Objekte werden auch dort detektiert, wo in Wirklichkeit im Bild gar keine zu sehen sind.

Im Falle der Zebrastreifenerkennung hat es sich schon früh gezeigt, daß eine feste Wahl des Schwellwertes unmöglich sein würde. Dies liegt vor allem in der Tatsache begründet, daß nicht von vornherein bekannt ist, welche Größe der Zebrastreifen im Bild haben wird, und aus welcher Richtung er zu sehen ist. Die Konsequenz ist, daß Linien mit mehreren 100 Pixeln Länge (und entsprechend hoher Genauigkeit) genauso berücksichtigt werden müssen wie Linien, deren Länge nicht einmal einige Dutzend Pixel beträgt, und daß der Abstand zwischen zwei Linien genausogut 2 wie 20 Pixel betragen kann.

Aus diesem Grunde hat es sich als essentiell erwiesen, von Anfang an mögliche Fehlerquellen in das Modell miteinzubeziehen und zu modellieren. Dadurch wird es zum einen möglich, die Primitiven gemäß Ihrer Glaubwürdigkeit zu gewichten, so daß z. B. bei der Berechnung des gemeinsamen Fluchtpunktes von 4 Linien diese unterschiedlich zur Berechnung beitragen. Zum anderen ist es jederzeit möglich, die Güte eines Zwischenergebnisses objektiv zu beurteilen.

In der Praxis hat es sich nahezu durchweg als ausreichend erwiesen, den Fehler als Gauß-verteilt zu modellieren und bei der Fehlerfortpflanzung mit der linearisierten Gleichung zu arbeiten. Abschnitt 2.3.1 beschreibt die bei der Berechnung der Fehlerfortpflanzung benutzen Mechanismen, und Abschnitt 2.3.2 das verwandte χ^2 Fehlermaß.

2.3.1 Fehlerfortpflanzung im linearisierten Fall

Ein simples Beispiel soll zeigen, warum es notwendig sein kann, zusammen mit einem Zwischenergebnis auch dessen Wahrscheinlichkeitsverteilung zu bestimmen. Angenommen, die Linie durch zwei Punkte soll berechnet werden. Wenn es sich bei den beiden Punkten nicht um perfekte Punkte, sondern z. B. um gemessene Bildkoordinaten handelt, dann ist offensichtlich, daß zwei unterschiedli-

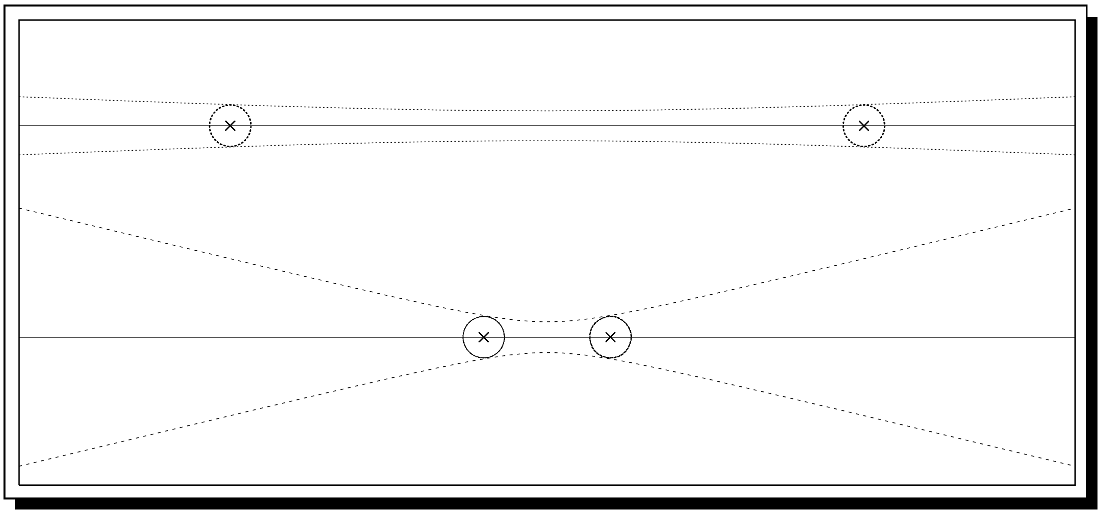


Abbildung 2: *Wahrscheinlichkeitsverteilung einer Linie durch zwei Punkte in Abhängigkeit von den Wahrscheinlichkeitsverteilungen der Punkte.*

che Messungen niemals die genau gleichen Koordinaten zurückgeben und infolgedessen auch niemals dieselbe Linie. Stattdessen werden die Koordinaten der beiden Punkte einer gewissen Wahrscheinlichkeitsverteilung unterliegen, und in Abhängigkeit davon wird sich auch für die Linie eine bestimmte Wahrscheinlichkeitsverteilung einstellen. Abbildung 2 zeigt diesen Effekt für zwei unterschiedliche Konstellationen der Punkte. Es ist offensichtlich, daß die Genauigkeit, mit der sich die Linie berechnen läßt, maßgeblich nicht nur von der Genauigkeit der gemessenen Koordinaten, sondern zusätzlich von deren Lage zueinander abhängt.

Wie kann man nun allgemein die Verteilung berechneter Größen bestimmen? Genauer, wenn die Verteilung $\mathbf{p}(\mathbf{x})$ einer Zufallsvariable $\mathbf{x} \in \mathbb{R}^n$ bekannt ist (in unserem Beispiel die Koordinaten der Punkte), wie kann man dann die Verteilung einer abhängigen Zufallsvariablen $\mathbf{y} = \mathbf{f}(\mathbf{x})$, $\mathbf{y} \in \mathbb{R}^m$ bestimmen, in unserem Beispiel die der Linie? Leider ist es nicht möglich, eine analytische Antwort für beliebige Verteilungen $\mathbf{p}(\mathbf{x})$ oder Funktionen $\mathbf{f}(\mathbf{x})$ zu geben. Für den häufigen Fall, daß $\mathbf{x}$ Gauß-verteilt (Kovarianzmatrix $\mathtt{K}_\mathbf{x}$) und $\mathbf{f}(\mathbf{x})$ eine lineare Funktion ist, ist das Ergebnis jedoch bekannt; $\mathbf{y}$ ist dann ebenfalls Gauß-verteilt und hat die Kovarianzmatrix

$$\mathtt{K}_\mathbf{y} = \mathtt{J}_\mathbf{y}\mathtt{K}_\mathbf{x}\mathtt{J}_\mathbf{y}^\mathsf{T} \tag{2}$$

Hierbei ist $\mathtt{J}_\mathbf{y}$ die Jacobi-Matrix von $\mathbf{y} = \mathbf{f}(\mathbf{x})$ bezüglich $\mathbf{x}$. Im hier betrachteten Fall einer linearen Funktion ist die Jacobi-Matrix natürlich trivial. Für das Beispiel in Abbildung 2 ergibt sich ein hyperbelförmiges Vertrauensintervall.

Komplizierter wird der Sachverhalt, wenn $\mathbf{y} = \mathbf{f}(\mathbf{x})$ keine lineare Funktion mehr ist. Um dennoch Gleichung (2) anwenden zu können ist es nötig, $\mathbf{f}(\mathbf{x})$ zuerst zu linearisieren. Dies geschieht durch Taylorentwicklung und Abbruch nach dem ersten Glied. Es ergibt sich

$$\mathbf{y} = \mathbf{g}(\mathbf{x}_0 + \Delta\mathbf{x}) = \mathbf{g}(\mathbf{x}_0) + \mathtt{J}_{\mathbf{g}(\mathbf{x}_0)}\Delta\mathbf{x} + \mathcal{O}(||\Delta\mathbf{x}||^2) \tag{3}$$

Selbstverständlich handelt es sich in diesem Fall bei Gleichung (2) nicht mehr um eine exakte Beziehung. Wie gut diese Näherung ist, hängt sowohl von der jeweiligen Form der Funktion $\mathbf{g}(\mathbf{x})$ als auch von der Kovarianzmatrix $\mathtt{K}_\mathbf{x}$ ab; je kleiner die Elemente in $\mathtt{K}_\mathbf{x}$ — der Fehler — sind, desto geringer ist der Fehler in Gleichung (2). Aus diesem Grund wird oft von der Näherung des kleinen Fehlers gesprochen. Der Fehler läßt sich abschätzen, indem eine Obergrenze für $\mathcal{O}(||\Delta\mathbf{x}||^2)$ berechnet wird, hierfür existieren diverse Näherungsformeln.

Gleichung (2) läßt sich selbst dann anwenden, wenn eine Lösung für $\mathbf{y}_0$ durch Minimierung gefunden wurde, d. h. die Funktion $\mathbf{g}(\mathbf{x})$ und damit auch ihre Jacobi-Matrix $\mathtt{J}_{\mathbf{g}(\mathbf{x})}$ sind nicht explizit bekannt. Bekannt ist nur die Kostenfunktion $C(\mathbf{x}, \mathbf{y})$, die minimiert wurde. Die notwendige Bedingung für ein Minimum ist $\frac{\partial C(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}} = 0$, und mit Hilfe des Satzes der impliziten Differentiation [12, 16] läßt sich die Jacobi-Matrix einer *unbekannten* Funktion $\mathbf{y} = \mathbf{g}(\mathbf{x})$ berechnen als

$$\mathtt{J}_{\mathbf{g}(\mathbf{x})} = -\left(\frac{\partial^2 C(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}^2}\right)^{-1} \left(\frac{\partial^2 C(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x} \partial \mathbf{y}}\right)^{\mathsf{T}} \tag{4}$$

Falls die Hesse-Matrix $\mathtt{H} = \frac{\partial^2 C(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}^2}$ im Minimum $\mathbf{y}_0$ invertierbar ist. Unter Benutzung vom Lagrangeschen Multiplikatoren läßt sich (4) auch noch auf Minimierungsaufgaben mit Nebenbedingungen anwenden.

Interessant ist, daß man mit Hilfe von Gleichung (2) selbstverständlich auch dann die Verteilung eines Ergebnisses berechnen kann, wenn der zur Berechnung des Ergebnisses genutzte Algorithmus nur eine Näherung statt des korrekten Ergebnisses berechnet. So ist es zum Beispiel möglich, $\mathbf{y}_0$ mit Hilfe eines beliebigen, suboptimale Algorithmus zu bestimmen und trotzdem die Verteilung $\mathtt{K}_\mathbf{y}$ zu bestimmen. Es sollte jedoch darauf hingewiesen werden, daß suboptimale Algorithmen oftmals einen beträchtlichen Bias bewirken oder die Kovarianzmatrix künstlich aufblähen können.

2.3.2 Das χ^2 Fehlermaß

Dieser Abschnitt gibt eine kurze Einführung in Anpassungstests, insbesondere den im Rahmen von MOVIS verwandten χ^2-Anpassungstest. Weitergehende Informationen finden sich zum Beispiel in [8] oder beliebigen Einführungen in die mathematische Statistik.

In Algorithmen der Bildverarbeitung stellt sich häufig die Frage: „Glaube ich, daß der beobachtete Wert dem erwarteten Wert entspricht?". Solche Fragen[2] beantworten Anpassungstests. Wenn es sich bei der zugrundeliegenden Verteilung um eine Gauß-Verteilung handelt, wird man häufig den χ^2-Anpassungstest verwenden. Seine Anwendung ist einfach. Sie wird weiter unten am Beispiel zweier Linien erklärt.

Angenommen, zwei Linien sind durch ihren Winkel θ mit der x-Achse und ihren Abstand d vom Ursprung parametrisiert[3], $\ell_i = (\theta_i, d_i)^\mathsf{T};\ i = 1, 2$, und ihre Kovarianzmatrix ist durch $\mathtt{K}_\ell \in I\!R^{2\times 2}$ gegeben. Einfach ausgedrückt beantwortet der χ^2-Anpassungstest die Frage: „Bin ich zu $p\,\%$ überzeugt, daß der Unterschied zwischen den beiden Linien ℓ_1 und ℓ_2 auf zufällige Schwankungen zurückzuführen ist, wie sie durch $\mathtt{K}_\ell$ beschrieben werden?" indem der Ausdruck

$$(\ell_1 - \ell_2)^\mathsf{T} \mathtt{K}_\ell^{-1} (\ell_1 - \ell_2) \leq \chi^2_{p,2} \tag{5}$$

ausgewertet wird. Dabei ist 2 die Anzahl der Freiheitsgrade (θ und d in unserem Beispiel).

2.4 Low-Level Routinen

Dieser Abschnitt beschreibt die Routinen der Bildvorverarbeitung, wie sie im Rahmen des Zebrastreifenfinders im Projekt MOVIS verwandt werden. Hierbei handelt es sich überwiegend um Standardalgorithmen. Im Folgenden werden die Algorithmen zur subpixelgenauen Kantenpunktextraktion (Abschnitt 2.4.1), zur Konturverfolgung und gleichzeitigen Hysterese (Abschnitt 2.4.2) und zur Linienextraktion (Abschnitt 2.4.3) vorgestellt.

2.4.1 Subpixelgenaue Kantenpunktextraktion

Als Algorithmus zur subpixelgenauen Kantenpunktextraktion wird eine simplifizierte Version des Canny Kantenpunktfinders benutzt [11]. Im Gegensatz zu [11] werden Kantenpunkte hier nur auf einer Auflösungsebene bestimmt und die Ableitungen nur in x- und y-Richtung berechnet.

Beim Canny-Kantenpunktfinder in der hier verwandten einfachen Form handelt es sich um eine Tiefpaßfilterung mit einem normierten Gauß-Kern der Standardabweichung σ und anschließende Ableitung in x- und y-Richtung. Der Parameter σ bestimmt zum einen die Ortsauflösung des Kantenpunktfinders (je kleiner σ

[2]Oder genauer die Frage: „Habe ich keinen Grund zu der Annahme, daß der beobachtete Wert nicht dem erwarteten Wert entspricht".

[3]Tatsächlich hat eine derartige Parametrisierung erhebliche Nachteile und wurde nur gewählt, um ein einfaches Beispiel zu bieten.

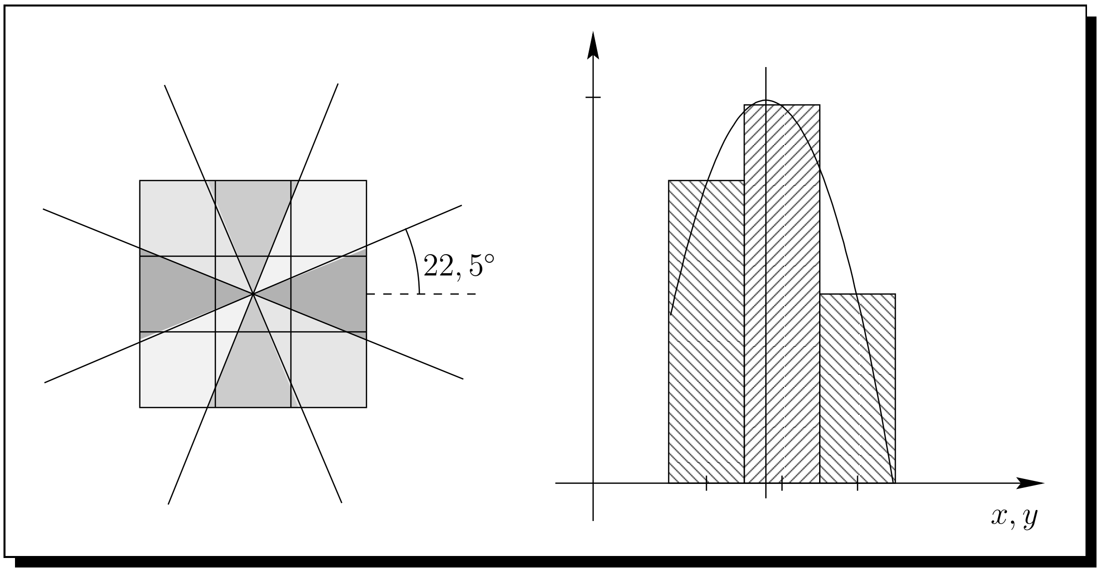


Abbildung 3: *Non-Maximum-Suppression.*

ist, desto besser ist die Ortsauflösung), und zum anderen die Störanfälligkeit der Ableitung (je größer σ ist, desto robuster ist der Algorithmus). Damit hängt die Wahl eines geeigneten Wertes σ theoretisch stark von der Qualität des zugrundeliegenden Bildmaterials sowie der Ortsfrequenz des zu findenden Zebrastreifens ab. Ausgiebige Versuche haben jedoch gezeigt, daß der tatsächliche Einfluß von σ auf die Erkennungsrate über einen relativ großen Bereich gering ist, $\sigma = 1.0\,\mathrm{pxl}$ ist typischerweise ein geeigneter Wert [2].

Anschließend findet eine non-Maximum-Suppression statt, wie sie auch in [11] beschrieben wird. Dazu wird jeder Wert (Betrag der Ableitung) mit seinen beiden direkten Nachbarn in Richtung der größten Steigung verglichen (siehe Abbildung 3). Ist einer der beiden Nachbarn größer, so wird der Kantenpunkt verworfen, andernfalls wird eine Parabel durch die drei Punkte gelegt und der mittlere Kantenpunkt durch Ort und Größe des neuen Maximums ersetzt. Das Ergebnis ist ein Kantenbild, in dem im Gegensatz zur reinen Ableitung nur noch die Maxima eingetragen sind, und in dem Kantenpunkte an beliebiger Position außerhalb des Pixelraster liegen können.

2.4.2 Konturfolger und Hysterese

Das Ergebnis der Kantenpunktdetektion ist eine im wesentlichen ungeordnete Menge einzelner Kantenpunkte. Es entspricht jedoch dem menschlichen Verständnis von Kanten, mehrere dieser Kantenpunkte zu Konturzügen (Kanten) zu verbinden. Eine ausführliche Beschreibung des verwendeten Konturfolgers findet sich

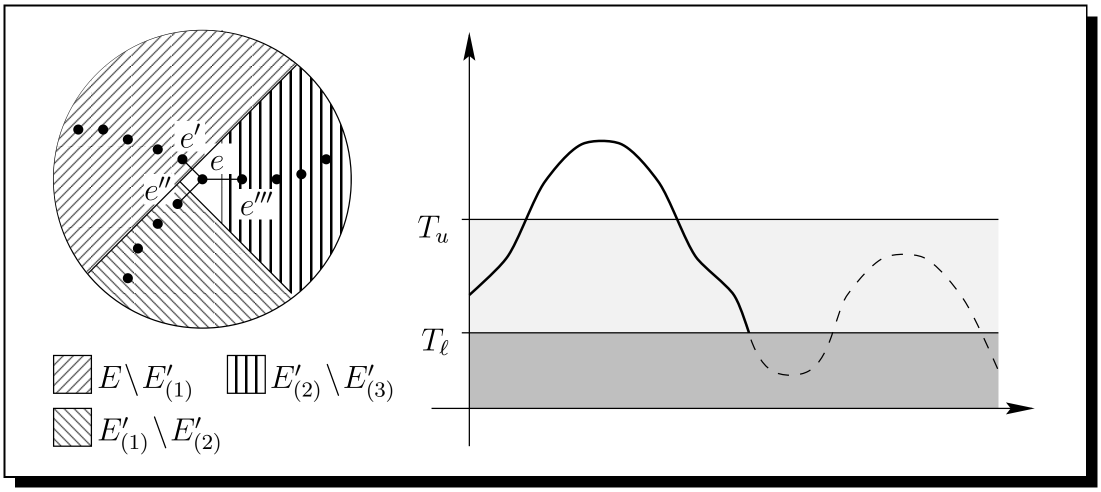


Abbildung 4: *Konturfolger und Hysterese*

in [6, 28]. Zeitgleich mit der Konturverfolgung kann die in [11] beschriebene Hysterese durchgeführt werden.

Wesentlicher Parameter für die Konturverfolgung ist der Nachbarschaftsradius r. Zwei Kantenpunkte werden nur dann als benachbart betrachtet, falls ihr Euklidischer Abstand maximal r beträgt. Je größer dieser Radius gewählt wird, desto größer ist die Chance, eventuelle Lücken in der Kontur zu schließen, desto größer ist aber auch die Gefahr, Verbindungen herzustellen, zu denen es im Bild keine Entsprechungen gibt. In der praktischen Anwendung hat sich für r ein Wert $2 \leq r \leq 2\sqrt{2}$ als sinnvoll erwiesen. Zusätzlich sind der obere und untere Schwellwert für die Hysterese, T_u und T_ℓ, anzugeben.

In einem ersten Schritt werden alle Kantenpunkte aus der Menge E der zu betrachtenden Kantenpunkte entfernt, deren Betrag (der Ableitung) unter dem unteren Schwellwert T_ℓ liegt. Daran anschließend folgt für jeden Kantenpunkt e die Bestimmung aller seiner Nachbarn. Dazu werden zunächst alle Kantenpunkte ermittelt, die sich innerhalb eines Kreises mit Radius r um den Kantenpunkt e befinden. Diese Kantenpunkte werden zur Menge der Kantenpunkte E' zusammengefaßt. Der Kantenpunkt e' mit dem kleinsten Abstand zu e ist automatisch ein direkter Nachbar von e. Aus der Menge der möglichen Nachbarn E' werden nun alle die Kantenpunkte entfernt, deren Abstand zu e' kleiner ist als zu e. Unter den verbleibenden Kantenpunkten ist wieder der Kantenpunkt e'' mit dem jetzt kleinsten Abstand zu e ein direkter Nachbar von e, und der Algorithmus startet von vorne, solange, bis die Menge der Kantenpunkte E' leer ist. Abbildung 4 macht dies noch einmal anschaulich deutlich.

Das Ergebnis dieses Vorgehens ist ein ungerichteter, planarer Graph. Je nach Anzahl der Nachbarn unterscheiden wir Subkonturen — jeder Kantenpunkt hat genau zwei Nachbarn — und Knoten — ein Kantenpunkt hat entweder einen oder mindestens 3 Nachbarn. In einem letzten Schritt werden jetzt all die Subkonturen verworfen, die nicht mindestens einen Kantenpunkt mit einer Kantenstärke größer T_u enthalten, sowie alle Knoten, die hierdurch isoliert würden. Vergleiche hierzu auch Abbildung 4.

2.4.3 Linienfinder

Nachdem alle Konturen identifiziert wurden, kann jede Kontur daraufhin untersucht werden, ob sie ein oder mehrere Geradenstücke enthält. Hierbei kommt eine modifizierte Version des in [37] beschriebenen iterativen Algorithmus zum Einsatz. Im Folgenden wird eine kurze Einführung in den Original-Algorithmus gegeben sowie die vorgenommenen Änderungen beschrieben.

In [37] wird die Gerade durch n Punkte $\mathbf{x}_i$, $i = 1 \ldots n$ bestimmt als die Gerade durch den Schwerpunkt aller Pixel $\bar{\mathbf{x}} = (\bar{x}, \bar{y})^\mathsf{T} = \frac{1}{n}\sum_{i=1}^{n}(x_i, y_i)^\mathsf{T}$ und mit dem Eigenvektor zum kleineren Eigenwert $\lambda_{\min}$ von (6) als Normalenvektor,

$$\frac{1}{n}\sum_{i=1}^{n}\begin{pmatrix} x_i^2 - \bar{x}^2 & x_i y_i - \bar{x}\bar{y} \\ x_i y_i - \bar{x}\bar{y} & y_i^2 - \bar{y}^2 \end{pmatrix} = \frac{1}{n}\sum_{i=1}^{n}\begin{pmatrix} (x_i - \bar{x})^2 & (x_i - \bar{x})(y_i - \bar{y}) \\ (x_i - \bar{x})(y_i - \bar{y}) & (y_i - \bar{y})^2 \end{pmatrix} \tag{6}$$

Der Eigenwert $\lambda_{\min}$ ist der mittlere quadratische Abstand der Kantenpunkte von der Geraden.

Es ist offensichtlich, daß sich mit Hilfe von (6) sehr leicht ein iterativer Algorithmus entwerfen läßt. Dazu wird probeweise eine Gerade durch eine kleine Anzahl von Punkten (typischerweise 13 in [37], die sogenannte Seed-Region) gelegt. Liegt der resultierende Fehler $\lambda_{\min}$ oberhalb eines vorgegebenen Schwellwertes, so wird der 1 Kantenpunkt der Seed-Region verworfen und durch den $n+1$ ten Kantenpunkt ersetzt. Liegt andererseits $\lambda_{\min}$ unterhalb des Schwellwertes, so wird ein weiterer Kantenpunkt hinzugenommen und der Algorithmus solange iteriert, bis der Schwellwert überschritten ist.

Im Rahmen von MOVIS wurden fünf wesentliche Anpassungen dieses Algorithmus vorgenommen. Zum einen unterscheiden sich die zugrundeliegenden Graphen. Wird in [37] von verzweigungsfreien Konturen ausgegangen, so dienen im Rahmen von MOVIS teilweise komplexe Graphen als Ausgangsmaterial (vgl. Abschn. 2.4.2). Aus diesem Grund wurde der Algorithmus so abgeändert, daß bei erreichen eines Knoten die Gerade in alle möglichen Richtungen ausgedehnt und letztendlich die Richtung gewählt wird, in der die Gerade am längsten fortgesetzt werden kann.

Ein zweiter Unterschied liegt in der Behandlung der Orientierung von Kantenpunkten. Im ursprünglichen Algorithmus werden Orientierungen Modulo π betrachtet, d. h. es macht keinen Unterschied, ob eine Kante einem Übergang von Dunkel nach Hell entspricht oder umgekehrt. Dies bedeutet insbesondere z. B., daß bei einem Schachbrett pro Reihe (und Spalte) eine durchgehende Linie gefunden würde.

Im Gegensatz dazu ist für die Zebrastreifenerkennung die Orientierung der gefundenen Linien signifikant Modulo 2π. Aus diesem Grund mußte sichergestellt werden, daß Geraden grundsätzlich niemals über eine sprunghafte Änderung der Richtung hinübergehen. Dies ist leicht durch eine Abfrage der Orientierung eines jeden Kantenpunktes zu implementieren.

Der dritte Unterschied liegt in der Behandlung des Abbruchkriteriums und der daraus resultierenden minimal-Länge der Seed-Region. In [37] wurde hierfür ein konstanter Schwellwert gewählt. Hierdurch ergibt sich das Problem einen Schwellwert zu wählen, der sowohl mit einigen wenigen Kantenpunkten — der Seed-Region — zu sinnvollen Ergebnissen führt, als auch Linien mit mehreren Hundert Kantenpunkten erkennen kann. Dabei kann es sich immer nur um einen Kompromiß handeln, und typische Probleme sind zum einen, daß die Seed-Region bereits relativ viele Punkte enthalten muß, sehr kurze Geradenstücke folglich gar nicht gefunden werden können, und zum anderen, daß die Seed-Region oft bereits an nicht ganz gerade Konturenstücke gefittet wird, was die Ungenauigkeiten in den Parametern der letztendlich gefundenen Gerade teilweise erheblich erhöhen kann.

Beide Probleme lassen sich umgehen, indem statt eines festen Thresholds ein Threshold gewählt wird, dessen Höhe von der Anzahl der verwandten Kantenpunkte abhängt. Hierzu bietet sich der χ^2-Anpassungstest förmlich an, die auszuwertende Gleichung ist

$$\lambda'_{\min} = \frac{1}{n} \sum_{i=1}^{n} \frac{\ell^{\mathsf{T}} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}} \ell}{\sigma_{\mathbf{x}}^2} \leq \chi^2_{p,n-2} \tag{7}$$

Wie in Gleichung (5) ist p die gewünschte „Sicherheit“ und $n-2$ die Anzahl der Freiheitsgrade.

Tatsächlich stecken in Gleichung (7) bereits einige vereinfachende Annahmen, die tatsächlich auszuwertende Gleichung ist

$$\lambda'_{\min} = \frac{1}{n} \sum_{i=1}^{n} \frac{\ell^{\mathsf{T}} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}} \ell}{\ell^{\mathsf{T}} \mathtt{K}_{\mathbf{x}_i} \ell} \leq \chi^2_{p,n-2} \tag{8}$$

Leider gibt es für Gleichung (8) i. a. keine geschlossene Lösung, es sei denn, die Kovarianzmatrix $\mathtt{K}_{\mathbf{x}_i}$ ist unabhängig von $\mathbf{x}_i$ und hat — wie in Gleichung (7) zugrundegelegt — die Form

$$\mathtt{K}_{\mathbf{x}_i} = \begin{pmatrix} \sigma_{\mathbf{x}}^2 & 0 & 0 \\ 0 & \sigma_{\mathbf{x}}^2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Tatsächlich ergeben sich aus der subpixelgenauen Kantenberechnung des Canny jedoch stark unterschiedliche Varianzen parallel ($\sigma_\parallel^2$) und senkrecht ($\sigma_\perp^2$) zur Kante, mit $\sigma_\perp^2 \ll \sigma_\parallel^2$, d. h.

$$\mathtt{K}_{\mathbf{x}_i} = R^{-\mathsf{T}} \begin{pmatrix} \sigma_\parallel^2 & 0 & 0 \\ 0 & \sigma_\perp^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} R^{-1}$$

mit

$$R = \begin{pmatrix} \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

α_i ist der Winkel zwischen der Kante durch $\mathbf{x}_i$ und der x-Achse.

Der Term im Nenner von Gleichung (8) ist

$$\ell^\mathsf{T} \mathtt{K}_{\mathbf{x}_i} \ell = \sigma_i^2 = \sigma_\perp^2 \cos^2(\alpha - \alpha_i) + \sigma_\parallel^2 \sin^2(\alpha - \alpha_i) \tag{9}$$

wenn α den Winkel zwischen ℓ und der x-Achse bezeichnet. Es ist jedoch offensichtlich, daß der durch die Approximation in Gleichung (7) entstandene Fehler ($\sigma^2 = \sigma_\perp^2$) um so kleiner ist, je kleiner die Differenz $||\alpha - \alpha_i||_2$ wird, so daß die Verwendung der Approximation in Gleichung (7) letztendlich doch zu rechtfertigen ist.

Ein vierter Unterschied liegt in der Parametrisierung der Geraden ℓ. In den Gleichungen (7) und (8) wurde von einer Parametrisierung

$$\ell = (a, b, c)^\mathsf{T} \qquad \text{mit} \qquad a^2 + b^2 = 1 \tag{10}$$

ausgegangen. Tatsächlich hat sich für praktische Anwendungen jedoch eine Parametrisierung der Form

$$\ell' = (\alpha, \bar{x}, \bar{y})^\mathsf{T} \tag{11}$$

als geeigneter erwiesen. Vorteilhaft sind hier der volle Rang und die einfache Form der Kovarianzmatrix (die Kovarianzmatrix ist in erster Näherung eine Diagonalmatrix). Zudem ist es trivial, aus (11) auf die Form (10) zu schließen, während der umgekehrte Weg wesentlich schwieriger ist.

Damit wurde auch schon der fünfte und letzte Unterschied zu dem in [37] beschriebenen Algorithmus angesprochen, die Konsequente Berechnung von (Ko)varianz-Information mittels Gleichung (2).

2.4.4 Nichtlineare Verzerrungen

Die letzten Abschnitte beschreiben alle Schritte zur Vorverarbeitung projektiv verzerrter Aufnahmen. Leider hat es sich in ersten Experimenten jedoch gezeigt,

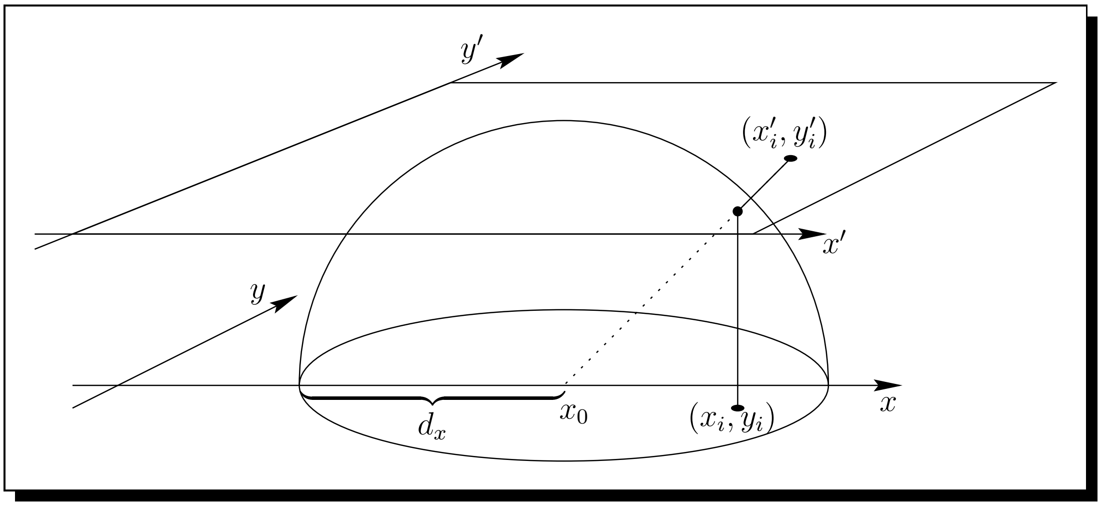


Abbildung 5: *Einfaches Modell zur Modellierung nichtlinearer Verzerrungen.*

daß die im Rahmen des Projektes MOVIS für den Online-Demonstrator (vgl. Abschn. 5) verwandte Kamera teli CS5132P mit dem 7,5 mm Toshiba Objektiv zusätzlich zu den rein projektiven Verzerrungen auch nichtlineare, Kissenförmige Abbildungsfehler aufweist. Derartige Verzerrungen werden für gewöhnlich durch Polynome höherer Ordnung (typischerweise fünfter oder siebter Ordnung) modelliert [42], und im Rahmen von MOVIS wurde ein entsprechender Algorithmus entwickelt [54].

In der Praxis hat es sich jedoch als ausreichend erwiesen, den Fehler durch das folgende Modell zu approximieren

$$x' \leftarrow (x - x_0)\sqrt{\frac{1}{1 - \frac{(x-x_0)^2}{d_x^2} - \frac{(y-y_0)^2}{d_y^2}}} + x_0 \tag{12}$$

$$y' \leftarrow (y - y_0)\sqrt{\frac{1}{1 - \frac{(x-x_0)^2}{d_x^2} - \frac{(y-y_0)^2}{d_y^2}}} + y_0 \tag{13}$$

Hierbei ist $(x_0, y_0)^\mathsf{T}$ der Bildmittelpunkt und $d_{x,y}$ ist ein Maß für die Verzerrung in x- bzw. y-Richtung[4]. Das zugrundeliegende Modell ist hierbei die Abbildung der Gegenstandsebene auf ein Ellipsoid und von dort auf die Bildebene, vergleiche Abbildung 5. d_x und d_y sind in diesem Modell die Halbachsen in x- und y-Richtung.

[4]Für den Demonstrator: $(x_0, y_0) = (255.5\,\text{pxl}, 255.5\,\text{pxl})$, $d_x = 1024.0\,\text{pxl}$, $d_y = 745.0\,\text{pxl}$.

2.5 High-Level Routinen

Dieser Abschnitt beschreibt die High-Level Routinen zur Zebrastreifendetektion, wie sie im Projekt MOVIS verwandt werden. Abschnitt 2.5.1 gibt einen Überblick über die gewählten Algorithmen, die dann in den Abschnitten 2.5.2 bis 2.5.5 näher erläutert werden.

2.5.1 Übersicht

Wie wir in Abschnitt 2.2 gesehen haben, läßt sich ein Zebrastreifen in einer beliebigen projektiven Aufnahme beschreiben als eine Menge von koinzidenten Geraden mit festem Doppelverhältnis, insbesondere ist das Doppelverhältnis von vier aufeinanderfolgenden Geraden 4/3. Aus dem bekannten Doppelverhältnis ergibt sich, daß für jede Menge von mindestens 3 Geraden der Ort des Horizonts für diese Geraden bestimmbar ist. Abschnitt 2.5.2 beschreibt den Prozeß, mit dessen Hilfe mögliche einzelne Streifen identifiziert werden. Hierbei wurde der Schwerpunkt auf robuste Detektion gelegt. Jeweils zwei solcher Streifen (vier Geraden) können dann auf Koinzidenz (vgl. Abschn. 2.5.3) und das korrekte Doppelverhältnis (Abschn. 2.5.4) überprüft werden, bevor durch Rückprojektion nach weiterem Support im Bild gesucht wird (Abschn. 2.5.5).

Tatsächlich lassen sich darüber hinaus noch weitere Aussagen machen, wenn ein gewisses Vorwissen über die Aufnahme existiert. Im Rahmen des Projektes MOVIS ist zum Beispiel die Orientierung der Kamera im Raum zum Zeitpunkt der Aufnahme bekannt, sowie die internen Kamera Parameter. Zusätzlich wird angenommen, daß die ungefähre Größe des Blinden bekannt ist. Das erlaubt die folgenden zusätzlichen Tests (vgl. Abschn. 2.5.5):

- Stimmt die berechnete Lage des Horizonts ungefähr mit der Orientierung der Kamera im Raum überein?
- Liegen die Streifen unterhalb des Horizonts? Andernfalls wäre der Zebrastreifen an der Decke!
- Stimmt die beobachtete absolute Breite der Streifen mit der erwarteten Breite von 50 cm überein?

Schließlich stellt Abschnitt 2.6 die Ergebnisse dieses Algorithmus bei der Anwendung auf Aufnahmen unterschiedlichster Auflösung vor.

2.5.2 Perceptual Grouping

Theoretisch wäre es möglich, Zebrastreifen in einem Bild zu identifizieren, indem für jede Kombination von 4 aus insgesamt N Linien die Tests in den Abschnitten 2.5.3 bis 2.5.5 durchgeführt würden. Es ist jedoch leicht einsichtig, daß der damit verbundene Aufwand der Ordnung $\mathcal{O}\left(\binom{N}{4}\right)$ bereits für kleine N weit außerhalb der im Rahmen von MOVIS gegebenen Möglichkeiten liegt. Aus diesem Grund muß eine gewisse Vorauswahl in Frage kommender Kombinationen von Geraden getroffen werden.

Eine Möglichkeit besteht darin, Fluchtpunkte im Bild zu identifizieren, z. B. mit einem der Algorithmen in [7, 13, 27, 30, 52], weitere Literaturhinweise finden sich in [7, 27]. Collins und Weiss [13] haben jedoch darauf hingewiesen, daß die Berechnung von Fluchtpunkten nicht sehr zuverlässig ist, wenn wie in unserem Fall nur einige wenige Linien zu dem jeweiligen Fluchtpunkt beitragen. Auch die Entwicklung von Algorithmen, die die statistischen Eigenschaften einer Linie berücksichtigen [30] werden dies nur langsam ändern.

Aus diesem Grund wurde im Rahmen von MOVIS ein anderer Weg verfolgt. Wenn man davon ausgeht, daß es sich bei den zu untersuchenden Aufnahmen nicht um beliebige projektive Verzerrungen eines Zebrastreifens handelt, sondern um Aufnahmen, die unter realen Bedingungen (Zentralprojektion, ein gewisser Mindestabstand vom Zebrastreifen) zustande gekommen sind, dann lassen sich Schranken für die maximal möglichen Verzerrungen berechnen. So ergibt sich zum Beispiel, daß sich die beiden Begrenzungslinien eines Streifens nur um maximal etwa $7°$ unterscheiden können. Mit diesem Vorwissen ist es relativ einfach, alle Linien in eine zweidimensionale Tabelle einzutragen (Winkel θ und Ursprungsabstand d) um dann in quasi linearer Zeit zu jeder Linie alle in Frage kommenden gegenüberliegenden Linien zu identifizieren. Zusätzliche Heuristiken bestimmen den maximalen Abstand der Linien zueinander und den maximalen seitlichen Versatz der Linien. Zu beachten ist, daß *nicht* überprüft wird, ob der Zwischenraum zwischen den Linien einheitlich hell gefüllt ist, da viele, insbesondere schlechter erhaltene Zebrastreifen an einer solchen Bedingung scheitern würden.

Um zwei Streifen — 4 Geraden — zu erhalten, werden die möglichen Streifen noch einmal in eine entsprechende Tabelle eingetragen und diese benutzt, um für jeden Streifen nach passenden Gegenstücken zu suchen. Wie bei heuristischen Verfahren üblich ist der Aufwand nur schwer abzuschätzen, liegt aber im schlechtesten Fall bei etwa $\mathcal{O}(N^2)$, typischerweise weit darunter.

2.5.3 Fluchtpunkte

Wie kann man die Wahrscheinlichkeit bestimmen, daß eine Menge von mehr als zwei Geraden koinzident ist, und falls die Koinzidenz der Linien wahrscheinlich ist, wie bestimmt man, wo sich der Schnittpunkt befindet? In Abschnitt 2.4.3 wurde das Problem gelöst, die wahrscheinlichste Gerade durch eine Menge von Punkten zu finden und deren Wahrscheinlichkeit zu bestimmen. Es ist von daher verlockend, den gleichen Algorithmus auch auf dieses Problem anzuwenden, zumal aus der projektiven Geometrie bekannt ist, daß Punkte und Linien dual zueinander sind.

Wie man leicht einsieht ist dies leider nicht möglich. Angenommen der Schnittpunkt der Linien befindet sich im unendlichen. In homogenen Koordinaten wäre dies der Punkt $\mathbf{p} = (x, y, z = 0)^\mathsf{T}$. Um jedoch die Formeln in Abschnitt 2.4.3 anwenden zu können ist eine der Voraussetzungen, daß $z = 1$ ist — das Konzept des Abstands zwischen Punkt und Linie ist für den Fall eines Punktes im unendlichen Bedeutungslos. Tatsächlich ist dies nicht überraschend. Die Formeln in Abschnitt 2.4.3 minimieren den quadratischen Euklidischen Abstand der Punkte von der Geraden — ein Konzept, das Sinn macht, solange wir uns innerhalb des Bildes bewegen, generell unter projektiven Transformationen aber bestenfalls als Fragwürdig bezeichnet werden kann.

Stattdessen benutzen wir im Rahmen des Projektes MOVIS einen Algorithmus, der zuerst von Kanatani vorgestellt wurde [20]. Kanatani benutzte sogenannte *N-Vektoren* — skalierte, normierte, homogene Vektoren — um Punkte und Linien darzustellen. Dies entspricht einer Projektion auf die Einheitskugel. Hierdurch ist es möglich, das Problem, den wahrscheinlichsten Schnittpunkt von M Geraden zu finden, auf eine Version von Gleichung (8) zurückzuführen, für dessen Lösung Kanatani einen typischerweise schnell konvergierenden iterativen Algorithmus vorstellt. Dieser Algorithmus berechnet den Fluchtpunkt $\mathbf{v}$ der am wahrscheinlichsten der Schnittpunkt der M Geraden sein könnte, sowie ein Maß für dessen Wahrscheinlichkeit und günstigerweise auch gleich dessen Kovarianzmatrix.

Abbildung 6 zeigt die Ergebnisse einer Monte-Carlo Simulation der Fluchtpunkt-Berechnung für drei typische Konstellation, nämlich für einen Fluchtpunkt im Bild (Abb. 6 oben links), einen Fluchtpunkt außerhalb des Bildes, aber nahe am Bild (Abb. 6 oben rechts), und einen Fluchtpunkt weit außerhalb des Bildes (Abb. 6 unten). Zugrundegelegt war ein Bild der Größe $512\,\text{pxl} \times 512\,\text{pxl}$. Die Ellipsen in den beiden oberen Bildern markieren die Grenzen der theoretisch berechneten 99 %-Intervalle. Es ist offensichtlich, daß das theoretische Ergebnis sehr gut mit dem simulierten übereinstimmt. Das hyperbolische Vertrauensintervall in Abb. 6 unten entsteht durch den Übergang von N-Vektoren zu kartesischen Koordinaten. Es ist sofort offensichtlich, daß das Problem, den Schnittpunkt quasi

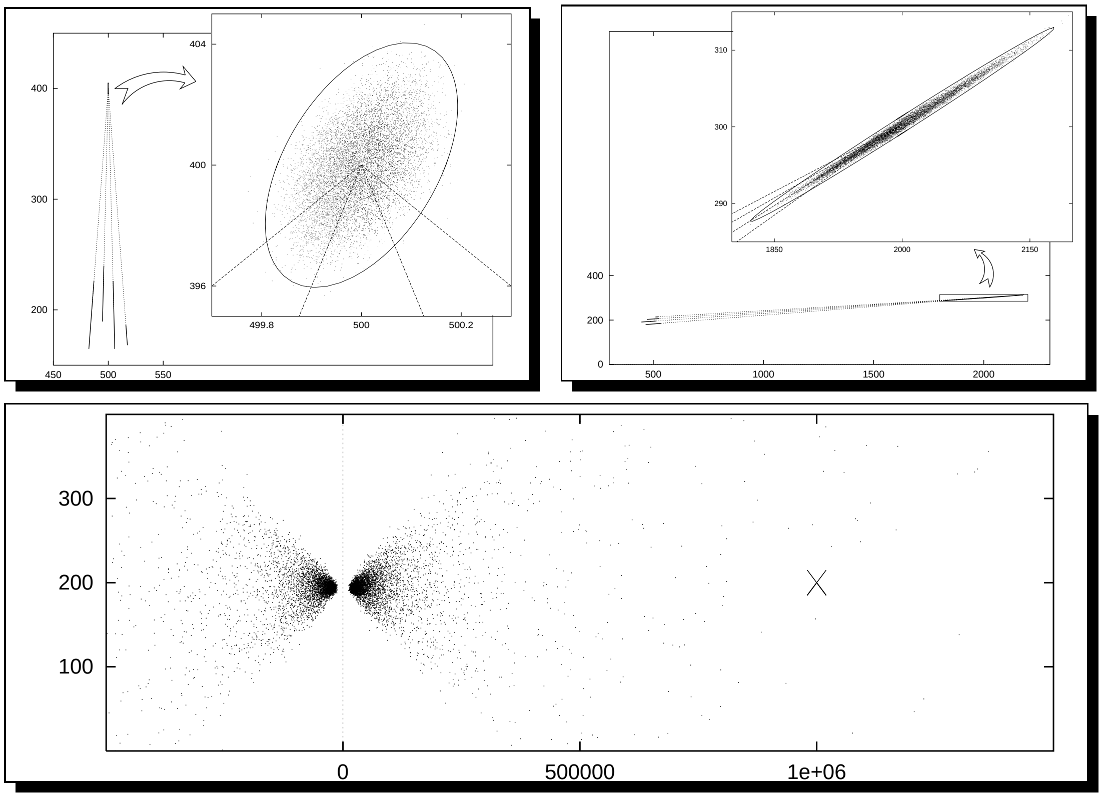


Abbildung 6: *Monte-Carlo Simulation der Fluchtpunkt-Berechnung nach [20].*

paralleler Geraden zu bestimmen ein äußerst schlecht gestelltes Problem ist, und daß kartesische Koordinaten die zugrundeliegende Mathematik nur sehr schlecht modellieren.

Die Entscheidung, ob die Hypothese, daß die untersuchten M Linien koinzident waren, angenommen wird, kann wie üblich mit Hilfe eines χ^2-Tests mit $M-2$ Freiheitsgraden gefällt werden.

2.5.4 Doppelverhältnis

Das Doppelverhältnis von vier koinzidenten Geraden — oder, analog, vier kollinearen Punkten — ist vermutlich die wichtigste projektive Invariante. Leider wird man es aufgrund unterschiedlichster Störungen in realen Aufnahmen nur selten mit wirklich koinzidenten Geraden zu tun haben. Aus diesem Grund wurde in Abschnitt 2.5.3 der wahrscheinlichste Fluchtpunkt von M Geraden bestimmt. Entsprechend würde man, um das Doppelverhältnis von vier Geraden zu bestimmen, erst deren Fluchtpunkt berechnen, um dann einen neuen, koinzidenten Geradensatz zu generieren, dessen Doppelverhältnis man bestimmt (vgl. z. B. [21]). Es zeigt sich, daß fast 90 % des Gesamtaufwandes für die Berechnung des Doppelverhältnisses auf die Berechnung des Fluchtpunktes entfallen, und nur etwa 10 % auf die des Doppelverhältnisses.

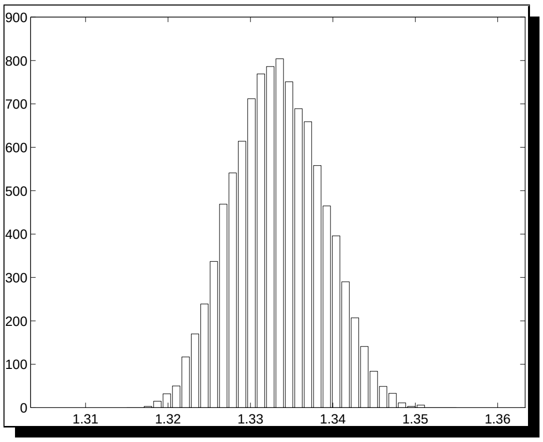
Aus diesem Grunde haben wir uns im Projekt MOVIS die Frage gestellt, ob es nicht möglich ist, das Doppelverhältnis ohne vorherige Bestimmung des Fluchtpunktes zu berechnen. In der üblicherweise zur Berechnung des Doppelverhältnisses herangezogenen Gleichung

$$[ABCD] = \frac{|ACX|}{|ABX|}\frac{|BDX|}{|ADX|}$$

taucht der Fluchtpunkt nicht explizit auf (vgl. Gleichung (1)). Stattdessen wird zur Berechnung eine willkürlich wählbare Gerade $\mathbf{X}$ benutzt, deren einzige Bedingung ist, daß sie nicht durch den Fluchtpunkt der vier anderen Geraden gehen darf[5]. Selbstverständlich ist das Doppelverhältnis überhaupt nur definiert, wenn die vier Geraden koinzident sind, aber reicht es eventuell zu wissen, daß die betrachteten Geraden eigentlich koinzident sein müßten, ohne diese Bedingung zu erzwingen?

Anders als bei koinzidenten Geraden hängt das Ergebnis hier offensichtlich von der Wahl von $\mathbf{X}$ ab. Monte-Carlo Simulationen belegen, daß bei vorsichtiger Wahl von $\mathbf{X}$ die Ergebnisse ohne Berechnung des Fluchtpunktes nur unwesentlich schlechter sind als mit, jedoch etwa um den Faktor 8 schneller (vergleiche

[5]Tatsächlich kann es bei ungeschickter Wahl von $\mathbf{X}$ zu numerischen Problemen kommen.



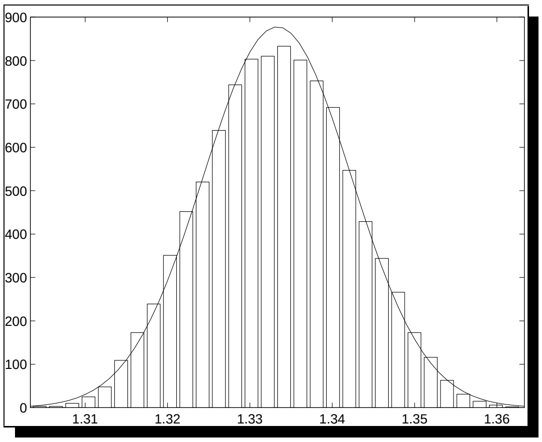


Abbildung 7: *Monte-Carlo Simulation für die Berechnung des Doppelverhältnisses mit (links) und ohne (rechts) Kenntnis des Fluchtpunktes.*

Abbildung 7 und 8). Darüber hinaus wird die Zuverlässigkeit des berechneten Ergebnisses in beiden Fällen korrekt abgeschätzt, so daß das folgende Vorgehen möglich wird: Statt zuerst den Fluchtpunkt und dann gegebenenfalls das Doppelverhältnis zu bestimmen, ist es jetzt möglich, zuerst das Doppelverhältnis zu berechnen. Da das Doppelverhältnis ein wesentlich stärkeres Kriterium ist als die Bedingung des gemeinsamen Fluchtpunktes, müssen anschließend zur Überprüfung der Koinzidenz-Hypothese typischerweise nur noch sehr wenige Fluchtpunkte berechnet werden. Dadurch sinkt die Gesamtrechenzeit erheblich.

Bleibt die Frage, wie der Hilfsvektor $\mathbf{X}$ zu wählen ist. Es ist möglich, $\mathbf{X}$ als eine Gerade aufzufassen, die die anderen vier Geraden schneidet. Es ist von daher offensichtlich, daß $\mathbf{X}$ die vier Geraden möglichst in der Nähe der Punkte, an die sie gefittet wurden, schneiden sollte (vergleiche Abbildung 9). Die optimale Wahl von $\mathbf{X}$ hängt hierbei von den vier Linien $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ und $\mathbf{D}$ sowie deren Varianzen ab. Sie läßt sich im Prinzip berechnen, indem man die Varianz des Doppelverhältnisses bezüglich $\mathbf{X}$ minimiert. Leider zeigt sich, daß eine geschlossene Lösung nicht möglich ist, und numerische Minimierung brächte keinen Geschwindigkeitsvorteil gegenüber der Berechnung des Fluchtpunktes. Unsere Untersuchungen haben jedoch gezeigt, daß es typischerweise ausreicht, $\mathbf{X}$ möglichst gut durch die Schwerpunkte der vier Linien zu legen, und hierfür existiert in der Tat eine geschlossene Lösung.

Es ist ganz interessant, Abbildung 8 links noch einmal unter diesem Gesichtspunkt zu betrachten. Die Abbildung zeigt die Entwicklung der (gemessenen) Varianz des Doppelverhältnisses über der relativen Linienlänge (Anzahl Kantenpunkte) im Doppelt-Logarithmischen Maßstab. Wir sehen, daß mit abnehmender Kantenlänge die Kovarianz des mit dem schnellen Algorithmus berechneten Doppelverhältnisses quadratisch ansteigt, während die Kovarianz des unter Benut-

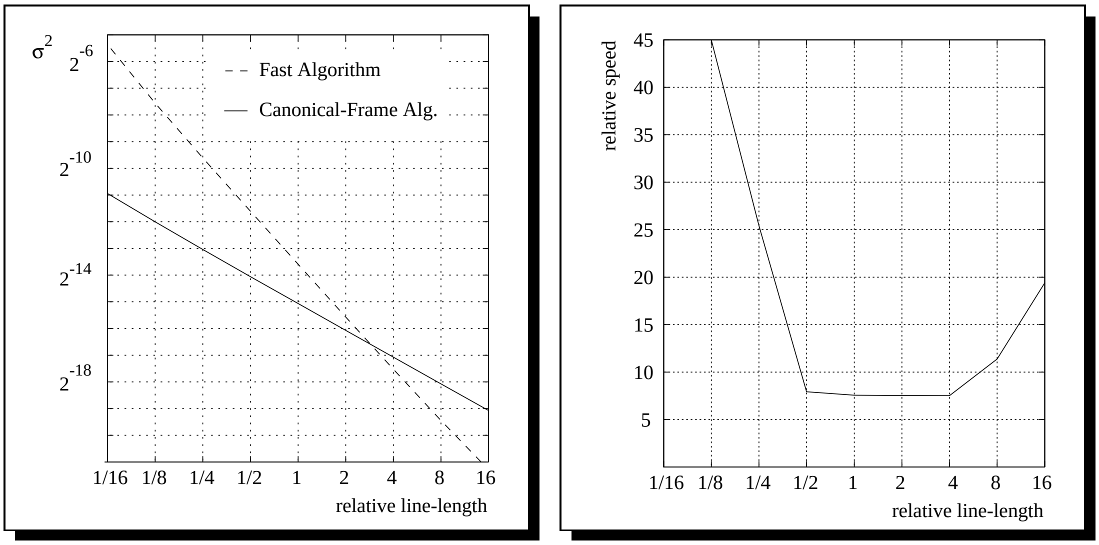


Abbildung 8: *Vergleich der Geschwindigkeit und Genauigkeit zweier Algorithmen zur Berechnung des Doppelverhältnisses.*

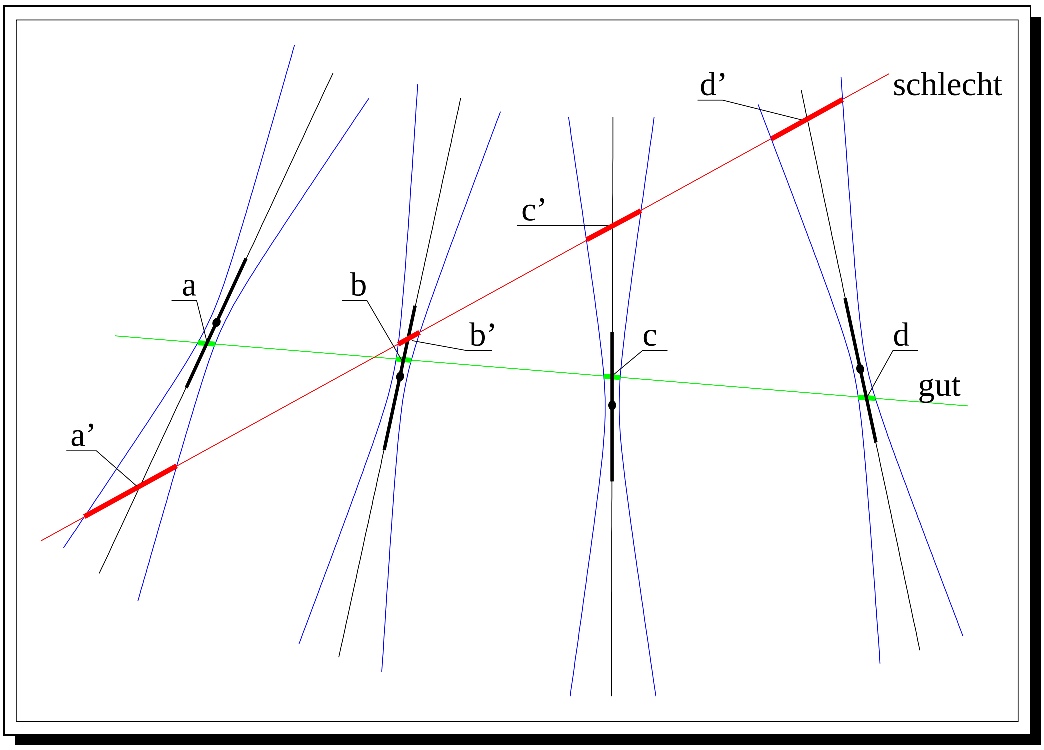


Abbildung 9: *Gute und schlechte Wahl der Hilfsgeraden* **X**.

zung des Fluchtpunktes berechneten Doppelverhältnisses nur linear zunimmt. Der Grund dafür liegt in der unterschiedlichen Abhängigkeit des Schwerpunktes und des Winkels von der Anzahl der Kantenpunkte. Während die Varianz des Schwerpunktes invers proportional zur Anzahl der Kantenpunkte n ist, $\sigma_{\bar{x}}^2, \sigma_{\bar{y}}^2 \propto \frac{1}{n}$, ist die Varianz des Winkels invers Proportional zum Quadrat der Kantenpunkte, $\sigma_\alpha^2 \propto \frac{1}{n^2}$. Da durch die Einbeziehung des Fluchtpunktes in die Berechnung des Doppelverhältnisses im wesentlichen Fehler im Winkel behoben werden, während unsere Wahl von $\mathbf{X}$ nur die Schwerpunkte der Liniensegmente berücksichtigt, ist umgekehrt das unterschiedliche Fehlerverhalten sofort verständlich.

Ob der berechnete Wert für das Doppelverhältnis CR dem erwarteten Wert $\overline{\mathrm{CR}}$ entspricht, kann wie üblich mit dem χ^2-Anpassungstest überprüft werden,

$$\frac{\left(\mathrm{CR} - \overline{\mathrm{CR}}\right)^2}{\sigma_{\mathrm{CR}}^2 + \sigma_{\overline{\mathrm{CR}}}^2} \leq \chi_{p,1}^2$$

Hierbei ist σ_{CR}^2 die berechnete Varianz des Doppelverhältnisses und $\sigma_{\overline{\mathrm{CR}}}^2$ die Varianz des Modells, eventuell $\sigma_{\overline{\mathrm{CR}}}^2 = 0$.

2.5.5 Rückprojektion und weitere Tests

Sobald zwei Streifen bzw. vier Geraden[6] eines Zebrastreifens identifiziert wurden ist es möglich, die Lage weiterer Streifen im Bild zu berechnen. Damit kann direkt im Bild nach Support für eine Zebrastreifenhypothese gesucht werden. Zusätzlich kann die Bildposition der Geraden im Unendlichen bestimmt und diese für weitere Tests herangezogen werden.

In der Literatur wird die Bestimmung der Bildposition weiterer Linien häufig so durchgeführt, daß drei schon bekannte Linien als sogenannte projektive Basis gewählt werden und davon ausgehend die Lage weitere Linien mit bekanntem Doppelverhältnis berechnet wird[19, 41]. In der Praxis zeigt sich jedoch, daß dieses Vorgehen extrem Fehleranfällig ist. Darüber hinaus ist es auch methodisch unerfreulich, daß hierdurch niemals die gesamte zur Verfügung stehende Information (typischerweise sind die Positionen von mehr als drei Geraden bekannt) genutzt wird.

Um optimale Ergebnisse zu erreichen, müßte man statt dessen die Transformation T^T berechnen, die M Geraden aus einer Normalform (z. B. horizontale Geraden mit bekannter Position) so in das Bild projiziert, daß der Abstand zwischen den berechneten und gemessenen Geraden minimal wird. Ist diese Transformation einmal berechnet, dann ist es einfach, die Position anderer Geraden vorherzusagen, in unserem Beispiel $\mathsf{T}^\mathsf{T}\ell_i''$. Dabei ist $\ell_i'' = (0, 1, c_i'')^\mathsf{T}$ eine der Geraden aus der

[6] Tatsächlich reichen drei Geraden.

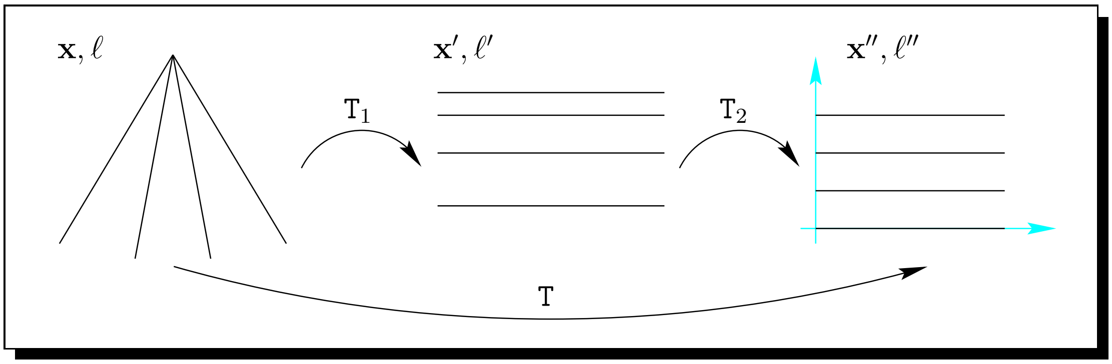


Abbildung 10: *Zerlegung der Transformation* $\mathtt{T}$ *in zwei Transformationen* $\mathtt{T}_1$ *und* $\mathtt{T}_2$*.*

Normalform, vgl. Abbildung 10. Entsprechend berechnet sich die Bildposition der Linie im Unendlichen als $\mathtt{T}^{\mathsf{T}}(0,0,1)^{\mathsf{T}}$. Die Matrix $\mathtt{T} \in I\!R^{3\times3}$ hat 5 Freiheitsgrade.

Da es für $\mathtt{T}$ keine geschlossene Lösung gibt, wird im Rahmen des Projektes MOVIS ein geringfügig anderer Weg gegangen. Statt $\mathtt{T}^{\mathsf{T}}$ zu berechnen, wird die Transformation $\mathtt{T}^{-\mathsf{T}}$ berechnet, die den Abstand der Linien in der Normalform minimiert (statt im Bild). Für die Berechnung dieser Matrix gibt es unter der Annahme kleiner Fehler eine sehr effektive Approximation, und zwar ist es möglich, $\mathtt{T}$ in zwei Teilmatrizen $\mathtt{T}_1$ und $\mathtt{T}_2$ zu zerlegen, die sich separat berechnen lassen. Die Matrix $\mathtt{T}_1^{-\mathsf{T}}$ projiziert alle Linien in eine Normalform, in dem die Geraden (so gut wie möglich) horizontal verlaufen, während die Transformation $\mathtt{T}_2^{-\mathsf{T}}$ die Geraden in ihre endgültige Position bringt (vgl. Abbildung 10). Damit hängt Matrix $\mathtt{T}_1^{-\mathsf{T}}$ nur vom Fluchtpunkt $\mathbf{v} = (x, y, z)^{\mathsf{T}}$ ab, der Bereits in Abschnitt 2.5.3 bestimmt wurde; es ist o. B. d. A.

$$\mathtt{T}_1 = \begin{pmatrix} x & y & 0 \\ -y & x & 0 \\ \frac{-xz}{\sqrt{x^2+y^2}} & \frac{-yz}{\sqrt{x^2+y^2}} & \sqrt{x^2+y^2} \end{pmatrix} \tag{14}$$

(zwei Freiheitsgrade). Die Matrix $\mathtt{T}_2^{-\mathsf{T}}$ hat die Form

$$\mathtt{T}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & t_y \\ 0 & p_y & s \end{pmatrix} \tag{15}$$

Eine geschlossene Lösung für $\mathtt{T}_2^{-\mathsf{T}}$ existiert.

Die Zerlegung der Matrix $\mathtt{T}^{-\mathsf{T}} = \mathtt{T}_2^{-\mathsf{T}}\mathtt{T}_1^{-\mathsf{T}}$ ist genaugenommen nur möglich, wenn entweder alle Geraden nach der Transformation mit $\mathtt{T}_1$ genau horizontal sind, oder wenn $p_y = 0$ in $\mathtt{T}_2$, da ein $p_y \neq 0$ die Winkel aller nicht-horizontalen Geraden verändern würde. Diese Änderung ist jedoch klein für nahezu horizontale Geraden und kann von daher unter der Annahme kleiner Fehler vernachlässigt werden.

Nachdem weiterer Support im Bild identifiziert wurde, können mehrere möglicherweise überlappende Hypothesen verschmolzen und eine Reihe weiterer Tests durchgeführt werden. Hierzu kann mit $\mathtt{T}^{-\mathsf{T}}(0,0,1)^{\mathsf{T}}$ die Bildposition der Linie im Unendlichen bestimmt werden. Der Zebrastreifen muß ganz unterhalb dieser Linie liegen. Da im Rahmen des Projektes MOVIS die ungefähre Orientierung der Kamera gegenüber der Senkrechten sowie der Öffnungswinkel der Kamera bekannt sind, kann zusätzlich die berechnete Bildlage der Linie im unendlichen (Horizont) mit der vermuteten Lage verglichen werden.

Schließlich kann die Größe des hypothetischen Zebrastreifens bestimmt werden unter der Annahme, daß der Abstand Grundfläche–Kamera (im wesentlichen die Größe des Blinden) gegeben ist. Dazu wird der Zebrastreifen auf die Grundfläche projiziert und der Abstand z. B. des Schwerpunktes einer Linie zur Nachbargeraden bestimmt (sinnvollerweise wird über alle so ermittelten Abstände gemittelt). Die Projektionsmatrix ist relativ komplex,

$$\mathtt{P} = \begin{pmatrix} h\cos(\alpha)s_x & h\sin(\alpha)s_y & -h\cos(\alpha)x_c - h\sin(\alpha)y_c \\ -fhp_y\sin(\alpha)s_x & fhp_y\cos(\alpha)s_y & fhp_y(\sin(\alpha)x_c - \cos(\alpha)y_c + t_y) + fh \\ -(1+t_yp_y)\sin(\alpha)s_x & (1+t_yp_y)\cos(\alpha)s_y & (1+t_yp_y)(\sin(\alpha)x_c - \cos(\alpha)y_c + t_y) + t_y \end{pmatrix} \tag{16}$$

dies läßt sich jedoch in mehrere Einzeloperationen zerlegen:

$$P = P_2T_2P_1T_2R_1T_1S_1 = \begin{pmatrix} h & 0 & 0 \\ 0 & 0 & fh \\ 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & p_y & 1 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{17}$$

In dieser Formel sind h die Höhe der Kamera über der Grundfläche (Körpergröße), f die Brennweite der Kamera, α der Winkel zwischen Horizont und x-Achse, $(x_c, y_c)^{\mathsf{T}}$ der Bildmittelpunkt, s_x und s_y voneinander abhängige Skalierungsfaktoren (typischerweise wird man einen der beiden Werte auf 1 setzen und den anderen auf das Seitenverhältnis der Kamera). Weiterhin sind

$$t_y = f * \tan\left(\frac{1}{2}\arctan\left(\frac{d}{f}\right)\right)$$

und

$$p_y = -\frac{d}{f^2 - dt_y}$$

Dabei ist d der Bildabstand des Horizonts vom Bildmittelpunkt.

Die auf diese Art ermittelte durchschnittliche Breite der Streifen kann dann mit der gewünschten Breite von 500 mm verglichen werden. Hier reichen relativ großzügige Grenzwerte, typischerweise werden Breiten von 300 mm bis 1000 mm akzeptiert[7].

2.6 Beispiele und Ergebnisse

Abbildung 11 zeigt einige Beispiele erkannter Zebrastreifen. Insgesamt wurde der Zebrastreifen-Erkenner in MOVIS an weit über 1000 Bildern sowohl mit als auch ohne Zebrastreifen getestet, die von verschiedenen Kameras generiert wurden und deren Auflösung von 439 pxl × 299 pxl bis 1024 pxl × 682 pxl reichten[8]. Die Erkennungsrate betrug unabhängig von der verwendeten Kamera und Auflösung relativ konstant um 70 % (selbstverständlich müßen die Parameter des Algorithmus zu Anfang an die verwandte Kamera angepaßt werden, bleiben danach jedoch konstant). Ein False-Positive wurde nicht einmal beobachtet, ist aber Aufgrund der gewählten Vorgehensweise nicht auszuschließen. Weiter Beispiele sind in Abschnitt 5.5 zu sehen.

[7] Es ist schwierig, h genau abzuschätzen, da der Blinde möglicherweise erhöht stehen könnte. Hierdurch, und durch Ungenauigkeiten in den anderen Variablen, sind relativ starke Schwankungen in der ermittelten Breite der Streifen möglich, die sich nur noch schwer modellieren lassen.

[8] Die im Rahmen von MOVIS standardmäßig verwandte Auflösung ist 512 pxl × 286 pxl.

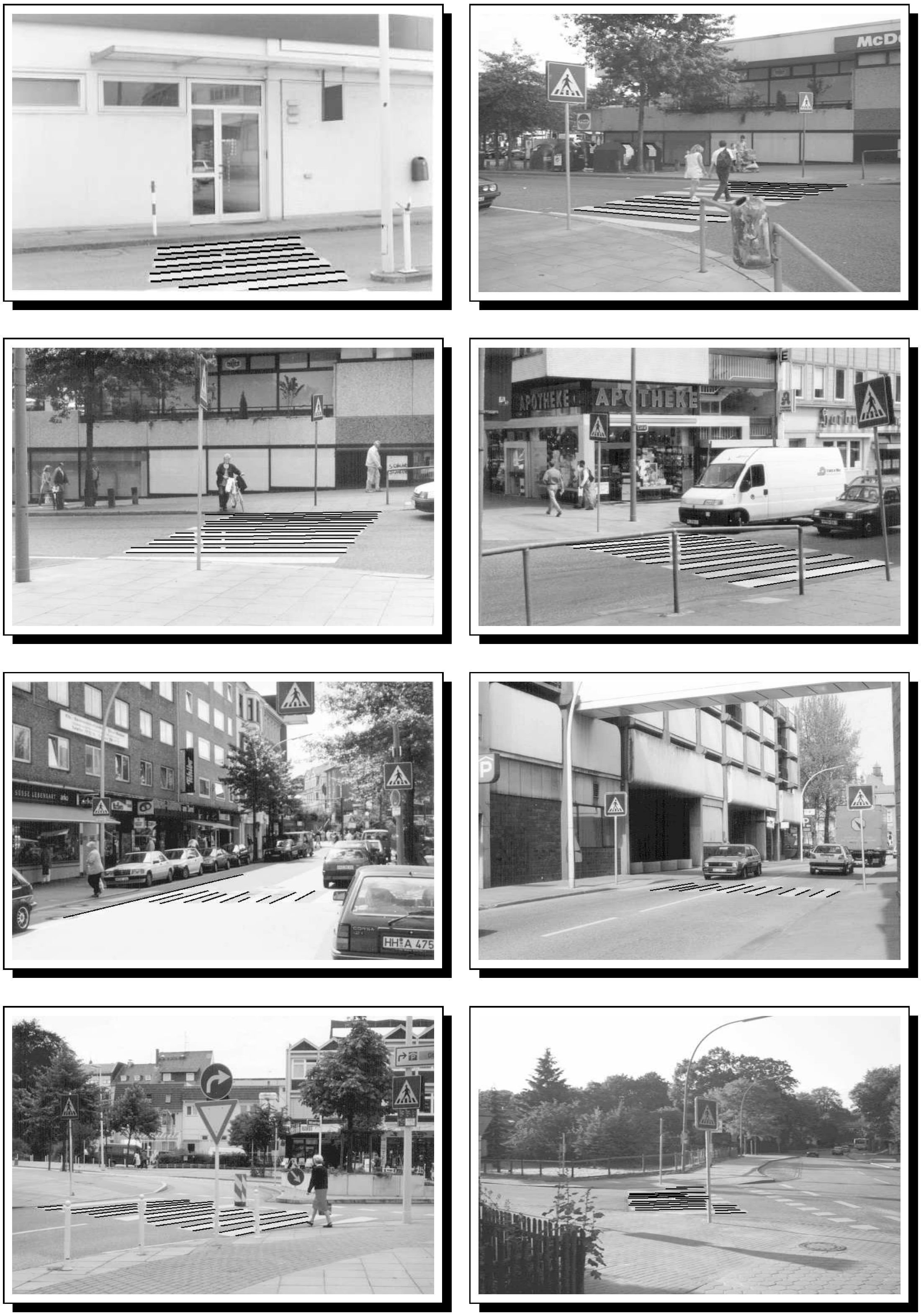


Abbildung 11: *Beispiele für die Zebrastreifenerkennung in* MOVIS

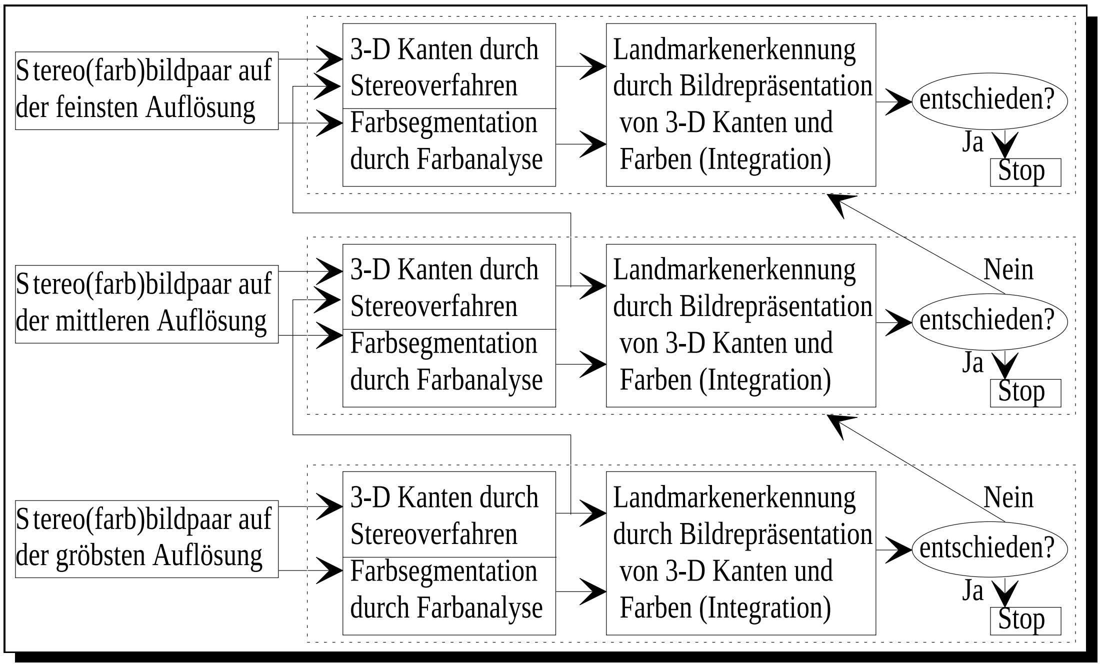


Abbildung 12: *Übersicht über die Fokussierung zur Landmarkenerkennung mit Hilfe von Tiefen- und Farbinformation.*

3 Erkennung von 3-D-Landmarken

3.1 Einleitung

Wie in Abschnitt 1 erwähnt, sind viele der für Blinde und Sehbehinderte wichtigen Landmarken künstliche, dreidimensionale Gegenstände, die eine regelmäßige Form aufweisen. Diese 3-D-Formen können sehr oft durch gerade Kanten eindeutig charakterisiert werden. Zur Beschreibung und Erkennung von 3-D-Landmarken spielt deshalb kantenbasierte Tiefeninformation eine wichtige Rolle; häufig reicht sie jedoch allein nicht aus, um Landmarken mit der nötigen Sicherheit zu identifizieren. Ergänzend können deshalb andere Merkmale, wie z. B. Farbe, Textur und Schrift verwendet werden.

Im Rahmen von MOVIS wurde eine pyramidale Bildrepräsentation mit 3-D-Kanten und Farbinformationen zur Beschreibung und Erkennung von 3-D-Landmarken entwickelt [25]. Wie beim menschlichen Sehen wird hierbei die Szene zunächst global analysiert, um danach unter Verwendung einer feineren Auflösung eine Landmarke zu bestätigen. Sobald in einer (groben) Auflösungsebene entweder eine Landmarke deutlich erkannt wird oder kein Bereich als mögliche Landmarke in Frage kommt, wird die Bildrepräsentation auf dieser Auflösung als ausreichend angesehen und die Analyse der Szene beendet; ansonsten wird die Bild-

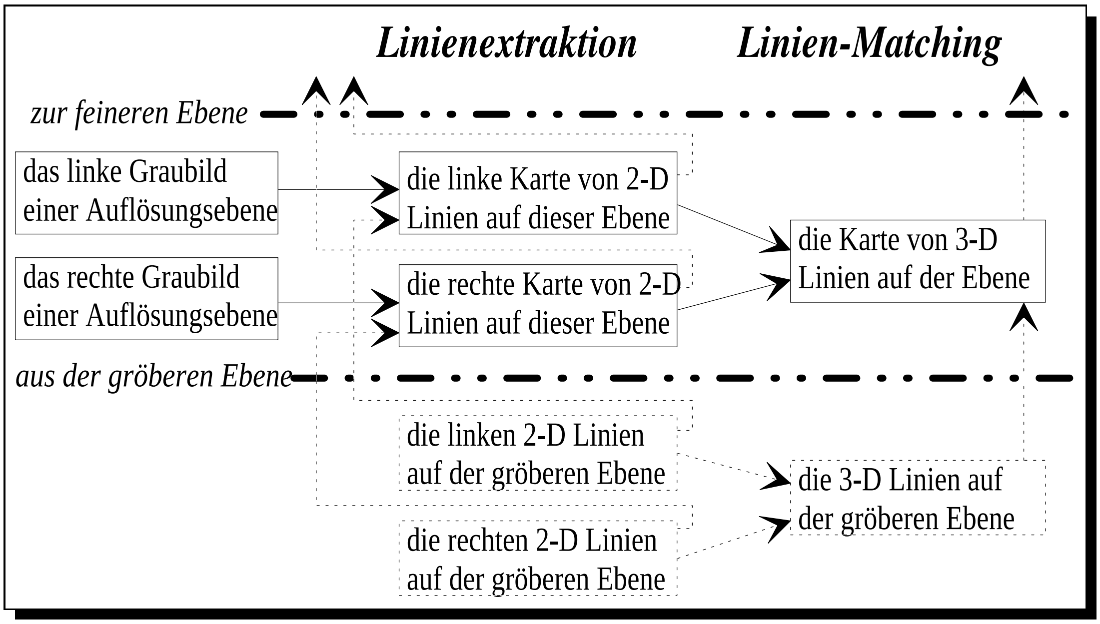


Abbildung 13: *Überblick über den pyramidalen, Linien-basierten Stereo-Algorithmus (nur eine Ebene wird dargestellt).*

repräsentation in die nächstfeinere Auflösungsebene übertragen und dort weiter verbessert (vgl. Abb. 12).

3.2 Extraktion von 3-D-Kanten

Das Stereoverfahren ist die wichtigste passive Methode zur 3-D-Tiefendatenschätzung. In [24] geben wir einen Überblick über verschiedene Ansätze und diskutieren die entsprechenden Vor- und Nachteile einzelner Verfahren. In [23] diskutieren wir einige der Probleme bei der Konfiguration von Stereokameras sowie der Einflüsse linearer und nichtlinearer Verzerrungen auf die Schätzfehler; dort stellen wir ebenfalls einen einfachen und effektiven Ansatz zur Kalibrierung der Kameras in der Praxis vor. Für den Fall, daß eine sehr genaue Tiefeninformation benötigt wird, wurde bei uns im Rahmen des Projektes MOVIS ein robusteres und genaueres Verfahren zur Kalibrierung mit 3-D-Gitterpunkten entwickelt [54].

Wie in der Einleitung erwähnt, wird für die Beschreibung und Wiedererkennung von Landmarken häufig nicht die vollständige Tiefeninformationen benötigt; stattdessen kommt man in vielen Fällen mit Information über 3-D-Kanten aus. Deshalb wurde für MOVIS ein neues, linienbasiertes Stereoverfahren entwickelt. In [26] werden die Vorzüge unseres Verfahrens gegenüber gängigen linienbasierten Methoden diskutiert (z. B. [22, 29, 31]). In unserem neuen Verfahren werden nur die in den beiden Stereobildern einer Szene extrahierten geraden Linien als Merkmale miteinander zur Deckung gebracht, und dadurch 3-D-Kanten basierend auf der Triangulationsgeometrie ermittelt. Das Stereoverfahren besteht demnach aus der Linienextraktion und gleichzeitigem Stereo-Matching; dieses wurde mit Hilfe einer Auflösungspyramide zuverlässig und rechenaufwandsgünstig implementiert. Abbildung 13 stellt eine Auflösungsebene der Algorithmenstruktur dar.

In diesem Lösungsansatz spielen das Mehrgitterverfahren und die Fuzzy-Evaluation über extrahierte 2-D- bzw. 3-D-Linien eine wichtige Rolle; einen Überblick über Fuzzy-Ansätzen in der Bildverarbeitung gibt [39]. Neben den üblichen Eigenschaften wie z. B. den geometrischen und Grauwertparametern, berechnen wir zusätzlich einen Wert für die Zuverlässigkeit der extrahierten Linien und verwenden diesen in allen Stufen: Linking, Merging, Fokussierung und Matching. Hierdurch konnten bessere Ergebnisse als z. B. in [4] und [49] erzielt werden. Weiterhin hilft diese quantitative Auswertung bei der Erkennung und Verifikation von Landmarken.

3.2.1 Extraktion von 2-D-Linien

Eine Reihe von Kantendetektoren wurden in den letzten Jahren vorgestellt, darunter die von Robert, Prewitt und Kirsch; Marrs und Hildreths zero-crossing of Laplacian Gaussian; Haralicks zero-crossing of second directional derivatives und Cannys optimaler Kanten-Detektor. Um die Grauwerte gerader Linien für das Linien-Matching zu nutzen, sind die Gradienten-basierten Kanten-Detektoren besser geeignet. Im Rahmen des Projektes MOVIS verwenden wir einen Sobel-Operator. Aus dem Gradientenbild können anschließend Linien extrahiert werden.

Die Linien-Extraktion ist ein High-Level Prozeß, der die primitiven Kanten-Pixel in einer Linien-Struktur gruppiert. Hough-Verfahren können die geraden Linien unmittelbar extrahieren, liefern jedoch keine Informationen über die Endpunkte eines Linie-Stücks und seine Grauwert-Attribute. Sie sind deshalb für Linien-basierte Stereoverfahren nicht sehr geeignet. Eine andere Methode zur Linienextraktion wurde von Burns [10] vorgestellt. Diese Methode segmentiert zuerst benachbarte Pixel mit ähnlichen Gradientenrichtungen in die Regionen der Linien-Unterstützung und extrahiert dann die geraden Linien unmittelbar durch Ebenen-Annäherung. Diese Methode hat sich aber für uns ebenfalls als ungeeignet erwiesen, da es zum einen schwer ist, zuverlässige vollständige Regionen der Linien-Unterstützung in Aufnahmen von Realszene zu ermitteln. Der andere Nachteil ist die hohe algorithmische Komplexität.

Eine primitive Methode, die häufig in linienbasierten Stereoverfahren zur Anwendung kommt, verbindet nach Kanten-Detektion und -Verdünnung die Kanten-Pixel und nähert sie durch eine Linie an. In [49] wurde eine Strategie des Scan-

Label-Link-Merging vorgestellt, um die Zuverlässigkeit zu verbessern und die Fragmentierung und Komplexität des Algorithmus zu reduzieren. In [4] wird eine hierarchische Kanten-Verfeinerung (edge focusing) vorgestellt, um Rauschen und unnötige Details zu unterdrücken. Der Nachteil der Methode ist das starre Tracking vom Gröberen zum Feineren. Hierdurch bleiben unzuverlässige grobe Merkmale stets erhalten, während neue zuverlässige Merkmale, die auf der gröberen Auflösungsebene nicht gefunden wurden, nicht mehr berücksichtigt werden können.

Um die oben erwähnten Methoden zu verbessern, stellen wir eine auf einer Fuzzy-Strategie basierende Multilevel-Methode zur Linien-Extraktion vor, die mehrere Vorteile gegenüber [4] hat. Anders als dort verwendet der Algorithmus die groben Merkmale zur Leitung des gesamten Prozesses des Scan-Label-Link-Merging, berücksichtigt aber die mit einem Fuzzy-Maßstab bestimmte Zuverlässigkeit der Merkmale. Hierdurch können neue, zuverlässige Merkmale, die in gröberen Stufen nicht existierten, in feineren Stufen des Linkens und Mergens mit Fuzzy-Regel erzeugt werden. Die Implementierung des Algorithmus in mehreren Stufen wird in Abb. 13 dargestellt. Die Linienextraktion jeder Stufe hat die folgenden Schritten:

- Berechnung des Gradientenbildes mit Hilfe des Sobel Operators.
- Auswahl von Kanten-Pixeln entlang der Gradientenrichtungen mit dem Verfahren der Non-Maximum Suppression.
- Gruppierung der Kantenpixel in Segmente, wobei die ursprüngliche Methode des Scan-Labeling durch eine robuste Suche und einen Schwellwert zur Verringerung der Fragmentierung in allen Richtungen verbessert wird.
- Linien-Segmenten werden durch Fuzzy-Regeln für das Linking und die Leitung mit den zuverlässigen gröberen geraden Linien ermittelt. Die Linien-Segmenten werden mit einer Fuzzy Gütefunktion basierend auf ihrem Grauwert, Betrag des Gradienten, geometrischen Abweichungen, Fragmentierung und der Zuverlässigkeit ihrer Vorgänger in den gröberen Auflösungsebenen bewertet.
- Vollständige gerade Linien werden mit Hilfe von Fuzzy-Regeln zum Merging und der Leitung durch die zuverlässigen geraden Linien in der gröberen Auflösungsebene ermittelt. Zusätzlich wird die Verschmelzung von Liniensegmenten durch eine Fuzzy Gütefunktion bestimmt, in den wie zuvor der Grauwert, Betrag des Gradienten, geometrische Abweichungen und die Fragmentierung eingehen.

Dank der Leitung durch die gröberen Auflösungsebenen und des daraus und aus dem Fuzzy-Ansatz resultierendem robustem Linking und Merging können möglichst vollständige gerade Linien mit wenigen Fragmenten extrahiert werden. Dies erleichtert die nachfolgenden Schritten des Linien Matching und der Objekterkennung. Wichtig für unseren Ansatz ist, daß unzuverlässige Informationen der gröberen Auflösungsebenen nicht zwingend erhalten bleiben. Die hohe positionelle Genauigkeit der Detektion von geraden Linien, die zur guten 3-D-Tiefenschätzung führt, wird in der entsprechenden Stufe erreicht. Diese geraden Linien können, wenn nötig, in feineren Stufen verwendet werden.

3.2.2 Linien-Matching

Um die 3-D-Kanten von Objekten aus Stereobildpaaren zu schätzen ist es nötig, in einem Matching-Prozeß die 2-D-Kanten aus beiden Stereobildern einander korrekt zuzuordnen. Um Mehrdeutigkeiten aufzulösen, werden in fast allen Stereoverfahren Nebenbedingungen herangezogen, die sich aus Modellen der Grauwerte und der Geometrie ergeben, insbesondere die lokale Ähnlichkeit der Merkmale, epipolare Beschränkungen aus der Stereo Triangulation und weitere Beschränkungen der Kontinuität. Aufgrund der relativ hohen Komplexität verwenden die meisten vorhandenen Linien-basierten Stereoverfahren die Strategie der Relaxation oder der dynamischen Programmierung zur Ermittlung von eindeutigen eins-zu-eins Korrespondenzen. Hierzu ist häufig die Initialisierung des Matching-Prozesses durch eine Menge bekannter Zuordnungen zwingend erforderlich [22, 31]. In [29] wurde eine Methode des Linien-Matchings in einem Durchlauf vorgestellt, wobei eine Matching-Funktion die Güte des Matching auf Grundlage vieler Beschränkungen berechnet. Tatsächlich ist es schwer, alle diese Beschränkungen geeignet zu kombinieren, zumal die Zuverlässigkeit des Matchings durch die Gewichtung der Beschränkungen in der Matching-Funktion beeinflußt wird. Für alle vorhandenen Linien-basierten Methoden des Stereo-Matching ist deshalb die sorgfältige Auswahl der expliziten Matching-Regeln wesentlich, um falsche Zuordnungen zu vermeiden.

Um die Komplexität zu reduzieren und zuverlässige Zuordnungen von Linien zu ermitteln, stellen wir einen neuen pyramidalen Algorithmus mit Fuzzy-Verfahren vor. Der Prozeß des Linien-Matching wird hierbei in nur einem Schritt erledigt. Verschiedene Matching-Beschränkungen werden mit einer Fuzzy Gütefunktion behandelt, ihre Zusammenhänge innerhalb des Matching-Prozesses werden durch einige Fuzzy-Regeln leicht vereinfacht modelliert. Der abzusuchende Raum für Zuordnungen wird durch die Ergebnisse gröberer Auflösungsebenen erheblich reduziert, so daß mehrfache Zuordnungen leicht zu vermeiden sind. Die Einflüsse unzuverlässiger oder falscher Ergebnisse in den gröberen Stufen können durch Fuzzy-Gütefunktionen und -Tracking effektiv untergedrückt werden.

Die pyramidale Implementierung des Linien-Matching wird in Abb. 13 dargestellt.

Der Algorithmus läßt sich auf jeder Auflösungsebene wie folgt beschreiben:

Linien-Matching: für die verbleibende stärkste Linie L_i^l des linken Bildes (bis keine geeigneten Linien mehr existieren)

- Probiere alle möglichen Zuordnungen mit den Linien L_k^r in einem lokalen Fenster des rechten Bildes: $m_M(\text{match}(L_i^l, L_k^r))$ (siehe Abschnitt 3.2.3)
- Finde die zuverlässigste Zuordnung: $m_M(\text{match}(L_i^l, L_j^r)) = \max\limits_k(m_M(\text{match}(L_i^l, L_k^r)))$
- Entferne die Linie L_i^l aus dem linken Bild. Wenn $m_M(\text{match}(L_i^l, L_j^r))$ einen Fuzzy-Schwellwert übersteigt, dann berechne die entsprechende 3-D-Kante und entferne auch die Linie L_j^r aus dem rechten Bild.

Verbesserung der 3-D-Kanten: verbinde unterbrochene 3-D-Kanten und unvollständige Strukturen.

- Wenn sich ein Paar von benachbarten 3-D-Kanten entlang einer 3-D-Linie befindet, dann verbinde die beiden 3-D-Kanten und berechne eine neue Fuzzy-Gütefunktion für diese Linie aufgrund der Zuverlässigkeiten ihrer Vorgänger und deren Nachbarschaft.
- Wenn die beide Endpunkte einer nicht zugeordneten Linie im 2-D-Bildraum in der Nähe von einigen 3-D-Kanten liegen, dann ermittle die Tiefe dieser Linie und deren Fuzzy-Gütefunktion aufgrund der relevanten 3-D-Kanten und ihrer Abstände zur 2-D-Kante.

Die Unterbrechungen von 3-D-Linien werden in diesen zwei Schritten zufriedenstellend beseitigt.

3.2.3 Fuzzy Matching

Um die Komplexität der Suche zu reduzieren, verwendet unser Algorithmus, wie allgemein üblich, die folgenden Nebenbedingungen: epipolare Einschränkung, Annahme der Kontinuität, Annahme der grauwertigen und geometrischen Ähnlichkeit der Linien. Es hat sich gezeigt, daß diese Einschränkungen besonders effizient behandelt werden können, wenn sie in den folgenden Fuzzy-Regeln kombiniert werden:

Wenn: Linie L_i^l auf dem linken Bild und Linie L_k^r auf dem rechten Bild

1. überlappen sich gut in vertikaler Richtung: $m_1(l_i^l, l_k^r)$ (epipolare Einschränkung)

2. haben den gleichen Winkel: $m_2(\theta_i^l, \theta_k^r)$ (Einschränkung der geometrischen Ähnlichkeit)
3. haben die gleiche Gradientengröße mit identischem Vorzeichen: $m_3(g_i^l, g_k^r)$ (Einschränkung der Grauwert-Ähnlichkeit)
4. haben eine ähnliche Disparität wie benachbarte, zuverlässig zugeordnete, starke 3-D-Kanten: $m_4(d)$ (Einschränkung der Kontinuität)
5. haben eine ähnliche Disparität wie die benachbarten in der gröberen Stufe ermittelten 3-D-Kanten: $m_5(d)$ (Multilevel-Transfer)

Dann: L_i^l und L_k^r bilden ein Paar,

wobei l die vertikale Projektion von L ist, θ der Winkel von L, g die Gradientengröße und d die Disparität zwischen zwei Linien L_i^l und L_k^r. Die oben beschriebene Fuzzy-Regel kann mathematisch wie folgt ausgedrückt werden:

Definition der Fuzzy-Gütefunktion zum Linien-Matching

$$m_M(match(L_i^l, L_k^r)) = \min(m_1, m_2, m_3, m_4)\ m_5$$

Definiert sind die einzelnen Regeln wie folgt:

Mit der verkürzten Funktion

$$S_w(x) = \begin{cases} 1 & : \quad |x| \leq w \\ 2 - (|x|/w) & : \quad w < |x| < 2w \\ 0 & : \quad 2w \leq |x| \end{cases}$$

definieren wir

$$\begin{aligned}
m_1(l_i^l, l_k^r) &= S_{\Delta l}(\log(l_i^l/l_k^r)) \frac{\text{overlap}(l_i^l/l_k^r)}{\min(l_i^l, l_k^r)} && (18) \\
m_2(\theta_i^l, \theta_k^r) &= S_{\Delta\theta}(\theta_i^l - \theta_k^r) && (19) \\
m_3(g_i^l, g_k^r) &= S_{\Delta g}(\log(g_i^l/g_k^r)) && (20) \\
m_4(d) &= \frac{(k_0 + \sum_j k_j m_M(d_j) S_{\Delta d}(d - d_j))}{(k_0 + \sum_j k_j m_M(d_j))} && (21) \\
m_5(d) &= \frac{(k_0 + \sum_i k_i m_M(d_i) S_{\Delta d}(d - d_i))}{(k_0 + \sum_i k_i m_M(d_i))} && (22)
\end{aligned}$$

wobei der Koeffizient i in Gl. (22) über alle in der gröberen Stufe ermittelten, benachbarten 3-D-Kanten läuft und j in Gl. (21) über alle benachbarten und zuverlässigen 3-D-Kanten der gleichen Auflösungsebene iteriert (eventuell die leere Menge). d_i und d_j sind die Disparitäten der jeweiligen 3-D-Kanten, $m_M(d_i)$ und

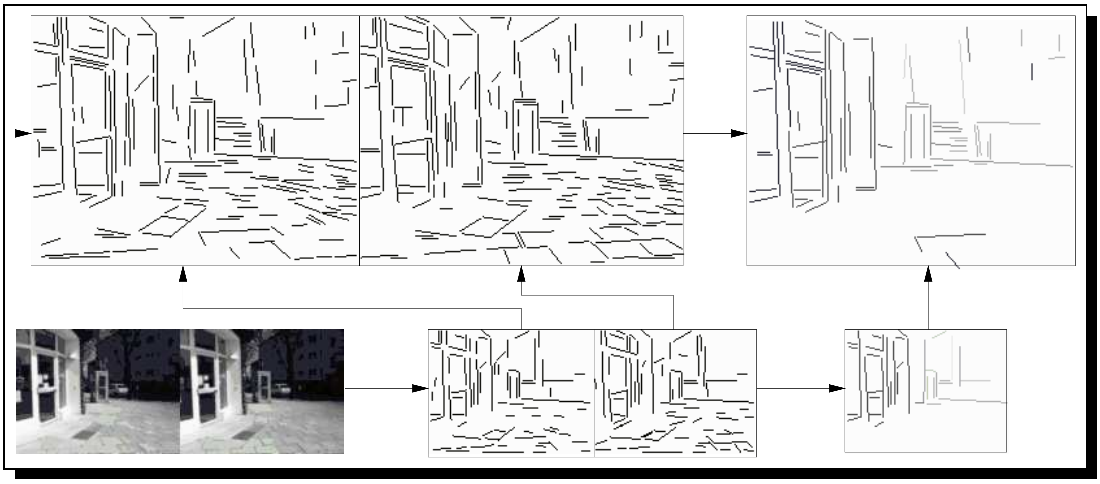


Abbildung 14: *Die Telefonzelle ist ca. 15 m weit von den Kameras entfernt: 3-D-Kanten auf der gröbsten und mittleren Auflösung sind auf der rechten Seite dargestellt. (Die Resultate bei voller Auflösung wurden nicht dargestellt.)*

$m_M(d_j)$ deren Gütefunktion. Die Gewichts-Koeffizient k_i und k_j hängen vom Abstand zwischen der aktuellen Linie d und der vorhandenen benachbarten Linie d_i ab:

$$k_i = S_{x_{\max}}(|x(d) - x(d_i)| + x_{\max}) \frac{\text{overlap}(l(d), l(d_i))}{\min(l(d), l(d_i))}$$

mit dem maximalen Abstand der Korrelation $x_{\max}$ und der horizontalen Koordinate x.

Aufgrund der Fragmentierung von Linien und Abweichungen in der Linien-Extraktion des linken und rechten Bildes wird die epipolare Einschränkung in Gleichung (18) nicht zwingend exakt erfüllt.

3.2.4 Beispiele für die Kantenextraktion

Um die Vorteile des neues Ansatzes, insbesondere die leistungsfähige Linien-Extraktion unter Erhaltung vollständiger Linien und Vermeidung unnötiger Details sowie das robustes Linien-Matching mit hoher Zuverlässigkeit und niedrigem Rechenaufwand zu demonstrieren, werden in Abb. 14 und Abb. 15 zwei Beispiele anhand realer Straßenszene dargestellt. Man kann sehen, daß die für die Landmarkenerkennung notwendigen Tiefeninformationen tatsächlich auf der mittleren Auflösungsebene (sogar schon auf der gröbsten Ebene) sehr zufriedenstellend ermittelt werden. Aus Aufwandsgründen werden nur relevante Merkmale gut fokussiert und die Schätzung anderer kleiner Details unterdrückt.

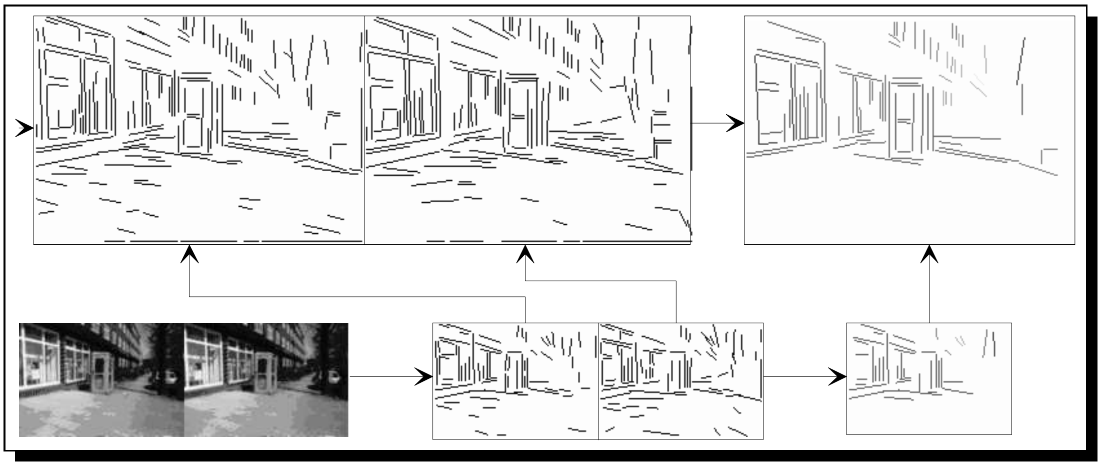


Abbildung 15: *Die Telefonzelle ist ca. 10 m weit von den Kameras entfernt: 3-D-Kanten auf der gröbsten und mittleren Auflösung sind auf der rechten Seite dargestellt.*

3.3 Farbanalyse

Wie die Farbwahrnehmung eine wichtige Rolle im menschlichen Sehen spielt, so kann die Verwendung von Farbinformation beim Rechnersehen die Szenenbeschreibung und Objekterkennung ebenfalls oft vereinfachen. Farbbildverarbeitung und -analyse wurden aus diesem Grund in den vergangenen Jahren zunehmend untersucht (z. B. [33]). Dieser Abschnitt beschränkt sich auf die Extraktion der entsprechenden Farbinformationen zur Landmarkenerkennung. Dafür wurde ein neuer Fuzzy-Algorithmus entwickelt.

Zur Verbesserung der Farbsegmentierung wurde die folgende neue Fuzzy-Regel verwendet:

IF ein Farb-Pixel hat mehr oder weniger die visuelle Farbe vC und eine benachbarte Region der visuellen Farbe vC

Then das Pixel hat die echte Farbe eC.

Daraus ergibt sich beispielsweise die folgende **Definition**:

$$m_{eC}(P) = \min(\sqrt{m_{vC}(P)}, \max_i(m_{vC}(R_i)))$$

mit $m_{vC}(R_i) = \text{average}_{P' \in R_i} m_{vC}(P')$

Die Membership-Funktion eines Pixels bezüglich der visuellen Farbe vC kann wie folgt mit Hilfe des durch $I = (R+G+B)$ normierten Farbraums definiert werden:

$$m_{vC}(R, G, B) \quad = \quad f(I) m_{vC}(r, g)$$

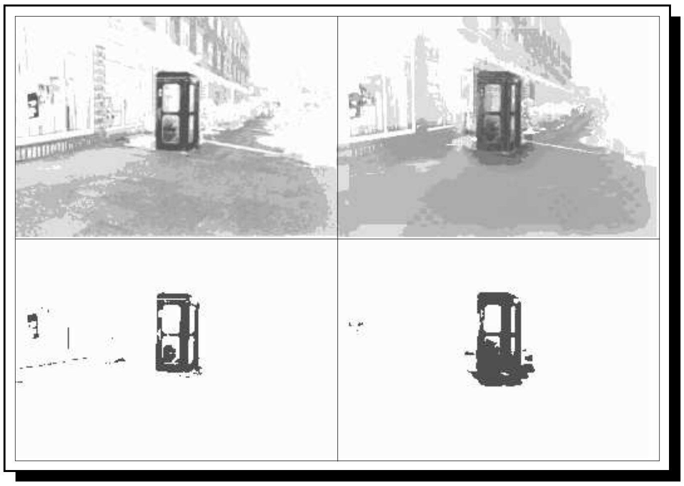


Abbildung 16: *oben links: Membershipwerte der visuellen Gelbfarbe, oben rechts: Membership der echten Gelbfarbe, unten links und rechts: Binarisierung der obigen Bilder (Schwellwert 0.55).*

$$m_{vC}(r,g) \quad = \quad \left(1 - \frac{x^2}{a^2} - \frac{y^2}{b^2}\right), \text{ mit } \begin{pmatrix} x \\ y \end{pmatrix} = R \begin{pmatrix} r - r_0 \\ g - g_0 \end{pmatrix}$$

Hierbei ist $r = 255R/I$ und $g = 255G/I$; (r_0, g_0) ist das Zentrum der visuellen Farbe vC im normierten Farbraum. Die Parameter a, b und R charakterisieren die Verteilung der Zugehörigkeit zur Farbe vC im normierten Farbraum.

In Abb. 16 wird die Segmentierung der gelben Farbe in dem Farbbild aus Abbildung 15 dargestellt, es sind $r_0 = 155$, $g_0 = 100$, $a = 100$, $b = 35$ und $R = I$. Das Verfahren sorgt dafür, daß nach der Vorverarbeitung die anderen kleinen Farbsegmente als Störung unterdrückt und die Farbsegmente auf der Landmarke verstärkt werden können.

Entsprechende Fuzzy-Regeln für die Farbanalyse lassen sich selbstverständlich in gleicher Weise für den IHS-Farbraum aufstellen. Diese haben bei uns zu hervorragende Segmentierungsergebnissen geführt, werden hier aber nicht näher diskutiert.

Basierend auf der Farbvorverarbeitung erfolgt ein Region-Merging zur Ermittlung der endgültigen Farbsegmentierung. Die daraus gewonnene Farbsegmentierung wird dann durch Integration mit der Tiefeninformationen zur Landmarkenerkennung verwendet.

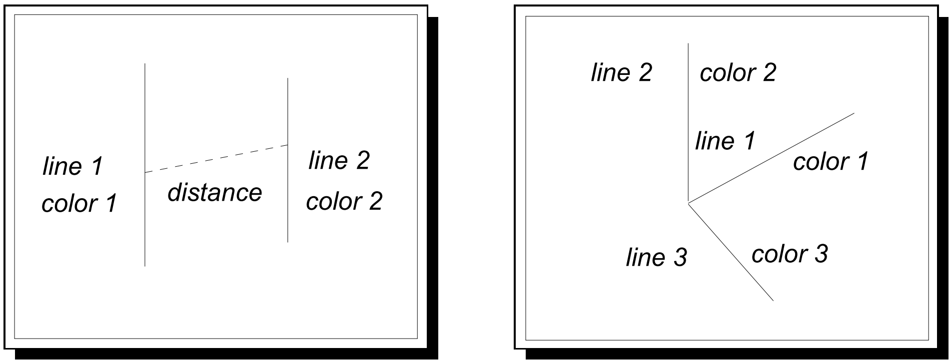
3.4 Der Gesamtalgorithmus

Basierend auf Stereo- und Farbinformationen haben wir ein Verfahren zur Segmentierung und Repräsentation von 3-D-Szenen für die 3-D-Landmarkenerkennung vorgeschlagen. Da fast alle Objekte, die eine größere Bedeutung für die freie Navigation von Blinden im Außenraum haben, künstliche Objekte sind, die sich nur durch ihre Farbe und 3-D-Kanten beschreiben lassen, ist es möglich, eine Szene nur auf Grundlage dieser beiden Merkmale zu segmentieren und zu analysieren. Wie weiter oben beschrieben, wird hierbei ein Lösungsansatz zur Integration der Tiefen- und Farbinformation zur eindeutigen Erkennung von 3-D-Objekten verwendet.

Die vollständige Modellierung eines 3-D-Objektes benötigt mehrere Aufnahmen aus verschiedenen Ansichten. Dieses gilt ebenfalls für die anschließende Modell-gestützte Erkennung. Für reale Problemstellungen wie die unsere sind diese Anforderungen jedoch nicht oder nur schwer zu erfüllen. Typischerweise erhält man nur eine Ansicht einer Landmarke und muß die Landmarke basierend auf dieser einen Ansicht lokalisieren und erkennen. Um dieses Problem zu lösen, haben wir ein Verfahren entwickelt, das auf der von Stein und Medioni in [43, 44] vorgeschlagenen Methode des *structure indexing* basiert. Anders als das ursprüngliche Verfahren, das nur einen Typ der lokalen Struktur — ein *Super-Segment* — verwendet, berücksichtigen wir mehrere Super-Segmente, um die komplizierten Objekten eindeutig zu modellieren und falsche Hypothesen zu unterdrücken.

In unserem Verfahren wird ein Objekt oder eine Szene durch Super-Segmente repräsentiert. Basierend auf dieser Repräsentation kann eine Hashing-Tabelle angelegt werden, die die Super-Segmente aller Objekte mit den jeweiligen Modellen verknüpft. Der Matching-Prozeß besteht dann nur noch aus einem Tabellenzugriff basierend auf den in einer Szene detektierten Segmenten. Hierdurch wird die Definition geeigneter Super-Segmente zum wichtigsten Schritt dieses Verfahrens. Ist die Definition zu allgemein, werden zu viele Hypothesen ermittelt (False-Positives). Ist andererseits die Definition zu speziell, so werden zu wenige Hypothesen generiert und deshalb letztendlich Landmarken übersehen (False-Negatives).

Entsprechend den typischen 3-D-Szenen im Außerraum, insbesondere den für Blinde wichtigen Landmarken wie z. B. Telefonzelle, Verkehrsampel und Briefkasten, führen wir zwei Arten von Super-Segmenten ein. Sie basieren auf 3-D-Kanten und Farbinformationen, wie sie durch das Stereo-Matching [26] und die Farbsegmentierung [25] ermittelt wurden. Das erste Super-Segment S_1 besteht aus einem Paar paralleler Linien mit in etwa gleicher Tiefe und Höhe (vgl. Abb. 17 a). Deren Attribute — die wirklichen Längen und Lage, der Abstand und die Farben um die Linien — werden zur Beschreibung verwendet. Das zweite Super-Segment S_2 besteht aus drei in etwa koinzidenten Linien (ein 3-D rechter



(a) (b)

Abbildung 17: *Zwei Arten von Super-Segmenten: (a) ein Paar paralleler Linien und (b) drei koinzidente Linien, ein 3-D rechter Winkel.*

(a) (b) (c)

Abbildung 18: *(a) Szenebild eines Stereopaares, (b) 3-D-Tiefenkarte, (c) Erkennungsergebnisse: Telefonzelle und Briefkasten.*

Winkel, vgl. Abb. 17 b). Dieses wird durch die wirkliche Länge und Lage der Linien sowie deren Farbe beschrieben.

Unter Verwendung der oben definierten Super-Segmente können wir eine Hashing-Tabelle aufbauen, in der die Zusammenhänge zwischen Super-Segmenten und allen verwandten 3-D-Modellen gespeichert sind. Basierend auf dieser Tabelle ist der Erkennungsprozeß durchzuführen. Als Beispiel betrachten wir die Szene in Abb. 18 a), in der neben einer Reihe irrelevanter Objekte eine Telefonzelle und ein Briefkasten als wichtige Landmarken zu sehen sind. Aufgrund der vorhandenen Stereoinformation (nicht dargestellt) berechnen wir die 3-D-Kanten (Abb. 18 b). Zusätzlich extrahieren wir die Farben in der Nähe dieser Kanten (nicht dargestellt). Wir identifizieren in Frage kommende Super-Segmente und suchen in der Hashing-Tabelle nach passenden Modell-Kandidaten. Das Ergebnisse besteht normalerweise aus mehrere Hypothesen. Aus diesem Grund ist die Einführung eines Prozesses notwendig, der die Konsistenz zwischen Szene und

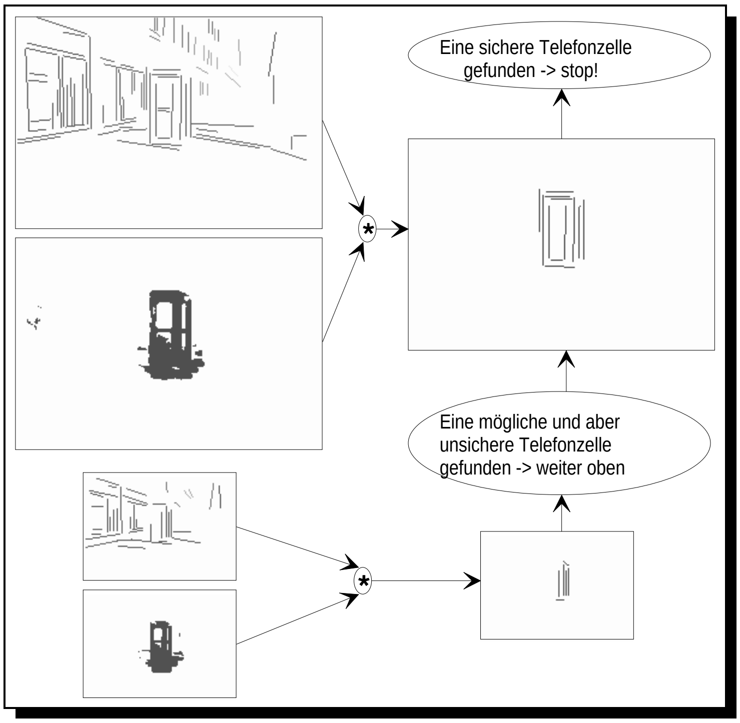


Abbildung 19: *Das Beispiel aus Abbildung 15 und 16 zur Demonstration der Integration von Tiefen- und Farbinformationen: Die Telefonzelle wird schon auf der gröbsten Auflösungsebene unsicher bzw. auf der mittleren Ebene sicher erkannt; damit ist die Analyse auf der feinsten Auflösungsebene unnötig und deshalb hier auch nicht dargestellt (links: Tiefenkarte und Farbsegmentierung, rechts: selektierte Telefonzelle).*

Modell berechnet. Nur der Match mit dem höchsten Wert der Konsistenz wird akzeptiert. In Abb. 18 c) werden die Erkennungsergebnisse durch die graue Unterlegung markiert.

Zusammenfassend verwendet das hier vorgeschlagene Verfahren die Fuzzy-Technik zur zuverlässigen Extraktion der grundlegenden Merkmale (3-D-Kanten und Farbinformation) und charakterisiert die 3-D-Szene durch eine neue Modellierung, die es erlaubt, 3-D-Landmarken mit geringem Rechenaufwand zuverlässig zu erkennen, ohne daß eine exakte Modellierung der gesuchten Objekte benötigt wird. Ein weiterer wichtiger Vorteil des Verfahrens ist die Robustheit der Erkennung gegenüber teilweiser Verdeckung der Landmarken durch andere Objekte, und gegenüber unterschiedlichen Ansichten.

In Abb. 19 wird noch einmal der pyramidale Aufbau des Algorithmus dargestellt. Es zeigt sich, daß das Verfahren für reale Szenen mit zufriedenstellenden Ergebnissen verwendet werden kann.

4 Voruntersuchung zur Erkennung von Verkehrszeichen

4.1 Einleitung

In diesem Abschnitt geht es um die Erkennung von Verkehrszeichen. Im Rahmen des Projektes MOVIS haben wir uns auf eine Reihe von für Blinde wichtiger Verkehrszeichen konzentriert [15], insbesondere die Zeichen „Fußgängerüberweg“ (StVO Zeichen 350), das Schild „Bus-Haltestelle“ (StVO Zeichen 224), das Schild „Taxenstand“ (StVO Zeichen 229) und das Hinweisschild mit dem Emblem der Telekom, welches sich auf dem Dach der neueren Telefonzellen befindet.

In der heutigen Zeit spielen Schilder jeglicher Art eine immer größere Rolle. Sie regeln das in den letzten Jahrzehnten erhöhte Verkehrsaufkommen, sie weisen uns den Weg zu vielen Infrastruktureinrichtungen, die wir tagtäglich nutzen, und geben uns so eine Hilfestellung, uns gerade in dicht besiedelten und vielleicht fremden Gegenden besser zurecht zu finden. Der normalsehende Mensch, der die Informationsfülle durch Schilder mit wenigen Blicken erfaßt und verarbeitet, wird vermutlich als selbstverständlich erachten, was blinden und sehbehinderten Menschen verborgen bleibt.

Im Rahmen des BMBF-Projekts MOVIS wurde untersucht, in wieweit es möglich ist, ein System zu entwickeln, das Bilder aus Alltagssituationen mit einer sehr kleinen Videokamera aufnimmt, verarbeitet und dem Blinden dann relevante Informationen in akustischer Form übermittelt. Auf diesem Wege sollen ihm neue Möglichkeiten eröffnet werden, sich außerhalb seiner gewohnten Umgebung zurechtzufinden. Die Anforderungen unterscheiden sich hierbei stark von denen der Verkehrszeichenerkennung in Systemen zur automatischen Fahrzeugführung oder Fahrerunterstützung (vgl. z. B. [36]). Insbesondere sind wir im Rahmen des Projektes MOVIS aus Platz- und Gewichtsgründen gezwungen, Kameras mit wesentlich geringerer farblicher und räumlicher Auflösung zu verwenden. Darüber hinaus erscheinen Verkehrszeichen aus der Sicht eines Fußgängers wesentlich stärker (projektiv) verzerrt als aus der Sicht eines Autofahrers, der für ihn relevante Verkehrszeichen in aller Regel immer frontal von vorne betrachtet. Und schließlich kann für ein Auto relativ sicher vorhergesagt werden, in welchem Bereich des Bildes sich Verkehrszeichen befinden können, während dies für einen Fußgänger, insbesondere ohne Information über die Kameraposition relativ zur Straße, unmöglich ist.

Die Abschnitte 4.2 bis 4.4 beschreiben erste, vielversprechende Untersuchungen auf der Basis eingescannter Papierbilder, die im Arbeitsbereich Technische Informatik I im Rahmen von Studienarbeiten realisiert wurden [2, 6, 50, 53]. Abschnitt 4.5 geht auf die Probleme bei der Anpassung der Algorithmen an die im Rahmen des Projektes MOVIS verwandte Kamera der Firma teli und den Framegrabber der Firma Imaging Technologie ein (vgl. [50]).

Das verwandte Vorgehen läßt sich grob in zwei Teilschritte unterteilen. In einem ersten Schritt werden aufgrund von Farbinformationen Regions of Interest bestimmt. Abschnitt 4.2 stellt den verwandten Algorithmus sowie die zugrundeliegende Theorie vor. Daran anschließend werden nacheinander auf jeder dieser Regions Geometrie-Merkmale berechnet und mit dem erwarteten Erscheinungsbild des betreffenden Schildes unter projektiver Verzerrung verglichen (Abschnitt 4.3). Einige Ergebnisse dieses Vorgehens werden schließlich in Abschnitt 4.4 vorgestellt, und Abschnitt 4.5 gibt einen kurzen Überblick über die Probleme beim Übergang auf die im Rahmen des Projektes MOVIS verwandte Miniaturkamera, sowie einen Ausblick auf mögliche zukünftige Entwicklungen.

4.2 Farbsegmentierung

Wesentliches Merkmal beim Entwurf von Verkehrszeichen ist die Signalwirkung der Schilder. Aus diesem Grund beschränken sich Schilder auf einige wenige, stark gesättigte Farben [18] und simple Formen wie Kreise, Dreiecke und Vierecke[9]. Deshalb lag es nahe, in einem Farbbild zuerst nach Gebieten (Regions of Interest) zu suchen, deren Farbe das Vorhandensein eines Verkehrszeichens vermuten laßt, und erst daran anschließend die jeweiligen Gebiete auf entsprechende Geometriemerkmale zu untersuchen.

Standardmäßig werden die Farbwerte von der Kamera als RGB-Tripel geliefert. Eine derartige Darstellung ist jedoch für die weitere Verarbeitung nicht sehr geeignet, da sie nicht helligkeitsinvariant ist. Besser geeignete Farbräume sind zum Beispiel der HSV-Hexkonus, der HSL-Doppelhexkonus oder der IHS-Doppelkegel (vergleiche z. B. [3, 17, 34, 38]). In allen drei Fällen ist der Farbton ($H =$ Hue) weitgehend Invariant gegenüber Helligkeitsschwankungen.

Im Rahmen des Projektes MOVIS haben wir uns für eine leicht abgewandelte Form des IHS-Modells entschieden (Intensity, Hue, Saturation). In [14] wurde ein Algorithmus für die Konvertierung vorgeschlagen, der den Farbton vereinfacht und damit genähert berechnet, der aber dennoch für uns ausreichend ist. In [38] wird eine Vereinfachung für die Sättigung angegeben. Die Berechnung der Intensität ist in allen genannten Literaturstellen einheitlich. Für die Transformation sind folgende Gleichungen implementiert:

[9] Eine offensichtliche Ausnahme ist hier das Stoppschild.

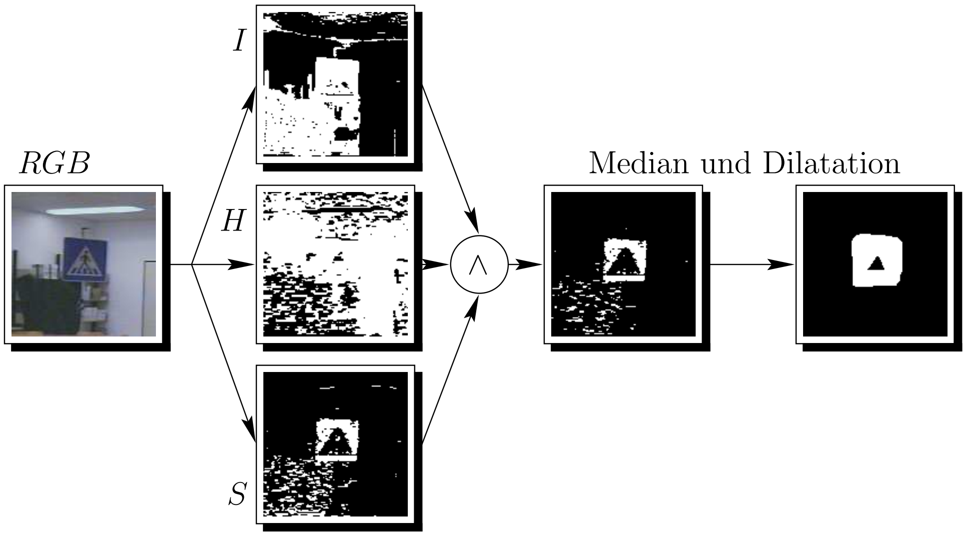


Abbildung 20: *Farbsegmentierung*

Der Intensität wird als arithmetischer Mittelwert der drei Bänder berechnet:

$$I = \frac{R+G+B}{3} \tag{23}$$

Die Sättigung ist gegeben durch:

$$S = (1 - \frac{\min(R,G,B)}{I}) \tag{24}$$

Aufgrund der Vereinfachung wird der IHS-Raum nur noch als Einfachkegel statt als Doppelkegel dargestellt. Der Algorithmus für die Bestimmung des Farbtons (Hue) ist ebenfalls vereinfacht, er nähert den Winkel mit Verhältnissen. Die Berechnung ist davon abhängig, welcher der drei Farbwerte R, G und B maximal ist, daher muß hier eine Fallunterscheidung getroffen werden:

$$H = \begin{cases} & \frac{60(G-B)}{\max(R,G,B)-\min(R,G,B)} & , R > G, B \\ 120.0 & + \quad \frac{60(B-R)}{\max(R,G,B)-\min(R,G,B)} & , G > R, B \\ 240.0 & + \quad \frac{60(R-G)}{\max(R,G,B)-\min(R,G,B)} & , B > R, G \end{cases} \tag{25}$$

Gleichung 25 berechnet den Farbton als Winkel zur Rotachse. Entsprechend kann natürlich eine beliebige andere Nullage verwendet werden. Dieses ist Aufgrund der Modulo-Eigenschaften des Farbtons (Farbwinkels) häufig sinnvoll.

Im folgenden wird nur noch auf den I, H und S Bändern des Bildes gearbeitet (vgl. Abb. 20). Zuerst wird jedes Band einer Schwellwert-Operation unterzogen, die nur Pixel mit dem richtigen Farbton selektiert. Zusätzlich wird durch die Einschränkung der für Sättigung und Intensität möglichen Werte der in Frage kommende Bereich weiter eingeschränkt. Die resultierenden Masken werden mit logischen Operatoren verknüpft (typischerweise eine UND-Verknüpfung). Anschließend wird die so entstandene Maske einer Medianfilterung[10] und daran anschließend einer morphologischen Operation (typischerweise einer Dilatation) unterzogen. Das Resultat sind i. a. mehrere geschlossene Regionen. Ein simpler Konturfolger [46] findet für jede Region die sie umgebende Bounding-Box. Diese wird, um einige Prozent vergrößert, den nachfolgenden Algorithmen (vgl. Abschnitt 4.3) zur Verfügung gestellt. Wesentliche Teile des Ablaufs sind noch einmal in Abbildung 20 dargestellt.

4.3 Projektive Geometrie

Dieser Abschnitt beschreibt die Routinen vom vorliegen einer Region of Interest bis hin zum abschließenden Mustervergleich (vgl. Abbildung 21). Alle hier beschriebenen Routinen arbeiten statt auf dem ganzen Bild nur auf den in Abschnitt 4.2 detektierten Regions of Interest (ROIs). Dadurch wird es möglich, relativ komplexe Operationen effizient zu verwenden.

Zu Beginn werden auf jeder ROI Kantenpunkte detektiert. Hierzu wird das in Abschnitt 2.4.1 und 2.4.2 beschriebene Verfahren nach Canny [11] eingesetzt. Durch die derart gefundenen Kantenpunkte werden dann je nach gesuchtem Schild Geraden oder Ellipsen gelegt. Um die Geraden zu finden wird der in Abschnitt 2.4.3 beschriebene Algorithmus benutzt, Ellipsen werden mit Hilfe der von Bookstein [5] und Pratt [35] beschriebenen Algorithmen gefunden.

Daran anschließend werden für das gesuchte Schild charakteristische Konstellationen von Linien und gegebenenfalls Ellipsen gesucht. Z. B. wird für das Zeichen Fußgängerüberweg (StVO Zeichen 350) nach einem hellen Dreieck und einem das Dreieck umschließenden Viereck gesucht. Eine gewisse Toleranz gegenüber Verdeckungen kommt dadurch zustande, daß nicht zwingend alle vier Linien des Vierecks wirklich detektiert werden müssen.

Mit Hilfe der so gefundenen Linien und Kreise wird eine Transformation bestimmt, die das vermutete Schild in eine Normalform abbildet. Dieses ist auf mehrere Arten möglich. So ist es z. B. möglich die Transformation T zu bestimmen, welche die gefundenen Linien möglichst gut auf die Kanten eines bekannten Prototypen projiziert. Stattdessen wurde im Rahmen des Projektes MOVIS der Weg verfolgt, erst einmal die Schnittpunkte aller Geraden und Ellipsen miteinander sowie deren Kovarianzen zu bestimmen. Die so gefundenen Schnittpunkte

[10] Im binären Fall ist die Medianfilterung trivial.

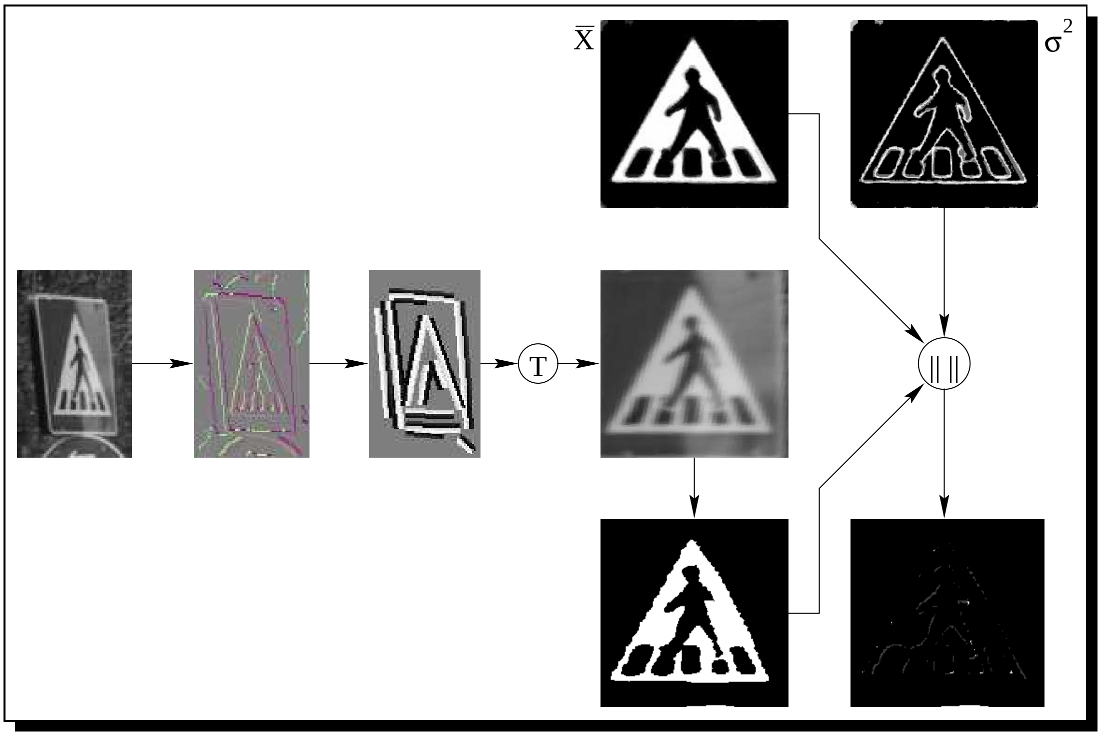


Abbildung 21: *Grouping und Verifikation.*

wurden dann in Normalform transformiert.

Schließlich wird die gesammte ROI in Normalform transformiert, das resultierende Bild binarisiert und der Abstand zwischen der transformierten ROI und der Normalform bestimmt. Um Problemen an Helligkeitsübergängen aus dem Weg zu gehen, die nie in genau die gleiche Position transformiert werden, wird als Abstandsmaß ein gewichteter Euklidischer Abstand benutzt, der durch eine entsprechend hohe Varianz an den Helligkeitübergängen Probleme mit der Positionierung ausgleicht.

4.4 Beispiele und Ergebnisse

Insgesamt konnte in [53] ein sehr robustes Verfahren implementiert werden, daß auf eingescannten Fotografien der Auflösung 1024 pxl × 682 pxl × 24 Bit eine Erkennungsrate von rund 60 % erreichte. Nicht erkannte Schilder fielen fast ausnahmslos in eine der drei Kategorien

1. Falsch belichtete (zu dunkle) Schilder.

2. Zu kleine Schilder (Mindestgröße ca. 25 pxl × 25 pxl).

Abbildung 22: *Beispiel für nicht erkannte Schilder (eingekreist). Nur das mittlere Bushaltestellenschild wurde erkannt, das linke Schild ist zu stark verdeckt, das rechte zu dunkel.*

3. Teilweise verdeckte Schilder.

Abbildung 22 zeigt ein Beispiel für teilweise verdeckte oder zu dunkle Schilder. Die Mehrzahl der Verkehrszeichen konnte jedoch problemlos erkannt werden, und die Abbildungen 23 und 24 zeigen einige Beispiele für erkannte Zebrastreifen-Schilder. Hierbei sind die näher untersuchten Regions of Interest in grün markiert, und die gefundenen Schilder in rot.

Abbildung 23: *Beispiele erkannter Schilder. ROIs sind grün umrahmt, die erkannten Zebrastreifenschilder rot.*

Abbildung 24: *Beispiele erkannter Schilder. ROIs sind grün umrahmt, die erkannten Zebrastreifenschilder rot.*

4.5 Probleme und Ausblick

Obwohl es in[2, 6, 53] gelungen ist, einen Verkehrszeichenerkenner zu entwerfen, der auf eingescannten Fotografien unterschiedlicher Herkunft robust Schilder mit Erkennungsraten um 60 % erkennen konnte, haben sich bei der Portierung des Algorithmus auf die im Rahmen des Projekts MOVIS benutzte Hardware, insbesondere die Kamera der Firma teli und den Framegrabber der Firma Imaging Technologie einige Probleme ergeben. Diese Probleme gehen im wesentlichen auf drei Ursachen zurück, nämlich die geringere räumliche Auflösung, die geringere farbliche Auflösung und Probleme bei der IHS-Konvertierung.

In [2, 53] wurden Aufnahmen der Auflösung 1024 pxl × 682 pxl zugrunde gelegt. Die im Rahmen des Projektes MOVIS verwandte Kamera der Firma teli hat jedoch nur eine Auflösung von 752 pxl × 582 pxl interlaced. Um Problemen mit einem bewegungsbedingten Versatz zwischen den Halbbildern aus dem Weg zu gehen, arbeiten die Algorithmen nur auf einem Halbbild, die Auflösung verringert sich folglich weiter zu 752 pxl × 291 pxl. Darüber hinaus verringert sich die tatsächlich zur Verfügung stehende Auflösung noch einmal durch die in der Kamera zum Einsatz kommende 1-Chip Technik. Hierbei werden jeweils vier benachbarte Pixel der Farben Gelb, Cyan, Magenta und Grün zu einem Farbwert zusammengefaßt. Hierdurch ergibt sich letztendlich eine Linienauflösung von etwa 470 Linien × 420 Linien für ein Vollbild oder entsprechend 470 Linien × 210 Linien für ein Halbbild[11]. Dies entspricht einer um 85 % reduzierten Fläche gegenüber den hochaufgelösten Aufnahmen in [53]. Darüber hinaus bringt die Verwendung einer 1-Chip Kamera noch weitere Probleme mit sich. Typischerweise weisen die Farbwerte zwar den gleichen Mittelwert, aber eine weit größere Streuung auf. Der Farbwert eines Pixels hängt stark nicht nur von der beobachteten Objektfarbe ab, sondern ebenso von der jeweiligen Position des Pixels im Bild.

Das dritte Problem ergibt sich durch die IHS-Konvertierung. Aus Zeitgründen wurden statt der Gleichungen (23), (24) und (25) die in den Framegrabber der Firma Imaging Technologie eingebaute Konvertierungsroutine benutzt. Diese implementiert die folgenden Gleichungen[12]:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} 0.498 & -0.249 & -0.249 \\ 0.333 & 0.333 & 0.333 \\ 0.0 & 0.431 & -0.413 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{26}$$

$$H = \arctan\left(\frac{B'}{R'}\right) \tag{27}$$

[11] Aus Gründen, die mit dem verwendeten Framegrabber zusammenhängen, verwenden wir in MOVIS im Allgemeinen eine Auflösung von 512 pxl × 286 pxl.

[12] Nachdem wir in Zusammenarbeit mit dem Hersteller einige Algorithmische Fehler beseitigen konnten.

$$I = G' \tag{28}$$
$$S = \sqrt{R'^2 + B'^2} \tag{29}$$

Wie man leicht sieht, realisiert Gleichung (29) keinen Kegel, sondern bloß einen Zylinder. Es zeigt sich, daß der damit zur Verfügung stehende Wert der Sättigung quasi wertlos ist, für eine erfolgreiche Klassifizierung muß zeitaufwendig noch einmal jeder S-Wert durch den zugehörigen I-Wert (oder $\min(I, 255 - I)$ — ein Doppelkegel) geteilt werden.

Trotzdem ist es in [50] gelungen, das Verfahren soweit an die erschwerten Randbedingungen anzupassen, daß auch hier Erkennungsraten von ca. 60 % erzielt werden konnten. Leider hat es sich jedoch gezeigt, daß der Algorithmus hierdurch einen Großteil seiner ursprünglichen Robustheit eingebüßt hat, er reagiert in seiner jetzigen Form extrem empfindlich auf Änderungen in der Beleuchtungsfarbe. Als eine mögliche Lösung dieses Problems wurde in [50] ein Algorithmus entwickelt, der ein kontinuierliches Lernen während des Betriebs erlaubt und so in der Lage sein könnte, sich an veränderte Beleuchtungsbedingungen zu adaptieren. Inwieweit sich dieses Vorgehen in der Praxis bewähren würde konnte aber im Rahmen der ersten Phase von MOVIS noch nicht getestet werden.

Andere Möglichkeiten, der durch die Verwendung einer 1-Chip Kamera aufgeworfenen Probleme — insbesondere der starken Schwankung der Farbwerte von Pixel zu Pixel — Herr zu werden, bietet zum Beispiel die Wahl von Schwellwerten in Abhängigkeit von der Pixel-Position oder die Einbeziehung von Nachbarschafts-Information, zum Beispiel in einem pyramidalen Ansatz ähnlich Abschnitt 3. Inwieweit dieser Ansatz, der für relativ große Farbflächen (Telefonzellen, Briefkästen) entwickelt wurde, sich auf die relativ kleinen Verkehrszeichen übertragen läßt konnte jedoch in der ersten Phase nicht mehr geprüft werden.

Abbildung 25: *Kamerakopf mit Objektiv in Originalgröße* 60 mm × 17 mm *(L×ϕ).*

Abbildung 26: *Brillenteil des Online-Demonstrators.*

5 Die MOVIS-Blindbrille: Eine Online-Demonstration

5.1 Einleitung

Im Rahmen des Projektes MOVIS wurde ein Online-Demonstrator gebaut, der bei der Begehung durch das DLR am 2. Oktober 1997 zum Einsatz kam. Hiermit konnte eindrucksvoll die Wirkungsweise und volle Funktionalität der in den Abschnitten 2 und 3 beschriebenen Verfahren inklusive einer Sprachausgabe der Detektionsergebnisse demonstriert werden.

Im folgenden werden die einzelnen Komponenten des Demonstrationsaufbaus beschrieben. Der eigentliche Demonstrator bestand aus einer mobilen und einer stationären Komponente (Abschnitt 5.2 und 5.3), die durch ein ca. 30 m langes Kabel miteinander verbunden wurden. Die Ausgabe der Ergebnisse erfolgte akustisch (mittels Sprachausgabe, Abschnitt 5.4), entweder direkt durch die Stereokopfhörer des Brillenteils, oder aber (zu Demonstrationszwecken) durch Lautsprecher. Abschnitt 5.5 beschreibt den Demonstrationsaufbau, so wie er am 2. Oktober 1997 existierte.

Abbildung 27: *Stationäre Rechnerkomponente.*

5.2 Die mobile Komponente

Im Demonstrator kam als mobile Komponente eine von der Firma Jurca entworfene und gefertigte Brille mit Taschenteil zum Einsatz. Diese enthält neben zwei Miniaturkameras der Firma teli (vgl. Abbildung 25 und 26) noch ein elektronisches Lot, Stereokopfhörer sowie die gesamte Steuerelektronik und Stromversorgung. Da diese Brille in einem separaten Bericht der Firma Jurca Optoelektronik genauer beschrieben ist, wird hier nicht weiter auf sie eingegangen.

5.3 Die stationäre Komponente

Für den Demonstrator wurde die mobile Komponente über ein ca. 30 m langes Kabel mit einem stationären Rechner der Firma Parsytec verbunden. Ursprünglich war geplant gewesen, die Algorithmen auf einem handelsüblichen PC unter Einsatz einer speziellen Bildvorverarbeitungshardware zu implementieren. Da diese Hardware (das vom Mikroelektronik Anwendungs Zentrum — MAZ — entwickelte Board Gipsi) jedoch letztendlich nicht zur Verfügung stand, kam statt dessen ein Parallelrechner der Firma Parsytec mit 8 PowerPC Prozessoren zum Einsatz.

Auf diesem Rechner wurden mit Hilfe des parallelen Bildverarbeitungssystems PIPS (Parallel Image Processing System, vgl. Abschn. 5.3.1) die besonders zeitaufwendigen Algorithmen der Bildvorverarbeitung implementiert, insbesondere z. B. die Farbsegmentierung in Abschnitt 3.3 oder der Canny Kantenfilter bis hin zur Linienextraktion in Abschnitt 2.4 (siehe auch [28]). Das der Parallelrechner tatsächlich nur die Aufgabe spezieller Bildverarbeitungshardware übernimmt zeigt sich in der Tatsache, daß quasi alle High-Level Routinen bereits in der aktuellen Version nur auf einem Prozessor implementiert sind. Zwar würde eine weitergehende Parallelisierung hier noch (geringe) Zeitvorteile bringen, die exemplarische Realisierung auf nur einem Prozessor zeigt jedoch, daß MOVIS nicht mehr sehr weit von einem komplett mobilem Gerät entfernt ist.

Der folgende Abschnitt gibt einen kurzen Überblick über das parallele Bildverarbeitungssystem PIPS.

5.3.1 PIPS

Aufbauend auf Forschungsergebnissen zur Parallelisierung von Algorithmen der digitalen Bildverarbeitung und Mustererkennung wurde das Bildverarbeitungssystem PIPS für massiv parallele Rechner implementiert [32]. Abbildung 28 links zeigt eine typische Systemkonfiguration. Dargestellt ist die Verteilung eines von der Kamera mittels eines Graphik-Prozessors aufgenommenen Bildes auf vier Prozessoren des parallelen Systems. Neben den Prozessoren des parallelen Systems und den für die Bildaufnahme und -wiedergabe nötigen Komponenten ist ein Prozessor als Interface zum Host-Rechner bzw. zur Anbindung an ein Netzwerk erforderlich. Diese Aufgabe wird im Falle des CC-Systems der Firma Parsytec zusätzlich von einem der 8 Prozessoren übernommen.

Um eine größtmögliche Portierbarkeit des Bildverarbeitungssystems zu erreichen, wurden sämtliche Hardwareabhängigen Systemkomponenten in separaten Modulen gekapselt (vgl. Abbildung. 28 rechts). Durch den schichtenweisen Aufbau wird es möglich, die Komponenten des Bildverarbeitungssystems unabhängig voneinander zu optimieren, und sich so dem Ziel einer möglichst effizienten Implementierung der Algorithmen auf unterschiedlichen Ebenen zu nähern.

Basis des Systems ist ein Kommunikations-Modul, dessen Funktionalität (im wesentlichen Datenaustausch zu benachbarten Prozessoren) auf eine Vielzahl von Parallelrechnerarchitekturen abgebildet werden kann. Durch dieses Konzept wird eine einfache Portierbarkeit von PIPS gewährleistet. Darauf aufbauend wurde ein Message Passing System (MEPS) entwickelt. Aus Benutzersicht stellt MEPS eine virtuell vollständige Verbindung zwischen allen auf den Prozessoren residierenden Prozessen her.

Das auf MEPS basierende PIPS-Modul stellt Mechanismen zur Verfügung, die Bilddaten dem Algorithmus angepaßt zu verteilen, dem Anwender transparent zuzuführen und nach der Berechnung zur Ausgabe einzusammeln. Eine Vielzahl von grundlegenden Algorithmen der Bildverarbeitung wurde in Form einer Bibliothek

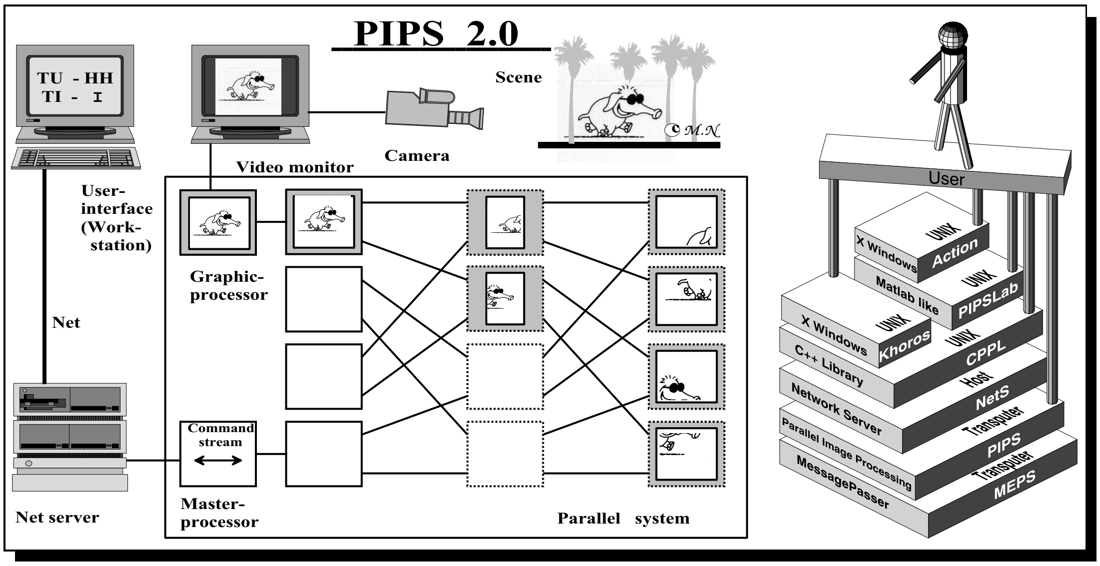


Abbildung 28: *Typische System-Architektur für das Bildverarbeitungssystem PIPS.*

in dieses Modul integriert. Auf dieser Ebene kann die Effizienz der Algorithmen beispielsweise durch Minimierung von Kommunikations- und Rechenaufwand, der Wahl einer geeigneten Datenverteilung sowie durch Überlappung von Berechnung und Kommunikation gesteigert werden.

Grundlage der Implementierung sind theoretisch erarbeitete Methoden zur effizienten Einbettung von Algorithmen mit lokalen Kommunikationsanforderungen in ein für Algorithmen mit globalen Abhängigkeiten optimales Netzwerk sowie Ergebnisse über die Skalierbarkeit der Algorithmen auf Netzwerke verschiedener Dimension. (vgl. z. B. auch [1]).

Von vernetzten Workstations aus kann mittels der in C++ implementierten Bibliothek PAPI (PIPS Application Programming Interface) auf den Parallelrechner zugegriffen werden. Aufbauend auf parallelisierten Einzeloperationen können auf dieser Ebene Bildverarbeitungsalgorithmen implementiert werden, wobei der Parallelrechner aus Benutzersicht einem schnellen Bildverarbeitungs-Coprozessor entspricht.

Mit Hilfe von PIPS sind im Rahmen des Projektes MOVIS mehrere der Low-Level Bildverarbeitungsroutinen auf einem skalierbaren CC-System der Firma Parsytec mit zur Zeit 8 PowerPC-Prozessoren implementiert worden, um die Erkennung von 2-D- und 3-D-Landmarken in quasi-Echtzeit zu gewährleisten. Die Low-Level Bildverarbeitungsroutinen wie z. B. der Canny-Kantenfilter eignen sich hierzu besonders gut, da sie zum einen sehr Rechenzeitintensiv sind, sich jedoch zum anderen besonders effizient parallelisieren lassen, da es sich hierbei überwiegend um lokale Operationen handelt. Da das CC-System über einen PCI-Bus mit dem Framegrabber verbunden ist, ist es darüber hinaus leicht möglich, das System im Nachhinein durch spezifische zusätzliche Hardware wie z. B. einen speziellen Prozessor zur Bildvorverarbeitung (z. B. Gipsi) zu ergänzen. Die Ergebnisse der auf dem Parallelrechner implementierten Low-Level Routinen können anschließend direkt auf einem der Knoten oder auf einem gesonderten Rechner zur Weiterverarbeitung übernommen werden.

5.4 Die Sprachausgabe

Die berechneten Ergebnisse werden schließlich über eine Sprachausgabe dem Blinden zur Verfügung gestellt. Da das CC-System der Firma Parsytec dem CHRP-Standard (Common Hardware Reference Platform) entspricht, verfügt es selbstverständlich auch über eine Karte zur Audioausgabe. Aus diesem Grund wurde die Sprachausgabe direkt vom CC-System aus gesteuert, wir haben dafür den Soundtreiber „Open Sound System for AIX“ der Firma 4Front Technologies zusammen mit IBMs text-to-speech Demo verwandt, die als Bestandteil der Ultimedia Services (UMS) zum Betriebssystem AIX gehört.

5.5 Der Demonstrationsaufbau

Um den oben beschriebenen Demonstrator unter möglichst realitätsnahen Bedingungen testen zu können, wurden im Foyer eines Gebäudes der TU Hamburg-Harburg in der Harburger Schloßstraße 20 eine originale Telefonzelle und Briefkasten sowie ein auf 80 % der Originalgröße verkleinerter Zebrastreifen aufgebaut. Abbildung 29 zeigt den Grundriß des Demo-Aufbaus, und Abbildung 30 einige typische Ansichten, die direkt mit dem Demonstrator aufgenommen wurden.

Anhand des Demo-Aufbaus konnte bewiesen werden, daß es prinzipiell möglich ist, Telefonzellen, Briefkästen und Zebrastreifen in quasi-Echtzeit unter realistischen Bedingungen und mit im wesentlichen bereits portabler Harware zu erkennen und dem Blinden somit ein wertvolles Hilfsmittel für die Orientierung im typischen Alltags-Situationen an die Hand zu geben. Abbildung 31 zeigt zum Abschluß noch einmal eine Reihe von Zebrastreifen, wie sie vom Demonstrator in unserem Demo-Aufbau identifiziert wurden, sowie die zugehörige errechnete Aufsicht.

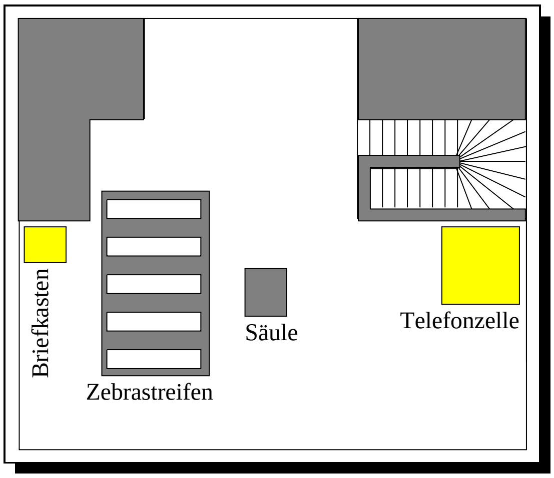


Abbildung 29: *Schematischer Aufbau der Demo-Szene.*

Abbildung 30: *Einige Beispielansichten des Demoaufbaus, aufgenommen mit dem Demonstrator.*

Abbildung 31: *Einige Beispiele für erkannte Zebrastreifen im Demoaufbau. Rechts ist jeweils der Zebrastreifen und links eine simulierte Sicht aus der Vogelperspektive.*

6 Zusammenfassung und Ausblick

In diesem Bericht wurden die Ergebnisse beschrieben, die am Arbeitsbereich Technische Informatik I der Technischen Universität Hamburg Harburg unter Leitung von Prof. Dr.-Ing. H. Burkhardt in der ersten Phase des Projekts MOVIS erzielt werden konnten. Insbesondere handelt es sich hierbei um die Entwicklung und Implementierung eines 3-D-Landmarken Erkenners am Beispiel von Telefonzellen und Zebrastreifen, eines 2-D-Landmarken Erkenners am Beispiel von Zebrastreifen und die Integration dieser beiden Module in einem Online-Demonstrator, der überzeugend die Durchführbarkeit eines Projektes wie MOVIS demonstrieren konnte. Zusätzlich wurden erste Voruntersuchung zur Erkennung von Verkehrsschildern aus beliebigen Richtungen (projektive Verzerrungen) durchgeführt

Zur Erkennung von 3-D-Landmarken haben wir einen neuen pyramidalen Ansatz unter Verwendung von Tiefen- und Farbinformationen vorgestellt. Zur Extraktion der Merkmale dreidimensionaler Landmarken werden zwei Algorithmen entwickelt, eine Linien-basierte pyramidale Stereobildauswertung und eine Farbanalyse. In beiden Algorithmen spielt Fuzzy Logik eine wichtige Rolle. 3-D-Kanten und Farbsegmenten werden dazu benutzt, die 3-D-Landmarken mit Hilfe von geometrischem Hashing zu modelliert, was zu einer einfachen Beschreibung von Landmarken führt, so daß sie zuverlässig erkannt werden können. In diesem Bericht wurden Telefonzellen und Briefkästen verschiedener Bauformen als typische 3-D-Landmarken verwendet.

Der Algorithmus zur Erkennung von 2-D-Landmarken ist in der Lage, Zebrastreifen selbst bei teilweiser Verdeckung oder starker Beschädigung sicher und zuverlässig zu erkennen. Möglich wurde dies durch eine Kombination aus projektiver Geometrie und Statistik, insbesondere durch die konsequente Anwendung von Methoden der Fehlerfortpflanzung. Hierdurch ist es gelungen einen Algorithmus zu entwickeln, der mit einem Minimum an Parametern, die noch dazu rein vom Bildaufnahmesystem abhängen und, einmal gewählt, nicht mehr verändert werden müssen, eine robuste Erkennung unter den unterschiedlichsten Aufnahmebedingungen ermöglicht. Beide Algorithmen wurden in einem Demonstrator kombiniert, der Online die Funktionsfähigkeit der hier vorgestellten Algorithmen unter Beweis gestellt hat.

Darüber hinaus wurden erste, vielversprechende Voruntersuchungen zur Erkennung von Verkehrszeichen in Farbbildern minderer Qualität (geringe räumliche und farbliche Auflösung) durchgeführt. Auch wenn noch keine abschließende Lösung präsentiert wurde zeichnet sich doch ab, daß die Erkennung von Verkehrszeichen unter beliebigen projektiven Verzerrungen selbst mit einer 1-Chip Miniaturkamera, wie sie im Rahmen des Projektes MOVIS Verwendung findet, durchaus möglich ist.

Das endgültige Ziel des Projektes MOVIS ist es, ein komplettes System zur Navigationshilfe für Blinde zu entwickeln.

Basierend auf Bildmerkmalen, insbesondere Farb- und geometrischen Informationen, und Tiefendaten ist der Ausbau hin zu einer verallgemeinerte Modellierung von 2-D- und 3-D-Landmarken zu erwarten, der das automatische oder doch zumindest halb-automatische erlernen weiterer Landmarken in greifbare Nähe rückt, selbst wenn dies zur Zeit aus Gründen der erwünschten Verarbeitungszeit noch nicht realisiert ist. Insgesamt konnte in der ersten Phase des Projektes MOVIS die prinzipielle Durchführbarkeit unseres ambitionierten Vorhabens unter Beweis gestellt werden. Für die nächste Phase wäre schon bald ein erster, halb-kommerzieller Prototyp zu erwarten, der Freilandversuche stark erleichtern würde. Insgesamt zeichnet sich für MOVIS eine arbeitsreiche aber vielversprechende Zukunft ab, soweit eine weitere Förderung sichergestellt werden kann.

Literatur

[1] T. Andreae, M. Nölle und G. Schreiber. Cartesian Products of Graphs as Spanning Subgraphs of de Bruijn Graphs. In *Proceedings of the 20. Workshop on Graph-Theoretic Concepts in Computer Science (WG94)*, S. 140–150. Lecture Notes in Computer Science 903, Springer, Juni 1994.

[2] Mirko Appel. Erkennung ausgesuchter Schilder unter Berücksichtigung von Meßungenauigkeiten. Studienarbeit, Arbeitsbereich Technische Informatik I, TU Hamburg Harburg, Januar 1997.

[3] Dana H. Ballard und Christopher M. Brown. *Computer Vision*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[4] F. Bergholm. Edge focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:729–741, 1987.

[5] Fred L. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9:56–71, 1979.

[6] Markus Breyer. Detektion von Polygonen geringer Größe und Komplexität in natürlichen Szenen basierend auf Kanteninformation. Studienarbeit, Arbeitsbereich Technische Informatik I, TU Hamburg Harburg, April 1997.

[7] B. Brillault-O'Mahony. New method for vanishing point detection. *CVGIP: Image Understanding*, 54(2):289–300, September 1991.

[8] I. N. Bronstein und K. A. Semendjajew. *Taschenbuch der Mathematik*. G. Grosche and V. Ziegler and D. Ziegler, Harri Deutsch, Thun und Frankfurt (Main), 23. Ausgabe, 1987.

[9] Bundesanstalt für Straßenwesen (BASt). Zusätzliche technische Vorschriften und Richtlinien für Markierungen auf Straßen ZTV-M 84. FGSV, September 1984.

[10] J. B. Burns, A. R. Hanson und E. M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:425–455, 1986.

[11] John F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[12] John C. Clarke. Modelling uncertainty: A primer. Submitted to Image and Vision Computing, Dezember 1996.

[13] Robert T. Collins und Richard S. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *Third International Conference on Computer Vision*, S. 400–403, Osaka, Japan, Dezember 1990. IEEE Computer Society, IEEE Computer Society Press.

[14] Conrac Corporation. *Raster Graphics Handbook*. Van Nostrand Reinhold, New York, 2 Ausgabe, 1985.

[15] empirica. Mögliche Anwendungsszenarien für Movis. MOVIS interner Bericht, April 1995.

[16] Olivier Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, Cambridge, Massachusetts, 1. Ausgabe, 1993.

[17] Rafael C. Gonzalez und Paul Wintz. *Digital Image Processing*. Addison-Wesley, Reading, Mass., 1987.

[18] Normenausschuß Farbe (FNF) im DIN Deutsches Institut für Normung e.V. DIN 6171, Teil 1: Aufsichtfarben für Verkehrszeichen. Beuth Verlag GmbH, Berlin 30, März 1989.

[19] Kenichi Kanatani. Computational cross ratio for computer vision. *CVGIP: Image Understanding*, 60(3):371–381, November 1994.

[20] Kenichi Kanatani. Statistical analysis of geometric computation. *CVGIP: Image Understanding*, 59(3):286–306, Mai 1994.

[21] Kenichi Kanatani. Statistical foundation for hypothesis testing of image data. *CVGIP: Image Understanding*, 60(3):382–391, November 1994.

[22] Y. Liu und T. S. Huang. Determining straight line correspondences from intensity images. *Pattern Recognition*, 24:489–504, 1991.

[23] A. Luo und H. Burkhardt. Ein helligkeitsbasiertes Stereoverfahren zur Tiefenschätzung. In G. Sagerer, S. Posch und F. Kummert, Hrsg., *17. DAGM - Symposium "Mustererkennung"*, S. 508–515, Bielefeld, 1995. Reihe Informatik aktuell, Springer.

[24] A. Luo und H. Burkhardt. An Intensity-Based Cooperative Bidirectional Stereo Matching with Simultaneous Detection of Discontinuities and Occlusions. *International Journal of Computer Vision*, 15:171–188, 1995.

[25] A. Luo, W. Tao und H. Burkhardt. Erkennung von 3-D Landmarken basierend auf Stereo-Tiefendaten und Farbinformationen zur Navigationshilfe für Blinde. In *18. DAGM - Symposium "Mustererkennung"*, Heidelberg, 1996. Reihe Informatik aktuell, Springer.

[26] A. Luo, W. Tao und H. Burkhardt. A new multilevel line-based stereo vision algorithm based on fuzzy techniques. In *Proc. of the 13th International Conference on Pattern Recognition*, S. 383–387, Vienna, Austria, August 1996.

[27] Evelyne Lutton, Henri Maître und Jaime Lopez-Krahe. Contribution to the determination of vanishing points using hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):430–438, April 1994.

[28] Heiko Marschner. Parallelisierung von Algorithmen zur Gewinnung von Konturdaten und Linien aus Grauwertbildern. Diplomarbeit, Arbeitsbereich Technische Informatik I, TU Hamburg Harburg, Juli 1997.

[29] J. H. McIntosh und K. M. Mutch. Matching straight lines. *Computer Vision, Graphics and Image Processing*, 43:386–408, 1988.

[30] G. F. McLean und D. Kotturi. Vanishing point detection by line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1090–1095, November 1995.

[31] G. Medioni und R. Nevatia. Matching images using linear feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:675–685, 1984.

[32] M. Nölle, G. Schreiber und H. Schulz-Mirbach. PIPS–a general purpose Parallel Image Processing System. In W.G. Kropatsch und H. Bischof, Hrsg., *16. DAGM - Symposium "Mustererkennung"*, S. 609–623, Wien, September 1994. Reihe Informatik XPress, TU-Wien.

[33] A. E. K. Pienkowski. *Artificial Colour Perception using Fuzzy Techniques in Digital Image Processing*. Verlag TÜV Rheinland, 1989.

[34] Charles A. Poynton. *A Technical Introduction to Digital Video*. John Wiley & Sons, New York, 1995.

[35] Vaughan Pratt. Direct least-squares fitting of algebraic surfaces. *Computer Graphics*, 21(4):145–152, Juli 1987.

[36] Lutz Priese, Jens Klieber, Raimund Lakman, Volker Rehrman und Rainer Schian. Echtzeit–Verkehrszeichenerkennung mit dem Colour Strucure Code. Interner bericht, Labor Bilderkennen, Universität Koblenz–Landau, Universität Koblenz–Landau, Rheinau 1 56075 Koblenz, Germany, 1995.

[37] Charles A. Rothwell. *Recognition Using Projective Invariance*. Dissertation, University of Oxford, Oxford, U.K., Juli 1993.

[38] John C. Russ. *The Image Processing Handbook.* CRC Press, Boca Raton, Florida, 1992.

[39] Fabrizio Russo. Fuzzy techniques in image processing. European Course on Recent Advances in Image Processing, Arbeitsbereich Technische Informatik I, TU Hamburg-Harburg, September 1995.

[40] J.G. Semple und G.T. Kneebone. *Algebraic Projective Geometry.* Clarendon Press, Oxford, 1952.

[41] John G. Semple und G. T. Kneebone. *Algebraic Projective Geometry.* Clarendon Press, Oxford, U.K., 1952 (reprinted 1979).

[42] Shishir Shah und J.K. Aggarwal. Intrinsic parameter calibration procedure for a (high distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition*, 29(11):1775–1788, 1996.

[43] F. Stein und G. Medioni. Efficient two dimensional object recognition. In *10th International Conference on Pattern Recognition*, Atlantic City, USA, Juni 1990.

[44] F. Stein und G. Medioni. Structural indexing: Efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):125–125, 1992.

[45] Fahrbahnmarkierungen, internationale Übereinkommen. Straßenbau von A-Z. 1276–1986.

[46] Ralph-Peter Suhr. Entwurf und Parallelisierung eines Konturverfolgers für Binärbilder zur parallelen konturbasierten lageinvarianten Merkmalsgewinnung. Diplomarbeit, TU Hamburg-Harburg, Dezember 1996.

[47] UNO. Übereinkommen über Straßenverkehrszeichen, Wien, November 1968. in [45].

[48] H.A. vanDijck, M.J. Korsten und F. van der Heijden. Robust 3-dimensional object recognition using stereo vision and geometric hashing. In *Third International Conference on Image Processing*, S. 16–19, Lausanne, September 1996.

[49] V. Venkateswar und R. Chellapa. Extraction of straight lines in aerial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:1111–1114, 1992.

[50] Frank Vietze. Erkennung ausgewählter Verkehrszeichen unter Verwendung einer Kleinstkamera. Studienarbeit, Arbeitsbereich Technische Informatik I, TU Hamburg Harburg, Juli 1997.

[51] David H. Warren und Edward R. Strelow, Hrsg. *NATO Advanced Research Workshop on Visual Spatial Prostheses for the Blind*, Band E of *NATO ASI Series*, Lake Arrowhead, California, September 1984. NATO, Martinus Nijhoff Publishers, Dordrecht.

[52] Richard S. Weiss, Hiromasa Nakatami und Edward M. Riseman. An error analysis for surface orientation from vanishing points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1179–1185, Dezember 1990.

[53] Amin Yousif. Erkennung ausgesuchter Schilder basierend auf Farb- und Geometrieinformation. Studienarbeit, Arbeitsbereich Technische Informatik I, TU Hamburg Harburg, Juli 1996.

[54] X.-F. Zhang, A. Luo, W. Tao und H. Burkhardt. Camera calibration based on 3d-point-grid. In *Nineth International Conference on Image Analysis and Processing*, Florence, Italy, September 1997. Lecture Notes in Computer Science, 1310, Springer Verlag.