

Universität Dortmund

# Bewegungsschätzung mit Methoden der Objekterkennung

Diplomarbeit

**Aishy Amer**

Dezember 1994

Fakultät für Elektrotechnik  
Lehrstuhl für Nachrichtentechnik  
AG Schaltungen der  
Informationsverarbeitung  
Prof. Dr. Ing. Hartmut Schröder  
Dipl. Ing. Holger Blume

Fachbereich für Informatik  
Lehrstuhl Informatik I  
Prof. Dr. Bernd Reusch  
Dipl. Inform. Thomas Dettmer



*für Sara ...*

Diese Arbeit widme ich jeder Frau und jedem Mädchen in meiner Heimat, die den Willen zur Freiheit hat.

Ich danke meinen Freundinnen und Freunden und meiner Familie, die mir auf meinem Weg in der BRD viel Wärme und viel Liebe geschenkt haben.

Meinem Betreuer Herrn Dipl. Ing. Holger Blume danke ich für seine hervorragende Betreuung und die vielen hilfreichen Diskussionen und Anregungen während der Durchführung meiner Arbeit.

Mein Dank gilt auch meinem Betreuer Herrn Dipl. Inform. Thomas Dettmer für die wertvollen Impulse und für die jederzeit gewährte Unterstützung.

Herrn Prof. Dr. Ing. Hartmut Schröder und Herrn Prof. Dr. Bernd Reusch danke ich für die Übernahme der Begutachtung und dafür, daß ich meine Arbeit an ihren Lehrstühlen, dem Lehrstuhl für Nachrichtentechnik, AG Schaltungen der Informationsverarbeitung und dem Lehrstuhl Informatik I der Universität Dortmund, durchführen und anfertigen konnte.

Darüber hinaus danke ich dem katholischen akademischen Ausländerdienst (KAAD) für seine finanzielle Unterstützung.

*Es war einmal in Deutschland ...*

*Aishy Amer, Dezember 1994*

إلى ساره ...

أهدي دراسة الماجستير هذه إلى كل امرأة وكل فتاة في وطني تتوق إلى الحرية .  
عائلي ، صديقاتي وأصدقائي هُنَّ وهُم إمتناني لما أعطين وأعطوا لي من محبة وحنان على دربي  
في جمهورية ألمانيا الاتحادية .

جزيل شكري أوجهه إلى المشرف السيد هولجير بلومه لإشرافه الزايع خلال إعداد هذه  
الدراسة . وأشكر أيضاً المشرف السيد توماس دتمير على توجيهاته المختلفة .  
لأستاذي السيد شرودير ، البروفيسور في كلية هندسة الكهرباء ولأستاذي السيد رويش ،  
البروفيسور في كلية علم الحاسوب في جامعة دورتموند شكري على إشرافهم العام والنهاي على  
هذه الدراسة .

وأخيراً أوجه شكري إلى مؤسسة المنح ك آ آ د .

كان يا ما كان في ألمانيا . . .

عائشه عامر ، كانون أول ١٩٩٤

# Motion Estimation Using Object Recognition Methods

## Abstract

Motion estimation is one of the key techniques of modern digital video signal processing. Most of the estimation algorithms which are used today and which are implemented in dedicated hardware are block based algorithms (block-matching algorithms). Although they are easy to implement and the quality of their resulting vector fields is better than that of the methods which were used formerly (gradient methods, phase plane correlation) they still suffer from some drawbacks. Most important is that using block based algorithms there will arise also a block pattern in the vector field (block pattern noise).

Real objects in real scenes do not coincide with the block boundaries. Therefore object recognition methods shall be introduced into motion estimation. Within this work an object based motion estimation algorithm is carried out by a morphological segmentation of the image followed by a contour analysis of the objects and a contour filling. This object information is used to enhance the quality of the vector fields which are generated by an efficient parallel predictive motion estimation algorithms which is suitable for a receiver based motion estimation.

The vector field is homogenised within the object regions. If non translatory motion (e.g. camera zoom, rotation) is detected by an analysis of the vector histogram in object regions the homogenisation process is stopped and the original vector field of the parallel predictive algorithm is kept.

As a result of these improved motion estimation it can be seen that using the vector field for a vector based upconversion algorithm better interpolation quality is gained. Moving textured pictures content isn't disrupted any more as it was the case using standard non homogeneous vector fields.

At the end of this work the calculational costs of the elements of the new algorithm (morphological segmentation, contour analysis of objects, homogenisation of the vector fields by object information) are evaluated. It can be seen, that the calculational cost is only about 1/14 of the calculational cost of parallel predictive motion estimation which is not yet optimised in its operation count.

Therefore this new algorithm also seems to be well suited for realisation as it improves the quality of the vector fields and raises the calculational cost of the whole concept only by a small amount.

# Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Kurzfassung</b>  | <b>1</b>  |
| <b>2</b> | <b>Grundlagen der Bewegungsschätzung</b>                                    | <b>3</b>  |
| 2.1      | Einleitung . . . . .  | 3         |
| 2.2      | Bildpunktbasierte Bewegungsschätzung . . . . .                              | 5         |
| 2.2.1    | Vorbemerkung . . . . .  | 5         |
| 2.2.2    | Gradientenmethode . . . . .   | 6         |
| 2.2.3    | Die Fouriermethode . . . . .  | 8         |
| 2.2.4    | Die Suchmethode (Das Blockmatching) . . . . .                               | 8         |
| 2.2.5    | Das Grundkonzept des prädiktiven Blockmatching . . . . .                    | 10        |
| 2.3      | Objektbasierte Bewegungsschätzung . . . . .                                 | 14        |
| 2.3.1    | Vorbemerkung . . . . .  | 14        |
| 2.3.2    | Verbesserung der Bewegungsschätzung durch Vektorfeldsegmentierung . . . . . | 16        |
| <b>3</b> | <b>Prinzipielle Methoden der Objekterkennung</b>                            | <b>18</b> |
| 3.1      | Grundlagen und Vorbemerkung . . . . .                                       | 18        |
| 3.2      | Konturorientierte Objekterkennung . . . . .                                 | 21        |
| 3.2.1    | Grundlegende Definitionen . . . . .   | 21        |
| 3.2.2    | Objektkonturextraktion . . . . .  | 22        |
| 3.3      | Mathematische Morphologie . . . . .   | 30        |
| 3.3.1    | Vorbemerkung . . . . .  | 30        |
| 3.3.2    | Grundbegriffe . . . . .   | 30        |
| 3.3.3    | Einige morphologische Grundalgorithmen . . . . .                            | 34        |
| 3.4      | Regionenorientierte Objekterkennung in Bildsequenzen . . . . .              | 37        |
| 3.5      | Konturorientierte versus regionenorientierte Methoden . . . . .             | 38        |
| <b>4</b> | <b>Ein Konzept zur Erkennung von Objekten in Bildsequenzen</b>              | <b>41</b> |
| 4.1      | Problemstellung und Motivation . . . . .                                    | 41        |
| 4.2      | Das neue konturorientierte Objekterkennungsverfahren . . . . .              | 42        |
| 4.3      | Schwellwertbasierte Objektisolierung . . . . .                              | 49        |
| 4.4      | Morphologische Konturpunktdetektion . . . . .                               | 50        |
| 4.4.1    | Vorbemerkung . . . . .  | 50        |
| 4.4.2    | Realisierung eines morphologischen Konturpunktdetektors . . . . .           | 52        |
| 4.5      | Erkennung von geschlossenen Konturen . . . . .                              | 55        |

|          |   |           |
|----------|---|-----------|
| 4.5.1    | Vorbemerkung . . . . .  | 55        |
| 4.5.2    | Konturextraktion . . . . .  | 56        |
| 4.5.3    | Algorithmische Aspekte des Verfahrens . . . . .                         | 60        |
| 4.6      | Konturorientierte Rekonstruktion von Objekten . . . . .                 | 63        |
| 4.6.1    | Vorbemerkung . . . . .  | 63        |
| 4.6.2    | Konturenausfüllung . . . . .  | 63        |
| <b>5</b> | <b>Objektbasierte Bewegungsschätzung</b>                                | <b>69</b> |
| 5.1      | Einleitung und Problemstellung . . . . .                                | 69        |
| 5.2      | Integration von Objektinformationen in die Bewegungsschätzung . . . . . | 72        |
| 5.2.1    | Begriffsbestimmung . . . . .  | 72        |
| 5.2.2    | Homogenisierung der Objektbewegung . . . . .                            | 75        |
| 5.2.3    | Homogenisierung der Hintergrundbewegung . . . . .                       | 77        |
| 5.2.4    | Ergänzung der Objektpunkte . . . . .                                    | 78        |
| 5.2.5    | Betrachtung anderer Bewegungsformen . . . . .                           | 78        |
| <b>6</b> | <b>Simulationsergebnisse</b>  | <b>81</b> |
| 6.1      | Vorbemerkung . . . . .  | 81        |
| 6.2      | Das Simulationsprogramm MORPHOBJ zur Erkennung von Objekten . . . . .   | 81        |
| 6.3      | Das Simulationsprogramm OBJMOTION . . . . .                             | 83        |
| 6.4      | Vergleichende Ergebnisse . . . . .                                      | 85        |
| 6.5      | Einige Ergebnisse des Simulationsprogramms MORPHOBJ . . . . .           | 88        |
| <b>7</b> | <b>Abschließende Betrachtungen</b>                                      | <b>93</b> |
| 7.1      | Zusammenfassung . . . . .   | 93        |
| 7.2      | Ausblick . . . . .  | 95        |
|          | <b>Literaturverzeichnis</b>   | <b>98</b> |

# Abbildungsverzeichnis

|      |   |    |
|------|---|----|
| 2.1  | Blockschaltbild eines konventionellen Fernsehsystems . . . . .        | 4  |
| 2.2  | Zeilensprungverfahren . . . . .                                       | 5  |
| 2.3  | Interpolation von Zwischenbildern . . . . .                           | 6  |
| 2.4  | Gradient einer translatorisch bewegten Kante . . . . .                | 7  |
| 2.5  | Das Prinzip der Suchmethode . . . . .                                 | 9  |
| 2.6  | Anordnung der Prädiktionsblöcke . . . . .                             | 11 |
| 2.7  | Anordnung der Korrekturwerte . . . . .                                | 12 |
| 2.8  | Rasterangepaßte Uminterpolation . . . . .                             | 13 |
| 2.9  | Anordnung der Kontrollblöcke . . . . .                                | 13 |
| 2.10 | Flußdiagramm des Blockmatching nach [Blum92] . . . . .                | 15 |
|      |   |    |
| 3.1  | Bestandteile einer Szene . . . . .                                    | 20 |
| 3.2  | Darstellung von Nachbarschaftsrelationen . . . . .                    | 22 |
| 3.3  | Lokale Faltung mittels eines Gradientenoperators . . . . .            | 25 |
| 3.4  | Typische Masken einiger Konturpunktdetektoren . . . . .               | 26 |
| 3.5  | Vergleich von Konturpunktdetektoren: Originalbild . . . . .           | 26 |
| 3.6  | Konturpunktdetektion des Robertsoperators . . . . .                   | 27 |
| 3.7  | Konturpunktdetektion des Sobeloperators . . . . .                     | 27 |
| 3.8  | Konturpunktdetektion des Laplaceoperators . . . . .                   | 28 |
| 3.9  | Konturpunktdetektion des Operators nach Wu . . . . .                  | 28 |
| 3.10 | Ergebnis des im Rahmen dieser Arbeit entwickelten Operators . . . . . | 29 |
| 3.11 | Translation einer morphologischen Menge . . . . .                     | 31 |
| 3.12 | Häufig verwendete Formen von Strukturelementen . . . . .              | 32 |
| 3.13 | Morphologische Operationen: Das Originalbild $S$ . . . . .            | 32 |
| 3.14 | Auswirkung einer Dilatation . . . . .                                 | 33 |
| 3.15 | Auswirkung einer Erosion . . . . .                                    | 33 |
| 3.16 | Dilatation und Erosion eines Gebiets mit einem Diskelement . . . . .  | 34 |
| 3.17 | Opening von $S$ durch $E$ . . . . .                                   | 35 |
| 3.18 | Closing von $S$ durch $E$ . . . . .                                   | 36 |
| 3.19 | Ergebnis eines morphologischen Konturpunktdetektors . . . . .         | 36 |
| 3.20 | Grundkonzept regionenorientierter Verfahren . . . . .                 | 37 |
|      |   |    |
| 4.1  | Flußdiagramm des vierstufigen Objekterkennungsverfahrens . . . . .    | 43 |
| 4.2  | Ein Originalbild aus der „car“-Sequenz . . . . .                      | 45 |
| 4.3  | Das schwellwertbasierte Zweipegebild zu Bild 4.2 . . . . .            | 46 |
| 4.4  | Das resultierende Konturpunktbild zu Bild 4.3 . . . . .               | 47 |



|      |   |    |
|------|---|----|
| 4.5  | Das Konturenbild zu Bild 4.4 . . . . .                                      | 48 |
| 4.6  | Die rekonstruierten (erkannten) Objekte aus Bild 4.5 . . . . .              | 49 |
| 4.7  | Die ausgewählten Strukturelemente . . . . .                                 | 52 |
| 4.8  | Das dilatierte Bild . . . . .   | 53 |
| 4.9  | Das aus der XOR-Operation resultierende Bild . . . . .                      | 54 |
| 4.10 | Flußdiagramm des entwickelten Konturpunktdetektors . . . . .                | 55 |
| 4.11 | Darstellung einer Kontur . . . . .  | 58 |
| 4.12 | Globale Suchstrategie zur Schließung von Lücken . . . . .                   | 60 |
| 4.13 | Flußdiagramm des Konturerkennungsverfahrens . . . . .                       | 62 |
| 4.14 | Ausfüllung von sich überschneidenden Konturen . . . . .                     | 65 |
| 4.15 | Suche nach einem inneren Punkt . . . . .                                    | 66 |
| 4.16 | Ein Originalbild aus der „tussy“ -Sequenz . . . . .                         | 67 |
| 4.17 | Bild von sich überschneidenden Konturen . . . . .                           | 67 |
| 4.18 | Die rekonstruierten (erkannten) Objekte . . . . .                           | 68 |
| 4.19 | Das homogenisierte Vektorfeld . . . . .                                     | 68 |
| 5.1  | Blockdiagramm des objektbasierten Bewegungsschätzers . . . . .              | 71 |
| 5.2  | Integration der Objektinformationen in die Bewegungsschätzung . . . . .     | 72 |
| 5.3  | Objektbasierte Optimierung der Bewegungsvektoren . . . . .                  | 75 |
| 5.4  | Beispiel einer Rotation . . . . .   | 79 |
| 5.5  | Flußdiagramm der objektbasierten Homogenisierung der Vektorfelder . . . . . | 80 |
| 6.1  | Das zu optimierende Vektorfeld (X-Komponente) . . . . .                     | 83 |
| 6.2  | Resultierendes Vektorfeld (X-Komponente) . . . . .                          | 85 |
| 6.3  | Beispiel 1: Ein Bild aus einer künstlich erzeugten Sequenz . . . . .        | 89 |
| 6.4  | Die erkannten Objekte aus Bild 6.3 . . . . .                                | 89 |
| 6.5  | Beispiel 2: Ein Bild aus der „prlcar“ -Sequenz . . . . .                    | 90 |
| 6.6  | Die erkannten Objekte aus Bild 6.5 . . . . .                                | 90 |
| 6.7  | Beispiel 3: Ein Bild aus der „tennis“ -Sequenz . . . . .                    | 91 |
| 6.8  | Die erkannten Objekte aus Bild 6.7 . . . . .                                | 91 |
| 6.9  | Beispiel 4: Ein Bild aus der „voiture“ -Sequenz . . . . .                   | 92 |
| 6.10 | Die erkannten Objekte aus Bild 6.9 . . . . .                                | 92 |

# Kapitel 1

## Kurzfassung

Der größte Teil der Informationen über den Zustand und die Veränderung unserer Umgebung wird über das wichtigste Sinnesorgan des Menschen, das Auge, aufgenommen. Menschen orientieren sich bei der Wahrnehmung von Objektbewegungen im wesentlichen an der Umrandung von Objekten, d.h. an ihren Ecken und Kanten und ordnen dabei jedem Objekt eine Bewegung zu. Das Interesse an einer automatischen Erkennung und Bestimmung von Bewegungen, auf optischen Informationen und charakteristischen Objekteigenschaften basierend, wuchs in den letzten Jahren erheblich. Die Vielfalt der Einsatzmöglichkeiten, wie z.B. die Auswertung von Satellitenaufnahmen, die Überwachung von Gefahrenbereichen, die Anwendungen in der Fernsehtechnik usw., begründet das zunehmende Interesse an leistungsfähigen Algorithmen zur Bewegungsschätzung.

In der Videosignalverarbeitung ist die Bedeutung der Bewegungsschätzung erheblich gestiegen, da Bewegungsinformationen über die sich bewegenden Objekte in Bildern, vor allem zur Qualitätsverbesserung der Bildwiedergabe, eine wichtige und notwendige Grundlage darstellen. Hierbei sind die Anforderungen an einen Algorithmus zur Bewegungsschätzung sehr hoch, da einerseits die wahre Bewegung der Objekte so gut wie möglich ermittelt, andererseits aber der Hardwareaufwand auf der Empfängerseite gering gehalten werden soll. Eine Bewegungsschätzung, die sich an den menschlichen Wahrnehmungseigenschaften und an den Objekten orientiert, stellt eine wichtige Grundlage für leistungsfähige Verfahren dar, die möglichst genau einen Bewegungsvektor pro Objekt (Bewegung/Objekt-Beziehung) liefern sollen.

Das Ziel des im Rahmen der vorliegenden Arbeit vorgestellten, in mehreren Schritten entwickelten, objektbasierten Verfahrens, liegt zunächst in der Erkennung und Extraktion von den im Bild enthaltenen Strukturen, die möglichst gut mit den realen Objekten übereinstimmen. Die so gewonnenen Objektinformationen werden dann zur Verbesserung und Unterstützung der Bewegungsschätzung verwendet.

Nach einer Einführung in den Themenbereich „Bewegungsschätzung“ folgt eine Beschreibung von Methoden zur Erkennung von Objekten in Bildsequenzen. Diese Beschreibung wird durch grundlegende Begriffsdefinitionen ergänzt. Die ersten beiden Kapitel beinhalten die theoretischen Grundlagen für die entwickelten Verfahren zur objektbasierten Bewegungsschätzung. Die folgenden Kapitel enthalten zahlreiche Verweise auf diese theoretischen Grundlagen.

Ein mehrstufiges Verfahren zur konturorientierten Erkennung von Objekten in Bildern wird in Kapitel 3 entworfen, welches, durch Extraktion der Objektkonturen und anschließender Rekonstruktion der Objekte, Strukturen in Bilddaten herausfindet und sie eindeutig beschreibt. Um die Bildstrukturen, besonders ihre Form, zu extrahieren, wird die mathematische Morphologie (Lehre von Strukturen und Formen) als Grundlage verwendet. Morphologische Grundoperatoren bilden hier den Kern des vorgestellten Erkennungsalgorithmus.

Das vierte Kapitel enthält eine Beschreibung eines objektbasierten Ansatzes, welcher gewonnene Objektinformationen zur Optimierung von Vektorfeldern, mit Hilfe von aufgestellten Regeln, in die Bewegungsschätzung integriert.

In Kapitel 5 werden Simulationsergebnisse, zur Verdeutlichung der Wirksamkeit der entwickelten Verfahren, der Objekterkennung und der Bewegungsschätzung aufgezeigt. In Kapitel 6 werden noch einmal die wichtigsten Aspekte des entwickelten Verfahrens zur objektbasierten Bewegungsschätzung zusammengefaßt. Hier werden auch einige Erweiterungsmöglichkeiten und Anregungen für weitere Forschungsarbeiten gegeben.

# Kapitel 2

## Grundlagen der Bewegungsschätzung

### 2.1 Einleitung

In einem Fernsehübertragungssystem werden die Bilder zeilenweise abgetastet. Abbildung 2.1 zeigt ein Blockschaltbild eines konventionellen Fernsehsystems. Die Kamera projiziert eine dreidimensionale Szene auf eine zweidimensionale Ebene. In der Regel arbeiten die heutigen Fernsehsysteme mit der sogenannten Zeilensprungabtastung. Die Abtastung des projizierten Bildsignals  $s(x,y,t)$  erfolgt zeitlich und örtlich in einer Sequenz von Halbbildern im Abstand  $T$ . Das Zeilenraster in jedem Halbbild besitzt dabei einen Rasterabstand von  $2d$ , mit  $d$  als der vertikale Abstand zwischen zwei Zeilen eines Vollbildes. Von Halbbild zu Halbbild wird das Zeilenraster um einen Abstand  $d$  versetzt. Nach dem zeilenweisen Durchlaufen des ersten Halbbildes bewegt sich der abtastende Strahlspot zur ersten Zeile des nächsten Halbbildes fort (Abbildung 2.2<sup>1</sup>).

Ein Bildwiedergabegerät in einem Fernsehübertragungssystem hat die Aufgabe, in einer angemessenen Zeit und mit möglichst wenig Hardwareaufwand das abgetastete, übertragene Bild zu rekonstruieren.

Trotz der großen Erfolge des Fernsehens besteht die Meinung, daß die Bildqualität unzureichend ist<sup>2</sup>. Hier stören insbesondere Effekte wie das 50Hz Großflächenflimmern oder das Kantenflackern und Zeilenwandern die subjektiv wahrgenommene Bildqualität. Zur Verbesserung der Bildqualität in Videosequenzen auf der Empfängerseite werden Wiedergabekonversionen eingesetzt, d.h. Zwischenbilder werden interpoliert (z.B. mittels einer Aufwärtskonversion 50Hz Zeilensprung auf 100Hz Zeilensprung). Abbildung 2.3 zeigt das Grundkonzept der Zwischenbildinterpolation.

Zur Interpolation von Zwischenbildern existieren eine Vielzahl von Algorithmen. Diese werden in erster Linie in statische, bewegungsadaptive und bewegungsvektorgestützte Verfahren unterteilt. Während die statischen Verfahren aus zwei Originalbildern ohne Verwendung einer Zusatzinformation ein Zwischenbild berechnen, verwenden be-

---

<sup>1</sup> vgl. [Wend88]

<sup>2</sup>[Wend88]

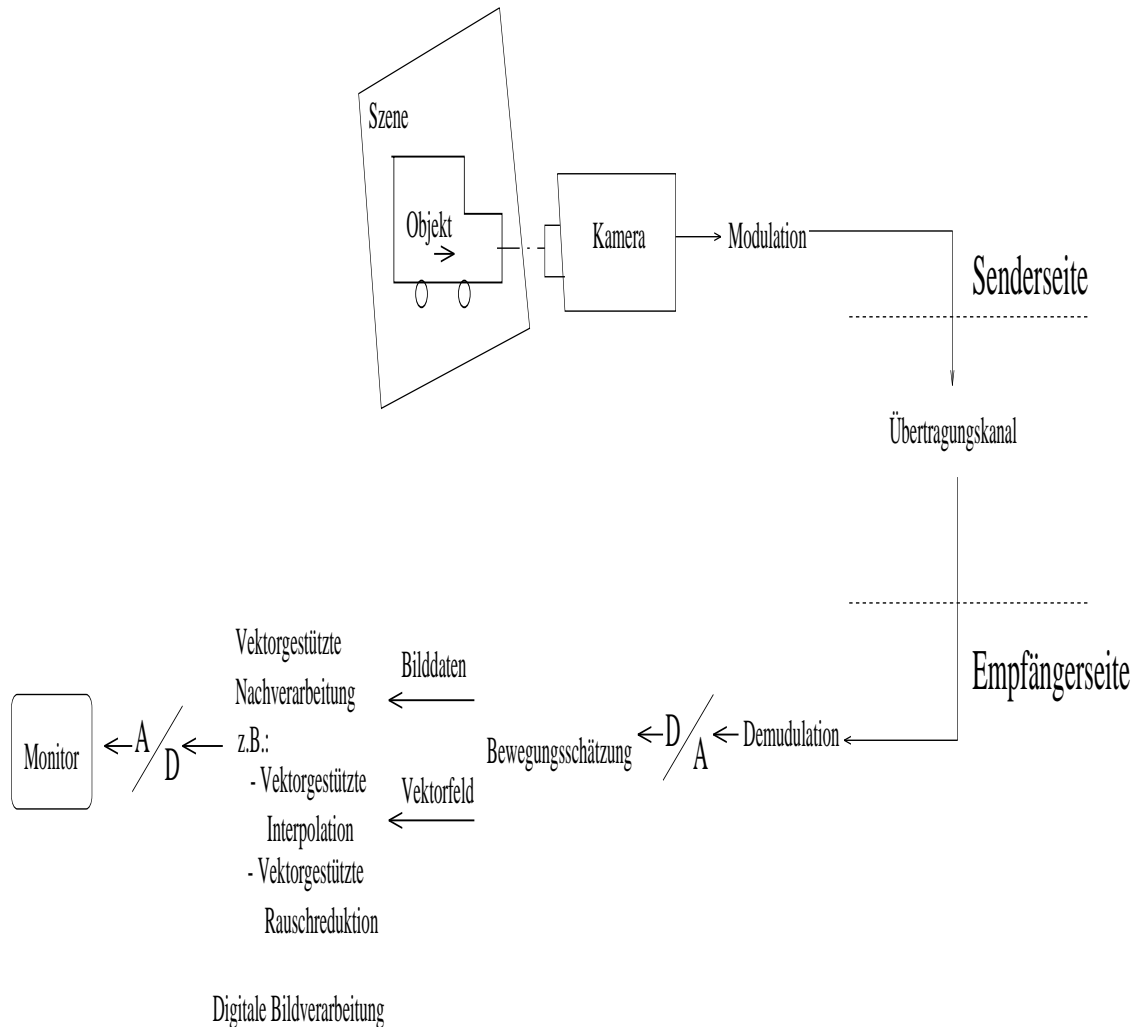


Abbildung 2.1: Blockschaltbild eines konventionellen Fernsehsystems

wegungsadaptive Verfahren die Information eines Bewegungsdetektors, um zwischen verschiedenen statischen Verfahren umzublenden. Wie bereits häufig nachgewiesen<sup>3</sup>, lassen sich die besten Interpolationsergebnisse mit bewegungsvektorgestützten Algorithmen erzielen, die Vektoren zur Interpolation verwenden, die die Bewegungen von Objekten nach Betrag und Richtung beschreiben. Die möglichst genaue Schätzung der Bewegung der Objekte zwischen zwei Halbbildern ist für die bewegungsbasierte Interpolation also ein wichtiges Instrument zur Verbesserung der Bildqualität in der Bildwiedergabeeinheit.

Für die Bewegungsschätzung existieren verschiedene Methoden. Im allgemeinen kann zwischen *bildpunktbasierter* Bewegungsschätzung und *objektbasierter* Bewegungsschätzung unterschieden werden. Bei der bildpunktbasierten Bewegungsschätzung wird

<sup>3</sup>vgl. [Blum94] und [Hent85]

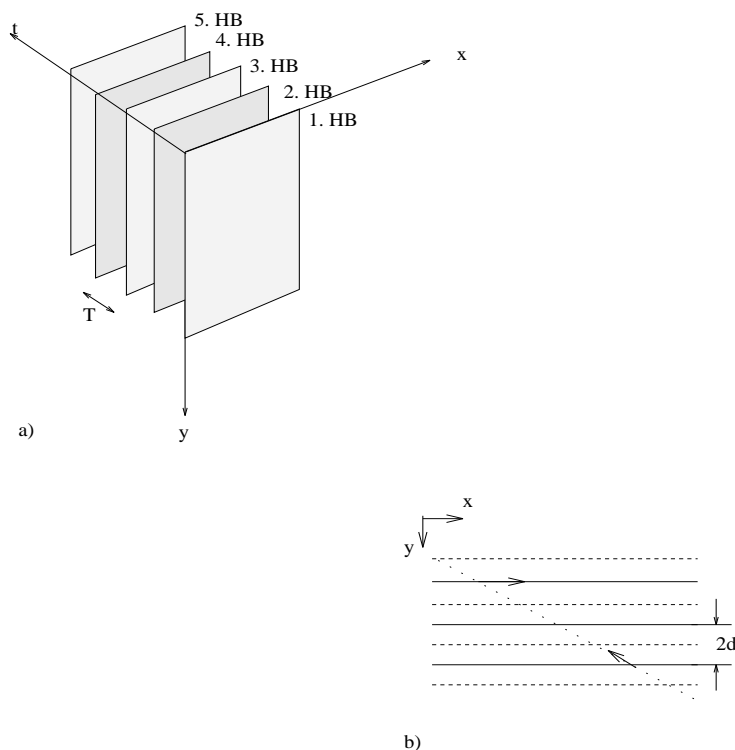


Abbildung 2.2: Zeilensprungabtastung  
 a. Abtastungsschema dreidimensional  
 b. Abtastung in der  $x,y$ -Ebene

jedem Pixel im Bild eine Bewegung zugeordnet. Dabei wird der Bildinhalt nicht berücksichtigt. Als bildpunkt-basierte Methoden finden die Gradientenmethode, Fouriermethode und die Suchmethode die häufigsten Anwendungen. Kapitel 2.2 gibt eine kurze Übersicht über diese Methoden. Objektbasierte Bewegungsschätzung arbeitet bildinhaltsangepaßt. Hier wird für jedes Bildobjekt (wobei der Begriff „Objekt“ nicht unbedingt dem realen Objektbegriff entsprechen muß) ein Bewegungsvektor errechnet. In Kapitel 2.3 werden einige objektbasierte Bewegungsschätzungsverfahren kurz erläutert. Ein im Rahmen dieser Arbeit entwickeltes Konzept zur objektbasierten Bewegungsschätzung wird in Kapitel 5 vorgestellt. Es ist anzumerken, daß diese Methoden sich im Forschungsstadium befinden und daher bisher noch nicht häufig angewendet werden.

## 2.2 Bildpunkt-basierte Bewegungsschätzung

### 2.2.1 Vorbemerkung

Die bildpunkt-basierten Bewegungsschätzer errechnen die Verschiebungsvektoren ohne die im Bild enthaltenen Strukturen zu berücksichtigen. Dies kann zu erheblichen Bewegungsverfälschungen führen. Bei dem häufig angewandten Blockmatching wird das Bild in Blöcke gleicher Größe unterteilt und dann wird die Bewegung blockweise er-

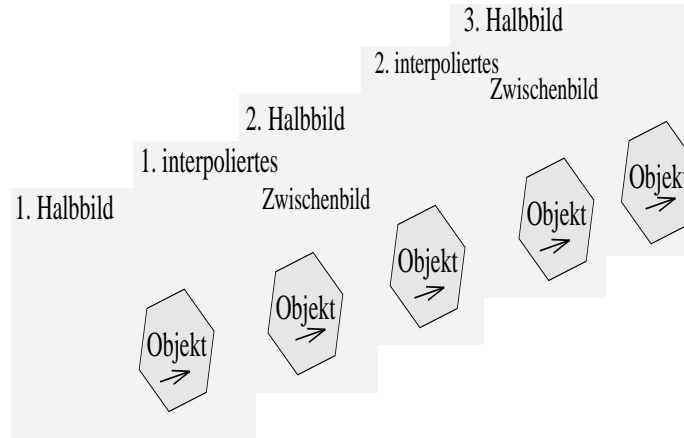


Abbildung 2.3: Interpolation von Zwischenbildern

rechnet. Wenn aber ein Block Teile verschiedener bewegter und nicht bewegter Objekte enthält, dann erhalten homogene Teile bei der Interpolation verschiedene Bewegungen. Außerdem stützt sich die Beschreibung der bildpunkt-basierten Bewegungsschätzung auf ein translatorisches Bewegungsmodell, wobei andere Bewegungsformen aus der Translation approximiert werden. Auf der anderen Seite sind die bildpunkt-basierten Bewegungsschätzer einfach zu implementieren und weisen bei der Hardwarerealisierung geringen Aufwand auf.

### 2.2.2 Gradientenmethode

Mit der Gradientenmethode wird jedem Pixel eine Bewegung zugeordnet. Seien  $s_k$  und  $s_{k-1}$  zwei aufeinanderfolgende Bilder einer Bildsequenz, die unter der Annahme eines vereinfachten Bewegungsmodells durch eine Translation,  $\vec{d}(x, y) = (d_x, d_y)$ , miteinander verknüpft sind, mit  $s_k(x, y)$  als die Luminanz eines Punktes des Bildes  $k$  an der Stelle  $x, y$ :

$$s_{k-1}(x, y) = s_k(x + dx, y + dy) \quad (2.1)$$

$\Leftrightarrow$

$$s_k(x, y) = s_{k-1}(x - dx, y - dy) \quad (2.2)$$

Ausgangspunkt der Betrachtung ist die Gleichung 2.3.  $D_s$  sei die Bilddifferenz zweier zueinandergehöriger Bildpunkte:

$$D_s(x, y) = s_k(x, y) - s_{k-1}(x, y) \quad (2.3)$$

aus Gleichung 2.1 folgt,

$$D_s(x, y) = s_k(x, y) - s_k(x + dx, y + dy) \quad (2.4)$$

Wird der Term  $s_k(x + dx, y + dy)$  in einer Taylorreihe entwickelt<sup>4</sup>, so wird

$$s_k(x + dx, y + dy) = s_k(x, y) + \frac{\partial s_k(x, y)}{\partial x} dx + \frac{\partial s_k(x, y)}{\partial y} dy + R(x, y) \quad (2.5)$$

wobei das Restglied  $R(x, y)$  Elemente höherer Ordnung enthält.

Unter Vernachlässigung des Restgliedes erfolgt aus den Gleichungen 2.4 und 2.5 für die Bilddifferenz  $D_s$ :

$$D_s(x, y) = -\frac{\partial s_k(x, y)}{\partial x} dx - \frac{\partial s_k(x, y)}{\partial y} dy \quad (2.6)$$

mit

$$\begin{pmatrix} dx \\ dy \end{pmatrix} \quad \text{Verschiebungsvektor und}$$

$$\begin{pmatrix} -\frac{\partial s_k(x, y)}{\partial x} dx \\ -\frac{\partial s_k(x, y)}{\partial y} dy \end{pmatrix} \quad \text{Gradientenvektor}$$

Aus Gleichung 2.6 folgt, daß die Schätzung der Bewegung in starkem Verhältnis zum Grauwertgradienten steht (Abbildung 2.4). Gleichung 2.6 gilt nur für kleine Verschie-

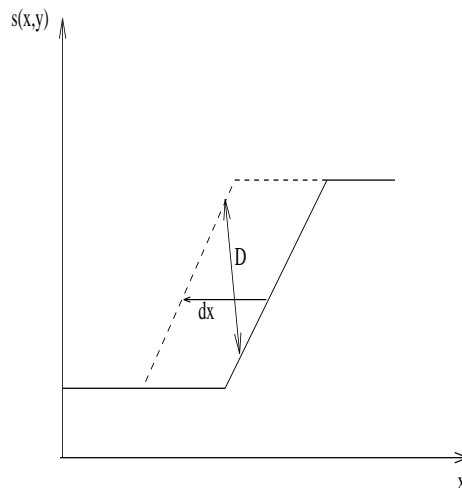


Abbildung 2.4: Gradient einer translatorisch in x-Richtung bewegten Kante

$s_k$ : —,  $s_{k-1}$  : ...

bungen. Bei großen Verschiebungen liefert die Gleichung 2.6 fehlerhafte Vektoren. Abhilfe kann hier eine iterative rekursive Schätzung der Verschiebungsvektoren schaffen<sup>5</sup>. In diesem Fall wird der Verschiebungsvektor durch folgende rekursive Schätzung approximiert:

$$\hat{d}(x, y)|_{i+1} = \hat{d}(x, y)|_i + \Delta d(x, y)|_i \quad (2.7)$$

<sup>4</sup>vgl. [WSch92]

<sup>5</sup>vgl. [WSch92]



Der Verschiebungsvektor besteht nun aus dem Vektor nach Gleichung 2.6 und aus einem Korrekturwert  $\Delta d(x, y)$ <sup>6</sup>.

### 2.2.3 Die Fouriermethode

Die Verschiebung der Bildsignale  $s_{k-1}(x, y)$  und  $s_k(x, y)$  wird mit dieser Methode aus der Phasendifferenz ihrer Fouriertransformierten  $S_{k-1}(u, v)$  und  $S_k(u, v)$  ermittelt. Dabei wird für jeden Block die Fouriertransformation gebildet. Aus der Phasenkorrelationsfunktion, Gleichung 2.8, wird dann auf den Verschiebungsvektor geschlossen.

$$P_k = \frac{S_k^*(u, v) \cdot S_{k-1}(u, v)}{|S_k(u, v) \cdot S_{k-1}(u, v)|} \quad (2.8)$$

$S_k^*(u, v)$  ist die konjugierte komplexe Fouriertransformierte.

Da die Bilder durch eine Translation verbunden sind (Gleichung 2.1 bzw. 2.2), wird die Phasenkorrelation wie folgt hergeleitet:

Die zugehörige Fouriertransformierte von  $s_k(x, y)$  ist:

$$S_k(u, v) = S_{k-1}(u, v) \cdot e^{-j2\pi(udx+vdy)} \quad (2.9)$$

eingesetzt in Gleichung 2.8 ergibt sich für die normierte Phasenkorrelation:

$$P_k(u, v) = \frac{|S_{k-1}(u, v)| e^{j\varphi_{k-1}(u, v)} e^{+j2\pi(udx+vdy)} |S_{k-1}(u, v)| e^{-j\varphi_{k-1}(u, v)}}{|S_{k-1}(u, v)|^2}$$

daraus folgt:

$$P_k(u, v) = e^{+j2\pi(udx+vdy)} \bullet \longrightarrow \delta(x + dx, y + dy) \quad (2.10)$$

$\delta(x + dx, y + dy)$  ist die Fouriertransformierte von  $P_k(u, v)$ , die die Verschiebung eines Blockes darstellt.

Mit der Transformationsmethode ist das Problem der großen Verschiebungen gelöst, da Gleichung 2.10 für beliebig große Verschiebungen gilt. Auf der anderen Seite hat die Phasenkorrelationsmethode wegen der zweimaligen Fouriertransformation und der durchzuführenden Division einen hohen Hardwareaufwand. Dies ist für die *empfängerseitige* Zwischenbildinterpolation<sup>7</sup> ein bedeutender Nachteil.

### 2.2.4 Die Suchmethode (Das Blockmatching)

Das Grundkonzept der Suchmethoden ist der Vergleich eines Blockes aus dem Bild  $k - 1$  mit Blöcken aus dem folgenden Bild  $k$ . Abbildung 2.5 zeigt das Prinzip der Suchmethode: Ein Referenzfenster (Block) aus Bild  $k - 1$  wird über einen vordefinierten Suchbereich im Bild  $k$  geschoben und ein Vergleichswert des Referenzblockes und des Suchblockes wird mittels einer Gütefunktion ermittelt. Je nach Suchstrategie werden einige Suchmethoden unterschieden<sup>8</sup>:

<sup>6</sup>vgl. [WSch92]

<sup>7</sup>vgl. Abbildung 2.1

<sup>8</sup>für mehr Details vgl. [WSch91]

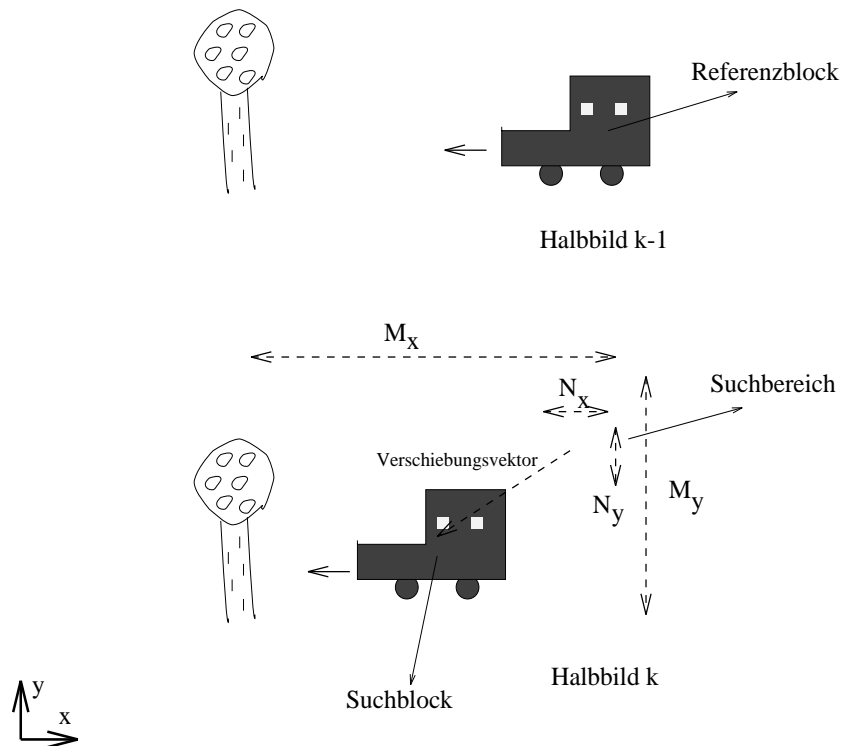


Abbildung 2.5: Das Prinzip der Suchmethode zur Bewegungsschätzung  
 $N_x \times N_y$ : Größe des Referenzblockes  
 $M_x \times M_y$ : Größe des Suchbereiches

- Die vollständige Suche (full-search Matching).
- Dreischrittsuche:
  1. In einem sehr groben Abtastraster wird ein Verschiebungsvektor gesucht.
  2. Um den Verschiebungswert aus Schritt eins wird mit einem feineren Raster gesucht.
  3. Wiederholung von Schritt zwei mit noch feinerem Raster.
- Suche in separierten Richtungen (conjugate direction search):  
 Zunächst wird nur in  $x$ -Richtung nach dem Fehlermaßminimum gesucht z.B. ausgehend vom Suchbereichszentrum. Von dort aus erfolgt die Suche in  $y$ -Richtung, bis das Fehlermaßminimum ebenfalls aufgefunden wird.
- Die prädiktive Suche (prädiktives Matching).

Beim full-search Matching werden die Halbbilder, zwischen denen die Bewegung geschätzt werden soll, in Blöcke gleicher Größe unterteilt. Der Referenzblock aus dem Bild  $k - 1$  wird dann mit *allen* Blöcken des Suchbereiches des  $k$ -ten Bildes verglichen. Dazu wird eine Gütefunktion (z.B. die DFD: Displaced Frame Difference<sup>9</sup>) über die Summe der Beträge der Pixeldifferenz zwischen Referenzblock und Suchblock gebildet. Es wird dann der Kandidatenvektor ausgewählt, dessen Gütewert den geringsten Betrag besitzt. Da die Rechenzeit dieses Verfahrens quadratisch zum Suchbereich steigt, eignet sich dieses Verfahren nicht zur *empfängerseitigen*<sup>10</sup> Schätzung von Verschiebungsvektoren.

Häufig werden aber Suchmethoden eingesetzt, die den Aufwand dadurch reduzieren, daß der Referenzblock nur mit bestimmten Suchblöcken getestet wird. Eine der bekannten Methoden dieser Art ist das prädiktive Blockmatching, welches in Kapitel 2.2.5 behandelt wird.

### 2.2.5 Das Grundkonzept des prädiktiven Blockmatching

Für die *empfängerseitige* Bewegungsschätzung<sup>11</sup> sind die prädiktiven Verfahren wegen ihres geringen Hardwareaufwandes und ihrer einfachen Implementierbarkeit besonders geeignet. Zur Senkung des Rechenaufwandes bei der Suchmethode werden die zeitlichen und die örtlichen Nachbarschaftsblöcke des Referenzblockes betrachtet. Die prädiktiven Verfahren gehen davon aus, daß die Bewegung eines Blockes sich nicht drastisch von einem Halbbild zum nächsten Halbbild ändert. Dies trifft besonders in homogenen Bereichen zu (z.B. Blöcke eines Objektes). Im Suchbereich um den Referenzblock wird eine bestimmte Anordnung von Blöcken, sogenannten *Prädiktionsblöcke* festgelegt, bei denen angenommen wird, die dazugehörigen Vektoren (*Prädiktionsvektoren*) haben mit dem zum Referenzblock gehörigen Bewegungsvektor etwas zu tun. Die Gütefunktion wird dann auf Basis der Prädiktionsvektoren, die den örtlich benachbarten Prädiktionsblöcken entnommen werden, sowie einigen weiteren<sup>12</sup> aus diesen Prädiktionsvektoren abgeleiteten Kandidatenvektoren berechnet. Die Verwendung von Informationen aus anderen Blöcken (Prädiktionsblöcken) zur Bewegungsschätzung eines Blockes verlieh diesen Methoden ihren Namen.

Eine Vielzahl von Algorithmen zum prädiktiven Matching sind entwickelt worden. Diese Algorithmen unterscheiden sich dadurch, wie die Anordnung der Prädiktionsblöcke vorgenommen wird. Ein bekanntes prädiktives Verfahren ist das Verfahren nach de Haan. In den nächsten Abschnitten werden die Grundkonzepte des Verfahrens nach de Haan<sup>13</sup> und eines auf dem Verfahren nach de Haan basierenden parallel-prädiktiven Blockmatching<sup>14</sup> erläutert.

**Grundkonzept des Verfahrens nach de Haan:** Bei dem Verfahren nach de Haan wird *eine Anordnung* von vier *Prädiktionsvektoren* und eine Zahl von zugeordneten

---

<sup>9</sup>vgl. Gleichung 2.11

<sup>10</sup>vgl. Abbildung 2.1

<sup>11</sup>vgl. Abbildung 2.1

<sup>12</sup>vgl. Abbildung 2.6

<sup>13</sup>[Haan91]

<sup>14</sup>[Blum92]

Korrekturwerten zur Bestimmung des *Bewegungsvektors* des *aktuellen Blockes* festgelegt. Zwei der vier Prädiktionsblöcke werden aus dem gleichen (örtlichen) Halbbild und die anderen zwei aus dem vorherigen (zeitlichen) Halbbild ausgewählt (Abbildung 2.6).

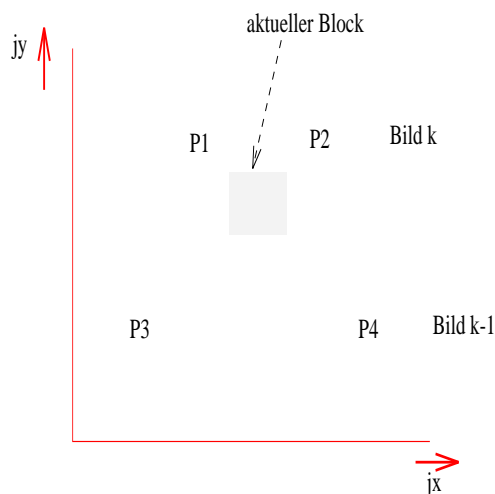


Abbildung 2.6: Anordnung der Prädiktionsblöcke.  $P_i$ :  $i$ -ter Prädiktionsblock

Die vier Prädiktionsvektoren sind nicht ausreichend, um große zeitliche als auch örtliche Bewegungsänderungen zu erfassen. Zusätzlich wird für jeden Prädiktionsvektor eine Zahl von *Korrekturwerten* (*Korrekturstern* genannt) festgelegt. Diese Korrekturwerte dienen der Zulassung von höheren Geschwindigkeiten, als dies die Prädiktionsvektoren erlauben. Aus Rechenzeitüberlegungen werden aber nur bestimmte Geschwindigkeiten erlaubt. Die günstigste<sup>15</sup> Anordnung ist in Abbildung 2.7 dargestellt. Da zur Bestimmung des Bewegungsvektors des Referenzblockes nur bestimmte Bewegungsvektoren (Prädiktionsvektoren und Korrekturstern, Abbildungen 2.6 und 2.7) in Betracht gezogen werden, konvergiert die wahre Geschwindigkeit eines Objektes erst nach Schätzungen von mehreren Objektblöcken. Dieses Konvergenzverhalten ist ein typisches Merkmal parallel-prädiktiver Blockmatchingverfahren.

Eine Gütefunktion wird zwischen dem Referenzblock und jeweils einem der festgelegten Suchblöcke (Prädiktionsblöcke) gebildet. Bei dem Verfahren nach de Haan wird die DFD (Displaced Frame Difference) gewählt. In Gleichung 2.11 zur Berechnung der DFD werden folgende Definitionen verwendet:

$P$ : Menge aller Punkte, die zum Referenzblock gehören,

$\vec{V} = (V_x, V_y)$ : Vektor des Suchblockes, für welchen die DFD ermittelt wird,

$T_i$ : Zeitpunkt der Bewegungsschätzung,

$s(x, y, T_i)$ : Der Pixelgrauwert an der Stelle  $x, y$  zum Zeitpunkt  $T_i$ ,

$N_x \times N_y$ : Die Größe eines Blockes.

<sup>15</sup>[Blum92]

$$DFD(P, \vec{V}, T_i) = \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} |s(x, y, T_i) - s(x - V_x, y - V_y, T_{i-1})| \quad (2.11)$$

Der Kandidatenvektor mit der minimalen DFD wird dann als Bewegungsvektor des Referenzblockes ausgewählt. De Haan setzt für die Anordnung von Abbildung 2.6 und

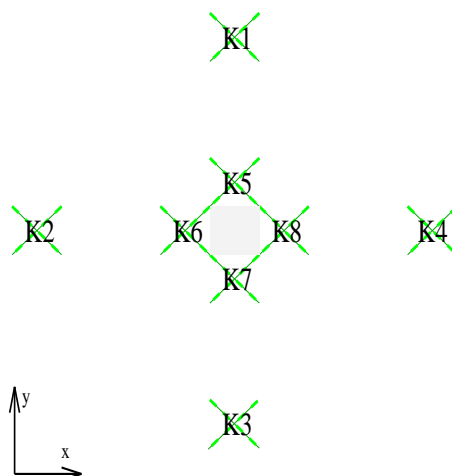


Abbildung 2.7: Anordnung der Korrekturwerte für einen Prädiktionsvektor  
 $K_i$ : der  $i$ -te Korrekturwert

2.7 zwei separate Schätzer ein. Bei jedem Schätzer wird nach Gleichung 2.11 die DFD berechnet. Dann wird aus den beiden Werten der separaten Schätzer der Vektor mit der besseren DFD ausgesucht. Dieser ausgesuchte Vektor wird dann jedem Blockpixel zugeordnet.

**Weiterentwicklung des parallel-prädiktiven Verfahrens nach [Blum92]:** Blume hat das Verfahren nach de Haan weiterentwickelt<sup>16</sup>. Dabei wurde das Grundkonzept dieses Verfahrens nicht verändert. Die wesentlichen Unterschiede sind im folgenden zusammengefaßt:

- Rasterangepaßte Uminterpolation  
 Wie bei der Einleitung zu diesem Kapitel beschrieben, werden Bilder im Zeilensprungformat abgetastet und übertragen, d.h. die Bildobjekte werden in zwei unterschiedlichen Rasterlagen übertragen. Dies führt zu Problemen bei der Bestimmung von Gütefunktionen zwischen zwei Halbbildern. Blume entwickelte ein intrafeld Verfahren zur „rasterangepaßten Uminterpolation“, das den negativen Einfluß des Zeilensprungverfahrens erheblich mindert. Dabei wird die Rasterzeile eines Halbbildes und die nächste Rasterzeile dieses Halbbildes verschieden gewichtet und zu einer Zwischenzeile summiert<sup>17</sup>. Somit wird intrafeld eine neue

<sup>16</sup>[Blum92], [Blum93]

<sup>17</sup>Abbildung 2.8, entnommen aus [Blum92]

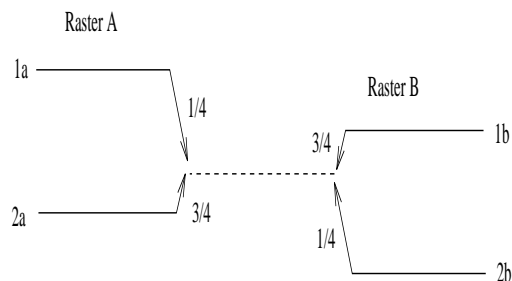


Abbildung 2.8: Rasterangepaßte Uminterpolation

einheitliche Rasterlage für die Bildobjekte erreicht. Diese Maßnahme wird als Vorverarbeitungsschritt vor der Bewegungsschätzung eingesetzt.

- Plausibilitätsregeln

Für die Auswahl zwischen den beiden Schätzern wird nicht nur die DFD betrachtet. Blume bezieht für die Entscheidung zwischen den beiden Schätzern Bewegungsinformationen aus Kontrollblöcken des gleichen Halbbildes ein, die von den örtlichen Prädiktionsvektoren einen Block zurückliegen (Abbildung 2.9).

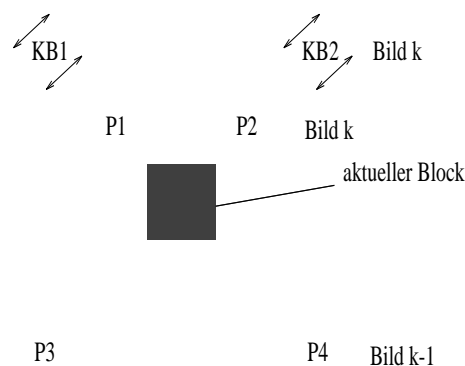


Abbildung 2.9: Anordnung der Kontrollblöcke zur Auswahl zwischen den Schätzern  
 $P_i$ :  $i$ -ter Prädiktionsblock,  $KB_i$ :  $i$ -ter Kontrollblock

Blume stellt folgende „Vektor-Plausibilitätsregeln“ auf, die das Konvergenzverhalten der Blockbewegung zur Schätzerauswahl betrachten:

- Wenn die beiden Schätzer die gleiche DFD liefern, dann wird einer der beiden zugehörigen Kandidatenvektoren genommen.
  - Wenn die Schätzer verschiedene DFD's liefern, dann werden die Vektoren der Kontrollblöcke (Abbildung 2.9) betrachtet. Wenn bei einem Schätzer die Vektoren der Kontrollblöcke gleich sind, dann wird dieser Bereich als auskonvergiert betrachtet.
  - Wenn die ersten beiden Regeln nicht gelten, dann wird der Vektor mit der betragsmäßig kleineren DFD gewählt.
- Nachverarbeitung der Vektorfelder  
 Durch Fehlschätzungen von Verschiebungsvektoren kann die Interpolation von Zwischenbildern stark beeinträchtigt werden. Fehlschätzungen können aufgrund von *Rauschen*, aufgrund des *Konvergenzverhaltens* des Algorithmus oder aufgrund von *Blockstrukturbildung* entstehen. Mittels entsprechender Nachverarbeitung der Vektorfelder können solche Artefakte beseitigt werden. Durch das Konvergenzverhalten des Algorithmus entstehen sogenannte *Konvergenzränder*. Dies sind die Ränder von Vektorbereichen. Auf diesen Vektorfeldberandungen springt der Vektor nicht in einem Schritt zum Hintergrundvektor auf den Objektvektor, sondern benötigt hierfür einige Schritte. Dies liegt daran, daß im Korrekturstern<sup>18</sup> nicht sämtliche mögliche Verschiebungen enthalten sind, sondern nur einige diskrete Werte. So können Verschiebungen auftreten, die einige Korrekturen (Updates) erfordern, bis der Prädiktionsvektor zum richtigen Objektvektor wird. *Blockstrukturbildung* heißt, daß durch die Zuordnung des ermittelten Vektors dem aktuellen Block eine Blockstruktur der Bewegung in dem interpolierten Zwischenbild erkennbar wird. Um dieses Artefakt zu verhindern, wird in der Nachverarbeitungskomponente eine sogenannte *Blockerosion* unternommen. Blockerosion heißt: der aktuelle Block wird in 4 Subblöcke unterteilt und für jeden Subblock wird die Bewegung durch Einbeziehen von Bewegungsvektoren der Nachbarblöcke modifiziert<sup>19</sup>.

Abbildung 2.10<sup>20</sup> gibt eine Gesamtübersicht über das weiterentwickelte Verfahren nach [Blum92].

## 2.3 Objektbasierte Bewegungsschätzung

### 2.3.1 Vorbemerkung

Die objektbasierte Bewegungsschätzung ist ein Forschungsgebiet, in dem objektbasierte Ansätze auf ihre Tauglichkeit beim Einsatz in verschiedenen Anwendungen der Bewegungsschätzung (Interpolation, Bildcodierung, ...) geprüft werden. Das Idealziel

---

<sup>18</sup>Abbildung 2.7

<sup>19</sup>für mehr Details vgl. [Blum92], Kapitel 5

<sup>20</sup>vgl. [Blum92]

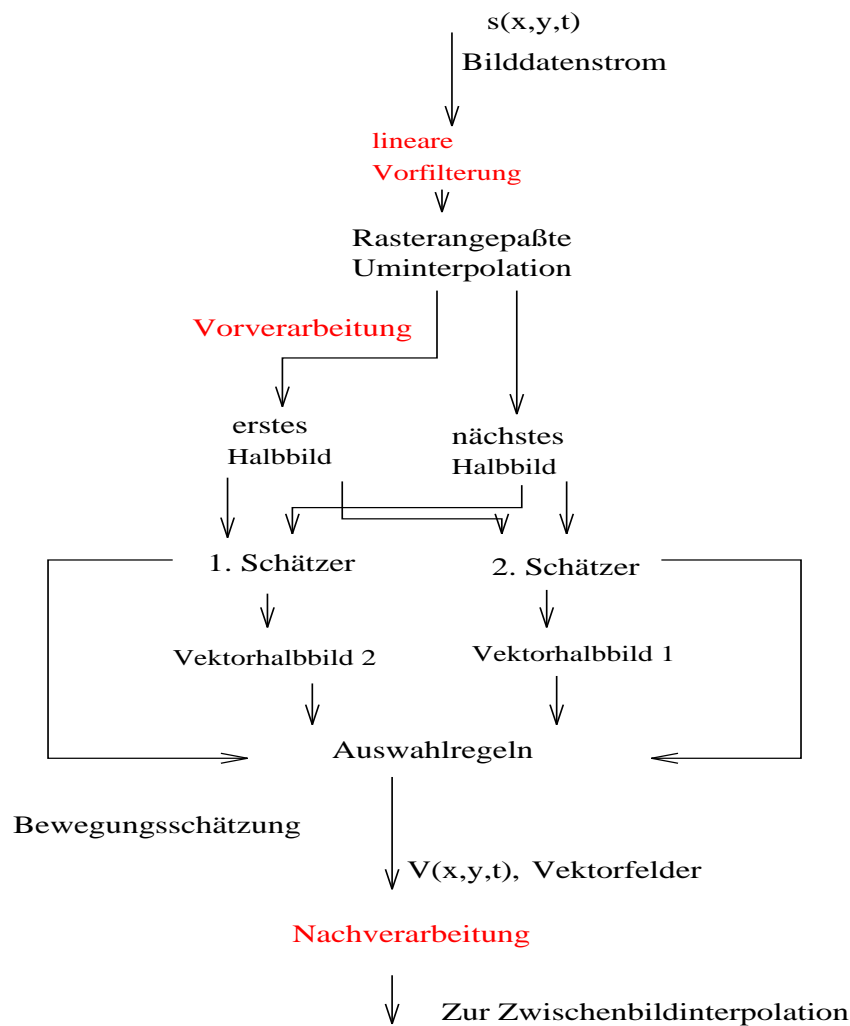


Abbildung 2.10: Flußdiagramm des Blockmatching nach [Blum92]

objektbasierter Bewegungsschätzung ist die Zuordnung von jeweils **einen** Bewegungsvektor zu **jedem** realen Objekt der aufgenommenen Szene. Eine wichtige Voraussetzung für diese Zuordnung und für einen leistungsfähigen objektbasierten Schätzalgorithmus ist die Erkennung von Bildteilen aus dem Bildsignal, die **so gut wie möglich** mit den realen Objekten übereinstimmen<sup>21</sup>. Im allgemeinen kann zwischen objektbasierter Bewegungsschätzung auf Basis erkannter Objektmerkmale (z.B. Objektkonturen) und objektbasierter Bewegungsschätzung auf Basis segmentierter Vektorfelder unterschieden werden. Die objektbasierte Bewegungsschätzung auf Basis segmentierter Vektorfelder kann in drei Schritte gegliedert werden:

1. Bildpunktbasierter Bewegungsschätzung
2. Segmentierung der resultierenden Vektorfelder

---

<sup>21</sup>vgl. Definition 3.5



3. Verbesserung der Bewegungsschätzung auf der Basis segmentierter bewegungshomogener Bildteile

Die objektbasierte Bewegungsschätzung auf der Basis von merkmalgestützter Objekterkennung wird in folgende Schritte unterteilt (vgl. das Blockdiagramm in Abbildung 5.1):

1. Bildpunktbasierte Bewegungsschätzung
2. Erkennung von Bildobjekten allein auf der Basis von Objektmerkmalen
3. **Optimierung** der *Bewegungsinformationen*, die mittels eines Bewegungsschätzers (z.B. prädiktives Blockmatching) ermittelt werden, *anhand* der gewonnenen *Objektinformationen*

Aus dem Literaturstudium zu dieser Arbeit geht hervor, daß verschiedene Ansätze zur objektbasierten Bewegungsschätzung auf der Basis bewegungsvektorgestützter Bildsegmentierung entwickelt worden sind. Im folgenden Kapitel 2.3.2 sind einige Ansätze kurz erläutert. Ein Dilemma und Problem solcher Verfahren ist, daß zur Schätzung der Bewegung eine Bildsegmentierung in disjunkten Bildteilen vorhanden sein muß, andererseits kann ohne Bewegungsinformationen keine Segmentierung vorgenommen werden. Außerdem ist die gegenseitige Abhängigkeit des Segmentierungsvorgangs und der Bewegungsschätzung sehr erheblich und führt bei falschen Ergebnissen in einer der beiden Vorgänge zu erheblichen und irreversiblen Fehlern in dem anderen Vorgang. Ein anderes Problem vektorfeldbasierter Schätzer ist, daß die Segmentierung der aus bildpunktbasierter Bewegungsschätzern resultierenden Vektorfelder, um homogene Bildteile zu erhalten, relativ kleine zusammenhängende Bildbereiche liefert, die mit den im Bild enthaltenen realen Objekten nicht übereinstimmen. Deshalb stellen solche erkannten Bildteile für leistungsfähige **objektbasierte** Schätzalgorithmen eine nicht günstige Basis dar. Ansätze zur objektbasierten Bewegungsschätzung mit Methoden der merkmalgestützten Objekterkennung sind der Autorin dieser Arbeit nicht bekannt. Ein im Rahmen dieser Arbeit entwickeltes Verfahren zur objektbasierten Bewegungsschätzung unter Verwendung von Objektmerkmalen zur Objekterkennung wird in Kapitel 5 vorgestellt und eine Definition des Terminus „objektbasierte Bewegungsschätzung“ wird dort angegeben.

### 2.3.2 Verbesserung der Bewegungsschätzung durch Vektorfeldsegmentierung

**Segment-Matchingmethode:** Guse<sup>22</sup> entwickelte ein Verfahren, das von einer Startunterteilung des Bildes in Initialsegmente ausgeht. Diese Initialsegmente können Blöcke des Bildes sein oder das gesamte Bild kann als Startsegment genommen werden. Mit Hilfe von gewonnenen Bewegungsinformationen werden die Initialsegmente an die realen Objekte der Szene approximiert. Die Segmentierung des Bildes wird wie folgt verbessert:

---

<sup>22</sup>[Guse92]

- Falls ein Segment Teile mehrerer Objekte enthält, wird dieses Segment in mehrere Segmente aufgeteilt.
- Kleine Segmente werden zu einem zusammenhängenden Segment mit Hilfe der ermittelten Bewegungsinformation zusammengefaßt.
- Zu kleine Segmente werden mit benachbarten Segmenten verschmolzen.

Die Bewegungsschätzung erfolgt bei Guse mit Hilfe eines von ihm entwickelten Segment-Matchingverfahrens, das Bewegung für beliebig geformte Segmente ermittelt.

**Änderungsdetektionsmethode:** Die Bewegung wird in diesem Verfahren durch die bildpunktgenauen Verschiebungsvektoren beschrieben. Mit der Gradientenmethode werden die Verschiebungsvektoren zuerst geschätzt. Die Bereiche des Bildes, in denen die Beschreibung der Bewegung nicht exakt war, werden dann mit einem Änderungsdetektor<sup>23</sup> ermittelt. Diese detektierten Bereiche werden als Objekte gefaßt, für die eine neue Iteration der Bewegungsschätzung durchgeführt wird. Dieser Ansatz scheint wegen seiner Änderungsdetektion sehr aufwendig zu sein.

**Kantenregressionsmethode:** Durch Zusammenfassung und Klassifizierung von benachbarten Blöcken eines Bildes, die starken Unterschiede zu den entsprechenden Blöcken im vorherigen Bild aufweisen, zu einem bewegten „Objekt“ wird die Bewegung in dem Ansatz von Kummerfeldt, May und Wolf<sup>24</sup> für Grauwertkanten durchgeführt. Die „Objekt“-Bewegung (genauer Blockbewegung) wird durch eine Regressionsanalyse der ermittelten Kantenbewegungen des „Objektes“ errechnet.

**Bewegungsgestütztes Clustering:** Das bewegungsgestützte Clustering<sup>25</sup> von gleichförmig bewegten Bildpunkten zu Objekten basiert auf der Bewegungsinformationen, die aufgrund der Gradientenmethode bildpunktgenau ermittelt worden sind. Dabei wird die Rauschempfindlichkeit nicht in Betracht gezogen.

---

<sup>23</sup>[Hött90]

<sup>24</sup>[KMW85]

<sup>25</sup>vgl. [Guse92] Seite 7

# Kapitel 3

## Prinzipielle Methoden der Objekterkennung in Bildsequenzen

### 3.1 Grundlagen und Vorbemerkung

Die menschliche Fähigkeit, in verschiedenen Situationen des Alltags Objekte zu erkennen und Probleme zu lösen wird zur Lösung von verschiedenen Fragestellungen in technischen, medizinischen, industriellen und militärischen Anwendungen mit Hilfe der Methoden der Objekterkennung nachgebildet. Die Objekterkennung ist ein traditionsreiches Gebiet der Informatik, das seinen Ursprung in den 60er Jahren hat. Heute hat sich in der Fachsprache der Begriff *Mustererkennung* anstatt des Begriffs *Objekterkennung* durchgesetzt. Leider gibt es derzeit keine einheitliche Terminologie und Bezeichnungsweise in diesem Gebiet. Die Begriffe *Mustererkennung* und *Objekterkennung* sowie andere Grundbegriffe des Gebiets haben in der umfangreichen Literatur zu diesem Thema je nach Problemkreis andere Definitionen. Eine umfassende Definition von Bezdek für den Terminus *Mustererkennung*, die sich nicht nur auf einen spezifischen Problemkreis begrenzt, lautet: „pattern recognition is a search for structure in data“<sup>1</sup>. In dieser Arbeit wird für den Begriff *Mustererkennung* bzw. *Objekterkennung im weiteren Sinne (i.w.S.)* folgende Definition verwendet:

**Definition 3.1 (Mustererkennung bzw. Objekterkennung i.w.S.)**

*Mustererkennung bzw. Objekterkennung i.w.S. ist das Herausfinden von Informationen über Strukturen in Daten.*<sup>2</sup>

□

Für konkrete Fragestellungen der Objekterkennung werden bestimmte Strukturen in den vorhandenen Daten betrachtet. Diese Strukturen werden hier als *Objekte im weiteren Sinne (i.w.S.)* bezeichnet.

---

<sup>1</sup>vgl. [Bez81] S. 1

<sup>2</sup>vgl. [Bez81] S. 1

**Definition 3.2 (Objekt)**

*Ein Objekt i.w.S. ist eine diskrete Abbildung einer physikalischen Größe.*<sup>3</sup>

□

Zur Darstellung von Objekten werden Informationen (Merkmale) über diese Objekte extrahiert.

**Definition 3.3 (Merkmal)**

*Ein Merkmal ist eine charakteristische Eigenschaft eines Objektes.*<sup>4</sup>

□

Durch das Herausfinden von Merkmalen über die Strukturen in den vorhandenen Daten werden Klassen von Objekten gebildet.

**Definition 3.4 (Klasse)**

*Eine Klasse ist eine Menge von Objekten, die gemeinsame Merkmale aufweisen.*

□

Die Daten, die einem mustererkennenden bzw. objekterkennenden System zugrunde liegen, werden in bildhafte und nicht bildhafte Daten (z.B. akustische Daten oder Sinn-daten (wie z.B. Geruch)) unterteilt. In der digitalen Bildverarbeitung und in der Signalverarbeitung wird u.a. mit Verfahren der Mustererkennung bzw. Objekterkennung versucht, aus bildhaften Daten Strukturen herauszufinden und für die anwendungsspezifischen Gebiete (wie z.B. die medizinische Tomographie, die Kriminologie, die Fern-sehetechnik, die Wettervorhersage und die Astronomie) zufriedenstellende Ergebnisse zu erzielen.

Eine reale Szene, die in bildhafte, digitalisierte und rechnerkompatible Daten transformiert wird, besteht aus mehreren Objekten und aus einem Hintergrund (Abbildung 3.1). Leistungsfähige Objekterkennungsalgorithmen sind von besonderer Bedeutung, um Bildvorlagen in disjunkte Bereiche zu trennen, die den Objekten der realen Szene zumindest näherungsweise entsprechen. Merkmale von Szenenobjekten (Bildobjekten), wie Form, Umfang (bzw. Kontur), Farbe (bzw. Grauwert), Oberflächenstruktur usw., sind hilfreiche und unabdingbare Mittel, um Bildobjekte vom Hintergrund zu trennen und zu beschreiben. Die topologischen Merkmale, wie Fläche, Umfang usw., sind wichtige Anhaltspunkte für die Erkennung von Bildobjekten. Objekterkennung in Bildsequenzen zerlegt ein Bild in zwei disjunkte Mengen, die Menge der Punkte, die zu einem Objekt gehören, im weiteren Objektpunkte genannt, und die Menge der Punkte, die zu keinem Objekt gehören, im folgenden Nichtobjektpunkte genannt.

In der Literatur werden im wesentlichen zwei verschiedene Verfahren zur Erkennung von Objekten in Bildsequenzen vorgeschlagen: die *konturorientierten* und die *regionenorientierten* Verfahren. In beiden Verfahren wird die Luminanz (bzw. der Grauwert) der

---

<sup>3</sup>vgl. [Bock87], S. 23 und [Niem89], S. 3

<sup>4</sup>vgl. [Bock87], S. 25

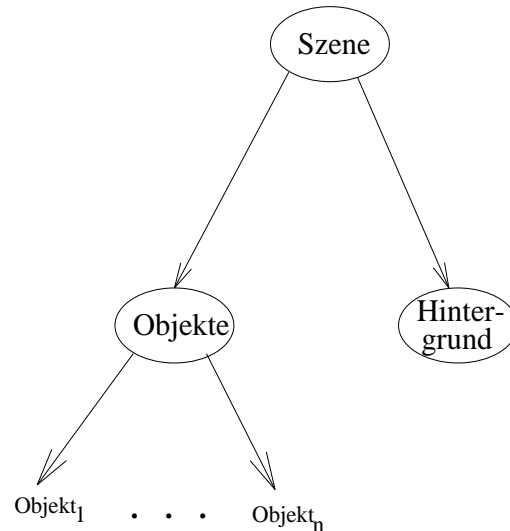


Abbildung 3.1: Bestandteile einer Szene

Bildpunkte zur Gliederung und zur anschließenden Erkennung von homogenen Bildteilen betrachtet (*luminanzbasierte Erkennung*). Eine andere Möglichkeit für die Erkennung von homogenen Bildteilen ist die Segmentierung der aus einem Bewegungsschätzverfahren resultierenden Vektorfelder, so daß Bildpunkte mit gleichen Bewegungsvektoren zu einem zusammenhängenden Bildteil zusammengefaßt werden (*bewegungsbasierte Erkennung*). Die bewegungsbasierte Erkennung hängt aber sehr stark von der Leistungsfähigkeit des Bewegungsschätzverfahrens ab. Bei Betrachtung der resultierenden Vektorfelder heutiger Bewegungsschätzer lassen sich sehr kleine zusammenhängende Bildteile segmentieren. Diese kleinen Bildteile entsprechen im sehr geringen Maß den im Bild enthaltenen eigentlichen Objekten, so daß ihr Einsatz als Grundlage für eine leistungsfähige **objektbasierte** Bewegungsschätzung ungünstig scheint<sup>5</sup>. In den Kapiteln 3.2, 3.3 und 3.4 werden die Grundprinzipien der luminanzbasierten Verfahren erläutert, wobei der Schwerpunkt auf *konturorientierte* Verfahren gelegt wird. In Kapitel 3.5 wird auf die Frage „konturorientierte oder regionenorientierte Bildinterpretation“ eingegangen und es werden sowohl Vor- als auch Nachteile solcher Methoden beschrieben.

---

<sup>5</sup>vgl. Kapitel 2.3.1, Definition 3.5 und Kapitel 4

## 3.2 Konturorientierte Objekterkennung in Bildsequenzen

### 3.2.1 Grundlegende Definitionen

In der Bildverarbeitung spielen Objektkonturextraktionsverfahren zur Trennung von Bildbereichen eine entscheidende Rolle. Die Kenntnis dieser Konturen erlaubt Aussagen über Form und Größe von Bildbereichen sowie die Bestimmung statistischer Bildeigenschaften innerhalb dieser Bereiche. Darüber hinaus können Beziehungen verschiedenster Art zwischen den Bereichen hergestellt und damit weitere Aussagen über das Gesamtbild gemacht werden. Experimente haben gezeigt, daß die menschliche Bildwahrnehmung sich hauptsächlich auf Konturen von Objekten richtet<sup>6</sup>. Für die in Kapitel 4 erläuterten Arbeitsweisen des im Rahmen dieser Arbeit vorgeschlagenen objekterkennenden Konzeptes werden folgende Definitionen gegeben, wobei die Begriffe *Objekterkennung im engeren Sinne (i.e.S)* und *Objekt im engeren Sinne (i.e.S)* im Vergleich zu den Definitionen 3.1 und 3.2 unterschieden werden.

#### Definition 3.5 (Objekterkennung i.e.S.)

**Objekterkennung i.e.S.** ist die Extraktion von homogenen Strukturen (Objekten) aus Bilddaten, so daß die Grenzen dieser Strukturen so gut wie möglich mit den realen Objektgrenzen übereinstimmen.

□

#### Definition 3.6 (Objekt i.e.S.)

**Objekt i.e.S.** ist eine Bildstruktur, die durch eine einfach geschlossene Kontur begrenzt ist.

□

Eine umfassende Definition des Begriffs *Kontur* ist schwierig. Die meisten Konturextraktionsverfahren basieren auf einer punktwisen Betrachtung des Bildes, wobei die unterschiedlichen Helligkeiten auf beiden Seiten des Bildpunktes als Hilfsmittel zur Erkennung nicht kontinuierlicher Bildübergänge (Konturen bzw. Kanten) verwendet werden. Hier wird der Begriff *Kontur* wie folgt definiert:

#### Definition 3.7 (Kontur)

Eine **Kontur** ist eine endlich geordnete Folge von benachbarten Bildpunkten (Konturpunkten):  $p_1 p_2 p_3 \dots p_{n-1} p_n p_1$ .

□

#### Definition 3.8 (Linie)

Eine **Linie** ist ein schmaler Streifen mit unterschiedlichen Helligkeiten auf beiden Seiten.<sup>7</sup>

□

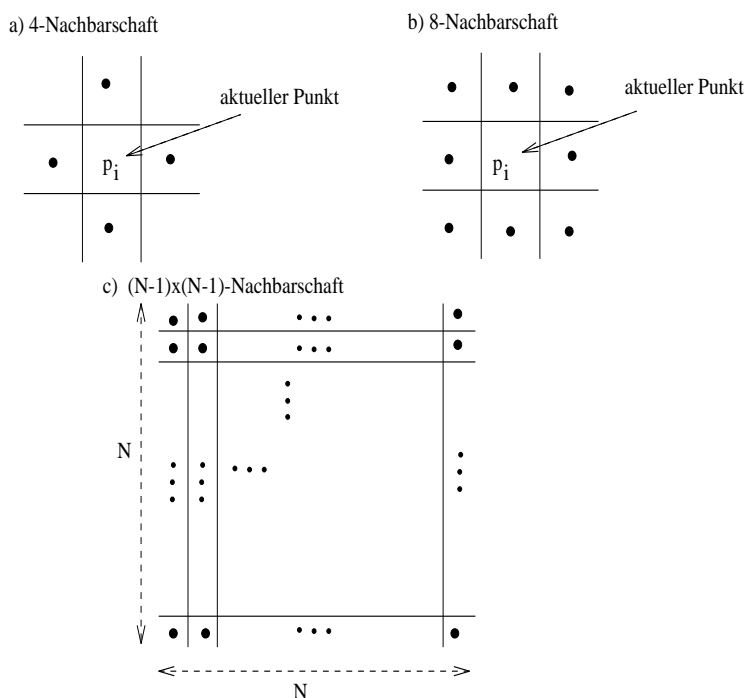


Abbildung 3.2: Darstellung von Nachbarschaftsrelationen:

- Darstellung der Vierernachbarschaft
- Darstellung der Achternachbarschaft
- Darstellung der  $(N-1) \times (N-1)$ -Nachbarschaft, mit einem  $(N \times N)$  großen Fenster

Ein Bildpunkt ist durch seine Ortskoordinaten  $(x, y)$  in der Bildebene definiert. Zwischen Bildpunkten besteht eine Nachbarschaftsrelation. Für die Darstellung dieser Relation existieren mehrere Möglichkeiten. Einige Formen sind in Abbildung 3.2 gezeigt.

### Definition 3.9 (Konturpunkt)

Ein **Konturpunkt**  $p_i$  ist ein Abtastwert  $s(x_i, y_i)$ , der die Bildhelligkeit an den diskreten Ortskoordinaten  $x_i, y_i$  repräsentiert und in seiner Vierernachbarschaft mindestens einen Hintergrundpunkt besitzt.

□

### 3.2.2 Objektkonturextraktion

Das Grundprinzip der meisten Konturextraktionsverfahren kann wie folgt formuliert werden: zunächst werden Konturpunktkandidaten mittels eines *Konturpunktdetektors*

<sup>6</sup>[Geue83], [Haus94]

<sup>7</sup>vgl. [Proj93]

aus der Menge *aller* Bildpunkte ausgesucht und anschließend durch einen *Konturpunktverfolger* zu einer endlich geordneten Folge von Nachbarpunkten  $p_1 p_2 p_3 \dots p_{n-1} p_n$  *verkettet*. Diese Folge repräsentiert eine Kontur, die möglichst geschlossen sein soll, d.h.  $p_1 = p_n$ .

### Konturpunktdetektion

In der Literatur werden zahlreiche Konturpunktdetektoren (auch Kantendetektoren genannt) vorgeschlagen. Bekannte Konturpunktdetektoren können im wesentlichen in zwei Gruppen eingeteilt werden: auf der einen Seite die Gruppe der klassischen (lokalen) Detektoren und auf der anderen Seite die Gruppe der morphologischen Detektoren<sup>8</sup>. Zur Beurteilung von Konturpunktdetektoren sind zwei Kriterien von besonderer Bedeutung:

- Grad der Rauschempfindlichkeit eines Detektors
- Grad der Detektionsgenauigkeit, d.h. welche Breite die detektierten Linien<sup>9</sup> haben

**Klassische Konturpunktdetektoren:** Klassische Detektoren bauen auf *abstrakten*, mathematischen Beschreibungen des Bildmaterials auf. Hier werden Eigenschaften der Umgebung des betrachteten Punktes analysiert, um diesen Punkt als Konturpunkt oder Nicht-Konturpunkt zu klassifizieren. Klassische Verfahren werden grob in globale und lokale Operatoren unterteilt. Globale Operatoren betrachten für die Detektion von Konturpunkten eine größere Umgebung des zu klassifizierenden Punktes. Dies führt dazu, daß die globalen Operatoren viel aufwendiger als die lokalen Operatoren sind. Daher werden in der Praxis, besonders in Echtzeitanwendungen, die lokalen Operatoren bevorzugt. Bei bestimmter Beleuchtung und Aufnahmeansicht entsprechen die Konturen (bzw. Kanten) der physikalischen Körper den Intensitätskonturen (bzw. Intensitätskanten) der Bilder. Daher haben die klassischen Konturpunktdetektionsverfahren in der digitalen Bild(-signal)verarbeitung eine wichtige Rolle. Die weitaus am häufigsten angewandten lokalen Verfahren basieren auf der mathematischen Formulierung für die Amplitudenänderung des Bildsignals. Bekannte Vertreter dieser Gruppe sind die Gradientenoperatoren und der Laplace-Operator.

**Gradientenoperatoren:** Der Gradient einer zweidimensionalen Intensitätsfunktion (eines Bildsignals) eignet sich zur mathematischen Darstellung von Intensitätsänderungen in einem Grauwertbild. Sei  $s(x, y)$  die Funktion des Grauwertbildsignals. Für den Gradienten folgt<sup>10</sup>:

$$\vec{G}(s(x, y)) = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{\partial s(x, y)}{\partial x} \\ \frac{\partial s(x, y)}{\partial y} \end{pmatrix} \quad (3.1)$$

---

<sup>8</sup>vgl. Kapitel 3.3

<sup>9</sup>vgl. Definition 3.8

<sup>10</sup>vgl. [GoWo92], S 418-419



Der Betrag des Gradienten ist:

$$|\vec{G}(s(x, y))| = \sqrt{\left(\frac{\partial s(x, y)}{\partial x}\right)^2 + \left(\frac{\partial s(x, y)}{\partial y}\right)^2} \quad (3.2)$$

Für die Richtung gilt:

$$\Theta(\vec{G}(s(x, y))) = \arctan\left(\frac{\partial s(x, y)}{\partial x} / \frac{\partial s(x, y)}{\partial y}\right) \quad (3.3)$$

Der Vektor  $\vec{G}$  zeigt in die Richtung der maximalen Änderung des Bildsignals ( $s$ ) an der Stelle  $(x, y)$ , wo der Gradientenbetrag  $G$  durch Gleichung 3.2 gegeben ist. Die Detektion von Änderungen des Bildsignals erfordert in jedem Bildpunkt die Differenzierung in zwei orthogonale Richtungen der Ortskoordinaten<sup>11</sup>. Aus Aufwandsgründen wird eine Approximation des Gradienten wie folgt vorgenommen:

$$|\vec{G}(s(x, y))| \approx \left|\frac{\partial s(x, y)}{\partial x}\right| + \left|\frac{\partial s(x, y)}{\partial y}\right| \quad (3.4)$$

Die Gleichungen 3.1 und 3.2 gelten für den kontinuierlichen Fall. In dem diskreten Fall werden die Differentialquotienten durch endliche Differenzen ersetzt. Bei einem Summengradienten z.B. ergibt sich folgende Gleichung<sup>12</sup>:

$$|\vec{G}(s(x, y))| \approx |s(x, y) - s(x + 1, y)| + |s(x, y) - s(x, y + 1)| \quad (3.5)$$

Für die Approximation des Gradienten aus Gleichung 3.2 existieren in der Literatur verschiedene Operatoren. Als Beispiele seien hier der Robertsoperator (vgl. Gleichung 3.6 und Abbildung 3.6) und der Sobeloperator (vgl. Gleichung 3.7 und Abbildung 3.7) genannt.

$$|\vec{G}(s(x, y))| \approx \max(|s(x, y) - s(x - 1, y - 1)|, |s(x, y - 1) - s(x - 1, y)|) \quad (3.6)$$

$$|\vec{G}(s(x, y))| \approx |M_x| + |M_y| \quad (3.7)$$

$M_x$  und  $M_y$  sind dabei durch die Masken in Abbildung 3.4 b) definiert sind. Abbildung 3.3 veranschaulicht die Grundidee der Gradientenoperatoren. Für ein Bildsignal  $s$  wird durch die Berechnung der Gradienten die Intensitätsänderung in jedem Punkt bestimmt. Für die Realisierung der approximativen partiellen Ableitung nach  $x$  und  $y$  werden zwei Masken benötigt. Das Bildsignal wird mit diesen Masken gefaltet. Bei einer Maske von  $n \times n$  ergibt sich für die Faltung folgender Ausdruck:

$$|\vec{G}(s(x, y))| \approx \sum_{i=1}^{n \times n} (s(x, y) \times h_i), \quad (3.8)$$

mit  $h_i$  als Koeffizient  $i$  der Faltungsmaske. Gradientenoperatoren unterscheiden sich dadurch, welche Masken sie für die Faltung festlegen. Typische Masken sind in Abbildung 3.4 gezeigt.

<sup>11</sup>vgl. [Zamp89], S. 56-57

<sup>12</sup>vgl. [Proj93]

Das Bildsignal  $s(x,y)$

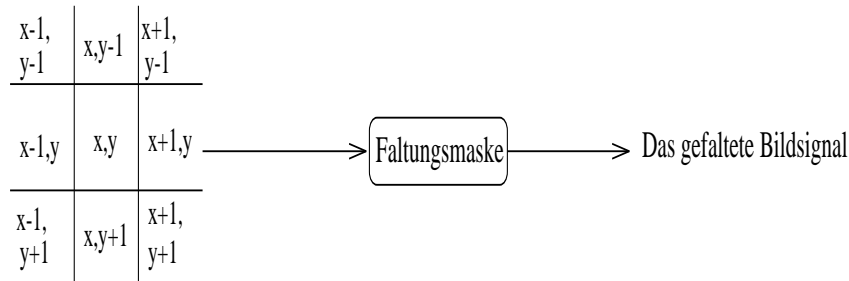


Abbildung 3.3: Lokale Faltung mittels eines Gradientenoperators

**Laplace-Operator:** Bei dem Laplace-Operator wird die zweite Ableitung betrachtet. Die Intensitätsänderungen werden nach Gleichung 3.9 berechnet:

$$\vec{G}^2(s(x, y)) = \begin{pmatrix} G_x^2 \\ G_y^2 \end{pmatrix} = \begin{pmatrix} \frac{\partial^2 s(x, y)}{\partial x^2} \\ \frac{\partial^2 s(x, y)}{\partial y^2} \end{pmatrix} \quad (3.9)$$

Für den diskreten Fall sind die Laplace-Maskenrealisierungen wie in Abbildung 3.4 a) festgelegt, so daß sich für die Intensitätsänderung in jedem Punkt an der Stelle  $x, y$  folgendes ergibt<sup>13</sup>:

$$\left| G^2(s(x, y)) \right| \approx 4s(x, y) - (s(x, y - 1) + s(x - 1, y) + s(x + 1, y) + s(x, y + 1)) \quad (3.10)$$

Dieser Operator ist durch die Berechnung der zweiten Ableitung zur Bestimmung der Intensitätsänderung sehr rauschempfindlich (vgl. Abbildung 3.8), deshalb findet er in der Praxis nur in Kombination mit anderen Operatoren Verwendung.

Die folgenden Abbildungen 3.5–3.9 stellen die Ergebnisse verschiedener Konturpunktdetektoren dar. Die darauf folgende Abbildung 3.10 zeigt ein Ergebnis der Konturpunktlokalisierung des im Rahmen dieser Arbeit entwickelten morphologischen Verfahrens zur Konturpunktdetektion, das in Kapitel 4 ausführlich beschrieben wird. Eine

<sup>13</sup>vgl. [GoWo92], S 420

a) Laplace-operator

|   |    |   |
|---|----|---|
| 0 | 1  | 0 |
| 1 | -4 | 1 |
| 0 | 1  | 0 |

|   |    |   |
|---|----|---|
| 1 | 1  | 1 |
| 1 | -8 | 1 |
| 1 | 1  | 1 |

b) Sobel-operator

|    |   |   |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

|    |    |    |
|----|----|----|
| 1  | 2  | 1  |
| 0  | 0  | 0  |
| -1 | -2 | -1 |

Abbildung 3.4: Typische Masken einiger Konturpunktdetektoren

umfassende Einführung in verschiedene Konturpunktdetektionsverfahren geben Bässman und Besslich<sup>14</sup>.



Abbildung 3.5: Vergleich von Konturpunktdetektoren: Originalbild

---

<sup>14</sup>[BaBe89]



Abbildung 3.6: Konturpunktdetektion des Robertsoperator (invertiert):

- a) Hohe Rauschempfindlichkeit
- b) Schlechte Detektionsgenauigkeit

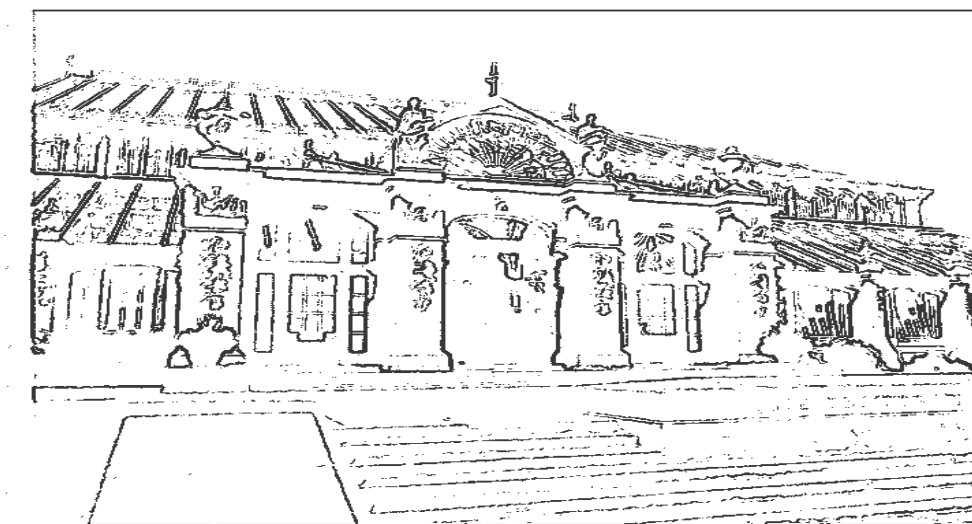


Abbildung 3.7: Konturpunktdetektion des Sobeloperator (invertiert):

- a) Sehr rauschunempfindlich
- b) Schlechte Detektionsgenauigkeit



Abbildung 3.8: Konturpunktdetektion des Laplaceoperators (invertiert):

- a) Sehr hohe Rauschempfindlichkeit
- b) Detektionsgenauigkeit zwischen 1-2 Pixels

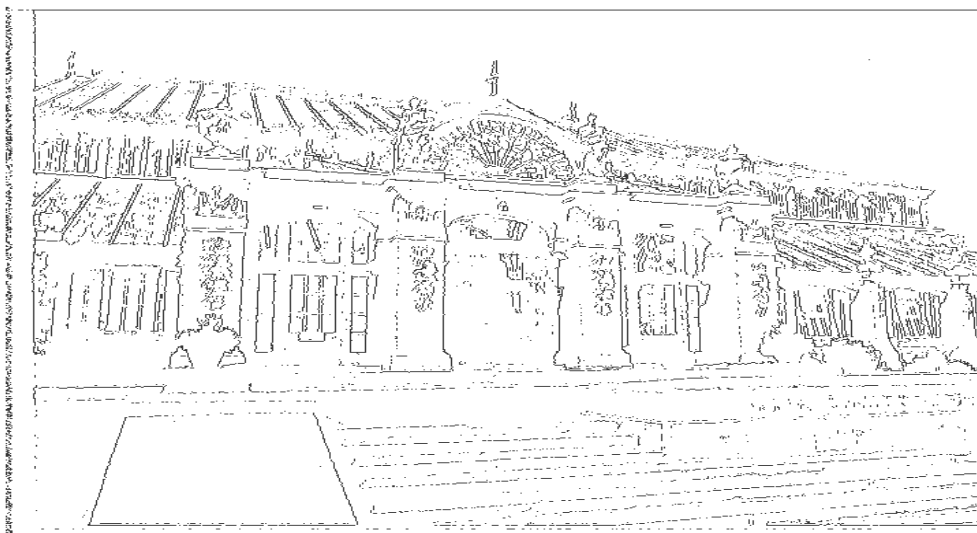


Abbildung 3.9: Konturpunktdetektion des Operators nach [Wu1994] (invertiert):

- a) Sehr gutes Rauschverhalten
- b) Genaue Konturpunktolokalisation



Abbildung 3.10: Ergebnis des im Rahmen dieser Arbeit entwickelten Operators (invertiert):  
 a) Ein gutes Rauschverhalten  
 b) Genaue Konturpunktlokalisierung  
 (meistens nur ein Pixel breite Konturen)

### Konturpunktverkettung

Die Techniken, die in den vorigen Abschnitten zur Konturpunktdetektion beschrieben wurden, sollten im Idealfall Konturpunkte liefern, die zwischen den Bildobjekten liegen. In der Praxis aber werden Konturbilder erzeugt, die von Rauschen, Fehldetektionen (Lücken) und anderen unerwünschten Nebeneffekten begleitet sind. Für weitere Verarbeitungsschritte im Prozeß der Objekterkennung müssen die ermittelten rohen Daten (die Konturpunkte) durch eine geeignete Struktur dargestellt werden. Für diesen Zweck werden die Konturpunkte in einer geordneten Folge miteinander verkettet, wobei der Anfangspunkt und der Endpunkt identisch sind. Ausgehend von einem der ermittelten Konturpunkte wird sukzessiv nach einem Fortsetzungspunkt gesucht. Für den Suchvorgang werden zwei Verfahren unterschieden. *Lokale Verfahren* suchen in einer begrenzten Nachbarschaft um den aktuellen Punkt nach dem Nachfolgepunkt. Sie sind deshalb wenig rechenintensiv, haben aber in komplexen Bildvorlagen Probleme bei der Bestimmung des geeigneten Nachfolgepunktes. *Globale Verfahren*, wie die Hough-Transformation, die heuristische Suche sowie die dynamische Programmierung, beziehen für die Bestimmung des Nachfolgepunktes Informationen und Vorwissen aus größeren Umgebungen des aktuellen Punktes und allgemeine Informationen über das gesamte Bild mit ein. Diese Eigenschaft macht die globalen Verfahren gegenüber Störungen im Bild robust. Ihr entscheidender Nachteil ist aber der erhebliche Rechenaufwand. Deshalb ist der Einsatz solcher Verfahren besonders für Echtzeitanwendungen kritisch. In [BaBe89] findet eine recht gute Einführung in die Arbeitsweise globaler Suchalgorithmen statt. Auf die Beschreibung solcher Algorithmen wird hier aus Platzgründen verzichtet.

Im Rahmen dieser Arbeit wurde ein Verfahren<sup>15</sup> entwickelt, das globale und lokale Informationen nutzt, um eine optimale Konturextraktion zu erreichen. Dies wird durch eine geeignete Reduktion der Menge der Konturpunkt-kandidaten, besonders aber durch die Erzeugung von nur einem Pixel breiten Linien von Konturpunkten erreicht.

## 3.3 Mathematische Morphologie

### 3.3.1 Vorbemerkung

Morphologie ist ursprünglich ein Teilgebiet der Biologie, der Geographie sowie der Linguistik, bei dem Formen und Strukturen untersucht werden. Die mathematische Morphologie in der Bildanalyse hat ihre Wurzeln in den 60er Jahren, als Georges Matheron die geometrischen Strukturen im Bild für die Bildanalyse betrachtet hat. Mittlerweile ist die mathematische Morphologie ein klassischer Bereich der digitalen Bild(-signal)verarbeitung geworden. Die Grundidee morphologischer Verfahren besteht darin, den Bildinhalt mittels einer geometrischen matrixförmigen Struktur, die über das gesamte Bild geschoben wird, so zu manipulieren, daß bestimmte erwünschte Merkmale von Bildstrukturen extrahiert werden. Diese matrixförmige Struktur wird *Strukturelement* genannt. Morphologische Verfahren können zur Lösung verschiedener Aufgaben in der Bildverarbeitung eingesetzt werden. Als Beispiele seien die Konturpunktdetektion (bzw. Kantendetektion), die Bildsegmentierung und die Filterung zur Extraktion von Signalcharakteristik genannt. Eine der wichtigsten Eigenschaften morphologischer Operatoren ist die Generierung von Merkmalen über die im Bild enthaltenen Strukturen bzw. Objekte. Diese Eigenschaft wird besonders im Gebiet der Mustererkennung zur Lösung von komplexen Problemen genutzt.

Im folgenden werden morphologische Grundoperatoren für Zweipegelebilder erläutert. In diesem Fall bestimmen die schwarz/weiß (0/1)-Übergänge die Form (und die Grenze) der Bildobjekte. Die Erweiterung auf Grauwertbilder wird hier nicht behandelt, es sei auf die entsprechende Literatur verwiesen.<sup>16</sup>

### 3.3.2 Grundbegriffe

Die *mathematische Morphologie* baut auf die Grundbegriffe der scharfen Mengenlehre auf. *Mengen* in der mathematischen Morphologie repräsentieren Formen von Objekten eines Bildes. Jedes *Element* einer Menge ist ein Tupel, dessen Koordinaten die  $(x, y)$  Koordinaten eines Pixels im Bild sind. Es seien  $S$  und  $E$  zwei Mengen in  $Z^2$ , wobei  $Z$  die Menge der ganzen Zahlen,  $S = (s_1, s_2, \dots, s_n)$  mit  $s_i = (x_i, y_i)$  und  $E = (e_1, e_2, \dots, e_n)$  mit  $e_i = (x_i, y_i)$ .

---

<sup>15</sup>vgl. Kapitel 4

<sup>16</sup>vgl. z.B. [GiDo87], [GoRi92]

**Definition 3.10 (Translation)**

Eine **Translation** von  $S$  durch den Vektor  $p = (x, y)$  ist definiert<sup>17</sup> als:

$$S + p = \{t \mid t = s + p; s \in S\} \quad (3.11)$$

□

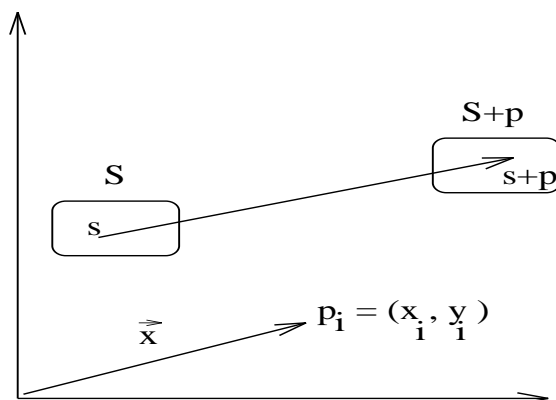


Abbildung 3.11: Translation einer morphologischen Menge:  
Translation:  $S$  wird entlang des Vektors  $\vec{x}$  übertragen

Abbildung 3.11 verdeutlicht Definition 3.10.

Die Grundinstrumente aller Verfahren, die auf mathematischer Morphologie aufbauen, bilden die Operationen *Dilatation* und *Erosion* sowie das *Strukturelement*.

**Definition 3.11 (Strukturelement)**

Ein **Strukturelement** ist eine morphologische Menge  $E$ , die eine geschlossene geometrische Form repräsentiert.

□

Beispiele typischer Strukturelemente zeigt Abbildung 3.12. Strukturelemente werden durch die Wahl der *Form* und der *Größe* des Strukturelementes anwendungsbezogen ausgesucht. Das meistverwendete Strukturelement ist das *Disk-* bzw. *Kreiselement*.

Eine *Dilatations*-Operation angewendet auf ein Bild (z.B. das Bild  $S$  in Abbildung 3.13), bewirkt eine Ausdehnung der Grenze der im Bild enthaltenen Objekte, entsprechend der Form des ausgewählten Strukturelementes, d.h. Objektränder werden hinzugefügt. Dieser Vorgang ist in Abbildung 3.14 verdeutlicht, dabei ist ein Diskelement<sup>18</sup> als Strukturelement verwendet worden.

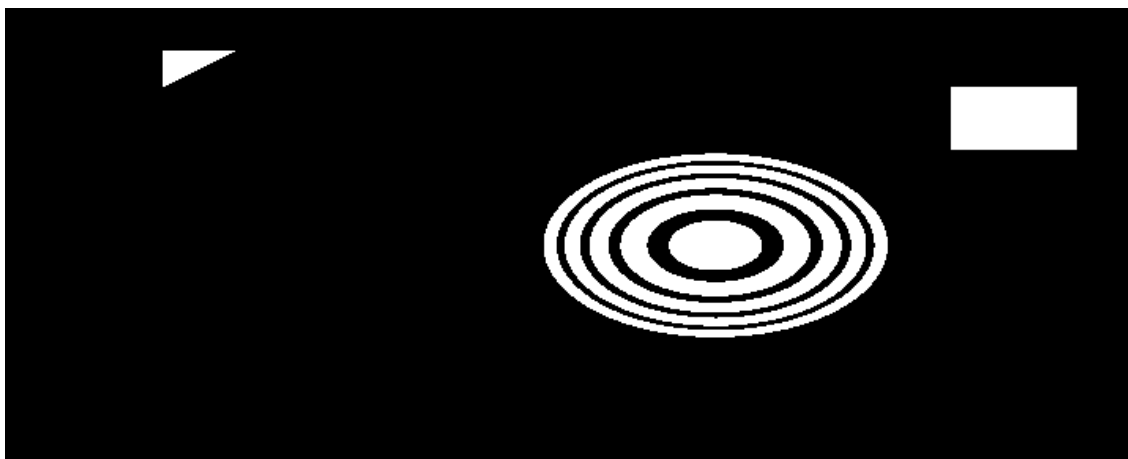
<sup>17</sup>vgl. [GoWo92], S. 519

<sup>18</sup>vgl. Abbildung 3.12 a)



| a) Diskelement  | b) Diamantelement |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <table style="border-collapse: collapse; margin: auto;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td></tr> </table> | 0                 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | <table style="border-collapse: collapse; margin: auto;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="padding: 2px 5px;">1</td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="padding: 2px 5px;">0</td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="border-right: 1px solid black; padding: 2px 5px;">1</td><td style="border-right: 1px solid black; padding: 2px 5px;">0</td><td style="padding: 2px 5px;">0</td></tr> </table> | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0   | 1                 | 1 | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1                 | 1 | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1                 | 1 | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1                 | 1 | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1                 | 1 | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 0                 | 1 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1                 | 1 | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 1   | 1                 | 1 | 1 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 1                 | 1 | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0   | 0                 | 1 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Abbildung 3.12: Häufig verwendete Formen von Strukturelementen

Abbildung 3.13: Morphologische Operationen: Das Originalbild  $S$ **Definition 3.12 (Dilatation)**

Seien  $S$  und  $E$  zwei Mengen in  $Z^2$ . Die **Dilatation** von  $S$  durch  $E$  ist definiert als:

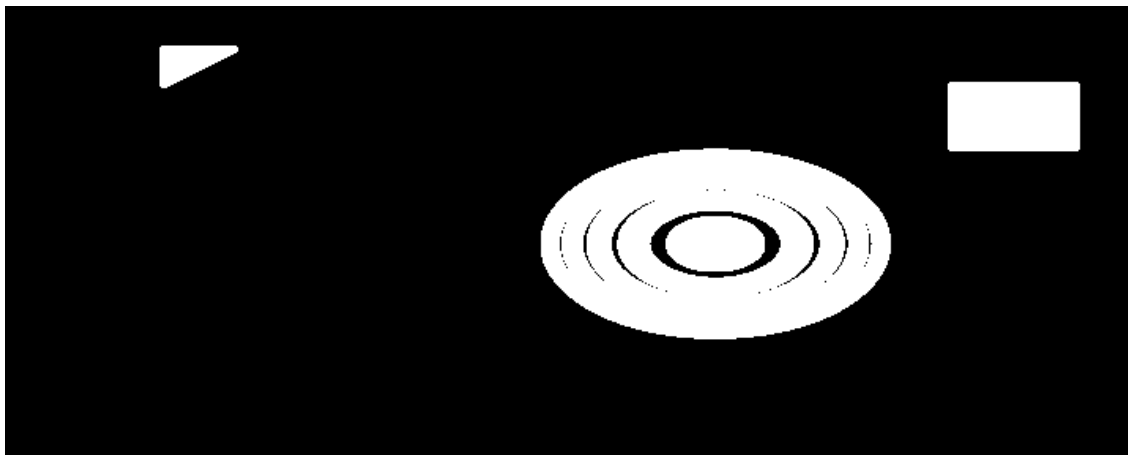
$$\mathcal{D}(S, E) = S \oplus E = \cup_{e \in E} (S + e) \quad (3.12)$$

□

Damit ist die Dilatation eine *Translation*<sup>19</sup> von  $S$  durch jedes Element der Menge  $E$  und eine *Vereinigung* der resultierenden Translationsmengen<sup>20</sup>.  $\mathcal{D}(S, E)$  ist eine Menge von Punkten, für welche  $S$  und  $E$  **mindestens** einen gemeinsamen Punkt haben. Die Menge  $E$  repräsentiert das ausgewählte *Strukturelement*.

<sup>19</sup>vgl. Gleichung 3.10

<sup>20</sup>vgl. [GiDo87], S. 4

Abbildung 3.14: Auswirkung einer Dilatation:  $S$  dilatiert durch  $E$ 

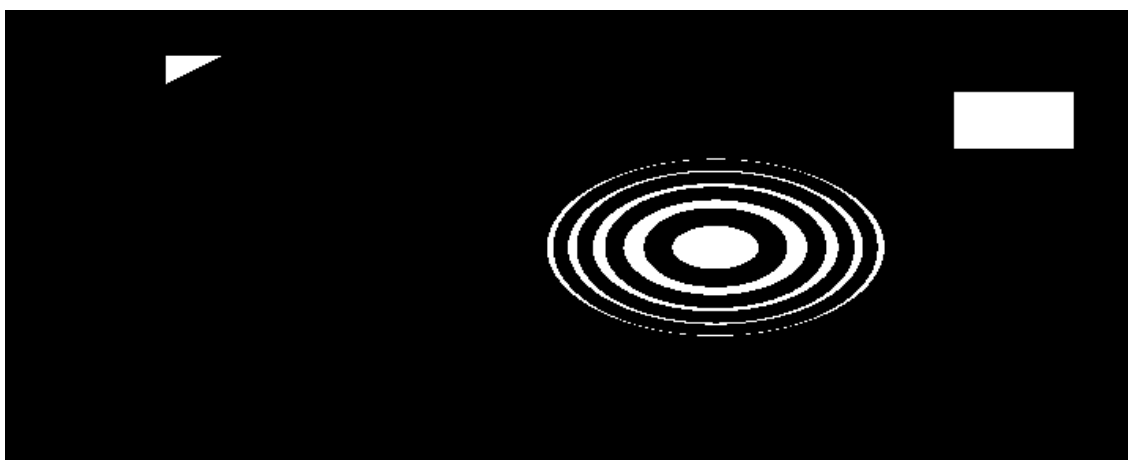
Die *Erosion* eines (Zweipegel-)Bildes bewirkt eine Schrumpfung der Grenze der im Bild enthaltenen Objekte, d.h. Objektränder werden abgetragen. Abbildung 3.15 verdeutlicht diesen Vorgang.

**Definition 3.13 (Erosion)**

Seien  $S$  und  $E$  zwei Mengen in  $Z^2$ . Die **Erosion** von  $S$  durch  $E$  ist definiert als:

$$\mathcal{E}(S, E) = S \ominus E = \bigcap_{e \in E} (S + e) \quad (3.13)$$

□

Abbildung 3.15: Auswirkung einer Erosion:  $S$  erodiert durch  $E$ 

D.h.: Aus der Erosion resultiert eine *Translation* von  $S$  durch jedes Element der Menge  $E$  und eine *Durchschnittsbildung* der resultierenden Translationsmengen.  $\mathcal{E}(S, E)$  ist eine Menge von Punkten, für welche  $E$  **ganz** in  $S$  enthalten ist.

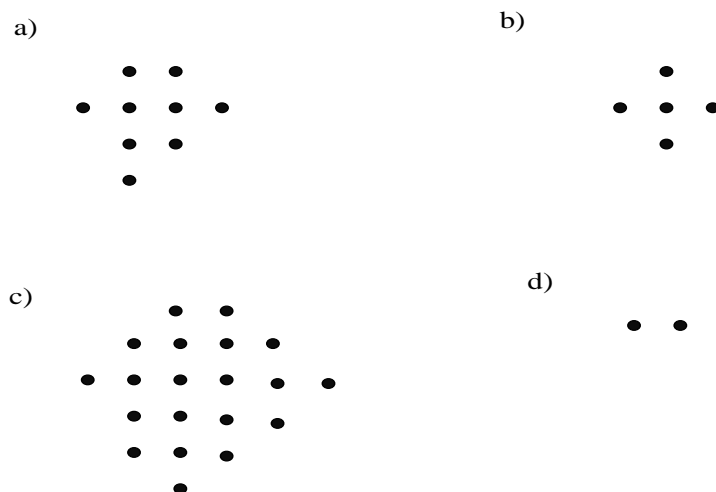


Abbildung 3.16: Dilatation und Erosion eines Gebiets mit einem Diskelement

- a) Das Gebiet  $G$ , b) Das Diskelement  $E$ ,  
 c) Dilatation  $G \oplus E$ , d) Erosion  $G \ominus E$

Abbildung 3.16<sup>21</sup> verdeutlicht noch einmal die Ausdehnung bzw. die Schrumpfung eines Gebiets mittels einer Dilatation bzw. Erosion. Dabei hat das Strukturelement die Form eines Kreises (Diskelement).

### 3.3.3 Einige morphologische Grundalgorithmen

Mittels bestimmter Kombinationen von Dilatationen und Erosionen lassen sich wichtige Objektmerkmale extrahieren, da sich Effekte einer Erosion mittels einer Dilatation kompensieren, aber nicht korrigieren lassen. In verschiedenen Fragestellungen der Bildanalyse ist die Extraktion von Objekten mit besonderen Formeigenschaften aus dem Gesamtbildinhalt ein zentrales Problem, d.h. bestimmte Objektteile sollten gelöscht werden und andere sollten beibehalten werden. Dieses Problem läßt sich dadurch lösen, daß eine geeignete Operationenfolge und ein geeignetes Strukturelement ausgewählt werden. Die Kunst des Einsatzes morphologischer Operatoren besteht in der Auswahl solcher geeigneter Kombinationen. Diese wichtige Eigenschaft wird deutlicher durch die Betrachtung von zwei weiteren Grundoperationen der mathematischen Morphologie: das *Opening* und das *Closing*.

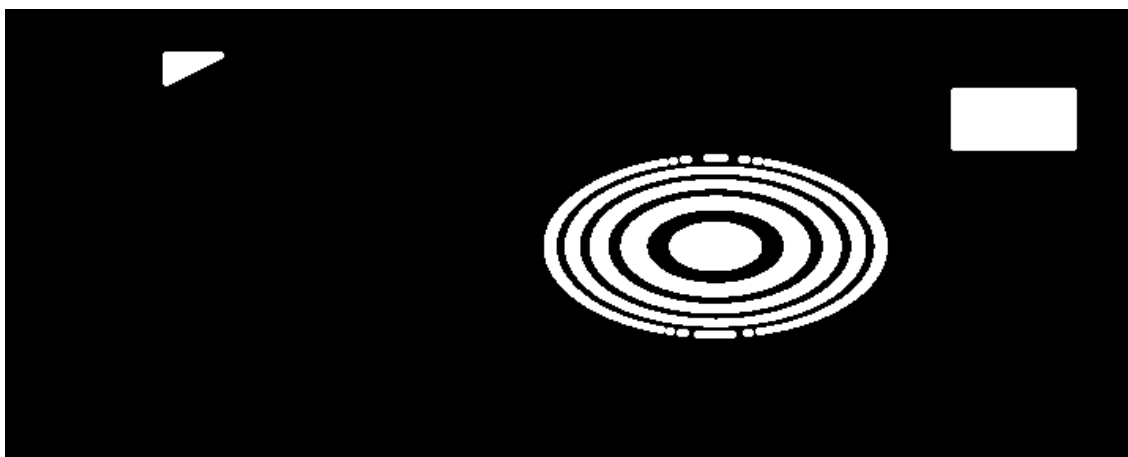
#### Definition 3.14 (Opening)

Seien  $S$  und  $E$  zwei Mengen in  $Z^2$ . Das **Opening** von  $S$  durch  $E$  ist definiert als:

$$\mathcal{O}(S, E) = (S \ominus E) \oplus E \quad (3.14)$$

<sup>21</sup>vgl. [PeCh94]

□

Abbildung 3.17: Opening von  $S$  durch  $E$ **Definition 3.15 (Closing)**

Seien  $S$  und  $E$  zwei Mengen in  $Z^2$ . Das **Closing** von  $S$  durch  $E$  ist definiert als:

$$\mathcal{C}(S, E) = (S \oplus E) \ominus E \quad (3.15)$$

□

Die Klammern bestimmen die Reihenfolge der Operatoren, die das Endergebnis stark beeinflusst (Erosion und Dilatation sind im allgemeinen nicht umkehrbar).

Das Opening dient im wesentlichen der Vereinfachung des Bildsignals durch das Herausheben von feinstrukturierten Bildanteilen (insbesondere Ränder) und von kleinen Bildteilen, wobei dieser Siebeffekt durch die Wahl der Form und insbesondere der Größe des Strukturelementes stark beeinflussbar ist. Nach einem Closing bleiben die groben Anteile des Originalbildes unverändert erhalten, kleine Risse, Lücken und feine Details werden dagegen aufgefüllt und mit den größeren Teilen *zusammengeschlossen*. Daher wird dieser morphologische Operator *Closing* genannt. In den Abbildungen 3.17 und 3.18 sind die Auswirkungen von Opening und Closing gezeigt.

Durch eine bestimmte, fallbezogene Kombination von Opening und Closing (open-close bzw. close-open) kann erreicht werden, daß das Bildsignal so gefiltert wird, daß Bildteile herausgesiebt werden und andere zusammenschlossen werden. Diese Filterung kann als ein Vorverarbeitungsschritt insbesondere vor einem Bildinterpretationsprozeß eingesetzt werden<sup>22</sup>.

Im folgenden ist eine Operationsfolge zur Konstruktion eines Konturpunktdetektors skizziert, die die oben genannten Eigenschaften morphologischer Operatoren verdeutlicht.

---

<sup>22</sup>vgl. [Sale94]

Abbildung 3.18: Closing von  $S$  durch  $E$ **Definition 3.16 (Morphologische Konturpunktdetektion)**

Seien  $S$  und  $E$  zwei Mengen in  $Z^2$ . Eine **morphologische Konturpunktdetektion** von  $S$  ist wie folgt definiert :

$$\mathcal{K}(S) = S - (S \ominus E) \quad (3.16)$$

□

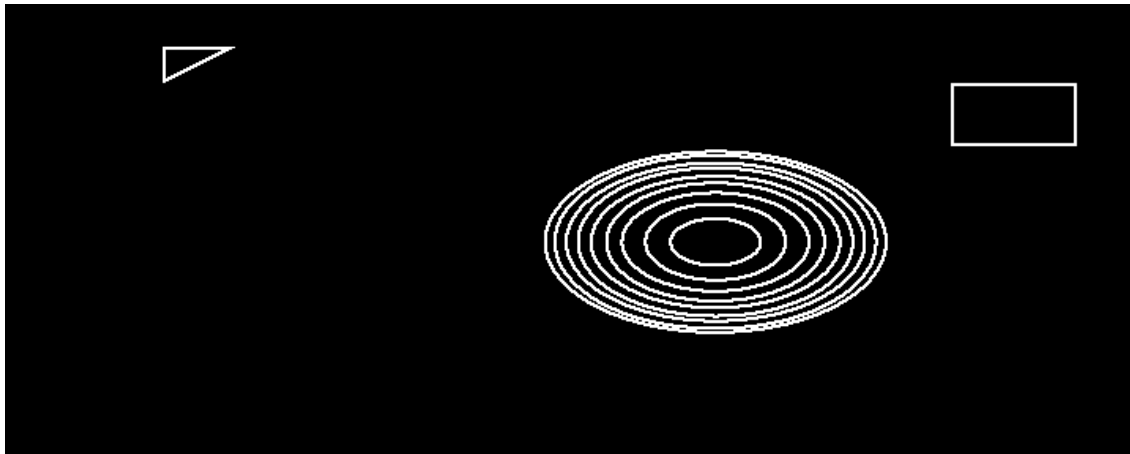


Abbildung 3.19: Ergebnis eines morphologischen Konturpunktdetektors

Die Operation „ $-$ “ ist durch eine Differenz zweier Mengen definiert. Die Abbildungen 3.13, 3.15 und 3.19 illustrieren den Mechanismus von Definition 3.16. Dieser Detektor weist ein gutes Rauschverhalten und eine sehr genaue Konturpunktllokalisierung (in den meisten Fällen liefert er eine Linienbreite von einem Pixel) auf. In Kapitel 4 wird eine andere Möglichkeit zur Auswahl einer Folge morphologischer Operationen für einen morphologischen Konturpunktdetektor gezeigt.

### 3.4 Regionenorientierte Objekterkennung in Bildsequenzen

Die Grundidee regionenorientierter Verfahren kann wie folgt formuliert werden: Ausgehend von einem *Keimpunkt*  $k_i$  wird eine *Region* durch sequentielle Angliederung *benachbarter Bildpunkte*  $p_1, p_2, \dots, p_n$  regelmäßig wachsen gelassen. Dabei wird ständig getestet, ob der *aktuelle Bildpunkt* ein vorgegebenes *Ähnlichkeitskriterium*  $A_i$  erfüllt. Günstige Keimpunkte und geeignete Ähnlichkeitskriterien sind wichtige Entscheidungen, die den Prozeß des Regionenwachstums stark beeinflussen. Abbildung 3.20 gibt

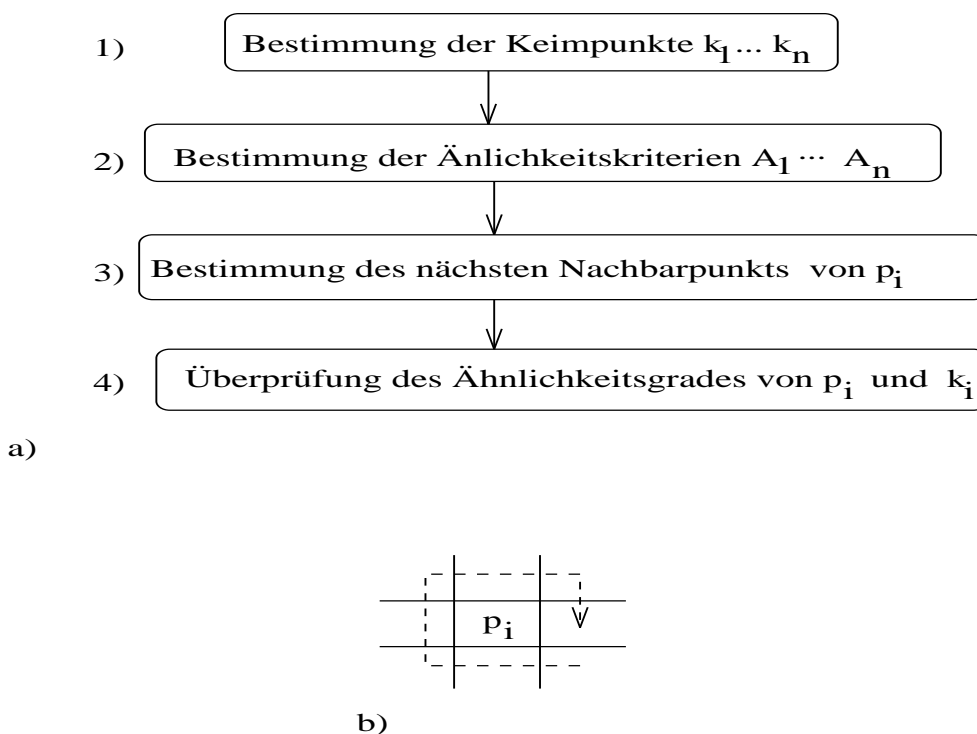


Abbildung 3.20: Grundkonzept regionenorientierter Verfahren

das Grundkonzept regionenorientierter Verfahren wieder:

1) **Bestimmung der Keimpunkte:**

Die Keimpunkte werden durch heuristische Verfahren bestimmt. Beispiel hierfür: im Bild werden kleine Gebiete gesucht, in denen statistische Merkmale, wie Mittelwert und Streuung, möglichst konstant bleiben. Dann werden die Schwerpunkte bestimmt und als Keimpunkte genommen.

2) **Bestimmung des Ähnlichkeitskriteriums:**

Zur Bestimmung des Ähnlichkeitskriteriums  $A_i$  wird im einfachsten Fall die absolute Abweichung des Grauwertes des aktuellen Punktes  $G_{p_i}$  vom Grauwert des Keimpunktes  $G_{k_i}$  genommen. Für eine angemessene Erkennung der Bildobjekte

werden oft zusätzlich andere Merkmale als der Grauwert berücksichtigt, z.B. das Texturmerkmal. In diesem Fall werden Methoden der Clusteranalyse angewendet, um den aktuell betrachteten Punkt anhand der verschiedenen, vorgegebenen Merkmale zu einem Cluster (Region) zuzuordnen<sup>23</sup>.

3) Reihenfolge der Nachbarpunkte:

Ausgehend vom Keimpunkt  $k_i$  werden die acht Nachbarpunkte des aktuellen Punktes auf ihre mögliche Regions- bzw. Clusterzugehörigkeit geprüft, wobei bereits klassifizierte Punkte markiert werden. Die Reihenfolge der Punktprüfung kann wie in Abbildung 3.20 b) festgelegt werden.

In der Literatur werden verschiedene regionenorientierte Verfahren genannt. Im wesentlichen lassen sich drei Techniken unterscheiden, die im folgenden kurz zusammengefaßt sind:

1. Lokale Techniken (engl. bottom-up)

Kleine benachbarte Regionen (im Initialzustand repräsentiert ein Bildpunkt eine Region) werden zu einer Region zusammengefaßt, falls die Merkmale dieser kleinen Regionen ein Ähnlichkeitskriterium erfüllen. Hierzu bieten sich Methoden der Clusteranalyse an<sup>24</sup>.

2. Globale Techniken (engl. top-down)

Zuerst wird das Gesamtbild betrachtet, so daß globale Merkmalsverteilungen (Homogenität) bestimmt werden. Ausgehend von diesen globalen Informationen wird eine Region (vom Gesamtbild aus) dann in kleinere Regionen geteilt, wenn sie ein Homogenitätskriterium nicht erfüllt (Verfahren, die ausschließlich auf globale Informationen aufbauen, werden in der Literatur wenig betrachtet).

3. Zusammengesetzte Techniken (engl. split and merge)

Lokale (räumliche) Ähnlichkeitskriterien und globale Merkmale (Homogenitätskriterien) werden verknüpft, so daß Regionen mit ähnlichen Eigenschaften miteinander verschmolzen werden und Regionen, die den Homogenitätstest verletzen, zerlegt werden.

## 3.5 Konturorientierte versus regionenorientierte Methoden

Ein gemeinsames Grundziel aller objekterkennenden Verfahren ist es, das Szenenbild in Bereiche zu zerlegen, deren Grenzen so gut wie möglich mit den realen Objektgrenzen übereinstimmen. Die Einteilung der Objekterkennungsmethoden in *konturorientierte* und *regionenorientierte* Verfahren folgt daraus, welches der beiden Merkmale der Strukturen im Bild zuerst extrahiert wird.

Verschiedene Konturextraktionsverfahren als Hilfsmittel zur Objekterkennung basieren auf der menschlichen Eigenschaft, Objekte in einer Bildszene gegenüber dem

---

<sup>23</sup>für Details vgl. [Zamp89]

<sup>24</sup>für Details vgl. [Mest89]

Hintergrund besser zu erkennen, wenn diese zumindest teilweise von einer Kontur umgeben sind. Der menschliche Betrachter einer Szene kann die Bewegung von Objekten schneller und besser durch ihre Kanten als durch ihre inneren Strukturen wahrnehmen. Denn Helligkeitsübergänge geben mehr Informationen als weitgehend homogene graue Bereiche. In den verschiedenen Anwendungen der Bildverarbeitung und Bildinterpretation sind die konturorientierten Verfahren mehr verbreitet. Zwei eindeutige Vorteile dieser Verfahren sind erstens die Schnelligkeit und zweitens der geringe Rechenaufwand. Im Hinblick auf die *empfängerseitige*<sup>25</sup> Zwischenbildinterpolation, bei der es auf Schnelligkeit und geringen Hardwareaufwand ankommt<sup>26</sup>, scheinen diese Verfahren sehr geeignet zu sein.

Ein Kritikpunkt an konturorientierten Verfahren betrifft die Verwendung von klassischen Konturpunktdetektoren als einen ersten Schritt der Konturextraktion. Diese Detektoren liefern im seltesten Fall geschlossene Konturen, so daß Nachverarbeitungsschritte sehr aufwendig sein können. Beim Einsatz von morphologischen Konturpunktdetektoren werden dagegen in den meisten Fällen geschlossene Konturen detektiert, so daß die entstandenen Konturen mit geringem Aufwand nachbearbeitet werden können<sup>27</sup>.

Ein anderer Kritikpunkt bei konturorientierten Verfahren ist ihre erhöhte Empfindlichkeit mit abnehmender Bildqualität bzw. zunehmendem Rauschgehalt eines Bildes. Denn sie betrachten für die Trennung von Konturpunktkandidaten von Hintergrundpunkten nur die lokale Nachbarschaft eines Kandidatenpunktes, um festzustellen, ob der Kandidatenpunkt als Konturpunkt tauglich ist. Dieser Kritikpunkt trifft bei klassischen Konturpunktdetektoren zu, die in ihrem Grundkonzept nur die  $\mathfrak{B}$ -Nachbarschaft des Kandidatenpunktes betrachten. Um trotz dieses Mangels eine ausreichend gute Konturextraktion zu erreichen, können folgende Vorverarbeitungsschritte unternommen werden:

- Geeignete Filterung des Bildes.
- Einbeziehung von anderen Punktmerkmalen und Vergrößerung des betrachteten Nachbarschaftsbereichs zur Auswahl der Konturpunktkandidaten.

Die morphologischen Konturpunktdetektoren sind weniger rauschempfindlich als die klassischen Detektoren, so daß sie statt der klassischen Detektoren eingesetzt werden können, um die o.g. Nachteile zu vermindern.

Zur Objekterkennung in Bildsequenzen werden regionenorientierte Verfahren eingesetzt, wegen ihrer Eigenschaft, auch globale Informationen (nicht nur lokale) in den Entscheidungsprozeß einzubeziehen. Sie arbeiten bereichsbezogen und nicht bildpunktbezogen, wie dies der Fall bei konturorientierten Verfahren ist, so daß sie alle Merkmale dieser Bereiche für die Zerlegung des Bildes in disjunkte Teilbereiche nutzen. Wegen der o.g. Eigenschaft regionenorientierter Verfahren schneiden sie in bezug auf Fehlersicherheit bei komplexem Bildmaterial besser ab. Auf der anderen Seite sind diese Verfahren langsam und aufwendig, da für die Zuordnung jedes Bildpunktes zu einer Region und

---

<sup>25</sup>vgl. Abbildung 2.1

<sup>26</sup>vgl. Kapitel 2

<sup>27</sup>vgl. Kapitel 4



für die Bestimmung von globalen Informationen zur Zerlegung von großen Regionen ein relativ großer Aufwand erforderlich ist. Ein anderes Problem dieser Verfahren ist auch die Schwierigkeit der Formulierung von Homogenitäts- und Ähnlichkeitskriterien (vgl. Kapitel 3.3).

In der Literatur sind *hybride* Verfahren bekannt, die regionen- und konturorientierte Strategien kombinieren<sup>28</sup>. Für die Erkennung von Objekten bei solchen Verfahren scheint die Extraktion von Kontur- und Regioneninformationen sowie die anschließende Integration beider Informationen sehr aufwendig zu sein.

---

<sup>28</sup>vgl. [Mest89]

# Kapitel 4

## Ein Konzept zur Erkennung von Objekten in Bildsequenzen

### 4.1 Problemstellung und Motivation

Verfahren der Bildanalyse und Bildinterpretation ermöglichen erforderliche Aussagen, insbesondere über strukturelle Anordnungen der verschiedenen Bildteile komplexer Bilder, für die jeweilige Anwendung herzuleiten. Diese quantitative Beschreibung des Bildinhaltes setzt leistungsfähige Isolierungs-, Erkennungs- und Beschreibungsmethoden dieser Bildteile voraus. Für einen menschlichen Betrachter wird das Problem der Erkennung von Bildobjekten durch Verknüpfung von *lokalen* und *globalen* Informationen über das Bild und die im Bild enthaltenen Strukturen (Objekte) einfach und unbewußt gelöst. Es ist deshalb plausibel, in einen Objekterkennungsalgorithmus *lokale*, wie *globale* Informationen mit einzubeziehen. Das Ziel dieses Kapitels besteht darin, die in dieser Arbeit entwickelte konturorientierte Methode aufzuzeigen, die sich für die Erkennung von Objekten in Bildern an der menschlichen Art, Objekte vorwiegend durch ihre Konturen und Ecken wahrzunehmen (Machscher Effekt, laterale Hemmung)<sup>1</sup>, orientiert und auf der mathematischen Morphologie basiert, die sehr vorteilhaft<sup>2</sup> für die formorientierte Bildinterpretation ist.

Ziel der automatischen Objekterkennung ist es, ein Bildsignal in zusammenhängende Komponenten zu zerlegen, die möglichst genau mit den empirisch wahrgenommenen Objekten übereinstimmen. Eine Interpretation des Bildinhaltes, verstanden als Erkennen von Objekten durch eine Wissensbasis („das ist ein Auto“), ist nicht Bestandteil des hier entwickelten Objekterkennungsverfahrens. Ein objekterkennendes System mit hinreichend allgemeiner Anwendbarkeit existiert nicht. Die bisherigen Verfahren sind für spezielle Anwendungen entwickelt worden, weil sich nur dadurch viele Probleme bei der Bildanalyse lösen lassen, aber insbesondere, weil sich Bilder in ihrer Beschaffenheit, Rauschcharakteristik, usw. stark unterscheiden.

Eine typische Eigenschaft natürlicher Bildsequenzen ist es, daß Bildobjekte eine Berandung haben und daß sie sich unabhängig voneinander bewegen können. In den

---

<sup>1</sup>vgl. [Haus94]

<sup>2</sup>vgl. Kapitel 4.4

verschiedenen Anwendungen der Bildanalyse werden die Objektkonturen bevorzugt analysiert, vor allem, weil sich durch die gewonnenen Konturinformationen topologische (globale) Informationen über die Bildobjekte sehr einfach ableiten lassen. Die Konturen als Grenze zwischen Objekten und Hintergrund sowie zwischen Objekten und ihren Nachbarobjekten sind sehr wichtig, da die Bildwahrnehmung sich hauptsächlich auf sie richtet. Experimente haben gezeigt, daß sich die Aufmerksamkeit der Menschen beim Betrachten eines Bildes in erster Linie auf Kanten und Ecken richtet und in einer zweiten Phase das Innere von Objekten vom Auge wahrgenommen wird. Das Idealziel bei der Konturanalyse ist die Extraktion von Konturen der im Bild enthaltenen Objekte, die *alle* realen, *exakten* Objektkonturen des Bildes, und *nur* sie, wiedergeben.

In der Regel werden an die Objekterkennung hohe Geschwindigkeitsanforderungen gestellt. Für die *empfängerseitige*<sup>3</sup> Zwischenbildinterpolation sind schnelle Bewegungsschätzverfahren eine wichtige Voraussetzung, um sie in praktischen Anwendungen einzusetzen. Um diesen hohen Geschwindigkeitsanforderungen gerecht zu werden, ist es notwendig, die Laufzeit der Algorithmen bei Beibehaltung ihrer Leistungsfähigkeit zu verringern. Um dies zu erreichen, soll ein Objekterkennungsverfahren in einfache Teilschritte gegliedert werden. Zusätzlich sollen diese Teilschritte mit möglichst geringem Hardwareaufwand realisierbar sein.

Um zu gewährleisten, daß der hier entwickelte Objekterkennungsalgorithmus für verschiedene Anwendungen nutzbar ist, wird eine modulare Gestaltung des Algorithmus angestrebt, wobei die Effektivität des Algorithmus im Hinblick auf die Abarbeitungszeit in praktischen Anwendungen im Vordergrund stehen soll. Das Problem der Objekterkennung im Sinne von Definition 3.5 wird durch eine „top-down“ Vorgehensweise behandelt, d.h. es werden ausgehend von einer globalen Zerlegung des Bildes schrittweise lokale Umgebungen zur Gewinnung von Objektinformationen betrachtet. Hier wird die Objekterkennung in mehrere Teilschritte zerlegt, so daß *Objektinformationen*, die in einem vorangegangenen Schritt ermittelt worden sind, in dem nachfolgenden Schritt zur Verbesserung der Objekterkennung eingebunden werden. Diese schrittweise Erkennung basiert auf dem *Prinzip der Reduktion* von Bildpunkten, die für die Weiterverarbeitung in dem folgenden Schritt nicht signifikant sind. *Die Transformation eines Bildes, durch die die Punkte der neuen Strukturen des transformierten Bildes auf für die Weiterverarbeitung signifikante Punkte reduziert werden, wird hier als Prinzip der Reduktion von Strukturpunkten eines Bildes bezeichnet.* In Kapitel 4.2 wird der Entwurf des entwickelten konturorientierten Erkennungskonzeptes grob beschrieben. In den darauf folgenden Kapiteln werden die einzelnen Komponenten dieses Konzeptes näher erläutert.

## 4.2 Das neue konturorientierte Objekterkennungsverfahren

Aufgabe dieses Kapitels ist es, einen kurzen, kompakten Überblick über die Komponenten des, in den darauf folgenden Kapiteln näher erläuterten, Erkennungsverfahrens

---

<sup>3</sup>vgl. Abbildung 2.1

zu geben. Ein Flußdiagramm wird dann den Entwurf dieses Algorithmus verdeutlichen.

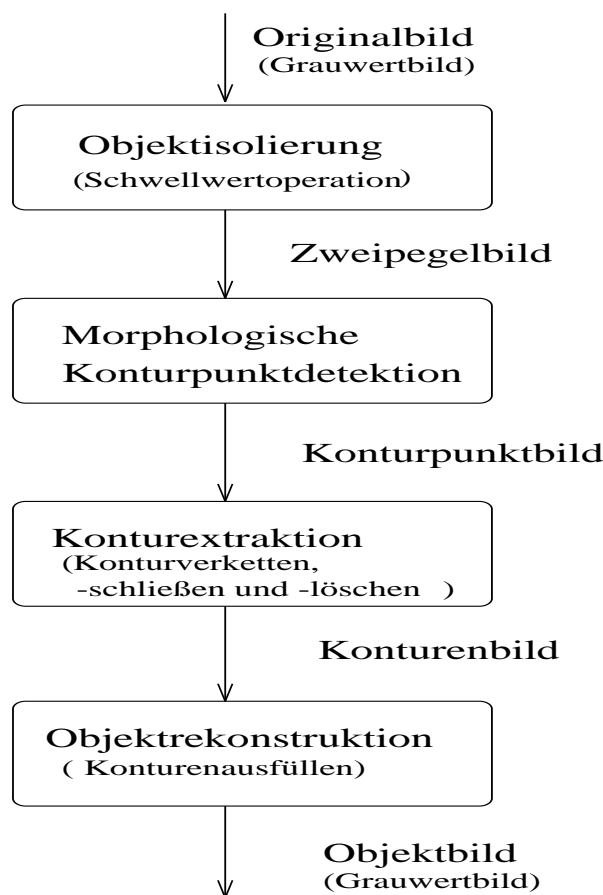


Abbildung 4.1: Flußdiagramm des vierstufigen Objekterkennungsverfahrens

Die Zielsetzung des neuen Verfahrens besteht darin, in erster Linie die wenigen, großen Objekte im Bild zu erkennen. Diese Objektauswahl erfolgt aus den physiologischen Wahrnehmungseigenschaften des menschlichen Auges, das in die Bewegung solcher Objekte einfällt, d.h. ihnen folgt („tracking“), und für diese Objekte dann ein erhöhtes Auflösungsvermögen besitzt.

Das Prinzip des Algorithmus kann wie folgt zusammengefaßt werden: Aus einem Originalbild (Halbbild) einer Sequenz (Abbildung 4.2) wird durch Trennung der *Objektpunkte* von den *Hintergrundpunkten* (*Nichtobjektpunkten*) zunächst ein *Zweipegelbild* (Abbildung 4.3) erzeugt. Dann wird mit Hilfe zusammengesetzter morphologischer Grundoperationen das *Zweipegelbild* in ein *Konturpunkt- bild* (Abbildung 4.4) transformiert, deren Punkte in drei Klassen unterteilt sind: in die Klasse der *Konturpunkte* (*Objektgrenzpunkte*), die einen vorgegebenen Grauwert erhalten, in die Klasse der *Objektuntergrundpunkte* (*Objektinnenpunkte*) und in die Klasse der *Hintergrundpunkte* (*Nichtobjektpunkte*). Diese Bildtransformation liefert unabhängige Konturelemente, die im Bild ohne räumliche Beziehung stehen. Um globale, für weitere Erkennungsschritte

notwendige, Objektinformationen zu gewinnen, müssen die Konturelemente in einem räumlichen Zusammenhang gebracht werden. Dies wird dadurch erreicht, daß die Konturpunkte des Konturpunktbildes, jeweils ausgehend von einem Konturanfangspunkt, miteinander zu Konturen verkettet werden. Dieser Verkettungsvorgang liefert ein Konturenbild (Abbildung 4.5), dessen Punkte in die Klasse der global miteinander *verketteten Konturpunkte*, in die Klasse der *Objektuntergrundpunkte* und in die Klasse der *Hintergrundpunkte* unterteilt sind. Bei diesem Verkettungsvorgang werden kleine und falsch erkannte Konturen eliminiert, die nicht signifikant für die Bewegungsschätzung sind, d.h. es werden die wenigen und großen Objektkonturen extrahiert. Dieses nur durch die Konturpunkte und ihre räumlichen Beziehungen charakterisierte Konturenbild ist als Grundlage für eine leistungsfähige *objektbasierte* Bewegungsschätzung, die auf die genaue Lage (Position) aller Objektpunkte angewiesen ist, allein nicht ausreichend. Deshalb wird aus dem Zweipegelebild (Konturenbild) ein Grauwertbild (Abbildung 4.6) rekonstruiert, dessen Punkte wieder in die Klasse der *Objektpunkte* und in die Klasse der *Hintergrundpunkte (Nichtobjektpunkte)* unterteilt sind. In diesem rekonstruierten Grauwertbild (Abbildung 4.6) sind die im Eingangsbild (Abbildung 4.2) enthaltenen Strukturen<sup>4</sup> als homogene unabhängige Objekte<sup>5</sup> erkannt worden.

Im folgenden sind die Komponenten des neuen vierstufigen Ansatzes zusammengestellt:

1. *Lokalisation von Objektpunkten* aus der Menge aller Bildpunkte (Kapitel 4.3: Schwellwertbasierte Objektisolierung)
2. *Detektion von Konturpunkten* aus der Menge aller Objektpunkte (Kapitel 4.4: Morphologische Konturpunktdetektion)
3. *Zusammenfassung der ermittelten Konturpunkte zu Konturen*, um zu einer globalen Beschreibung der Objekte zu gelangen. Dabei werden kleine, nicht signifikante Konturen eliminiert. (Kapitel 4.5: Erkennung von Objektkonturen)
4. Verwendung des „A-priori“-Konturwissens und der globalen Informationen, die aus den ermittelten Konturen zu gewinnen sind, zur *Rekonstruktion von homogenen Objekten*, wobei jedes Objekt mit einem einheitlichen Grauwert versehen wird. (Kapitel 4.6: Rekonstruktion von homogenen Objekten)

Jede der oben genannten Erkennungskomponenten ist grundsätzlich auf eine Vereinfachung des Bildinhaltes durch Entfernung von nicht signifikanten Informationen für die folgenden Erkennungskomponenten und auf eine Extraktion der signifikanten Merkmale der vereinfachten Bildstrukturen für die folgende Phase angewiesen<sup>6</sup>.

Abbildung 4.1 gibt einen Gesamtüberblick über die Komponenten des Algorithmus sowie die dadurch erzeugten Bildtransformationen. Die Abbildungen 4.2–4.6 verdeutlichen die Arbeitsweise dieser Komponenten anhand eines Anwendungsbeispiels.

---

<sup>4</sup>vgl. Definitionen 3.1, 3.5

<sup>5</sup>vgl. Definition 3.6

<sup>6</sup>Prinzip der Reduktion, vgl. Kapitel 4.1



Abbildung 4.2: Ein Originalbild aus der „car“-Sequenz  
(Ein Auto fährt von rechts nach links,  
der Hintergrund bewegt sich von links nach rechts  
(Kameraschwenk mit dem bewegten Foto))



Abbildung 4.3: Das schwellwertbasierte Zweipegelbild zu Bild 4.2 (invertiert)

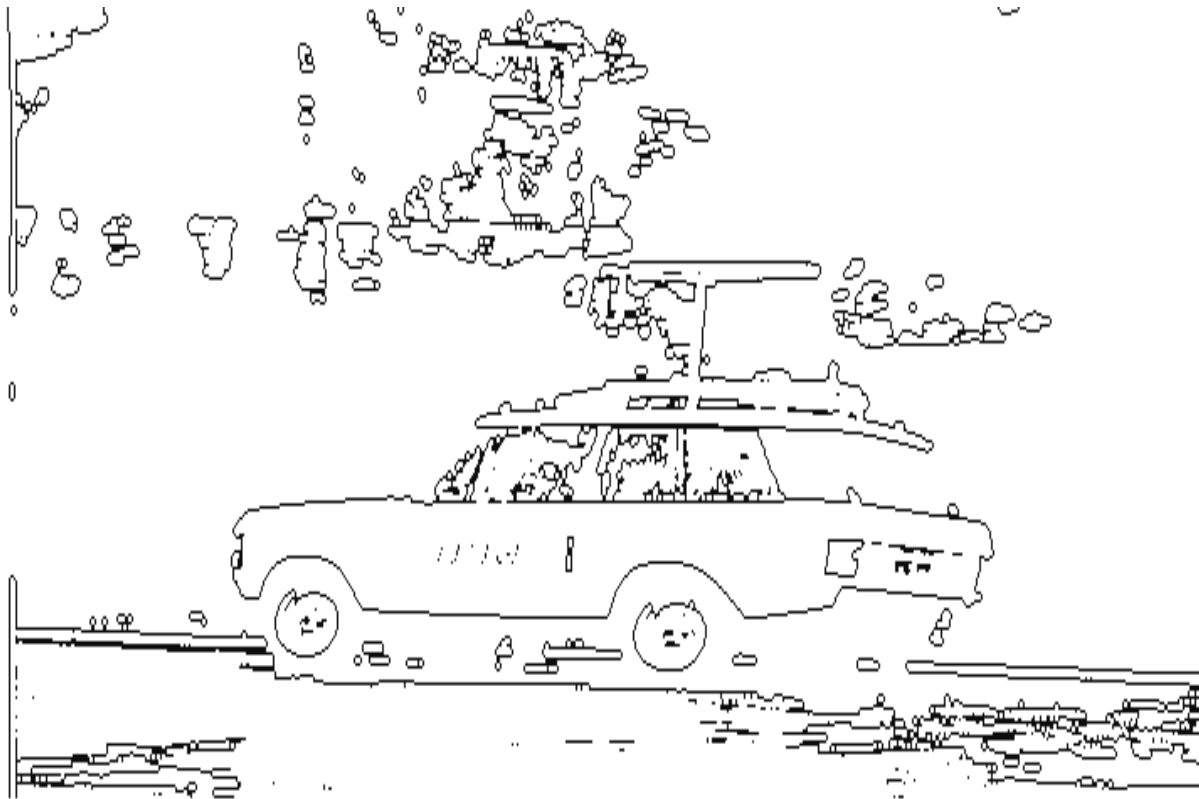


Abbildung 4.4: Konturpunktbild zu Bild 4.3:

Das Endergebnis des Detektors nach Definition 4.1 (invertiert)



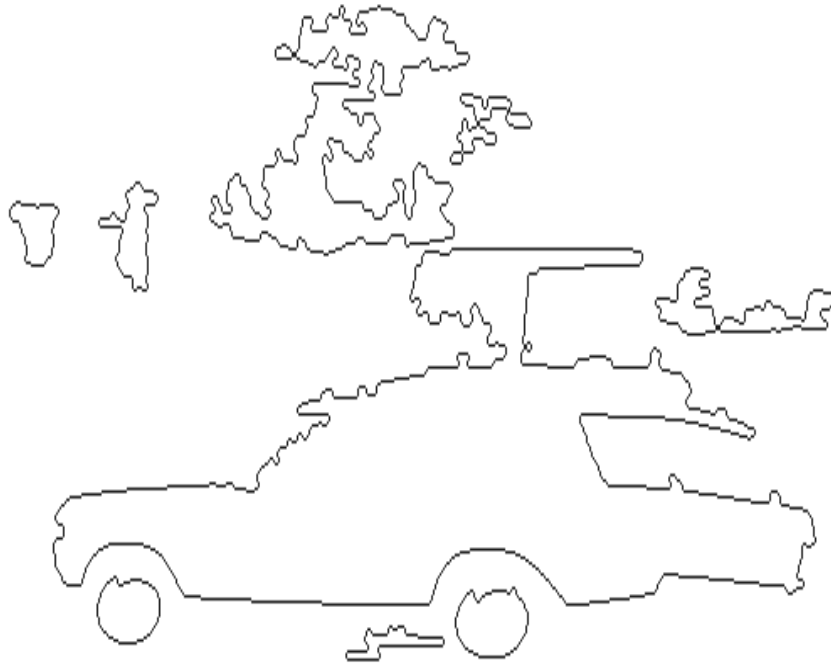


Abbildung 4.5: Das resultierende Konturenbild zu Bild 4.4 (invertiert)  
(einpixelbreite Konturen)



Abbildung 4.6: Die rekonstruierten (erkannten) Objekte aus Bild 4.5 (invertiert)

### 4.3 Schwellwertbasierte Objektisolierung

Schwellwertverfahren werden in vielen praktischen Anwendungen der automatischen Bildanalyse eingesetzt. Mit einer Schwellwertoperation wird durch eine einfache *lokale* Entscheidung eine *globale* Bildunterteilung erreicht. Die Schwellwertverfahren sind daher sehr effektiv und lassen sich leicht implementieren. Die Bildpunkte werden durch Festlegung einer Grauwertschwelle in zwei Klassen unterteilt, die Klasse der Bildpunkte, deren Grauwert größer oder gleich dem festgelegten Schwellwert ist, und die Klasse der Bildpunkte, deren Grauwert unterhalb des Schwellwertes liegt. Dabei wird davon ausgegangen, daß der Bildwertabstand zwischen Objektpunkten des Eingangsbildes und Hintergrundpunkten groß ist (starker Kontrast). Damit umfaßt eine Klasse die *Hintergrundpunkte* und die andere Klasse die *Objektpunkte*. Das entscheidende Problem bei einer solchen Bildpunktklassifizierung liegt in der Auswahl eines geeigneten Schwellwertes.

In der Regel wird die automatische Schwellwertauswahl durch die Auswertung des Bildhistogramms vollzogen: zunächst wird die Verteilung der Grauwerte aller Bildpunkte ermittelt, dann wird ein Grauwert aus dieser Verteilung nach einem bestimmten Kriterium als Schwelle ausgewählt. Als Kriterium kann z.B. das Minimum (bzw. das Maximum) des Grauerthistogramms gewählt werden. In dem Fall, daß das Histogramm mehrere ausgeprägte Maxima aufweist, werden mehrere Schwellen für die Unterteilung

von Bildpunkten in mehrere Klassen<sup>7</sup> (also nicht nur Objekt- und Hintergrundklassen) festgelegt. [BBbv91] gibt eine ausführliche Beschreibung von Algorithmen zur Bestimmung von Schwellwerten und zur Klassifikation von Bildpunkten aufgrund mehrerer Schwellen.

Das Originalgrauwertbild wird in dem hier entwickelten Erkennungsverfahren als erster Schritt zur Objekterkennung mittels eines vorgegebenen Schwellwertes (Steuerparameter) in zwei Klassen unterteilt: auf der einen Seite die Klasse der *Objektpunkte*, deren Grauwert dem vorgegebenen Schwellwert gleich oder ihn übersteigt, und auf der anderen Seite die Klasse der *Hintergrundpunkte*, deren Grauwert unterhalb des Schwellwertes liegt. Somit wird den Objektpunkten ein vorgegebener Grauwert (*Objektgrauwert*) und den Hintergrundpunkten ein *Hintergrundgrauwert* zugeordnet. Auf diese Weise wird durch die *Reduktion* von nicht signifikanten Bildpunkten ein *Zweipegelbild* erzeugt. Durch die Abbildungen 4.2 und 4.3 wird das Ergebnis der oben beschriebenen schwellwertbasierten Objektisolierung verdeutlicht.

In den seltensten Fällen kann es allein auf der Grundlage eines Schwellwertverfahrens eine korrekte und ausreichende Objekttrennung geben. In Kapitel 4.4 wird ein auf der Betrachtung der Objektpunktumgebung basiertes Verfahren zur Verbesserung der bisher ermittelten Objektinformationen beschrieben.

## 4.4 Morphologische Konturpunktdetektion

### 4.4.1 Vorbemerkung

Im allgemeinen ist die Konturpunktdetektion eine Erkennungstechnik, die auf dem Herausfinden von Diskontinuität der Bildfunktion basiert. Das Ziel besteht darin, die Bildpunkte, die auf der einen Seite an der Grenze zwischen Objekten und Hintergrund und auf der anderen Seite an der Grenze zwischen Objekten und anderen Objekten im Bild liegen, als *Konturpunkte* zu klassifizieren<sup>8</sup> und die anderen Bildpunkte zu Hintergrundpunkten zu erklären. Die Anwendung von Konturpunktdetektoren bewirkt somit nur eine lokale Bildtransformation, bei der unabhängige Konturelemente ohne räumliche Beziehung entstehen. Dieser Vorgang der Vereinfachung des Bildinhaltes basiert auf dem Prinzip der Reduktion von Bildpunkten<sup>9</sup>.

Für die Konstruktion von Konturpunktdetektoren sind insbesondere folgende Bedingungen zu beachten:

- Die Detektion der Konturpunkte muß gewährleistet sein.
- Die Detektion der Konturpunkte soll an der richtigen Position erfolgen.
- Die durch die Konturpunkte gebildeten Konturlinien sollen dünn und eindeutig sein.

---

<sup>7</sup>Beispiel hierfür vgl. [Wolf91]

<sup>8</sup>vgl. Definitionen 3.7, 3.8 und 3.9

<sup>9</sup>vgl. Kapitel 4.1

Der hier entwickelte Konturpunktdetektor basiert auf der mathematischen Morphologie. Wie bereits gezeigt<sup>10</sup>, können morphologische Operatoren sehr effizient zur Extraktion von geometrischen Objektmerkmalen wie Form, Größe oder Verbindung eingesetzt werden. Im folgenden werden die wesentlichen Vorteile für einen Einsatz von auf mathematischer Morphologie basierenden Algorithmen in der digitalen Bildverarbeitung kurz erläutert<sup>11</sup>:

- Es existiert eine recht gut entwickelte, morphologische Algebra, die sich auf Grundlagen der Booleschen Algebra und der klassischen Mengenlehre zurückführen läßt. Mit Hilfe dieser Algebra lassen sich Optimierungen und weitere Repräsentationsmöglichkeiten herleiten.
- Durch eine modulare Konstruktion von einfachen, morphologischen Grundoperationen lassen sich selbst komplexe Bildverarbeitungsaufgaben lösen.
- Für das Herausfinden von Strukturen<sup>12</sup> aus Bilddaten eignen sich zusammengesetzte, morphologische Operatoren, wegen ihrer Eigenschaft, bestimmte Objektteile zu eliminieren, aber auch andere Teile hinzuzufügen<sup>13</sup>, so daß Objekte mit besonderen Formeigenschaften und mit für den Prozeß der Bildanalyse oder der Mustererkennung erwünschten Merkmalen, aus dem Bild extrahiert werden können.
- Morphologische Operatoren schützen wegen ihrer Beschaffenheit die wesentlichen geometrischen Formen der Bildobjekte. Bei linearen Filtern z.B. wird dagegen die eigentliche geometrische Form von Objekten eines Bildes zerstört<sup>14</sup>.
- Die durch morphologische Operationen zusammengesetzten Verfahren sind gut parallelisierbar<sup>15</sup>.
- Besonders für einen praktischen Einsatz eines *objektbasierten* Verfahrens sind morphologische Operatoren vor allem wegen ihrer effizienten und einfachen Software- und Hardwarerealisierbarkeit sehr attraktiv.

Die Anwendung eines morphologischen Detektionsverfahrens auf ein Zweipegelbild bewirkt eine Bildtransformation, bei der folgende Punktklassen entstehen: die Klasse der *Konturpunkte*, die Klasse der *Objektuntergrundpunkte* und die Klasse der *Hintergrundpunkte*, wobei der Grauwert der Objektuntergrundpunkte und der Hintergrundpunkte in diesem Fall identisch ist.

---

<sup>10</sup>[Sale94], [GiDo87], [Zamp89]

<sup>11</sup>vgl. [GiDo87]

<sup>12</sup>vgl. Definition 3.1

<sup>13</sup>vgl. Kapitel 3.3

<sup>14</sup>[GiDo87]

<sup>15</sup>vgl. [GiDo87]

## 4.4.2 Realisierung eines morphologischen Konturpunktdetektors

Im folgenden wird ein Entwurf und die Implementierung eines morphologischen Konturpunktdetektors dargestellt. Wie bereits erwähnt<sup>16</sup>, besteht die Kunst bei einer Konstruktion von morphologisch basierenden Algorithmen in der Auswahl einer geeigneten Kombination aus morphologischen Grundoperatoren und den dazugehörigen Strukturelementen, so daß das gewünschte Ziel dieser Algorithmen effizient erreicht wird. Im folgenden wird eine Operationsfolge zur Konstruktion eines morphologischen Konturpunktdetektors vorgestellt, die als Ziel die Erkennung von gut positionierten Konturpunkten, die an der Grenze zwischen Objekten und Hintergrund und zwischen Objekten und anderen Objekten liegen, hat.

### Definition 4.1 (Morphologische Konturpunktdetektion)

Seien  $S$ ,  $E_1$  und  $E_2$  Mengen in  $Z^2$ . Eine **morphologische Konturpunktdetektion** von  $S$  ist wie folgt definiert:

$$\mathcal{K}(S) = (S \text{ XOR } (S \oplus E_1)) \ominus E_2 \quad (4.1)$$

□

a) Diskelement

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |

b) Quadratelement

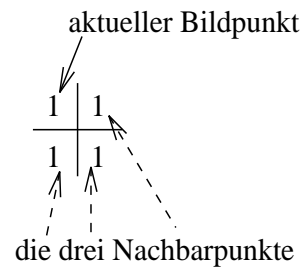


Abbildung 4.7: Die für die Realisierung des Detektors nach Definition 4.1 ausgewählten Strukturelemente

In dem Entwurf des morphologischen Konturpunktdetektors nach Definition 4.1 wird als erstes eine Dilatation des aus dem Schwellwertverfahren resultierenden Zweipegebildes (Abbildung 4.3) mittels eines Diskelementes der Größe  $5 \times 5$  (Abbildung 4.7) durchgeführt. Das Diskelement wird über das gesamte Bild geschoben. Zur Ermittlung des Zustandes jedes Punktes (gesetzt (weiß) oder nicht gesetzt (schwarz)) des dilatierten Bildes wird folgende Regel angewendet:

- **Wenn** das Diskelement **mindestens** einen gesetzten Punkt des Elementbereiches im Eingangsbild (im Zweipegebild) überdeckt, **dann** werden **alle** vom Diskelement überdeckten Bildpunkte im Ausgangsbild (das dilatierte Bild) gesetzt, falls der zugehörige Elementpunkt gesetzt ist.

<sup>16</sup>vgl. Kapitel 3.3



Abbildung 4.8: Das Ergebnisbild der Dilatation von Bild 4.3  
nach Definition 4.1 (invertiert)

Nun wird das Zweipegelebild (Abbildung 4.3) und das dilatierte Bild (Abbildung 4.8) mittels einer *XOR*-Funktion verknüpft. Das neue Zwischenbild enthält Konturlinien sowie eine Menge anderer, nicht signifikanter und kleinerer Bereiche. Diese Konturlinien geben im wesentlichen die Form der ursprünglichen Objekte wieder, sind aber für weitere Verarbeitungsschritte nicht ausreichend dünn. Mit Hilfe einer anschließenden Erosion werden dann aus diesem Zwischenbild kleinere Bereiche und unerwünschte Ränder der resultierenden Konturlinien eliminiert, so daß dünne und eindeutige Konturlinien generiert werden.

Die Erosion folgt mittels eines Quadratelementes (Abbildung 4.7), welches über das aus der *XOR*-Operation resultierende Zwischenbild (*XOR*-Bild) geschoben wird, so daß zur Ermittlung des Zustandes jedes Punktes des neuen Bildes folgende Regel angewendet wird:

- **Wenn nicht alle drei Nachbarpunkte** (Abbildung 4.7) *des betrachteten Punktes des Eingangsbildes (XOR-Bild), die vom Quadratelement überdeckt sind, gesetzt, dann wird der entsprechende Punkt im Ausgangsbild (im erodierten Bild) nicht gesetzt.*

Die Abbildungen 4.3, 4.4, 4.8 und 4.9 illustrieren den Mechanismus des nach Definition 4.1 entworfenen Detektors. Abbildung 4.10 verdeutlicht noch einmal den Algorithmus des entwickelten morphologischen Konturpunktdetektors.

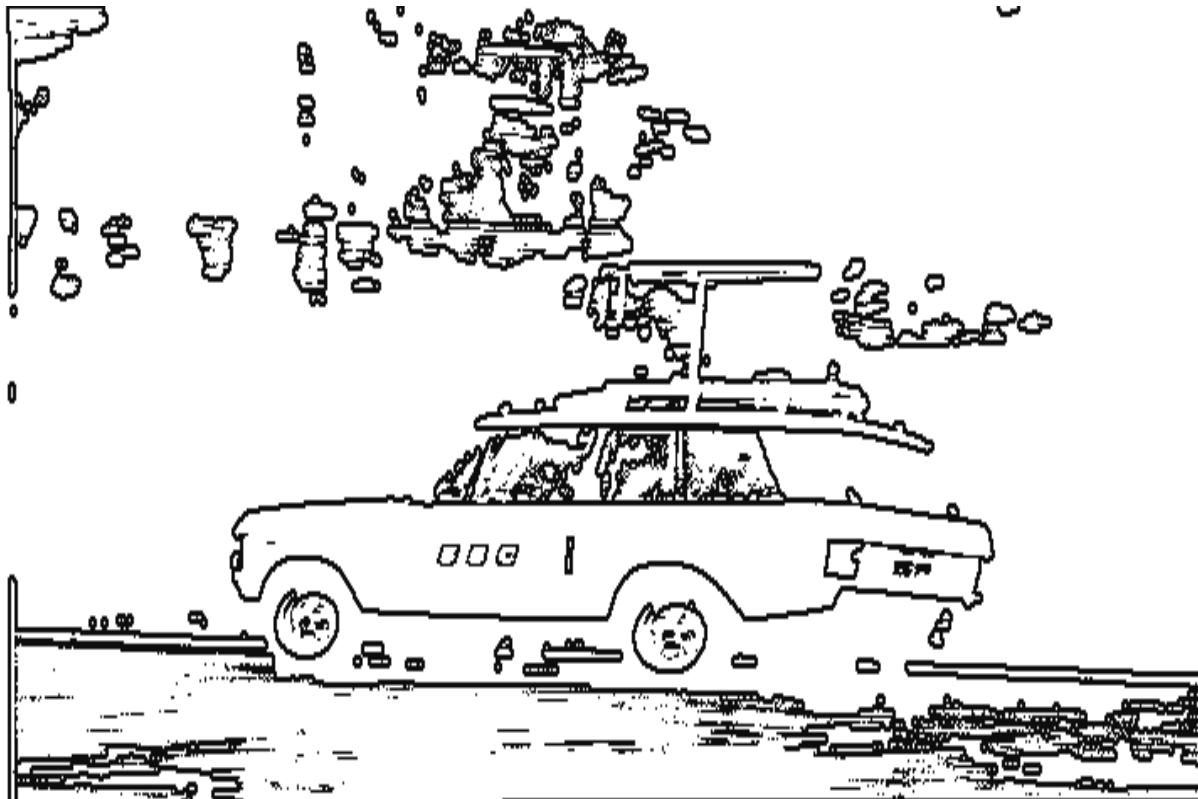


Abbildung 4.9: Das Ergebnisbild der XOR-Operation der Bilder 4.3 und 4.8 nach Definition 4.1 (invertiert)

Dieser Konturpunktdetektor hat folgende Eigenschaften:

- Der Detektor liefert gut positionierte Konturpunkte.
- Der Detektor weist eine genaue Konturpunktlokalisierung auf, d.h. er liefert dünne und eindeutige Konturlinien. In den meisten Fällen liefert er eine Linienbreite von einem Pixel. Diese Eigenschaft ist besonders für eine konturorientierte Erkennung von Objekten, an deren Grenze die detektierten Konturpunkte liegen, wichtig.
- Er weist ein gutes Rauschverhalten auf.

Ein wesentlicher Nachteil eines nur auf durch Konturpunktdetektoren ermittelten Konturlinien basierenden Algorithmus zur Objekterkennung besteht in der Schwierigkeit, aus dem Bild die für den Erkennungsprozeß erforderlichen globalen Objektinformationen zu gewinnen. Denn zunächst liegen nur separate Konturlinien vor und erst nach einem weiteren Verarbeitungsschritt der detektierten Konturpunkte (Verknüpfung der Konturlinien) lassen sich globale Objektmerkmale bestimmen. Dies bedeutet, daß einzelne Konturelemente zu einer Darstellung einer geschlossenen Linie verknüpft werden müssen, die eine semantische Interpretation zuläßt. In Kapitel 4.5 wird das Konzept eines auf den detektierten Konturpunkten basierenden Algorithmus zur Gewinnung von globalen Objektinformationen (Konturen) beschrieben.

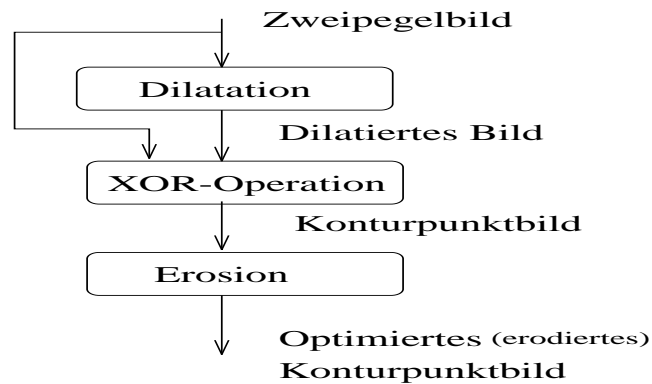


Abbildung 4.10: Flußdiagramm des morphologischen Konturpunktdetektors nach Definition 4.1

## 4.5 Erkennung von geschlossenen Konturen

### 4.5.1 Vorbemerkung

Konturpunktdetektoren bewirken nur eine lokale Bildtransformation, bei der unabhängige Konturelemente ohne räumliche Beziehung entstehen. In dem transformierten Bild können nicht signifikante Konturelemente, falsch detektierte Konturelemente und Lücken zwischen den ermittelten Konturelementen auftreten. Zur Bestimmung der eigentlichen Konturen von Objekten ist eine Zusammenhangsanalyse der detektierten Konturpunkte erforderlich, so daß eine Menge von endlich geordneten Folgen von benachbarten Bildpunkten,  $p_1 p_2 \cdots p_n p_1$ , (geschlossene Konturen) entsteht. Jede Folge von Konturpunkten stellt eine Kontur dar, die dann einem Bildobjekt eindeutig zugeordnet wird. Bei der Generierung von Konturen werden in erster Linie die wenigen und großen Objekte im Bild in Betracht gezogen.

Das Prinzip der Reduktion von nicht signifikanten Bildelementen wird hier auch berücksichtigt. Der Zusammenschluß von detektierten Konturelementen bedeutet demnach:

1. Eliminierung von als *Einzelpunkte* erkannten Konturpunkten nach folgender Regel:

Regel 1:

**WENN** der **Anfangspunkt** einer Kontur **keinen** Nachbarpunkt in seiner Achternachbarschaft hat,  
**DANN** wird er eliminiert.

2. Eliminierung von *kleinen, nicht signifikanten* entstandenen Konturen nach folgender Regel:



Regel 2:

**WENN** die Kontur geschlossen **und** „klein“ ist,  
**DANN** wird sie eliminiert.

3. Eliminierung von als falsch erkannten Konturelementen nach folgenden Regeln:

Regel 3:

**WENN** die Kontur „groß“ ist **und** eine kleine Lücke aufweist, deren Schließen eine Linie hervorruft, die „viele“ andere Konturpunkte überquert,  
**DANN** wird sie eliminiert.

Regel 4:

**WENN** die Kontur eine „große“ Lücke aufweist **und** „klein“ ist,  
**DANN** wird sie eliminiert.

Regel 5:

**WENN** die Kontur eine „große“ Lücke hat, deren Schließen eine Linie „quer“ durch das Bild hervorruft,  
**DANN** wird sie eliminiert.

Aus diesen drei Regeln ist der Fall des *Konturfehlers* ablesbar, d.h. das Konturverfolgungsverfahren hat falsche Verkettungswege durchlaufen. In diesem Fall werden die dadurch entstandenen Konturen als falsch und nicht signifikant für die Objekterkennung eingestuft und ignoriert.

4. Zusammenfassung von mehreren Konturelementen zu einer *geschlossenen Kontur*, die die oben genannten fünf Regeln nicht erfüllen.

Dieses Vorgehen richtet sich nach dem *Prinzip der Reduktion* der nicht signifikanten Bildpunkte zur Verringerung der Komplexität des Algorithmus. Im Gegensatz zu vielen bekannten Konturfindungsverfahren, die komplexe mathematische Kriterien zur Konturfindung heranziehen, wird in dieser Arbeit eine Methode der Konturfindung vorgestellt und angewendet, die sich auf eine einfache Konturpunktverkettungsstrategie stützt.

## 4.5.2 Konturextraktion

Das Ziel dieses Kapitels ist es, ein Verfahren zu erläutern, mittels dem aus den durch die Detektion der Konturpunkte in *lokaler* Form vorliegenden Bilddaten möglichst viele *globale* Informationen über die Bildobjekte hergeleitet werden. Hier muß jedoch vor allem die Frage der Effektivität des Verfahrens berücksichtigt werden. Bei dem hier angewendeten Konturextraktionsverfahren wird ausgehend von einem als Anfangspunkt eines Objektes erkannten Bildpunkt (Objektsuche), mittels einer zunächst *lokalen* (in der Achternachbarschaft) Suchstrategie von Punkt zu Punkt, der nächste Nachbarpunkt (Konturpunkt) gesucht. Falls aber in dieser lokalen Umgebung kein Folgepunkt

gefunden wird (Lücke), folgt der Suchvorgang dann schrittweise in einer *globaleren* Nachbarschaft (Lückenschließen). Die dadurch gefundenen Konturpunkte werden mittels eines *Codierungsalgorithmus*, der die Richtung der gefundenen Punkte codiert, und durch die Angabe der absoluten Ortskoordinaten der gefundenen Konturpunkte zu einer Kontur verkettet und dargestellt.

## Objektsuche

Das durch den morphologischen Detektor erzeugte Konturpunktbild wird zeilenweise abgetastet, bis ein Konturpunkt gefunden wird, der dann als *Anfangspunkt* der zu erkennenden Kontur bezeichnet wird. Das Finden eines Anfangspunktes reklamiert das Finden eines neuen Bildobjektes. Die Objektsuche wird nach der Vollendung der Beschreibung einer Kontur durch Weiterabtastung des Bildes ab dem Anfangspunkt der zuletzt festgelegten Kontur fortgesetzt, bis das Ende des Bildes erreicht ist. Wenn das Bild mehrere Objekte enthält, alternieren somit Objektfinden und Konturdarstellung so lange, bis das gesamte Bild auf alle Objekte abgetastet ist.

## Darstellung von Konturen

Der Vorgang der Konturerkennung wird nach einer erfolgreichen Objektsuche durch die folgenden Schritte vollzogen:

**Konturpunktverfolgung:** Ausgehend von dem gefundenen Konturanfangspunkt wird in der Achternachbarschaft<sup>17</sup> eines gefundenen Konturpunktes nach dem nächsten Folgekonturpunkt gesucht. Falls aber in der Achternachbarschaft *des Anfangspunktes* kein Folgepunkt gefunden wird, wird der Anfangspunkt als Einzelpunkt eingestuft und somit ignoriert (gelöscht) (*Regel 1: Wenn der Anfangspunkt einer Kontur keinen Nachbarpunkt in seiner Achternachbarschaft hat, dann wird er eliminiert*).

Im Falle einer erfolgreichen Suche in der Achternachbarschaft wird die gleiche Suchstrategie sukzessiv für jeden gefundenen Punkt angewendet. Diese lokale Suchstrategie ist wie folgt definiert: der Folgepunkt des aktuellen Punktes wird in den acht Zellen der Achternachbarschaft, ausgehend von der äußerst rechten, horizontalen Zelle (in Abbildung 4.11 Zelle Nummer 4) sukzessiv für jeden aktuellen Punkt gesucht, bis die Punktverfolgung den gleichen Anfangspunkt (zurück) erreicht hat. Dadurch wird eine Folge von Schritten erzeugt, wobei jeder Schritt eine aus acht möglichen Richtungen repräsentiert. Die Richtungen der Schritte können durch die Zahlen 0 bis 7, z.B. wie in Abbildung 4.11 angegeben ist, dargestellt werden.

Die Punktsuche wird solange fortgesetzt, bis der Anfangspunkt erreicht ist oder das Verfahren wegen eines Fehlers bei dem aktuell betrachteten Konturelement (Lücken) abgebrochen wird. Dabei wird jeder gefundene Punkt als gefunden markiert und somit für den Vorgang der Konturerkennung „unsichtbar“ gemacht. Anschaulich kann der Verfolgungsalgorithmus wie folgt beschrieben (vgl. Beispiel in Abbildung 4.11) werden:

*Ein Betrachter, der ausgehend von einem Anfangspunkt die Konturpunkte verfolgt, sucht jeweils den äußersten rechten, verfügbaren Konturpunkt aus, bis er*

---

<sup>17</sup>Abbildung 3.2

zurück zum Anfangspunkt wiederkehrt, dabei markiert er jeden gefundenen Punkt. Dann sucht er für einen neuen Vorgang einen neuen Anfangspunkt ab dem im vorangegangenen Durchlauf gefundenen Anfangspunkt aus.

a) Konturpunktbild

|   | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 |   | × |   | × |   |
| 1 | × |   | × |   | × |
| 2 | × |   |   |   | × |
| 3 |   | × |   | × |   |
| 4 |   |   | × |   |   |

Das Konturrechteck

↑ Spalte  
→ Zeile

b) Richtungen der Suchschritte

|          |    |          |
|----------|----|----------|
| <u>3</u> | 2↑ | <u>1</u> |
| <u>4</u> | ×  | <u>0</u> |
| <u>5</u> | 6↓ | <u>7</u> |

c) Darstellung der Kontur der Länge 10 durch :

1) Konturcode mit dem Anfangspunkt: p(0,1)

[ 5677112343 ]

2) Ortskoordinatenangabe:

[ (1,0), (2,0), (3,1), (4,2), (3,3), (2,4), (1,4), (0,3), (1,2), (0,1) ]

3) Konturrechteck:

- minimale Zeile: 0, maximale Zeile: 4, minimale Spalte: 0, maximale Spalte: 4

Abbildung 4.11: Darstellung einer Kontur: ein Anwendungsbeispiel

**Konturpunktverkettung:** Die durch den Verfolgungsvorgang gefundenen Konturpunkte müssen miteinander in Beziehung gebracht werden, um zu einer zusammenhängenden Punktfolge zu gelangen, die eine Kontur fehlerfrei und eindeutig repräsentiert. Eine eindeutige Beschreibung einer Objektkontur kann z.B. durch die Angabe einer Zahlenfolge  $c_1, c_2, \dots, c_n$ , mit  $c_i$  als Richtung des gefundenen Konturpunktes erfolgen. Diese Zahlenfolge ist wie folgt zu bestimmen<sup>18</sup>:

Durch die Verfolgung der Konturpunkte in der Achternachbarschaft, ausgehend von einem beliebigen Ausgangspunkt bis zum gleichen Punkt zurück, wird eine Folge von Schritten erzeugt, wobei jeder Schritt in eine von acht möglichen Richtungen zeigt. Die Richtungen der Schritte können durch die Zahlen 0 bis 7 (Abbildung 4.11) angegeben werden. Die erzeugte Zahlenfolge wird *Konturcode* genannt. Diese in der Literatur weit verbreitete Codierungsmethode wird zur Lösung von verschiedenen Problemen, wie die Formanalyse oder die Merkmalextraktion, eingesetzt. Mit Hilfe des Konturcodes kann z.B. die Lage eines Objektes durch die Bestimmung von Eckpunkten spezifiziert werden. Durch eine numerische Manipulation dieser Zahlenfolge können selbst komplexe

<sup>18</sup>vgl. [KlZa92]

Klassifikationsprobleme, ohne dabei auf die Bilddaten greifen zu müssen, in effektiver Weise gelöst werden<sup>19</sup>.

Eine andere Methode zur eindeutigen Darstellung von Objektkonturen ist die Angabe der absoluten Ortskoordinaten der gefundenen Konturpunkte. Durch diese Informationen kann u.a. die Lage (Position) eines Objektes im Bild genau spezifiziert werden. Diese Darstellungsart der Konturen wird hier bevorzugt, da sie für die spätere Rekonstruktion von Grauwertobjekten aus den ermittelten Konturen effektiv ist. Der Konturcode wird aber vollständigheitshalber als ein Element der festgelegten Datenstruktur einer Kontur angegeben<sup>20</sup>.

Um die Menge der durch die Konturbestimmung erkannten Bildstrukturen auf das erforderliche zu reduzieren, werden die verketteten Konturen, deren Länge eine vorgegebene Größe nicht überschreitet, als nicht signifikant für den folgenden Schritt der Objekterkennung eingestuft und somit ignoriert (gelöscht) (*Regel 2: diese Kontur ist geschlossen, aber zu klein*).

Dieser Verkettungsvorgang liefert ein Konturenbild (Abbildung 4.5), deren Punkte in die Klasse der global miteinander *verketteten Konturpunkte*, in die Klasse der *Objektuntergrundpunkte* und in die Klasse der *Hintergrundpunkte* unterteilt sind.

**Lückenschließen:** Falls in der Achternachbarschaft des *aktuellen Konturpunktes* kein Nachfolgepunkt gefunden wird, ist der Fall einer Lücke erkannt, und eine Behandlung dieses Lückenproblems wird erforderlich, um zu einer zusammenhängenden, geschlossenen Kontur zu gelangen. Der Algorithmus zur Behandlung des Lückenproblems ist im folgenden zusammengefaßt:

Falls die bisher verkettete Konturlinie, die jetzt durch eine Lücke behaftet ist, eine vorgegebene Größe nicht übersteigt, wird sie als zu klein klassifiziert und damit als nicht signifikant für den folgenden Objekterkennungsschritt eingestuft, und somit ignoriert (gelöscht). Falls aber die verkettete Konturpunktlinie als signifikant eingestuft wird, wird die Suche nach einem Konturpunkt spiralförmig in den nächsten *drei Nachbarschaften*, der 24er-Nachbarschaft, der 48er-Nachbarschaft und der 80er-Nachbarschaft, wie Abbildung 4.12 zeigt, fortgesetzt. Dabei kann die Größe der globalen Suchnachbarschaft durch einen Steuerparameter festgelegt werden. Hier soll die Effektivität (Rechenzeit) des Algorithmus miteinbezogen werden. Wenn ein Konturpunkt, im weiteren *Anschlußpunkt* genannt, in dieser spiralförmigen Umgebung gefunden wird, dann wird der *aktuelle Konturpunkt* (dies ist der Punkt, für den kein Folgepunkt in der Achternachbarschaft gefunden wurde) mit dem *Anschlußpunkt* durch eine Linie verbunden (Kontur wird geschlossen).

Falls aber diese Linie viele andere Konturpunkte überquert, dann wird dieser Fall als *Konturfehler*, d.h. das Konturverfolgungsverfahren hat falsche Verkettungswege durchlaufen, erkannt und damit wird diese Kontur als nicht signifikant eingestuft und ignoriert (*Regel 3: die Kontur ist „groß“ und weist eine kleine Lücke auf, deren Schließen eine Linie hervorruft, die viele andere Konturpunkte überquert*).

Dann wird weiter mittels der oben beschriebenen lokalen Suchstrategie (in der Ach-

---

<sup>19</sup>für Details vgl. [KlZa92]

<sup>20</sup>vgl. Kapitel 4.5.3

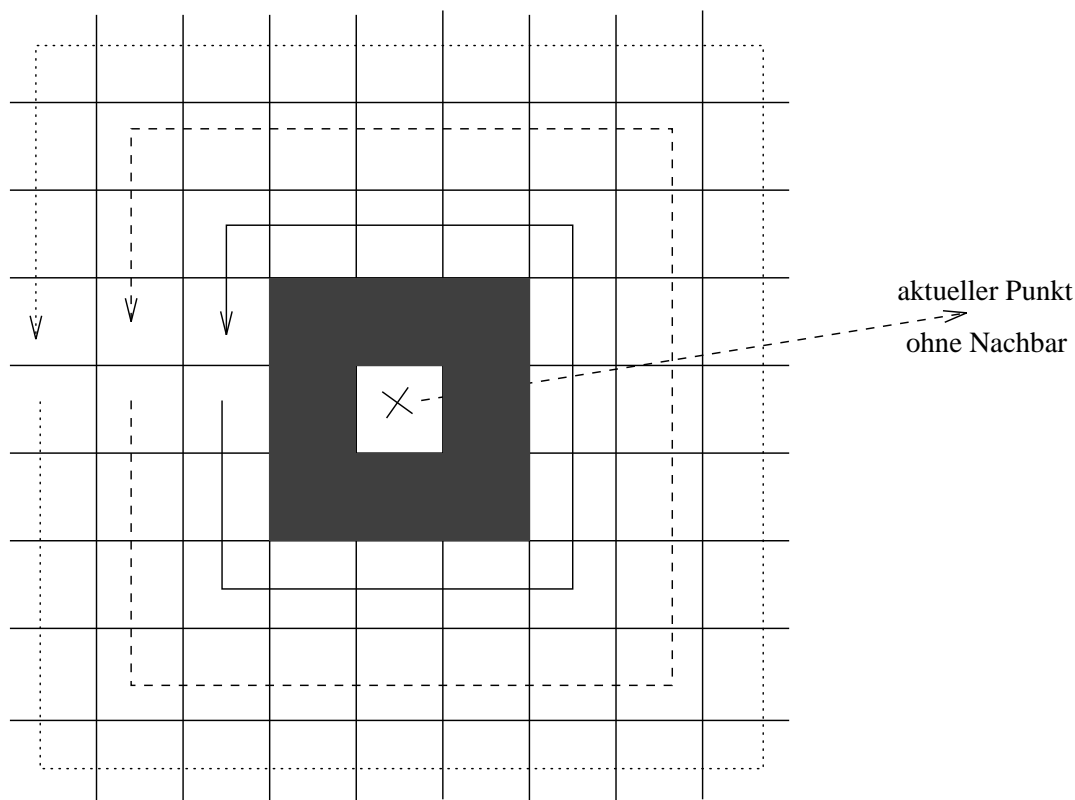


Abbildung 4.12: Globale Suchstrategie zur Schließung von Lücken

ternachbarschaft) nach einem Folgepunkt für den gefundenen *Anschlußpunkt* gesucht. Und somit wird der Verfolgungs- und Verkettungsvorgang fortgesetzt.

Falls aber in den festgelegten drei Nachbarschaften kein Folgepunkt gefunden wird (Lücke ist zu groß), wird der *aktuelle Konturpunkt* mit dem *Anfangspunkt* durch eine Linie verbunden und somit wird die Kontur geschlossen. Falls aber die bisher (bis zu dem aktuellen Punkt) verkettete Konturpunktlinie eine vorgegebene Größe nicht überschreitet (*Regel 4: diese Kontur weist eine große Lücke auf und ist klein*), oder aber der Abstand zwischen dem *aktuellen Konturpunkt* und dem *Anfangspunkt* eine vorgegebene Größe überschreitet (*Regel 5: diese Kontur weist eine große Lücke auf, deren Schließen eine Linie „quer“ durch das Bild hervorruft*), dann werden diese zwei Fälle als *Konturfehler*<sup>21</sup> angesehen, und diese verkettete Konturpunktlinie wird deshalb ignoriert (gelöscht).

### 4.5.3 Algorithmische Aspekte des Verfahrens

Im folgenden werden einige algorithmische Punkte der gewählten Lösungsansätze zur Konturerkennung bei der Implementierung des im Rahmen dieser Arbeit entwickelten *konturorientierten Objekterkennungsalgorithmus* in dem Simulationsprogramm *MORPHOBJ*<sup>22</sup> kurz erläutert:

<sup>21</sup>vgl. Regel 3

<sup>22</sup>vgl. Kapitel 6.2

- Es wird angenommen, daß die gesuchten Objekte bzw. Konturpunkte sich von den anderen Punkten durch einen klaren Bildwert unterscheiden (starker Kontrast).
- Die Konturpunktverfolgung erfolgt ausgehend vom Anfangspunkt, zurück zum Anfangspunkt rekursiv. Jeder gefundene Konturpunkt wird so markiert, daß er für den Vorgang der Konturerkennung und den Vorgang der Objektsuche unerkennbar gemacht wird.
- Für die Anwendung von Konturerkennungsmethoden sind neben den in der „C“ Programmiersprache vordefinierten Datentypen weitere Datentypen notwendig. Grundlegend ist die Darstellung von Konturen. Hierfür ist der Datentyp Kontur definiert. Die Datenrepräsentation für diesen Datentyp ist in der folgenden Datenstruktur festgelegt:

Eine **Kontur** ist ein **Verbund** von:

*Nummer* : INT

*Länge* : INT

*Grauwert* : INT

*Startpunkt* : POINT

*minimale Zeile* : INT

*maximale Zeile* : INT

*minimale Spalte* : INT

*maximale Spalte* : INT

*Konturcode* : **Feld** aus INT

*Konturkoordinaten* : **Feld** aus INT

Die *Nummer* bezeichnet eine aufsteigende Nummerierung der erkannten Konturen. Die *Länge* gibt die Anzahl der zu einer Kontur verketteten Punkte an. Der *Grauwert* der verketteten Punkte wird für jede Kontur neu berechnet. Die Komponente *Startpunkt* bezeichnet den Anfangspunkt einer Kontur. Die folgenden vier Komponenten (*minimale Zeile*, *maximale Zeile*, *minimale Spalte* und *maximale Spalte*) geben die Positionen der vier Eckpunkte der Kontur an. Durch diese vier Angaben ist eine Objektkontur in ein Rechteck (*Konturrechteck*) isoliert. In einem Rechteck können Konturen mehrerer Objekte eingeschlossen sein, aus diesem Grund wird jeder Kontur bei dem Verkettungsverfahren ein eindeutiger Grauwert zugeordnet. Die Komponente *Konturcode* gibt die Richtung<sup>23</sup> der ermittelten Konturpunkte an. Eine Kontur wird auch durch die Komponente *Konturkoordinaten* eindeutig dargestellt, die die absoluten Ortskoordinaten der verketteten Konturpunkte angibt.

- Mit der Angabe des *Konturrechteckes*, dem *Konturpunktgrauwert* und der *Konturkoordinaten* wird ein Objekt eindeutig isoliert und für den letzten Schritt der Objekterkennung, die Konturenausfüllung<sup>24</sup>, zur Verfügung gestellt.

---

<sup>23</sup>vgl. Abbildung 4.11

<sup>24</sup>Kapitel 4.6

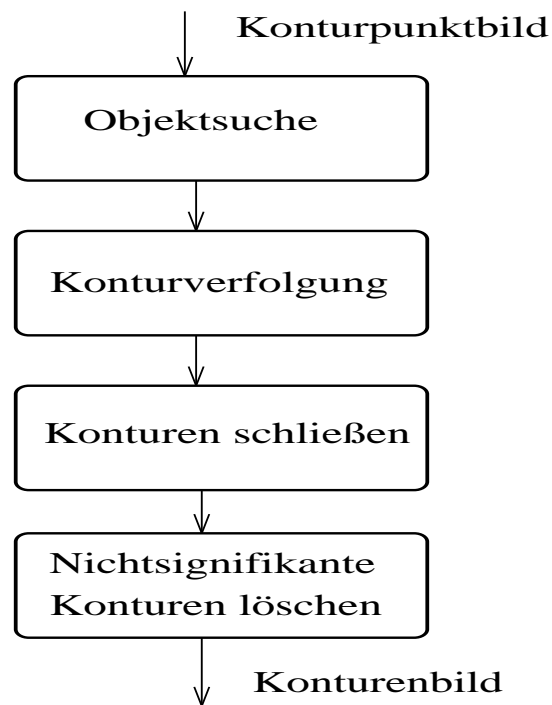


Abbildung 4.13: Flußdiagramm des Konturerkennungsverfahrens

Abbildung 4.13 verdeutlicht die grundlegende Struktur des oben beschriebenen Verfahrens zur Konturerkennung. Abbildung 4.5 zeigt ein Konturen bild, das mittels des vorgestellten Konturerkennungsverfahrens aus Bild 4.4 transformiert worden ist.

Im folgenden sind die Vorteile eines auf Konturpunktverkettung basierenden Verfahrens zusammengestellt:

- Das Verfahren führt immer zu geschlossenen Objektkonturen.
- Diese Methode ist effektiv, da für die Beschreibung einer Kontur nur die Konturpunkte und ihre benachbarten Bildpunkte zur Festlegung des Objektzusammenhanges herangezogen werden.
- Durch den erzeugten Konturcode und durch die Koordinatenangabe lassen sich in einfacher Weise globale Objektmerkmale, insbesondere Formmerkmale, bestimmen.
- Die Rekonstruktion der eliminierten inneren Bildstrukturen (Ausfüllen) kann, anhand der somit gewonnenen Konturinformationen, in einfacher Weise erfolgen.

## 4.6 Konturorientierte Rekonstruktion von Objekten

### 4.6.1 Vorbemerkung

Ziel der vorliegenden Arbeit ist es, einen objektbasierten Ansatz zu entwickeln, der auf erkannten, homogenen Objekten für die Schätzung der Objektbewegung basiert. Die erkannten Informationen über die Objektpunkte werden durch diesen Ansatz, zusammen mit den aus einem *bildpunktbasierten Bewegungsschätzverfahren*<sup>25</sup> gewonnenen Bewegungsvektoren der einzelnen Bildpunkte, verknüpft, um zu einer *besseren Bewegungsschätzung* zu gelangen. Voraussetzung für eine solche *objektbasierte Bewegungsschätzung* ist eine möglichst fehlerfreie, *vollständige* und robuste Objekterkennung. Die in dem resultierenden Konturenbild, das als Ausgangsbild des in Kapitel 4.5 beschriebenen Verfahrens zur Konturerkennung dient, gewonnenen Informationen über die Konturpunkte reichen nicht aus, um die für eine objektbasierte Bewegungsschätzung erforderlichen Informationen über *alle* Punkte der Bildobjekte zur Verfügung zu stellen. Denn die spezifizierten Punkte der Objektkonturen beschreiben nur die Grenze der Bildobjekte, also nur einen kleinen Teil der ursprünglichen Objektpunkte. Um alle Objektpunkte vollständig zu bestimmen, werden *die inneren Punkte* jeder gewonnenen Kontur (diese sind die mittels des Konturextraktionsverfahrens als *Objektuntergrundpunkte*<sup>26</sup> klassifizierten Punkte) bestimmt und mit einem bestimmten Grauwert gefüllt. Somit entstehen aus den Konturen Grauwertobjekte, deren Punkte durch die Angabe ihrer Koordinaten eindeutig definiert sind. Dieser Vorgang der Objektkonturausfüllung wird hier als *Objektrekonstruktion* bezeichnet.

### 4.6.2 Konturenausfüllung

Der Vorgang der Erkennung und der Ausfüllung der inneren Punkte einer Kontur zur Rekonstruktion der im Eingangsbild enthaltenen Strukturen (Objekte) wird, ausgehend von einem gefundenen *inneren Punkt*, rekursiv für alle anderen inneren Punkte der Kontur durchgeführt. Mittels des im folgenden beschriebenen Ansatzes wird jeweils für **jede Kontur** ein *innerer Punkt* bestimmt (vgl. Abbildung 4.15).

1. Finde in dem Konturrechteck einen Konturpunkt mit dem Grauwert der betrachteten Kontur.
2. Suche in der folgenden darunter liegenden Zeile nach einem Konturpunkt mit dem gleichen Grauwert.
3. Suche in der gleichen Zeile, ab der Spalte von Schritt 2, nach einem *Objektuntergrundpunkt*.
4. Überprüfe nach den folgenden drei Regeln (drei Richtungen, vgl. Abbildung 4.15 b), ob dieser gefundene Objektuntergrundpunkt im Inneren der Kontur liegt:

---

<sup>25</sup>vgl. Kapitel 2.2

<sup>26</sup>vgl. Kapitel 4.5.2



- (a) **WENN** ein Konturpunkt mit dem gleichen Konturgrauwert sich horizontal rechts von diesem Objektuntergrundpunkt befindet **und**
- (b) **WENN** ein Konturpunkt mit dem gleichen Konturgrauwert sich vertikal über diesem Objektuntergrundpunkt befindet **und**
- (c) **WENN** ein Konturpunkt mit dem gleichen Konturgrauwert sich vertikal unter diesem Objektuntergrundpunkt befindet,

**DANN** befindet sich der gefundene Objektuntergrundpunkt aus Schritt 3 im Inneren der Kontur.

Ausgehend von diesem inneren Punkt werden dann rekursiv die in der Vierernachbarschaft (vgl. Abbildung 4.15 c) liegenden Bildpunkte mit einem vorgegebenen Grauwert gefüllt, und damit zu *Objektpunkten* erklärt. Die Ausfüllung wird solange durchgeführt, bis die Grenze der Kontur erreicht ist, d.h. bis die Kontur „voll“ ist. Abbildung 4.15 verdeutlicht den Suchvorgang nach einem inneren Punkt in der Kontur, sowie den Füllvorgang.

In vielen Fällen werden durch den Konturerkennungprozeß sich überschneidende Konturen erzeugt (z.B. die Konturen in Abbildung 4.17), so daß die Bestimmung eines inneren Punktes nach dem oben beschriebenen Verfahren das Füllverfahren in ein kleines Gebiet der gesamten Konturfläche führt. In diesem Fall erreicht das rekursive Füllverfahren nach einem kurzen Fülldurchlauf die Konturgrenze und somit wird nur ein kleiner Teil der Fläche der Objektkontur gefüllt. Deshalb wird das Füllverfahren in solchen erkannten Fällen, ausgehend von einem neu zu bestimmenden Objektuntergrundpunkt, solange wiederholt, bis der „größte Teil“ der Konturfläche gefüllt ist. Zur Feststellung, ob das Verfahren sich in ein kleines Gebiet der gesamten Fläche der Kontur verlaufen hat, wird folgende heuristische Regel angewendet:

*Solange die (nicht ausgefüllte) Restfläche des Konturrechteckes größer als die Hälfte der Konturrechteckfläche ist, wiederhole den Füllvorgang, falls ein Objektuntergrundpunkt (innerer Konturpunkt) gefunden wird*<sup>27</sup>

Diese Regel ist, mathematisch vereinfacht, wie folgt zu begründen (Abbildung 4.14): Sei  $K$  eine sich überschneidende Kontur (Abbildung 4.14 a). Beim ersten Fülldurchgang wird ein innerer Punkt in der erste Hälfte der Kontur gefunden. Diese erste Hälfte  $((a \cdot b)/4$ , mit  $a \cdot b$  als die Fläche des Konturrechtecks) wird dann rekursiv gefärbt. Die nicht gefärbte Restfläche  $(3(a \cdot b)/4$ , Abbildung 4.14) ist größer als die Hälfte der Konturrechteckfläche  $((a \cdot b)/2)$ . Deshalb wird ein neuer Durchlauf, ausgehend von einem neu zu bestimmenden inneren Punkt, der in der zweite Hälfte liegt und nach dem oben beschriebenen Verfahren, zu finden ist, gestartet. Nach diesem zweiten Durchlauf ist die gesamte Kontur gefärbt. Die Auswertung der Regel nach diesem zweiten Durchlauf ergibt folgendes Ergebnis:  $(a \cdot b)/2 \not> (a \cdot b)/2$ , d.h. der „größte Teil“ der Konturfläche ist „voll“, deshalb wird der Füllvorgang abgebrochen. In dem Fall nach Abbildung 4.14 wird der Füllvorgang abgebrochen, auch weil sich kein Objektuntergrundpunkt (Innenpunkt) in der Kontur finden läßt.

<sup>27</sup>Falls dies nicht der Fall ist, ist die Kontur vollständig gefüllt. Ein solcher Fall tritt besonders bei schmalen Konturen auf

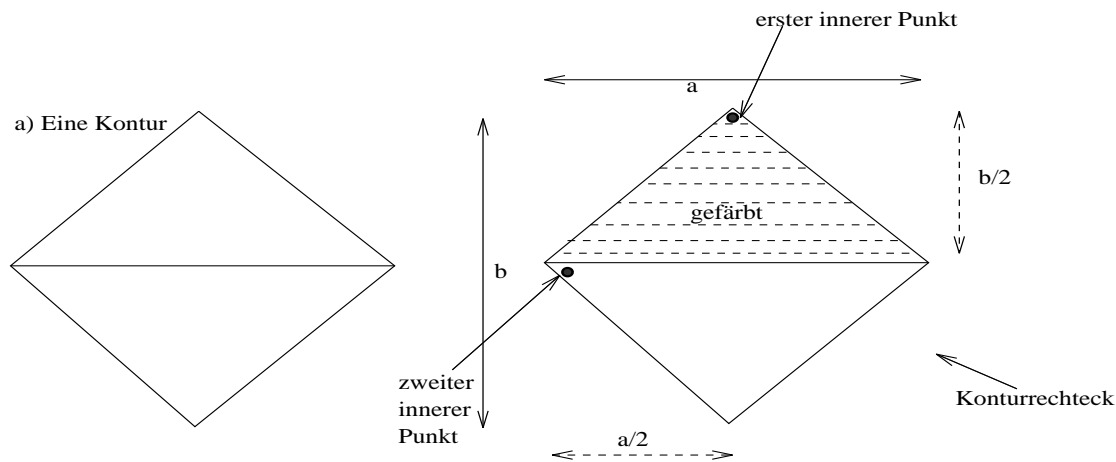


Abbildung 4.14: Ausfüllung von sich überschneidenden Konturen

Für die Repräsentation der Daten, die ein Objekt kennzeichnen, wird für die Implementierung des entwickelten *konturorientierten Objekterkennungsalgorithmus* in dem Simulationsprogramm *MORPHOBJ* folgende Datenstruktur gewählt:

Ein **Objekt** ist ein **Verbund** von:

- Nummer : INT
- Grauwert : INT
- Startpunkt : POINT
- minimale Zeile : INT
- maximale Zeile : INT
- minimale Spalte : INT
- maximale Spalte : INT
- Konturkoordinaten : **Feld** aus INT
- Objektkoordinaten : **Feld** aus INT

Die *Nummer* bezeichnet eine aufsteigende Nummerierung der rekonstruierten Objekte. Der *Grauwert* repräsentiert den für den Füllvorgang einer Kontur bestimmten Grauwert, der für jedes Objekt neu berechnet wird. Die Komponente *Startpunkt* bezeichnet den ersten gefundenen Objektuntergrundpunkt eines Objektes, der zuerst gefüllt wird. Die folgenden vier Komponenten (*minimale Zeile*, *maximale Zeile*, *minimale Spalte* und *maximale Spalte*) geben die Position der vier Eckpunkte eines Objektes an. Durch diese vier Angaben ist ein Objekt in einem Rechteck (*Objektrechteck*) isoliert. Die Kontur des Objektes (seine Grenze) wird durch die Komponente *Konturkoordinaten*

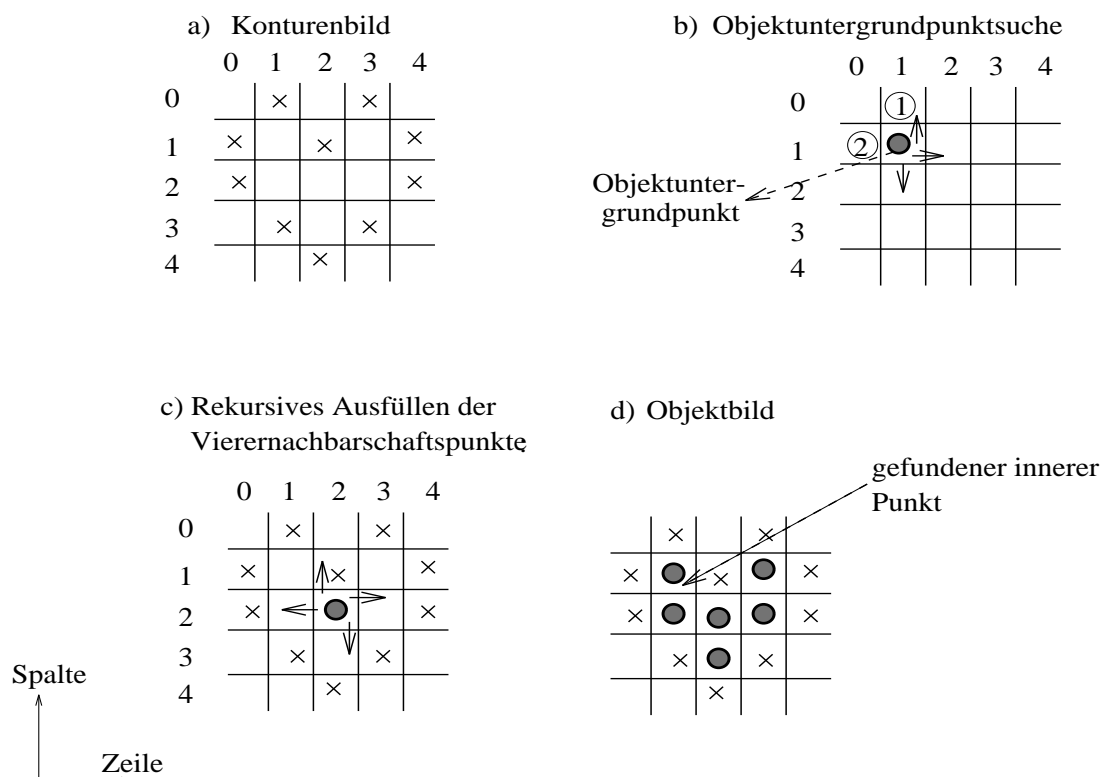


Abbildung 4.15: Suche nach einem inneren Punkt

dargestellt, die die absoluten Ortskoordinaten der verketteten Konturpunkte angibt. Alle Objektpunkte werden durch die Angabe ihrer Ortskoordinaten (die Komponente *Objektkoordinaten*) dargestellt.

Durch den Objekterkennungsalgorithmus wird somit eine Liste der erkannten Objekte erzeugt, die als Grundlage für die in Kapitel 5 vorgestellte *objektbasierte Bewegungsschätzung* dient. Diese Liste steuert gemeinsam mit den in dem Objektbild enthaltenen Klassifizierungsinformationen der Bildpunkte und mit den mittels eines *bildpunkt-basierten Bewegungsschätzer* gewonnenen Bewegungsinformationen anhand eines formulierten Regelsatzes die Bestimmung der neuen Verschiebungsvektoren. Der oben genannte Regelsatz stellt eine Abwägung zwischen den ermittelten Bewegungsinformationen und den erkannten Objektinformationen dar.

Beispiele für Erkennung von Objekten aus einem Halbbild einer realen Szene sind in den Abbildungen 4.2 bis 4.6 und in den Abbildungen 4.16 bis 4.18 verdeutlicht, die mit Hilfe des Simulationsprogramms *MORPHOBJ* erzeugt worden sind. Abbildung 4.19 zeigt eine mit Hilfe des hier entwickelten objektbasierten Bewegungsschätzers<sup>28</sup> ermittelte X-Komponente der Vektorfelder zu der „tussy“-Bildsequenz (Abbildung 4.16). Hier ist deutlich zu sehen, daß Bewegungsvektoren für die erkannten Objekte, die sich bewegen (die Sprecherin), geschätzt worden sind und daß das homogenisierte Vektorfeld durch die Form der erkannten (bewegten) Objekte geprägt ist.

<sup>28</sup>vgl. Kapitel 5



Abbildung 4.16: Ein Originalbild aus der „tussy“-Sequenz  
(Eine Sprecherin bewegt sich vor einem ruhenden Hintergrund)



Abbildung 4.17: Bild von sich überschneidenden Konturen (zu dem Bild 4.16)  
(invertiert)

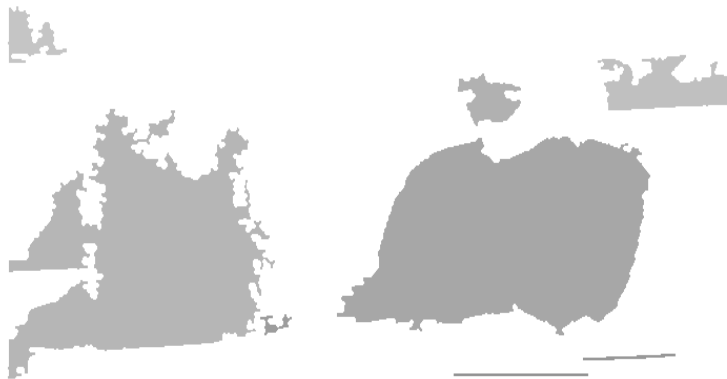


Abbildung 4.18: Die rekonstruierten (erkannten) Objekte aus den Konturen in Bild 4.17 (invertiert)



Abbildung 4.19: Das homogenisierte Vektorfeld zur der „tussy“-Sequenz

# Kapitel 5

## Objektbasierte Bewegungsschätzung

### 5.1 Einleitung und Problemstellung

Eines der spannendsten, aber auch schwierigsten Gebiete der digitalen Bild- und Signalverarbeitung ist die Verarbeitung von Bildsequenzen, deren Einzelbilder durch eine zeitlich periodische Zeilensprungabtastung des, durch eine Kamera projizierten, Bildsignals erzeugt werden. Die Bestimmung von Bewegungsvektoren erlangt für die Verarbeitung von Bildsequenzen eine immer größere Bedeutung, denn für viele Anwendungen ist eine möglichst genaue Bewegungsschätzung notwendig. Die Qualitätsverbesserung der Bildwiedergabe herkömmlicher Fernsehgeräte ist eine dieser Anwendungen, in der, besonders auf der Empfängerseite, keine aufwendigen Maßnahmen zur Qualitätsverbesserung (möglichst mit geringem Hardwareaufwand) erwünscht sind. Eine dieser Maßnahmen ist die Aufwärtskonversion, bei der ein 50Hz Zeilensprungsignal durch eine Zwischenbildinterpolation in ein Signal höherer Bildwiedergabefrequenz (vorzugsweise in ein 100Hz Zeilensprungsignal) umgewandelt wird.

Für die Zwischenbildinterpolation existieren verschiedene Verfahren, die in statische, bewegungsadaptive und bewegungsvektorgestützte Verfahren unterteilt werden<sup>1</sup>. Für leistungsfähige Interpolationsverfahren werden Informationen über die Bewegungen<sup>2</sup>, welche in zwei Halbbildern stattfinden, benötigt. Vektoren beschreiben für jeden Pixel im ersten Halbbild seine vermutliche Verschiebung (Richtung und Betrag der Verschiebung) ins zweite Halbbild. Die benötigten Bewegungsvektoren werden von Bewegungsschätzern bestimmt, die in *bildpunktbasierte* und *objektbasierte* Bewegungsschätzer unterteilt werden. Bei den bekanntesten, bildpunktbasierten Bewegungsschätzverfahren (Blockmatching) werden die Bilder in rechteckige Blöcke unterteilt, ohne dabei die im Bild enthaltenen Strukturen (die Grauwertverteilung) zu berücksichtigen. Für jeden dieser Blöcke wird dann ein Bewegungsvektor zwischen zwei aufeinanderfolgenden Halbbildern ermittelt. Diese blockweise Schätzung der Bewegung geschieht aus Gründen der geringen Hardwarerealisierbarkeit. Diesem Vorteil stehen aber folgende

---

<sup>1</sup>Kapitel 2

<sup>2</sup>[Blum92]

Nachteile<sup>3</sup> gegenüber, die im wesentlichen entstehen, weil innerhalb eines Blockes eine nicht immer vorhandene, konstante Verschiebung vorausgesetzt wird:

- Innerhalb eines Blockes können Teile verschieden bewegter Objekte auftreten. Die Zuordnung **eines** Vektors zu diesem Block führt zu falschen Verschiebungen dieser Objekte.
- Wenn in einem Block Objektteile enthalten sind, die keine Bewegung aufweisen, dann führt ihre Verschiebung zu sehr stark störenden Artefakten.
- Durch die blockweise Vektorzuordnung werden übergreifende Korrelationen mehrerer Blöcke nicht ausgenutzt.
- Die blockweise Vektorzuordnung verursacht in einigen Fällen blockförmige Muster (blocking Artefakte, fixed pattern noise), die sich bei den interpolierten Bildern störend auswirken.

Alle bildpunktbasieren Verfahren werden mit besonderen Problemen konfrontiert, wenn die Bewegung in Grenzbereichen zwischen zwei benachbarten Objekten geschätzt werden soll. Die Schätzung von hinreichend genauen Bewegungsvektoren ist für eine bewegungsvektorgestützte Interpolation besonders wichtig, da durch deutlich falsch geschätzte Bewegungen sehr störende Effekte bei den interpolierten Zwischenbildern auftreten können, wie z.B. das kurzzeitige Zerreißen eines Objektes.

Für einen menschlichen Betrachter beginnt die Analyse eines Bildinhaltes zunächst, bewußt oder unbewußt, durch die Suche nach berandeten und sich unabhängig bewegenden Objekten im Bild. Der Bildinhalt wird hiernach beurteilt. Aus den oben genannten Gründen geht hervor, daß es erfolgversprechend ist, objektbasierte Konzepte zur Bewegungsschätzung zu formulieren. Hier werden Informationen über die im Bild enthaltenen Strukturen zur Schätzung der Bewegung miteinbezogen. Die Objekte treten dann als separate Einheiten anstelle einzelner Bildpunkte bzw. rechteckiger Blöcke auf.

Das Interesse an einer automatischen Bestimmung von Bewegungen, auf der Basis objektbasierter Informationen, wuchs in den letzten Jahren erheblich. Die Grundidee der aus dem Literaturstudium<sup>4</sup> bekannten objektbasierten Verfahren kann wie folgt formuliert werden:

Durch eine Segmentierung der Vektorfelder, die mittels eines bildpunktbasieren Schätzers ermittelt werden, werden zusammenhängende Bildteile erzeugt. Auf der Grundlage dieser zusammenhängenden (vektorhomogenen) Bildteile werden dann die ursprünglichen Vektorfelder verbessert.

Nachteile solcher Verfahren<sup>5</sup> bestehen darin, daß zur Schätzung der Bewegung segmentierte homogene Bildteile vorhanden sein müssen, und für die Erzeugung von gut homogenisierten Bildteilen möglichst genau geschätzte Bewegungsinformationen zur Verfügung stehen müssen. Diese gegenseitige Abhängigkeit führt bei Fehlern in dem einen Vorgang zu erheblichen und irreversiblen Fehlern in dem anderen Vorgang. Durch

---

<sup>3</sup>vgl. [Guse92]

<sup>4</sup>vgl. Kapitel 2.3

<sup>5</sup>vgl. Kapitel 2.3

die Segmentierung der Vektorfelder werden als Grundlage für einen weiteren Schätzvorgang relativ kleine homogene Gebiete geliefert. Diese relativ kleinen Gebiete stellen keine günstige Basis zur objektbasierten Bewegungsschätzung dar, die als Idealziel die Bestimmung der Bewegungen von Bildteilen, die möglichst gut mit den realen Objekten der aufgenommenen Szene übereinstimmen<sup>6</sup>, hat.

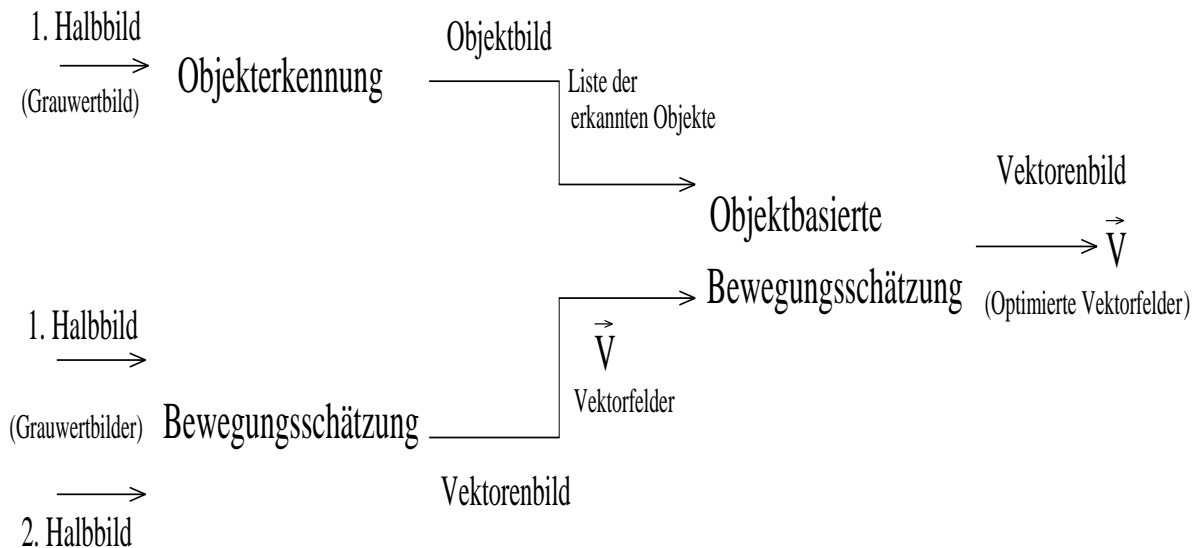


Abbildung 5.1: Blockdiagramm des objektbasierten Bewegungsschätzers

Das Ziel der Untersuchungen im Rahmen dieser Arbeit ist es, Konzepte zur objektbasierten Bewegungsschätzung zu entwickeln, die die Erkennung von beliebig berandeten Bildteilen (Objekten), und die Integration der dadurch gewonnenen Objektinformationen zur Unterstützung und Verbesserung der Bewegungsbestimmung ermöglichen. Die Erkennung von Bildobjekten (bewegt und nicht bewegt) orientiert sich dabei an den im Bild enthaltenen Strukturen, extrahierten Objektmerkmalen. In Kapitel 4 ist ein mehrstufiges Konzept zur Erkennung von Objekten in Bildsequenzen vorgestellt worden. In diesem Kapitel sollen die durch das konturorientierte Objekterkennungsverfahren gewonnenen Informationen als Zusatzinformationen in der Bewegungsschätzung zur Optimierung der Vektorfelder, die mittels eines separaten Bewegungsschätzers berechnet worden sind, herangezogen werden.

Als *objektbasierte Bewegungsschätzung* wird hier ein Vorgang bezeichnet, in dem die Bewegung eines Objektes<sup>7</sup> zwischen zwei Halbbildern einer Bildsequenz, aufgrund

<sup>6</sup>vgl. Kapitel 2.3 und Definition 3.5

<sup>7</sup>vgl. Definition 3.6



erkannter Objektinformationen (Position des Objektes in dem ersten Halbbild), berechnet wird. Die Abbildung 5.1 gibt die grundlegenden Strukturen des hier vorgestellten objektbasierten Bewegungsschätzverfahrens wieder.

## 5.2 Integration von Objektinformationen in die Bewegungsschätzung

### 5.2.1 Begriffsbestimmung

Drei wichtige Eigenschaften von Objekten in Bildsequenzen sind die Position (die Ortskoordinaten der Objektpunkte), die Form und die Bewegung. Durch die Erkennung der Positionen von Objekten in einem Bild, kann die Bewegung dieses Objektes zwischen zwei aufeinanderfolgenden Halbbildern geschätzt werden. In diesem Kapitel wird gezeigt, wie durch Einbeziehung der Objektinformationen in die Bewegungsschätzung die Bestimmung der Verschiebungsvektorfelder verbessert werden kann.

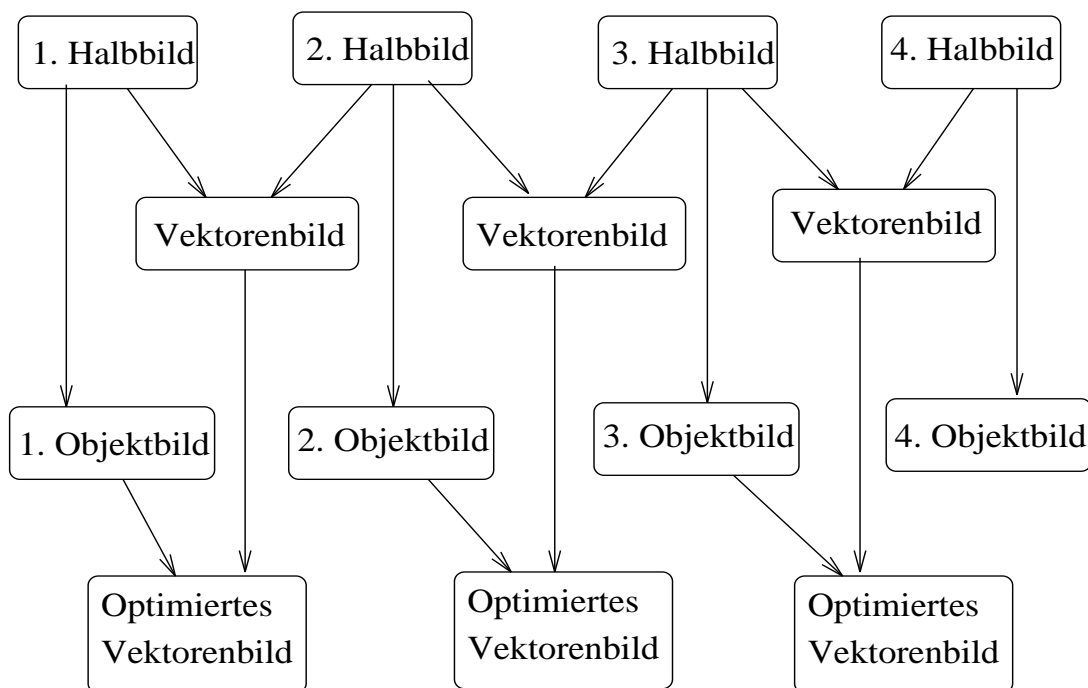


Abbildung 5.2: Integration der Objektinformationen in die Bewegungsschätzung

Die Grundidee des hier entwickelten objektbasierten Bewegungsschätzers verdeutlicht Abbildung 5.2. Zunächst werden zwischen zwei aufeinanderfolgenden Halbbildern einer Sequenz, mit Hilfe eines beliebigen Bewegungsschätzers, die Bewegungsvektoren aller Bildpunkte ermittelt. Die dadurch erzeugten Bewegungskomponenten in x- und y-Richtung, werden in Form einer Bildsequenz getrennt abgelegt. Die erzeugten Bilder

(Vektorenbildsequenz) sind Grauwertbilder, in welchen verschiedene Geschwindigkeiten verschiedenen Grauwerten zugeordnet sind. Dies bedeutet, daß aus zwei Halbbildern ein Grauwertbild (Vektorenbild) erzeugt wird, in dem für jeden Punkt die Bewegung in Form von Grauwertstufen abzulesen ist. Als nächstes werden aus einem Halbbild einer Sequenz, mit Hilfe des im Kapitel 4 vorgestellten Objekterkennungsverfahrens, die Strukturen (Objektpunkte und Nichtobjektpunkte) in diesem Bild extrahiert. Dadurch wird ein Grauwertbild (Objektbild) erzeugt, dessen Punkte anhand ihrer Grauwerte eindeutig entweder als Objektpunkte, oder als Hintergrundpunkte zu erkennen sind. Bei der Objekterkennung wird eine Liste der hierbei entstandenen Objekte erzeugt, die für jedes Objekt u.a. die Position (die absoluten Koordinaten), den Grauwert seiner Punkte und seine Gesamtfläche (Objektfläche) enthält. Diese Liste steuert, gemeinsam mit den in dem Objektbild enthaltenen Klassifizierungsinformationen der Bildpunkte und mit den in dem Vektorenbild eingetragenen Bewegungsinformationen, anhand eines *formulierten Regelsatzes*, die Bestimmung der neuen Verschiebungsvektoren.

Der oben genannte Regelsatz stellt eine Umblendung (Abwägung) zwischen den ermittelten Bewegungsinformationen und den erkannten Objektinformationen dar.

Die Bewegungsvektoren, die zwischen zwei aufeinanderfolgenden Bildern einer, mittels einer Kamera aufgenommenen, Realszene ermittelt werden können, werden in zwei Kategorien unterteilt: in die Kategorie der Bewegungsvektoren der sich bewegten Objektpunkte und in die Kategorie der Bewegungsvektoren der ruhenden Bereiche (Hintergrund).

Ein Hintergrundvektor ist entweder ein Nullvektor, der die „Bewegung“ der ruhenden Bereiche repräsentiert, oder ein Bewegungsvektor, der durch bestimmte Einflüsse (z.B. durch einen Kameraschwenk bzw. durch eine aus anderen Gründen verursachte Hintergrundbewegung) entstanden ist. Zur Bestimmung des *neuen Bewegungsvektors eines Bildpunktes* wird zwischen seiner vorliegenden Bewegungsangabe und zwischen seiner Klassenzugehörigkeit (Nichtobjekt- bzw. Objektpunkt, der zu einem genau isolierten Objekt gehört) umgeblendet. Bei dieser Abwägung und bei der Erstellung der Regeln werden folgende Festlegungen benutzt:

**Ein Objektpunkt** ist ein Punkt, der mittels des Objekterkennungsalgorithmus als ein Teil eines Objektes klassifiziert wird.

**Ein Nichtobjektpunkt** ist ein Punkt, der als Nichtteil eines Objektes klassifiziert wird.

**Ein Hintergrundvektor** ist ein Vektor der zu verbesserenden Vektorfelder ist, der entweder ein Vektor der ruhenden Bereiche (Nullvektor), oder ein Vektor, der durch bestimmte Einflüsse (z.B. durch einen Kameraschwenk) verursacht wird.

**Ein Hintergrundpunkt** ist ein Punkt, dessen Bewegung durch einen Hintergrundvektor gegeben wird.

**Der Majoritätsvektor** eines Objektes ist der Vektor, der innerhalb eines Objektbereiches am häufigsten auftritt. Wenn dieser Vektor nicht der Nullvektor ist, dann wird das Objekt als bewegt erkannt, andernfalls als nicht bewegt. Nicht bewegte Objekte, dies sind erkannte Objekte, deren Majoritätsvektor der Nullvektor

ist, werden weiterhin in dem neuen Algorithmus als nicht bewegt behandelt. Im allgemeinen wird der Majoritätsvektor nach den unten genannten Regeln als der neue Vektor gewählt, der einem erkannten Objekt zugeordnet wird.

**Der alte Bewegungsvektor** ist ein Vektor eines Bildpunktes der zu verbessernden Vektorfelder, die mit Hilfe eines bildpunktbasiereten Bewegungsschätzers (z.B. ein parallel-prädiktives Blockmatching) ermittelt werden.

**Der neue Bewegungsvektor** ist der Vektor eines Bildpunktes, der mit Hilfe des hier vorgestellten Bewegungsschätzers neu berechnet wird, und den alten Vektor ersetzt.

Bei der Abwägung zwischen Objektinformationen und Bewegungsinformationen sind folgende Punkte zu beachten:

1. Homogenisierung der Bewegung eines erkannten Objektes (eine Bewegung/Objekt-Beziehung).
2. Homogenisierung der Hintergrundbewegung (Eine Bewegung/Hintergrund-Beziehung).
3. Ergänzung der Objektpunkte durch Einbeziehung der aus den Vektorfeldern ablesbaren Bewegungen der Nichtobjektpunkte, die von dem Erkennungsalgorithmus als Nichtobjektpunkte klassifiziert werden, aber eine andere Bewegung aufweisen als der Hintergrundvektor.
4. Heuristische Überprüfung des Auftretens anderer Bewegungsformen neben der Translation, wie z.B. der Rotation, oder des Zooms, um sie durch eine Extrabehandlung zu berücksichtigen.

Für eine empfängerseitige Anwendung scheinen parallel-prädiktive Blockmatchingverfahren besonders geeignet zu sein<sup>8</sup>. Deshalb werden geschätzte Vektorfelder, die nach dem in [Blum92] vorgestellten parallel-prädiktiven Blockmatching ermittelt werden, als Ausgangsbasis für eine Optimierung der Bewegungsschätzung, durch die in den Kapiteln 4 und 5 vorgestellten Methoden, verwendet.

Der genannte Algorithmus liefert Vektorfelder, welche sich aus Bewegungsvektorinformationen für die X-Komponente und die Y-Komponente der geschätzten Bewegung zusammensetzt. Jede Bewegungskomponente wird in Form einer Bildsequenz (50Hz Zeilensprungformat) durch Codierung der Geschwindigkeit in Graustufen getrennt abgelegt. Die somit erzeugten Bilder (*Vektorenbilder*) sind Grauwertbilder, in welchen den verschiedenen Geschwindigkeiten verschiedene Grauwerte zugeordnet sind. Ruhige Bereiche in einer Bildsequenz werden durch ein neutrales Grau dargestellt. Bewegungen in positiver Richtung werden helle, Bewegungen in negativer Richtung dunkle Grautöne zugeordnet. *Schnellere* Bewegungen in positiver Richtung werden durch *hellere* Grautöne, *schnellere* Bewegungen in negativer Richtung werden durch *dunklere* Grautöne in den Grauwertbildsequenzen dargestellt.

---

<sup>8</sup>[Blum92]

In den folgenden Abschnitten werden die oben genannten Fälle zur Erstellung eines Regelsatzes, der die Bestimmung der neuen Vektorfelder steuert, behandelt. Abbildung 5.3 gibt einen Gesamtüberblick der objektbasierten Homogenisierung der Vektorfelder.

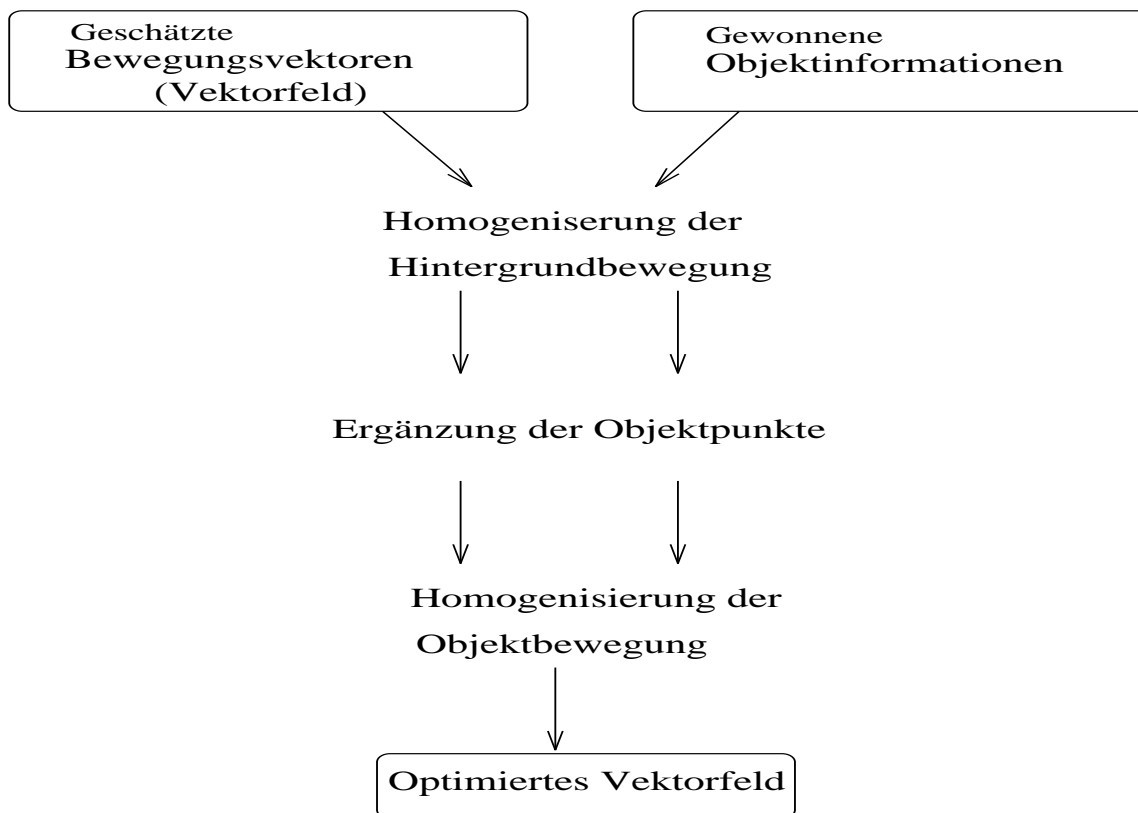


Abbildung 5.3: Objektbasierte Optimierung der Bewegungsvektoren

### 5.2.2 Homogenisierung der Objektbewegung

Hauptziel der objektbasierten Bewegungsschätzung ist die Nutzung von gewonnenen Informationen über die im Bild enthaltenen Objekte zur Ermittlung von einer Bewegung/Objekt-Beziehung. Zum Erreichen dieses Ziels wird hier für jedes Objekt ein Histogramm der im Objektbereich auftretenden Vektoren erstellt und ausgewertet. Aus dem erstellten Histogramm wird der Bewegungsvektor bestimmt, der am häufigsten auftritt (*Majoritätsvektor*). Die Häufigkeiten aller im Objektbereich vorkommenden Vektoren wird zu einer *Vektorenfläche* summiert. Die gesamte *Objektfläche* ist in der Liste der durch die Objekterkennung entstandenen Objekte<sup>9</sup> enthalten. Für **jeden Objektpunkt** (im folgenden „aktueller Punkt“) wird dann der *neue Bewegungsvektor*, durch Abwägung zwischen den Objektinformationen und den vorliegenden Bewegungs-

<sup>9</sup>vgl. Kapitel 4

informationen, anhand der unten aufgestellten Regeln bestimmt. Im folgenden ist ein Regelsatz, mittels dem die oben genannte Umblendung vollzogen wird, angegeben:

- Regel 1:

**WENN** der *alte Bewegungsvektor* des aktuellen Objektpunktes der Nullvektor ist **und** die Vektorenfläche *groß*<sup>10</sup> gegenüber der Objektfläche ist, **DANN** erhält der aktuelle Punkt als *neuen Bewegungsvektor* den Majoritätsvektor.

- Regel 2:

**WENN** der *alte Bewegungsvektor* des aktuellen Objektpunktes der Nullvektor ist **und** die Vektorenfläche *zu klein* gegenüber der Objektfläche ist, **DANN** erhält der aktuelle Punkt den Majoritätsvektor als *neuen Bewegungsvektor*.

- Regel 3:

**WENN** der *alte Bewegungsvektor* des aktuellen Objektpunktes nicht der Nullvektor ist **und** die Differenz zwischen dem alten Bewegungsvektor und dem Majoritätsvektor *klein* ist, **DANN** erhält der aktuelle Punkt den Majoritätsvektor als *neuen Bewegungsvektor*.

- Regel 4:

**WENN** der *alte Bewegungsvektor* des aktuellen Objektpunktes nicht der Nullvektor ist **und** die Differenz zwischen dem alten Bewegungsvektor und dem Majoritätsvektor *groß* ist **sowie** die Fläche des alten Bewegungsvektors *klein* gegenüber der Fläche des Majoritätsvektors ist, **DANN** erhält der aktuelle Punkt den Majoritätsvektor als *neuen Bewegungsvektor*.

- Regel 5:

In den anderen möglichen Fällen behält der aktuelle Punkt (Objektpunkt) seinen *alten Bewegungsvektor*.

---

<sup>10</sup>Für die Implementierung werden in einer PCF-Datei, in welcher Aufrufargumente eines Programms enthalten sind, die Werte solcher Variablen angegeben. Beispiel hierfür ist: Wenn die Vektorenfläche **mindestens ein Dittel** der Objektfläche ist, dann wird der Majoritätsvektor gewählt.

Wenn der alte Bewegungsvektor eines Objektpunktes der Nullvektor ist, dann gibt es zwei Möglichkeiten.

1. Dieser Punkt gehört zum Objekt, und deshalb sollte sein Vektor nicht der Nullvektor sein oder
2. der Objektpunkt als solcher ist falsch erkannt worden.

Diese zwei Fälle werden anhand der Regeln 1 und 2 überprüft. Regel 1 besagt, daß, wenn die gesamte Vektorenfläche zu klein ist, der Objektinformation mehr Vertrauen geschenkt wird. Bei Regel 2 ist die Vektorenfläche so groß, daß angenommen werden kann, daß der Nullvektor falsch ermittelt worden ist, d.h. der Objektinformation wird auch in diesem Fall mehr Vertrauen geschenkt und somit wird der Majoritätsvektor dem aktuellen Punkt zugeordnet. In allen anderen Fällen wird die Objektinformation als falsch eingestuft und der Punkt behält seinen alten Vektor.

Wenn der alte Vektor des Objektpunktes nicht der Nullvektor ist, dann wird angenommen, daß die Wahrscheinlichkeit groß ist, daß der Objektpunkt als solcher richtig erkannt worden ist. Unter dieser Bedingung wird in Regel 3 zwischen dem alten Bewegungsvektor und dem möglichen neuen Vektor (Majoritätsvektor) abgewogen. Wenn die Differenz zwischen den Vektoren nicht groß ist, wird zu Gunsten des Majoritätsvektors entschieden. Wenn aber die Differenz groß ist, dann wird in Regel 4 die zusätzliche Information über beide Vektoren miteinbezogen.

In Regel 4 wird das Verhältnis beider Flächen, der Fläche des alten Vektors und der Fläche des neuen Vektors (Majoritätsvektor), als Zusatzinformation genommen. Wenn die Fläche des alten Vektors klein gegenüber der Majoritätsfläche ist, wird der alte Vektor mit dem Majoritätsvektor überschrieben. Auf diese Weise wird eine Bewegung/Objekt-Beziehung erzeugt, bei der alle Punkte, die zu einem Objekt eingestuft sind, einen einheitlichen Bewegungsvektor erhalten.

### 5.2.3 Homogenisierung der Hintergrundbewegung

Zur Optimierung der Hintergrundbewegung wird davon ausgegangen, daß die Bewegung eines Hintergrundes einheitlich sein muß. Deshalb wird, durch Einbeziehung von Informationen über die Nichtobjektpunkte und über ihre aus dem zu verbessernden Vektorfeld ableitbaren, zugehörigen Bewegungsvektoren, angestrebt, einen einzigen Vektor als Hintergrundvektor zu ermitteln. Dieses Ziel wird durch das folgende Schema erreicht:

Für alle Nichtobjektpunkte (*Informationen aus den erkannten Positionen der Bildobjekte*) werden die am häufigsten auftretenden Vektoren (Majoritätsvektoren: *Informationen aus dem zu optimierenden Vektorfeld*) durch Auswertung des Histogramms aller Hintergrundvektoren, die innerhalb dieses Gebietes (Nichtobjektpunkte) auftreten, ermittelt. Dann wird anhand folgender Regel zwischen dem ersten Majoritätsvektor des Hintergrundes (in den meisten Fällen ist dieser der Nullvektor) und dem zweiten Majoritätsvektor (der Nichtnullvektor, der z.B. durch einen Kameraschwenk verursacht wird) ausgewählt:

- Regel 6:

**WENN** die Häufigkeit des Nullvektors gegenüber der Häufigkeit des Nichtnullvektors nicht **wesentlich groß** ist,

**DANN** wird der Nichtnullvektor als einheitlicher Hintergrundvektor gewählt.

Hier wird davon ausgegangen, daß die Bildobjekte bei ihrer Bewegung durch das Bild von verschiedenen Einflüssen (z.B. Kameranachwenk) beeinträchtigt werden, so daß der Hintergrundvektor (Vektor der ruhenden Bereiche) nicht immer der Nullvektor sein muß. Durch die oben erstellte Regel wird die Hintergrundinformation vervollständigt. Für den Ausdruck **wesentlich groß** wird hier z.B. folgender heuristischer Wert verwendet: Wenn die Häufigkeit des Nichtnullvektors **mindestens ein Drittel** der Häufigkeit des Nullvektors ist, dann wird der Nichtnullvektor als Hintergrundvektor gewählt.

#### 5.2.4 Ergänzung der Objektpunkte

Falls die Bewegung eines aus der Objekterkennung als Nichtobjektpunkt erkannten Bildpunktes nicht der oben ermittelten Hintergrundbewegung, oder der Nullbewegung entspricht, dann wird davon ausgegangen, daß dieser als Nichtobjektpunkt erkannte Punkt als solcher falsch erkannt ist. Hier tritt der Fall auf, bei dem die Objektinformation und die Bewegungsinformation sich eindeutig widersprechen. In diesem Fall wird folgende Regel angewendet:

- Regel 7:

**WENN** der Bildpunkt als Nichtobjektpunkt erkannt wird **und** sein alter Bewegungsvektor weder mit dem neu ermittelten Hintergrundvektor noch mit dem Nullvektor identisch ist,

**DANN** behält er seinen alten Bewegungsvektor.

Die Bedeutung des Auftretens eines nach Regel 7 möglichen Falls ist, daß der Nichtobjektpunkt, anhand seines alten Bewegungsvektors, als ein Objektpunkt klassifiziert wird, und somit wird die Menge der Objektpunkte ergänzt.

#### 5.2.5 Betrachtung anderer Bewegungsformen

Die oben beschriebenen Regeln sind für ein translatorisches Bewegungsmodell erstellt worden. Für nicht translatorische Bewegungen sind andere Regeln erforderlich. Hier wird, anhand heuristisch entwickelter Regeln, versucht andere Bewegungen zu erkennen. Für diesen Zweck wird das Vektorenhistogramm jedes Objektes ausgewertet. Wenn die Vektoren innerhalb des Objektes *möglichst gleichverteilt* sind, dann wird eine nicht translatorische Bewegung, z.B. Rotation oder Zoom, angenommen (vgl. Abbildung 5.4). Für diese nicht translatorische Bewegung werden folgende Regeln aufgestellt:

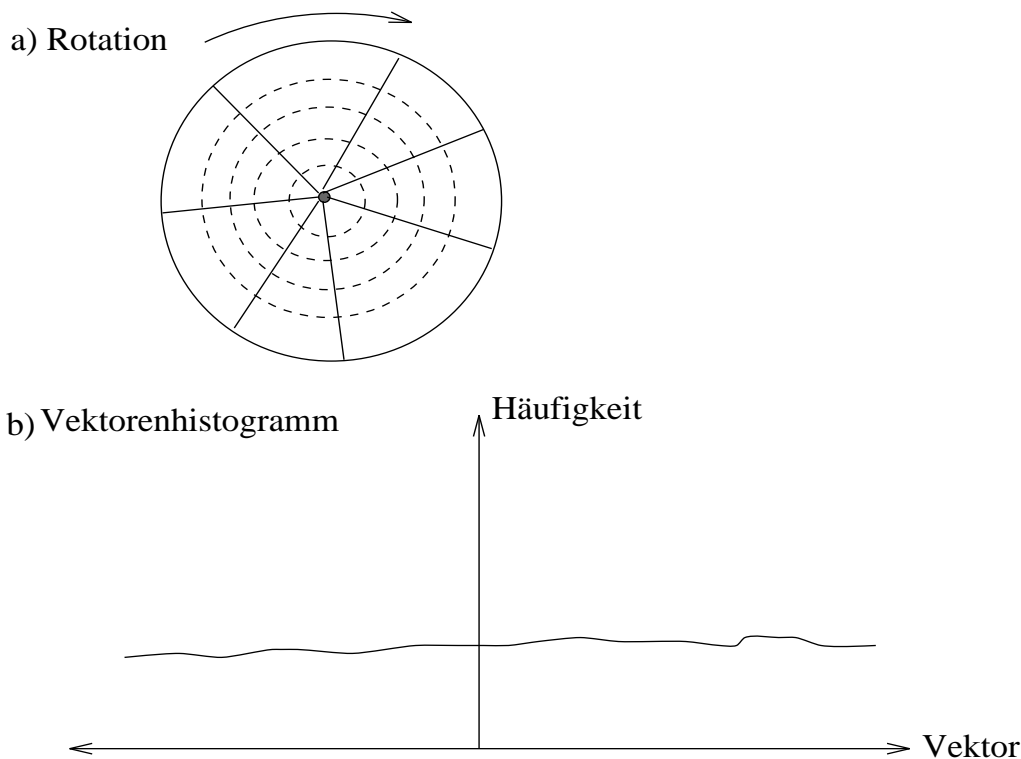


Abbildung 5.4: Beispiel einer Rotation

- Regel 8:

**WENN** der *alte Bewegungsvektor* des aktuellen Objektpunktes nicht der Nullvektor ist **und** die Differenz zwischen dem alten Bewegungsvektor und dem Majoritätsvektor *klein* ist **sowie** die Vektoren innerhalb des Objektes *möglichst gleichverteilt* sind<sup>11</sup>, **DANN** behält der aktuelle Punkt seinen alten Bewegungsvektor<sup>12</sup>.

- Regel 9:

**WENN** der *alte Bewegungsvektor* des aktuellen Objektpunktes nicht der Nullvektor ist **und** die Differenz zwischen dem alten Bewegungsvektor und dem Majoritätsvektor *groß* ist **sowie** die Vektoren innerhalb des Objektes *möglichst gleichverteilt* sind<sup>13</sup>, **DANN** behält der aktuelle Punkt seinen alten Bewegungsvektor<sup>14</sup>.

---

<sup>11</sup>vgl. Regel 3

<sup>12</sup>Eine nicht translatorische Bewegung wird vermutet

<sup>13</sup>vgl. Regel 4

<sup>14</sup>Eine nicht translatorische Bewegung wird vermutet



Abbildung 5.5 verdeutlicht noch einmal die verschiedenen Phasen der Integration der Objektinformationen in die Bewegungsschätzung.

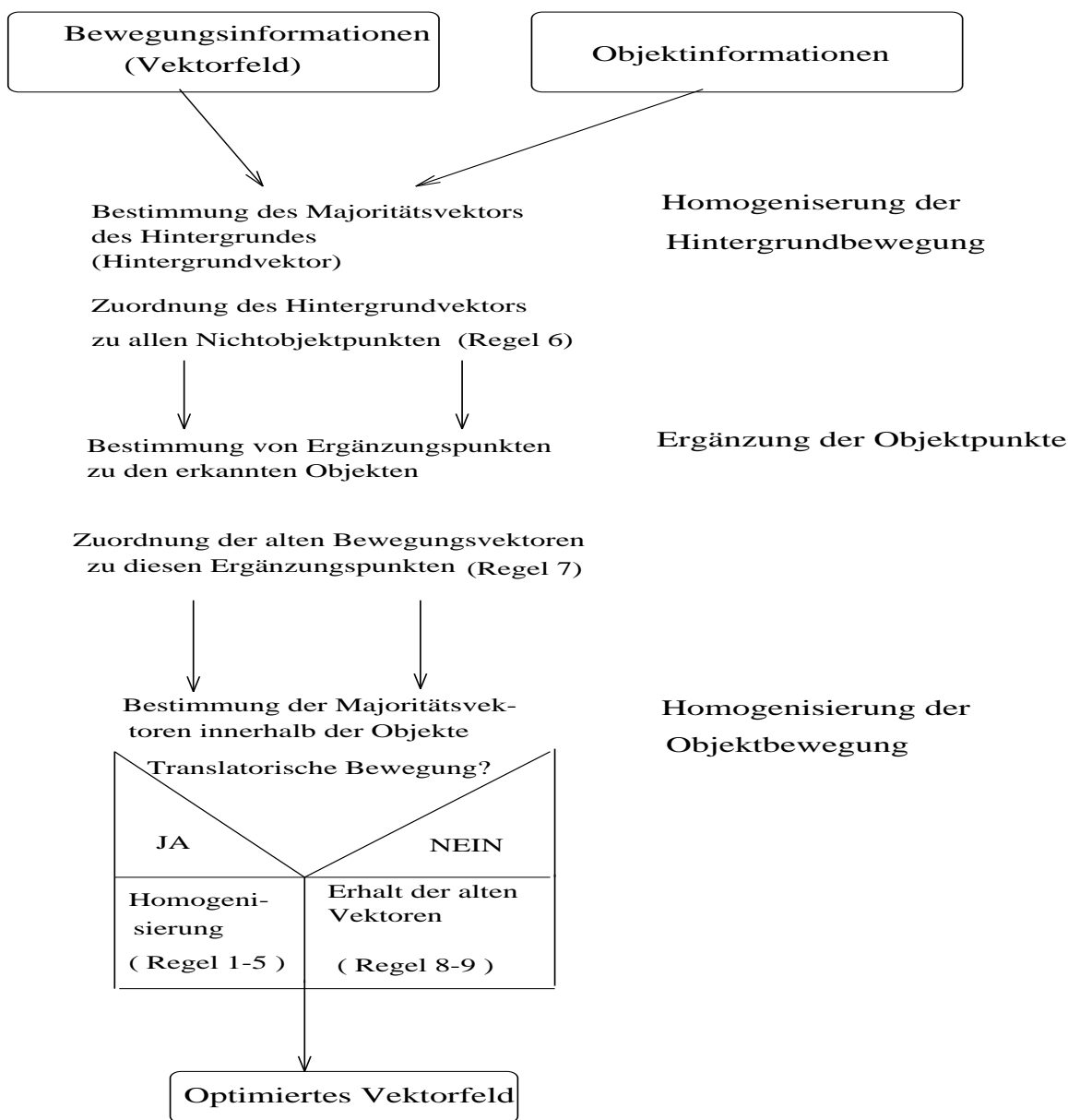


Abbildung 5.5: Flußdiagramm der objektbasierten Homogenisierung der Vektorfelder

# Kapitel 6

## Simulationsergebnisse

### 6.1 Vorbemerkung

In diesem Kapitel werden die Simulationsergebnisse der in Kapitel 4 und 5 vorgestellten Verfahren zusammengefaßt. Der zugrundeliegende Gedanke bei der Entwicklung der Verfahren war, die Bewegungsschätzung für beliebig geformte Bildbereiche durchzuführen. Eine wichtige Bedingung ist dabei die Verwendbarkeit des Algorithmus bei der empfängerseitigen Bewegungsschätzung von Bildsequenzen. Dies bedeutet, daß die Entwicklung von wenigen, kleinen und nicht aufwendigen Modulen im Vordergrund steht, denn die Realisierung bei einer empfängerseitigen Qualitätsverbesserung orientiert sich daran, ob sie mit möglichst wenig Chips auf einer Zusatzkarte des Endgerätes<sup>1</sup> zu implementieren ist. Eine andere wichtige Bedingung geht aus dem Prinzip des hier entwickelten objektbasierten Bewegungsschätzers hervor.

Im Rahmen dieser Arbeit wurden Möglichkeiten und Grenzen einer objektbasierten Bewegungsschätzung untersucht. Als ein erster Schritt von einem „Blockmatching“ zu einem „Objektmatching“ wurde ein Konzept entwickelt, bei dem die mit Hilfe eines auf der mathematischen Morphologie basierten Verfahrens zur Objekterkennung gewonnenen Objektinformationen zur Unterstützung und Verbesserung von bereits mit Hilfe eines bildpunktbasieren Bewegungsschätzers ermittelten Vektorfeldern eingesetzt werden. Aus diesem Prinzip geht hervor, daß der Aufwand der Objektinformationengewinnung und der Aufwand der Integrationen dieser Informationen im Vergleich zu dem Aufwand des gewählten bildpunktbasieren Bewegungsschätzers gering gehalten werden soll. Diese Erkenntnisse werden durch die im folgenden dargestellten Simulationsergebnisse verdeutlicht.

### 6.2 Das Simulationsprogramm MORPHOBJ zur Erkennung von Objekten

Die schrittweise Erkennung von Objekten aus Bildern, die nach dem in Kapitel 4 vorgestellten Verfahren arbeitet, kann durch den Einsatz von Steuerparametern beeinflusst

---

<sup>1</sup>[Blum92]

und damit der Problemstellung angepaßt werden. Das Simulationsprogramm MORPHOBJ erkennt nach dem in Kapitel 4 beschriebenen, auf morphologischen Grundoperationen basierten Verfahren zur Objekterkennung homogene Strukturen (Objekte) in den Halbbildern einer Sequenz. Der Aufruf des Simulationsprogramms MORPHOBJ hat folgende Syntax:

*morphobj* <PCF-Datei für die Erkennung von Objekten>

Als Aufrufargument ist eine *PCF-Datei* einzugeben, welche folgende Informationen enthält:

- Angaben über die zu bearbeitende Bildsequenz.
- Informationen zu der morphologischen Konturpunktdetektion (Angaben zu den gewünschten Strukturelementen).
- Den Schwellwert zur Trennung von Objektpunkten aus dem Hintergrund.
- Parameter zur Steuerung der Konturextraktion<sup>2</sup>. Dies sind: die minimale Konturlänge, die minimale Länge einer nicht geschlossenen Kontur, die eliminiert werden kann, der maximal erlaubte Abstand zwischen zwei zu verbindenden Punkten (Lückenschließen) einer nicht geschlossenen Linie, die maximale Anzahl von Konturpunkten, die eine Linie, bei Verbindung zweier Punkte (Lückenschließen), überqueren kann, ohne diesen Fall als Konturfehler einzustufen und die Größe der globalen Nachbarschaft zur Suche eines Anschlußpunktes bei möglichen Lücken.
- Wegen der in Kapitel 4.6.2 beschriebenen Konturüberschneidungen können bei der Konturausfüllung kleine gefüllte Gebiete entstehen, die für die Bewegungsschätzung nicht signifikant sind. Dies kann durch die Angabe einer minimalen Fläche, die ein rekonstruiertes Objekt haben kann, gesteuert werden.

Das Programm MORPHOBJ erzeugt aus einer realen Bildsequenz (50Hz Zeilensprungformat) eine andere Bildsequenz (auch 50Hz Zeilensprungformat), deren Grauwertbilder in zusammenhängende, unabhängige, homogene Objekte unterteilt sind, wobei jedes Objekt einen anderen Grauwert besitzt. Die extrahierten Informationen aller Objekte sind in einer Liste verfügbar. Mit Hilfe dieser Liste werden Zugriffe auf die Bilddateien vermieden und somit wird der Aufwand erheblich reduziert. In Kapitel 6.4 werden Ergebnisse über die Rechenzeiten der Module des entwickelten Algorithmus angegeben, die den oben genannten geringen Aufwand verdeutlichen. Als Beispiel für eine Objekterkennung sei die in Abbildung 4.2 gezeigte Szene gegeben:

*Ein Auto fährt von rechts nach links.*

*Dabei wird es mit einem Keraschwenk in gleicher Richtung verfolgt.*

Das Programm MORPHOBJ liefert die in Abbildung 4.6 dargestellten, erkannten Objekte. Hier sind die wesentlichen, für eine weitere Verarbeitung (z.B. Bewegungsschätzung) signifikanten Objekte, die den Definitionen 3.5 und 3.6 (vgl. Kapitel

---

<sup>2</sup>Diese Informationen sind ausführlich in Kapitel 4.5 beschrieben

3.1) entsprechen, aus dem Originalbild extrahiert. Die Abbildungen 6.3–6.10 zeigen mehrere Anwendungsbeispiele des Programms MORPHOBJ.

In Kapitel 6.3 ist das Simulationsprogramm OBJMOTION zur objektbasierten Bewegungsschätzung erläutert. Das Programm MORPHOBJ ist eine Teilmenge des Programms OBJMOTION. Da aber Objekterkennung auch für andere Anwendungen als die Bewegungsschätzung (z.B. Codierung von Bildsequenzen) von Interesse sein kann, sind zwei unabhängige Systeme, einmal für die Erkennung von Objekten und zum anderen für die Bewegungsschätzung, entstanden.

## 6.3 Das Simulationsprogramm OBJMOTION

Wie bereits mehrfach erwähnt, eignet sich für eine empfängerseitige Videosignalanwendung das parallel-prädiktive Blockmatching am besten<sup>3</sup>. Für Simulationszwecke des hier entwickelten objektbasierten Bewegungsschätzverfahrens werden deshalb die geschätzten Vektorfelder des Simulationsprogramms REBLOMAT, das Bewegungsvektoren nach dem in [Blum92] vorgestellten parallel-prädiktiven Blockmatching ermittelt, als Ausgangsbasis für eine Integration von Objektinformationen zur Verbesserung der Vektorfelder genommen. In Abbildung 6.1 ist die X-Komponente der mit Hilfe des Simulationsprogramms REBLOMAT ermittelten Vektorfelder gezeigt.



Abbildung 6.1: Das zu optimierende Vektorfeld (X-Komponente)

---

<sup>3</sup>[Blum92]

Das Programm OBJMOTION optimiert, nach dem in Kapitel 5 vorgestellten Verfahren, Bewegungsvektorfelder zu Bildsequenzen, die mittels eines beliebigen Bewegungsschätzers ermittelt sind. Der Aufruf des Simulationsprogramms OBJMOTION hat folgende Syntax:

```
objmotion <PCF-Datei für die Bildsequenz> <PCF-Datei für die Objekterkennung>
          <PCF-Datei für die Integration>
```

Im *ersten Aufrufargument* ist eine *PCF-Datei* einzugeben, welche Informationen über die zu schätzende Bildsequenz und über die zu optimierenden Vektorfelder (Vektorenbilder) enthält. Mit dem *zweiten Aufrufargument* wird eine *PCF-Datei* angegeben, in welcher die Steuerparameter für die Erkennung von Objekten aus den Bildern der realen Sequenz abgelegt sind. Diese Informationen sind in Kapitel 6.2 erläutert.

Als *drittes Aufrufargument* ist eine *PCF-Datei* einzugeben, welche Informationen, die, zur Verbesserung der Bewegungsinformation eines Bildpunktes, eine Umblendung zwischen ihrer alten Bewegungsinformation und ihrer erkannten Klassenzugehörigkeit darstellen, enthält. Diese Verbesserung und Unterstützung der Bewegungsschätzung wird, wie in Kapitel 5 ausführlich beschrieben, durch eine Gewichtung der in den Regeln auftretenden linguistischen Variablen, wie *kleine Fläche*, *möglichst gleichverteilte Vektoren* usw., modifiziert und damit der Problemstellung angepaßt<sup>4</sup>.

Durch den Einsatz von PCF-Dateien entfällt bei Veränderung der verschiedenen Parameter die Neucompilierung des Programms. Es ist dadurch auch möglich, Änderungen leicht vorzunehmen und mehrere Simulationen mit unterschiedlichen Parametern zum Testen der Auswirkung, z.B. zur Gewichtung der linguistischen Variablen in den oben genannten Regeln, durchzuführen, um auch eine Optimaleinstellung der Parameter zu erreichen.

Das Ergebnis der Verarbeitung von Bildsequenzen durch das Programm OBJMOTION ist ein Bewegungsvektorfeld, welches sich, wie beim Simulationsprogramm REBLOMAT, aus Bewegungsvektoren für die X-Komponente und die Y-Komponente der geschätzten Bewegungen zusammensetzt. Die X-Komponente, sowie die Y-Komponente werden in Form von Bildsequenzen (50Hz Zeilensprung) abgelegt. Den verschiedenen ermittelten Geschwindigkeiten werden verschiedene Grauwerte zugeordnet. Durch diese Zuordnung entstehen Grauwertbilder, die für jede Komponente als Sequenz zusammengelegt werden. Bei dem Blockmatching-Programm REBLOMAT werden Geschwindigkeiten in einem Bereich von -16 Pixel/Halbbild bis +15 Pixel/Halbbild ermittelt. Wenn größere bzw. kleinere Geschwindigkeiten auftreten, dann werden sie auf die nächste darstellbare Geschwindigkeitsstufe begrenzt. Denn Objekte mit solchen Bewegungen sind in den meisten Fällen ohnehin unscharf aufgenommen. Durch die gewählte Bewegungsgrenze ergeben sich bei einer Grauwert-/Geschwindigkeits-Zuordnung 32 unterschiedliche Grautöne. Die Darstellung der Bewegung durch Grauwerte ist hier, wie in dem REBLOMAT-Programm, angegeben.

---

<sup>4</sup>für eine genaue Beschreibung dieser Informationen vgl. die in Kapitel 5 aufgestellten Regeln 1-9

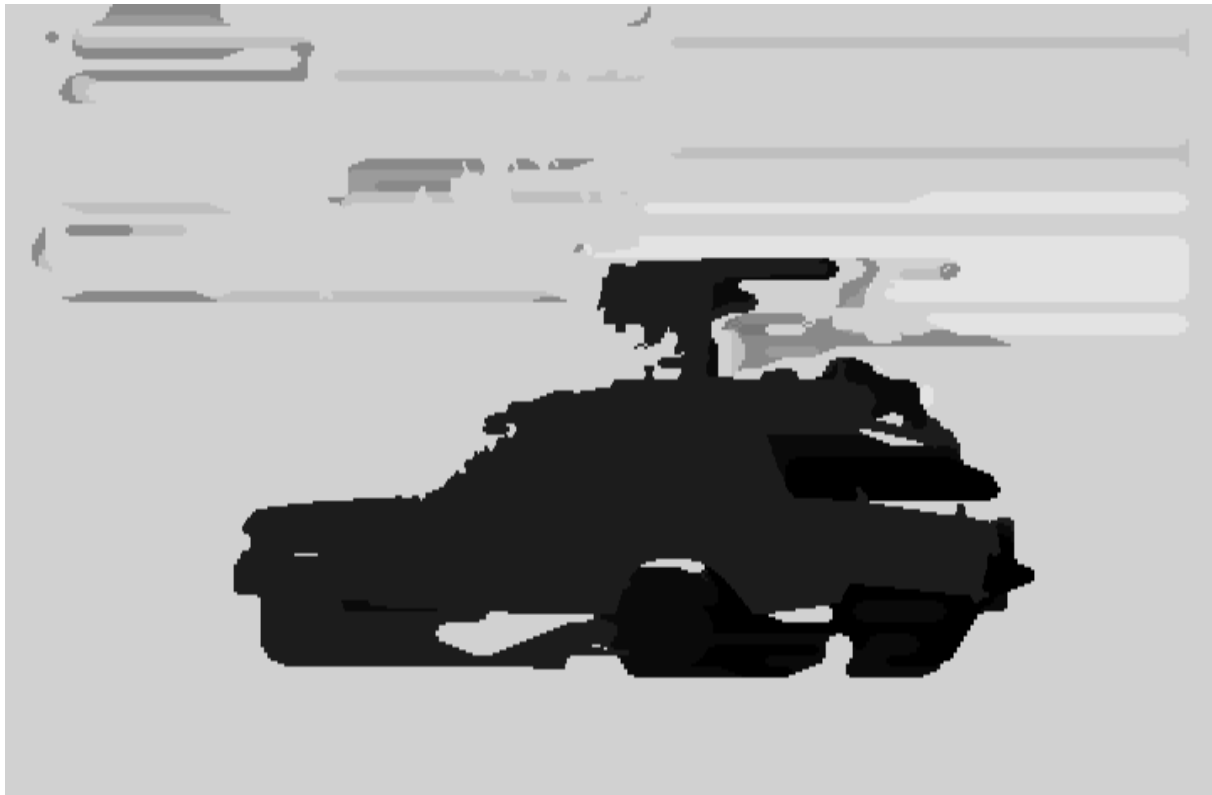


Abbildung 6.2: Resultierendes Vektorfeld des objektbasierten Bewegungsschätzers (X-Komponente)

Für eine reale Bewegungsschätzung ist als Beispiel die in Abbildung 4.2 dargestellte Szene gegeben:

*Ein Auto fährt von rechts nach links. Dabei wird es mit einem Kameranachweiser in gleicher Richtung verfolgt. Während das Auto sich langsam nach links bewegt, bewegt sich der Hintergrund wegen des Kameranachweisers schnell nach rechts.*

In Abbildung 6.2 ist die X-Komponente des, mit Hilfe des Programms OBJMOTION zu der oben beschriebenen Sequenz erzeugten, Vektorfeldes gezeigt. Deutlich zu sehen ist, daß das homogenisierte Vektorfeld, wegen der Einbeziehung von Objektinformationen, im wesentlichen durch die Form der erkannten (bewegten) Objekte geprägt ist. Ein anderes Beispiel von objektbasierter Homogenisierung von Vektorfeldern verdeutlicht Abbildung 4.19.

## 6.4 Vergleichende Ergebnisse

Das vorgestellte objektbasierte Verfahren zur Bewegungsschätzung wird von den in Kapitel 6.2 und 6.3 beschriebenen Parametern beeinflusst. Wichtig sind vor allem die Wahl des Schwellwertes und die Wahl der Konturparameter, wie z.B. die minimale Konturlänge. Das Ergebnis der Laufzeit ist *im allgemeinen* abhängig von der Anzahl

und Größe der Objekte, die im Bild enthalten sind und von dem Erkennungsverfahren lokalisiert werden, d.h. die Struktur des Bildes spielt eine wichtige Rolle.

Bei einem Vergleich zwischen der Rechenzeit, die das Simulationsprogramm REBLOMAT zur Erzeugung eines Vektorfeldes zwischen zwei Bildern der „car“-Sequenz (Abbildung 4.2) nach dem in [Blum92] vorgestellten Blockmatchingverfahren verbraucht, und der Rechenzeit, die das Simulationsprogramm OBJMOTION nach dem hier vorgestellten objektbasierten Verfahren zur Optimierung dieses Vektorfeldes benötigt, hat sich folgendes Ergebnis herausgestellt:

Das Programm REBLOMAT benötigt zur Schätzung der Bewegung zwischen zwei Halbbildern auf einer *Sun4-Workstation mit einem Sparc2-Prozessor* ca. 209 Sekunden. Diese Schätzung ist mit 22 Kandidatenvektoren pro Block, mit sogenannten „wide blocks“, und mit einer zweistufigen Blockerosion<sup>5</sup> vollzogen worden. Das Programm OBJMOTION benötigt zur Optimierung dieser Vektorfelder auf derselben Maschine, mit den in Kapitel 6.3 erläuterten Parametern, ca. 15 Sekunden. Der Zeitbedarf beider Verfahren bleibt über die Sequenz konstant. Für andere Sequenzen ergeben sich kleine Abweichungen.

Nach den in Kapitel 6.1 erläuterten Bedingungen zum Einsatz von Bewegungsschätzern für eine empfängerseitige Anwendung, scheint der Aufwand der vorgestellten, objektbasierten Bewegungsschätzung (Optimierung der Vektorfelder, d.h. Gewinnung von Objektinformationen und Integration dieser Informationen in die Bewegungsschätzung), im Vergleich zu dem Aufwand des gewählten bildpunkt-basierten Bewegungsschätzers, gut vertretbar. Bei einer möglichen Hardwarerealisierung des Algorithmus können die Objekterkennung und die bildpunkt-basierte Bewegungsschätzung parallel durchgeführt werden (vgl. Abbildung 5.1). Da die Objekterkennung (MORPH-OBJ) wesentlich schneller als die Bewegungsschätzung (REBLOMAT) ist (1:20), sind nur für die Integration der Objektinformationen in die Bewegungsschätzung zusätzliche Operationen im zeitkritischen Verarbeitungspfad erforderlich.

Dieser geringe Aufwand des entwickelten Verfahrens begründet sich durch die im folgenden beschriebenen Eigenschaften. Die morphologische Konturpunktdetektion nach Definition 4.1 liefert *gut positionierte* und sehr *dünne* Konturelemente. Der morphologische Detektor weist auch ein *gutes Rauschverhalten* auf. Diese Eigenschaften ermöglichen eine schnelle und robuste Konturerkennung, die eindeutig *lokalisierte und unabhängige* Konturen liefert. Die anschließende Objektrekonstruktion kann durch diese genaue Lokalisation der Konturen und durch die Erzeugung von *Konturlisten* in schneller und einfacher Weise erfolgen. Diese Listen repräsentieren die ermittelten Konturen und stellen ihre charakteristischen Daten zur Verfügung. Die Objektrekonstruktion wird dadurch besonders effizient, weil es anhand der erzeugten Konturlisten *nicht erforderlich* wird, auf die *Bilddaten zuzugreifen*.

Die darauf folgende Optimierung der Vektorfelder erfolgt durch die Verwendung von Operatoren, die wegen der *erzeugten Objektliste* selten auf die *Bilddaten zugreifen*. Denn diese Listen beinhalten die erforderlichen Objektinformationen, die für die objektbasierte Optimierung der Vektorfelder erforderlich sind. Zwei andere wichtige Eigenschaften des hier vorgestellten objektbasierten Verfahrens, nämlich die *modulare*

---

<sup>5</sup>vgl. [Blum92]

|                                     | 1. Bild  | 2. Bild  | 3. Bild  | 4. Bild  |
|-------------------------------------|----------|----------|----------|----------|
| morphologische Konturpunktdetektion | 7.599    | 7.633    | 7.649    | 7.649    |
| Konturpunktverkettung               | 1.166    | 1.133    | 1.099    | 1.116    |
| Objektrekonstruktion                | 2.449    | 2.033    | 2.199    | 2.016    |
| Gesamtlaufzeit von MORPHOBJ         | 11.214   | 10.799   | 10.947   | 10.781   |
| Objektbasierte Integration          | 3.683    | 3.633    | 3.566    | 3.616    |
| Gesamtlaufzeit von OBJMOTION        | 15.082   | 14.599   | 14.682   | 14.549   |
| Rechenzeit von REBLOMAT             | 208.841  | 209.108  | 208.574  | 208.574  |
| Verhältnis beider Rechenzeiten      | 1:13.847 | 1:14.323 | 1:14.206 | 1:14.335 |

Tabelle 6.1: Benötigte Rechenzeit bei der „car“-Sequenz

|                                     | seq1     | seq2     | seq3     | seq4     |
|-------------------------------------|----------|----------|----------|----------|
| morphologische Konturpunktdetektion | 7.599    | 7.066    | 5.283    | 6.616    |
| Konturpunktverkettung               | 1.166    | 1.433    | 0.899    | 0.849    |
| Objektrekonstruktion                | 2.449    | 4.366    | 1.449    | 2.549    |
| Gesamtlaufzeit von MORPHOBJ         | 11.214   | 12.865   | 7.631    | 10.014   |
| Objektbasierte Integration          | 3.683    | 3.499    | 3.149    | 3.799    |
| Gesamtlaufzeit von OBJMOTION        | 15.082   | 16.516   | 10.966   | 13.813   |
| Rechenzeit von REBLOMAT             | 208.841  | 210.141  | 208.641  | 212.058  |
| Verhältnis beider Rechenzeiten      | 1:13.847 | 1:12.723 | 1:19.026 | 1:15.352 |

Tabelle 6.2: Benötigte Rechenzeit bei verschiedenen Sequenzen

- seq1: „car“-Sequenz
- seq2: „tussy“-Sequenz
- seq3: „prlcar“-Sequenz
- seq4: eine künstlich erzeugte Sequenz

*Gestaltung* des Algorithmus, die aus der Zerlegung der Aufgabe der Objekterkennung und der Bewegungsschätzung in einfache Teilschritte erfolgt, und die *lokale* Arbeitsweise der Operatoren, die durch Vermeidung komplexer Rechenoperationen erfolgt, spielen eine entscheidende Rolle für den oben genannten geringen Aufwand.

Tabelle 6.1 zeigt die benötigte Rechenzeit<sup>6</sup> der einzelnen Module des entwickelten objektbasierten Schätzalgorithmus für mehrere Halbbilder einer Sequenz (Abbildung 4.2), die Rechenzeit des blockbasierten Schätzers nach [Blum92], sowie deren Verhältnis. Die Rechenzeit beider Verfahren bleibt für die Bilder einer Sequenz konstant. Das Verhältnis der Rechenzeiten beider Verfahren (Optimierung der Vektorfelder : Schätzung der Vektorfelder) ist 1:14. Der geringe Aufwand dieser Optimierung ist für einen Einsatz zur Zwischenbildinterpolation auf der *Empfängerseite* gut vertretbar.

In Tabelle 6.2 sind die benötigten Rechenzeiten der einzelnen Module des objektbasierten Schätzalgorithmus, jeweils für das erste Halbbild verschiedener Sequenzen (vgl. Abbildungen 4.2, 4.16, 6.3, und 6.5), die Rechenzeit des blockbasierten Schätzer

<sup>6</sup>Diese Zeit ist mit Hilfe der clock-Bibliotheksfunktion der C-Programmiersprache ermittelt



nach [Blum92], sowie deren Verhältnis angegeben. Die Rechenzeit des Verfahrens nach [Blum92] ist für die verschiedenen Sequenzen identisch. Die kleinen Abweichungen der Rechenzeit des hier entwickelten Verfahrens begründen sich mit den unterschiedlichen Strukturen der Bilder der Sequenzen. Das Modul „objektbasierte Integration“ (Integration der Objektinformation in die Bewegungsschätzung) weist eine ziemlich konstante Rechenzeit auf, die aus der Regelmäßigkeit des Integrationsalgorithmus erfolgt. Das Verhältnis der Rechenzeiten beider Verfahren bei verschiedenen Sequenzen (Optimierung der Vektorfelder : Schätzung der Vektorfelder) ist 1:16.

## **6.5 Einige Ergebnisse des Simulationsprogramms MORPHOBJ**

Im folgenden sind einige Anwendungsbeispiele für die, auf der Basis morphologischer Operationen entwickelte Objekterkennung in Bildsequenzen gezeigt. Es sind jeweils die Originalbilder und die mit Hilfe des Programms MORPHOBJ erzeugten Objektbilder (invertiert) angegeben.

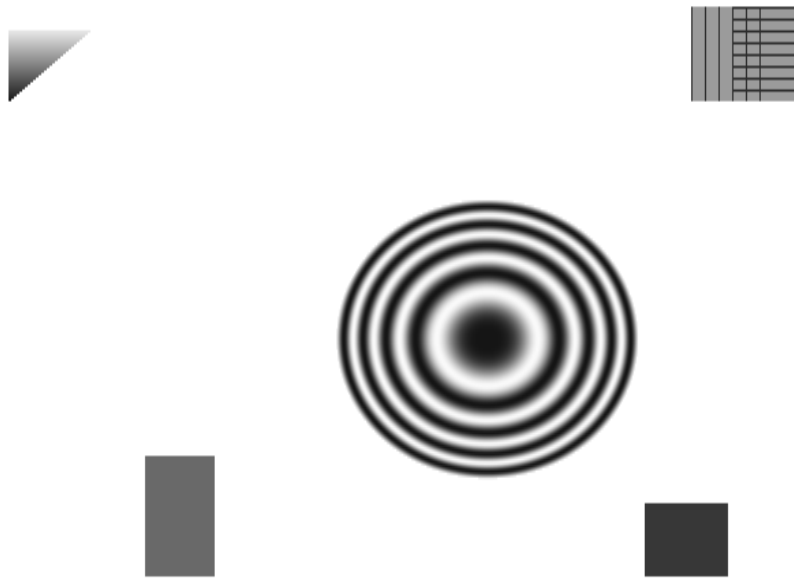


Abbildung 6.3: Beispiel 1: Ein Bild aus einer künstlich erzeugten Sequenz



Abbildung 6.4: Die erkannten Objekte aus Bild 6.3



Abbildung 6.5: Beispiel 2: Ein Bild aus der „prlcar“ -Sequenz



Abbildung 6.6: Die erkannten Objekte aus Bild 6.5



Abbildung 6.7: Beispiel 3: Ein Bild aus der „tennis“ –Sequenz



Abbildung 6.8: Die erkannten Objekte aus Bild 6.7



Abbildung 6.9: Beispiel 4: Ein Bild aus der „voiture“ –Sequenz



Abbildung 6.10: Die erkannten Objekte aus Bild 6.9

# Kapitel 7

## Abschließende Betrachtungen

### 7.1 Zusammenfassung

Die Bedeutung der Bewegungsschätzung in der heutigen Videosignalverarbeitung wird immer größer, besonders für die auf der Empfängerseite notwendige Zwischenbildinterpolation zur Verbesserung der Bildqualität von Videosignalen.

Im Rahmen dieser Arbeit wurde ein Bewegungsschätzer entwickelt, welcher, aufgrund aus dem Bild extrahierter Objektinformationen, die Bestimmung von Bewegungsvektoren unterstützt und verbessert. Somit können die Vektorfelder der Objekte homogenisiert werden. Eine Homogenisierung des Vektorfeldes, insbesondere innerhalb bewegter und texturierter Objekte, ist vorteilhaft, da somit deutlich störende Interpolationsartefakte vermieden werden können. Ist das Vektorfeld solch eines texturierten Objektes inhomogen, so ergeben sich bei der Interpolation Versätze zwischen den Texturen bzw. Teile des Objektes, die deutlich wahrgenommen werden können.

Die Extraktion der Objektinformationen basierte auf den für diesen Zweck vorteilhaften Algorithmen der mathematischen Morphologie.

Nach einer Einführung in den Themenbereich der Bewegungsschätzung wurden die im weiteren benötigten Konzepte, Methoden und Begriffsdefinitionen der Objekterkennung dargestellt. Darauf aufbauend erfolgte der Entwurf eines Softwarekonzeptes für die objektbasierte Bewegungsschätzung.

Um den speziellen Anforderungen an eine empfängerseitige Bewegungsschätzung möglichst gerecht zu werden, wobei geringer Hardwareaufwand im Vordergrund steht, wurde ein mehrstufiger Ansatz entwickelt, der die Aufgabe der objektbasierten Bewegungsschätzung in einfache Teilaufgaben zerlegt und komplexe Rechenoperationen vermeidet. Für eine objektbasierte Bewegungsschätzung in Bildsequenzen müssen zuerst Informationen über die im Bild enthaltenen Strukturen (Objekte) zur Verfügung gestellt werden. Um Strukturen aus dem Rest eines Bildes herauszufinden, wurde die mathematische Morphologie (Lehre von Strukturen und Formen) als Grundlage verwendet. Mit Hilfe morphologischer Grundoperatoren wurden zuerst die Grenzpunkte der Strukturen (Konturpunkte) lokalisiert und anschließend verarbeitet, so daß dünne, meist ein Pixel breite Konturpunktlinien an den Grenzen der Objekte entstanden.

Diese morphologische Verarbeitung der Strukturen des Bildes ermöglichte eine schnelle und einfache Erkennung von Objektkonturen. Die morphologisch gewonnenen

Konturpunkte wurden zu Konturen verkettet. Dabei wurden die dadurch erkannten kleinen, falschen und für die Bewegungsschätzung nicht signifikanten Objektkonturen eliminiert. Diese Reduktion der herausgefundenen Konturen und damit der zu erkennenden Objekte auf wenige, große Objekte ist für einen Einsatz in der Videosignaltechnik vorteilhaft. Die menschliche visuelle Wahrnehmung wird von den Bewegungen weniger großer Objekte geleitet, denen das Auge folgt („tracking“) und für die es ein erhöhtes Auflösungsvermögen besitzt.

Jede Kontur ist durch eine endlich geordnete Folge von Konturpunkten gegeben. Sie ist durch die Angabe der absoluten Ortskoordinaten ihrer Punkte eindeutig dargestellt. Die charakteristischen Eigenschaften der erkannten einzelnen Konturen sind in einer Datenstruktur abgelegt und für die darauf aufbauende Rekonstruktion von Bildobjekten, deren innere Strukturen durch die morphologische Konturpunktdetektion eliminiert wurden, zur Verfügung gestellt.

Für eine leistungsfähige objektbasierte Bewegungsschätzung, bei der jedem Objekt, d.h. allen Punkten eines Objektes, möglichst eine Bewegung zugeordnet wird (eine Bewegung/Objekt-Beziehung), stellen die ermittelten Objektkonturen allein keine ausreichende Grundlage dar. Aus diesem Grund wurde mit Hilfe der erzeugten Konturlisten, insbesondere mit Hilfe der angegebenen Koordinaten der Konturpunkte, das Innere einer Kontur mittels eines Ausfüllungsverfahrens, *ohne dabei auf die Bilddaten* zuzugreifen, markiert. Mit dem Ausfüllalgorithmus wurden auch die Koordinaten jedes inneren Punktes bestimmt. Dadurch wurden alle Punkte eines Objektes erkannt. Die Informationen, die ein Objekt kennzeichnen und es eindeutig beschreiben, wurden für die folgende Bewegungsschätzung in einer Objektliste abgelegt.

Für die Optimierung von Vektorfeldern wurde ein Regelsatz erstellt, der eine Umblendung zwischen den erkannten Objekten und den zu optimierenden Vektorfeldern ermöglicht. Hierfür wurden vier Problemfälle aufgeführt, die jeweils durch entsprechende Regeln behandelt wurden. Bei dem ersten Fall wurde eine Bewegung/Objekt-Beziehung angestrebt. Im zweiten Fall wurden die Hintergrundbewegungen betrachtet. Hier wurde eine Bewegung für den Hintergrund, der aus den erkannten Nichtobjektpunkten besteht, ermittelt. Für falsch erkannte Objektinformationen sind im dritten Fall entsprechende Regeln aufgestellt worden. Die Behandlung von nicht translatorischen Bewegungsformen wurde im vierten Fall beschrieben.

Durch verschiedene Simulationen wurde der geringe Aufwand des entwickelten Algorithmus nachgewiesen. Die Optimierung der Bewegungsinformationen mittels erkannter Objektinformationen wurde vor allem durch die Vermeidung von Zugriffen auf die Bilddaten durchgeführt. Der für die Optimierung von Vektorfeldern benötigte Zeitbedarf ist im Vergleich zu dem Zeitbedarf für die Ermittlung dieser Vektorfelder sehr gering. Verschiedene durchgeführte Simulationen mit Hilfe von Simulationsprogrammen, REBLOMAT nach [Blum92] und OBJMOTION nach dem hier entwickelten Verfahren, haben gezeigt, daß das Aufwandsverhältnis zwischen dem hier entwickelten objektbasierten Algorithmus zur Optimierung der Vektorfelder und dem nach [Blum92] entwickelten bildpunktbasierten (blockbasierten) Bewegungsschätzer 1:14 ist. Dieses Ergebnis ist im Hinblick auf einen empfängerseitigen Einsatz gut vertretbar. Bei einer möglichen Hardwarerealisierung des hier entwickelten Algorithmus können die Objekterkennung und die bildpunktbasierte Bewegungsschätzung parallel durchgeführt

werden. Da die Objekterkennung wesentlich schneller als die Bewegungsschätzung ist, entsteht keinen neuen zeitkritischen Verarbeitungspfad. Somit ist allein durch die Hinzufügung der Objekterkennung keine Erhöhung der Verarbeitungsfrequenz erforderlich. Lediglich das Verknüpfen der Objektinformationen mit den Bewegungsinformationen erfordert zusätzliche Operationen im zeitkritischen Pfad.

Insgesamt wurde ein Verfahren vorgestellt, das die vielversprechenden Möglichkeiten und Vorteile der *objektbasierten* Verarbeitung für einen zukünftigen langfristigen Einsatz verdeutlicht und unterstreicht.

## 7.2 Ausblick

Die Anforderungen an Algorithmen für die Bewegungsschätzung sind in vielen Anwendungsbereichen sehr hoch. Leistungsfähige Verfahren zur empfängerseitigen Zwischenbildinterpolation für die Verbesserung der Bildqualität benötigen möglichst genau ermittelte Vektorfelder, bei denen zu jedem Bildobjekt möglichst ein Vektor zugeordnet ist. Diese Algorithmen sollen auf der anderen Seite aber geringen Hardwareaufwand aufweisen. Die konventionellen bildpunktbasieren (blockbasierten) Methoden liefern Vektorfelder, die im Hinblick auf eine *Bewegung/Objekt-Beziehung* Spielraum für Verbesserung lassen.

Der im Rahmen dieser Arbeit vorgeschlagene Weg für eine *objektbasierte* Bewegungsschätzung konnte die Qualität der blockbasiert ermittelten Vektorfelder mit geringem Aufwand steigern. Für eine in vielen Anwendungen notwendige (erwünschte) *Bewegung/Objekt-Beziehung*, bei der jedem *realen Objekt* der Szene ein Bewegungsvektor zugeordnet wird, scheint es vielversprechend zu sein, die Strategie der *objektbasierten* Bewegungsschätzung zu verfolgen. So könnten zu den, in dieser Arbeit dargestellten Regeln und Kriterien für die Integration von gewonnenen Objektinformationen in die Bewegungsschätzung, weitere Regeln mit anderen möglichen Variablen, die die behandelten Fälle ergänzen, hinzugefügt werden.

Bei Betrachtung der vorgeschlagenen Regeln fällt auf, daß viele linguistische Variablen benutzt werden, um zwischen den Objektinformationen und den Bewegungsinformationen umzublenden. Für die Auswertung dieser Regeln bieten sich die Methoden der unscharfen Logik (engl. *fuzzy logic*) an, die vielversprechend und geeignet zur Beschreibung von vagen und nicht vollständigen Informationen sind. So könnten, z.B., Begriffe, wie „Wenn die Vektorfläche groß gegenüber der Objektfläche ist“, durch Aufstellung von *unscharfen Mengen* (engl. *fuzzy set*) ausgewertet werden.

Ein weiterer Aspekt zur Erweiterung des hier vorgestellten objektbasierten Ansatzes besteht darin, zur Erkennung von nicht translatorischen Bewegungsformen die Verteilung der Vektoren in einem Objektbereich durch das Vektorhistogramm intensiver auszuwerten.

Andere Erweiterungen könnten auch für die Objekterkennung erfolgen. Hier könnten Methoden der automatischen Bestimmung (Auswertung des Bildhistogramms) von geeigneten Schwellwerten zur Unterstützung der Objektisolierung eingestetzt werden. Ein anderer denkbarer Weg zur Verbesserung der Objektisolierung wäre, statt einer statischen Schwellwertentscheidung eine dynamische Schwellwertentscheidung durch



Einbeziehung der gewonnenen Objektinformationen zur Bestimmung eines bereichsabhängigen Schwellwertes durchzuführen.

Für das Modul „Objektrekonstruktion“, bei dem die verketteten Konturen mit Hilfe eines rekursiven Algorithmus ausgefüllt werden, könnte das „scanline“-Verfahren<sup>1</sup> zur Konturausfüllung als eine mögliche bessere Alternative geprüft werden.

Weiterhin ist zu überprüfen, in welchen Anwendungen weitere Einsatzgebiete der gewonnenen Objektinformationen liegen (z.B. Codierung von Bildsequenzen).

---

<sup>1</sup>vgl. [Raub93]

# Erklärung und Einwilligung

## Erklärung

Ich versichere, daß ich diese wissenschaftliche Arbeit selbständig verfaßt und keine anderen, als die angegebenen Quellen und Hilfsmittel, benutzt habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder Sinn nach entnommen sind, wurden in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Das gleiche gilt für die beigegebenen Skizzen und Darstellungen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Dortmund, den 23. Dezember 1994

Unterschrift\_\_\_\_\_

## Einwilligung

Hiermit erkläre ich mich damit einverstanden, daß diese wissenschaftliche Arbeit nach den Bestimmungen des §6 Absatz 1 Satz 1 des Gesetzes über Urheberrecht vom 9.9.1965 in die Bereichsbibliothek aufgenommen und damit für Leser der Bibliothek öffentlich zugänglich gemacht wird.

Ferner bin ich damit einverstanden, daß gemäß §54 Absatz 1 Satz 1 dieses Gesetzes, Leser zu persönlichen oder wissenschaftlichen Zwecken Kopien aus der Arbeit anfertigen dürfen.

Dortmund, den 23. Dezember 1994

Unterschrift\_\_\_\_\_

# Literaturverzeichnis

- [Bez81] Bezdek, J.  
*Pattern Recognition with Fuzzy Objective Function Algorithms*  
New York [u.a.]: Plenum Press, 1981
- [BBbv91] Bässmann, H., Besslich Ph.  
*Bildverarbeitung Ad Oculos*  
Berlin [u.a.]: Springer, 1991
- [BaBe91] Bässmann, H., Besslich Ph.  
*Konturorientierte Verfahren in der digitalen Bildverarbeitung*  
Berlin [u.a.]: Springer, 1989
- [Bier91] Bierling, M.  
*Hierarchische Displacementschätzung zur Bewegungskompensation  
in digitalen Fernsehsequenzen*  
VDI-Bericht Nr. 179  
Düsseldorf: VDI Verlag, 1991
- [Blum92] Blume, H.  
*Verfahren und Schaltungsstrukturen für eine Bewegungsschätzung  
mit örtlich-zeitlicher Prädiktion*  
Diplomarbeit D5-92, Lehrstuhl für Nachrichtentechnik  
Universität Dortmund  
Dortmund: 1992
- [Blum93] Blume, H.  
*Bewegungsschätzung in Videosignalen mit parallelen  
örtlich-zeitlichen Prädiktoren*  
Tagungsband des 5. Dortmunder Fernsehseminars, 29.9.–01.10.1993  
Dortmund: 1993
- [Blum94] Blume, H., Schwoerer, L., Zygis, K.  
*Subband based Upconversion Using Complementary Median Filter*  
Proceedings of the 7th International Congress on HDTV and Beyond,  
16.-28.10.1994  
Turino (Italy): 1994

- [Bock87] Bocklisch, S. F.  
*Prozeßanalyse mit unscharfen Verfahren*  
Berlin: Technik, 1987
- [Bron91] Bronstein, I.N., Semendjajew, K.A.  
*Taschenbuch der Mathematik*  
Moskau [u.a]: Verlag Nauka, Teubner, Verlag Harri Deutsch, 1991
- [Capp87] Cappellini, V.  
*Time-Varying image processing and moving object recognition*  
Proceedings of the International Workshop  
Florence (Italy): North-Holland, September 1986
- [Haan91] de Haan, G., Huijgen, H.  
*Motion estimator for TV-picture enhancement*  
4th International Workshop on HDTV and Beyond  
Torino (Italy): 1991
- [DoPe88] Dörfler, W., Peschek, W.  
*Einführung in die Mathematik für Informatiker*  
München;Wien: Hanser, 1988
- [DuHa73] Duda, R.O., Hart, M.P.  
*Pattern classification and scene analysis*  
New York [u.a.]: John Wiley & Sons, 1973
- [DuIn93] *Duden Informatik*  
Hg. Lektorat des BI-Wissenschaftsverlags  
2. Auflage  
Mannheim [u.a.]: Dudenverlag, 1993
- [Foit79] Foith, J.P.  
*Angewandte Szenenanalyse*  
DAGM Symposium, Karlsruhe  
Informatik-Fachberichte 20  
Berlin [u.a.]: Springer, 1979
- [Fran89] Franke, U.  
*Regionenorientierte Bildbeschreibung –  
Algorithmen und Möglichkeiten*  
VDI-Bericht Nr. 101  
Düsseldorf: VDI Verlag, 1989
- [Gerh88] Gerhard, A.  
*Bewegungsanalyse bei der Codierung von Bildsequenzen* Dissertation,  
Lehrstuhl für Nachrichtentechnik  
Technische Universität München  
München: 1988

- [Geue83] Geuen, W.  
*Konturfindung auf der Basis des visuellen Konturempfindens*  
Dissertation, Fakultät für Maschinenwesen  
Universität Hannover  
Hannover: 1983
- [GiDo88] Giardina C., Dougherty  
*Morphological Methods in Image and Signal Processing*  
New Jersey: Prentice Hall, 1988
- [Gilg90] Gilge, M.  
*Regionenorientierte Transformationscodierung  
in der Bildkommunikation*  
VDI-Bericht Nr. 128  
Düsseldorf: VDI Verlag, 1990
- [GoWo92] Gonzalez, R., Woods, R.  
*Digital Image Processing*  
Reading [u.a.]: Addison-Wesley, 1992
- [Groh91] Groh, C.  
*Rechnergestützte Analyse der Bewegungen von Samenzellen -  
Robuste Objekterkennung und Bahnverfolgung*  
Dissertation, Mathematisch-Naturwissenschaftliche Fakultät  
Humboldt-Universität zu Berlin  
Berlin: 1991
- [Guse92] Guse, W.  
*Objektorientierte Bewegungsanalyse in Bildfolgen*  
VDI-Bericht Nr. 223  
Düsseldorf: VDI Verlag, 1992
- [Haber89] Haberäcker, P.  
*Digitale Bildverarbeitung*  
Grundlagen und Anwendungen  
3. überarbeitete Auflage  
München [u.a.]: Hanser, 1989
- [Haus94] Hauske, G.  
*Systemtheorie der visuellen Wahrnehmung*  
Stuttgart: Teubner Verlag, 1994
- [Hent85] Hentschel, Ch.  
*Flimmerreduktionsverfahren zur Verbesserung  
der Fernseh-Bildwiedergabe*  
ntzArchiv Bd. 7 (1985) H. 9  
Braunschweig: 1985

- [Heri92] Hering, E.  
*Software Engineering*  
Braunschweig [u.a.]: Vieweg, 1992
- [Heuc86] Heuck, N.  
*Ein Verfahren zur Segmentierung von Bildern  
aus dynamischen Bildsequenzen*  
Dissertation, Fakultät für Elektrotechnik  
RWTH Aachen  
Aachen: 1986
- [Hott90] Hötter, M.  
*Object-oriented analysis-synthesis coding  
based on moving two-dimensional objects*  
Signal Processing: Image Communication, Band II  
Dezember 1992
- [Kett90] Kett, P.K.  
*Methoden der Bildverarbeitung für die automatische  
Segmentierung und Strukturanalyse von dreidimensionalen  
Bildern eines konfokalen Laser Scanning Mikroskopes*  
Dissertation, Naturwissenschaftliche–Mathematische Gesamtfakultät  
Ruprecht–Karl Universität Heidelberg  
Heidelberg: 1990
- [Kirc93] Kirchner, H.  
*Bewegungserkennung in Bildfolgen*  
Wiesbaden: DeutscherUniversitätsVerlag, 1993
- [KlZa92] Klette, R., Zamperoni, P.  
*Handbuch der Operatoren für die Bildbearbeitung*  
Braunschweig [u.a.]: Vieweg, 1992
- [KMW85] Kummerfeldt, G., May, F. und Wolf, W.  
*Coding television signals at 320 and 64 kbit/s*  
In 2nd International Technical Symposium on Optical and  
Electro-Optical Applied Science and Engineering  
Proceedings SPIE  
Cannes: Dezember 1985
- [Lehn94] Lehn, K.  
*Unschärfe Klassifizierung  
kontaminationsverdächtiger Flächen*  
Diplomarbeit, Fachbereich für Informatik  
Universität Dortmund  
Dortmund: 1994

- [Mest89] Mester, R.  
*Regionenorientierte Bildsegmentierung unter Verwendung stochastischer Bildmodelle*  
VDI-Bericht Nr. 106 Düsseldorf: VDI Verlag, 1989
- [Niem74] Nieman, H.  
*Methoden der Mustererkennung*  
Frankfurt am Main: Akademische Verlagsgesellschaft, 1974
- [Niem89] Niemann, H.  
*Pattern Analysis and Understanding*  
Second Edition  
Berlin [u.a.]: Springer, 1989
- [Pavl90] Pavlidis, Theo  
*Algorithmen zur Grafik und Bildverarbeitung*  
Hannover: Heise, 1990
- [PeCh94] Pei, S., Chen, F.  
*Image sampling structure conversion by morphological filters*  
in *Signal Processing: Image Communication* 6 (1994), 13-24  
Taipei (Taiwan): 1994
- [Raub93] Rauber, Th.  
*Algorithmen in der Computergraphik*  
Stuttgart: Teubner, 1993
- [Sale94] Salembier, P.  
*Morphological multiscale segmentation for image coding*  
in *Signal Processing* 38 (1994), 359-386  
Barcelona (Spain): 1994
- [Scho87] Schöning, U.  
*Logik für Informatiker*  
Koblenz: 1987
- [Serr82] Serra, Jean  
*Image analysis and mathematical morphology, Volume 1*  
London [u.a.]: Academic Press, 1982
- [Trie78] Triendl, E.  
*Bildverarbeitung und Mustererkennung*  
DAGM Symposium, Oberpfaffenhofen Informatik-Fachberichte 17  
Berlin [u.a.]: Springer, 1978

- [Proj93] Wawrzyniak, J., Rüschenbaum, W., Spork, F., Michau S.  
*Erstellung einer Multimedia-Präsentation  
zu ausgewählten Kapiteln der Bildverarbeitung*  
Projektgruppe P12 WS93/94, Lehrstuhl für Nachrichtentechnik  
Universität Dortmund  
Dortmund: 1993
- [Wend88] Wendland, B.  
*Fernsehtechnik, Band I:  
Grundlagen*  
Heidelberg: Hütig Verlag, 1988
- [WSch92] Wendland, B., Schröder, H.  
*Fernsehtechnik, Band II:  
Systeme und Komponenten zur Farbbildübertragung*  
Heidelberg: Hütig Verlag, 1992
- [Wolf91] Wolf, G.  
*Methoden zur Nutzung von A-priori-Wissen und  
globaler Information für die Objektisolierung*  
Dissertation, Mathematisch-Naturwissenschaftliche Fakultät  
Humboldt-Universität zu Berlin  
Berlin: 1991
- [Wu1994] Wu, Xiaofeng  
*Synthetische Kantenverteilung zur Verbesserung der Bildschärfe*  
VDI-Bericht Nr. 272  
Düsseldorf: VDI Verlag, 1994
- [Zamp89] Zamperoni, P.  
*Methoden der digitalen Bildsignalverarbeitung*  
Braunschweig [u.a.]: Vieweg, 1989