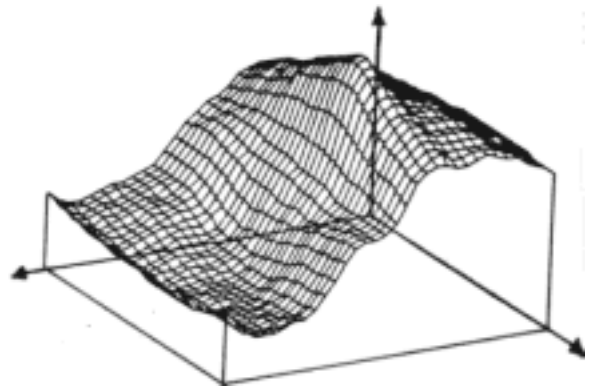
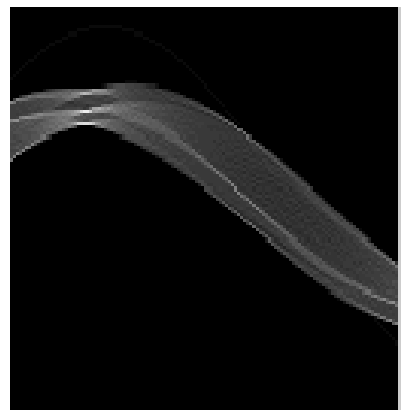
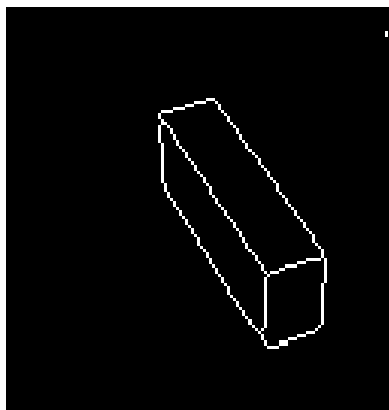
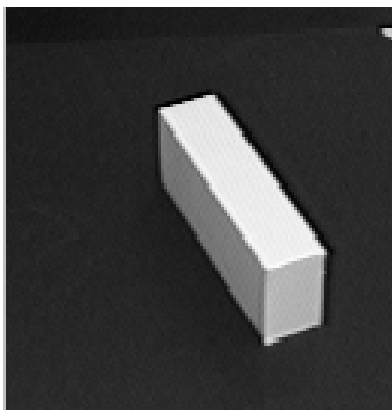


Computer Vision



W.-D. Groch

h_da - fbi



Inhaltsübersicht: Computer Vision

- 1 Einführung und Überblick**
- 2 Visuelle Wahrnehmung**
 - 2.1 Anatomie und Funktion des Sehsystems
 - 2.2 Visuelle Empfindungen und ihre Verarbeitung
 - 2.3 Zusammenfassung: Sicht des Wahrnehmungsprozesses
- 3 Vergleich bildhafter Informationen**
 - 3.1 Bilddifferenzen
 - 3.2 Quantitativer Vergleich gleich großer Bilder
 - 3.3 Quantitativer Vergleich ungleich großer Bilder
- 4 Konturorientierte Segmentierung, Kontur-Repräsentationen**
 - 4.1 Kanten- und Liniendetektion
 - 4.2 Nachverarbeitung der Ergebnisse der Kanten-/Liniendetektion
 - 4.3 Kanten-/Linienelement-Verkettung
 - 4.4 Hough-Transformation
 - 4.5 Sequentielle Kanten-/Linienextraktion
 - 4.6 Konturorientierte Repräsentationen
- 5 Interpretation von Strichzeichnungen**
 - 5.1 Allgemeine Vorbemerkungen zur 3D-Problematik
 - 5.2 Markierung einzelner Linien und Linienschnittpunkte
 - 5.3 Konsistente Markierung von Strichzeichnungen
 - 5.4 Erweiterungen der Verfahren
- 6 Stereobildauswertung**
 - 6.1 Hindernis-Ermittlung mit inverser Projektion
 - 6.2 Geometrische Grundlagen bzgl. Vergenz und Disparität
 - 6.3 Korrespondenzproblem
 - 6.4 Stereobildauswertung nach Marr und Poggio
 - 6.5 PMF-Algorithmus zur Stereobildauswertung
- 7 Bildfolgenauswertung**
 - 7.1 Grundlagen der Bildfolgenauswertung
 - 7.2 Kollisionsvorhersage
 - 7.3 Differentieller Ansatz zur Bildfolgenauswertung
- 8 Shape from X**
 - 8.1 3D-Form aus Beleuchtungsänderungen (Photometrisches Stereo)
 - 8.2 3D-Form aus Konturen
 - 8.3 3D-Form aus Texturen
- 9 Wissensbasierte Bildinterpretation**
 - 9.1 Komponenten einer wissensgesteuerten Bildanalyse
 - 9.2 Wissensbasis: Wissensinhalte und Wissensrepräsentation
 - 9.3 Wissensnutzung (Systemsteuerung, Kontrolle)
 - 9.4 Beispiel für ein allgemeines System-Modell
 - 9.5 Abschließende allgemeine Überlegungen zu Bildanalysesystemen

Literaturverzeichnis

Index

1 Einführung und Überblick

Nachdem in "Graphische Datenverarbeitung 1 und 2" bereits die Grundlagen der Bildbearbeitung (Originalbild → manipuliertes Bild) und Bildverarbeitung (Originalbild → segmentierte, einfache Objekte und Merkmale) behandelt wurden, sollen hier weiterführende Bildverarbeitungs-Techniken und -Methoden vorgestellt werden.

Die Zielsetzung ist hierbei, den menschlichen Seh- und Wahrnehmungsprozess auf den Rechner zu übertragen, um dort zu "maschinellern Sehen" oder gebräuchlicher zu "Computer Vision" zu kommen.

Bildverarbeitungssysteme, die zu einfachen automatischen Bildauswertungen in künstlichen Umgebungen in der Lage sind, werden derzeit z.B. massiv in der Fertigungsautomatisierung eingesetzt. Im Gegensatz hierzu ist die automatische Bildinterpretation, die beispielsweise einem Roboter die autonome Navigation in beliebigen, natürlichen Umgebungen erlauben würde, noch Gegenstand intensiver Forschungsarbeiten.

Bevor in den folgenden Kapiteln 3 bis 9 ausgewählte Computer-Vision-Ansätze vorgestellt werden, wird in Kapitel 2 zunächst das visuelle Wahrnehmungssystem des Menschen behandelt. Zahlreiche hierbei gewonnenen Erkenntnisse können direkt auf den Rechner zur Lösung von Computer-Vision-Aufgaben übertragen werden.

In Kapitel 3 wird auf den Vergleich bildhafter Information eingegangen. Dies kann beispielsweise eingesetzt werden, um in Bildfolgen eine Änderungsentdeckung (z.B. zu Überwachungszwecken) durchzuführen. Eine anderer Zielsetzung ist das Durchsuchen von "großen" Bildern nach Stellen, die einem "kleinen", vorgegeben Bildausschnitt möglichst ähnlich sind. Auf diese Weise können sowohl markante Bildstellen, wie beispielsweise Linien, Kanten oder Ecken, als auch Abbildungen einfacher Objekte (z.B. Muttern auf einem Fließband oder Blut- oder Gewebezellen in Abstrichen) automatisch gefunden werden. Ein weiteres Anwendungsgebiet ist die Stereo- und Bildfolgenauswertung (siehe Kapitel 6 und 7), bei der markante Stellen aus einem Bild im jeweils anderen Stereobild bzw. einem späteren Bild der Bildfolge wieder gefunden werden sollen, um so Tiefen- bzw. Bewegungsinformation zu gewinnen.

Für viele Computer-Vision-Ansätze ist es notwendig, die Bilddaten vollständig oder zumindest teilweise zu segmentieren. Deshalb werden in Kapitel 4 Methoden zur konturorientierten Segmentierung vorgestellt. (Der hierzu duale Ansatz zur flächenorientierten Segmentierung wird in "2D- und 3D-Bildverarbeitung" genauer betrachtet).

In Kapitel 5 werden erste, besonders einfache Ansätze zur räumlichen Interpretation von Bildern vorgestellt. Am Beispiel von Strichzeichnungen, wie sie beispielsweise bei der perfekten Segmentierung einfacher Szenen aus der "Klötzchenwelt" entstehen, wurden in der Frühphase des Computer Vision Methoden und Werkzeuge entwickelt, die zum Teil heute noch in fortgeschrittenen Systemen zum Einsatz kommen.

In Kapitel 6 werden die Grundlagen der Stereobildauswertung behandelt. Aus der unterschiedlichen Darstellung von Objekten in Stereobildern lassen sich Informationen über die dreidimensionale Gestalt der Objekte ableiten; z.B.:

- sind zwei Objekte gleichweit von den Stereokameras entfernt, oder ist ein Objekt (welches?) näher an den Kameras platziert, oder
- in welcher Entfernung von den Stereokameras befindet sich ein Objekt?

Kapitel 7 stellt einige grundlegende Methoden aus dem Bereich der Bildfolgenauswertung vor. Aus den zeitlichen Änderungen in aufeinanderfolgenden Bildern einer Bildfolge kann beispielsweise auf die Bewegungen der abgebildeten Objekte und/oder der Kamera geschlossen werden.

Neben der Stereobildauswertung gibt es noch eine ganze Reihe anderer Möglichkeiten, um die räumliche Gestalt von Objekten aus Bildern zu rekonstruieren. Von diesen sogenannten "Shape from X"-Verfahren werden in Kapitel 8 die Ansätze zur Ermittlung der 3D-Gestalt aus Beleuchtungsänderungen, aus Konturen und aus Texturen vorgestellt.

In vielen Fällen führt eine Bildauswertung, die sich nur auf die in den Bildern enthaltenen Informationen stützt, nicht zum Ziel, und es muss zusätzliches Wissen in den Interpretationsprozess eingebracht werden. Die Modellierung, Repräsentation und Nutzung solchen Zusatzwissens ist Gegenstand des 9. Kapitels zur wissensbasierten Bildinterpretation.

Der Einfachheit halber werden die meisten Verfahren am Beispiel von Grauwertbildern vorgestellt. Die Ansätze sind jedoch ohne weiteres auf Farbbilder übertragbar, indem man diese z.B. in den HSV-Farbraum transformiert und dann beispielsweise nur den V-Kanal der Helligkeiten auswertet, oder aber, indem man die einzelnen Farbkanäle getrennt oder parallel auswertet und die Ergebnisse dann geeignet miteinander verknüpft.

Immanuel Kant: „Es bleibt uns völlig unbekannt, was die Objekte sein können, an sich und unabhängig von der Aufnahme durch unsere Sinne. Wir wissen nichts außer der Art und Weise, wie wir sie wahrnehmen.“

2 Visuelle Wahrnehmung (siehe z.B. [Goldstein])

Der Mensch nimmt seine Umwelt hauptsächlich visuell, d.h. mit Hilfe der Augen wahr. Im Gegensatz dazu leben z.B. Hunde in einer "Riechwelt" und Fledermäuse in einer "Hörwelt". Die Wahrnehmung erfolgt subjektiv und für jedes Individuum unterschiedlich und ist stark durch Erfahrung geprägt.

Das Auge selbst ist kein Messgerät, das absolute Werte (Grau- oder Farbwerte) liefert, wie z.B. eine elektronische Kamera. Das Auge misst eher relativ und reagiert insbesondere auf Änderungen: räumliche Änderungen (z.B. Kontrastsprung zwischen zwei unterschiedlich eingefärbten Flächen) oder zeitliche Änderungen (z.B. Bewegungen im Blickfeld). Abbildung 2.1 zeigt ein Beispiel dafür, dass das Auge z.B. Helligkeiten nicht objektiv wahrnehmen kann.



Abb. 2.1: Beispiel dafür, dass das Auge kein objektives Messgerät ist: es sieht ein und denselben Grauwertfleck in Abhängigkeit von seiner Umgebung unterschiedlich hell.

Die Forschung auf dem Gebiet der visuellen Wahrnehmung ist schwierig, da nur der "Input" für den Wahrnehmungsprozess durch gezielte Versuchsaufbauten exakt bestimmbar ist. Das Ergebnis der Wahrnehmung ist i.d.R. nicht messbar. Insbesondere ist auch der Beitrag des Gedächtnisses, d.h. der Erfahrung und des Wissens nicht bestimmbar.

Das Verständnis der Wahrnehmung hat im Laufe der Jahrhunderte einen tiefgreifenden Wandel durchlaufen. Während Plato in seinen Schriften noch schrieb, dass "das Auge korpuskulare Sehstrahlen auf die betrachteten Objekte werfe", verglich Helmholtz das Auge mit einer photographischen Kamera (Prinzip der Lochkamera). Diese Erkenntnis gilt als physikalisches Grundprinzip auch heute noch, jedoch wurden erst in den letzten Jahrzehnten die entscheidenden Ergänzungen erforscht. Die Gestalts-Psychologen stellten fest, dass der Erkennung bestimmte Gesetzmäßigkeiten zugrunde liegen (siehe Kapitel 2.2.5). Die Neurophysiologen untersuchten den Zusammenhang zwischen bestimmten Basis-Erkennungsvorgängen und der Aktivität von Nervenzellen im Gehirn. Und in den letzten Jahren beschäftigen sich die Systemtheoretiker damit, Wahrnehmungsprozesse in Form von neuronalen Netzen zu modellieren.

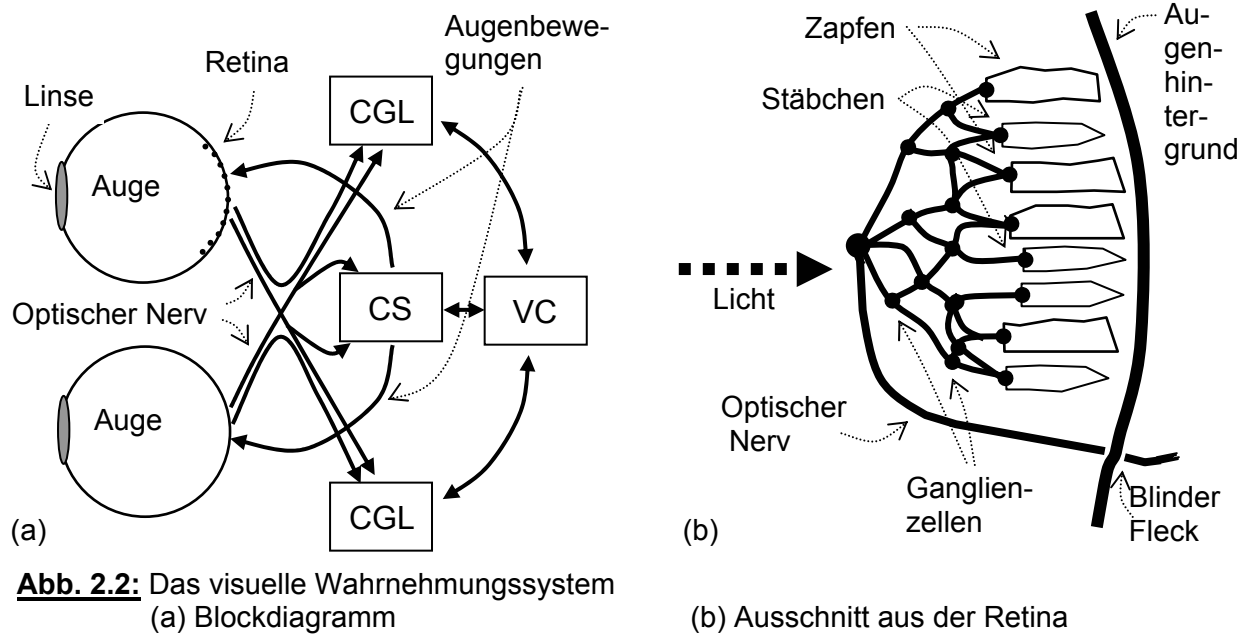
Ein erstaunliches Phänomen der Erkennung ist, dass sie unabhängig von der Beleuchtung oder einer (nicht zu schnellen) Bewegung der Szene oder des Betrachters funktioniert. Dies wird nicht zuletzt durch die Adaptionsfähigkeit der Wahrnehmungssinne erreicht, die sich z.B. an unterschiedliche Beleuchtungsstärken, spektrale Lichtzusammensetzungen und sogar geometrische Spiegelungen oder Deformationen anpassen können.

2.1 Anatomie und Funktion des Sehsystems

Der "Sensor" des Sehsystems (siehe Abbildung 2.2 (a)) ist das Auge. Das Licht tritt vorne durch die **Linse** ein und fällt hinten im Augapfel auf die **Netzhaut (Retina)**. Dort befinden sich zweierlei lichtempfindliche Rezeptoren (siehe Abbildung 2.2 (b)):

- **Stäbchen** (Schwarz/Weiß-Empfindung) und
- **Zapfen** (Farbempfindung).

In der Mitte der Retina befindet sich innerhalb des „**gelben Flecks**“ (**Makula**) die sogenannte **Sehgrube (Fovea)**, in der die Rezeptoren (nur Zapfen) besonders dicht gepackt sind, so dass hier die "schärfste" Wahrnehmung möglich ist.



Die Rezeptoren wandeln das einfallende Licht in neuronale Impulse und leiten diese über mehrere Ebenen von speziellen Nervenzellen (sogenannten **Ganglien**) weiter. In den Ganglien, die von Ebene zu Ebene die Reize von mehr Rezeptoren zusammenfassen (also eine größere räumliche Sicht besitzen), findet bereits eine erste, primitive Reizverarbeitung statt: bestimmte Reize können sich gegenseitig verstärken, andere dagegen schwächen sich gegenseitig ab; so können z.B. Linien und Kanten bestimmter Ausrichtungen erkannt werden. Die neuronalen Impulse (Reize) werden dann durch den **optischen Nerv** (mit mehr als 10^6 Nervenfasern) zum Gehirn weitergeleitet. Im Gehirn wiederum gibt es verschiedene Bereiche, die unabhängig voneinander bzw. im gegenseitigen Wechselspiel bestimmte Verarbeitungen vornehmen:

- 1.) **Corpus Geniculatum Laterale:** hier findet durch die Über-Kreuz-Verschaltung (siehe Abbildung 2.2 (a)) eine Überlagerung (aber noch nicht Verschmelzung) der beiden Retina-Bilder statt, so dass hier u.a. eine erste Stufe der Tiefenwahrnehmung angesiedelt ist;
- 2.) **Colliculus Superior:** hier ist die Schaltstelle zur Steuerung der Augenbewegungen. Außerdem wird hier die Lage und Größe von Objekten ermittelt, und es werden bereits Reize anderer Sinnesorgane einbezogen;
- 3.) **Visueller Cortex:** hier werden die Retina-Bilder "verschmolzen"; weiterhin findet hier die Tiefen- und die Formwahrnehmung statt.

Das genaue Spektrum der einzelnen Verarbeitungseinheiten, sowie insbesondere ihr Zusammenspiel, ist auch heute noch Gegenstand der Forschung.

Im Gehirn findet eine Parallelverarbeitung der anfallenden Reize statt (Neuronales Netz!). Während der Betrachtung eines Punktes (bzw. Objektes) ist das Auge keineswegs ständig auf diesen gerichtet. Das Auge fixiert immer nur kurz und "springt" ansonsten willkürlich "hin und her". Mit diesen sogenannten **Sakkaden** wird das Sehfeld unsystematisch abgetastet, um damit das Umfeld einer interessierenden Stelle mitzuerfassen. Der Ablauf der Sakkaden wird dabei sowohl durch äußere Beeinflussungen (Auffälligkeiten in der Szene, wie z.B. starke Kontraste), als auch durch innere, unbewusste Beeinflussungen (z.B. Erwartungshaltung zum Szeneninhalte) gesteuert. Trotz ständiger Augen-, Kopf- und Körperbewegungen und einer sich i.d.R. ständig ändernden Umgebung haben wir immer die Empfindung eines "stabilen" Bildes.

2.2 Visuelle Empfindungen und ihre Verarbeitung

Es werden vier Gruppen von Empfindungen unterschieden. Auf einer höheren Verarbeitungsstufe werden diese gemeinsam z.B. durch die Prinzipien der Gestalts-Psychologie aktiv verarbeitet.

2.2.1 Farbempfindung

Das menschliche Auge ist im Spektralbereich zwischen etwa 400 und 750 Nanometer empfindlich; siehe Abbildung 2.3. Die Farbempfindung ist dabei unabhängig von der Intensität. Sie erfolgt mit Hilfe der Zapfen, die im wesentlichen im Bereich der Fovea angesiedelt

sind. Es gibt drei Arten von Zapfen, die für drei unterschiedliche Wellenlängen-Bereiche empfindlich sind, aus denen sich jeder Farbton additiv mischen lässt. Das Auge kann ca. 200 Farbton-Stufen unterscheiden. Je Farbton können etwa 20 Sättigungs-Stufen und je Farbton/Sättigung ca. 500 Helligkeits-Stufen unterschieden werden. Dies ergibt insgesamt ungefähr zwei Millionen mögliche Farb-Abstufungen.

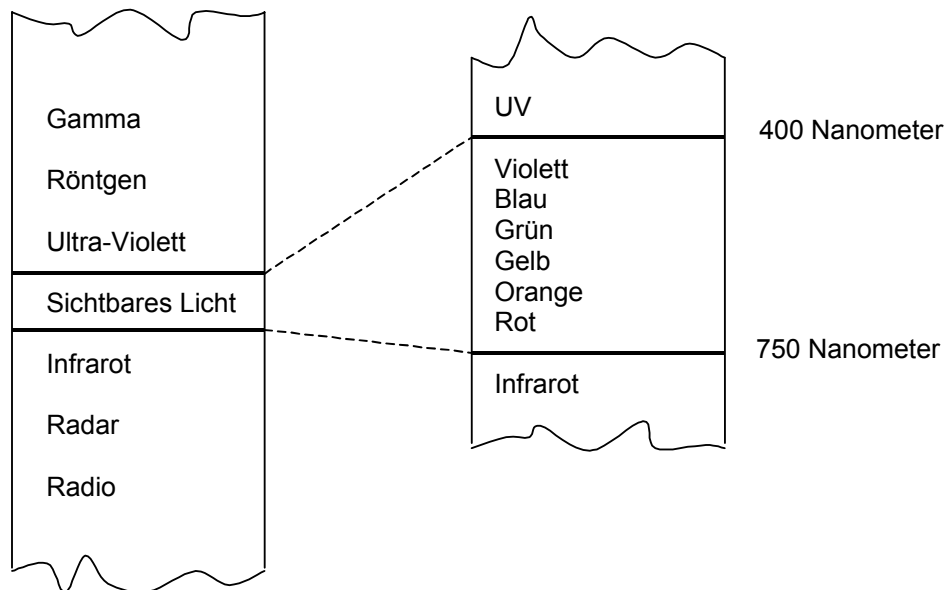


Abb. 2.3: Elektromagnetisches Spektrum und Unterbereich des sichtbaren Lichtes.

Neben den Zapfen gibt es im Auge noch die für das Schwarz/Weiß-Sehen zuständigen Stäbchen. Diese sind über die gesamte Retina verteilt und gewährleisten durch ihre wesentlich höhere Lichtempfindlichkeit, dass wir unsere Umwelt auch noch im Dämmerlicht, d.h. ohne Farben, wahrnehmen können.

2.2.2 Bewegungsempfindung

Die menschliche Wahrnehmung ist auf Bewegungserkennung optimiert, da diese Funktion entwicklungsgeschichtlich gesehen für das Überleben besonders wichtig ist: Bedrohungen sind i.d.R. mit Bewegungen verbunden, und diese müssen sofort und auch am Rande des Sehfeldes zuverlässig erkannt werden.

Die einzelnen Rezeptoren und Nervenzellen reagieren also besonders empfindlich auf zeitliche Veränderungen der Lichtreize, d.h. Bewegungen. Bewegungen in der Szene bzw. Bewegungen des Auges bewirken sogenannte **Fließmuster** auf der Retina. Auf die Bewegungserkennung (**optischer Fluß**, **optical flow**) wird in Kapitel 7 genauer eingegangen.

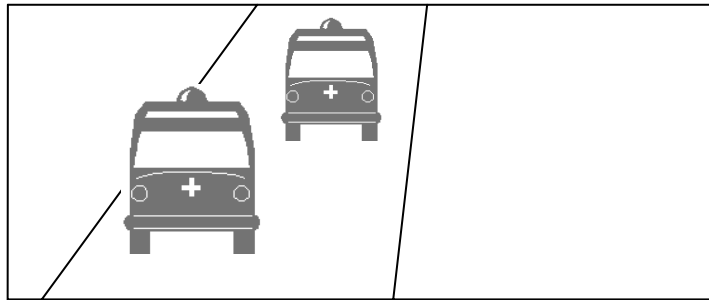
2.2.3 Tiefenempfindung

Das Tiefensehen ist besonders wichtig zur Orientierung in der Umwelt (z.B. Greifen und Fortbewegung mit Kollisionsvermeidung). Für die Tiefenempfindung unterscheidet man zwischen den monokularen und den binokularen Tiefenmerkmalen:

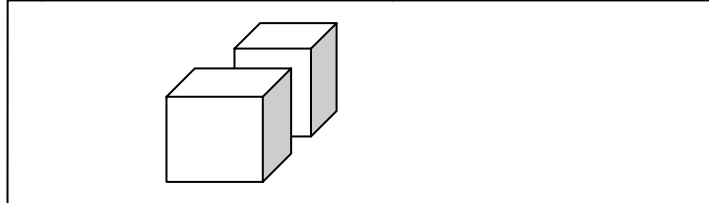
- **monokulare (einäugige) Tiefenmerkmale** (siehe Abbildung 2.4) sind die wahrgenommene Größe bekannter Objekte, die Verdeckung, die lineare Perspektive, die Dunstperspektive, die monokulare Bewegungsparallaxe (Winkelgeschwindigkeiten), Licht- und Schattenwurf, die Augenakkommodation (Linsenkrümmung zur Scharfstellung) sowie die Texturkompression.
- **Binokulare (zweiäugige) Tiefenmerkmale** sind die Vergenz (Anstellwinkel zwischen den Sehachsen beider Augen bei Fixierung eines Punktes; siehe Abbildung 2.5 (a)) sowie die Disparität (Differenzen in den Abbildungen einer Szene mit beiden Augen bei der Fixierung eines Punktes P; siehe Abbildung 2.5 (b); Genaueres siehe Kapitel 6).

Während man früher davon ausging, dass Tiefensehen an Formerkennung gebunden ist, hat man heute mit Hilfe sogenannter **Zufallsstereogramme** nachgewiesen, dass die Tiefenempfindung unabhängig von der Formerkennung erfolgt. Auf die Tiefenerkennung wird in Kapitel 6 (Stereobildauswertung) näher eingegangen.

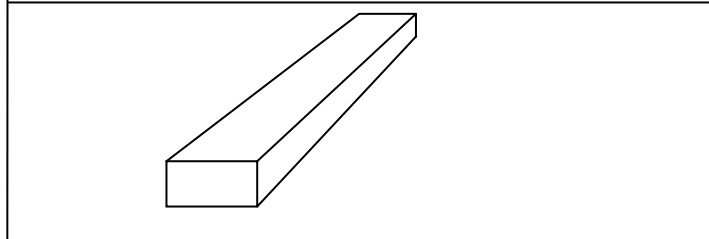
wahrgenommene Größe bekannter Objekte



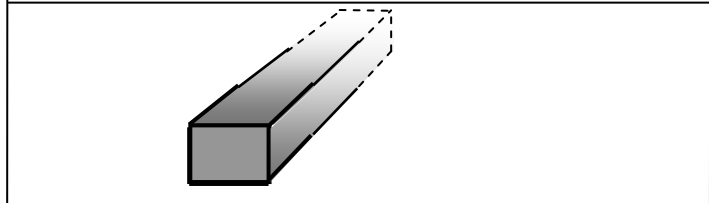
Verdeckung von Objekten



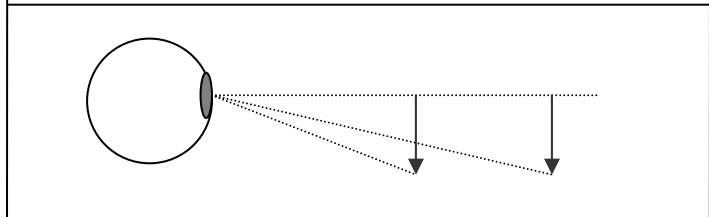
Lineare Perspektive



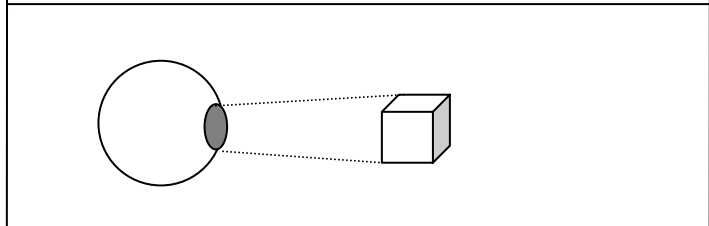
Dunstperspektive



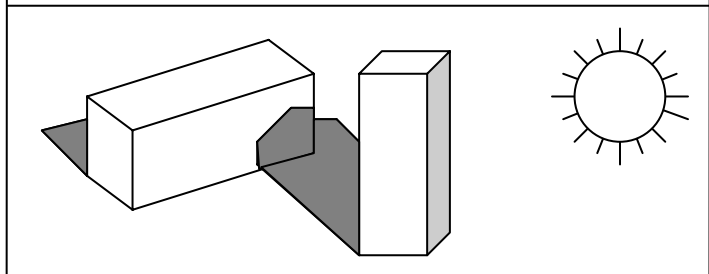
Monokulare Bewegungsparallaxe (Winkelgeschwindigkeiten)



Akkommodation auf einen Entfernungsbereich (Linsenkrümmung zur Scharfstellung)



Verteilung von Licht und Schatten



Textur-Kompression

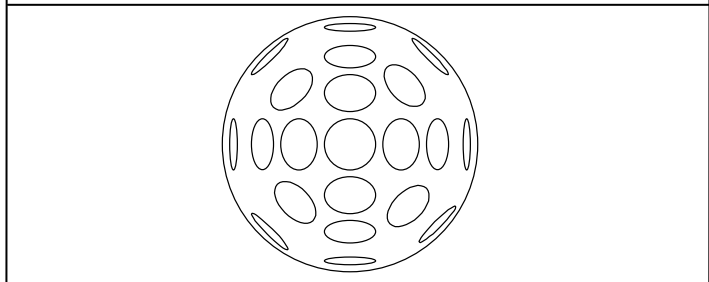
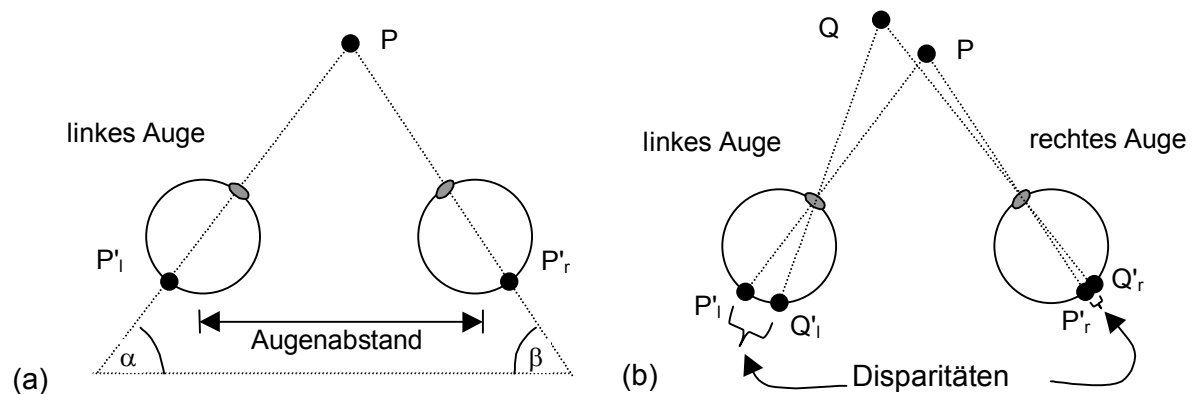


Abb. 2.4: Beispiele für monokulare Tiefenmerkmale.

**Abb. 2.5:** Beispiele für binokulare Tiefenmerkmale:

(a) Vergenz

(b) Disparität

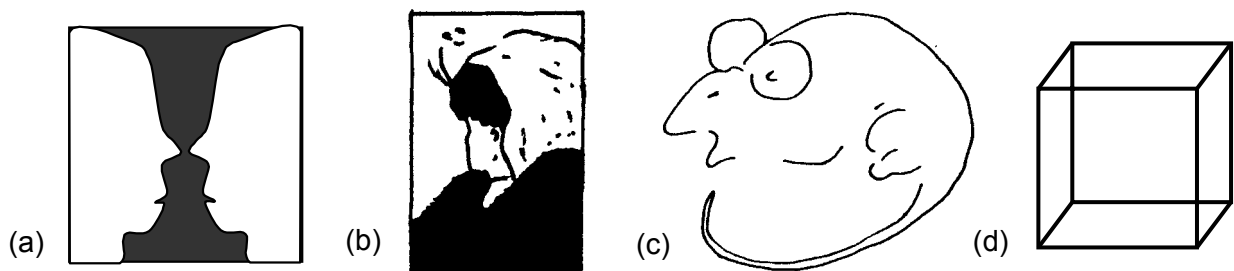
2.2.4 Formempfindung

Über die Formempfindung ist noch relativ wenig bekannt. Man weiß mittlerweile, dass diese im Gegensatz zur Fähigkeit der Trennung von Figur und Hintergrund erlernt werden muss. Bei der Formempfindung greifen insbesondere die Prinzipien der Gestalts-Psychologie, auf die im nächsten Teilkapitel eingegangen wird.

2.2.5 Psycho-Physik des Sehens

Die Fähigkeit zur Trennung von Figur und Hintergrund ist angeboren: stets ist "Etwas" die Figur und "Etwas" der Hintergrund. Die Entscheidung fällt spontan und erfolgt nach mehreren Kriterien; (z.B. sind Figuren eher klein, konvex und geschlossen).

Bei Mehrdeutigkeiten kann eine Bildstelle nie gleichzeitig als Figur und als Hintergrund wahrgenommen werden; siehe Abbildung 2.6 (a). In diesen Fällen ist meist ein bewusstes "Umschalten" von Figur auf Hintergrund und umgekehrt erforderlich, und dieses erfordert eine merkbare "Umschaltzeit". Derselbe Umschalteffekt kann auch bei Figuren mit mehrdeutiger Interpretation beobachtet werden; siehe Abbildung 2.6 (b) bis (d).

**Abb. 2.6:** Bilder mit mehrdeutigen Interpretationen

Die Schule der Gestalts-Psychologie hat herausgefunden, dass die Wahrnehmung nicht passiv auf Reize der Umwelt reagiert, sondern diese Reize aktiv organisiert. Dabei kommen die zuvor aufgezählten Einzelempfindungen nicht isoliert zum Tragen, sondern sie wirken gemeinsam als organisiertes Ganzes. Die Organisationsprinzipien sind angeboren und werden in Form der sogenannten **Gestalts-Gesetze** beschrieben. Die drei wichtigsten Gesetze sind:

- 1.) **Gesetz der Nähe und Ähnlichkeit:** Aufgrund der Nähe nimmt man in Abbildung 2.7 (a) Punktpaare wahr. Wegen der Ähnlichkeit der Symbole sieht man in Abbildung 2.7 (b) Spalten und nicht Zeilen.
- 2.) **Gesetz der Geschlossenheit:** In Abbildung 2.7 (c) sieht man bevorzugt schließende Klammern [] und nicht öffnende Klammern [].
- 3.) **Gesetz der guten Fortsetzung:** In Abbildung 2.7 (d) erkennt man eine "Gerade" und eine "Welle", obwohl auch andere Zerlegungen möglich sind.

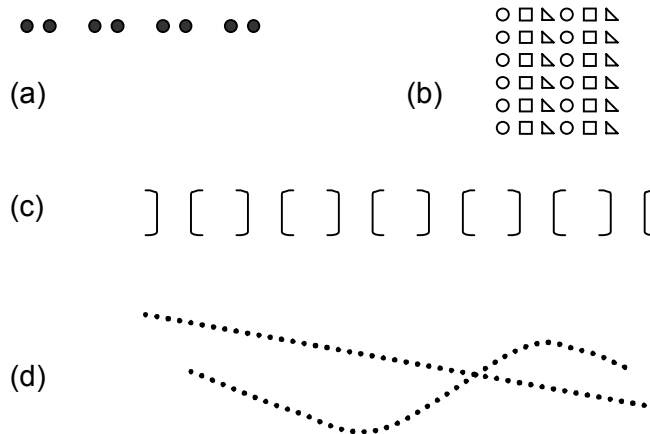


Abb. 2.7: Beispiele zu Gestalts-Gesetzen.

2.3 Zusammenfassung: Sicht des Wahrnehmungsprozesses

Seit Descartes und Helmholtz versuchte man, den gesamten visuellen Wahrnehmungsprozess durch die (passive) Kamera-Analogie zu erklären. Diese Sicht wurde z.B. von Wertheimer erweitert: "Die Sensoren der Retina liefern Informationen in Form von Reizen; diese werden passiv unter Einfluss von Assoziationen erkannt".

Im Gegensatz hierzu weiß man heute, dass dem Kamera-Prinzip des Auges ein aktiver Erkennungsvorgang nachgeschaltet ist, der die Reize aktiv unter Einfluss von Erfahrung und Wissen formiert bzw. gruppiert.

Dieser aktive Prozess lässt sich bewusst in die Irre führen, wie dies z.B. bei den optischen Täuschungen geschieht (siehe Abbildung 2.8). Dabei ist der Grund der Täuschungen vielfach unbekannt. Außerdem sind optische Täuschungen unabhängig vom Denken, denn auch erkannte Täuschungen verschwinden nicht!

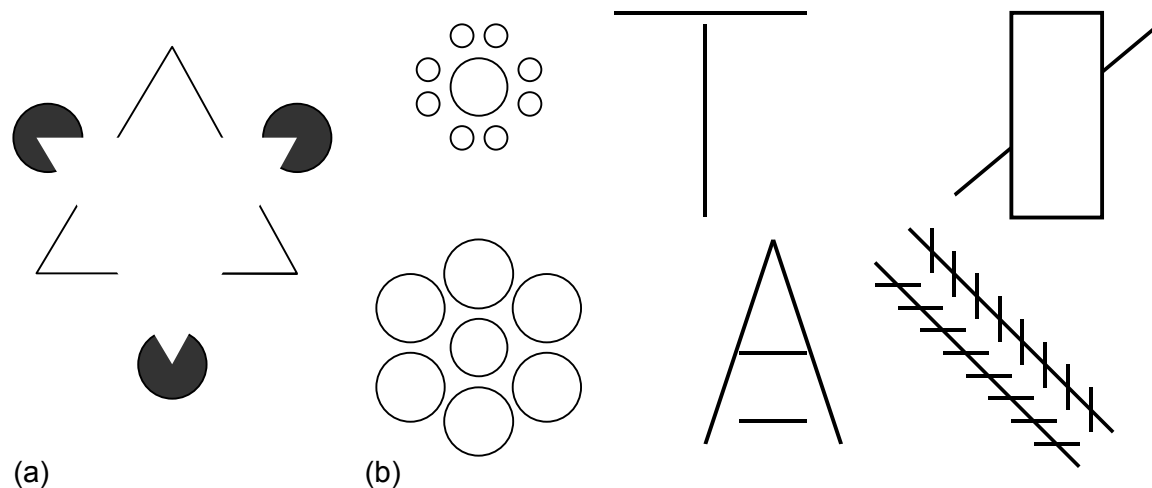


Abb. 2.8: Optische Täuschungen

(a) imaginäre Konturen;

(b) Täuschungen bzgl. Größe, Länge, Fortsetzung und Parallelität

Generell gilt für die Erkennung das folgende **Objekterkennungs-Paradoxon**: Um ein noch unbekanntes Objekt in einer noch unbekanntenen Szene zu identifizieren, muss das Objekt extrahiert, beschrieben und mit früher abgespeicherten (bekannten) Beschreibungen verglichen werden. Das unbekannte Objekt kann jedoch erst extrahiert werden, wenn seine Identität bekannt ist.

3 Vergleich bildhafter Information

In diesem Kapitel wird auf den Vergleich bildhafter Information eingegangen. Dies kann beispielsweise eingesetzt werden, um in Bildfolgen eine Änderungsentdeckung (z.B. zu Überwachungszwecken) durchzuführen. Eine anderer Zielsetzung ist das Durchsuchen von "großen" Bildern nach Stellen, die einem "kleinen", vorgegeben Bildausschnitt möglichst ähnlich sind. Auf diese Weise können beispielsweise markante Bildstellen, wie z.B. Linien, Kanten oder Ecken, aber auch Abbildungen einfacher Objekte, wie z.B. Muttern auf einem Fließband oder Blut- oder Gewebezellen in Abstrichen automatisch gefunden werden. Ein weiteres Anwendungsgebiet ist die Stereo- und Bildfolgenauswertung (siehe Kapitel 6 und 7), bei der markante Stellen aus einem Bild im jeweils anderen Stereobild bzw. einem späteren Bild der Bildfolge wieder gefunden werden sollen, um so Tiefen- bzw. Bewegungsinformation zu gewinnen.

Eine allgemeine Zielsetzung des Vergleichs ist die quantitative Bestimmung der Ähnlichkeit zweier Vorlagen, d.h. die Berechnung einer Maßzahl für die Ähnlichkeit bzw. Unähnlichkeit der beiden Bilder. Speziellere Zielsetzungen sind dann z.B. die Ermittlung der genauen Position, an der zwei vorgelegte Bilder am ähnlichsten sind, bzw. wo sie besonders unähnlich sind, d.h. wo Änderungen stattgefunden haben.

3.1 Bilddifferenzen

Unterschiede zwischen zwei gleich großen Bildern B und B' lassen sich direkt mit Hilfe der **Bilddifferenz** ermitteln. Neben dem Vergleich einzelner Bilder (z.B. geometrisch angepasste Luft- oder Satellitenbilder) kann beispielsweise auch eine Änderungsentdeckung in einer Bildfolge durchgeführt werden. So kann z.B. automatisch ein Alarm ausgelöst werden, wenn eine Überwachungskamera Änderungen detektiert und im Bereich der Bilddatenkompression (z.B. für die Video-Telefonie) erreicht man sehr hohe Kompressionsraten, indem man statt ganzer Bilder nur die Änderungen zwischen aufeinanderfolgenden Bildern überträgt.

Die Differenz zweier Bilder B und B' zur Ermittlung des Differenzbildes D wird durch die Subtraktion der Grauwerte g bzw. g' der beiden Bilder an den jeweils gleichen Bildpositionen ermittelt:

$$D = B - B' \quad \text{mit } g^*(i,j) = g(i,j) - g'(i,j) \quad \text{für alle } i, j. \quad (3-1)$$

Abschließend muss der Grauwertbereich von D noch linear in das zulässige Grauwertintervall transformiert werden.

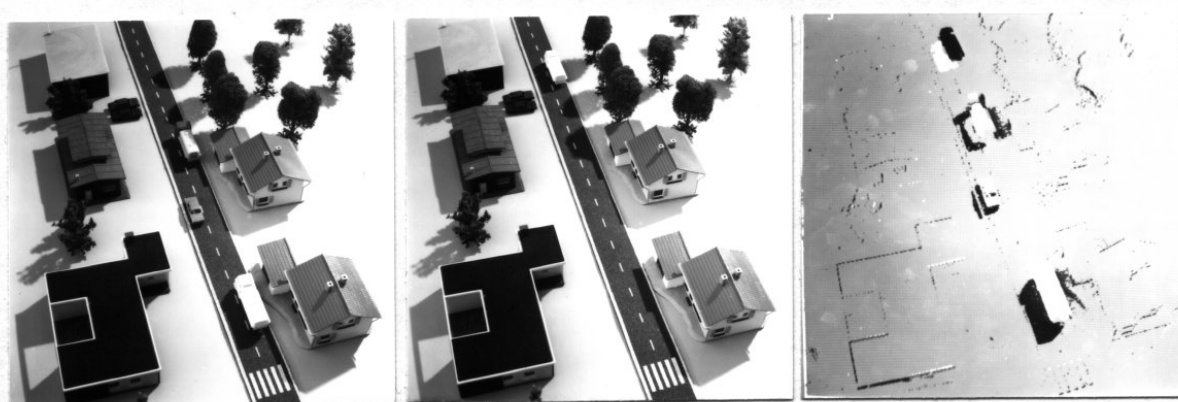


Abb. 3.1: Beispiel für eine Bilddifferenz in einer Bildfolge zur Änderungsentdeckung.

An Stellen, an denen B und B' übereinstimmen, ergibt sich für D ein konstanter "Hintergrund-Grauwert". An Stellen, an denen sich B und B' unterscheiden, treten vom Hintergrund-Grauwert verschiedene Grauwerte auf. Es muss berücksichtigt werden, dass z.B. als Folge des Rauschens, das jedem Bild überlagert ist, in der Regel auch für identische Bildteile kleine Änderungen angezeigt werden. Abbildung 3.1 zeigt ein Beispiel für eine Bilddifferenz.

3.2 Quantitativer Vergleich gleich großer Bilder

Die quantitative Bestimmung der Ähnlichkeit zweier Bilder, d.h. die Berechnung einer **Ähnlichkeits-Maßzahl** m , kann sich entweder direkt auf die Bildgrauwerte oder indirekt auf (aus den Bildern abgeleitete) Merkmale beziehen.

Bei der zweiten Möglichkeit werden je Bild bestimmte Merkmale gemessen, und die Merkmalswerte werden miteinander verglichen. Einfache Beispiele hierfür sind der Vergleich des mittleren Grauwertes und der Grauwert-Varianz der beiden Bilder. Da diese indirekten Vergleiche für viele Anwendungen zu grob sind, werden sie hier nicht weiter betrachtet.

Die in der Regel bessere, weil aussagekräftigere Vergleichsmöglichkeit ist der direkte, auf Grauwerte bezogene Vergleich. Dabei wird für dieses Teilkapitel vorausgesetzt, dass die beiden zu vergleichenden Bilder B und B' gleich groß sind; (Spaltenzahl $M =$ Zeilenzahl N).

Die Grauwerte von B und B' seien im weiteren mit g und g' bezeichnet. Verschiedene Maßzahlen lassen sich aus den Grauwert-Differenzen einander entsprechender Bildpositionen berechnen; ($i=0, \dots, M-1$ und $j=0, \dots, N-1$):

$$m_1 = \max_{i,j} (|g(i,j) - g'(i,j)|) \quad (3-2)$$

$$m_2 = \sum_j \sum_i |g(i,j) - g'(i,j)| \quad (3-3)$$

$$m_3 = \sum_j \sum_i (g(i,j) - g'(i,j))^2 \quad (3-4)$$

Die angegebenen Maßzahlen entsprechen nacheinander folgenden Metriken für den "Grauwert-Abstand" zweier Bilder: Maximum-, Summen- und (Quadrat der) Euklid-Metrik.

Bei einem Vergleich zweier Bilder ist die Übereinstimmung dann besonders groß, wenn die Werte der Maßzahlen m_1 bis m_3 besonders klein sind. Für identische Bilder erhält man jeweils den Wert Null.

Die Grauwerte von B und B' können auch multiplikativ miteinander verknüpft werden:

$$m_4 = \sum_j \sum_i (g(i,j) * g'(i,j)) \quad (3-5)$$

Die Maßzahl m_4 wird als **Korrelations-Koeffizient** bezeichnet. Im Gegensatz zu den vorherigen Maßzahlen zeigt sich die Ähnlichkeit zweier Bilder daran, dass m_4 sehr groß wird.

Leider ist die Maßzahl m_4 unter anderem von der mittleren Helligkeit der beiden Bilder abhängig. So liefern z.B. helle Bilder höhere Werte als dunkle, so dass ein objektiver Vergleich mit Hilfe von m_4 kaum möglich ist. Häufig wird statt dessen der **normierte Korrelations-Koeffizient** m_5 gewählt, der unabhängig von der Helligkeit der Bilder ist:

$$m_5 = \frac{\sum_j \sum_i (g(i,j) * g'(i,j))}{\sqrt{\sum_j \sum_i g(i,j)^2} * \sqrt{\sum_j \sum_i g'(i,j)^2}} \quad (3-6)$$

Die Werte von m_5 liegen zwischen 0 und 1, wobei der Wert 1 angenommen wird, wenn B und B' identisch sind.

Eine weitere Verbesserung berücksichtigt noch den mittleren Grauwert \bar{g} bzw. \bar{g}' der beiden Bilder, so dass die Maßzahl m_6 zwischen -1 (für invertierte Bilder) und 1 (für identische Bilder) liegt:

(3-7)

$$m_6 = \frac{\sum \sum ((g(i,j) - \bar{g}) * (g'(i,j) - \bar{g}'))}{\sqrt{\sum \sum (g(i,j) - \bar{g})^2} * \sqrt{\sum \sum (g'(i,j) - \bar{g}')^2}}$$

3.3 Quantitativer Vergleich ungleich großer Bilder

Bei den bisherigen Ansätzen waren die beiden Bilder B und B' gleich groß und wurden für den Vergleich genau "übereinander" gelegt. Im folgenden werden Ansätze vorgestellt, die ermitteln, ob ein im Vergleich zu B kleineres Bild B' ähnlich zu einem (achsenparallelen) Teilbild von Bild B ist. Falls eine solche Übereinstimmung festgestellt wird, so wird zusätzlich ihre genaue Position bestimmt. Diese Form des Vergleichs wird häufig als **Schablonen-Vergleich** (engl. **template matching**) bezeichnet; siehe Abbildung 3.2 (a). Um die Ähnlichkeit überall in B zu überprüfen, wird die Schablone B' systematisch z.B. von links oben nach rechts unten zeilenweise über B "geschoben". In jeder "Verschiebeposition" (i,j) wird die Ähnlichkeit zwischen B' und dem von B' aktuell abgedeckten Bereich von B bestimmt; siehe Abbildung 3.2 (b). Für diese Einzelvergleiche stehen die Ähnlichkeitsmaße m₁ bis m₆ zur Verfügung, wobei die Maximum-Bildungen bzw. Doppelsummen über B' und den abgedeckten Bereich von B laufen.

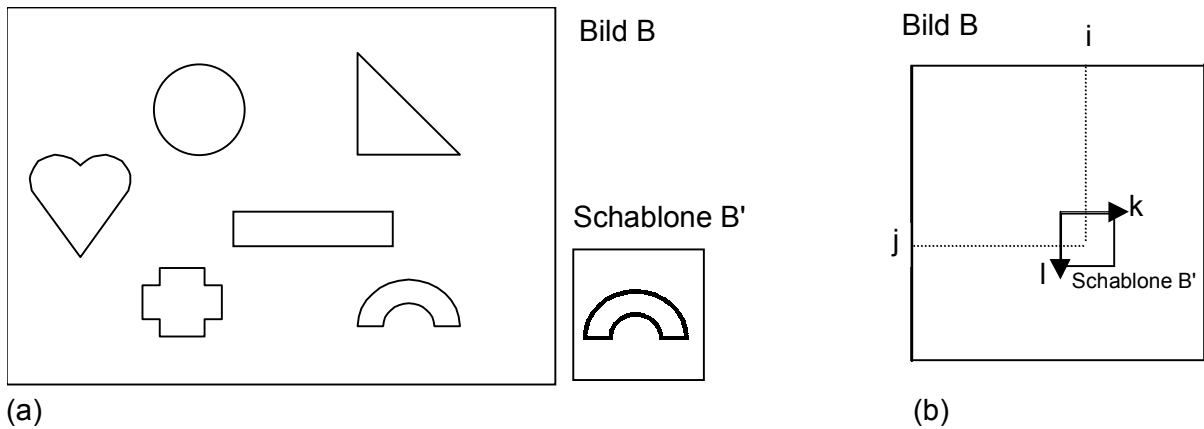


Abb. 3.2: Schema des Schablonen-Vergleichs.

Beispielsweise lautet die Formel zur Berechnung des Maßes m₄ zwischen einem Bild B und einer Schablone B' (mit ungerader Spaltenzahl K und ungerader Zeilenzahl L) an der "Verschiebeposition" (i,j):

(3-8)

$$m_4(i,j) = \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} g(i+k - \frac{K-1}{2}, j+l - \frac{L-1}{2}) * g'(k,l)$$

(Für K=L=3 entspricht diese Formel der Faltungsformel aus "Graphische Datenverarbeitung 1" Kapitel 6.1.2 und als Schablone können z.B. die dort angegebenen Kantendetektoren verwendet werden, die Kanten unterschiedlicher Ausrichtung modellieren).

Die Berechnungen, die sich auch auf alle anderen Maße übertragen lassen, liefern für jede "Verschiebeposition" einen Wert, der kennzeichnet, wie groß die Ähnlichkeit zwischen der Schablone B' und dem abgedeckten Bereich von B ist. (Dabei muss der Fall gesondert behandelt werden, dass die Schablone B' über den Rand von B "hinausragt").

Die Maßzahlen m(i,j) bilden eine Matrix, die z.B. für m₄ als **Korrelationsmatrix** bezeichnet wird. Wenn die Werte der Korrelationskoeffizienten in der dritten Dimension abgetragen werden, so erhält man ein sogenanntes **Korrelationsgebirge**; siehe Abbildung 3.3.

Wie beschrieben ist die Ähnlichkeit zwischen B und B' dann besonders groß, wenn die Korrelationskoeffizienten groß sind. Die Suche nach der Position mit der besten Übereinstim-

mung zwischen B und B' entspricht also der Suche nach dem absoluten Maximum in der Korrelationsmatrix bzw. dem Korrelationsgebirge.

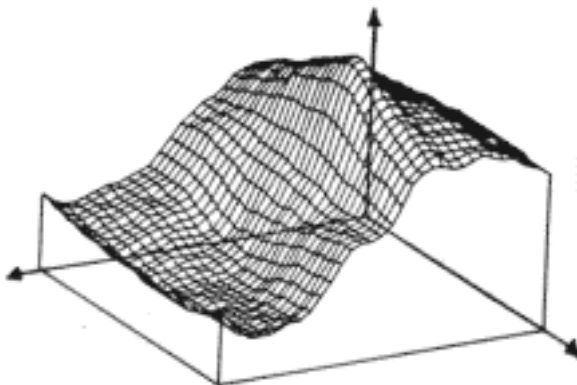


Abb. 3.3: Beispiel für ein Korrelationsgebirge.

Besonders bei der nicht-normierten Korrelation nach m_4 sind im Korrelationsgebirge häufig weitere, deutlich ausgeprägte Maxima vorhanden. Diese können im schlimmsten Fall sogar das "richtige" Maximum übertreffen.

Weiter sei darauf hingewiesen, dass die aufgeführten Ansätze nur funktionieren, wenn eine Übereinstimmung zwischen B' und einem Teilbild von B ausschließlich durch eine Translation erreicht werden kann. Bereits kleine gegenseitige Maßstabsänderungen oder Verdrehungen der in B bzw. B' dargestellten "gleichen" Muster beeinträchtigen die Maßzahlen empfindlich.

Die vorgestellten Ansätze mit beliebigem B und B' werden auch als **Kreuzkorrelation** bezeichnet. Im Gegensatz dazu spricht man von **Autokorrelation**, wenn B' als Teilbild aus B entnommen wird. Die Autokorrelation wird z.B. eingesetzt um festzustellen, ob ein Muster aus B an mehreren Stellen im selben Bild B auftritt. Bei der Texturanalyse (siehe "2D- und 3D-Bildverarbeitung") wird die Autokorrelation dazu verwendet, die Periodizität des Auftretens des Grundmusters der Textur zu bestimmen.

Weiter ist zu erwähnen, dass die Korrelation zwischen zwei Bildern B und B' statt im Ortsraum auch (nach entsprechenden Fourier-Transformationen) im Frequenzraum durchgeführt werden kann (**Faltungssatz**).

Schlußbemerkung:

Wegen der genannten Schwierigkeiten (Neben-Maxima, Empfindlichkeit gegen Rotation, Skalierung usw.) wird die Korrelation heute im wesentlichen dort eingesetzt, wo einfache Muster wie Kanten und Linien im Rahmen einer Kanten- bzw. Liniensextraktion aus einem Bild herauszufiltern sind. Ebenfalls ausgereift ist der Einsatz im Bereich der Stereobildauswertung z.B. bei der Auswertung von Stereoluftbildern (Passpunktsuche, d.h. Suche nach einander entsprechenden Stellen in den Bildern zur nachfolgenden Ermittlung von Geländehöhendaten) und die "Verfolgung" (d.h. das Wiederfinden) von markanten Objektstellen in Bildfolgen zur Bewegungsanalyse. Für eine allgemeine Objekterkennung (insbesondere bei komplexen oder dreidimensionalen Objekten) werden in der Regel andere Vorgehensweisen eingesetzt.

4 Konturorientierte Segmentierung, Kontur-Repräsentationen

Für viele Aufgaben im Bereich "Computer Vision" ist es erforderlich, die Bilddaten teilweise oder sogar vollständig in ihre relevanten Teile, die sogenannten **Regionen** zu zerlegen, um so z.B. die im Bild enthaltenen Objekte ermitteln zu können.

Für diese sogenannte Segmentierung gibt es zwei duale Ansätze: das flächen- und das konturorientierte Vorgehen. Während bei der **flächenorientierten Segmentierung** versucht wird, zu jedem homogenen Bildbereich (der z.B. einheitlich eingefärbt oder texturiert ist) die Menge aller zugehörigen Pixel zu erhalten, wird bei der **konturorientierten Segmentierung** nur der "Rand" dieser Bereiche ermittelt. Für den Rechner sind beide Vorgehensweisen gleichwertig; der Mensch hingegen reagiert eher auf Diskontinuitäten (Kanten) als auf Kontinuitäten (homogene Bildbereiche); siehe Kapitel 2.

Der flächenorientierte Ansatz wird in "2D- und 3D-Bildverarbeitung" genauer vorgestellt; hier wird nur die konturorientierte Vorgehensweise behandelt. Bevor die einzelnen Schritte des konturorientierten Ansatzes genauer besprochen werden, soll hier zunächst ein kurzer Überblick gegeben werden.

Zuerst werden mit Hilfe von Kantendetektoren überall im Bild **Kantenelemente**, d.h. kurze Abschnitte von Kanten gesucht. Diese Kantendetektion liefert in der Regel breite Streifen nicht zusammenhängender Kantenelemente, d.h. es werden Kantenelemente senkrecht zur Längsausrichtung der Kante mehrfach nebeneinander detektiert. Weiterhin weisen diese Kantestreifen häufig Lücken auf und es gibt meist noch zahlreiche isoliert liegende Kantenelemente im Hintergrund bzw. innerhalb von Objektflächen; siehe Abbildung 4.1.

Die weitere Aufgabe der konturorientierten Segmentierung besteht nun darin, aus den breiten Streifen nicht zusammenhängender Kantenelemente eine (geschlossene, ein Pixel breite) Folge von Konturpixeln zu gewinnen.

Da die Ermittlung linienhafter Bildstrukturen (z.B. Straßen in Luftbild) prinzipiell dasselbe Vorgehen und dieselben Probleme aufweist, werden diese beiden Themen im Folgenden gemeinsam behandelt.

4.1 Kanten- und Liniendetektion

Kanten- bzw. **Liniendetektoren** sind lokale Bildoperatoren, die ermitteln, wo im Bild Kanten- bzw. Linienelemente auftreten.

Ein Weg, Kanten- und Liniendetektoren zu erzeugen, besteht darin, kurze Kanten- und Linienstücke unterschiedlicher Ausrichtung mit Hilfe von kleinen Bildausschnitten zu modellieren, und dann z.B. mit den Korrelations-Maßen m_4 bis m_6 aus Kapitel 3 die Stellen im Bild zu suchen, an denen die Bildfunktion mit dem Kanten-/Linienmodell übereinstimmt.

Im Folgenden sind vier **Kanten-Schablonen**, mit denen verschiedene Kanten-Richtungen unterschieden werden können, sowie vier **Linien-Schablonen** unterschiedlicher Richtungen für helle Linien dargestellt. Ein Ergebnis der Anwendung der diagonal (/) verlaufenden Kantenschablone ist in Abbildung 4.1 (b) gezeigt. Mittelgrau eingefärbte Bereiche zeigen an, dass hier keine Kante vorhanden ist. Je heller eine Bildposition eingefärbt ist, je besser ist die Übereinstimmung zwischen der Bildstelle und der verwendeten Schablone. Eine Bildposition ist um so dunkler eingefärbt, je besser die Bildstelle zur invertierten Schablone (d.h. hier einer um 180 Grad gedrehten Schablone) passt.

(4-1)

Kanten-Schablonen:	-1 0 1 -1 0 1 -1 0 1	-1 -1 -1 0 0 0 1 1 1	0 -1 -1 1 0 -1 1 1 0	-1 -1 0 -1 0 1 0 1 1
Linien-Schablonen:	-1 2 -1 -1 2 -1 -1 2 -1	-1 -1 -1 2 2 2 -1 -1 -1	2 -1 -1 -1 2 -1 -1 -1 2	-1 -1 2 -1 2 -1 2 -1 -1

Die "fehlenden" vier Kantenausrichtungen bzw. die vier Linien-Schablonen für dunkle Linien müssen nicht separat betrachtet werden, da diese Fälle von den oben gezeigten Schablonen ebenfalls detektiert werden (siehe dunkle Stellen in Abbildung 4.1 (b)) und z.B. am Vorzeichen des Ergebnisses zu erkennen sind.

Häufig werden auch größere Kanten- bzw. Linien-Schablonen von z.B. 7*7 oder 9*9 Pixeln verwendet. Je größer die Schablonen sind, desto mehr Richtungen können unterschieden werden.

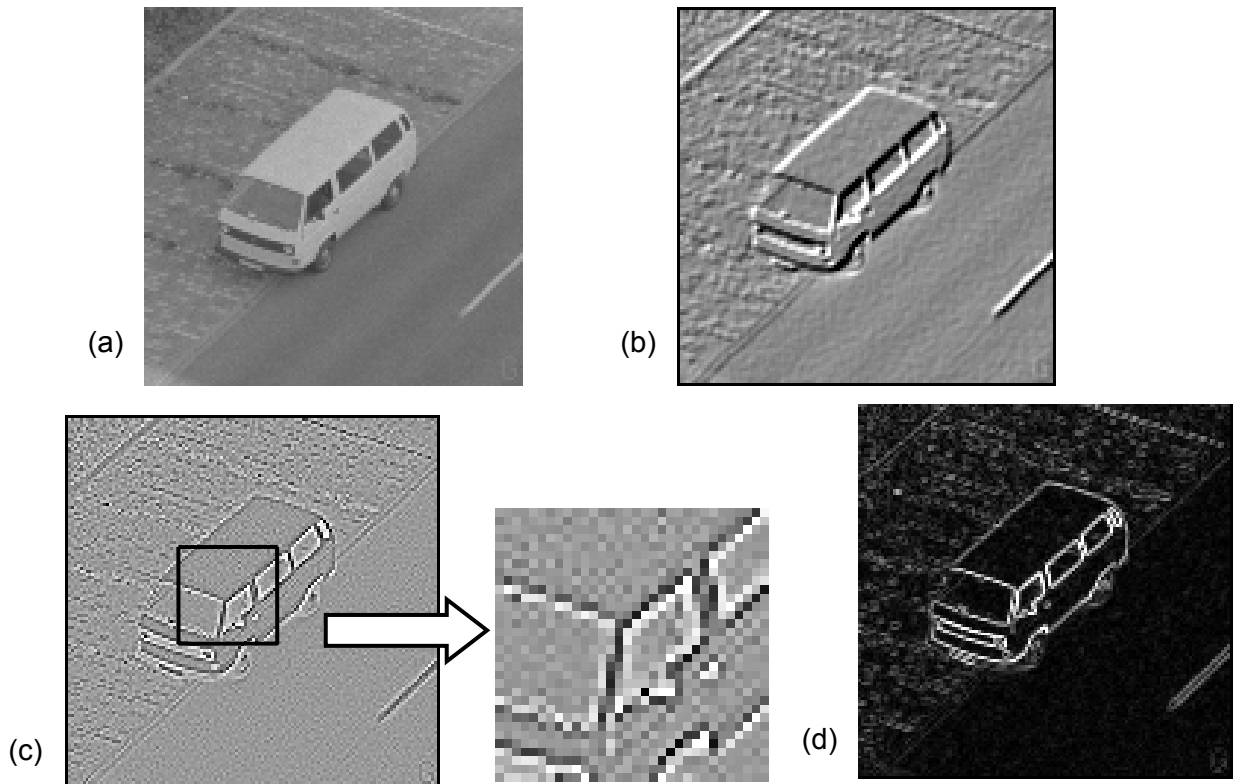


Abb. 4.1: Beispiele für die Anwendung von Kantendetektoren:

- (a) Originalbild,
- (b) Korrelation mit diagonal (/) verlaufender Kanten-Schablone,
- (c) Laplace-Operator,
- (d) Robert's-Kreuz-Operator.

Ein anderer Weg zur Herleitung von Kantendetektoren erfolgt mit Hilfe der (**partiellen**) **Ableitung** der Bildfunktion. Die Ableitung einer Funktion gibt die Steigung der Funktion im betrachteten Punkt an, und auf dieselbe Weise kann man in der zweidimensionalen Bildfunktion, die hier zweckmäßig als Grauwertgebirge betrachtet wird, den "Anstieg" zwischen unterschiedlich hellen Bildbereichen, d.h. Bereichen mit unterschiedlich hohen Grauwerten ermitteln. Ausgeprägte Kanten entsprechen "steilen Abhängen" im Grauwertgebirge, d.h. die partielle Ableitung senkrecht zum Verlauf einer Kante hat einen hohen Wert.

Im weiteren wird statt des zweidimensionalen Bildes zunächst nur ein eindimensionaler Grauwertverlauf entlang einer Linie durch das Bild betrachtet. Abbildung 4.2 zeigt als Beispiel das Grauwertprofil g entlang einer Linie, die in einem Bild ein helles Objekt auf dunklem Hintergrund schneidet.

Wenn man die 1D-Grauwertfunktion $g(i)$ als analoge Funktion $f(x)$ ansehen würde (siehe Abbildung 4.2, linke Seite), so ergäbe sich die erste (partielle) Ableitung $f'(x) = \frac{\partial f}{\partial x}$ zu:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (\text{für } h \rightarrow 0).$$

Im diskreten Fall (siehe Abbildung 4.2, rechte Seite) kann h nicht kleiner als 1 werden und für $h=1$ erhält man die sogenannte Rechts-Ableitung der Grauwertfunktion g , die in diesem Falle als **Rechts-Differenz** $\Delta_r g$ bezeichnet wird:

$$\Delta_r g(i) = \frac{g(i+1) - g(i)}{1} = g(i+1) - g(i). \quad \text{und dies entspricht der 1D-Schablone: } \begin{matrix} & i-1 & i & i+1 \\ \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \end{matrix} \quad (4-2)$$

Entsprechend erhält man die **Links-Differenz** $\Delta_l g$ durch Berücksichtigung des zentralen und des links davon gelegenen Bildpunktes. Wenn man die Elemente links und rechts der betrachteten Stelle i berücksichtigt, so erhält man die **symmetrische Differenz** $\Delta_s g$:

$$\Delta_l g(i) = \frac{g(i) - g(i-1)}{1} = g(i) - g(i-1) \quad \text{bzw.} \quad \Delta_s g(i) = g(i+1) - g(i-1). \quad (4-3)$$

(entspricht: $\begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$ bzw.: $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$)

Die symmetrische $\Delta_s g$ liefert genau die Zeilen der Kanten-Schablone zur Detektion senkrecht verlaufender Kanten in (4-1).

Da die Bildfunktion eine zweidimensionale Funktion ist, muss man die drei Differenzen korrekter Weise als $\Delta_l g(i,j)$, $\Delta_r g(i,j)$ und $\Delta_s g(i,j)$ schreiben, wobei j konstant ist und die jeweils betrachtete Bildzeile angibt. Alle drei Differenzen liefern eine partielle Ableitung der Bildfunktion g in waagrechter Richtung, die im weiteren als $\Delta_{\text{hori}} g$ ("hori" für horizontal) bezeichnet wird. Entsprechend erhält man bei spaltenweiser Betrachtung die partiellen Ableitungen in vertikaler Richtung $\Delta_o g(i,j)$, $\Delta_u g(i,j)$ und $\Delta_s g(i,j)$ ("o" für "oben", "u" für "unten" und "s" wieder für "symmetrisch"), die mit $\Delta_{\text{veri}} g(j)$ zusammengefasst werden.

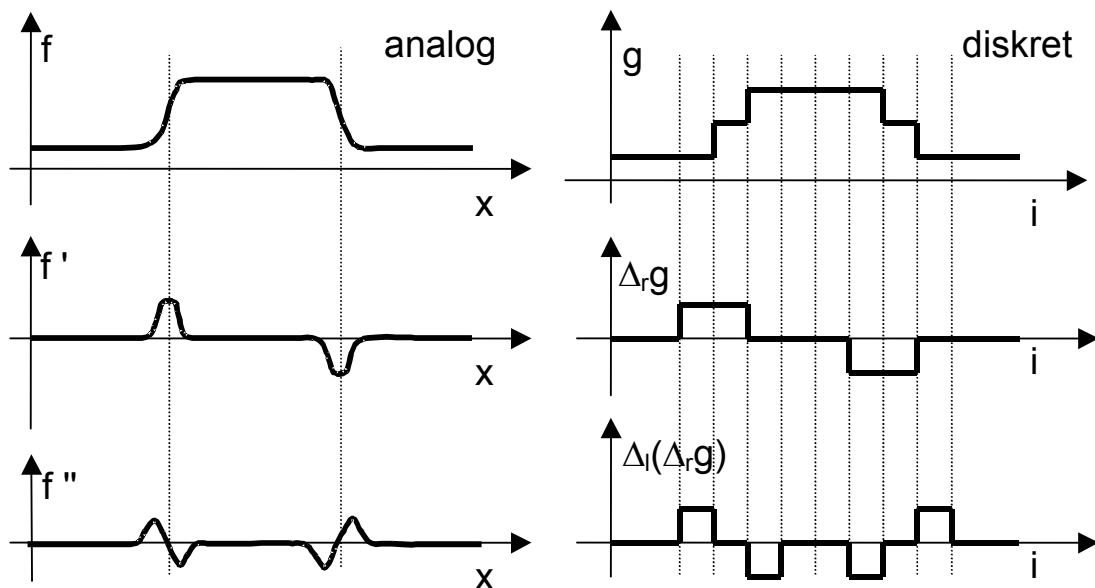


Abb. 4.2: Beispiel für die erste und zweite Ableitung einer analogen Funktion f bzw. einer diskreten 1D-Bildfunktion g .

Aus den partiellen Ableitungen $\frac{\partial f}{\partial x}$ und $\frac{\partial f}{\partial y}$ einer analogen zweidimensionalen Funktion f erhält man den sogenannten **Gradienten** $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)^t$. Der Gradient ist ein Vektor, der in Richtung der größten Steigung zeigt. Im Falle einer diskreten zweidimensionalen Bildfunktion g zeigt der Gradient in die Richtung des steilsten Anstiegs im Grauwertgebirge. Der Gradient einer Bildfunktion g ist definiert als:

$$\nabla g(i,j) = (\Delta_{\text{hori}} g(i,j), \Delta_{\text{veri}} g(i,j))^t. \quad (4-4)$$

Der Betrag des Gradienten (d.h. anschaulich die Stärke einer Kante) und seine Richtung ergeben sich zu:

$$\sqrt{|\Delta_{\text{hori}}g(i,j)|^2 + |\Delta_{\text{veri}}g(i,j)|^2} \quad \text{bzw.} \quad \arctan(\text{veri}g(i,j)/\text{hori}g(i,j)). \quad (4-5)$$

Die zweite Ableitung (d.h. die Ableitung der Ableitung) einer Funktion stellt ebenfalls einen Kantendetektor dar: die Kante ist am "Nulldurchgang" zu erkennen; (siehe Abbildung 4.2 unten). Um die seitliche "Verschiebung" der Kante durch die Rechts- bzw. Links-Differenz im diskreten Fall zu kompensieren, wählt man für die nochmalige Ableitung die jeweils andere Differenz und erhält z.B.:

$$\Delta_l(\Delta_r g(i)) = -\Delta_r g(i-1) + \Delta_r g(i) = -g(i) + g(i-1) + g(i+1) - g(i) = g(i-1) - 2g(i) + g(i+1);$$

(dies entspricht der 1D-Schablone:

1	-2	1
---	----	---

)

Aus den zweifachen partiellen Ableitungen $\frac{\partial^2 f}{\partial x^2}$ und $\frac{\partial^2 f}{\partial y^2}$ bzw. $\Delta_l(\Delta_r g(i,j))$ in horizontaler und $\Delta_o(\Delta_u g(i,j))$ in vertikaler Richtung bildet man als zweidimensionalen Kantendetektor den **Laplace-Operator**, der Kanten aller Ausrichtungen erkennt; siehe Abbildung 4.1 (c):

$$\nabla^2 f(x,y) = (\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}) \quad \text{bzw. diskret}$$

$$\nabla^2 g(i,j) = (\Delta_r g(i,j) + \Delta_o g(i,j)) = (g(i-1,j) - 2g(i,j) + g(i+1,j)) + (g(i,j-1) - 2g(i,j) + g(i,j+1)) = g(i,j-1) + g(i-1,j) - 4g(i,j) + g(i+1,j) + g(i,j+1)$$

Dies entspricht einer Überlagerung (Addition) der 1D-Schablonen

1	-2	1
---	----	---

 und

1	-2	1
---	----	---

, und liefert somit genau den Laplace-Operator aus "Graphische Datenverarbeitung 1", Kapitel 6, (6-16):

0	1	0
1	-4	1
0	1	0

Abschließend sollen noch zwei weitere Kanten-Detektoren kurz vorgestellt werden.

Der **Robert's-Kreuz-Operator** ist ein besonders einfacher Kantendetektor, der Kanten aller Ausrichtungen ermittelt, indem er die Grauwerte eines 2*2-Pixel-Blockes auswertet:

$$\text{Rob}(i,J) = |g(i,j) - g(i+1,j+1)| + |g(i,j+1) - g(i+1,j)|. \quad (4-8)$$

Der Robert's-Kreuz-Operator liefert Werte zwischen 0 und $2g_{\text{max}}$ (mit g_{max} maximal möglicher Grauwert), wobei niedrige Ergebniswerte auf homogene Bildbereiche hinweisen. Hohe Werte resultieren aus stark unterschiedlichen Grauwerten, was Indiz für das Vorhandensein einer Kante sein kann.

Augrund der geringen Größe seines Einflussbereiches von 2*2 Pixeln reagiert der Robert's-Kreuz-Operator sehr empfindlich auf Störungen der Bildfunktion, wie z.B. Rauschen. In Abbildung 4.1 (d) ist ein Ergebnis des Robert's-Kreuz-Operators dargestellt.

Als letztes soll noch der **Kirsch-Operator** betrachtet werden, der ebenfalls acht verschiedene Kanten-Richtungen unterscheiden kann.

Für die Berechnung an einer Bildposition (i,j) werden für $n=0$ bis $n=7$ jeweils zwei Anteile aus drei bzw. fünf zusammenhängenden Nachbar-Pixeln "gemessen". Mit

$$S_n(i,j) = 1/3 * (A_n + A_{n+1} + A_{n+2}) \quad (n \text{ modulo } 8) \quad \text{und}$$

$$T_n(i,j) = 1/5 * (A_{n+3} + \dots + A_{n+7}) \quad (n \text{ modulo } 8)$$

(4-9)

A ₃	A ₂	A ₁
A ₄	(i, j)	A ₀
A ₅	A ₆	A ₇

ergibt sich die "Stärke" der Kante, d.h. der Betrag des Kontrast-Gradienten zu

(4-10)

$$Kir(i,j) = \max(|S_n(i,j) - T_n(i,j)|)$$

wobei das Maximum über alle n von 0 bis 7 gebildet wird. Die Richtung der Kante an der betrachteten Stelle (i,j) entspricht dem Wert von n , für den dieses Maximum angenommen wird. Die Ergebnisse des Kirsch-Operators ähneln im Aussehen denen des Robert's-Kreuz-Operators, wobei der Kirsch-Operator aber unempfindlicher auf Bildstörungen reagiert und zusätzlich noch die Richtung der Kantenelemente liefert; (Beispielbilder zum Kirsch-Operator sind als Farbbilder am Ende dieser Datei angefügt, aber im Ausdruck nicht enthalten).

Auf weitere Kantendetektoren, wie z.B. den zu den Rangfolge-Operatoren gehörenden Spannweiten-Operator (siehe "Graphische Datenverarbeitung 1") oder den sehr aufwändigen Canny-Operator wird hier nicht eingegangen.

Obwohl das Ergebnis der Kanten-/Liniendetektion häufig schon ein fertiges Ergebnis vorspiegelt, ist zu beachten, dass alle Elemente unabhängig voneinander vorliegen und die oben aufgezählten Schwierigkeiten auftreten: breite Streifen, Lücken und isolierte Elemente.

4.2 Nachverarbeitung der Ergebnisse der Kanten-/Liniendetektion

Die Nachverarbeitung soll die durch Mehrfachdetektion entstandenen breiten Streifen auf ein Pixel breite Striche reduzieren, Lücken schließen und isoliert liegende Elemente löschen.

Im Folgenden wird vorausgesetzt, dass das Ergebnis der Kanten- bzw. Linien-Detektion als Ergebnisbild mit Grauwerten g vorliegt, die angeben, wie ausgeprägt an der betreffenden Stelle eine Kante/Linie vorhanden ist: niedrige Werte stehen für "keine Kante/Linie" und hohe Werte stehen für "gut ausgeprägte Kante/Linie". (Bei der Verwendung von Schablonen müssen die Ergebnisse erst noch entsprechend aufbereitet werden, da ausgeprägte Kanten/Linien sowohl durch sehr helle als auch durch sehr dunkle Bildpositionen gekennzeichnet werden; siehe Abbildung 4.1 (b)).

4.2.1 Binarisierung

Die einfachste Möglichkeit zur Nachverarbeitung eines Ergebnisbildes ist eine Binarisierung, wie sie auch als Vorstufe zur flächenorientierten Segmentierung eingesetzt werden kann. Bei einer **Binarisierung** wird ein Eingabebild B mit Grauwerten $g(i,j)$ mit Hilfe eines Schwellwertes $schwelle$ in ein binäres Ausgabebild B' mit Grauwerten $g'(i,j)$ überführt, wobei $g'(i,j)$ nur die beiden Werte g_1' bzw. g_2' annehmen kann:

(4-11)

$$g'(i,j) = \begin{cases} g_1' & \text{falls } g(i,j) \leq schwelle \\ g_2' & \text{falls } g(i,j) > schwelle \end{cases} \quad (\text{Häufig wird } g_1' \text{ als Schwarz und } g_2' \text{ als Weiß gewählt}).$$

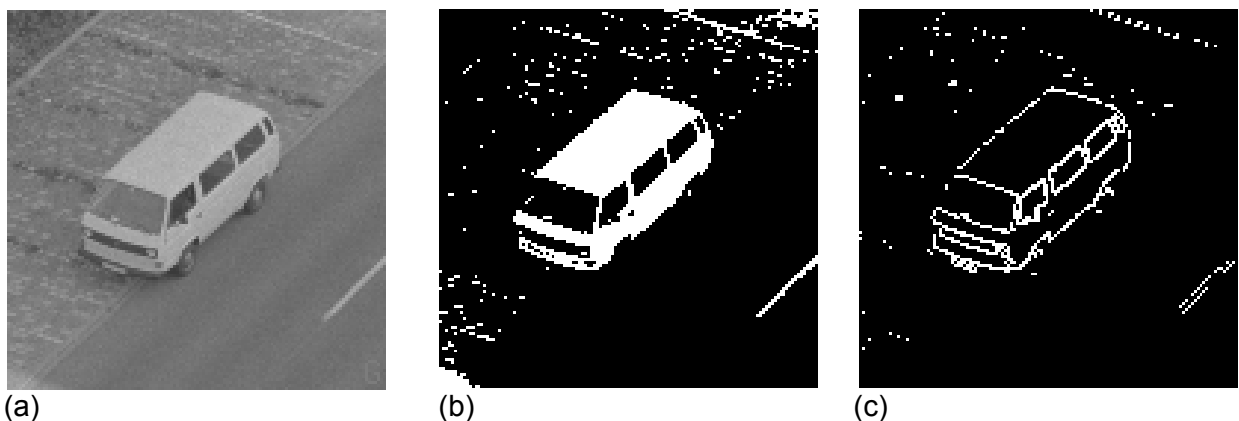


Abb. 4.3: Beispiele für Binarisierungen:

- (a) Originalbild,
- (b) Binarisierung des Originalbildes;
- (c) Binarisierung des Ergebnisses der Kanten-Detektion aus Abbildung 4.1 (d).

(Die Binarisierung gehört genau wie die lineare Grauwerttransformation (siehe "Graphische Datenverarbeitung 1") zu den **Punktoperatoren**, bei denen dieselbe Berechnungsvorschrift ohne Berücksichtigung von Nachbarpixeln auf jeden Bildpunkt angewendet wird.)

Bei günstiger Wahl des Schwellwertes (z.B. unter Berücksichtigung des Histogramms) erlaubt die Binarisierung bei geeigneten Grauwertbildern eine gute Trennung von Objekt und Hintergrund; siehe Abbildung 4.3 (b). Bei Ergebnisbildern einer Kanten-/Linien-Detektion sind nach einer Binarisierung nur noch die klar ausgeprägten Kanten-/Linienelemente vorhanden; siehe Abbildung 4.3 (c). Nach wie vor gilt, dass die übrig gebliebenen Pixel unabhängig voneinander sind, d.h. dass die weißen Pixel der Busfläche in (b) nicht "wissen", dass sie zu **einer** Zusammenhangskomponente gehören und dass die Konturpixel in (c) ihre benachbarten Konturpixel nicht "kennen".

Die Anwendung der Binarisierung auf das Ergebnisbild einer Kanten-/Linien-Detektion ist einfach aber unflexibel und kann unerwünschter Weise z.B. auch lange, aber nur schwach ausgeprägte Kanten-/Linien löschen. In der Regel entstehen keine ein Pixel breiten Linien von Konturpixeln und es können neue, zusätzliche Lücken entstehen.

4.2.2 Nicht-Maximum-Unterdrückung

Die oben genannten Nachteile vermeidet die sogenannte **Nicht-Maximum-Unterdrückung**, bei der quer zur Richtung der Kante/Linie alle Elemente mit nicht maximalem Wert gelöscht werden. Im Idealfall bleibt nur noch eine einpixel breite Linie erhalten.

4.2.3 Verdünnung

Eine weitere Möglichkeit stellen Verfahren dar, bei denen durch sukzessive **Erosion** (siehe "Graphische Datenverarbeitung 1", Kapitel 6) abwechselnd auf allen Seiten der breiten Streifen, die im weiteren als Region bezeichnet werden, je eine "Bildpunkt-Schicht abgetragen" wird. Dabei ist aber zu berücksichtigen, dass zusätzlich zu den vorhandenen Lücken neue Unterbrechungen entstehen können, wenn an schmalen oder in ihrer Breite variierenden Streifen zu viele Schichten abgetragen werden. Diesen Nachteil vermeidet die sogenannte Verdünnung.

Bei einer **Verdünnung** werden sukzessive immer wieder Randschichten des Objektes abgetragen, bis nur noch das „Skelett“ des Objektes übrig ist. Dabei wird aber im Gegensatz zur Rangfolge-Operation Erosion beachtet, dass eine Region nicht in mehrere Regionen zerfällt oder verschwindet, und dass ein Pixel breite Linien nicht verkürzt werden.

Um insbesondere das Verschwinden einer kleinen oder nur zwei Pixel breiten Region zu verhindern, darf das iterative Abschälen nicht gleichzeitig auf allen Seiten der Region vorgenommen werden. Statt dessen werden bei jedem Iterationsschritt z.B. zuerst nur die linken, dann nur die rechten, anschließend nur die oberen und zuletzt nur die unteren **Randpixel** der Region gelöscht. Randpixel sind Pixel, die zur Region gehören und in ihrer N4-Nachbarschaft ein Pixel besitzen, das nicht zur Region gehört.

Um zu verhindern, dass eine Region in zwei oder mehr Regionen zerfällt, dürfen nur solche Pixel gelöscht werden, die sogenannte **einfache Punkte** sind.

Um zu entscheiden, ob ein Randpunkt P ein N4- oder ein N8-einfacher Punkt ist, wird seine N8-Nachbarschaft betrachtet. Enthält diese Nachbarschaft genau eine N4- bzw. N8-ZHK (siehe "Integrierte Lehrveranstaltung"), so ist der betreffende Punkt ein N4- bzw. N8-einfacher Punkt und darf gelöscht werden. Andernfalls "verbindet" der Punkt zwei oder mehr ZHKs und seine Löschung würde die Region in zwei oder mehr Regionen zerlegen.

Die Entscheidung, ob ein Punkt ein einfacher Punkt ist, erfolgt am einfachsten mit Hilfe der Berechnung der **Connectivity-Numer**. Dazu nummeriert man die N8-Nachbarschaft eines Pixels P wie rechts gezeigt durch. Die Nachbarn D_0 bis D_7 sollen den Wert Eins haben, wenn die betreffende Position innerhalb der Region liegt; andernfalls sollen sie den Wert Null haben. Die Connectivity-Number CON_4 für einen N4-Zusammenhang ist gegeben durch:

D_3	D_2	D_1
D_4	P	D_0
D_5	D_6	D_7

(4-12)

$$CON_4 = D_0 + D_2 + D_4 + D_6 - D_0 * D_1 * D_2 - D_2 * D_3 * D_4 - D_4 * D_5 * D_6 - D_6 * D_7 * D_0.$$

Die ersten vier Summanden dieser Formel geben an, an wie vielen Seiten von P eine potentielle ZHK liegt. Die restlichen (negativen) Summanden korrigieren dieses Ergebnis, falls z.B. zwei dieser Kandidaten zur selben ZHK gehören, d.h. wenn Sie über das Eckpixel der Nachbarschaft von P miteinander verbunden sind.

Ein Punkt ist ein N4-einfacher Punkt, der gelöscht werden darf, wenn gilt: $CON_4 = 1$.

Zur Berechnung der N8-Connectivity-Number CON_8 wird ein duales Vorgehen gewählt. Zunächst führt man ein: $\underline{D}_i = 1 - D_i$. ($i \in \{0, \dots, 7\}$) und definiert dann entsprechend zu oben:

(4-13)

$$CON_8 = \underline{D}_0 + \underline{D}_2 + \underline{D}_4 + \underline{D}_6 - \underline{D}_0 * \underline{D}_1 * \underline{D}_2 - \underline{D}_2 * \underline{D}_3 * \underline{D}_4 - \underline{D}_4 * \underline{D}_5 * \underline{D}_6 - \underline{D}_6 * \underline{D}_7 * \underline{D}_0.$$

Und wie oben gilt, dass ein Punkt ein N8-einfacher Punkt ist, der gelöscht werden darf, wenn gilt: $CON_8 = 1$.

Beispiele:

0	1	1
0	P	1
0	0	0

P ist N4-einfach
P ist N8-einfach

0	1	1
0	P	1
0	1	0

P ist nicht N4-einfach
P ist N8-einfach

0	1	1
0	P	1
1	0	0

P ist N4-einfach
P ist nicht N8-einfach

0	1	1
0	P	0
0	1	0

P ist nicht N4-einfach
P ist nicht N8-einfach

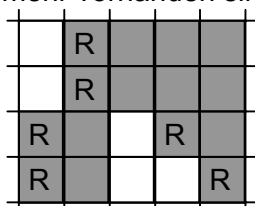
Die Verkürzung ein Pixel breiter Linien vermeidet man, indem man nur solche Randpixel löscht, die keine sogenannten **Endpunkte** sind. Ein N4- bzw. N8-Endpunkt ist ein Punkt der Region, der in seiner N4- bzw. N8-Nachbarschaft nur genau ein weiteres Pixel der Region besitzt.

Im Folgenden ist die N4- bzw. N8-Verdünnung noch einmal mit Hilfe eines Pseudo-Code Programmstückes zusammengefasst:

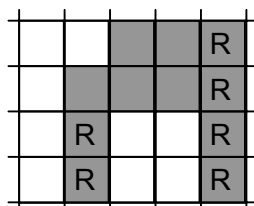
```

{
  Für alle 4 Richtungen (linke, rechte, obere dann untere Randpunkte) {
    Für alle Bildpunkte {
      Falls aktueller Bildpunkt ein N4/8-Randpunkt ist {
        Lösche den aktuellen Bildpunkt, falls er ein N4/8-einfacher
        Punkt, aber kein N4/8-Endpunkt ist;
      }
    }
  }
} Solange noch Löschungen möglich sind;
    
```

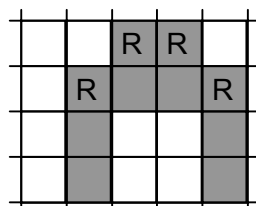
Das folgende Beispiel demonstriert den Ablauf einer N4- und einer N8-Verdünnung. Alle Pixel außerhalb des gezeigten Bildausschnittes sind (weiße) Hintergrundpixel. Es sind nur die Schritte gezeigt, in denen noch Änderungen auftreten. Das Verfahren darf aber erst abbrechen, wenn in einem vollständigen Durchgang über alle vier Richtungen keine Änderungen mehr vorhanden sind:



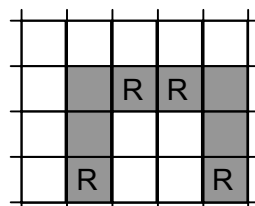
Originalbild-Ausschnitt mit markierten linken Randpunkten; das Objekt ist dunkel gezeichnet.



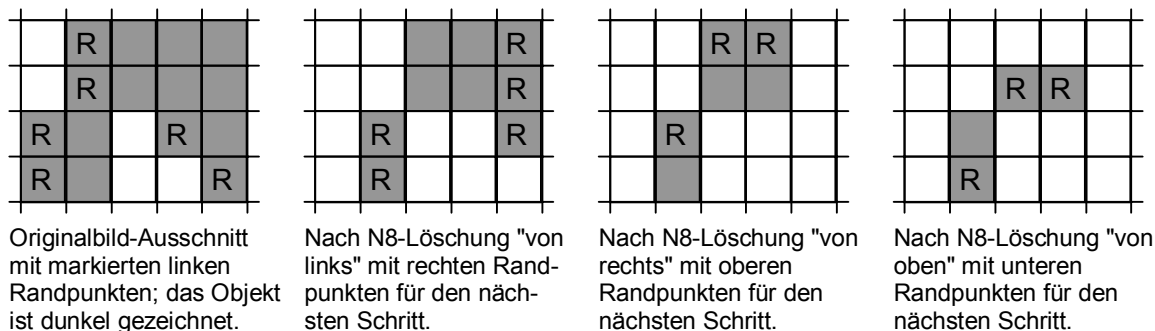
Nach N4-Löschung "von links" mit rechten Randpunkten für den nächsten Schritt.



Nach N4-Löschung "von rechts" mit oberen Randpunkten für den nächsten Schritt.



Nach N4-Löschung "von oben" mit unteren Randpunkten für den nächsten Schritt.



In Abbildung 4.4 ist das Ergebnis einer Verdünnung für ein reales Beispiel dargestellt.

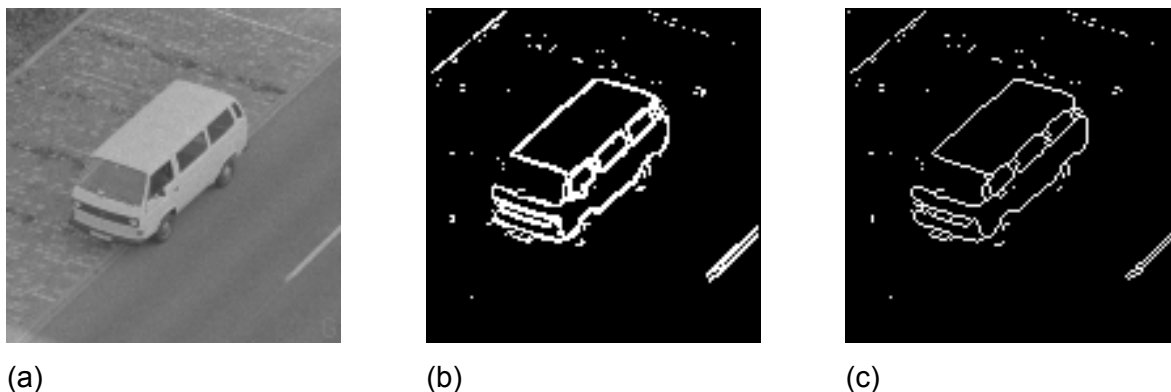


Abb. 4.4: Beispiel für eine Verdünnung:

- (a) Grauwertbild,
- (b) Binarisiertes Ergebnis des Sobel-Kantendetektors,
- (c) Ergebnis einer N8-Verdünnung.

Die Konturen bestehen nach der Verdünnung aus einer Menge einzelner Pixel, die noch keinerlei Verkettung zu einem Linienzug aufweisen. Wenn man abschließend eine Elementverketzung (siehe Kapitel 4.3) und danach eine Polygonzug-Approximation (siehe Kapitel 4.6.2.1) durchführt, so lässt sich die Kontur der Region (vektoriell) mit Hilfe weniger Stützstellen repräsentieren.

Schlussbemerkung: Kleine (ursprünglich vorhandene) Lücken können vor Anwendung der Erosion bzw. Verdünnung durch eine oder mehrere **Dilatationen** geschlossen werden, bei der die Streifen mit einer oder mehreren zusätzlichen "Schichten" umgeben werden.

4.2.4 Relaxation

Als letzte Möglichkeit seien hier die sogenannten kontinuierlichen **Relaxationsverfahren** genannt, die ein allgemeines Bildverarbeitungswerkzeug mit sehr weit gestreutem Einsatz darstellen. Ein Relaxationsverfahren ist ein iterativer Prozess, bei dem lokale Gegebenheiten durch ihren Kontext solange verändert werden, bis sich ein konsistentes Ergebnis einstellt. Im vorliegenden Fall heißt dies, dass sich benachbarte Kanten-/Linienelemente gegenseitig verstärken, wenn ihre Richtungen kompatibel sind, d.h. zueinander passen. Andernfalls schwächen sich die Elemente gegenseitig ab.

Beispiel für die Relaxation zur Nachverarbeitung von Linienelementen:

Für jedes Kanten-/Linienelement seien acht verschiedene Richtungen r_i ($i=0,\dots,7$) möglich. Für jede dieser (Fortsetzungs-) Richtungen besteht als Ergebnis der Kanten-/Linien-Detektion ein gewisses **Vertrauen** v_i ($i=0,\dots,7$). Für diese Initialvertrauen, die i.d.R. aus dem Ergebniswert des angewendeten Operators abgeleitet werden, gilt: $\sum v_i = 1$.

1. Iterationsschritt: (dieser und die weiteren erfolgen parallel für alle Elemente). Zur Neuberechnung der Vertrauenswerte eines Elementes wird seine Umgebung untersucht (z.B. bis zu

einem Abstand von vier Bildpunkten). In jeder Richtung, die einen Vertrauenswert $v_i \neq 0$ besitzt, wird nach Nachbarn gesucht, die mit dem aktuellen Element (genauer: mit seiner aktuellen Richtung) kompatibel sind.

Kompatibilität der Richtung eines Nachbarn mit der aktuellen Richtung bedeutet: der Nachbar besitzt eine Richtung mit hohem Vertrauen, die mit der aktuellen Richtung wenigstens annähernd übereinstimmt. Die Anzahl der kompatiblen Nachbarn, ihre Entfernung zum aktuellen Punkt und die Höhe der Initialvertrauenswerte in den interessierenden Richtungen ergeben zusammen den Beitrag der Nachbarschaft zur Neuberechnung des Vertrauens v_i der Richtung r_i (Verstärkung oder Abschwächung). Sind die Vertrauenswerte aller acht Richtungen für alle Punkte eines Bildes neu berechnet, beginnt der zweite Iterationsschritt. Dieser und die weiteren Schritte entsprechen dem ersten Schritt mit dem Unterschied, dass die jeweils betrachtete Nachbarschaft bereits durch deren Nachbarschaft beeinflusst ist. Auf diese Weise wird in die Bewertung des Vertrauens in eine Fortsetzungsrichtung zunehmend mehr Information über die Umgebung einbezogen.

Ein geeignetes Abbruchkriterium beendet die Iteration für jene Punkte, bei denen keine entscheidenden Änderungen mehr auftreten.

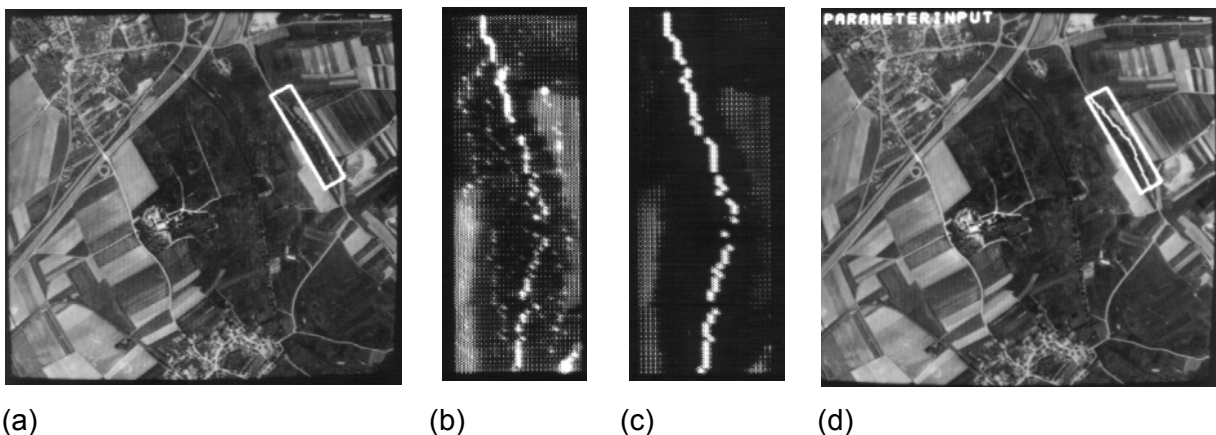


Abb. 4.4: Beispiel für die Anwendung eines Relaxationsverfahrens zur Linien-Extraktion:

- (a) Original-Luftbild mit schlecht erkennbarem Strassenstück;
- (b) Ergebnis der Linien-Detektion innerhalb des Rahmens aus (a);
- (c) Ergebnis der Anwendung eines Relaxationsverfahrens;
- (d) Einblendung des Relaxations-Ergebnisses in das ursprüngliche Luftbild.

Das Ergebnis der Relaxation ist noch keine Kontur bzw. Linie, sondern wieder eine Matrix einzelner Elemente mit Richtungsinformation. Im Gegensatz zur Ausgangssituation basieren die Hinweise auf mögliche Kanten-/Linien jetzt aber auf Informationen aus der weiteren Umgebung jedes Elementes. Außerdem können bei der gegenseitigen Verstärkung von Elementen auch Element-Verkettungen gebildet werden, die abschließend zur Gewinnung der zusammenhängenden Kontur bzw. Linie genutzt werden können.

Abbildung 4.4 zeigt ein Beispiel für die Anwendung eines Relaxationsverfahrens. In Abbildung 4.4 (a) ist ein Luftbild dargestellt, in dem im Bereich des weiß gezeichneten Rahmens der Straßenverlauf ermittelt werden soll. Abbildung 4.4 (b) zeigt den Rahmen in vergrößerter Darstellung, wobei zusätzlich als helle Punkte die Ergebnisse eines Linienelement-Detektors eingeblendet sind. In Abbildung 4.4 (c) ist das Ergebnis der Relaxation dargestellt, und Abbildung 4.4 (d) zeigt das ins Bild eingeblendete Endergebnis.

4.3 Kanten-/Linienelement-Verkettung

Das Ergebnis der Nachverarbeitung besteht in der Regel immer noch aus nicht zusammenhängenden Kanten-/Linienelementen. Im nächsten Verarbeitungsschritt müssen die einzelnen Elemente zu größeren Kanten-/Linienabschnitten zusammengefasst werden.

Die einfachste Möglichkeit besteht darin, zu jedem Element in seiner 3*3-Umgebung seinen Vorgänger und seinen Nachfolger zu suchen. Wegen der streng lokalen Sicht dieser

Methode sind die Ergebnisse jedoch häufig unbefriedigend (z.B. Lücken). Wenn an einer Lücke in Richtung der Objekt-Fortsetzung auch in mehreren Pixeln Entfernung nach einer Fortsetzung gesucht wird, so lassen sich auf diese Weise auch Lücken von mehreren Pixeln Breite schließen.

Andere Vorgehensweisen fassen die Element-Verkettung (Verfolgung) als Suchproblem auf. Durch Anwendung einer Bewertungsfunktion kann vermieden werden, dass der gesamte Suchbaum zur Suche nach dem optimalen Pfad aufgebaut werden muss. Die Definition der Bewertungsfunktion (Kostenfunktion) erlaubt die Berücksichtigung von Modellwissen über die gesuchten Kanten/Linien. Z.B. können gerade und kontrastreiche Kanten/Linien bevorzugt werden.

4.4 Hough-Transformation

Bei den bisher vorgestellten Ansätzen war für die Kanten-/Linienuche ein **implizites** Modell unterlegt, z.B. "glatter" Verlauf der Kante bzw. Linie. Die **Hough-Transformation**, die im weiteren am Beispiel der Ermittlung von Geradenstücken aus Kanten-/Linienelementen vorgestellt wird, gibt das zugrunde liegende Modell **explizit** vor, indem gezielt kollinear angeordnete Elemente, die im weiteren als Punkte bezeichnet werden, ausgewählt werden.

4.4.1 Hough-Transformation in den m-b-Parameterraum

Die Hough-Transformation löst das Kollinearitäts-Problem in einem **Parameterraum**. Dazu verwendet sie die Hauptform der Geraden im x-y-Raum: $y = m \cdot x + b$. Eine spezielle Gerade ist gegeben durch $y = m_0 \cdot x + b_0$, d.h. sie entspricht einem Punkt (m_0, b_0) im **m-b-Parameterraum**; siehe Abbildung 4.5.

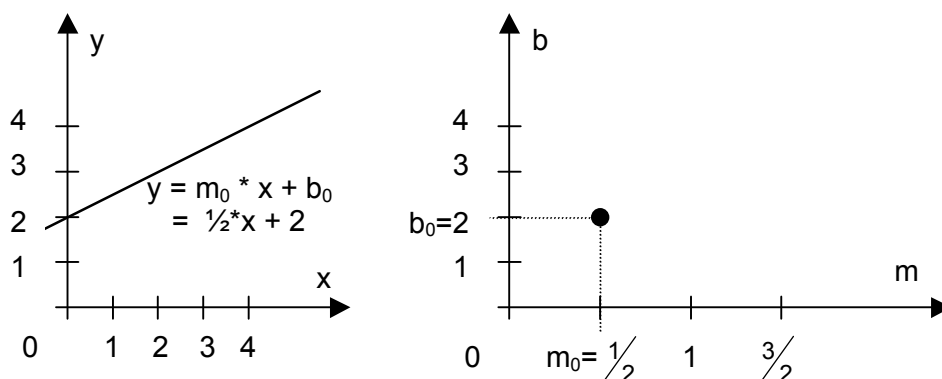


Abb. 4.5: Beispiel für die Hough-Transformation in den m-b-Raum: eine Gerade im x-y-Raum entspricht einem Punkt im m-b-Raum.

Als nächstes wird ein Punkt im x-y-Raum betrachtet. Der Punkt wird als Schnittpunkt beliebig vieler Geraden aufgefasst (Geradenbündel). Jede dieser Geraden entspricht einem Punkt im m-b-Raum, wobei alle diese Punkte auf einer Geraden liegen; siehe Abbildung 4.6.

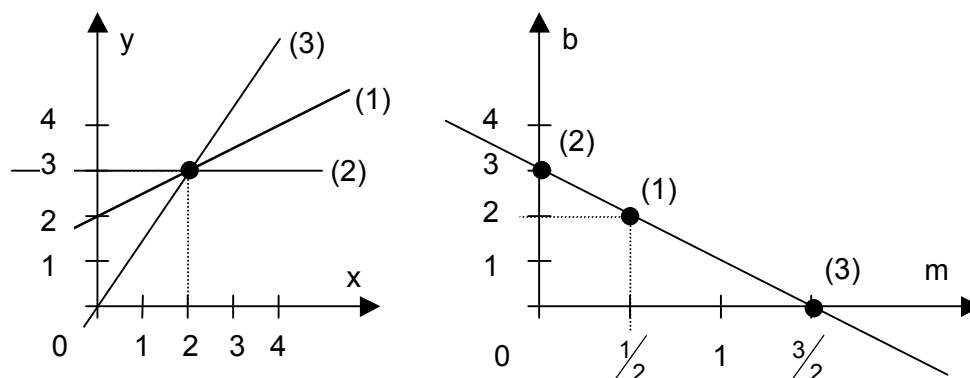


Abb. 4.6: Beispiel für die Hough-Transformation in den m-b-Raum: ein Punkt im x-y-Raum entspricht einer Geraden im m-b-Raum.

Zwei Punkte im x - y -Raum entsprechen zwei Geraden im m - b -Raum. Dabei korrespondiert die Gerade im x - y -Raum, auf der die beiden Punkte liegen, mit dem Schnittpunkt der beiden Geraden im m - b -Raum; siehe Abbildung 4.7.

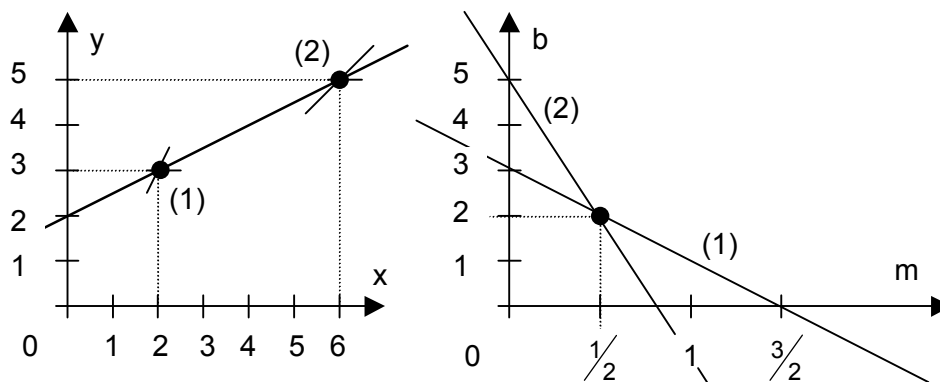


Abb. 4.7: Beispiel für die Hough-Transformation in den m - b -Raum: zwei Punkte im x - y -Raum entsprechen zwei Geraden im m - b -Raum, die sich in einem Punkt schneiden.

Im letzten Schritt werden im x - y -Raum n kollinear, d.h. auf einer Geraden angeordnete Punkte betrachtet: jeder dieser Punkte entspricht einer Geraden im m - b -Raum, und da die Punkte auf einer Geraden angeordnet sind, schneiden sich die zugehörigen Geraden in einem Punkt des m - b -Raumes (n -facher Schnittpunkt); siehe Abbildung 4.8.

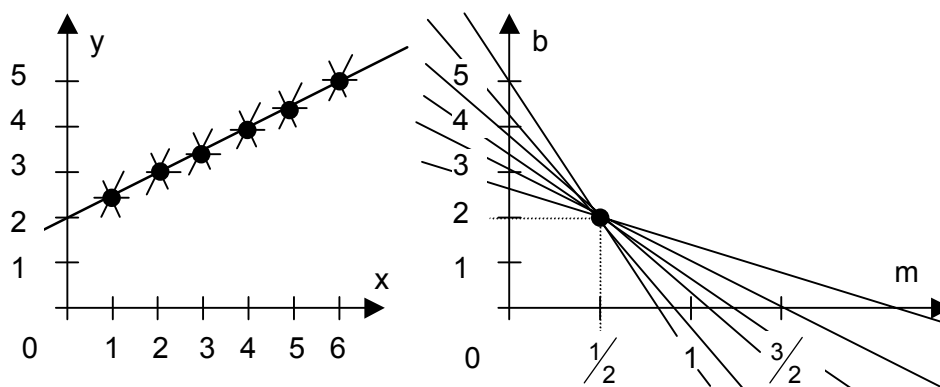


Abb. 4.8: Beispiel für die Hough-Transformation in den m - b -Raum: n kollinear angeordnete Punkte im x - y -Raum entsprechen n Geraden im m - b -Raum, die sich in einem Punkt schneiden.

Die obigen Überlegungen können nun direkt eingesetzt werden, um kollinear angeordnete Kanten-/Linienelemente zu finden: zuerst wendet man einen Kanten-/Liniendetektor auf das Bild an. Dann transformiert man die Positionen, an denen deutlich erkennbare Kanten- bzw. Linienelemente vorhanden sind - diese erhält man z.B. durch eine Binarisierung - in den m - b -Raum und sucht dort nach n -fachen Schnittpunkten ($n \gg 10$) der entsprechenden Geraden. Wenn man einen n -fachen Schnittpunkt im m - b -Raum gefunden hat, so weiß man, dass die zugehörigen n Kanten-/Linienelemente kollinear angeordnet sind.

Abbildung 4.9 zeigt ein synthetisches Beispiel zur Hough-Transformation in den m - b -Raum, an dem zusätzlich ein gravierender Nachteil der bisherigen Form erkennbar wird. In Abbildung 4.9 (a) ist das Originalbild dargestellt und Abbildung 4.9 (b) zeigt das (angenommene) Ergebnis einer Kantendetektion mit drei Mengen von kollinear angeordneten Kanten-Elementen. Im m - b -Raum (siehe Abbildung 4.9 (c)) sind zwei n -fache Schnittpunkte zu erkennen, die zu der waagrechten und der schräg verlaufenden Dreiecksseite gehören. Die linke, senkrecht verlaufende Dreiecksseite kann nicht ermittelt werden, da ihre Steigung unendlich groß ist, und der zugehörige Schnittpunkt deshalb im m - b -Raum im Unendlichen liegt.

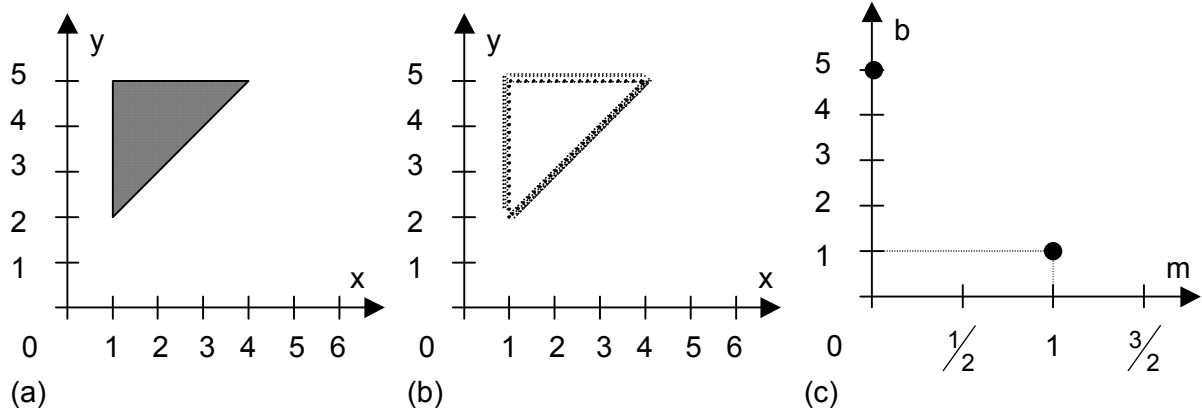


Abb. 4.9: Synthetisches Beispiel für die Hough-Transformation in den m-b-Raum:
 (a) Originalbild; (b) Ergebnis der Kantendetektion;
 (c) m-b-Raum mit zwei n-fachen Schnittpunkten.

4.4.2 Hough-Transformation in den ϕ -d- Parameterraum

Um zu vermeiden, dass Punkte, die kollinear auf einer senkrecht verlaufenden Gerade angeordnet sind, nicht erkannt werden können, wählt man statt des m-b-Raumes einen anderen Parameterraum.

Statt der Hauptform der Geraden verwendet man die Hesse'sche Normalform: $x \cdot \cos \phi + y \cdot \sin \phi - d = 0$ mit $\phi \in [0, \pi]$.

Eine spezielle Gerade $x \cdot \cos \phi_0 + y \cdot \sin \phi_0 - d_0 = 0$ des x-y-Raumes entspricht dann einem Punkt (ϕ_0, d_0) im ϕ -d-Parameterraum, wobei ϕ_0 und d_0 endlich sind; siehe Abbildung 4.10.

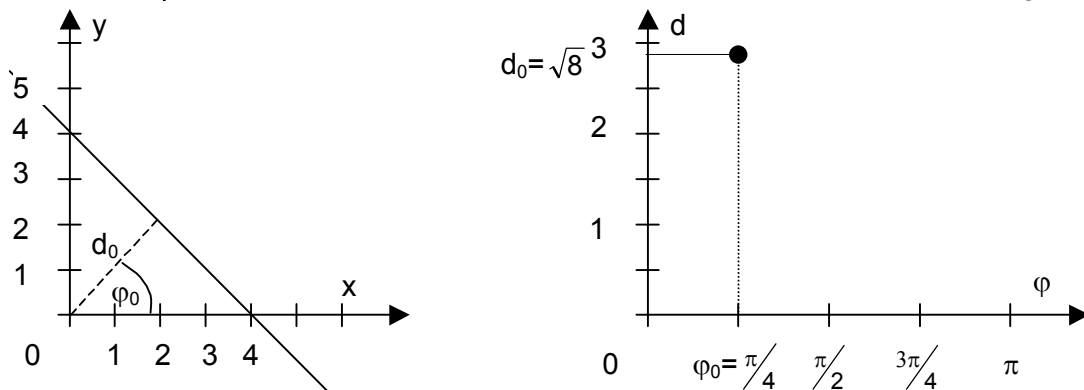


Abb. 4.10: Beispiel für die Hough-Transformation in den ϕ -d-Raum: eine Gerade im x-y-Raum entspricht einem Punkt im ϕ -d-Raum.

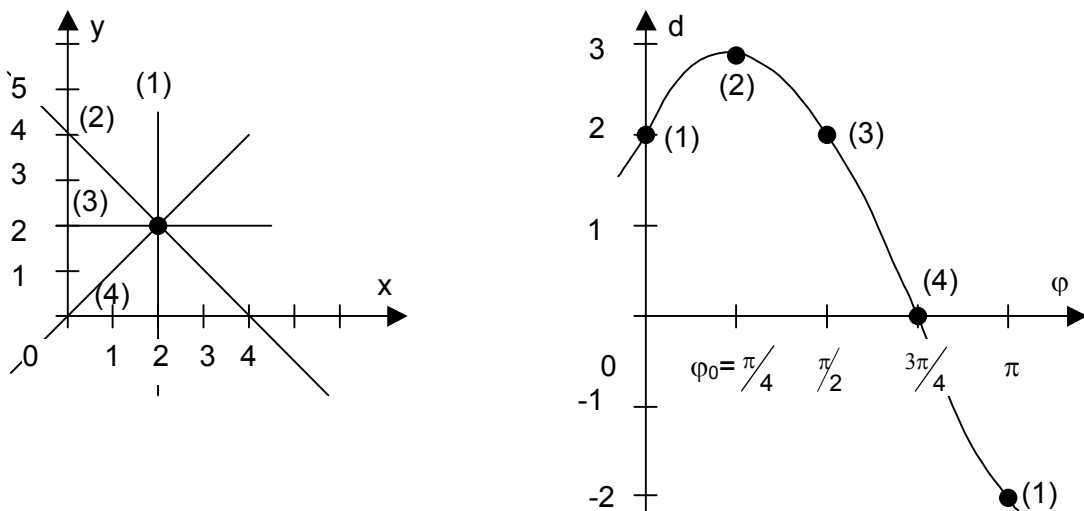


Abb. 4.11: Beispiel für die Hough-Transformation in den ϕ -d-Raum: ein Punkt im x-y-Raum entspricht einem Abschnitt einer sinusförmigen Schwingung im ϕ -d-Raum.

Ein Punkt (Geradenbüschel!) im x - y -Raum entspricht nun einem Abschnitt einer sinusförmigen Schwingung im φ - d -Raum; siehe Abbildung 4.11. Und n kollinear angeordnete Punkte des x - y -Raumes entsprechen nun n sinusförmigen Schwingungen im φ - d -Raum, die sich in einem Punkt (n -facher Schnittpunkt) schneiden.

Abbildung 4.12 zeigt ein Beispiel für zwei (kleine) kollinear angeordnete Punktmengen im x - y -Raum. Die zugehörigen Kurven schneiden sich im φ - d -Raum in zwei Punkten. Der Punkt B, der zu beiden kollinearen Anordnungen gehört, entspricht dabei einer Schwingung, die durch beide 3-fachen Schnittpunkte verläuft.

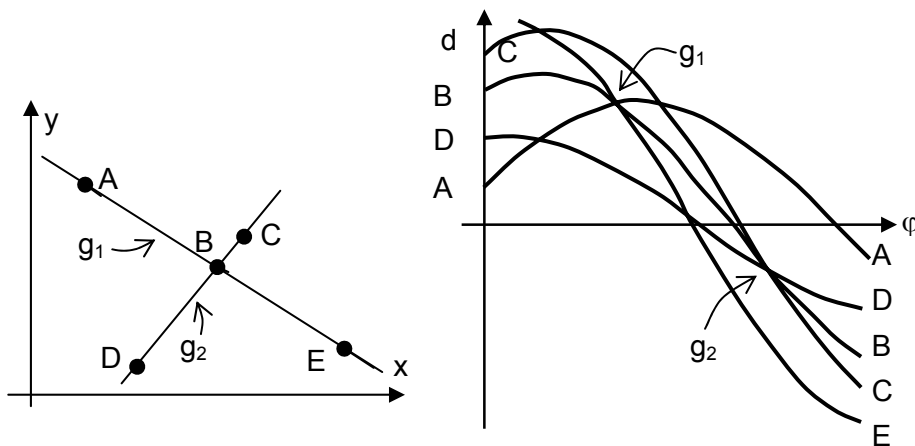


Abb. 4.12: Beispiel für die Hough-Transformation in den φ - d -Raum.

In Abbildung 4.13 ist ein reales Beispiel für die Hough-Transformation in den φ - d -Raum dargestellt. Auf das Originalbild (4.13 (a)) werden eine Kantendetektion und eine nachfolgende Verdünnung angewendet (siehe Abbildung 4.13 (b)). Im φ - d -Raum sind drei (unmittelbar übereinanderliegende) Schnittpunkte der sinusförmigen Schwingungen besonders deutlich zu erkennen (siehe Pfeile in Abbildung 4.13 (c)); diese Schnittpunkte entsprechen den drei Längsseiten des Quaders.

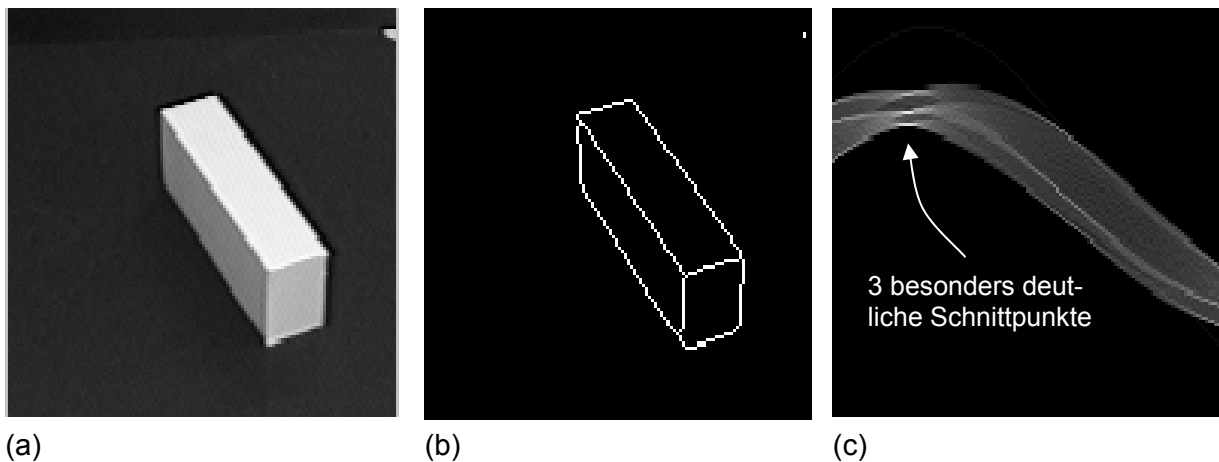


Abb. 4.13: Reales Beispiel für die Hough-Transformation in den φ - d -Raum: (a) Originalbild; (b) Kantendetektion und Verdünnung; (c) φ - d -Raum.

4.4.3 φ - d -Akkumulatorfeld

Der Wechsel des Parameterraumes erlaubt nun auch die Erkennung von senkrecht verlaufenden Kollinearitäten, jedoch treten noch zwei weitere Probleme auf:

- 1.) Um die Kollinearität von n Punkten im x - y -Raum zu überprüfen, müssen die Schnittpunkte von n Kurven errechnet werden, d.h. es müssen insgesamt $\frac{1}{2} * n * (n - 1)$ Schnittpunkts-Berechnungen durchgeführt werden, was einen enormen Rechenaufwand darstellt.

2.) Die Punkte des x-y-Raumes sind in der Bildverarbeitung ganzzahlige Pixel-Positionen, und außerdem liefern die Kanten-/Linien-detektoren breite Streifen von Kanten-/Linien-Elementen, so dass die Kollinearität in der Praxis nur näherungsweise erfüllt ist. Dies führt dazu, dass sich im φ -d-Raum kein exakter Schnittpunkt der Schwingungen ergibt.

Um diesen beiden Probleme zu begegnen, wird der φ -d-Raum gerastert und man erhält ein sogenanntes **φ -d-Akkumulatorfeld**; siehe Abbildung 4.14. Bei der Übertragung eines Punktes vom x-y-Raum in den φ -d-Raum ermittelt man, welche Zellen des Akkumulatorfeldes von der sinusförmigen Kurve, die dem Punkt entspricht, "getroffen" werden und inkrementiert den Zähler dieser Zellen um Eins.

Wenn alle Punkte des x-y-Raumes übertragen sind, durchsucht man das φ -d-Akkumulatorfeld nach Häufungspunkten, d.h. nach Zellen, die besonders hohe Einträge beinhalten.

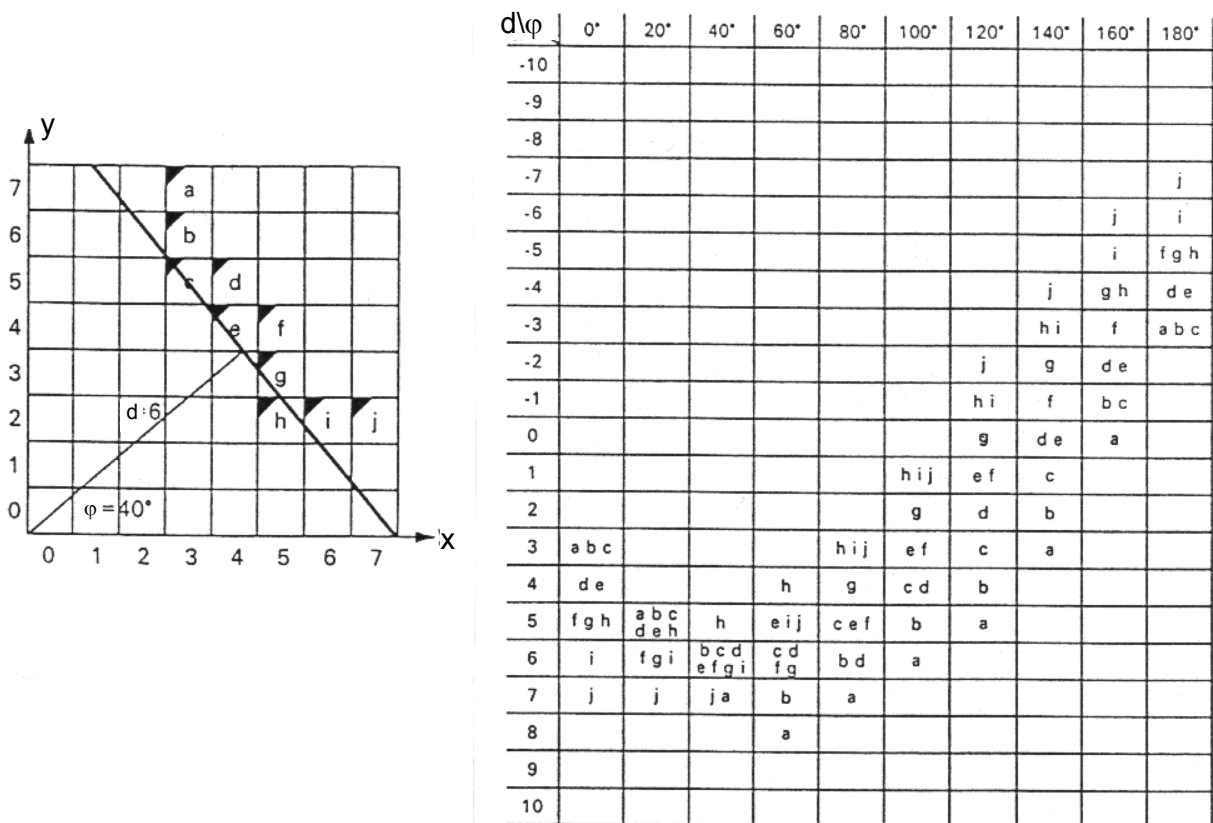


Abb. 4.14: Beispiel für ein Akkumulatorfeld bei der Hough-Transformation (Quelle: Bässmann, Besslich: "Konturorientierte Verfahren in der DBV", Springer-Verlag).

Durch dieses Vorgehen können auch näherungsweise Kollinearitäten erkannt werden und der Aufwand zur Ermittlung der Kurvenschnittpunkte steigt bei der Hinzunahme weiterer Punkte nur linear an.

4.4.4 Modifikation der Hough-Transformation nach Gorman/Clowes

Bei dem bisherigen Vorgehen wurde für die Kanten-/Linien-elemente, die zum Zwecke der Kanten- bzw. Linienfindung auf Kollinearität untersucht werden sollen, nur die Position der Elemente, nicht aber ihre Richtung r ausgewertet. Falls die Richtung r eines Elementes bekannt ist (z.B. Kirsch-Operator), so ergibt sich φ als $\varphi = (r+90 \text{ Grad})_{\text{modulo } 180 \text{ Grad}}$ und die Hough-Transformation vereinfacht sich erheblich. Dem Element mit Richtung r entspricht dann im φ -d-Raum nur noch ein Punkt, da φ festgelegt ist. Dieser Punkt liegt in genau **einer** Akkumulatorzelle, und nur diese wird für das betreffende Element um eins hochgezählt.

Schlussbemerkungen: Ein Nachteil der Hough-Transformation ist, dass sie keine Aussage darüber zulässt, ob kollinear angeordnete Punkte dicht beieinander oder weit voneinander

entfernt sind und z.B. auf zwei verschiedenen, kollinear "hintereinander" angeordneten Strecken liegen.

Abschließend sei nur erwähnt, dass sich die Hough-Transformation auf beliebige parametrische Kurven (z.B. Kreise, Ellipsen, ...) erweitern lässt.

4.5 Sequentielle Kanten-/Linienextraktion

Die bislang vorgestellten Verfahren haben den Vorteil der Parallelisierbarkeit. Dem steht der Nachteil gegenüber, dass im ganzen Bild gesucht wird, also auch dort, wo gar keine Kanten/Linien vorhanden sind. (Insbesondere im Bereich der Volumenbildverarbeitung stellt dies ein gravierendes Problem dar). Diesen enormen Aufwand vermeiden die **sequentiellen Extraktionsverfahren**, die sich zuerst deutlich erkennbare Kanten-/Linienelemente als sogenannte **Ansatzstellen** suchen. Von diesen ausgehend folgen die Verfahren den Kanten/Linien schrittweise. Da in jedem Schritt die Lage und das Erscheinungsbild des nächsten Elementes aufgrund der aktuellen Kenntnisse vorhergesagt werden können, sind die Verfahren adaptiv und führen auch bei gestörten Bildern zu vollständigeren und zuverlässigeren Ergebnissen.

Abbildung 4.15 zeigt ein Beispiel für die sequentielle Extraktion eines Straßennetzes aus einem Luftbild. In Abbildung 4.15 (a) sind die ermittelten Ansatzstellen dargestellt. Abbildung 4.15 (b) zeigt ein Zwischenergebnis bei der Verfolgung der Straße. Oben in der Bildmitte ist das bisher extrahierte Straßenstück zu sehen. Um die Fortsetzung der Straße zu finden, wird der Grauwertverlauf entlang einer gebogenen Testlinie untersucht, die "jenseits" des zuletzt gefundenen Straßenstückes so platziert wird, dass sie den zu erwartenden Straßenverlauf rechtwinklig schneidet. Das Grauwertprofil entlang der Teststrecke ist links in Abbildung 4.15 (b) eingezeichnet. Der Ausschlag nach oben entspricht dem Schnitt der Testlinie mit der Straße. Abbildung 4.15 (c) zeigt das Endergebnis der Linienextraktion.



(a)

(b)

(c)

Abb. 4.15: Beispiel für ein sequentielles Linienextraktionsverfahren.

4.6 Konturorientierte Repräsentationen

Bei der konturorientierten Segmentierung werden die segmentierten Bereiche des Bildes, d.h. die Regionen, anhand ihrer Kontur beschrieben. Diese besteht aus einer ein Pixel breiten, lückenlosen und geschlossenen Folge von aufeinanderfolgenden Konturpixeln.

Bei jeder Kontur-Repräsentation wird kurz darauf eingegangen, ob sich aus dieser Darstellung bestimmte Eigenschaften (z.B. „kreisförmig“ oder „(in Teilen) geradlinig“) einfach erkennen lassen. Weiterhin wird untersucht, ob die Repräsentation translations, rotations- oder skalierungs-invariant ist. (Damit kann z.B. einfach untersucht werden, ob ein Objekt in identischer oder ähnlicher Form mehrfach in einem Bild auftritt oder aber ein Objekt kann in den aufeinanderfolgenden Bildern einer Bildfolge einfach wiedergefunden werden).

Im Folgenden werden verschiedene Möglichkeiten zur Repräsentation der Konturen vorgestellt. Dabei wird zwischen vollständigen und unvollständigen Repräsentationen unterschieden.

4.6.1 Vollständige Kontur-Repräsentationen

Bei den vollständigen Kontur-Repräsentationen kann die Kontur (und damit auch die Region) pixelgenau rekonstruiert werden.

4.6.1.1 Folge absoluter Koordinaten

Bei dieser einfachsten Möglichkeit zur vollständigen Repräsentation einer Kontur werden die Konturpunkte ausgehend von einem Startpunkt (z.B. oberster, linkerster Punkt) der Reihe nach mit ihren **absoluten Koordinaten** (x, y) aufgelistet.

Aus dieser Form der Repräsentation sind Merkmale wie „kreisförmig“ oder „geradlinig“ nicht direkt erkennbar und die gesamte Kontur-Beschreibung ändert sich, wenn die Kontur verschoben, gedreht oder in ihrer Größe verändert wird.

4.6.1.2 Folge relativer Koordinaten

Die Erfassung der Kontur mit Hilfe **relativer Koordinaten** vermeidet die Abspeicherung "großer" Zahlenwerte, wie sie bei großen Bildern durch hohe Koordinatenwerte verursacht werden. Statt sich die absoluten Koordinaten des Folgepunktes (x_{n+1}, y_{n+1}) eines Punktes (x_n, y_n) zu merken, erfasst man nur die Inkremente dx und dy so dass gilt:

$$x_{n+1} = x_n + dx \quad \text{und} \quad y_{n+1} = y_n + dy \quad \text{mit } dx, dy \in \{-1, 0, 1\}.$$

Bei Konturen, die mit relativen Koordinaten beschrieben sind, können geradlinige Anteile recht einfach daran erkannt werden, dass bestimmte Strings von dx - dy -Inkrementen mehrmals hintereinander auftreten. Auch parallel verlaufende Konturteile können einfach entdeckt werden. Bei einer Translation der Region ändert sich nur der Startpunkt der Kontur. Bei einer Rotation oder einer Skalierung der Region ändern sich alle relativen Koordinaten.

4.6.1.3 Kettencodierung

Die **Kettencodierung (chain code)** stellt eine Weiterentwicklung der relativen Koordinaten dar.

Bei einer N4-Kontur kommen für die Nachfolge eines Konturelementes nur die vier N4-Nachbarn und bei einer N8-Kontur nur die acht N8-Nachbarn in Frage. Indem man die Elemente der N8-Nachbarschaft z.B. nach dem rechts gezeigten Schema durchnummeriert, kann ein Folgepixel durch die Angabe eines Kettencode-Elementes $\in \{0, \dots, 3\}$ bzw. $\in \{0, \dots, 7\}$ eindeutig adressiert werden.

5	1	4
2	P	0
6	3	7

Für die Kettencodierung gelten grundsätzlich dieselben Aussagen, wie für die relativen Koordinaten in Kapitel 4.6.1.2.

4.6.2 Unvollständige Kontur-Repräsentationen

Bei den unvollständigen Kontur-Repräsentationen wird die Kontur durch eine einfachere Kurve (z.B. einen Polygonzug) approximiert, so dass i.d.R. keine pixelgenaue Rekonstruktion der ursprünglichen Folge von Konturpixeln mehr möglich ist.

4.6.2.1 Polygonzug-Approximation

Bei der **Polygonzug-Approximation** wird die Kontur mit Hilfe eines Abstands-Schwellwertes durch eine Folge von Strecken approximiert. Dieser Schwellwert S gibt an, wie weit die tatsächlichen Konturpixel maximal vom approximierenden Streckenzug entfernt sein dürfen.

Der Ergebnis-Polygonzug wird aus der Eingabe-Kontur, die als Folge von aufeinander folgenden Pixeln vorliegt, iterativ berechnet:

- 1.) zuerst ermittelt man zwei diametral gelegene Extrempunkte der Kontur; z.B. das linkeste Pixel P_0 und das rechteste Pixel P_1 der Kontur; siehe Abbildung 4.16 (a). Die Punkte P_0 und P_1 zerlegen die Kontur in zwei Abschnitte (einen oberen und einen unteren), die im Folgenden getrennt bearbeitet werden.

Dazu wird die Verbindungsstrecke von P_0 nach P_1 als Approximation für die obere Konturhälfte und die Verbindungsstrecke von P_1 nach P_0 als Approximation für die untere Konturhälfte angenommen; siehe Abbildung 4.16 (a).

- 2.) Für jede Strecke mit Anfangspunkt P_A und Endpunkt P_E wird nun aus der Menge der zugehörigen Konturpixel derjenige Punkt P_M ermittelt, der den größten Abstand zu der approximierenden Strecke hat; siehe Abbildung 4.16 (b).

Ist dieser Abstand kleiner als der Abstands-Schwellwert S , so stellt die Strecke von P_A nach P_E die endgültige Approximation des zugehörigen Konturstückes dar. Andernfalls wird die aktuell untersuchte Strecke in zwei Strecken zerlegt, die von P_A nach P_M und von P_M nach P_E verlaufen.

Diese beiden Strecken werden wieder einzeln untersucht, d.h. das Vorgehen wird mit dem Beginn von 2.) fortgesetzt.

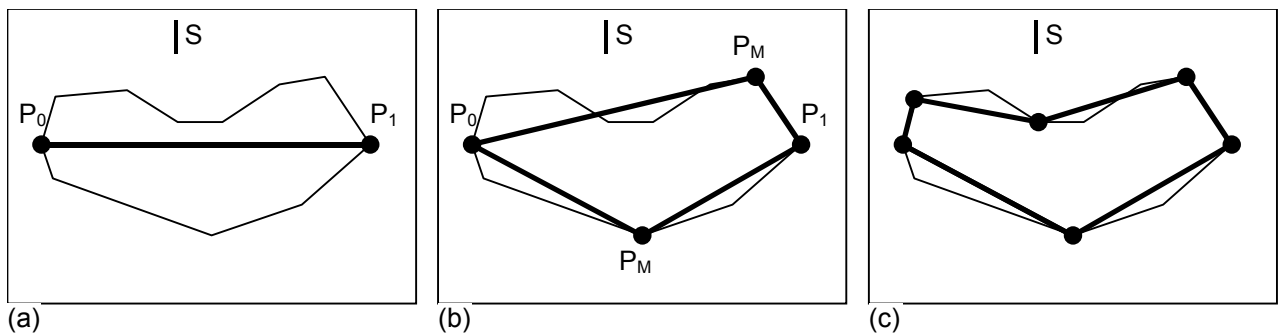


Abb. 4.16: Beispiel für die Polygonzug-Approximation einer Kontur:
 (a) Originalkontur mit diametral angeordneten Punkten P_0 und P_1 , einem initialen Polygonzug (P_0 nach P_1 und zurück) und dem Abstands-Schwellwert S ,
 (b) die beiden Strecken des initialen Polygonzuges werden aufgespalten,
 (c) endgültige Polygonzugapproximation (abhängig vom Abstands-Schwellwert S).

Das Gesamtverfahren endet, wenn kein Konturpixel mehr weiter als der Abstands-Schwellwert S von einer Strecke des approximierenden Polygonzuges entfernt ist; siehe Abbildung 4.16 (c).

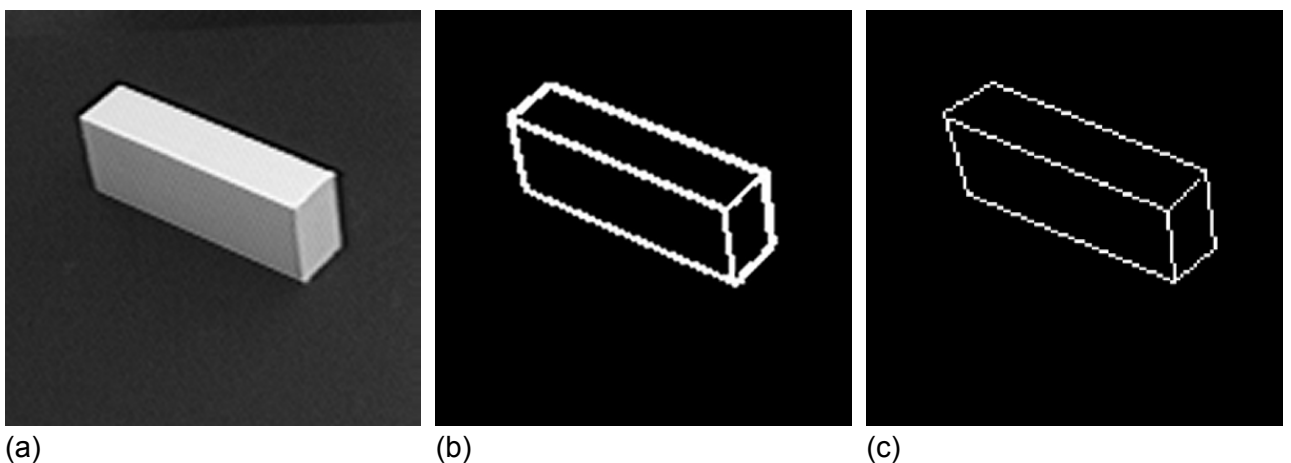


Abb. 4.17: Beispiel für die Polygonzug-Approximation einer Kontur:
 (a) Originalbild,
 (b) binarisiertes Ergebnis einer Kantendetektion; (Ergebnis ist rasterorientiert),
 (c) Polygonzug-Approximation des Ergebnisses der Verdünnung von Bild (b) und nachfolgender Linienverfolgung; (Ergebnis ist vektororientiert).

Abbildung 4.17 zeigt ein Beispiel für die unvollständige konturorientierte Repräsentation des Ergebnisses einer konturorientierten Segmentierung. In Abbildung 4.17 (a) ist das Originalbild dargestellt. In Abbildung 4.17 (b) wurde ein Kantendetektor angewendet und das Kan-

tenbild wurde binarisiert. Abbildung 4.17 (c) zeigt das Endergebnis nach einer Verdünnung, einer Elementverkettung zur Ermittlung der Folge von Konturpixeln und einer abschließenden Polygonzug-Approximation. Die Kontur wird durch wenige Eckpunkte und die dazwischen aufgespannten Strecken repräsentiert.

Die Polygonzug-Approximation eignet sich (in Abhängigkeit vom verwendeten Schwellwert) bedingt, um geradlinige Konturstücke zu erkennen. Ansonsten ist diese Repräsentationsform weder translations-, rotations- noch skalierungs-invariant.

4.6.2.2 Polar-Koordinaten-Repräsentation

Bei der Polar-Koordinaten-Repräsentation wird zuerst der Schwerpunkt S der Region bestimmt. Ausgehend von diesem Schwerpunkt ermittelt man nun z.B., waagrecht nach rechts die Entfernung bis zur Kontur, und trägt diesen Wert über dem Winkel 0° als Radius ins Polar-Koordinaten-System ein. Danach wird diese Entfernungsmessung schrittweise wiederholt, wobei bei jedem Schritt der Winkel, in dem bis zur Kontur gemessen wird, um eine bestimmte Gradzahl inkrementiert wird.

In Abbildung 4.18 (b) ist die Polar-Koordinaten-Repräsentation der in Abbildung 4.18 (a) gezeigten rechteckigen Kontur dargestellt.

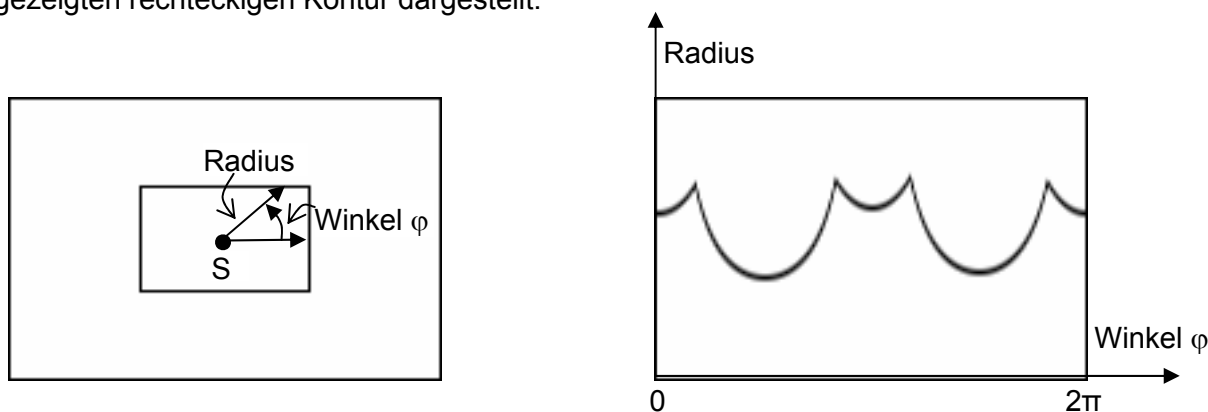


Abb. 4.18: Beispiel für die Polar-Koordinaten-Repräsentation einer Kontur:
 (a) Originalbild,
 (b) Polar-Koordinaten-Darstellung

Diese Form der Kontur-Repräsentation eignet sich nur für sternförmig konvexe Regionen, d.h. Regionen, bei denen jeder Strahl, der vom Schwerpunkt der Region ausgeht, die Kontur nur in einem Punkt schneidet. Die Polar-Koordinaten-Repräsentation bietet sich besonders zur Ermittlung von kreisförmigen Konturen oder Konturteilen an. Bei einer Translation ändert sich nur der Schwerpunkt der Region, bei einer Rotation wird die Kurve im Polar-Koordinaten-System in waagrechter Richtung und bei einer Skalierung senkrechter Richtung verschoben.

5 Interpretation von Strichzeichnungen

Dieses und die folgenden drei Kapitel beschäftigen sich damit, dreidimensionale Strukturen aus zweidimensionalen Grauwertbildern abzuleiten. Bevor auf die Interpretation von Strichzeichnungen eingegangen wird, sollen aber noch einige allgemeine Aspekte zur 3D-Problematik erörtert werden.

5.1 Allgemeine Vorbemerkungen zur 3D-Problematik

Das größte Problem bei der Ableitung von 3D-Information aus Bildern besteht darin, dass die bilderzeugende Projektion, die einen 3D-Weltausschnitt in ein 2D-Bild abbildet, nicht umkehrbar ist: ein 2D-Bild kann die Projektion von unendlich vielen verschiedenen 3D-Szenen sein (siehe Abbildung 5.1).

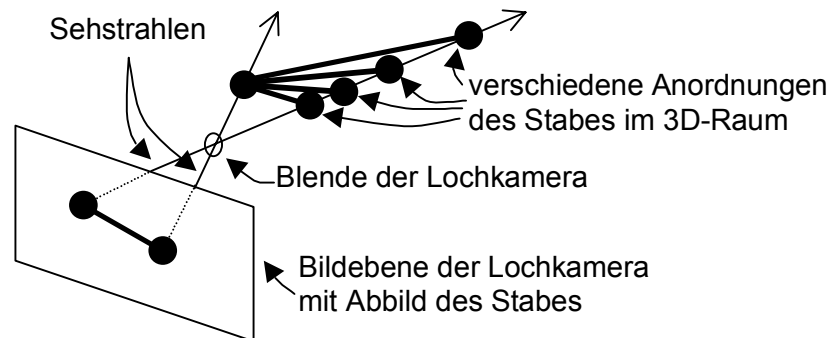


Abb. 5.1: Beispiel für die Nicht-Eindeutigkeit der Projektion von 3D nach 2D.

In der Praxis (d.h. bei realen Daten) ist der Mensch jedoch meist problemlos in der Lage, den Bildinhalt räumlich zu interpretieren, indem er 3D-Szenenmerkmale aus 2D-Bildmerkmalen ableitet. Solche sogenannten **inneren Charakteristika (intrinsic characteristics)** sind z.B. der Abstand von Oberflächen zum Betrachter, ihre Neigung und ihr Reflektionsverhalten.

Für die automatische Bildauswertung werden in der Regel mehrere "verschiedene" Eingabebilder derselben Szene gefordert:

- Stereobilder (verschiedene Kamerastandpunkte); siehe Kapitel 6,
- Bildfolgen (zeitlich gestaffelte Bilder bei bewegter Szene und/oder bewegter Kamera); siehe Kapitel 7,
- photometrische Stereobilder (verschiedene Lichtquellen bei gleichem Kamerastandpunkt); siehe Kapitel 8.1.

Wenn nur ein Bild zur Verfügung steht, so müssen für den Versuch der Erkennung von 3D-Strukturen zusätzliche Kenntnisse vorliegen (**Modellwissen**) z.B.:

- Beleuchtungsmodell: Beleuchtung, Reflektivität,
- Sensormodell: z.B. Perspektiv-Abbildung,
- Weltmodell: z.B. "Klötzchenwelt".

Der Einfluss von Modellen auf die Bildinterpretation wird in Kapitel 9 behandelt.

Bevor in den Kapiteln 6 bis 8 Abbildungen von Szenen der realen Umwelt ausgewertet werden, sollen in diesem Kapitel künstliche Szenen aus der sogenannten **Klötzchenwelt** behandelt werden. Für diese Szenen mit ihren einfachen Gegebenheiten (einfache Körper, eindeutige Abbildungen, optimierte Beleuchtung) kann noch am wahrscheinlichsten erwartet werden, dass flächen- oder konturorientierte Segmentierungsverfahren vollständige und richtige Ergebnisse liefern. Zwar trifft dies in der Praxis meist doch nicht hundertprozentig zu, aber die vorgestellten, weiterverarbeitenden Verfahren können so modifiziert werden, dass sie auch nicht ganz perfekte Segmentationen verarbeiten können. Zur Schilderung der grundlegenden Verfahren wird im weiteren jedoch vorausgesetzt, dass die Segmentierung z.B. mit Hilfe der in Kapitel 4 vorgestellten Verfahren perfekte Strichzeichnungen der Objektkonturen ohne Lücken oder fehlende Linien und ohne zusätzliche störende Linien liefert.

Aus der Vielzahl von Verfahren zur Analyse solcher perfekter Strichzeichnungen soll hier nur der von Huffman und Clowes unabhängig voneinander entwickelte Ansatz zur Markierung

einzelner Linien und Linienschnittpunkte vorgestellt werden (Kapitel 5.2). Darauf bauen dann zwei Verfahren auf, die aus diesen Linienschnittpunkt-Einzelmarkierungen konsistente Markierungen für eine ganze Strichzeichnung erzeugen, (Kapitel 5.3). Diese Gesamtmarkierung erlaubt die Trennung unterschiedlicher Objekte in der Szene sowie ihre Interpretation. Kapitel 5.4 geht kurz auf Erweiterungen und Verallgemeinerungen der Verfahren ein.

5.2 Markierung einzelner Linien und Linienschnittpunkte

Bei dem Verfahren von Huffman und Clowes werden nur sehr einfache Objekte zugelassen: massive Klötzchen, bei denen sich in jeder Klötzchenecke nur drei Flächen treffen (siehe Abbildung 5.2 (a)). Objekte dürfen sich teilweise verdecken, aber die Kamera muss sich in einem sogenannten **allgemeinen Kamerastandort** befinden, bei dem eine geringfügige Positionsänderung die "Art der Schnittpunkte" in den Strichzeichnungen nicht verändert (siehe Abbildung 5.2 (b)). Weiterhin sollen keine Schatten vorhanden sein und Objekte dürfen nicht aufeinander stehen, d.h. es sollen keine sogenannten Spalten (cracks) auftreten.

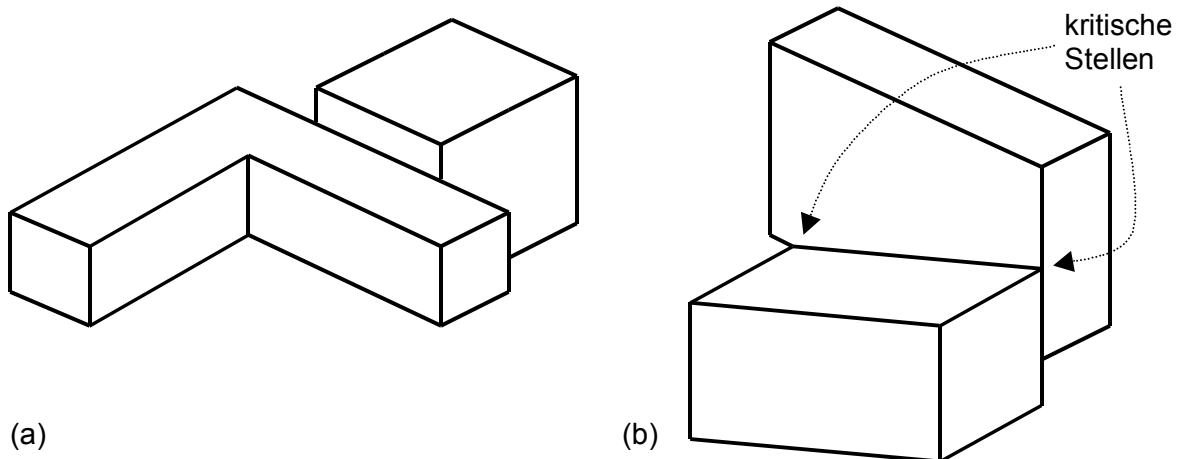


Abb. 5.2: (a) Beispiel aus der Klötzchenwelt für zulässige Objekte und eine mögliche Anordnung,
 (b) Beispiel für eine Abbildung, die gegen den allgemeinen Kamerastandort verstößt.

Eine Analyse solcher Strichzeichnungen zeigt, dass nur vier Arten von Linien auftreten können: **Grenzlinien** (zwei Richtungen), sowie **konvexe** und **konkave "innere" Linien**. Diese vier Arten von Linien sind in Abbildung 5.3 (a) mit einem " \rightarrow " bzw. " \leftarrow " (Fläche sitzt jeweils rechts), oder einem "+" oder "-" gekennzeichnet.

Weiterhin können nur folgende Arten von Linienschnittpunkten auftreten (siehe Abbildung 5.3 (b)): **L**, **Pfeil**, **Gabel** und **T**. (Ein T tritt dort auf, wo ein Objekt ein anderes teilweise verdeckt). Rein rechnerisch können für jede der an einem Schnittpunkt beteiligten Linien vier Linienmarkierungen (\rightarrow , \leftarrow , +, -) auftreten. Für den Pfeil würden sich also $4^3=64$ Markierungsmöglichkeiten ergeben. In der Realität treten aber nur 3 Markierungskombinationen für den Pfeil auf. Der vollständige Katalog der insgesamt 18 möglichen Linienschnittpunkt-Markierungen ist in Abbildung 5.4 gezeigt.

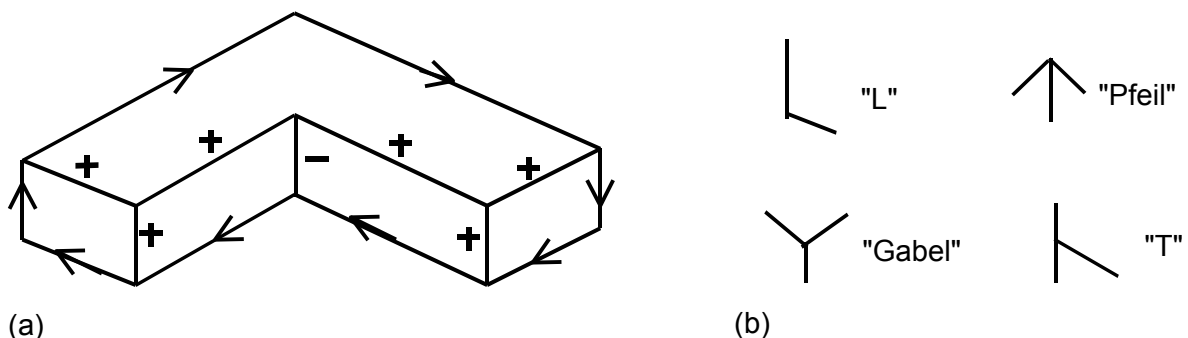


Abb. 5.3: (a) Mögliche Linienmarkierungen.
 (b) Mögliche Linienschnittpunkte.

5.3 Konsistente Markierung von Strichzeichnungen

Nachdem nun anhand vorgegebener, interpretierter Strichzeichnungen alle möglichen Linien und Linienschnittpunkte samt Markierungen katalogisiert sind, kann das Vorgehen umgekehrt werden: zu einer vorgelegten (perfekten) Strichzeichnung sollen eine oder mehrere konsistente Gesamtmarkierungen aller vorhandenen Linienschnittpunkte gefunden werden. Dies erlaubt dann die Trennung und Interpretation der Objekte in der Szene.

Bei der einfachsten Vorgehensweise werden zu jedem Linienschnittpunkt alle möglichen Markierungen vermerkt. Danach werden in der Gesamtzeichnung alle Kombinationsmöglichkeiten für alle Linienschnittpunkte-Markierungen nacheinander getestet, um konsistente Markierungen der Szene zu finden. Es ist unmittelbar einleuchtend, dass ein derartiges "Brute-Force"-Vorgehen wegen der immensen kombinatorischen Vielfalt nur bei Szenen mit sehr wenigen Linien eingesetzt werden kann.

Intelligentere Verfahren stützen sich daher auf eine (eigentlich triviale) sogenannte **Zwangsbedingung (constraint)**: Jede Linie muss an ihren beiden Enden dieselbe Markierung besitzen! Mit Hilfe dieser Zwangsbedingung lassen sich zwei unterschiedliche Markierungsverfahren aufstellen.

5.3.1 Sequentielles Markierungsverfahren

Zu Beginn dieses auf Waltz zurückgehenden Verfahrens wird ein beliebiger "außen" liegender Linienschnittpunkt X als Startposition ausgewählt. Zu diesem werden nach Katalog alle möglichen Schnittpunkte-Markierungen abgespeichert. Als nächstes betrachtet man einen beliebigen benachbarten Schnittpunkt Y, zu dem eine Verbindungslinie existiert und speichert zu diesem alle möglichen Schnittpunkte-Markierungen ab. Dann werden alle Markierungen von Y, die wegen der Zwangsbedingung nicht mit den Markierungen von X verträglich sind, gelöscht. Dieser Schritt wird als **Vorwärtsausbreitung der Zwangsbedingung (forward constraint propagation)** bezeichnet. Es folgt eine **Rückwärtsausbreitung der Zwangsbedingung (backward constraint propagation)**, bei der für den Schnittpunkt X alle Markierungen gelöscht werden, die nicht mit den Markierungen von Y verträglich sind.

Danach wird ein neuer benachbarter Schnittpunkt Z ausgewählt, und dort wird wie oben beschrieben vorgegangen. Die Vorwärts- und Rückwärtsausbreitung erstreckt sich dabei bis zu Punkt X. Anschließend wird ein weiterer benachbarter Linienschnittpunkt behandelt usw.. Das Verfahren, das der Abarbeitung eines Suchbaumes entspricht, endet, wenn alle Linienschnittpunkte untersucht wurden.

Ein Beispiel für das sequentielle Markierungsverfahren ist in Abbildung 5.5 (a) vorbereitet.

5.3.2 Markierung durch diskrete Relaxation

Neben dem sequentiellen Markierungsverfahren wird vor allem der im folgenden vorgestellte **Waltz'sche Filteralgorithmus** eingesetzt. Dabei handelt es sich um ein sogenanntes diskretes Relaxationsverfahren. **Relaxationsverfahren** (vgl. Kapitel 4.2.4) sind parallele, iterative Verfahren, bei denen die Markierungen von Elementen durch den Einfluss ihrer Umgebung solange beeinflusst werden, bis sich ein konsistentes Gesamtergebnis einstellt. Im vorliegenden Fall sind die Elemente die Linienschnittpunkte und die Markierungen sind die möglichen Schnittpunkte-Markierungen aus Abbildung 5.4. Da diese Markierungen nur vorhanden (Wahrscheinlichkeit Eins) oder nicht vorhanden (Wahrscheinlichkeit Null) sein können, wird das Verfahren als **diskretes** Relaxationsverfahren bezeichnet.

Zu Beginn des Relaxationsverfahrens werden für alle Linienschnittpunkte alle (lokal) möglichen Schnittpunkte-Markierungen aufgelistet. Bei jedem Schritt des iterativen Vorgehens werden nun parallel für alle Schnittpunkte alle die Markierungen gelöscht, die nicht mit den Markierungen benachbarter Schnittpunkte verträglich sind (constraint propagation). Das Verfahren konvergiert i.d.R. rasch, d.h. bereits nach wenigen Iterationsschritten können keine weiteren Markierungen mehr entfernt werden, so dass eine oder mehrere Interpretationen der vorgelegten Strichzeichnung ermittelt wurden.

Ein Beispiel für das Markierungsverfahren durch diskrete Relaxation ist in Abbildung 5.5 (b) vorbereitet.

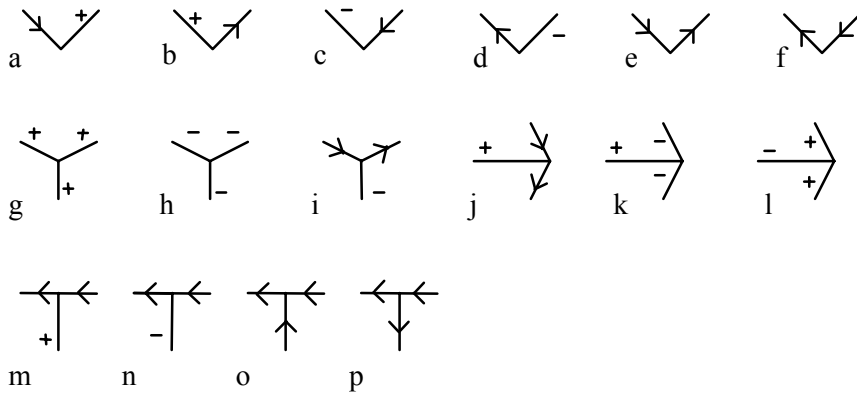


Abb. 5.4: Katalog aller möglichen Linienschnittpunkts-Markierungen.

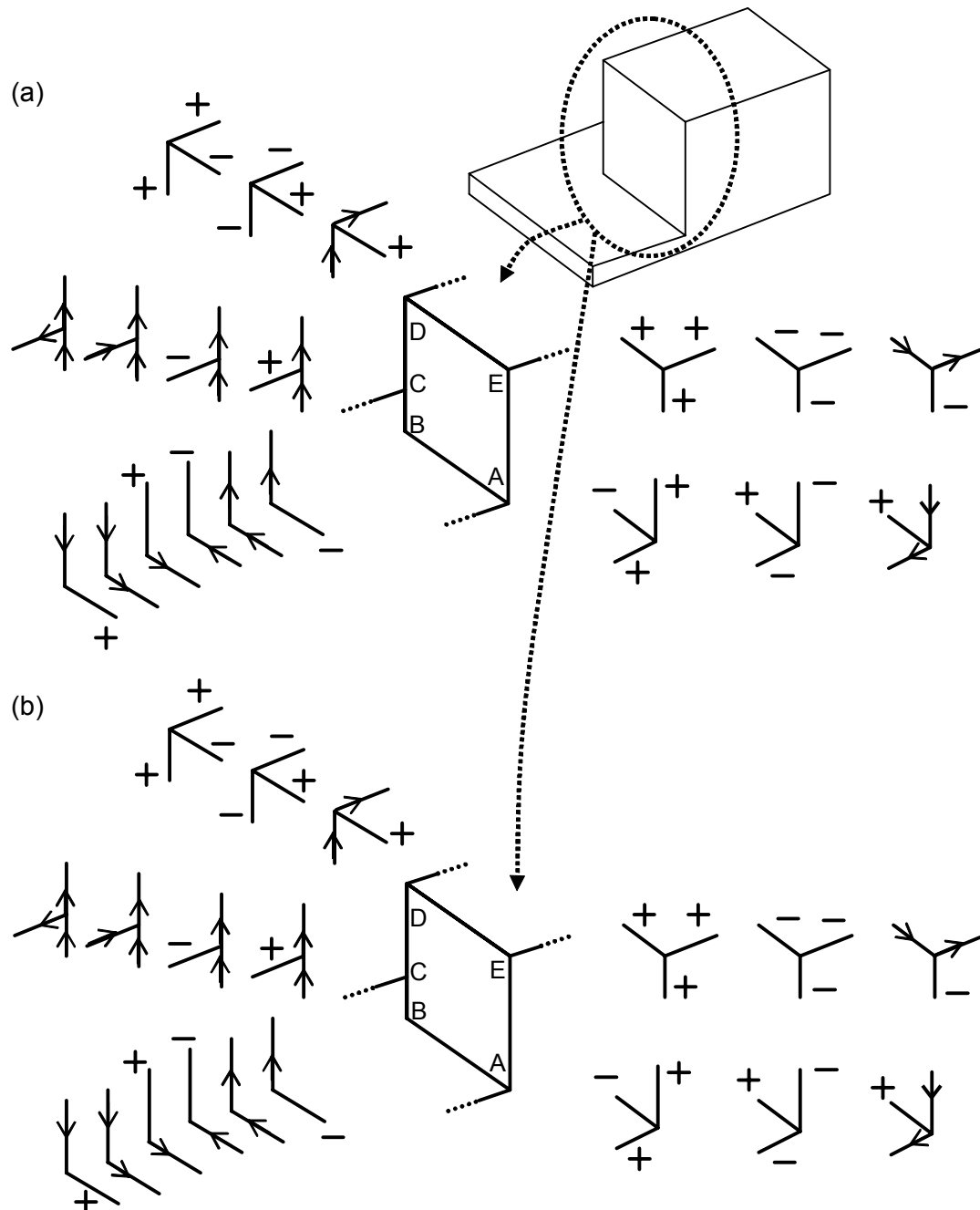


Abb. 5.5: Vorbereitetes Beispiel für die Markierung einer Strichzeichnung:
 (a) sequentielles Verfahren;
 (b) Verfahren mit diskreter Relaxation.

Die durch das sequentielle Verfahren bzw. das Relaxationsverfahren gefundenen konsistenten Markierungen der Strichzeichnungen müssen nicht eindeutig sein: z.B. kann ein isoliert gezeichneter Würfel frei im Raum schweben oder mit einer seiner drei verdeckten Flächen an einer "großen" Fläche wie z.B. einer Wand befestigt sein. Wenn für eine Strichzeichnung keine konsistente Markierung gefunden werden kann, so stellt die Linienzeichnung ein **unmögliches Objekt** dar (siehe Abbildung 5.6). Unmögliche Objekte sind i.d.R. lokal, (d.h. isoliert an jeder Ecke betrachtet) widerspruchsfrei, aber es existiert keine konsistente Gesamtinterpretation.

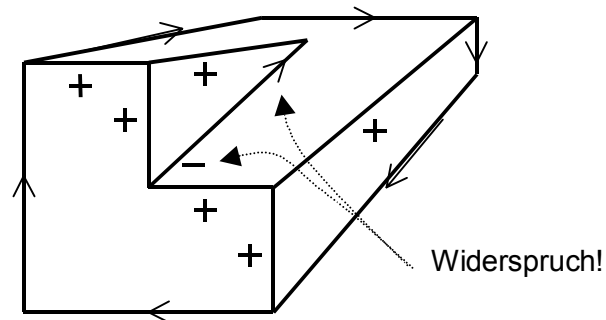


Abb. 5.6: Beispiel für ein unmögliches Objekt (die mittlere Kante ist vorne und hinten unterschiedlich markiert)

5.4 Erweiterungen der Verfahren

Die angegebenen Markierungsverfahren lassen sich auch für komplexere Szenen z.B. mit Schatten, Spalten und Hohlräumen einsetzen. Für diese Fälle müssen erweiterte Kataloge zur Linienmarkierung und zur Schnittpunkts-Markierung aufgestellt werden und es müssen neue Schnittpunkts-Typen berücksichtigt werden. Auf den ersten Blick steigt dadurch die Anzahl möglicher Markierungen für die Schnittpunkte dramatisch an: z.B. sind nun rein rechnerisch für einen Pfeil über 120.000 verschiedene Markierungen möglich, in der Realität können aber "nur" 70 Markierungen auftreten. Besonders der Waltz'sche Filteralgorithmus führt hier vergleichsweise rasch zum Ziel, da bei je zwei benachbarten Linienschnittpunkten immer nur sehr wenige Markierungen zueinander passen.

Dies bedeutet, dass die Hinzunahme weiterer Informationen (z.B. Spalten, Schatten usw.) zwar die Zahl der möglichen Einzelschnittpunkts-Interpretationen enorm ansteigen lässt, die Gesamtinterpretation der Zeichnung i.d.R. aber sehr schnell eindeutig wird.

Andere Verfahrenserweiterungen lassen andere Objekte zu, bei denen sich z.B. mehr als drei Flächen in einer Körperecke treffen können, oder aber es können sogenannte **Origami-Objekte** bearbeitet werden. (Origami ist eine japanische Kunst, bei der offene oder geschlossene 3D-Körper aus Papier gefaltet werden). Bei noch weitergehenden Verfahrenserweiterungen können auch Objekte mit bestimmten gekrümmten Oberflächen behandelt werden, und auch die Forderung nach einem allgemeinen Kamerastandpunkt kann aufgegeben werden.

Die schwierigste Erweiterung für die Verfahren stellt die Einbeziehung unvollständiger bzw. fehlerhafter Linienzeichnungen dar. Hier wurden mehrere Ansätze entwickelt, die z.B. Hypothesen aufstellen, wo bestimmte nicht extrahierte Linien erwartet werden können. Die Hypothesen werden anschließend im Originalbild überprüft. Diese Verfahren arbeiten aber nach wie vor nur mit mäßigem Erfolg und eine Übertragung der Verfahren auf reale Szenen ist bis heute nur sehr eingeschränkt gelungen.

6 Stereobildauswertung

Bei der **photogrammetrischen Stereobildauswertung** wird eine Szene von zwei (oder mehr) unterschiedlichen Kamerastandorten aus aufgenommen. Durch einen Vergleich der Bilder kann die dreidimensionale Struktur des aufgenommenen Raumes rekonstruiert werden, d.h. es können die Abstände zwischen den Kameras und den abgebildeten Objekten bestimmt werden.

Die Stereobildauswertung spielt z.B. im Bereich der Robotik, aber auch bei der Luftbildauswertung zur Erzeugung von digitalen Geländemodellen und Orthophotos (Photos mit Kartengrundriss, d.h. ohne perspektivische Verzerrungen) eine große Rolle.

In Kapitel 6.1 wird zunächst ein einfacher Ansatz vorgestellt, mit dem erkannt werden kann, ob sich vor den Stereokameras ein Hindernis befindet. In Kapitel 6.2 wird dann gezeigt, wie man mit Hilfe der binokularen Merkmale "Vergenz" und "Disparität" Abstände zu Objekten ermitteln kann. In Kapitel 6.3 wird auf das sogenannte Korrespondenzproblem eingegangen und in den Kapiteln 6.4 und 6.5 werden zwei Verfahren zur Auflösung von Mehrdeutigkeiten beim Korrespondenzproblem vorgestellt. Auf alternative Methoden, die zur Abstandsbestimmung beispielsweise Spaltlichtbeleuchtungen (auch: Streifenprojektion oder Lichtschnittverfahren) oder aktive Sensoren (z.B. Laser-Entfernungsmesser) einsetzen, wird hier nicht eingegangen.

6.1 Hindernis-Ermittlung mit inverser Projektion

Mit dem im Folgenden vorgestellten **Verfahren mit inverser Projektion** (s. z.B. [Mallot]) kann beispielsweise im Bereich der Robotik relativ einfach festgestellt werden, ob sich auf dem Boden vor einem Roboter ein Hindernis befindet (**Hindernis-Ermittlung**), so dass er sich z.B. nicht ungehindert vorwärts bewegen kann.

Für dieses Verfahren wird vorausgesetzt, dass die gegenseitige Anordnung der beiden Stereokameras (gegenseitiger Abstand der beiden Kameras und Ausrichtung der optischen Achsen), sowie ihre "Höhe über Grund" und der Neigungswinkel, unter dem die Kameras auf die Bodenfläche schauen, bekannt sind; siehe Abbildung 6.1. (Die **optische Achse** einer Kamera ist dabei die Gerade, die senkrecht auf der Bildebene steht und durch die Mitte der Kamerablende verläuft).

Unter diesen Voraussetzungen ist die Projektion der Punkte der Bodenfläche ins Bild umkehrbar (inverse Projektion), d.h. zu einem Bildpunkt im 2D-Bild lässt sich bezüglich des Kamera-Koordinatensystems der zugehörige 3D-Punkt auf der Bodenfläche berechnen.

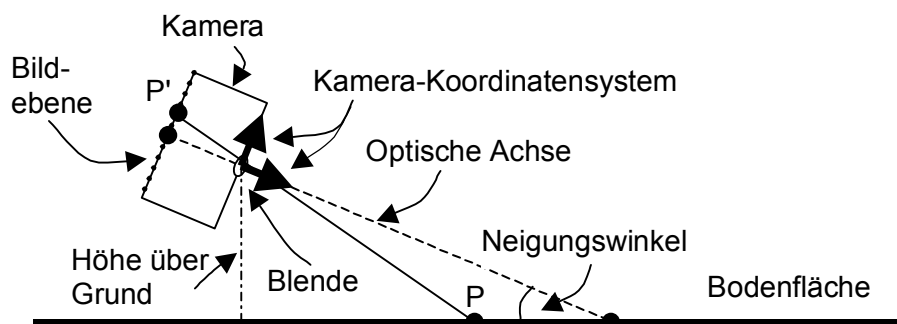


Abb. 6.1: Prinzipbild in Seitenansicht zur inversen Projektion: es gibt eine eindeutige Entsprechung zwischen jedem Punkt P der Bodenfläche und dem zugehörigen Bildpunkt P'.

Im Folgenden kurz die Grundidee des Verfahrens mit inverser Projektion:

- Die erste Kamera nimmt ein Bild der Szene auf.
- Da die Abbildungsgeometrie laut Voraussetzung bekannt ist, kann zu jedem Bildpunkt der zugehörige Punkt auf der Bodenfläche in Kamera-Koordinaten berechnet werden; hierbei

handelt es sich um eine inverse Projektion unter der Annahme, dass auf der Bodenfläche kein Hindernis steht.

- Da auch die gegenseitige Anordnung der beiden Kameras bekannt ist, kann nun berechnet werden, an welche Bildposition im Bild der zweiten Kamera dieser Punkt der Bodenfläche abgebildet werden würde und das Pixel wird entsprechend eingefärbt, d.h. es erhält die Farbe des ursprünglichen Pixels aus dem Bild der ersten Kamera. Auf diese Weise kann für die zweite Kamera aus dem Bild der ersten Kamera durch inverse Projektion bezüglich der ersten Kamera und nachfolgende Projektion bezüglich der zweiten Kamera ein vollständiges Abbild der Bodenfläche berechnet werden.
- Im letzten Schritt wird nun mit der zweiten Kamera ein tatsächliches Bild der Szene aufgenommen und dieses Bild wird mit dem künstlich erzeugten Bild verglichen. Wenn die beiden Bilder identisch sind, so befindet sich vor den Kameras kein Hindernis. Falls ein Hindernis vorhanden war, so differieren die beiden Bilder, da das Hindernis bei der künstlichen Berechnung des zweiten Bildes so behandelt wurde, als wäre es auf die Bodenfläche "aufgemalt".

Abbildung 6.2 verdeutlicht an einem Beispiel das Vorgehen, wobei hier der Einfachheit halber von einer **idealen Stereo-Konfiguration** ausgegangen wird, bei der die optischen Achsen der beiden Kameras parallel zueinander verlaufen. Abbildung 6.2 (a) zeigt in einer Seitenansicht die erste Kamera und die Bodenfläche mit einem Hindernis. Wäre das Hindernis nicht vorhanden, so entspräche der Bildpunkt P' dem Punkt P auf der Bodenfläche. Im vorliegenden Fall entspricht P' aber dem Punkt $P^\#$ auf dem Hindernis.

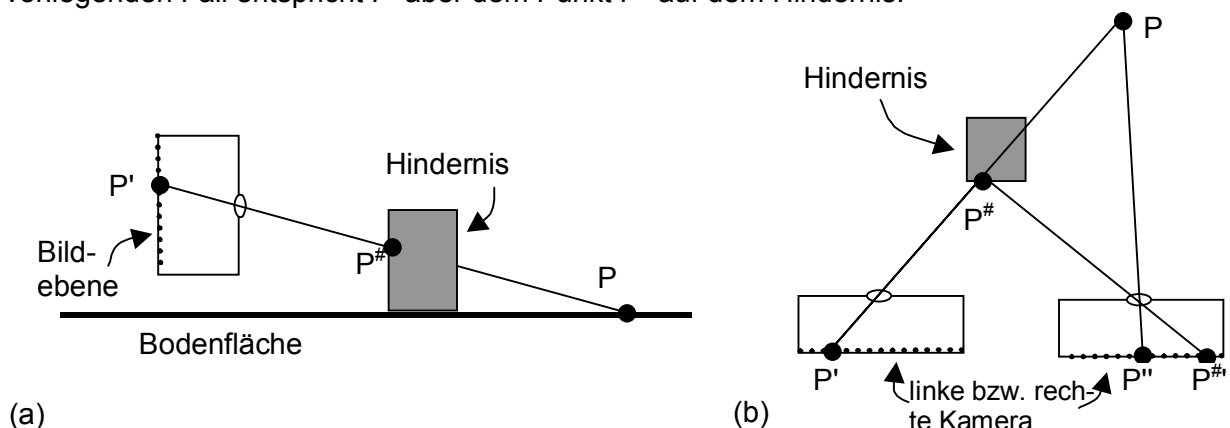


Abb. 6.2: Beispielskizze für die inverse Projektion mit einem Hindernis auf der Bodenfläche:

- Seitenansicht mit erster Kamera, Bodenfläche und Hindernis;
- Draufsicht auf die beiden Kameras und das Hindernis.

Abbildung 6.2 (b) zeigt die Stereo-Kameras und das Hindernis aus der Vogelperspektive (Draufsicht). Für den Bildpunkt P' wird - in der Annahme, dass kein Hindernis vorhanden ist - der Punkt P der Bodenfläche und anschließend der Punkt P'' im Bild der zweiten Kamera berechnet und P'' wird wie P' eingefärbt.

In der Realität, d.h. in dem tatsächlich mit der zweiten Kamera erzeugten Bild, wird der Raumpunkt $P^\#$, der ja die Einfärbung von P' und damit auch von P'' bewirkt, aber auf die Bildposition $P^\#$ abgebildet, die von P'' verschieden ist. Das errechnete und das tatsächlich Bild unterscheiden sich also in dieser und allen anderen dem Hindernis entsprechenden Positionen.

In Abbildung 6.3 ist ein Ergebnis des Verfahrens mit inverser Projektion dargestellt. Abbildung 6.3 (a) zeigt das von der linken Kamera aufgenommene Bild, und in Abbildung 6.3 (b) ist das tatsächlich von der rechten Kamera aufgenommene Bild zu sehen. Das aus dem linken Bild für die rechte Kamera berechnete Bild ist in Abbildung 6.3 (c) zu sehen. Die Differenzbildung zwischen dem berechneten und dem tatsächlich aufgenommenen Bild (siehe Abbildung 6.3 (d)) zeigt, dass die Bilder verschieden sind, dass sich also vor den Kameras ein Hindernis befindet.

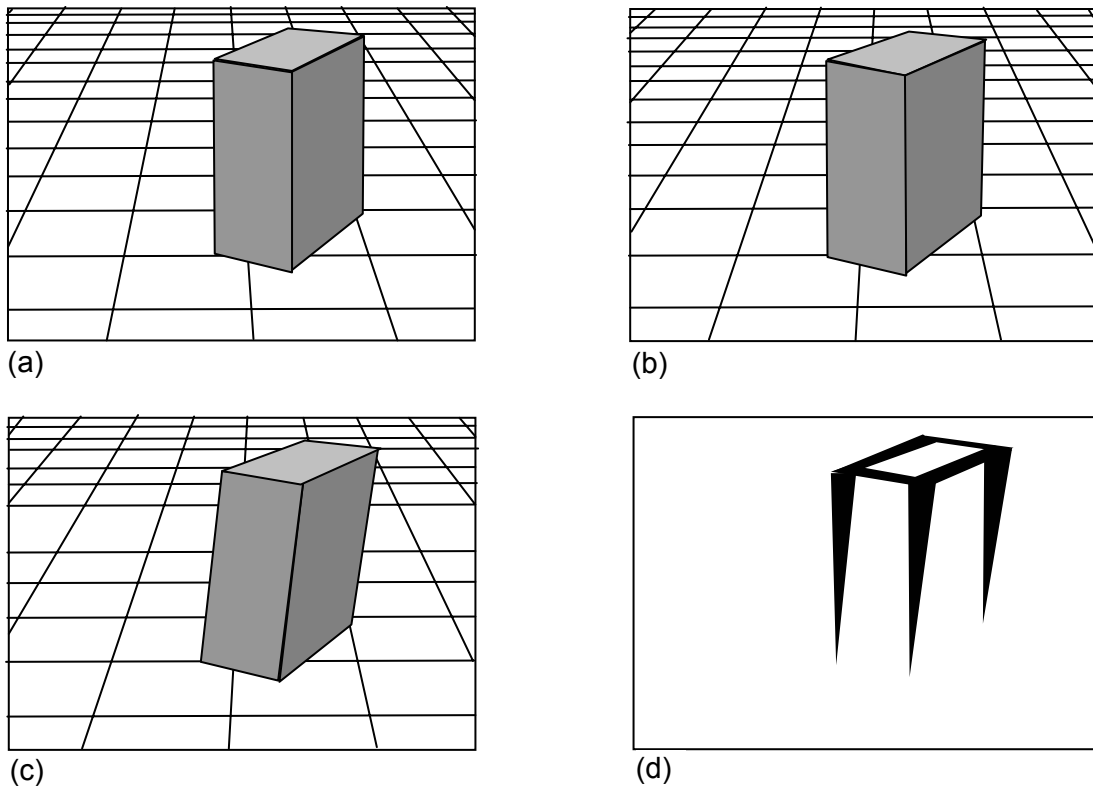


Abb. 6.3: Beispiel für die inverse Projektion mit einem Hindernis auf der Bodenfläche:
 (a) Bild der linken Kamera; (b) tatsächliches Bild der rechten Kamera;
 (c) aus (a) berechnetes Bild der rechten Kamera, (d) Differenzbild.

6.2 Geometrische Grundlagen bzgl. Vergenz und Disparität

Unter **Vergenz** versteht man den bzw. die Anstellwinkel, die die Sehachsen beider Augen (bzw. die optischen Achsen zweier Kameras) bei der Fixierung auf ein in endlicher Entfernung platziertes Objekt einnehmen. Mit Hilfe dieser Winkel kann durch Triangulation die Entfernung zu dem Objekt bzw. einem Raumpunkt P bestimmt werden. Aus Abbildung 6.4 (a) ergibt sich die Entfernung (Tiefe) z zu einem Punkt P, der aktuell fixiert wird, wie folgt:

Es gilt $\cot\alpha = a/z$ und $\cot\beta = b/z$. Durch Addition der beiden Gleichungen und Ersetzen nach $2d = a + b$ ergibt sich $(\cot\alpha + \cot\beta) = 2d / z$ und daraus:

$$(6-1)$$

$$z = 2d / (\cot\alpha + \cot\beta) \quad (2d \text{ ist der Augenabstand bzw. der Abstand der beiden Kameras}).$$

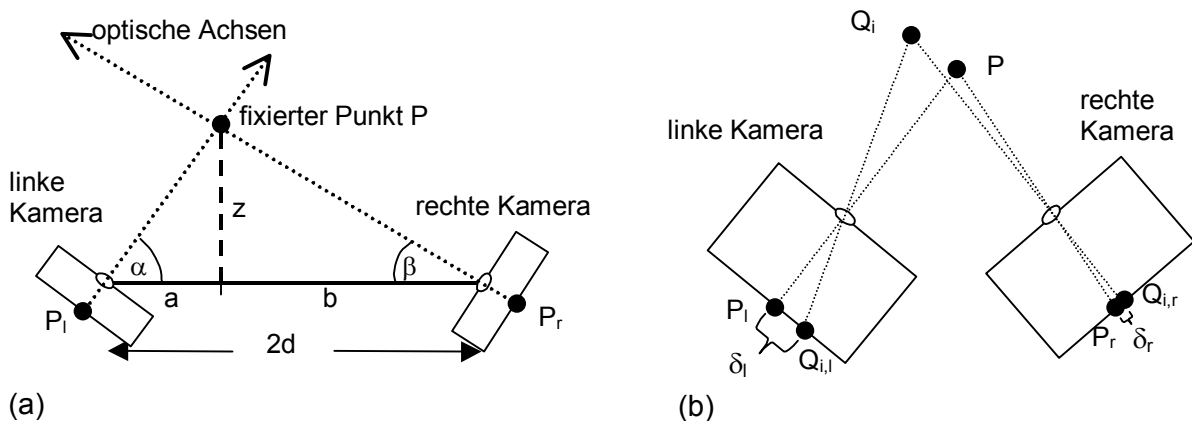


Abb. 6.4: (a) Beispiel aus der Vogelperspektive für eine Triangulation mit Stereokameras (Vergenz);
 (b) Beispiel aus der Vogelperspektive für Disparitäten.

Bei der Fixierung auf einen Punkt P bilden sich alle anderen Raumpunkte Q_i , die einen von P verschiedenen Abstand zum Betrachter haben, auf der Netzhaut bzw. in den Kamera-Bildern in Punkte $Q_{i,l}$ (links) bzw. $Q_{i,r}$ (rechts) ab; siehe Abbildung 6.4 (b). Diese Punkte besitzen im linken und im rechten Bild einen unterschiedlichen Abstand zum Abbild des Punktes P (Daumenexperiment!). Diese Differenzen der Abbilder werden als **Disparitäten** δ_l und δ_r (für linke und rechte Disparität) bezeichnet. Anhand dieser Disparitäten lässt sich für jeden Raumpunkt Q_i seine Tiefe bestimmen.

Diese Berechnung sei am Beispiel zweier Kameras K_l und K_r in der x - z -Ebene, d.h. aus der Vogelperspektive erläutert (siehe Abbildung 6.5). Die beiden Kameras, bzw. stellvertretend die beiden punktförmigen Blenden seien auf der x -Achse bei $x_{K,l} = -d$ und $x_{K,r} = d$ angebracht, d.h. $K_l = (-d, 0)$ und $K_r = (d, 0)$ im x - z -Koordinatensystem. Die Kameras sollen der Einfachheit halber parallel zur z -Achse orientiert sein, d.h. auf $z = \infty$ fixiert sein (ideale Stereo-Konfiguration). Beide sollen die Brennweite f haben. Der Fixationspunkt ∞ wird also in der linken Bildebene auf $\infty_l = (-d, -f)$ und rechts auf $\infty_r = (d, -f)$ abgebildet.

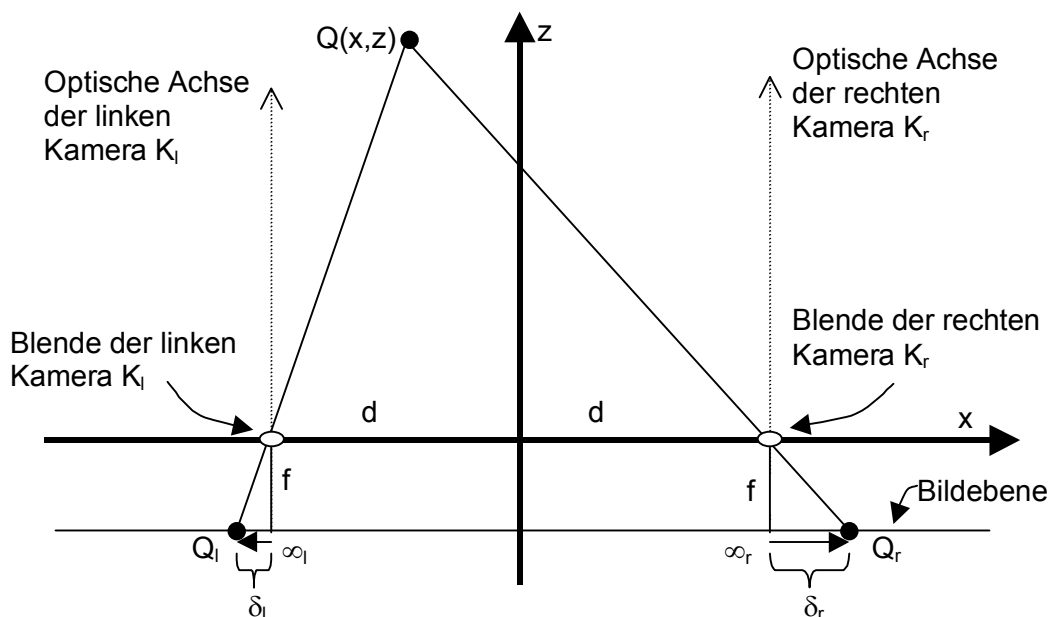


Abb. 6.5: Beispiel für die Berechnung der Tiefe z eines Punktes Q ; (Vogelperspektive).

Ein beliebiger Punkt $Q(x, z)$ werde links auf $Q_l = (x_{Q,l}, -f)$ und rechts auf $Q_r = (x_{Q,r}, -f)$ abgebildet. Die Disparitäten δ_l und δ_r ergeben sich zu: $\delta_l = x_{Q,l} - x_{K,l} = x_{Q,l} + d$ und $\delta_r = x_{Q,r} - x_{K,r} = x_{Q,r} - d$; (δ_l und δ_r sind verschieden groß und im Beispiel auch unterschiedlich orientiert).

Links sind die Dreiecke mit den Hypotenusen $\overline{Q_l K_l}$ und $\overline{K_l Q}$ ähnlich und rechts die mit den Hypotenusen $\overline{Q_r K_r}$ und $\overline{K_r Q}$. Daraus folgt:

$$\delta_l / f = -(d+x)/z \quad \text{bzw.} \quad \delta_r / f = (d-x)/z$$

Hieraus folgt durch Elimination von x :

$$z * (\delta_r - \delta_l) = 2df$$

und weiter das Endergebnis:

$$z = 2df / (\delta_r - \delta_l), \quad \text{sowie durch Einsetzen in oben: } x = \frac{-d * (\delta_r + \delta_l)}{(\delta_r - \delta_l)}. \quad (6-2)$$

6.3 Korrespondenzproblem

Die oben angegebene Formel erlaubt also direkt die Berechnung der Tiefe eines beliebigen Punktes Q in der x - z -Ebene (und entsprechend erweitert im x - y - z -Raum), sofern die beiden korrespondierenden Punkte Q_l und Q_r im linken und im rechten Bild bekannt sind. In der Regel ist dies aber nicht der Fall, und man muss zu einer Bildstelle Q_l im linken Bild erst die

entsprechende Stelle Q_r im rechten Bild suchen. Diese Suche wird als **Korrespondenzproblem** bezeichnet.

Für das Korrespondenzproblem gibt es in der Praxis drei unterschiedliche Ansätze: den grauwert-, den merkmals- und den inhaltsorientierten Ansatz; siehe Abbildung 6.6.

Beim **grauwertorientierten Ansatz** wird zu einem "markanten" Teilbild aus dem einen Bild das entsprechende Teilbild im anderen Bild durch direkten (bildhaften) Vergleich der Grauwertuntermatrizen der Teilbilder automatisch gefunden. Hierfür wird i.d.R. die normalisierte Korrelation verwendet (siehe Kapitel 3.3).

Beim **merkmalsorientierten Ansatz** werden die beiden Bilder zuerst verarbeitet, indem z.B. mit einem Kanten- oder Liniendetektor Kanten bzw. Linien extrahiert werden. Zur Ermittlung korrespondierender Stellen werden nun markante Kanten oder Linien (z.B. signifikante Ecken, Verzweigungen oder Kreuzungen) ausgewählt, und diese werden danach im anderen Bild gesucht. Bei nebeneinander angeordneten Kameras, die mit demselben Neigungswinkel in dieselbe Richtung schauen, sind z.B. auch senkrecht verlaufende Kanten- und Linienstücke für diese Zuordnung geeignet, waagrechte dagegen nicht (warum?).

Und bei der **inhaltsorientierten** dritten Alternative werden beide Bilder zuerst interpretiert und dann werden die korrespondierenden Objekte gesucht; z.B.: wo liegt der rote Ball in beiden Bildern.

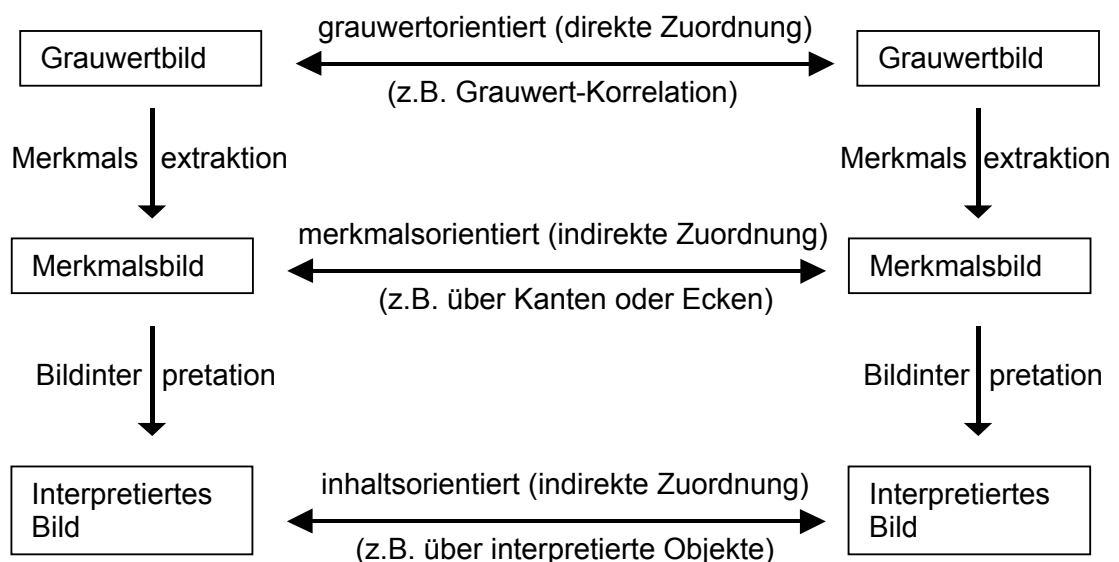


Abb. 6.6: Drei unterschiedliche Ansätze zur Lösung des Korrespondenzproblems.

Zur Suche nach einer Bildstelle Q_r im rechten Bild, die mit einer Bildstelle Q_l im linken Bild korrespondiert, muss nicht das gesamte rechte Bild abgesucht werden. Der **Sehstrahl**, der bzgl. der ersten Bildstelle Q_l durch die Kamerablende in den 3D-Raum zum betrachteten Objekt Q führt (und dessen Länge bis zum Punkt Q im Endeffekt ermittelt werden soll), wird im zweiten Bild auf eine Gerade, die sogenannte **Epipolare**, abgebildet (siehe Abbildung 6.7). Nur entlang dieser Geraden muss im rechten Bild nach der entsprechenden Bildstelle gesucht werden.

Wenn es zu jeder markanten Bildstelle im linken Bild genau eine "passende" Bildstelle im rechten Bild gibt, so kann diese mit also mit dem grauwert-, merkmals- oder inhaltsbasierten Ansatz gefunden werden, und das Korrespondenzproblem ist gelöst, so dass anschließend die Entfernungen zu den zugehörigen Objekten berechnet werden können.

In der Praxis kann es aber (vor allem beim grauwert- oder merkmalsorientierten Ansatz) vorkommen, dass z.B. mehrere Bildstellen im rechten Bild zu einer aktuell im linken Bild untersuchten Bildstelle passen und umgekehrt.

- 2.) **Kontinuität:** Die Disparität ändert sich i.d.R. langsam in Abhängigkeit vom Ort, (d.h. reale Objekte haben kontinuierliche Oberflächen, und Sprünge in der Tiefe treten nur an Objektgrenzen auf).

Während die erste Regel hilft, widersprüchliche Kombinationen zwischen den L_i und R_i zu eliminieren, hilft die zweite bei der Entscheidung zwischen mehreren möglichen Anordnungen (und bevorzugt bei dem Beispiel in Abbildung 6.8 die Interpretation mit den schwarzen Kreisscheiben).

Ausgehend von diesen beiden Regeln schlagen die Autoren ein **kooperatives Netzwerk** vor. Dieses und die darauf ablaufende Relaxation sollen im weiteren am Beispiel zweier Zeilen aus einem Zufallsstereogramm erläutert werden (siehe Abbildung 6.9). Abbildung 6.9 (a) zeigt die Zeilen aus dem linken bzw. rechten Bild des Stereogrammes: das durch O gekennzeichnete "Objekt" ist im rechten Bild vor dem durch H gekennzeichneten "Hintergrund" um einen Bildpunkt nach links verschoben, besitzt also die Disparität 1.

Die beiden Stereogrammzeilen wirken nun von links oben und rechts oben diagonal (\backslash bzw. $/$), d.h. entlang der jeweiligen Sehstrahlen auf das Netzwerk ein. Die Zellen des Netzwerkes können die Werte 1 (aktiv) und 0 (inaktiv) einnehmen. Zunächst wird ein Anfangszustand C^0 für das gesamte Netzwerk eingestellt (Abbildung 6.9 (b)). Ein Element erhält dabei den Wert Eins zugewiesen (Zelle aktiv, da mögliche Korrespondenz der entsprechenden Bildelemente), wenn es im Schnittpunkt zweier Diagonalen liegt, die in der linken bzw. rechten Stereogrammzeile von gleich markierten Bildelementen ausgehen. Bei Ungleichheit der betreffenden Bildelemente wird Null ins Netzwerk eingetragen (Zelle inaktiv; keine Korrespondenz der entsprechenden Bildelemente möglich). Die Vielzahl möglicher Einzel-Korrespondenzen wird also mit Einsen bezeichnet.

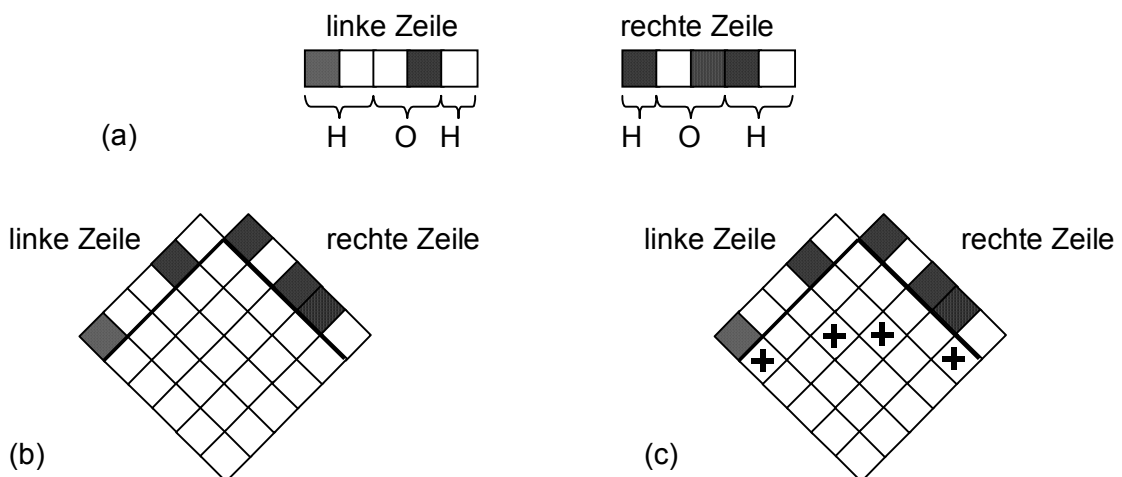


Abb. 6.9: Vorbereitetes Beispiel für ein kooperatives Netz.

Die dem Netzwerk zugrunde liegende Idee besagt, dass einerseits benachbarte aktive Zellen der selben Disparitätsebene (d.h. Waagrechten) sich gegenseitig verstärken sollen, um dadurch "glatte" Oberflächen zu erzeugen (Zwangsbedingung 2), andererseits sollen sich entlang der Sehstrahlen (Diagonalen \backslash und $/$) benachbarte aktive Zellen gegenseitig abschwächen um dadurch Fehlzuordnungen zu eliminieren (Zwangsbedingung 1). Die Erregungen bzw. Hemmungen wirken nun im Rahmen eines iterativen Prozesses auf die einzelnen Elemente des Netzwerkes ein, bis sich ein konsistentes Ergebnis einstellt (Relaxation!). Abbildung 6.9 (c) zeigt das Endergebnis, bei dem das Objekt in der Disparitätsebene 1 gefunden wird. (Bemerkung: bei einem derart kleinen Netz kann das Endergebnis nicht rechnerisch nachvollzogen werden!)

Das geschilderte Vorgehen entspricht für alle Zellen des Netzwerkes dem folgenden iterativen Algorithmus:

$$C^{n+1} = \sigma \{ \sum(C^n) - H * \sum(C^n) + C^0 \} \quad (6-3)$$

Dabei läuft die erste Summe über das jeweils betrachtete Element von C^n sowie seine verstärkenden, d.h. waagrechten Nachbarn. Die zweite Summe läuft über das jeweils betrachtete Element von C^n sowie seine abschwächenden, d.h. diagonalen Nachbarn. H ist ein zusätzlicher Hemmfaktor und σ ist eine sogenannte Sprungfunktion bzgl. eines Schwellwertes S . Wenn der Wert des Klammerausdruckes bei σ den Schwellwert S übersteigt, liefert σ den Wert 1, andernfalls den Wert 0 (vgl. Binarisierung).

Die Autoren haben gezeigt, dass das Verfahren für eine Vielzahl von Werten H und S konvergiert und das korrekte Korrespondenzergebnis liefert. Das Verfahren kann auf den räumlichen Anwendungsfall erweitert werden, indem zu den übereinanderliegenden Bildzeilen der beiden Stereobilder ein dreidimensionales Netzwerk gewählt wird. Eine weitere Fortführung des Verfahrens durch Grimson erlaubt (in Grenzen) die Übertragung auf Bilder natürlicher Szenen. In beiden Bildern eines Stereopaars wird dabei versucht, einander entsprechende Konturabschnitte zu finden.

6.5 PMF-Algorithmus zur Stereobildauswertung

Der **PMF-Algorithmus** nach Pollard, Mayew und Frisby (s. z.B. [Sonka et al.]) löst bei idealer Stereo-Konfiguration Mehrdeutigkeiten beim Korrespondenzproblem durch Relaxation mit Hilfe des sogenannten Disparitätsgradienten.

Auch hier sollen wieder möglichst kontinuierliche Oberflächen gegenüber „Tiefensprüngen“ bevorzugt werden; siehe Zwangsbedingungen am Anfang von Kapitel 6.4.

Die Grundidee des PMF-Algorithmus ist, dass aus den Kandidaten im linken bzw. rechten Bild alle möglichen Kandidaten-Paare (d.h. Positionen) im Raum vor den Kameras gebildet werden; vgl. Abbildung 6.8. Aus diesen Positionen wird diejenige ausgewählt, zu der es die meisten auf anderen Sehstrahlen liegenden Positionen in ungefähr gleicher Entfernung zur Kamera gibt. Anhand dieser Position wird einem Kandidaten im linken Bild ein Kandidat im rechten Bild als „korrespondierend“ zugeordnet. Zu den verbleibenden Kandidaten im linken bzw. rechten Bild werden wieder alle möglichen Positionen vor den Kameras betrachtet; wie zuvor wird diejenige ausgewählt, die mit den meisten anderen in ähnlicher Entfernung zur Kamera liegt, usw..

Zur Bewertung, ob zwei Positionen im Raum vor den Kameras ungefähr im gleichen Abstand liegen, wird der sogenannte „Disparitätsgradient“ berechnet. Dazu führen die Autoren ein künstliches "Zyklopenbild" ein, das von einem fiktiven Zyklopedenauge, bzw. einer entsprechenden Kamera aufgenommen würde, die in der Mitte zwischen den beiden Stereokameras sitzt; siehe Abbildung 6.10.

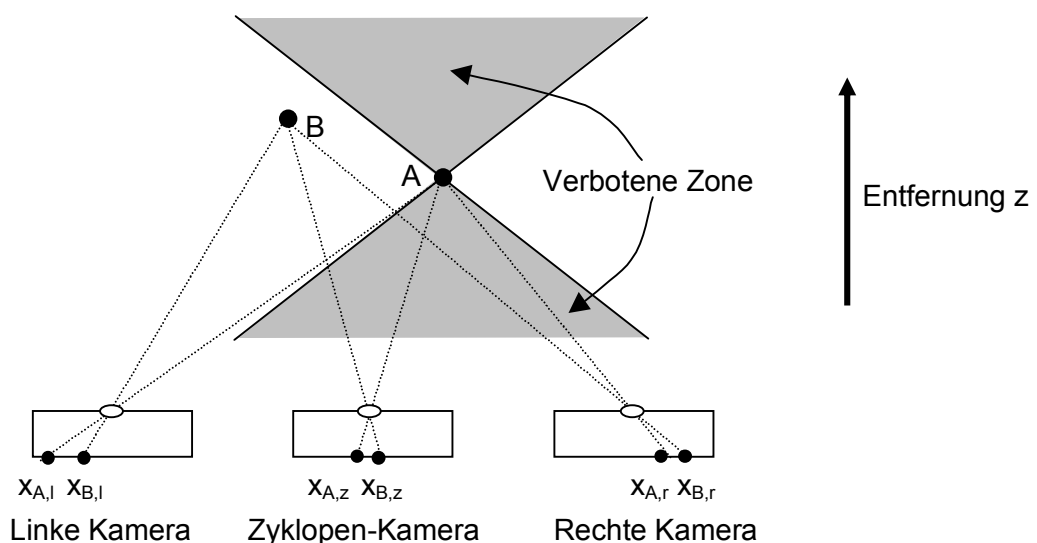


Abb. 6.10: Prinzipbild zum Zyklopedenbild und zur "verbotenen Zone" (Draufsicht).

Seien A und B zwei Punkte in der Szene, zu denen im Weiteren der Disparitätsgradient berechnet werden soll. Die Punkte A und B werden im linken und im rechten Bild auf $A_l=(x_{A,l}, y_A)$ und $B_l=(x_{B,l}, y_B)$ bzw. $A_r=(x_{A,r}, y_A)$ und $B_r=(x_{B,r}, y_B)$ abgebildet. (Aufgrund der Epipolar-Geometrie der idealen Stereo-Konfiguration liegen die beiden Punkte A_l und A_r bzw. B_l und B_r in beiden Bildern auf derselben Höhe, d.h. es gilt: $y_{A,l}=y_{A,r}$ und $y_{B,l}=y_{B,r}$; diese Höhen werden im Weiteren als y_A bzw. y_B bezeichnet).

Die zugehörigen Punkte A_z und B_z des Zyklopenbildes ergeben sich zu:

$$A_z = \left(\frac{x_{A,l} + x_{A,r}}{2}, y_A \right) \text{ und } B_z = \left(\frac{x_{B,l} + x_{B,r}}{2}, y_B \right) \quad (6-4)$$

Zur Berechnung des Disparitätsgradienten $DG(A,B)$ zu den beiden Punkten A und B werden nun zwei Hilfsgrößen eingeführt:

1.) **Zyklopen-Abstand $S(A,B)$** , d.h. Abstand der Bildpunkte A_z und B_z im Zyklopenbild; siehe Abbildung 6.11:

$$\begin{aligned} S(A,B) &= \sqrt{(x_{A,z} - x_{B,z})^2 + (y_A - y_B)^2} = \sqrt{\left(\left(\frac{x_{A,l} + x_{A,r}}{2} \right) - \left(\frac{x_{B,l} + x_{B,r}}{2} \right) \right)^2 + (y_A - y_B)^2} = \\ &= \sqrt{\left(\frac{1}{2} * ((x_{A,l} - x_{B,l}) + (x_{A,r} - x_{B,r})) \right)^2 + (y_A - y_B)^2} = \sqrt{\left(\frac{1}{2} * (\delta_l + \delta_r) \right)^2 + (y_A - y_B)^2} \end{aligned}$$

(Geometrisches Mittel der beiden Disparitäten) \longleftarrow

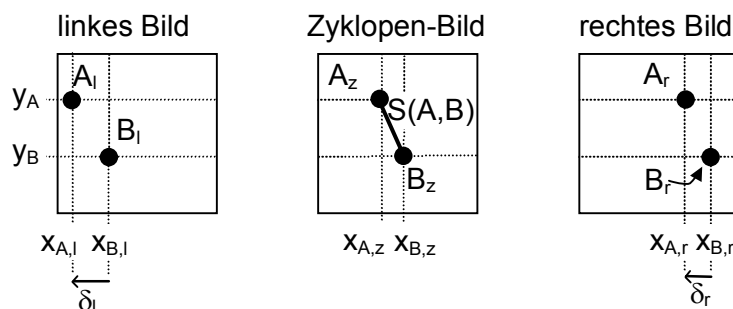


Abb. 6.11: Prinzipbild zur Herleitung des Zyklopen-Abstandes und der Disparitäten-Differenz.

2.) **Disparitäten-Differenz $D(A,B)$** , d.h. Differenz der Disparitäten δ_l und δ_r ; s. Abbildung 6.11:

$$D(A,B) = (x_{A,r} - x_{A,l}) - (x_{B,r} - x_{B,l}) = (x_{A,r} - x_{B,r}) - (x_{A,l} - x_{B,l}) = \delta_r - \delta_l \quad (6-6)$$

Aus (6-5) und (6-6) wird der **Disparitäts-Gradient $DG(A,B)$** gebildet:

$$DG(A,B) = \frac{|D(A,B)|}{|S(A,B)|} \quad (6-7)$$

Der Disparitäts-Gradient kann als Maß für die Steigung der Geraden angesehen werden, die durch die Punkte A und B verläuft; siehe Abbildung 6.10.

Psychophysikalische Experimente legen den Schluss nahe, dass die Zuverlässigkeit der Tiefenwahrnehmung beim Menschen bei wachsendem Disparitäts-Gradienten abnimmt. Das bedeutet, dass z.B. zwei in x-Richtung dicht benachbarte Punkte nur dann in ihrer gegenseitigen räumlichen Beziehung richtig erkannt werden können, wenn sich ihre Disparitäten nur geringfügig unterscheiden.

In Abbildung 6.10 ist als Beispiel der Bereich der Szene als "verbotene Zone" eingezeichnet, in dem der Disparitäts-Gradient den Wert 1 übersteigt. In diesem Bereich wäre die Erkennung der räumlichen Lage eines Punktes B in Bezug auf Punkt A unsicher oder sogar unmöglich.

Die oben erläuterte Beschränkung des Disparitäts-Gradienten stellt eine der beiden Zwangsbedingungen dar (**disparity gradient constraint**), die beim PMF-Algorithmus zur Lösung des Korrespondenzproblems eingesetzt werden. Die andere Zwangsbedingung ist - wie bei Marr und Poggio - die Eindeutigkeit der Raumtiefe.

Im Folgenden ist der Ablauf des PMF-Algorithmus geschildert, der aus der Menge der möglichen Punkt-Korrespondenzen (Positionen) iterativ die „besten“ Positionen auswählt. Das Vorgehen basiert auf einem Relaxationsverfahren, das zu jeder möglichen Position untersucht, wie gut sie durch die anderen Positionen unterstützt wird:

1. Ermittlung markanter Stellen in den beiden Stereobildern.
2. Bildung aller möglichen Kandidaten-Paare (Positionen) von korrespondierenden Bildstellen im Raum vor den Kameras; (beispielsweise wird zu jeder Bildstelle im linken Bild getestet, welche Bildstellen im rechten Bild unter Berücksichtigung der Epipolargeometrie als korrespondierende Bildstellen in Frage kommen).
3. Berechnung des Disparitäts-Gradienten für jede Zweier-Kombination von Kandidaten-Paaren, die die Eindeutigkeits-Zwangsbedingung nicht verletzt. Falls der Disparitäts-Gradient einen vorgegebenen Schwellwert nicht überschreitet, so wird für die beiden beteiligten Kandidaten-Paare ein "Wahrscheinlichkeits-Zähler" um Eins erhöht.
4. Das Kandidaten-Paar mit dem höchsten Wert des Wahrscheinlichkeits-Zählers wird als "korrekt zugeordnet" angesehen. Alle anderen Kandidaten-Paare, die einen der beiden Kandidaten dieses Paares beinhalten, werden wegen der Eindeutigkeits-Constraint im Hinblick auf die weitere Verarbeitung gelöscht.
5. Weiter mit 2.), wobei die bereits zugeordneten Kandidaten-Paare bei der Berechnung der Wahrscheinlichkeits-Zähler berücksichtigt werden. (Die Disparitäts-Gradienten müssen dabei nicht neu berechnet werden).
6. Das Verfahren endet, wenn iterativ alle Bildstellen zugeordnet wurden.

In Abbildung 6.12 ist ein Beispiel für den PMF-Algorithmus vorbereitet.

Erweiterung des Verfahrens: Bei der Bewertung der Kombinationen der Kandidaten-Paare muss berücksichtigt werden, dass die Unterschreitung des Schwellwertes für den Disparitäts-Gradienten umso wahrscheinlicher wird, je weiter die Kandidaten-Paare auseinander liegen. Um dies zu berücksichtigen ...

... können z.B. nur solche Kandidaten-Paare kombiniert werden, die nicht zu weit auseinander liegen;

... können z.B. die "erfolgreichen" Kombinationen von Kandidaten-Paaren mit dem Kehrwert ihres Abstandes gewichtet werden. Dadurch würden weiter voneinander entfernte Kandidaten-Paare geringer bewertet.

Beispiel für den Ablauf des PMF-Algorithmus (siehe Abbildung 6.12)

Der Einfachheit halber wird hier vorausgesetzt, dass alle markanten Bildstellen in derselben "Höhe" im Bild liegen, so dass der Ausdruck $(y_A - y_B)$ in (6-5) weggelassen werden kann und deshalb auch auf das Quadrieren und Wurzelziehen verzichtet werden kann:

$$DG(b1, a3) = \frac{|D(b1, a3)|}{|S(b1, a3)|} = \frac{|4 - 10|}{|-2 + 1|} = \frac{6}{1} > 1$$

$$DG(b1, c2) = \frac{|D(b1, c2)|}{|S(b1, c2)|} = \frac{|4 - 4|}{|-2 - 0|} = \frac{0}{2} \leq 1 \quad \text{deshalb: } WZ(b1) = 1 \quad \text{und } WZ(c2) = 2$$

$$DG(b1, c3) = \frac{|D(b1, c3)|}{|S(b1, c3)|} = \frac{|4 - 6|}{|-2 - 1|} = \frac{2}{3} \leq 1 \quad \text{deshalb: } WZ(b1) = 2 \quad \text{und } WZ(c3) = 2$$

$$DG(c1, b2) = \frac{|D(c1, b2)|}{|S(c1, b2)|} = \frac{|2 - 6|}{|-1 + 1|} = \frac{4}{0} \quad \text{nicht definiert}$$

$$DG(c1, b3) = \frac{|D(c1, b3)|}{|S(c1, b3)|} = \frac{|2 - 8|}{|-1 - 0|} = \frac{6}{1} > 1$$

$$DG(c1, a2) = \frac{|D(c1, a2)|}{|S(c1, a2)|} = \frac{|2 - 8|}{|-1 + 2|} = \frac{6}{1} > 1$$

$$DG(c1, a3) = \frac{|D(c1, a3)|}{|S(c1, a3)|} = \frac{|2 - 10|}{|-1 + 1|} = \frac{8}{0} \quad \text{nicht definiert}$$

$$DG(a2, b3) = \frac{|D(a2, b3)|}{|S(a2, b3)|} = \frac{|8 - 8|}{|-2 - 0|} = \frac{0}{2} \leq 1 \quad \text{deshalb: } WZ(a2) = 1 \quad \text{und } WZ(b3) = 2$$

$$DG(a2, c3) = \frac{|D(a2, c3)|}{|S(a2, c3)|} = \frac{|8 - 6|}{|-2 - 1|} = \frac{2}{3} \leq 1 \quad \text{deshalb: } WZ(a2) = 2 \quad \text{und } WZ(c3) = 3$$

$$DG(b2, a3) = \frac{|D(b2, a3)|}{|S(b2, a3)|} = \frac{|6 - 10|}{|-1 + 1|} = \frac{4}{0} \quad \text{nicht definiert}$$

$$DG(b2, c3) = \frac{|D(b2, c3)|}{|S(b2, c3)|} = \frac{|6 - 6|}{|-1 - 1|} = \frac{0}{2} \leq 1 \quad \text{deshalb: } WZ(b2) = 2 \quad \text{und } WZ(c3) = 4$$

$$DG(c2, a3) = \frac{|D(c2, a3)|}{|S(c2, a3)|} = \frac{|4 - 10|}{|0 + 1|} = \frac{6}{1} > 1$$

$$DG(c2, b3) = \frac{|D(c2, b3)|}{|S(c2, b3)|} = \frac{|4 - 8|}{|0 - 0|} = \frac{4}{0} \quad \text{nicht definiert}$$

Die Zählerstände ergeben sich zu:

$$WZ(a1) = 4, \quad WZ(a2) = 2, \quad WZ(a3) = 0,$$

$$WZ(b1) = 2, \quad WZ(b2) = 2, \quad WZ(b3) = 2,$$

$$WZ(c1) = 0, \quad WZ(c2) = 2 \quad WZ(c3) = 4.$$

4. Der höchste Zählerstand ist 4 und er tritt bei den Kandidaten-Paaren a1 und c3 auf.
5. Nach Entfernung von a, c, 1 und 3 verbleibt als mögliches Kandidaten-Paar nur b2, das infolge dessen auch als korrekt angesehen wird.
6. Ende des Verfahrens, da alle Bildstellen zugeordnet wurden.

Der PMF-Algorithmus hat also aus der Menge der korrekten Zuordnungen diejenige ausgewählt, bei der die Objekte alle in derselben Raumtiefe (= Disparitäts-Ebene 0) liegen. Dieses Ergebnis entspricht auch der zweiten Zwangsbedingung (Kontinuität) beim Algorithmus von Marr und Poggio.

7 Bildfolgenauswertung

Biologische (visuelle) Wahrnehmungssysteme sind darauf optimiert, Änderungen, d.h. z.B. Bewegungen im Blickfeld zu erfassen. Da die Systeme sich in einer sich ständig verändernden Umgebung selbst bewegen, ist dies eine Grundvoraussetzung für das Überleben (z.B. Entdeckung von Angreifern, Beutefang...).

Der größte Teil der automatischen Bildauswertung ist im Gegensatz dazu auf die Auswertung stationärer Szenen bei stationärem Sensor ausgerichtet. Für zahlreiche Anwendungsfälle ist aber eine Auswertung zeitlich gestaffelter Bildinformation, d.h. sogenannter **Bildfolgen** erforderlich, z.B.:

- Auswertung von medizinischen Röntgenbildfolgen unter Verwendung von Kontrastmitteln zur Untersuchung von Kreislaufphänomenen, Stoffwechselfvorgängen oder Ähnlichem,
- Überwachung von Verkehrsanlagen, wie z.B. Bahnsteigen,
- Auswertung von Wolkenverschiebungen in Luft- oder Satellitenbildern zur Erfassung der aktuellen Windverteilungen für die Erstellung von Wettervorhersagen.
- Überwachung von Fließbändern oder Arbeitsräumen von industriellen Robotern in der Fertigung,
- Steuerung autonomer Roboter

Mitunter ist die Aufgabe der automatischen Bildfolgenauswertung nicht die Bewegungserkennung, sondern nur die einfache Änderungsentdeckung (z.B. zur Auslösung eines Alarms bei einer Überwachungsanlage). Diese eingeschränkte Aufgabe lässt sich z.B. mit Hilfe der Bilddifferenz lösen (siehe Kapitel 3.1). Die Subtraktion der Grauwerte zweier aufeinanderfolgender Bilder der Bildfolge liefert direkt alle Änderungen: alle Bildstellen, an denen die Grauwertdifferenz ungleich null ist.

Neben der Änderungsentdeckung und Bewegungserkennung erlaubt die Auswertung von Bildfolgen auch Rückschlüsse auf die 3D-Gestalt der abgebildeten Umgebung bzw. der Objekte (**structure from motion**). So können aus den Bewegungsparallaxen (Winkelgeschwindigkeiten) Entfernungen bzw. relative Entfernungen bzgl. des Sensors abgeleitet werden. Ebenso ist unter bestimmten Zusatzvoraussetzungen die Ermittlung von Oberflächenneigungen aus Bildfolgen möglich.

Eine der Kernfragen bei der Bewegungsverarbeitung ist, zu welchem Zeitpunkt diese während des Gesamtlaufes der Auswertung einer Bildfolge stattfinden soll:

- wertet man zuerst jedes Einzelbild aus (Segmentierung in Regionen) und versucht danach die zeitliche Zuordnung zwischen den Regionen in aufeinanderfolgenden Bildern der Bildfolge, oder
- ermittelt man zuerst Änderungen bzw. Bewegungen und stützt damit die nachfolgende Segmentierung. (Ein gut getarntes Tier ist in einem Einzelbild nicht erkennbar, und erst seine Bewegungen in einer Bildfolge ermöglichen seine Entdeckung).

Für beide Vorgehensweisen gibt es Bildverarbeitungsansätze, und diese werden im weiteren kurz vorgestellt. In Kapitel 7.1 werden zunächst die Grundlagen der Bildfolgenauswertung besprochen. In Kapitel 7.2 wird dann ein relativ einfaches Verfahren vorgestellt, mit dem Kollisionen zwischen der Kamera (d.h. z.B. einem Roboter) und Szenenobjekten vorhergesagt und somit vermieden werden können. In Kapitel 7.3 wird ein alternativer Ansatz mit differentieller Betrachtungsweise zur Bildfolgenauswertung erläutert.

7.1 Grundlagen der Bildfolgenauswertung

Die Änderungsentdeckung bzw. Bewegungserkennung soll im weiteren am Beispiel von Grauwert-Bildfolgen $(B_t) = (g(x, y, t))$ ($t = 0, 1, \dots, T$) betrachtet werden. Die vorgestellten Verfahren beziehen sich dabei i.d.R. auf zwei zeitlich aufeinanderfolgende Bilder $B_t = g(x, y, t)$ und $B_{t+1} = g(x, y, t+1)$; daraus lässt sich das Vorgehen zur Verarbeitung einer gesamten Bildfolge ableiten.

Bei Bildfolgen, die bei einer festen Bewegungs- und Blickrichtung des Sensors erzeugt wurden, scheint die Landschaft aus einem Punkt, dem sogenannten **Fluchtpunkt (focus of expansion)** "herauszufließen", (z.B. bei Fahrt auf einer langen geraden Straße mit Blickrich-

tung nach vorne). Das "Fließen", das sich radial vom Fluchtpunkt weg bewegt, verläuft in der Nähe des Fluchtpunktes langsam und nimmt nach außen hin in der Geschwindigkeit zu. Objekte, die sich relativ zum Sensor bewegen, rufen also in der Bildfolge (bzw. auf der Retina des Auges) ein Fließmuster hervor, das als **optischer Fluss** (*optical flow*) bezeichnet wird; siehe Abbildung 7.1.

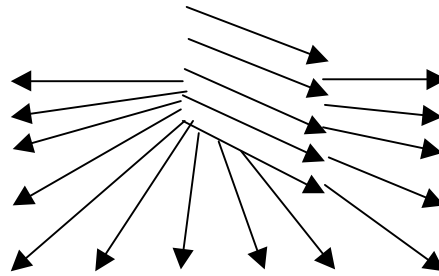


Abb. 7.1: Beispiel für einen optischen Fluss: die Kamera bewegt sich geradlinig "ins Bild hinein" und gleichzeitig bewegt sich ein Objekt nach rechts.

Aus diesem Flussfeld lassen sich ohne Kenntnis von "höheren" Informationen bereits z.B. Zeitpunkt und Ort einer möglichen Kollision ableiten; siehe auch Kapitel 7.2.

Weiterhin offenbart das Flussfeld über die **Bewegungsparallaxen** (Winkelgeschwindigkeiten) direkt weitere Erkenntnisse über den abgebildeten Raum: entfernte Objekte bewegen sich bei einer seitlichen Verschiebung des Betrachters langsamer als nahe gelegene Objekte, so dass die relative Entfernung, d.h. das Abstandsverhältnis) dieser Objekte geschätzt werden kann (structure from motion). Auf die 3D-Strukturerkennung soll hier nicht näher eingegangen werden.

Durch den optischen Fluss wird jedem Bildpunkt ein zweidimensionaler **Verschiebevektor** VV (mit Richtung und Betrag) zugewiesen. Die Gesamtheit der Verschiebevektoren wird als **Verschiebevektorfeld** VVF bezeichnet; siehe Abbildung 7.2.

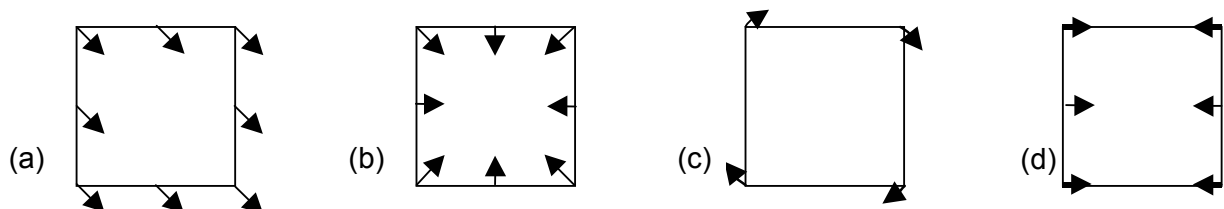


Abb. 7.2: Beispiele für einfache Verschiebevektorfelder:

- (a) Translation parallel bzw. (b) senkrecht zur Bildebene;
 (c) Rotation um eine Achse senkrecht bzw. (d) parallel zur Bildebene.

In der Realität lassen sich die Verschiebevektoren nicht für alle Bildpunkte bestimmen. Z.B. lassen sich in homogenen Bereichen und entlang von Kanten keine Verschiebungen ermitteln (**Blendenproblem**); siehe Abbildung 7.3. Bewegungen können nur dort entdeckt werden, wo sie zu Änderungen von Grauwerten führen, d.h. wo wandernde Helligkeitsmuster auftreten. Umgekehrt entspricht aber nicht jede Grauwertänderung einer Bewegung, denn die Ursache für die Änderung kann z.B. in einer Beleuchtungsänderung liegen.

Außerdem muss beachtet werden, dass ein isoliert betrachteter 2D-Verschiebevektor beliebig viele 3D-Interpretationen besitzt. Diese Mehrdeutigkeit kann erst durch den räumlichen und zeitlichen Kontext aufgelöst werden.

Bei der Bewegungserkennung ist die Schwierigkeit der Aufgabe stark abhängig vom zeitlichen Abstand aufeinanderfolgender Bilder der Bildfolge und der Geschwindigkeit der Bewegungen. Bei geringfügigen örtlichen Veränderungen von Bild zu Bild ist die Ermittlung korrespondierender Bildstellen einfach; bei größeren Änderungen tritt wie bei der Auswertung von Stereobildern das **Korrespondenzproblem** (vgl. Kapitel 6.3) auf. Während korrespondierende Bildstellen bei der Stereobildauswertung im Partnerbild nur entlang einer Geraden gesucht

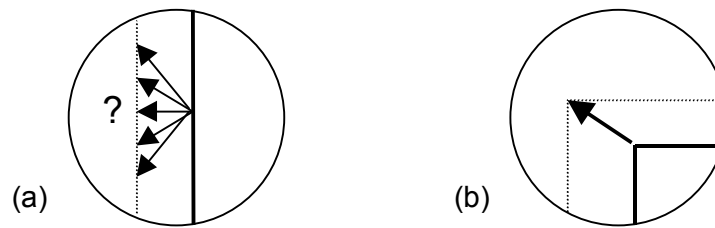


Abb. 7.3: Beispiele zum Blendenproblem:

- (a) Mehrdeutigkeit der Verschiebevektoren an einer Kante,
 (b) Eindeutigkeit der Verschiebevektoren an einer Ecke.

werden mussten (Epipolare), treten hier zweidimensionale Suchbereiche auf, die lediglich durch die folgenden fünf Heuristiken eingeschränkt werden:

- 1.) Für die Bewegungen können i.d.R. maximale Geschwindigkeiten angegeben werden (Objekt-Masse!).
- 2.) Bewegungen können ihre Richtung von Bild zu Bild i.d.R. nur geringfügig ändern (Objekt-Masse!).
- 3.) Bewegungen betreffen i.d.R. ganze Objekte, so dass in der Bildfolge ganze Bildbereiche und nicht nur einzelne Bildpunkte verschoben sind (Objekt-Kohärenz!).
- 4.) Zuordnungen zwischen korrespondierenden Stellen sind i.d.R. konsistent, d.h. (von Verdeckungen abgesehen) entsprechen n Stellen im einen Bild auch n gleichartig angeordnete Stellen im anderen Bild.
- 5.) Häufig ist die aktuelle Bewegung a priori näherungsweise bekannt; z.B. erfolgt eine Fallbewegung i.d.R. abwärts (Weltmodell!).

Zur Ermittlung von Kandidaten für korrespondierende Bildstellen können wieder grauwert-, merkmals- oder inhaltsorientierte Ansätze verwendet werden; siehe auch Kapitel 6.3.

Da sich Ecken besonders gut zur Bestimmung von Verschiebevektoren eignen (siehe Abbildung 7.3 (b)), soll im weiteren ein spezieller **Ecken-Detektor** vorgestellt werden, der Ecken unabhängig von ihrer Orientierung (also rotationsinvariant) ermittelt.

Die Grundidee zu diesem Detektor stammt aus der Differential-Geometrie und stützt sich darauf, dass eine Ecke aus zwei zusammentreffenden Kanten besteht. In der Umgebung der Ecke gibt es also zwei ausgeprägte und in ihrer Richtung deutlich unterschiedliche Gradienten.

Um solche Bildstellen zu ermitteln, berechnet man für jede Bildposition (i,j) die horizontale partielle Ableitung g_{hori} und die vertikale partielle Ableitung g_{veri} . Daraus werden dann zu jeder Bildposition (i,j) innerhalb einer "kleinen" Umgebung (z.B. 3×3 - oder 5×5 - bis 15×15 -Umgebung) die folgenden drei Werte ermittelt:

$$\begin{aligned}
 A(i,j) &= \sum_l \sum_k g_{\text{hori}}^2(i+k, j+l), & B(i,j) &= \sum_l \sum_k g_{\text{veri}}^2(i+k, j+l) & \text{und} \\
 C(i,j) &= \sum_l \sum_k g_{\text{hori}}(i+k, j+l) * g_{\text{veri}}(i+k, j+l)
 \end{aligned}
 \tag{7-1}$$

(Dabei laufen die Doppelsummen über die Elemente der "kleinen" Umgebung der Position (i,j) ; also z.B. k und l von -1 bis $+1$ für eine 3×3 -Umgebung).

Danach werden die drei Matrizen mit den Werten von $A(i,j)$, $B(i,j)$ und $C(i,j)$ i.d.R. mit einem Gauss-Filter geglättet und mit den geglätteten Werten \bar{A} , \bar{B} und \bar{C} wird an jeder Bildposition (i,j) die sogenannte **Strukturmatrix** $M(i,j)$ berechnet:

$$M(i,j) = \begin{vmatrix} \bar{A}(i,j) & \bar{C}(i,j) \\ \bar{C}(i,j) & \bar{B}(i,j) \end{vmatrix}
 \tag{7-2}$$

Für jede Strukturmatrix, d.h. für jede Position (i,j), werden die Eigenwerte λ_1 und λ_2 bestimmt, da sich damit Aussagen über die lokale Bildstruktur an der Stelle (i,j) ableiten lassen:

$$\lambda_{1,2} = \frac{\text{spur}(M)}{2} \pm \sqrt{\left(\frac{\text{spur}(M)}{2}\right)^2 - \det(M)} = \frac{1}{2} * \left(\bar{A} + \bar{B} \pm \sqrt{\bar{A}^2 - 2\bar{A}\bar{B} + \bar{B}^2 + 4\bar{C}^2}\right) \tag{7-3}$$

Man unterscheidet folgende drei Fälle:

- 1.) λ_1 und λ_2 sind beide betragsmäßig klein: in der Nähe der Position (i,j) gibt es keine Kante und folglich auch keine Ecke.
- 2.) λ_1 und λ_2 sind in der Größe sehr verschieden: in der Nähe der Position (i,j) gibt es **eine** Kante aber keine Ecke.
- 3.) λ_1 und λ_2 sind beide deutlich von Null verschieden und betragsmäßig etwa gleich groß: in der Nähe der Position (i,j) gibt es eine Ecke. Die zu λ_1 und λ_2 gehörenden Eigenvektoren stehen senkrecht auf den beiden Kanten, die die Ecke bilden.

Die Eigenwerte kennzeichnen also die "Stärke" der Kanten in der Umgebung der entsprechenden Bildposition; die zugehörigen Eigenvektoren geben die Richtung der jeweiligen Kante an.

Aus der Tatsache, dass an einer Ecke beide Eigenwerte "groß" sein sollen, leitet Harris für den nach ihm benannten **Harris-Corner-Detector** (s. z.B. [Burger]) die sogenannte "Corner-Strength"-Funktion CSF(i,j) ab:

$$CSF(i,j) = \det(M(i,j)) - \alpha * (\text{spur}(M(i,j)))^2 = (\bar{A}\bar{B} - \bar{C}^2) - \alpha * (\bar{A} + \bar{B})^2 \tag{7-4}$$

Der Parameter α steuert die Empfindlichkeit des Harris-Corner-Detectors; er kann maximal den Wert 0.25 annehmen, und wird i.d.R. auf einen Wert zwischen 0.04 und 0.06 gesetzt.

Je größer der Wert der sogenannten "corner-strength"-Funktion CSF ist, je deutlicher zeichnet sich an der betreffenden Stelle eine Ecke ab.

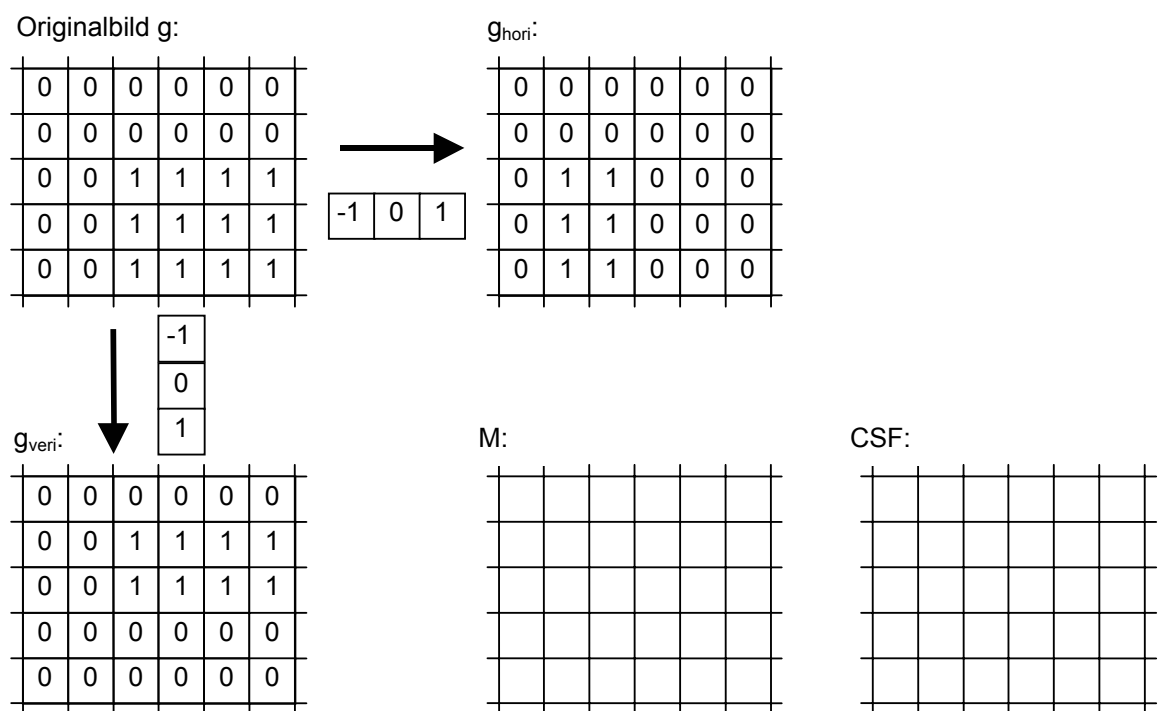


Abb. 7.4: Vorbereitetes Beispiel für den Harris-Corner-Detector.

Abschließend muss aus dicht beieinander gefundenen Eckpunkten, die ja mit hoher Wahrscheinlichkeit dieselbe Ecke im Bild bezeichnen, stellvertretend ein Eckpunkt ausgewählt werden.

Das in Abbildung 7.4 vorbereitete Beispiel demonstriert die Ecken-Detektion.

7.2 Kollisionsvorhersage

Die **Kollisionsvorhersage** spielt beispielsweise im Bereich der Robotik eine wichtige Rolle: wann bzw. nach welcher Reststrecke kollidiert ein Roboter mit einem vor ihm befindlichen Hindernis?

Im folgenden wird vorausgesetzt, dass der Roboter jeweils nach einem Schritt der Länge Δs ein Bild B_i der Szene aufnimmt, es auswertet und dabei die aktuelle Höhe h_i des Hindernisses im Bild (d.h. in Pixeln) bestimmt. Auch ohne Kenntnis der tatsächlichen Höhe h des Hindernisses und seiner ursprünglichen Entfernung zur Kamera ist es möglich, die Reststrecke s_{rest} nach Aufnahme eines (späteren) Bildes B_{i+1} bis zum Hindernis zu bestimmen (s. z.B. [Trucco, Verri]).

Abbildung 7.5 demonstriert diesen Sachverhalt, wobei hier statt des Roboters (bzw. seiner Kamera) das Hindernis bewegt wurde.

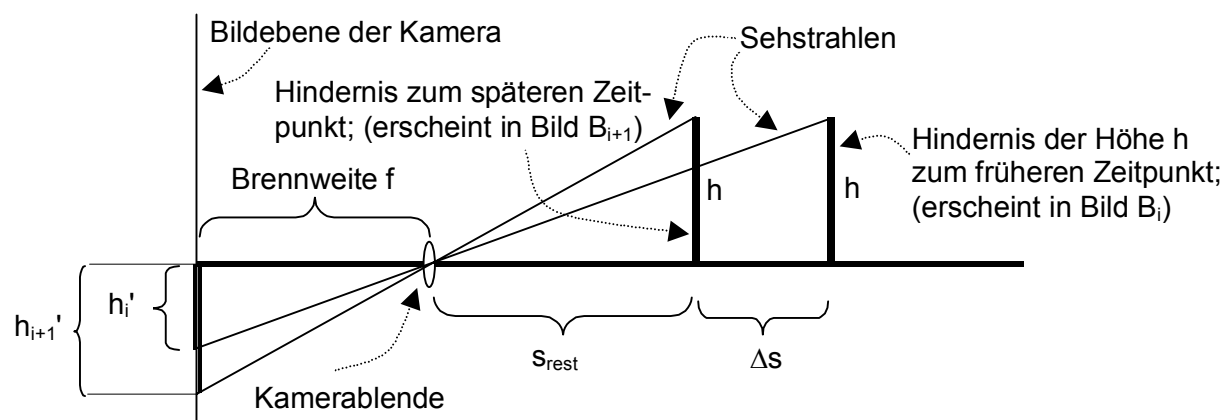


Abb. 7.5: Prinzipbild zur Kollisionsvorhersage; (Seitenansicht).

In Abbildung 7.5 gibt es zwei Paare von ähnlichen Dreiecken, und nach dem Strahlensatz erhält man:

$$1.) \frac{h}{s_{\text{rest}} + \Delta s} = \frac{h_i'}{f} \quad \text{und daraus: } h = \frac{h_i'}{f} * (s_{\text{rest}} + \Delta s)$$

$$2.) \frac{h}{s_{\text{rest}}} = \frac{h_{i+1}'}{f} \quad \text{und daraus: } h = \frac{h_{i+1}'}{f} * s_{\text{rest}}$$

Durch Gleichsetzen ergibt sich:

$$\frac{h_i'}{f} * (s_{\text{rest}} + \Delta s) = \frac{h_{i+1}'}{f} * s_{\text{rest}}$$

und daraus:

$$s_{\text{rest}} = \frac{h_i'}{h_{i+1}' - h_i'} * \Delta s.$$

(7-5)

Da die auf der rechten Seite auftretenden Größen alle bekannt sind, kann die Reststrecke s_{rest} bis zur Kollision auf ganz einfache Weise berechnet werden.

Auf dieselbe Weise, bzw. mit Hilfe der Beziehung $s = v * t$ (Strecke = Geschwindigkeit * Zeit) kann man ermitteln, welche Restzeit t_{rest} noch bis zur Kollision zwischen einem Roboter und einem Hindernis verbleibt (time to impact). Als einzige Voraussetzung muss hier das Zeitintervall Δt zwischen zwei aufeinanderfolgenden Bildern bekannt sein:

(7-6)

$$t_{\text{rest}} = \frac{h_i'}{h_{i+1}' - h_i'} * \Delta t.$$

7.3 Differentieller Ansatz zur Bildfolgenauswertung

Beim **differentiellen Ansatz** nach Horn und Schunk (s. z.B. [Jähne]) wird die Bewegung als stetige Änderung der Bildfunktion g hinsichtlich der Position und der Zeit modelliert. Das bedeutet, dass das Helligkeitsmuster, das sich zum Zeitpunkt $t'+1$ an der Stelle (x,y) befindet, zum früheren Zeitpunkt t' an der um den Verschiebevektor $[\delta x, \delta y]^t$ verschobenen Position $[x+\delta x, y+\delta y]^t$ vorhanden war:

$$g(x, y, t'+1) = g(x+\delta x, y+\delta y, t') \quad (7-7)$$

Das Ziel ist, den unbekanntem Verschiebevektor $[\delta x, \delta y]^t$ zu ermitteln. Abbildung 7.6 demonstriert diesen Fall an einem eindimensionalen Beispiel. Die Bestimmung des gesuchten Verschiebevektors δx ist hier eigentlich trivial, jedoch soll im folgenden die Grundidee der späteren zweidimensionalen Lösung vorgestellt werden.

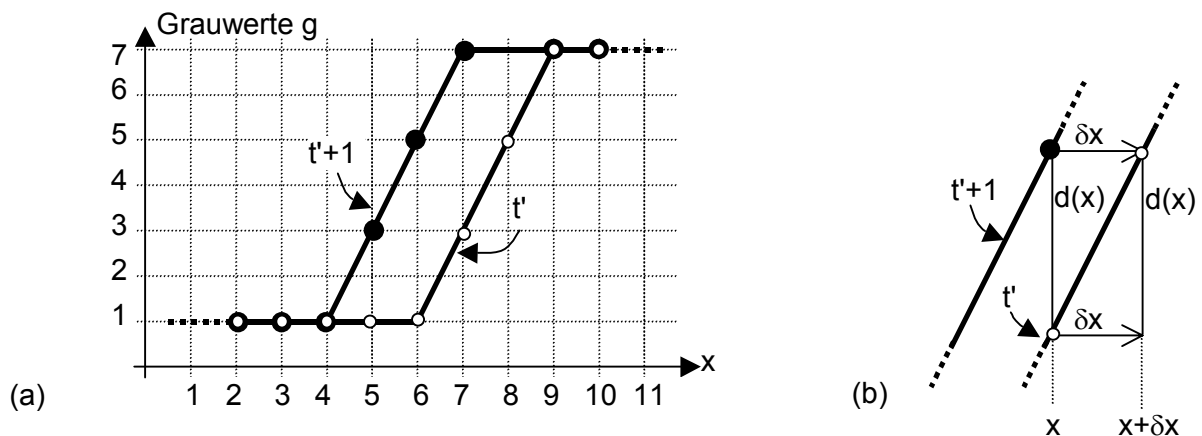


Abb. 7.6: Beispiel (eindimensional) zur differentiellen Betrachtungsweise.

In Abbildung 7.6 (b) ist die Steigung der Funktion g zum Zeitpunkt t' an der Stelle x gegeben durch

$$g'(x, t') = \frac{d(x)}{\delta x}.$$

Dabei gibt $d(x)$ die Grauwertdifferenz zwischen den um δx verschobenen Grauwertfunktionen an der Stelle x an: $d(x) = g(x, t'+1) - g(x, t')$.

Der gesuchte Verschiebevektor δx ergibt sich also zu:

$$\delta(x) = \frac{d(x)}{g'(x, t')}. \quad (7-8)$$

Um im zweidimensionalen Fall die Verschiebung δx und δy in einem Punkt zu bestimmen, bildet man entsprechend zu oben die Grauwert-Differenzfunktion d :

$$\begin{aligned} d(x, y) &= g(x, y, t'+1) - g(x, y, t') \\ &= g(x+\delta x, y+\delta y, t') - g(x, y, t') \quad \text{wegen (7-7)}. \end{aligned}$$

Mit Hilfe der Taylor-Reihenentwicklung ergibt sich:

$$d(x, y) = g(x, y, t') + g_{\text{hori}} * \delta x + g_{\text{veri}} * \delta y + E - g(x, y, t') = g_{\text{hori}} * \delta x + g_{\text{veri}} * \delta y + E.$$

Dabei bezeichnen g_{hori} und g_{veri} die partiellen Ableitungen in horizontaler bzw. vertikaler Richtung (vgl. Kapitel 4.1) und E fasst die Glieder höherer Ordnung der Taylor-Reihenent-

wicklung zusammen, die im weiteren Verlauf vernachlässigt, d.h. weggelassen werden. $d(x, y)$ lautet dann in Matrix-Schreibweise:

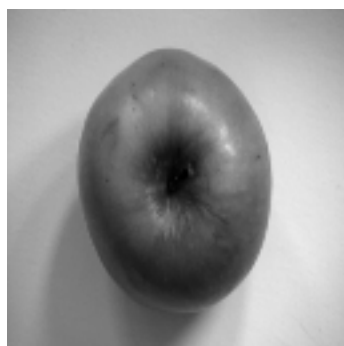
$$d(x, y) = [g_{\text{hori}}, g_{\text{veri}}]^t [\delta x, \delta y]^t. \quad (7-9)$$

(Auf den eindimensionalen Fall reduziert entspricht dies genau der Gleichung (7-8)).

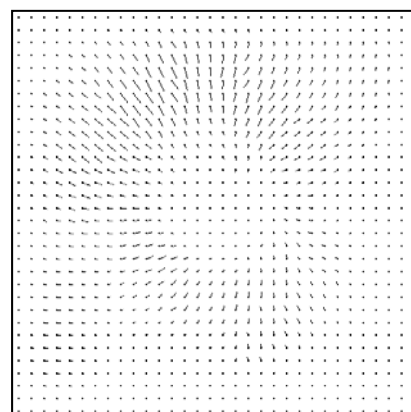
Die partiellen Ableitungen sowie die Differenzfunktion lassen sich für alle Bildpunkte direkt aus der Bildfolge berechnen, aber diese eine Gleichung reicht nicht aus, um die beiden Unbekannten δx und δy zu berechnen. Um den Verschiebevektor $[\delta x, \delta y]^t$ dennoch zu bestimmen, verwendet man das sogenannte **Smoothness-Constraint**. Dieses setzt voraus, dass sich die einzelnen Bildpunkte von Bild zu Bild nicht willkürlich bewegen, sondern dass benachbarte Bildpunkte ähnliche Bewegungen durchführen, d.h. dass "glatte" Verschiebevektorfelder vorliegen. Die Aufgabe lässt sich (mathematisch aufwändig) als Fehlerminimierungsproblem formulieren, das durch einen iterativen Ansatz im Rahmen eines Relaxationsverfahrens gelöst werden kann.

Abbildung 7.7 zeigt ein reales Beispiel zur Bestimmung eines Verschiebevektorfeldes. In Abbildung 7.7 (a) ist eines von zwei Bildern einer Bildfolge zu sehen. Das zweite Bild, in dem sich die Kamera vom Objekt entfernt hat, ist nicht dargestellt. Abbildung 7.7 (b) zeigt das resultierende Verschiebevektorfeld und in den Abbildungen (c) bis (e) sind als Zwischenergebnisse die Ableitungen g_{hori} , g_{veri} sowie das Differenzbild d dargestellt.

Ein Problem bei diesem Vorgehen, das auch in Abbildung 7.7 sichtbar wird, stellen die Konturen sich bewegender Objekte dar. Die an diesen Konturen bevorzugt gefundenen Verschiebevektoren beeinflussen bei der Relaxation ihre Umgebung zu beiden Seiten der Kontur, so dass auch Verschiebungen auf der Seite der Kontur entstehen, die zum ruhenden Hintergrund gehört. Dieser Effekt muss in der Praxis entsprechend berücksichtigt und verhindert werden.



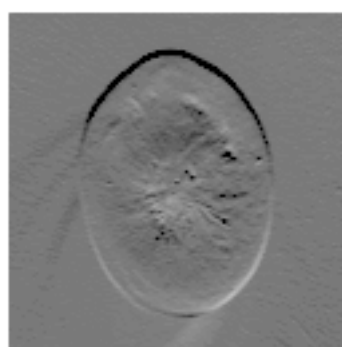
(a)



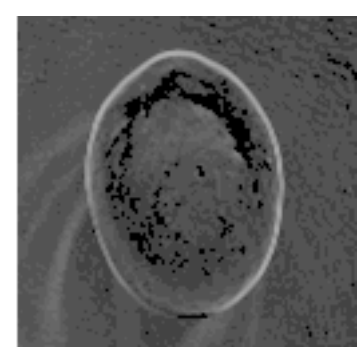
(b)



(c)



(d)



(e)

Abb. 7.7: Beispiel für die Bestimmung eines Verschiebevektorfeldes; (Verschiebevektorfeld erzeugt mit "Bildverarbeitung Ad Oculos" von H. Bässmann und J. Kreys).

8 Shape from X

Im Prinzip ist es unmöglich, dreidimensionale Information aus **einem** monokularen Bild abzuleiten; siehe Kapitel 5, Abbildung 5.1. Unter bestimmten Zusatzvoraussetzungen (z.B. Wissen über die abgebildete Szene) können jedoch auch aus einem oder mehreren vom gleichen Kamerastandort aus aufgenommenen Bildern Oberflächenorientierungen abgeleitet werden. Daraus kann dann die Form der Objekte bestimmt werden. In Abhängigkeit von den verfügbaren Zusatzvoraussetzungen (Zusatzinformationen) werden diese Methoden bezeichnet als **Form aus Schattierung (shape from shading)**, **Form aus Form (shape from shape)** oder **Form aus Textur (shape from texture)**. Die Darstellung der ermittelten Oberflächenorientierungen erfolgt häufig in Form sogenannter **Nadel-Diagramme**; dabei sind nur die Oberflächen-Einheitsnormalen dargestellt (siehe Abbildung 8.1).

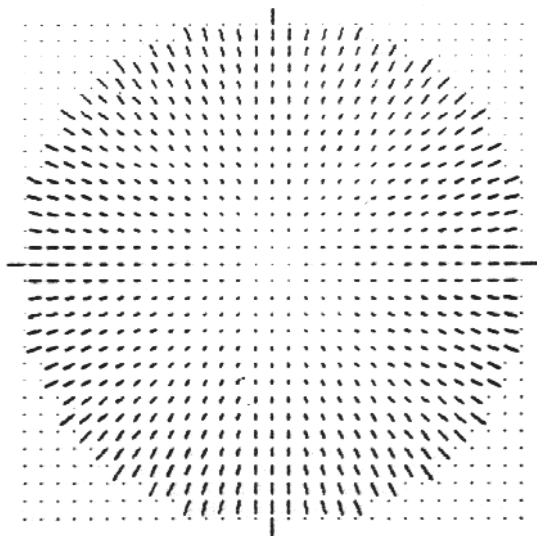


Abb. 8.1: Beispiel für ein Nadeldiagramm einer Kugel.

8.1 Form aus Schattierung

Bei der "normalen" Bildsegmentierung erschweren die Helligkeitsänderungen im Bild, die auf gleichmäßige Krümmungen von Oberflächen zurückzuführen sind, den Segmentierungsprozess. Andererseits erlauben diese Änderungen aber die Ableitung von Hinweisen auf die 3D-Gestalt der Oberflächen.

Die Helligkeit eines Bildpunktes in einem Grauwertbild hängt ab von der Beleuchtung, dem Reflektionsverhalten, der Oberflächenneigung sowie dem Abstand des Oberflächenelementes, das in dem betreffenden Bildpunkt abgebildet wird. Wenn die Beleuchtung und das Reflektionsverhalten der Objekte einer Szene bekannt sind, lässt sich daraus die Oberflächenneigung für alle Bildpunkte bestimmen.

Sei im weiteren O ein Oberflächenelement an der Stelle (x,y,z) mit einer Oberflächennormalen \vec{n} . Die Beleuchtung erfolge durch eine Lichtquelle aus der Richtung \vec{s} und die Kamera befinde sich in der Richtung \vec{v} und schaue entlang der z -Achse; (\vec{n} , \vec{s} und \vec{v} sind Vektoren, siehe Abbildung 8.2). Mit $a(x,y,z)$ sei die Intensität des einfallenden Lichtes an der Stelle (x,y,z) bezeichnet, und $\Phi(\vec{n}, \vec{s}, \vec{v})$ stehe für die Reflektion. Weiter sei $I'(x',y')$ die Beleuchtungs-Intensität (Helligkeit) im Bildpunkt (x',y') , der dem Oberflächenelement O entsprechen soll. $I'(x',y')$ lässt sich darstellen als:

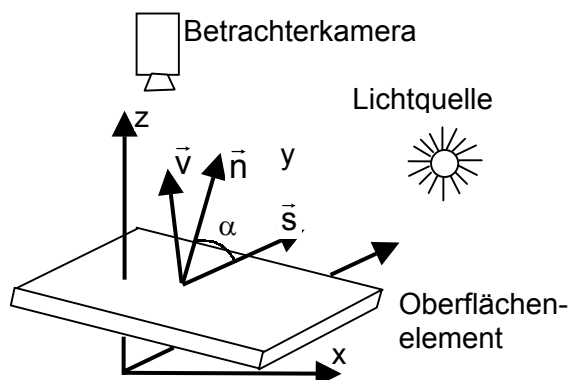
$$I'(x',y') = k * a(x,y,z) * \Phi(\vec{n}, \vec{s}, \vec{v}) \quad (8-1)$$

wobei k eine Konstante ist, die durch die Charakteristik der Kamera festgelegt ist. Die Gleichung (8-1) ist **nicht** linear und kann **nicht** analytisch gelöst werden.

Der Einfachheit halber sei die Lichtquelle nun sehr weit entfernt, damit $a(x,y,z)$ und \vec{s} für jede Position der Szene konstant sind, d.h. $a(x,y,z)=a$. Gleichung (8-1) vereinfacht sich so zu:

(8-2)

$$I'(x',y') = k * a * \Phi(n, s, v).$$

**Abb. 8.2:** Geometrie der Reflektion.

Nun sollen vereinfachende Annahmen für die Reflektionsfunktion $\Phi(\vec{n}, \vec{s}, \vec{v})$ getroffen werden. Eines der einfachsten Reflektionsmodelle ist das einer perfekt diffus rückstrahlenden Oberfläche (**Lambert'scher Strahler**, siehe Graphische Datenverarbeitung). Die Reflektion ist hier unabhängig von der Betrachter-Richtung \vec{v} und proportional zum Cosinus des Lichteinfallswinkels α zwischen \vec{n} und \vec{s} :

$$\Phi(\vec{n}, \vec{s}, \vec{v}) = r \cdot \cos \alpha.$$

(8-3)

Dabei ist r der Reflektions-Koeffizient der Oberfläche. Da \vec{s} als konstant vorausgesetzt wird, hängt der Winkel α nur von der Flächennormalen \vec{n} ab. Da die Orientierung einer Fläche O (also die Richtung \vec{n} ihrer Normalen) durch die beiden partiellen Ableitungen $p = \partial O / \partial x$ und $q = \partial O / \partial y$ beschrieben werden kann, ist die Reflektion also eine Funktion von p und q :

$$\Phi(\vec{n}, \vec{s}, \vec{v}) = R(p, q)$$

(8-4)

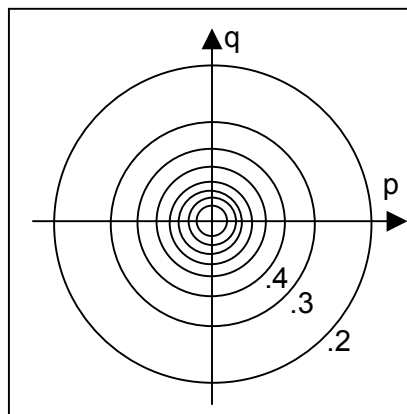
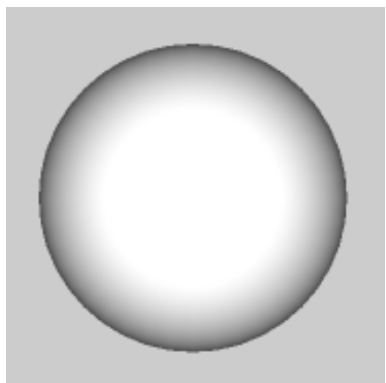
Da $R(p, q)$ die Bildhelligkeit als Funktion von p und q angibt, wird $R(p, q)$ als **Reflektionskarte (reflection map)** bezeichnet. Üblicherweise wird $R(p, q)$ als (Bild-)Matrix mit einer Folge von **Isolinien** gleicher Helligkeit dargestellt; siehe Abbildung 8.3. (Anschaulich entspricht $R(p, q)$ einer Höhenlinien-Karte, nur dass die Höhenlinien nicht gleiche Höhen sondern gleiche Helligkeiten im p - q -Raum bezeichnen). Jede Isolinie gibt also Kombinationen von p und q (d.h. Neigungen des Oberflächenelementes O bzgl. der Einstrahlrichtung \vec{s}) an, bei denen das Oberflächenelement unter der gleichen, dieser Isolinie zugeordneten Helligkeit erscheint.

Wenn die Betrachtung der durch p und q beschriebenen Oberflächenelemente aus der Richtung der Lichtquelle erfolgt (d.h. $\vec{s} = \vec{v}$ gilt), so bilden die Isolinien konzentrische Kreise um den Nullpunkt der p - q -Ebene (siehe Abbildung 8.3 (a)). Erfolgt die Betrachtung nicht aus der Beleuchtungsrichtung, so bilden die Isolinien Ellipsen um den maximalen Wert 1 (maximale Helligkeit), die nach außen mit abfallenden Reflektionswerten von Ellipsen über Parabeln in Hyperbeln übergehen (siehe Abbildung 8.3 (b) und (c)).

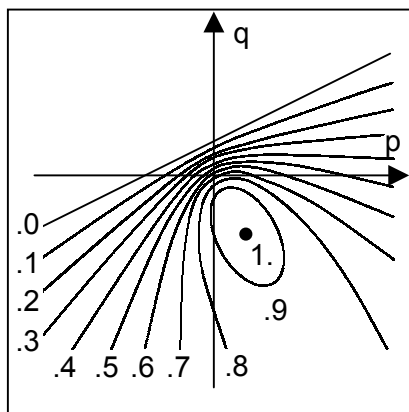
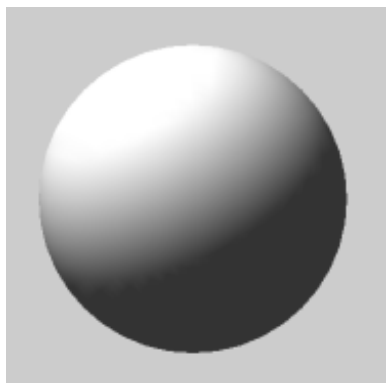
In der Praxis können die Reflektionskarten für unterschiedliche Materialien, d.h. unterschiedliche Reflektions-Koeffizienten r experimentell bestimmt, bzw. berechnet werden. Bei bekanntem Reflektions-Koeffizienten r lässt sich aus der Bildhelligkeit und dem bekannten Winkel α also zu jedem Bildpunkt die entsprechende Isolinie im p - q -Raum angeben. Die Oberflächenneigung eines Oberflächenelementes ist damit noch nicht eindeutig festgelegt, aber es kommen nur noch bestimmte Konstellationen von p und q in Frage, d.h. die Aufgabe ist auf einen Freiheitsgrad reduziert (unbekannter Punkt auf der ausgewählten Isolinie).

Die Hinzunahme weiterer Bilder, die vom gleichen Kamerastandort aus, aber bei unterschiedlicher Beleuchtungsrichtung aufgenommen wurden, macht die Bestimmung der Oberflächenorientierung eindeutig. Diese Methode wird als **photometrisches Stereo** bezeichnet.

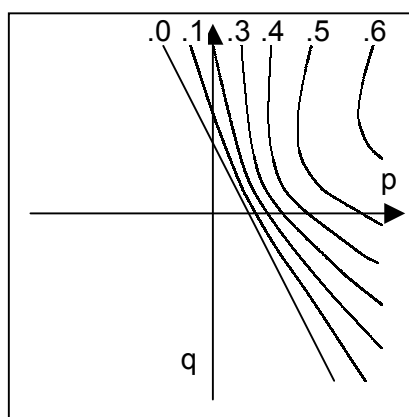
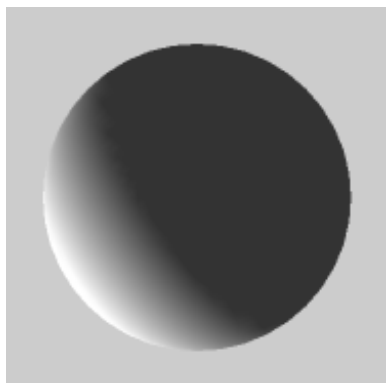
(a)



(b)



(c)



(d)

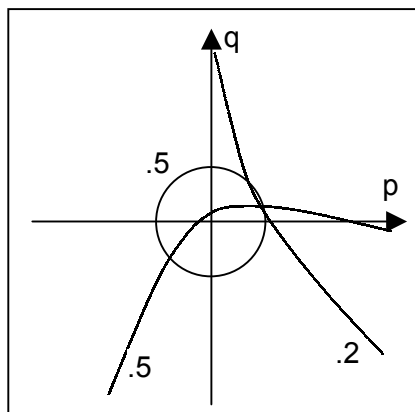


Abb. 8.3: Beispiel für die Erstellung und Auswertung von Reflektionskarten:

- (a) Reflektionskarte bei Beleuchtung von vorne,
- (b) Reflektionskarte bei Beleuchtung von links oben,
- (c) Reflektionskarte bei Beleuchtung von links unten und leicht von hinten,
- (d) Überlagerung von drei Isolinien aus (a), (b) und (c) führt zur eindeutigen Bestimmung der Oberflächenorientierung eines Oberflächenelementes.

Zwei derartige Bilder liefern in jedem Bildpunkt zwei Helligkeiten, also zwei Isolinien. Diese schneiden sich im p-q-Raum in zwei Punkten, so dass die Orientierung eines Oberflächenelementes nun bis auf eine Doppeldeutigkeit bestimmt ist. Durch die Hinzunahme eines dritten photometrischen Stereobildes ist die Oberflächenorientierung eindeutig ermittelt; siehe Abbildung 8.3 (d). (Voraussetzung ist dabei, dass die Oberflächenelemente für keine der drei Lichtquellen im abgeschatteten Bereich liegen).

(Im Gegensatz zur üblichen Stereobild-Auswertung tritt hier kein Korrespondenzproblem auf, da die Bilder bildpunktgenau "zueinander passen".)

In Abbildung 8.4 ist ein Beispiel zur Berechnung der Oberflächenorientierung mit Hilfe von gerasterten Reflektionskarten vorbereitet.

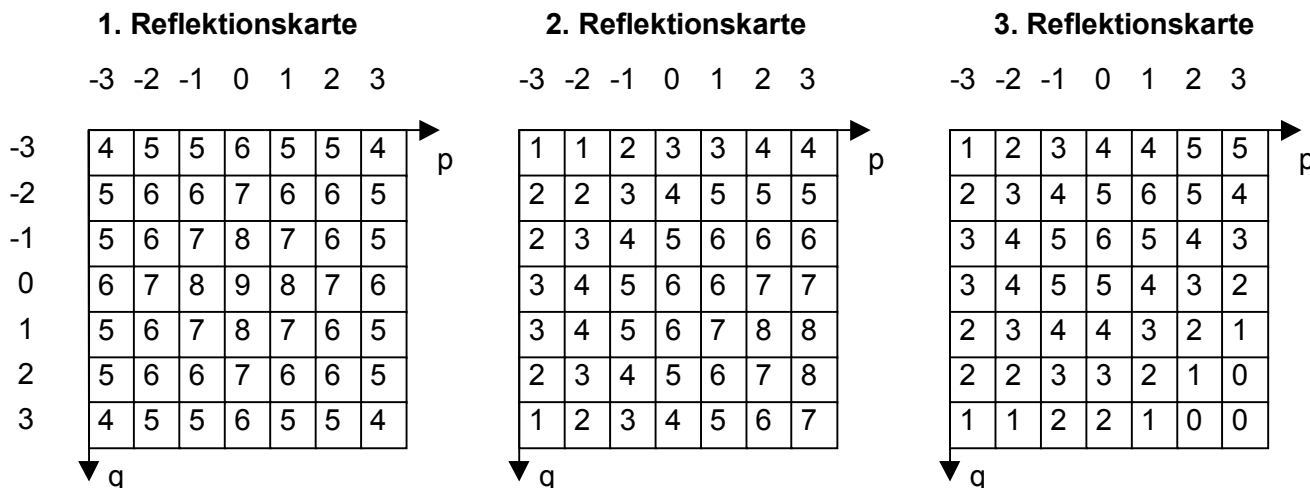


Abb. 8.4: Vorbereitetes Beispiel für die Ermittlung der Oberflächenorientierung: die drei Reflektionskarten zeigen die Isolinien bzgl. einer Halbkugel bei drei bestimmten Lampenpositionen und einem speziellen Material. Das auszuwertende Oberflächenelement habe in den zugehörigen drei Bildern die Grauwerte: $g_1 = \dots$, $g_2 = \dots$ und $g_3 = \dots$.

Für ein anderes einfaches Reflektionsmodell ist die Berechnung der Oberflächenorientierung in einigen, ausgezeichneten Punkten sehr einfach. Für eine ideal spiegelnde Oberfläche gilt, dass bzgl. der Flächennormalen der Einfallswinkel und der Ausfallswinkel des Lichtes gleich sind. An Stellen, an denen im Bild Schlaglichter auftreten, d.h. der Lichtausfall auf die Kamera trifft, ist die Flächennormale also direkt bestimmt. In der Realität treten diffuse und spiegelnde Reflektion i.d.R. gemischt auf, und die Berechnung der Reflektions-Karte gestaltet sich erheblich komplizierter.

8.2 Form aus Konturen

Der Mensch ist in der Lage, aus einfachen Strichzeichnungen einer dreidimensionalen Szene, wie sie z.B. bei einer perfekten Bildsegmentierung entstehen, die Oberflächenorientierung abzuleiten; siehe Abbildung 8.5 (a). Im wesentlichen gibt es in solchen Strichzeichnungen zwei Arten von Konturlinien: sogenannte **äußere Kanten** (eine betrachtete Oberfläche verschwindet durch stetige Krümmung; siehe Abbildung 8.5 (b) seitlich am Eimer) und sogenannte **Abbruch-Kanten** (eine ansonsten glatte Oberfläche weist einen Knick auf oder endet; siehe Abbildung 8.5 (b) oberer Eimerrand).

An einer äußeren Kante kann die Oberflächenorientierung exakt angegeben werden: die Flächennormale steht einerseits senkrecht auf der äußeren Kante im Bild und andererseits senkrecht auf der Blickrichtung.

Für Abbruch-Kanten lässt sich die Oberflächenorientierung nur auf einen Freiheitsgrad einschränken: die Flächennormale steht senkrecht auf der Abbruch-Kante im Bild; dies entspricht der unbekanntem Lage eines Vektors in einer Ebenen des 3D-Raumes.

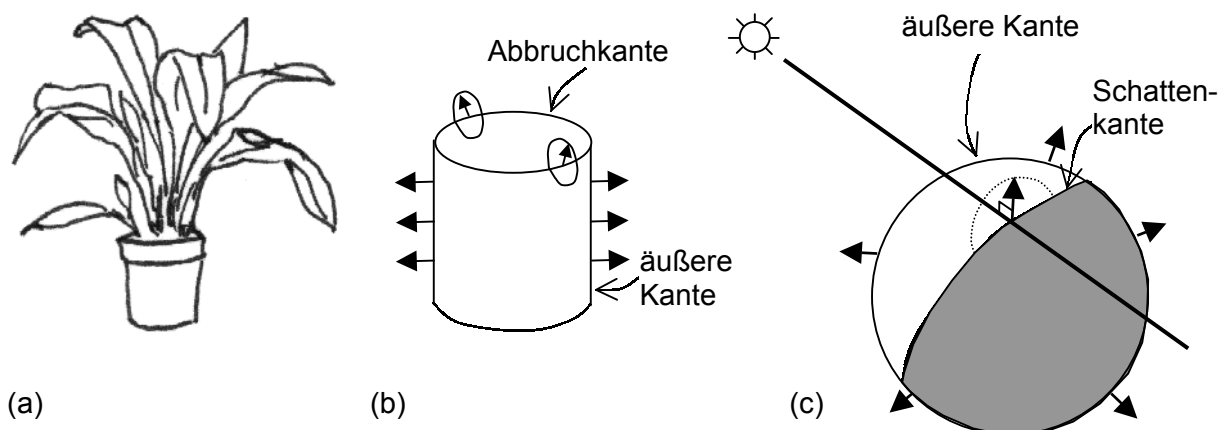


Abb. 8.5: Beispiele für die Bestimmung der Oberflächenorientierungen in Strichzeichnungen.

Durch den räumlichen Zusammenhang der Objekte (Kohärenz) und unter der Voraussetzung, dass Objektflächen stetig gekrümmt sind, lässt sich durch Interpolation aus den an den Kanten ermittelten Oberflächenorientierungen die Orientierung in großen Teilen des Bildes ableiten. Auch hier kommen insbesondere wieder Relaxationsverfahren zum Einsatz.

Eine weitere Möglichkeit zur Einschränkung von Oberflächenorientierungen ist an den sogenannten **Schatten-Kanten**, d.h. den durch Schattenwurf verursachten "Scheinkanten" möglich (siehe Abbildung 8.5 (c)). Entlang der Schatten-Kanten steht die Oberflächennormale senkrecht auf der Beleuchtungsrichtung, d.h. es bleibt nur noch ein Freiheitsgrad (Ebene) offen.

Ausgehend von Stellen, an denen die Schatten-Kanten auf die äußeren Kanten treffen, lässt sich die Oberflächenorientierung entlang der Schatten-Kanten durch Interpolation meist eindeutig bestimmen.

8.3 Form aus Textur

Die Texturierung von Oberflächen liefert sehr starke Hinweise zur Ableitung der Oberflächenorientierung. Bei in der Realität einheitlich texturierten Oberflächen unterliegen die texturbildenden Primitive im Bild häufig Variationen hinsichtlich Größe, Orientierung, Aspekt-Verhältnis und Dichte der Anordnung (**Texturkompression**). Diese Variationen sind im wesentlichen auf zwei Ursachen zurückzuführen:

- erstens wirkt eine entfernungsabhängige Skalierung, und
- zweitens ergibt sich bei der perspektivischen Abbildung eine von Änderungen der Oberflächenneigung abhängige Verkürzung.

Alle Ausdehnungsrichtungen eines Texturprimitivs im Bild werden in Abhängigkeit von der Entfernung zur Kamera im gleichen Maßstab verkleinert. Im Gegensatz dazu wird nur in Richtung der Oberflächenneigung eine Verkürzung vorgenommen. Diese Ausdehnungsrichtung wird um einen Faktor verkürzt, der proportional ist zum Cosinus des Winkels zwischen der Oberflächennormale und der Blickrichtung; siehe Abbildung 8.6.

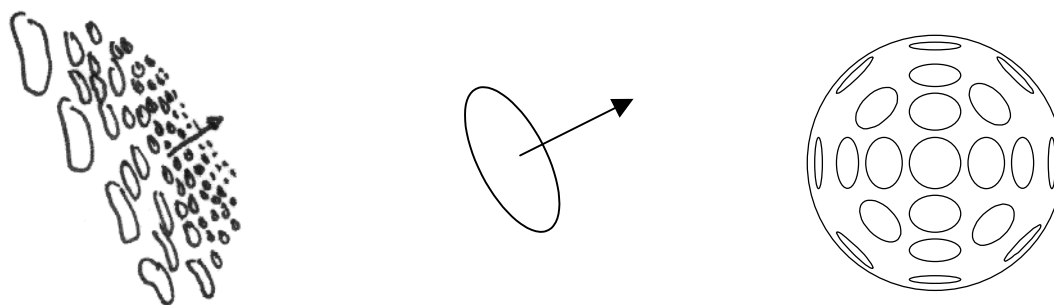


Abb. 8.6: Beispiele für die Bestimmung der Oberflächenorientierung aus einer Textur.

9 Wissensbasierte Bildinterpretation

Die vorangegangenen Kapitel zeigten die Lösung spezieller Aufgabenstellungen und die Ableitung folgender innerer Charakteristika (intrinsic characteristics): Oberflächenorientierung und -distanz. Eine vollständige Bildbeschreibung können jedoch (wenn überhaupt) nur komplexe Bildanalyse-Systeme ableiten; (diese Systeme verfügen häufig über die aufgeführten Verfahren). Eine solche Bildbeschreibung soll folgendes umfassen:

- die Bestandteile (Objekte) der Szene,
- die Eigenschaften der Objekte,
- die Beziehungen zwischen den Objekten und
- die Interpretation der Objekte.

Für eine Bildanalyse sind komplexe Verarbeitungsstrukturen erforderlich, die durch ein wissens- und modellgesteuertes Vorgehen die Gesamtaufgabe in Form organisierter Teilaufgaben lösen. Das Vorgehen wird deshalb auch als **wissens-** oder **modellgesteuerte Bildanalyse** oder als **Bildverstehen** bezeichnet. Der Terminus Bildverstehen rührt dabei daher, dass die Ableitung der Beschreibung eines nicht-trivialen Bildes meist bis zu einem gewissen Grade ein "Verstehen" des Bildinhaltes voraussetzt. Das bedeutet, dass bestimmten Bestandteilen eines Bildes bzw. einer Bildfolge eine Bedeutung aus dem zugrunde liegenden Problemkreis zugeordnet wird. (Im Falle der Analyse elektrischer Schaltpläne kann man z.B. einer bestimmten räumlichen Anordnung von Strichen die Bedeutung "Widerstand" zuordnen).

Bildverstehen heißt also, dass ein Zusammenhang hergestellt wird zwischen Größen, die mehr oder weniger direkt aus dem Bild ermittelt werden können und Objekten, Ereignissen usw. aus dem zugrunde liegenden Problemkreis. Kenntnisse über die Menge dieser Objekte und Ereignisse, ihre gegenseitigen Relationen örtlicher, zeitlicher und anderer Natur sowie ihre Erscheinungsform in den Bildern werden im allgemeinen als **Wissen** über den betrachteten Problemkreis bezeichnet. Von wissensgesteuerter Bildanalyse (Bildverstehen) spricht man dann, wenn dieses Wissen **explizit** als separater Modul vorhanden ist und deutlich vom Rest des Systems getrennt ist. Ein solcher Modul wird auch als **Wissensbasis** oder **Modell** (des Problemkreises) für die Bildanalyse bezeichnet.

Abbildung 9.1 zeigt eine grundlegende Struktur für eine wissensgesteuerte Bildanalyse. Es gibt eine klare Trennung zwischen den eher numerischen Methoden zur Extraktion elementarer Bildbestandteile, der Wissensbasis und einem Modul, dem die Aufgabe der Wissensnutzung zukommt. Das bedeutet, dass dieser Modul mit eher symbolischen Methoden unter Rückgriff auf das vorhandene Wissen entscheidet, wann welche Verarbeitungsschritte ausgeführt werden sollen (**Ablaufkontrolle**).

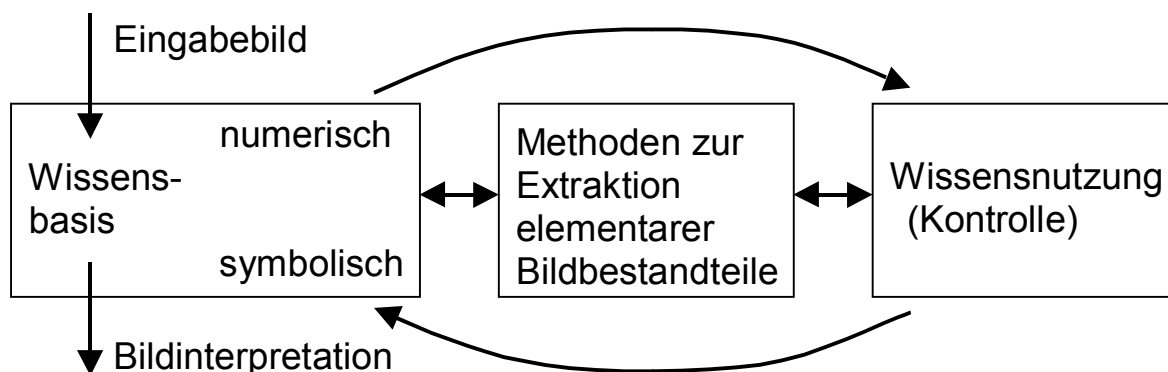


Abb. 9.1: Grundlegende Struktur eines wissensgesteuerten Bildanalyse-Systems.

Der Vorteil eines Systemaufbaus nach Abbildung 9.1 liegt klar auf der Hand: die Modularität des Systems gestattet es, einzelne Module auszutauschen, ohne dass dies für den Rest des Systems möglicherweise fatale Auswirkungen hat. Damit können Fehlerkorrekturen und Systemverbesserungen schneller und sicherer durchgeführt werden. Auch die Umstellung auf eine komplett neue Aufgabe, wie dies z.B. im Bereich der Fertigungs-Automatisierung häufig

auftritt, lässt sich schnell erreichen. Die Trennung von Wissensbasis, Wissensnutzung und Methoden zur Extraktion elementarer Bestandteile trägt weiterhin zur Transparenz des Systems bei. Dies gilt sowohl für die Sicht des Benutzers, als auch die des Entwicklers. Bei früheren Ansätzen lag häufig eine untrennbare Vermischung insbesondere von Wissensbasis und Kontrolle vor (**implizites Wissen**), was der Verständlichkeit, Änderbarkeit und Erweiterbarkeit des Systems starke Grenzen setzte.

Während Methoden zur Extraktion elementarer Bestandteile bereits ausgiebig diskutiert wurden, soll im folgenden auf die Wissensbasis und die Wissensnutzung eingegangen werden. Diese beiden Themen sind Kernthemen der **Künstlichen Intelligenz** und werden dort ausführlich behandelt; hier kann nur ein kurzer und vereinfachter Einblick aus der Sicht der automatischen Bildauswertung gegeben werden. Zuvor soll kurz auf den Gesamtaufbau einer wissensgesteuerten Bildverarbeitung eingegangen werden.

9.1 Komponenten einer wissensgesteuerten Bildanalyse

Als Vorgabe für eine wissensgesteuerte Bildanalyse werden benötigt:

- das zu analysierende Eingabebild,
- die Methoden zur Bildverarbeitung und Bedeutungszuweisung,
- die Wissensbasis, die die für die Analyse des Bildes relevanten Wissensinhalte enthält,
- die Systemsteuerung, die unter Rückgriff auf die Wissensbasis zur rechten Zeit die richtigen Systemaktionen veranlasst, und
- die Spezifikation des Zieles der Bildanalyse (Aufgabenstellung).

Bei realen Bilddaten kann das Ergebnis nur in zahlreichen Verarbeitungsschritten aus den Eingabedaten abgeleitet werden. Der Interpretationsprozess wird in der Literatur häufig in drei Stufen eingeteilt; jeder dieser Stufen ist eine bestimmte Form der Repräsentation der aktuellen Zwischenergebnisse zugeordnet.

- 1.) Low-Level-Verarbeitung: Auf dieser Stufe werden die Bilddaten einfachen Bildbe- und Bildverarbeitungsoperationen unterzogen; z.B. Bildverbesserung zur Hervorhebung relevanter Information. Diese Verarbeitung, die Bilddaten wieder in Bilddaten überführt, wird häufig auch als **ikonische Verarbeitung** bezeichnet.
- 2.) Medium-Level-Verarbeitung: Auf dieser Zwischenstufe wird aus den ikonischen Bilddaten eine erste Beschreibung z.B. in Form von Bildregionen mit symbolischen Namen (z.B. Region R_i) sowie zugehörigen Attributen (z.B. Größe der Region R_i) und Relationen (z.B. R_i benachbart zu R_j) erzeugt. Die Ergebnisse, die auch als erste **symbolische Beschreibung** bezeichnet werden, liegen i.d.R. nicht mehr bildhaft, sondern z.B. in Form von Tabellen oder Graphen vor.
- 3.) High-Level-Verarbeitung: Auf dieser höchsten Verarbeitungsstufe erfolgt die **Interpretation** der ersten symbolischen Beschreibung. Diese Stufe wird im Gegensatz zur ikonischen Verarbeitung als **symbolische Verarbeitung** bezeichnet. Im Rahmen der Interpretation werden die symbolischen Namen und Bezeichnungen durch die tatsächlichen realen Bedeutungen (z.B. Objektnamen wie "Haus") ersetzt. Das Ergebnis muss die bzgl. der jeweiligen Aufgabenstellung geforderte (mehr oder weniger detaillierte, allgemein verständliche oder maschinenlesbare) Beschreibung besitzen. Z.B. muss es im Rahmen der automatischen Fertigung direkt zur Durchführung der jeweils nächsten Fertigungsschritte (z.B. Verschweißen zweier Teile) genutzt werden können.

Abbildung 9.2 zeigt ein Beispiel mit Ergebnissen der verschiedenen Verarbeitungs- und Beschreibungsebenen. Das reale Eingabebild, das ein Montageblech auf einem Fließband darstellt, ist hier nicht wiedergegeben. Aus diesen ikonischen Daten wird — z.B. nach einer Grauwertspreizung — durch ein flächenorientiertes Segmentierungsverfahren das in Abbildung 9.2 (a) gezeigte Zwischenergebnis erzeugt, das die in Abbildung 9.2 (b) dargestellte symbolische Beschreibung besitzt. Diese symbolische Beschreibung wird durch die symbolischen Interpretationsverfahren weiterverarbeitet, bis schließlich das gewünschte Endergebnis

vorliegt. Dieses wird hier z.B. in Form eines attributierten relationalen Graphen dargestellt; siehe Abbildung 9.2 (c).

Die Gesamtaufgabe ist dadurch charakterisiert, dass während der Auswertung ständig entschieden werden muss, welcher Auswerteschritt (bzw. welche alternativen Auswerteschritte parallel) als nächstes ausgeführt werden soll(en). Die Entscheidungen sind i.d.R. von Mehrdeutigkeiten geprägt, da die Ergebnisse, die bei der Extraktion elementarer Bildbestandteile gewonnen werden, oft bis zu einem gewissen Grade unvollständig, inkonsistent oder sogar unkorrekt sind. Diese Entscheidungen werden auf der Basis des verfügbaren Wissens im Rahmen der Wissensnutzung (Kontrolle) getroffen.

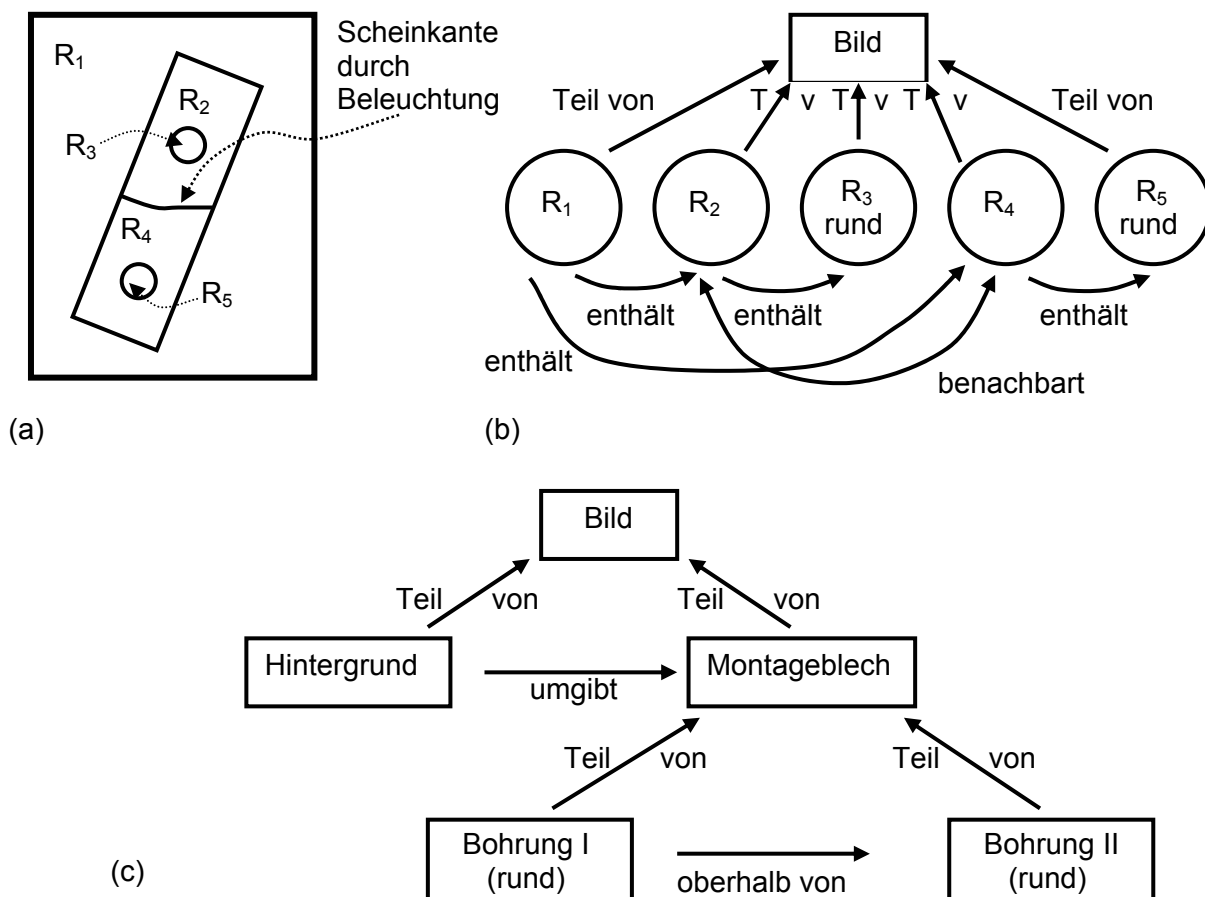


Abb. 9.2: Beispiel für die verschiedenen Verarbeitungs- und Beschreibungsebenen.

Nur sehr einfache Auswerteaufgaben lassen sich ohne eine solche flexible Ablaufsteuerung im Rahmen eines streng hierarchischen und vorgegebenen Verfahrensablaufes lösen: erst Verfahren A, dann Verfahren B oder C, anschließend D usw.. Üblicherweise sind wirklich flexible Steuerungen notwendig, wie sie in Kapitel 9.3 beschrieben werden.

9.2 Wissensbasis: Wissensinhalte und Wissensrepräsentation

"Representation seems to be the key to creating computer intelligence" (Winston)

Bei der Wissensbasis unterscheidet man zwischen Langzeitwissen und Kurzzeitwissen. Beim **Langzeitwissen** handelt es sich um allgemeingültiges Wissen, das generell auf den betrachteten Problembereich zutrifft und unabhängig vom betrachteten Bild ist. Im Gegensatz dazu bezieht sich das **Kurzzeitwissen** auf die in jedem Auswerteschritt vorliegenden Eingabedaten. Langzeitwissen ist also statischer Natur, während Kurzzeitwissen dynamische Züge trägt und sich von Bild zu Bild, aber oft auch von Bildteil zu Bildteil ändert. (Zwischen-) Ergebnisse sind in diesem Sinne auch dem Wissen zuzurechnen; sie zählen zum Kurzzeitwissen.

Das Wissen liegt häufig in Form sogenannter Modelle vor, weshalb man oft auch von modellgesteuerter Bildanalyse spricht. Die Modelle werden eingesetzt, um den Abbildungsprozess des Sensors zu invertieren. Das bedeutet, dass die Bildanalyse ein aktiver Prozess ist, der in Abhängigkeit von den Modellen zu den Sensordaten eine Interpretation erzeugt. Die wichtigsten Modelle sind:

- 1.) Weltmodell: Das Weltmodell umfasst das Wissen über den Bildinhalt, d.h. die abgebildete Welt. Dabei kann es sich um sogenannte **generische Modelle** (wie sieht z.B. ein Haus ganz allgemein aus) oder um spezielle Modelle (wie sieht z.B. das "Weiße Haus" aus) handeln. Insbesondere das Allgemeinwissen über das Aussehen der Welt, die Zusammenhänge zwischen den Weltobjekten usw. sind hier enthalten.
- 2.) Beleuchtungs- und Reflektionsmodell: Diese Modelle umfassen die Beschreibung der Hauptlichtquellen (Position, räumliche Ausdehnung, Intensität, Farbe und Strahlungsart, z.B. diffuse Beleuchtung), sowie die Reflektions- und Refraktionseigenschaften der Weltobjekte.
- 3.) Sensormodell: Das Sensormodell beschreibt die photometrischen und geometrischen Eigenschaften der Sensoren. Es gibt an, wie die Punkte der dreidimensionalen Welt auf Punkte eines zweidimensionalen Bildes abgebildet werden: wie ergibt sich für eine spezielle 3D-Szene mit dem betreffenden Sensor aus einer bestimmten Position und unter bestimmten Beleuchtungsverhältnissen eine 2D-Helligkeitsmatrix (Bildmatrix).

Das verfügbare Wissen muss in der Wissensbasis in geeigneter Form repräsentiert werden, damit zur richtigen Zeit in möglichst einfacher Weise auf die richtigen Wissensinhalte zugegriffen werden kann. Unter **Wissensrepräsentation** versteht man dabei die Kombination von Datenstrukturen zur Wissensabspeicherung und von Prozeduren, um Manipulationen an den Datenstrukturen vorzunehmen bzw. mit Hilfe des Wissens Schlussfolgerungen zu ziehen. Ziel der Wissensrepräsentation ist, dass die Bildanalyse so vorgehen kann, als ob sie etwas über die zu lösenden Probleme wüsste (Bildverstehen!).

Bei der Wissensrepräsentation wird — wie bereits oben erwähnt — grundsätzlich zwischen zwei verschiedenen Techniken unterschieden:

- bei der **impliziten Repräsentation** ist das Wissen in die Verarbeitungsroutinen eingearbeitet. Dem Vorteil der schnellen und direkten Programmerstellung stehen als Nachteile gegenüber, dass das Wissen schwer erkennbar und schwer (z.B. auf Vollständigkeit und Widerspruchsfreiheit) zu überprüfen ist. Diese Repräsentationstechnik, die in der Frühphase der automatischen Bildauswertung fast ausschließlich eingesetzt wurde, wird heute im Rahmen der wissensgesteuerten Bildauswertung möglichst vermieden.
- bei der **expliziten Repräsentation** liegen das Wissen und die Verarbeitungskomponenten getrennt vor (vgl. Abbildung 9.1). Das Wissen kann z.B. durch sogenannte Wissenserkomponenten "eingelernt" sein. Es ist explizit verfügbar und daher leicht zu erkennen, zu überprüfen und zu ändern bzw. zu ergänzen. Da die Verarbeitungsprogramme das Wissen nicht selbst enthalten, sind sie allgemeiner gehalten und somit flexibler einsetzbar.

Im folgenden werden kurz die beiden im Rahmen der Bildanalyse am häufigsten verwendeten expliziten Wissensrepräsentationsformen vorgestellt. Dazu ist es zweckmäßig, für das Wissen eine weitere Unterteilung vorzunehmen:

- Daten und/oder Fakten (sogenanntes **deklaratives Wissen**), und
- Wissen darüber, wie die Daten bzw. Fakten genutzt werden können (sogenanntes **prozedurales Wissen**).

Bei der praktischen Anwendung muss das Faktenwissen strukturiert werden. Inhaltlich zusammengehörige Fakten bzw. Daten werden zu Datenobjekten zusammengefasst. Bei der Repräsentation stehen die Eigenschaften der Objekte sowie ihre gegenseitigen Beziehungen im Vordergrund. Zur expliziten Abspeicherung des deklarativen Wissens werden häufig semantische Netze verwendet. Im Gegensatz dazu eignen sich zur expliziten Repräsentation prozeduraler Wissensinhalte insbesondere sogenannte Regeln.

9.2.1 Repräsentation von Wissen durch semantische Netze

Semantische Netze repräsentieren Wissen als hochgradig verknüpfte Struktur, die durch einen verallgemeinerten Graphen beschrieben wird. Ein semantisches Netz besteht aus **Knoten** und **Kanten**, die die Knoten verbinden. Sowohl den Knoten als auch den Kanten sind Bedeutungen (Semantik) zugeordnet. Die Knoten stehen für **generische Objekte**, d.h. allgemeine Objekte (z.B. "Haus") bzw. **Objektklassen** oder für **individuelle Objekte** (z.B. "Weißes Haus"), die auch als **Instanzen** oder **Ausprägungen** bezeichnet werden. Bei den Kanten gibt es im wesentlichen zwei Arten von Kanten:

- 1.) **Beziehungskanten** setzen Objekte bzw. Objektklassen zueinander in Beziehung. Neben den allgemeinen Beziehungen, wie z.B. "liegt oberhalb", "berührt" usw., sind zwei Beziehungen und ihre Inversen von besonderer Bedeutung:
 - "ist ein" bzw. "hat als Beispiel"
 - "ist Teil von" bzw. "hat als Teil".

Beispiele:



Mit Hilfe von Beziehungskanten **vererben** sich z.B. die Eigenschaften der generischen Objekte auf ihre Ausprägungen.

- 2.) **Eigenschaftskanten** weisen einem Knoten ein Attribut zu; z.B.



Abbildung 9.3 zeigt ein stark vereinfachtes semantisches Netz zu einer allgemeinen Vorstadtszene (Weltmodell, d.h. Langzeitwissen). Unterhalb der Trennlinie ist eingetragen, wie während der Bildanalyse bestimmte Zwischenergebnisse (d.h. Kurzzeitwissen) zum Weltmodell in Beziehung gesetzt werden können.

9.2.2 Repräsentation von Wissen durch Regeln

Regeln werden in der Literatur häufig auch als **Produktionen** bezeichnet, und Systeme, die Regeln zur expliziten, prozeduralen Wissensrepräsentation verwenden, werden deshalb auch **Produktionssysteme (PS)** genannt. Im Bereich der künstlichen Intelligenz sind mit dem Begriff Produktionssystem ganz spezifische Eigenschaften verbunden. Insbesondere umfasst ein PS drei wesentliche Komponenten:

- die globale Wissensbasis,
- die Menge der Regeln und
- die Systemsteuerung.

Es soll hier nur auf die Regeln eingegangen werden. Eine Regel hat einen **Bedingungs-** und einen **Aktionsteil**:

WENN eine wohldefinierte Situation vorhanden ist,

DANN wird eine bestimmte Aktion durchgeführt.

Die Erfüllung des Bedingungssteils wird anhand der Wissensbasis überprüft. Falls die Bedingung erfüllt ist und die Regel zur Ausführung kommt, wird durch den Aktionsteil der Inhalt der Wissensbasis verändert. Bei den üblichen algorithmischen Programmiersprachen ist mit der Reihenfolge der Befehle die Vorgabe eines Kontrollflusses verbunden, der angibt, in welcher Reihenfolge die Befehle abgearbeitet werden sollen. Im Gegensatz dazu wird die Reihenfolge, in der die Regeln eines PS abgearbeitet werden, nicht durch die Reihenfolge ihrer Anordnung, sondern ausschließlich durch den Zustand der Wissensbasis und durch die Systemsteuerung bestimmt.

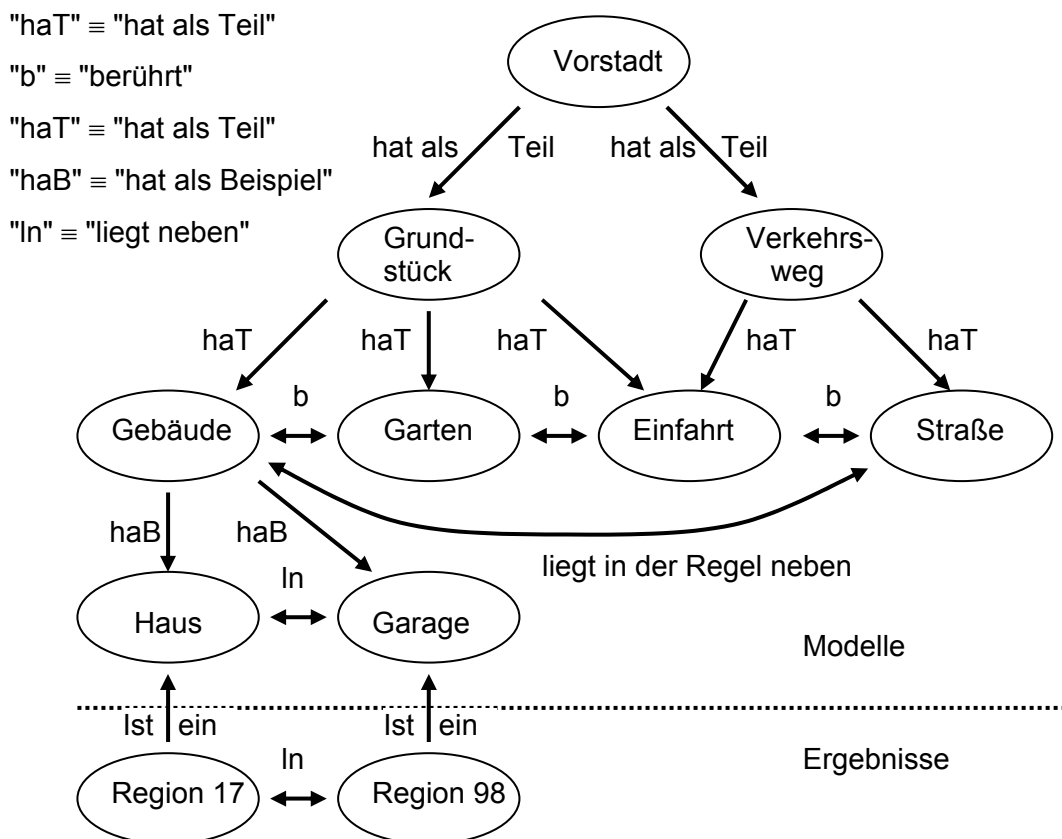


Abb. 9.3: Beispiel für ein semantisches Netz.

Dies hat den Nachteil, dass im Gegensatz zum Algorithmus ein erheblicher Aufwand benötigt wird, um herauszufinden, welche Regel als nächste ausgeführt werden soll. Dem steht jedoch der Vorteil gegenüber, dass das Wissen durch Hinzunehmen, Herausnehmen oder Modifizieren von Regeln einfach zu ändern ist. Einerseits kann die Wissensbasis auf diese Weise inkrementell aufgebaut werden, andererseits kann das System schnell und einfach an geänderte Voraussetzungen und neue Situationen angepasst werden. Bei jeder Änderung der Wissensbasis muss aber auf Konsistenz und Widerspruchsfreiheit der Regeln geachtet werden.

Im folgenden sind mehrere einfache Regeln angegeben, wie sie z.B. im Rahmen eines Luftbildinterpretations-Systems zum Einsatz kommen könnten:

Regel 1: WENN eine Region ein schmaler langer Streifen ist, und die Region als "Asphalt" klassifiziert wurde, DANN stellt die Region eine Straße dar.

Regel 2: WENN eine Region ein langer schmaler Streifen ist und wenn die Region als "Wasser" klassifiziert wurde, DANN stellt die Region einen Wasserweg (Fluss, Kanal ...) dar.

Regel 3: WENN eine Region klein ist und die Region als "Beton" oder "Eisen" klassifiziert wurde, DANN stellt die Region ein künstliches Objekt dar.

Regel 4: WENN eine Region im Schnittpunkt einer Straße und eines Wasserweges liegt und wenn die Region als künstliches Objekt klassifiziert wurde, DANN stellt die Region eine Brücke dar.

Ein Hauptanwendungsbereich für Produktionssysteme sind die sogenannten **Experten-systeme**, die z.B. für medizinische Diagnosen eingesetzt werden.

9.2.3 Weitere Bemerkungen, Alternativen

In der Praxis werden die beiden oben aufgeführten expliziten Wissensrepräsentationsformen häufig auch gemeinsam eingesetzt. Z.B. können die Modelle und die (Zwischen-)Ergebnisse in Form von semantischen Netzen abgespeichert werden; und die Veränderung

des Wissens, d.h. die Ableitung weiterer Ergebnisse bis hin zum Endergebnis wird durch die Anwendung von Regeln erreicht.

Auf andere Wissensrepräsentationsformen, wie z.B. den **Prädikatenkalkül erster Ordnung** aus der formalen Logik oder **Frames-and-Slots-Repräsentationen** wird hier nicht eingegangen.

9.3 Wissensnutzung (Systemsteuerung, Kontrolle)

"More knowledge means less search"

Das Ziel der wissensgesteuerten Bildanalyse besteht darin, aus einem Eingabebild eine Beschreibung und Interpretation abzuleiten. In einem komplexen System stehen i.A. für jedes Zwischenergebnis jeder beliebigen Verarbeitungsstufe zahlreiche Möglichkeiten zur Fortsetzung der Verarbeitung zur Verfügung. Darüber hinaus sind die zu einem beliebigen Zeitpunkt verfügbaren Zwischenergebnisse üblicherweise mehrdeutig, so dass die Zahl der Alternativen noch größer wird. Somit entsteht bei der Wissensnutzung das Problem, die Aktionen des Systems sowie die Auswahl der (Teil-)Daten, auf die die Aktionen angewandt werden sollen, geeignet zu steuern.

Allgemein kann die Wissensnutzung als Folge von Transformationen angesehen werden, die ausgehend von einem initialen Zustand der Wissensbasis das System in einen gewünschten Zielzustand überführt, der die Interpretation der Szene darstellt. Da zu jedem Zeitpunkt verschiedene alternative Transformationen anwendbar sind, kann die Analyse als Suchproblem im Graphen der Systemzustände angesehen werden: die Aufgabe der Systemkontrolle ist es, die Systemaktivitäten so zu steuern, dass ein "optimaler" Pfad zwischen dem initialen Systemzustand und dem gewünschten Endzustand gefunden wird.

Es ist unmittelbar einleuchtend, dass bei komplexen Systemen nicht blind alle Alternativen entwickelt, d.h. alle möglichen Pfade ausprobiert werden können. Derartige **tiefen-** oder **breitenorientierten Suchverfahren (breadth first bzw. depth first)** scheitern an der sogenannten **kombinatorischen Explosion**: die riesige Zahl möglicher Alternativen kann nicht mehr in vertretbarer Zeit untersucht werden. Im Bereich der künstlichen Intelligenz hat man daher eine ganze Reihe von Suchverfahren gesammelt bzw. entwickelt, die problemspezifisches Wissen einsetzen, um den Aufwand in Grenzen zu halten. Mit Hilfe dieser sogenannten **heuristischen Information** wird versucht, die Zahl der tatsächlich zu entwickelnden Alternativen möglichst gering zu halten. Auf die Vielzahl dieser Verfahren kann hier nicht eingegangen werden.

Im folgenden sollen einige spezielle Kontrollmechanismen vorgestellt werden, die Wissen einsetzen, um möglichst günstig die Interpretation einer Szene aus den Eingabebilddaten abzuleiten.

9.3.1 Bottom-Up-Kontrolle

Bei der **Bottom-Up-Vorgehensweise** (siehe Abbildung 9.4 (a)) werden die Eingangs-Bilddaten Schritt für Schritt in "organisiertere" und im Hinblick auf das Endergebnis nützlichere Information umgesetzt. Die Verarbeitung erfolgt dabei in der Low-Level- und Medium-Level-Verarbeitung nur in Abhängigkeit von den Eingabe-Daten und ohne den Einfluss von Modellen d.h. Wissen. Die Vorgehensweise wird daher auch als **datengetriebene Vorgehensweise** bezeichnet. Die Datenmenge wird von Verarbeitungsschritt zu Verarbeitungsschritt reduziert. Erst gegen Ende der Auswertung kommt das Wissen zum Tragen, um aus der symbolischen Szenenbeschreibung die Interpretation abzuleiten.

Ein Vorteil der Bottom-Up-Vorgehensweise ist, dass sie leicht auf andere Aufgabenstellungen übertragen werden kann. Da die grundlegenden Verfahren wissensunabhängig arbeiten, müssen sie nicht verändert werden; Änderungen fallen erst bei der High-Level-Verarbeitung an, wo das Wissen zur Interpretation eingesetzt wird. Die Tatsache, dass die "frühen" Verarbeitungsstufen ohne Wissen ablaufen, birgt aber auch Nachteile in sich: da die Verfahren uninformatiert, d.h. pauschal vorgehen, sind die jeweils erzielten Ergebnisse u.U. grob und fehlerträchtig. Lückenhafte Zwischenergebnisse einerseits erschweren dann die Verarbeitung

auf den höheren Ebenen. Fehler andererseits können nicht frühzeitig erkannt werden und führen so zu zahlreichen überflüssigen Berechnungen auf höheren Ebenen.

Die Bottom-Up-Strategie stellt beim derzeitigen Stand der Technik insbesondere bei der Verarbeitung industrieller Szenen die am häufigsten verwendete Bildanalyse-Strategie dar.

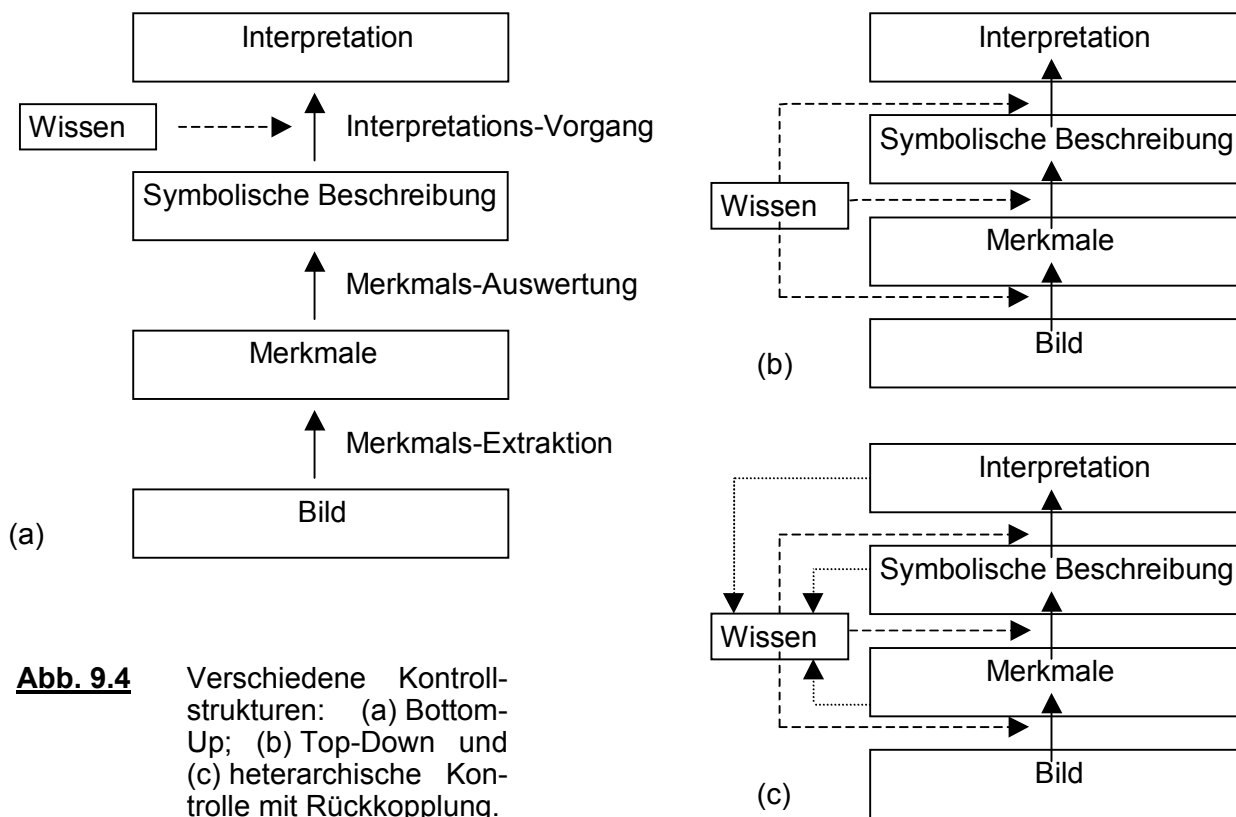


Abb. 9.4 Verschiedene Kontrollstrukturen: (a) Bottom-Up; (b) Top-Down und (c) heterarchische Kontrolle mit Rückkopplung.

9.3.2 Top-Down-Kontrolle

Das Gegenteil zur Bottom-Up-Vorgehensweise stellt die **Top-Down-Vorgehensweise** dar (siehe Abbildung 9.4 (b)), bei der die gesamte Verarbeitung durch das Wissen kontrolliert und geleitet wird. Diese Vorgehensweise wird daher auch oft als **modellgetriebene Vorgehensweise** bezeichnet.

Typische Top-Down-Vorgehensweisen sind **Hypothese-Test-** bzw. **Prädiktions-Verifikations-Schemata**. Dabei werden aufgrund des Wissens über die Szene auf jeder Verarbeitungsstufe Hypothesen aufgestellt, und diese werden dann gezielt getestet, d.h. überprüft. Dies bedeutet konkret, dass z.B. aufgrund des bislang angesammelten Wissens bestimmte Objekte mit einem bestimmten Erscheinungsbild an einer bestimmten Bildposition vorhergesagt werden (Prädiktion). Bereits die Low-Level-Verarbeitungsverfahren können dann ganz gezielt und adaptiert eingesetzt werden, um eben diese Hypothesen zu überprüfen, d.h. die Prädiktion zu verifizieren oder zu falsifizieren (verwerfen).

Bei einer reinen Top-Down-Vorgehensweise können eine Vielzahl falscher Hypothesen aufgestellt werden, die wiederum einen enormen, überflüssigen Rechenaufwand nach sich ziehen. In der Praxis werden reine Top-Down-Methoden daher nur dort eingesetzt, wo für die Szenarien starke Zwangsbedingungen (constraints) existieren, d.h. die Zahl möglicher Hypothesen nicht zu groß ist.

9.3.3 Heterarchische Kontrolle mit Rückkopplung

Bottom-Up- und Top-Down-Vorgehensweisen werden auch als **hierarchische Kontrollstrukturen** bezeichnet, da die Systemaktivitäten in einer klaren Hierarchie gegliedert sind. Im Gegensatz dazu zeigt Abbildung 9.4 (c) eine **heterarchische Kontrollstruktur mit Rückkopplung (feed back)**, bei der die Ergebnisse jeder Verarbeitungsstufe auf einer höheren

Ebene jeweils überprüft und in Frage gestellt werden können. Üblicherweise verlaufen Teile der Auswertung nach der Bottom-Up- und Teile nach der Top-Down-Vorgehensweise: häufig wird ein Eingabebild bis zu einem gewissen Stadium erst einmal ohne Wissen, d.h. bottom-up, ausgewertet. Die weitere Verarbeitung dagegen ist nicht festgelegt, sondern hängt vom Bildmaterial und den jeweiligen Zwischenergebnissen ab. Auf jeder Stufe wird entschieden, ob die Verfahren systematisch und ohne Wissen eingesetzt werden sollen, um z.B. möglichst schnell möglichst viele Merkmale zu erzeugen, oder ob mit Hilfe des Wissens gezielt Hypothesen aufgestellt und überprüft werden sollen. Die Rückkopplung ermöglicht in diesem Zusammenhang, dass die Ergebnisse bestimmter Auswertestufen an frühere Auswerteeinheiten zurückgeführt werden können: z.B. können die bislang ermittelten Merkmale mit Hilfe des Wissens überprüft werden und der Merkmalsextraktionsprozess kann danach nochmals in verbesserter Form oder mit verbesserten Einstellungen wiederholt werden.

Heterarchische Systeme sind insbesondere bei komplexen Szenen den hierarchischen Ansätzen überlegen. Sie sind jedoch schwierig zu realisieren und sind deshalb noch weitgehend Gegenstand der Forschung.

9.4 Beispiel für ein allgemeines System-Modell

Das im folgende vorgestellte allgemeine System-Modell (siehe Abbildung 9.5) für eine wissensbasierte Bildinterpretation geht im wesentlichen auf Kanade und Pinz zurück; (s. z.B. [Pinz]).

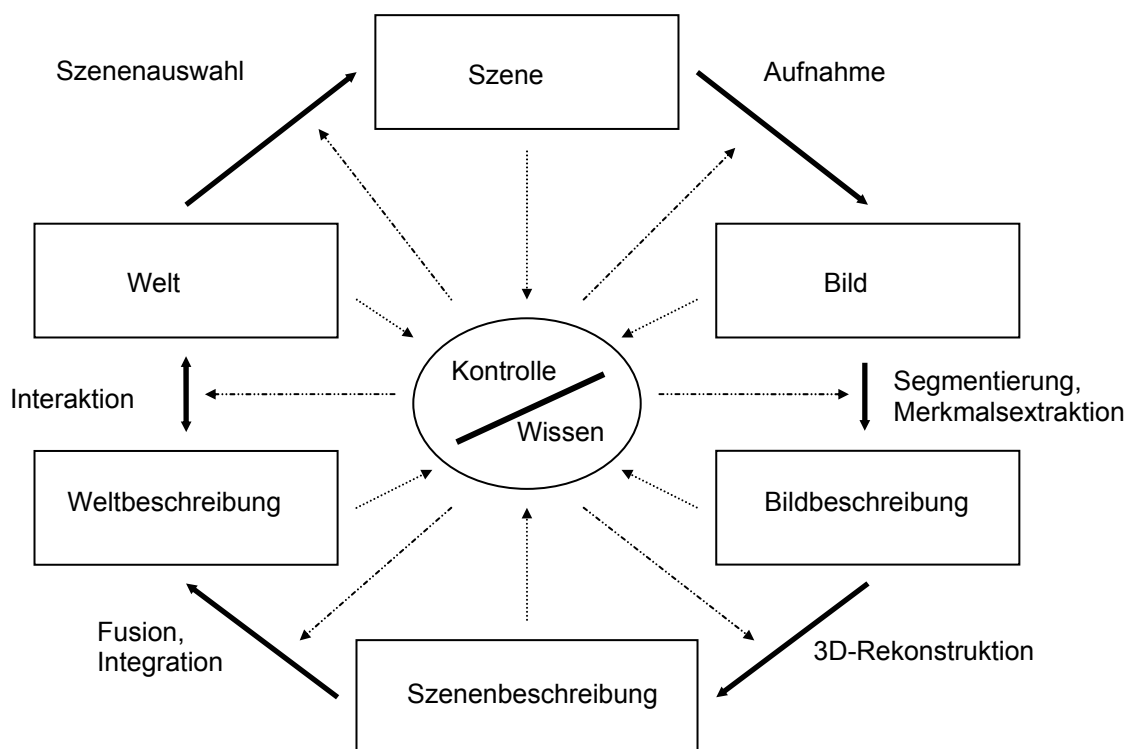


Abb. 9.5: Allgemeines System-Modell für eine wissensbasierte Bildinterpretation nach Kanade und Pinz.

Die Kästchen in Abbildung 9.5 bezeichnen die verschiedenen Repräsentationsebenen. Die durchgezogenen Pfeile stehen für die Prozesse, die die Übergänge zwischen den jeweiligen Repräsentationsebenen bewirken. Die Kontrolle steuert die Prozesse (strich-punktierte Pfeile), um eine vorgegebene Aufgabenstellung zu erfüllen. Dabei wird auf das allgemeine Wissen und die jeweils neu gewonnenen Erkenntnisse (gestrichelte Pfeile) zurückgegriffen.

"**WELT**" steht für eine Repräsentation der 4D-Welt (x,y,z,t) mit Hilfe generischer Objekt-Klassen. Diese Repräsentationsebene enthält also eine allgemeine, z.B. parametrische

Beschreibung möglicher Objekte, ihrer Eigenschaften und ihrer gegenseitigen Beziehungen; (Beispiel: Reihenhauszelle).

Eine "**SCENE**" ist eine spezielle Auswahl eines 3D-Weltausschnittes zu einem bestimmten Zeitpunkt. Eine Szene wird durch die Anordnungen von Instanzen aus den allgemeinen Klassen festgelegt; (Beispiel: Parkplatz der FH am 1. April 2006 um 8.10 Uhr).

Ein "**BILD**" ist die 2D-Projektion der ausgewählten Szene unter Festlegung bestimmter Kameraparameter; (z.B. Kamerastandort, Blickrichtung).

Die "**BILDBESCHREIBUNG**" ist die mit Hilfe von Bildverarbeitungsverfahren gewonnene symbolische Beschreibung des 2D-Bildinhaltes; (z.B. Liste der segmentierten Regionen einschließlich der ermittelten Merkmale wie Größe, Form usw.).

Die "**SCENENBESCHREIBUNG**" versucht, die aus einem oder mehreren Bildern gewonnenen 3D-Erkenntnisse über die ausgewählte Szene darzustellen; (z.B. 3D-Approximation der in der Szene enthaltenen Objekte durch verallgemeinerte Zylinder oder Angabe der Oberflächenorientierung für die sichtbaren Flächen; intrinsic images).

Anmerkung: es gibt auch Ansätze, die versuchen, aus einem oder mehreren Bildern direkt, d.h. unter Umgehung der Bildbeschreibung die Szenenbeschreibung zu erzeugen; (z.B. "Shape from Shading" und andere "Shape from X" -Ansätze; siehe Kapitel 8).

Die "**WELTBESCHREIBUNG**" muss in der Lage sein, mehrere Szenen und den räumlichen und zeitlichen Bezug zwischen diesen Szenen zu repräsentieren. (Integration der Gemeinsamkeiten der räumlich und/oder zeitlich überlappenden Szenenbeschreibungen und Fusion mit weiteren Wissensquellen).

Das Ergebnis der Auswertung kann unter dem Aspekt der gestellten Aufgabe z.B. von einem (menschlichen) Benutzer durch Vergleich mit der Realität interaktiv kontrolliert und gegebenenfalls korrigiert werden. Eine andere Möglichkeit besteht darin, dass ein Roboter mit der ihn umgebenden Welt interagiert (z.B. Ergreifen eines Objektes oder eigener Positionswechsel), um eine gestellte Aufgabe zu erfüllen.

Neben der Kontrolle spielt das Wissen bei dem vorgestellten System-Modell eine wesentliche Rolle. Die statischen Teile des Wissens liegen meist in Form von Modellen vor; (z.B. Weltmodell, Beleuchtungs- und Reflektionsmodell, Sensormodell). Die dynamischen Teile des Wissens stellen im wesentlichen die sich ständig ändernden Zwischenergebnisse dar. Wichtig, aber besonders schwierig zu formulieren ist das Allgemeinwissen (Common-Sense).

9.5 Abschließende allgemeine Überlegungen zu Bildanalyse-Systemen

Bildanalyse-Systeme bearbeiten i.d.R. Schwarz/Weiß- oder Farbbilder, die sie entweder direkt über eine angeschlossene Kamera einziehen (on-line-Betrieb), oder die ihnen nach externer Aufzeichnung überspielt werden (off-line-Betrieb). Manche Systeme verarbeiten zusätzlich noch weitere Sensor-Informationen, wie z.B. Entfernungsdaten, die etwa mit Hilfe eines Laser-Entfernungsmessers ermittelt werden.

Der Systemmodul mit den Verfahren zur Ermittlung elementarer Bildbestandteile (siehe Abbildung 9.1) verfügt i.d.R. über Methoden zur Bildrestaurierung (z.B. Elimination von Sensorfehlern), Bildverbesserung (z.B. Kontrastverbesserung) und Bildaufbereitung (z.B. Vergrößerung). Weiterhin findet sich hier meist eine ganze Reihe alternativer Segmentierungsverfahren, die flächen- und/oder konturorientiert vorgehen. Diese Verfahren werden im Anfangsstadium der Analyse meist bottom-up, d.h. ohne Einfluss des Wissens eingesetzt. Leistungsfähige Systeme arbeiten später häufig iterativ und rufen die Segmentierungsverfahren gegebenenfalls zu einem späteren Zeitpunkt noch einmal auf (feed back), um dann top-down, d.h. wissens- bzw. modellgesteuert

- bisherige grobe Ergebnisse zu überprüfen und/oder zu verfeinern (Nachsegmentierung), oder
- gezielt in bestimmten Bildbereichen bestimmte Objekt-Erscheinungsformen zu ermitteln (Hypothesengesteuerte Segmentierung).

Ein zentraler Bestandteil aller Systeme ist neben der getrennt vorliegenden Wissensbasis die Datenstruktur zur Aufnahme der (Zwischen-)Ergebnisse. Häufig ist dieser "Speicher" identisch oder ähnlich zur Wissensbasis aufgebaut. Der Ergebnisspeicher dient als Vermittler zwischen den verschiedenen Verarbeitungsstufen (low level, medium level, high level) und wird auch innerhalb jeder Verarbeitungsstufe von den aufgerufenen Verfahren verwendet.

Im Bereich der Wissensnutzung unterscheiden sich die Systeme am meisten. Da in Kapitel 9.3 die Wissensnutzung im wesentlichen unter dem Aspekt der Kontrollstrategie behandelt wurde, sollen hier noch ergänzend zwei häufig verwendete Ansätze angesprochen werden, die direkt auf die Interpretation hin ausgerichtet sind.

Eine ganz wesentliche Rolle bei der Interpretation spielen Relaxationsverfahren (diskrete oder kontinuierliche). Da diese Verfahren in den zurückliegenden Kapiteln an zahlreichen Stellen aufgeführt wurden, sollen sie hier nicht noch einmal erläutert werden. Es sei in diesem Zusammenhang insbesondere auf Kapitel 5 hingewiesen, wo der Waltz'sche Filteralgorithmus (diskrete Relaxation) direkt zur Interpretation von Strichzeichnungen eingesetzt wurde.

Eine weitere Klasse von Interpretationsverfahren sind die sogenannten **Graph-Match-Verfahren** oder **Graphen-Vergleichsverfahren**. In Kapitel 9.2.1 wurde erläutert, wie Graphen—speziell semantische Netze— zur Repräsentation von Wissen (Modellen) aber auch zur Ab-speicherung von Ergebnissen verwendet werden können. Bei der Wissensnutzung geht es darum, eine Korrespondenz zwischen Modell und Eingabebild bzw. einer daraus abgeleiteten symbolischen Beschreibung zu erzielen. Das Aufstellen einer derartigen Korrespondenz wird als **Matching** bezeichnet. Wenn sowohl das Modell, als auch das symbolisch beschriebene Eingabebild in Form von Graphen (z.B. semantischen Netzen) dargestellt sind, können die aus der Graphentheorie bekannten Verfahren zur Bestimmung von Korrespondenzen zwischen zwei Graphen G_1 und G_2 angewandt werden. Ein **Graphen-Isomorphismus** liegt vor, wenn beide Graphen G_1 und G_2 strukturidentisch sind. Ein **Teilgraphen-Isomorphismus** liegt vor, wenn zu G_1 und G_2 zwei Teilgraphen $G' \leq G_1$ bzw. $G'' \leq G_2$ existieren, so dass G' und G'' isomorph sind. Wenn zu einem Eingabebild ein solcher (Teil-) Graphen-Isomorphismus zu einem Modell gefunden werden kann, so ist das Eingabebild bzw. der dem Teilgraphen entsprechende Teil des Eingabebildes interpretiert.

Das Problem, das bei der Anwendung solcher Zuordnungsverfahren bei der Bildinterpretation auftritt, besteht darin, dass in dem Graphen, der aus dem Eingabebild abgeleitet ist, i.A. mehr oder weniger starke Abweichungen gegenüber dem Modell auftreten. Eine Lösung bilden hier die sogenannten **unexakten Vergleichsverfahren (inexact matching)**, die in gewissem Umfange Unterschiede in den Markierungen oder Attributwerten bei Kanten und Knoten sowie in der globalen Struktur tolerieren. Diese unexakte Zuordnung entspricht dann der bestmöglichen Interpretation der Eingabedaten anhand des gegebenen Modells.

Ein wichtiger Aspekt, der thematisch in engem Zusammenhang mit Fragen der Wissensnutzung steht, und der in vielen Bildanalyse-Systemen unabhängig vom verwendeten Ansatz zur Wissensdarstellung und -nutzung von Bedeutung ist, ist die sogenannte Planung. Unter **Planung** versteht man in der Bildanalyse und im Bereich der künstlichen Intelligenz die Aufstellung eines Planes, der eine Folge von Aktionen definiert, die der Erreichung eines komplexen Zieles dienen. Planung im Bereich der Bildinterpretation bedeutet dabei meist, dass mit Hilfe einiger schneller oder besonders zuverlässiger Verfahren zuerst nur grobe Analyse-Ergebnisse erzielt werden. Diese werden nachfolgend schrittweise verfeinert. Ein Gewinn an Rechenzeit und/oder Zuverlässigkeit ergibt sich dadurch, dass in der Verfeinerungsphase nur noch bestimmte Operationen und diese nur noch in bestimmten Untermengen von Daten, die sich durch die vorhergehende Bearbeitung ergeben, angewandt werden.

Literatur (allgemeine Lehrbücher)

- Bässmann H., Kreyss J., "Bildverarbeitung Ad Oculos", Springer, 2004
- Bennamoun M., Mamic G., "Object Recognition", Springer, 2002
- Bischof H., Kropatsch W., "Digital Image Analysis", Springer, 2001
- Burger W., Burge M.J., "Digitale Bildverarbeitung", Springer, examen press, 2005
- Demant C., Streicher-Abel B., Waskewitz P., "Industrial Image Processing", Springer, 1999
oder auf deutsch: "Industrielle Bildverarbeitung", Springer, 2002
- Faugeras O., "Three-Dimensional Computer Vision", MIT Press, 1994
- Forsyth D. A., Ponce J., "Computer Vision", Prentice Hall, Pearson Education, 2003
- Goldstein E. B., "Wahrnehmungspsychologie", Spektrum Akademischer Verlag, 2002
- Gonzales R., Woods R., "Digital Image Processing", Addison Wesley, 2003
- Horn B. K. P., "Robot Vision", McGraw Hill, 1986
- Jähne B., "Digitale Bildverarbeitung", Springer, 2002
- Klette R., Koschan A., Schlüns K., "Computer Vision", Vieweg, 1998
- Mallot H. A., "Sehen und die Verarbeitung visueller Informationen", Vieweg, 2000
- Nischwitz A., Haberäcker P., "Masterkurs Computergrafik und Bildverarbeitung", Vieweg, 2004
- Pinz A., "Bildverstehen", Springer, 1997
- Russ J. C., "The Image Processing Handbook", Springer, 1999
- Sonka M. et al., "Image Processing, Analysis and Machine Vision", Thomson Computer Press, 2007
- Trucco E., Verri A., "Introductory Techniques for 3D-Computer Vision", Prentice Hall, 1998
- Winston P. H., "Artificial Intelligence", Addison-Wesley, 2005

φ -d-Akkumulatorfeld	4-14	Ecken-Detektor	7-3
φ -d-Parameterraum	4-12	Eigenschaftskante	9-5
3D-Problematik	5-1	einäugige Tiefenmerkmale	2-3
Abbruchkante	8-4	einfacher Punkt	4-6
Ablaufkontrolle	9-1	Empfindungen, visuelle	2-2
Ableitung	4-2	Endpunkt	4-7
absolute Koordinaten	4-16	Epipolare	6-5
Ähnlichkeits-Maßzahl	3-2	Erosion	4-6
Akkomodation	2-4	Expertensystem	9-5
Akkumulatorfeld, φ -d-	4-14	explizite Repräsentation	9-4
Aktionsteil	9-5	Explosion, kombinatorische	9-7
allgemeiner Kamerastandort	5-2	Extraktionsverfahren, sequent.	4-15
Ansatz, grauwertorientierter	6-5	Faltungssatz	3-4
Ansatz, inhaltsorientierter	6-5	feed back	9-8
Ansatz, merkmalsorientierter	6-5	flächenorientierte Segmentier.	4-1
Ansatzstelle	4-15	Fleck, gelber	2-1
Approximation, Polygonzug	4-16	Fließmuster	2-3
Ausprägung	9-5	Fluchtpunkt	7-1
äußere Kante	8-4	Fluss, optischer	2-3
Autokorrelation	3-4	focus of expansion	7-1
backward constraint propag.	5-3	Form aus Form	8-1
Bedingungsteil	9-5	Form aus Schattierung	8-1
Beschreibung, symbolische	9-2	Form aus Textur	8-1
Bewegungsparallaxe	2-4, 7-2	forward constraint propagation	5-3
Beziehungskante	9-5	Fovea	2-1
Bildanalyse, modellgesteuerte	9-1	Frames-and-Slots-Repräsent.	9-7
Bildanalyse, wissensgesteuerte	9-1	Gabel	5-2
Bilddifferenz	3-1	Ganglien	2-2
Bildfolge	7-1	gelber Fleck	2-1
Bildfolgenauswertung	7-1	generisches Modell	9-4
Bildverstehen	9-1	generisches Objekt	9-5
Binarisierung	4-5	Gestalts-Gesetze	2-5
binokulare Tiefenmerkmale	2-3	Gradient	4-3
Blendenproblem	7-2	Gradient (Disparität)	6-9
Bottom-Up-Vorgehensweise	9-7	Graph-Match-Verfahren	9-11
breadth first	9-7	Graphen-Isomorphismus	9-11
breitenorientierte Suchverfah.	9-7	Graphen-Vergleichsverfahren	9-11
chain code	4-16	grauwertorientierter Ansatz	6-5
Charakteristika, innere	5-1	Grenzlinie	5-2
Colliculus Superior	2-2	Harris-Corner-Detektor	7-4
Connectivity-Number	4-6	heterarchische Kontrollstruktur	9-8
constraint	5-3, 6-6, 7-7	Heuristische Information	9-7
Corpus Geniculatum Laterale	2-2	hierarchische Kontrollstruktur	9-8
datengetriebene Vorgehensw.	9-7	Hindernis-Ermittlung	6-1
deklaratives Wissen	9-4	Hough-Transformation	4-10
depth first	9-7	Hypothese-Test-Schemata	9-8
differentieller Ansatz (Bildfolg.)	7-6	ideale Stereo-Konfiguration	6-2
Differenz	4-3	ikonische Verarbeitung	9-2
Dilatation	4-7	implizite Repräsentation	9-4
diskrete Relaxation	5-3	inexact matching	9-11
Disparität	2-3, 6-4	inhaltsorientierter Ansatz	6-5
Disparitäten-Differenz	6-9	innere Charakteristika	5-1
Disparitäts-Gradient	6-9	innere Linie	5-2
Disparitätsebene	6-6	Instanz	9-5
disparity gradient constraint	6-10	Interpretation	5-1, 9-2
Dunstperspektive	2-4	intrinsic characteristics	5-1

inverse Projektion	6-1	matching, inexact	9-11
Isolinie (gleicher Helligkeit)	8-2	merkmalsorientierter Ansatz	6-5
Isomorphismus, Graphen	9-11	Modell	9-1
Isomorphismus, Teilgraphen	9-11	Modell, generisches	9-4
Kamerastandort, allgemeiner	5-2	modellgesteuerte Bildanalyse	9-1
Kante	9-5	modellgetriebene Vorgehensw.	9-8
Kante, äußere	8-4	Modellwissen	5-1
Kante, Schatten-	8-4	monokulare Bewegungsparall.	2-4, 7-2
Kanten-/Linienelement-Verkett.	4-9	monokulare Tiefenmerkmale	2-3
Kanten-Schablonen	4-1	Nadel-Diagramm	8-1
Kantendetektion	4-1	Nerv, optischer	2-2
Kantendetektor	4-1	Netz, semantisches	9-5
Kantenelemente	4-1	Netzhaut	2-1
Kantenextraktion	4-15	Nicht-Maximum-Unterdrückung	4-6
Kettencodierung	4-16	normierter Korrelations-Koeff.	3-2
Kirsch-Operator	4-4	Objekt, generisches	9-5
Klötzchenwelt	5-1	Objekt, unmögliches	5-5
Knoten	9-5	Objekterkennungs-Paradoxon	2-6
Kollisionsvorhersage	7-5	Objektklasse	9-5
kombinatorische Explosion	9-7	optical flow	2-3, 7-2
konkave innere Linie	5-2	optische Achse	6-1
Kontrollstruktur, heterarchische	9-8	optischer Fluss	2-3, 7-2
Kontrollstruktur, hierarchische	9-8	optischer Nerv	2-2
Kontur-Repräsentation, unvoll.	4-16	Origami-Objekte	5-5
Kontur-Repräsentation, vollst.	4-15	Parameterraum	4-10, 4-12
Kontur-Repräsentationen	4-15	Parameterraum, ϕ -d-	4-12
konturorientierte Segmentier.	4-1	Parameterraum, m-b-	4-10
konvexe innere Linie	5-2	partielle Ableitung	4-2
kooperatives Netzwerk	6-7	Perspektive, Dunst-	2-4
Koordinaten, absolute	4-16	Perspektive, lineare	2-4
Koordinaten, relative	4-16	Pfeil	5-2
Korrelations-Koeffizient	3-2	photogrammetrisches Stereo	6-1
Korrelations-Koeffizient, norm.	3-2	photometrisches Stereo	8-2
Korrelationsgebirge	3-3	Planung	9-11
Korrelationsmatrix	3-3	PMF-Algorithmus	9-8
Korrespondenzproblem	6-5, 7-2	Polygonzug-Approximation	4-16
Kreuzkorrelation	3-4	Prädikatenkalkül erster Ord.	9-7
Künstliche Intelligenz	9-2	Prädiktions-Verifikations-Sche.	9-8
Kurzzeitwissen	9-3	Produktion	9-5
L	5-2	Produktionssystem	9-5
Lambert'scher Strahler	8-2	prozedurales Wissen	9-4
Langzeitwissen	9-3	Punkt, einfacher	4-6
Laplace-Operator	4-4	Punktoperator	4-6
lineare Perspektive	2-4	quantitativer Vergleich	3-2, 3-3
Linie, innere	5-2	Randpixel	4-6
Linien-Schablonen	4-1	Rechts-Differenz	4-3
Liniendetektion	4-1	reflection map	8-2
Liniendetektor	4-1	Reflektionskarte	8-2
Liniensextraktion	4-15	Regel	9-5
Links-Differenz	4-3	Regionen	4-1
Linse	2-1	relative Koordinaten	4-16
m-b-Parameterraum	4-10	Relaxation	4-8
Makula	2-1	Relaxation, diskrete	5-3
Markierung, Linien	5-2	Relaxationsverfahren	4-8
Marr und Poggio, Algorithmus	6-6	Repräsentation, explizite	9-4
Matching	9-11	Repräsentation, implizite	9-4

Repräsentationen, Frames-&	9-7	Verschiebevektorfeld	7-2
Repräsentationen, Kontur-	4-15	Vertrauen	4-7
Retina	2-1	Visuelle Empfindungen	2-2
Robert's-Kreuz-Operator	4-4	visuelle Wahrnehmung	2-1
Rückkopplung	9-8	Visueller Cortex	2-2
Rückwärtsausbreitung der Zw.	5-3	vollständige Kontur-Repräs.	4-15
Sakkaden	2-2	Vorgehensweise, Bottom-Up-	9-7
Schablonen-Vergleich	3-3	Vorgehensweise, datengetrieb.	9-7
Schatten-Kante	8-5	Vorgehensweise, modellgetr.	9-8
Segmentierung	4-1	Vorgehensweise, Top-Down-	9-8
Segmentierung	4-1	Vorwärtsausbreitung der Zw.	5-3
Sehstrahl	6-5	Wahrnehmung, visuelle	2-1
Sehgrube	2-1	Waltz'scher Filteralgorithmus	5-3
Sehsystem	2-1	Winkelgeschwindigkeit	2-4
semantisches Netz	9-5	Wissen	9-1
sequentielle Extraktionsverf.	4-15	Wissen, deklaratives	9-4
shape from shading	8-1	Wissen, prozedurales	9-4
shape from shape	8-1	Wissensbasis	9-1, 9-3
shape from texture	8-1	wissensgesteuerte Bildanalyse	9-1
Smoothness-Constraint	7-7	Wissensrepräsentation	9-4
Stäbchen	2-1	Zapfen	2-1
Stereo-Konfiguration, ideale	6-2	Zufallsstereogramm	2-3
Stereo, photogrammetrisches	6-1	Zwangsbedingung	5-3, 6-6, 7-7
Stereo, photometrisches	8-2	zweiäugige Tiefenmerkmale	2-3
Stereobildauswertung	6-1, 8-2	Zyklopen-Abstand	6-9
Strahler, Lambert'scher	8-2		
Strichzeichnung	5-1		
structure from motion	7-1		
Strukturmatrix	7-3		
Suchverfahren, breitenorient.	9-7		
Suchverfahren, tiefenorientierte	9-7		
symbolische Beschreibung	9-2		
symbolische Verarbeitung	9-2		
Symmetrische Differenz	4-3		
System-Modell	9-9		
T	5-2		
Teilgraphen-Isomorphismus	9-11		
template matching	3-3		
Texturkompression	2-4, 8-5		
Tiefenmerkmale	2-3		
tiefenorientierte Suchverfahren	9-7		
Top-Down-Vorgehensweise	9-8		
Transformation, Hough	4-10		
unexakte Vergleichsverfahren	9-11		
unmögliches Objekt	5-5		
unvollständige Kontur-Repräs.	4-16		
Verarbeitung, ikonische	9-2		
Verarbeitung, symbolische	9-2		
Verdünnung	4-6		
Vererbung	9-5		
Verfahren mit inverser Proj.	6-1		
Vergenz	2-3, 6-3		
Vergleich, quantitativer	3-2, 3-3		
Vergleichsverfahren, unexakte	9-11		
Verkettung, Kanten-/Linienele.	4-9		
Verschiebevektor	7-2		