

ALBERT-LUDWIGS-UNIVERSITÄT
FREIBURG
INSTITUT FÜR INFORMATIK

Lehrstuhl für Mustererkennung und Bildverarbeitung

Prof. Dr.-Ing. Hans Burkhardt



Bildverarbeitungspraktikum I

Praktikumsanleitungen

Ausgabe 2005

VORWORT

Der hier vorliegende Band enthält die Versuchsanleitungen von Teil I eines Bildverarbeitungspraktikums, in welchem die folgenden Themenschwerpunkte behandelt werden:

1. Bildvorverarbeitung
2. Kamerakalibrierung
3. Farbe
4. Autofokussysteme
5. Mustererkennung

Diese Veranstaltung gibt Gelegenheit zum praktischen Arbeiten zu ausgewählten Themengebieten. Dabei wird zum einen der Vorlesungsstoff ergänzt und zum anderen Gelegenheit zum Experimentieren mit digitaler Bildverarbeitungs/Hardware und Software gegeben. Die Themengebiete sind so ausgewählt, daß sie eigenständig an einem Nachmittag bearbeitet werden können.

Freiburg, im WS 2005

Prof. Dr.-Ing. H. Burkhardt

Nachdruck und Kopie, auch auszugsweise ist nur mit Genehmigung des Verfassers gestattet.

Inhaltsverzeichnis

1	Bildvorverarbeitung	11
1.1	Einleitung	11
1.2	Bildverbesserung durch Punktoperatoren und lokale Filterung	12
1.2.1	Grauwerttransformationen	13
1.2.2	Lineare Lokaloperatoren	19
1.2.3	Medianfilter	23
1.3	Kantenextraktion	29
1.3.1	Gradientenoperator	30
1.3.2	Laplaceoperator	33
1.3.3	Differenzen von Mittelwerten	35
1.3.4	Cannyoperator	37
1.4	Binarisierung und morphologische Binärbildverarbeitung	41
1.4.1	Binarisierung	41
1.4.2	Dilatation	42
1.4.3	Erosion	46
1.4.4	Open und Close	48
1.5	Aufgaben	51
1.5.1	Theoretische Aufgaben	51
1.5.2	Praktische Aufgaben	52
2	Kamerakalibrierung	57
2.1	Einleitung	57
2.2	Projektive Geometrie	58
2.2.1	Einleitung	58
2.2.2	Die projektive Ebene	60
2.2.3	Transformationen und ihre Hierarchie	61
2.2.4	Berechnung einer Homographie	64
2.3	Kameramodell	66

2.3.1	Die Lochkamera	66
2.3.2	Linsenverzerrungen	68
2.3.3	Projektion einer Ebene	70
2.4	Kalibrierung	71
2.4.1	Einleitung und Überblick	71
2.4.2	Kalibriermuster	72
2.4.3	3D Kalibrierung	73
2.4.4	2D Kalibrierung	76
2.4.5	Selbstkalibrierung	79
2.5	Numerisches Handwerkszeug	81
2.5.1	Fehlerquadrat Methode	81
2.5.2	Normalisierung	82
2.6	Literatur	83
2.7	Aufgaben	83
2.7.1	Theoretischer Teil	83
2.7.2	Praktischer Teil	85
3	Farbe	88
3.1	Einleitung	88
3.2	Menschliche Farbwahrnehmung	89
3.3	Farbdarstellung	92
3.3.1	Dreifarbentheorie	92
3.3.2	Farbsysteme	95
3.4	Farbnormalisierung	105
3.4.1	Modellbildung	105
3.4.2	Normalisierung bzgl. der Beleuchtungsgeometrie	106
3.4.3	Normalisierung bzgl. des Beleuchtungsspektrums	107
3.4.4	Normalisierung bzgl. beider Effekte	109
3.5	Versuchsteil	110
3.5.1	Versuchsaufbau	110
3.5.2	Grundlegende Einführung in Matlab	112
3.5.3	Theoretische Aufgaben	114
3.5.4	Aufgaben zur menschlichen Farbwahrnehmung	114
3.5.5	Aufgaben zur Farbdarstellung	115
3.5.6	Aufgaben zur Farbnormalisierung	116
4	Autofokus	119
4.1	Einleitung	119

4.2	Kontrastbasierte Verfahren	120
4.2.1	Ursachen der Unschärfe	120
4.2.2	Relative Verfahren	123
4.2.3	Absolute Verfahren	129
4.3	Triangulationsverfahren	129
4.3.1	Passive Verfahren	130
4.3.2	Aktive Verfahren	135
4.4	Weitere Verfahren	135
4.4.1	Mechanische Kopplung	136
4.4.2	Laufzeitmessungen	136
4.4.3	Bildlagemessung	136
4.5	Aufgaben	137
4.5.1	Vorbereitende Aufgaben	138
4.5.2	Versuchsaufgaben	140
5	Mustererkennung	145
5.1	Einleitung	145
5.2	Konturfindung	147
5.2.1	Konturverfolgung	148
5.2.2	Konturglättung	150
5.2.3	Kettenkodierung	151
5.3	Geometrische Merkmale	153
5.3.1	Grundlagen der lageinvarianten Mustererkennung	154
5.3.2	Umfang, Fläche und Kompaktheit	156
5.3.3	Momente	157
5.3.4	Polar-check	158
5.4	Graph-matching	159
5.4.1	Grundlagen der Darstellung eines Musters als Graph	160
5.4.2	Graph-Isomorphismus	162
5.4.3	Attributierung	163
5.5	Subgraph-matching	165
5.5.1	Subgraph-Isomorphismus und Assoziationsgraph	166
5.5.2	Koordinatentransformation	169
5.5.3	Berechnung	170
5.6	Aufgaben im Praktikum	171

Praktikumsversuch

Bildvorverarbeitung

R. Klaus, K. Peschke und H. Burkhardt

1 Bildvorverarbeitung

1.1 Einleitung

Die digitale Bildverarbeitung ist in den letzten Jahren zu einem eigenständigen und für die Anwendung in vielen Bereichen bedeutenden Gebiet der Informationsverarbeitung geworden. Die universellen Methoden finden beispielhafte Anwendungen im Bereich der Medizin, Biologie, Werkstückerkennung und Fernerkundung sowohl innerhalb der Industrie als auch der Wissenschaft. Die Abb.1.1 enthält eine schematische Vorgehensweise bei Problemen der Bildverarbeitung und Mustererkennung.

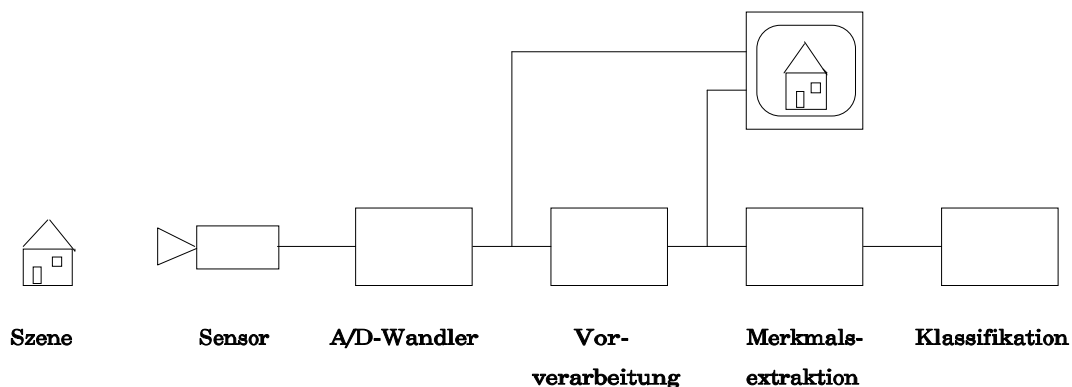


Abbildung 1.1: Allgemeines Schema zur Bildverarbeitung und Mustererkennung

Im vorliegenden Versuch beschäftigen wir uns mit Methoden der digitalen *Bildvorverarbeitung*. Sie dienen dazu, eine bereits digitalisierte Bildvorlage zu bearbeiten, um darin enthaltene Informationen für den menschlichen Betrachter besser visuell sichtbar zu

machen (Bildverbesserung) oder für die weiteren Verarbeitungsstufen der rechnergestützten Mustererkennung (Merkmalsextraktion und Klassifikation) aufzubereiten. Die Methoden der Bildvorverarbeitung bauen weitgehend auf lokale lineare und nichtlineare Filteroperationen auf. Es soll dafür eine einheitliche Darstellungsform gegeben werden, die sich programmtechnisch einfach umsetzen läßt.

Der Schwerpunkt des Praktikumversuches liegt auf den beiden Teilen Bildverbesserung durch Punktoperatoren und lokale Filterung. Auf den Teil Binarisierung und morphologische Bildverarbeitung wird nur kurz eingegangen, da dieser ausführlich in einem eigenen Versuch im Bildverarbeitungspraktikum II abgehandelt wird. Die Aufgaben im praktischen Teil wurden ursprünglich mit der Sprache APL2 bearbeitet. Seit dem WS 03/04 kommt stattdessen Matlab zum Einsatz.

1.2 Bildverbesserung durch Punktoperatoren und lokale Filterung

Die Aufgabe von *Bildverbesserungsverfahren* besteht darin, Bildsignale so zu bearbeiten, daß die für die bestehende Aufgabe entscheidende Information besser visuell oder maschinell extrahierbar wird, indem die dem Bild überlagerten Rauschanteile unterdrückt werden, bei gleichzeitiger Hervorhebung gewisser Bildnutzinformationen. Der in Kapitel 1.2.1 vorgestellte *Histogrammausgleich* versucht wenig kontrastreiche Bilder in kontrastreichere Bilder zu überführen. Die benutzte Operation bezieht sich auf isolierte Pixel und wird dementsprechend als *Punktoperation* bezeichnet. Kapitel 1.2.2 führt *Lokaloperatoren* ein, deren Aufgabe die schon angesprochene Rauschunterdrückung bei möglichst unveränderter Originalbildqualität ist. Dabei werden lineare gewichtete oder gar nur ungewichtete Mittelwerte einer durch das Lokaloperatorfenster festgelegten Umgebung gebildet. Wir erhalten gefilterte Bilder, die weniger Rauschanteile enthalten, deren feinstrukturierten Bildanteile aber mehr oder weniger verschwommen erscheinen. Die oben schon im Ortsbereich formulierte Wirkung einer linearen Mittelwertbildung läßt sich im Frequenzbereich als Tiefpaßfilterung beschreiben. Die hochfrequenten Rauschanteile und Bildsignalanteile werden reduziert, insgesamt aber verbessert sich in der Regel das Signal-/Rauschleistungsverhältnis.

Die in Kapitel 1.2.2 vorzustellenden Lokaloperatoren bilden gewichtete Mittelwerte aus der Lokaloperatorumgebung. Es war erwähnt worden, daß feine Bilddetails verschwommen erscheinen, also Kanten verschliffen werden. In Kapitel 1.2.3 werden wir eine nicht-

lineare Variante der Lokaloperatorfilterung kennenlernen, die wir im allgemeinen Fall als *Rangordnungsfiler*, im hier nur behandelten speziellen Fall als *Medianfilter* [52] bezeichnen. Die im Operatorfenster auftretenden Originalbildpixel werden der Größe nach sortiert und der mittlere Wert dieser geordneten Kette, der Median als Ausgangswert genommen. Analog zum Median könnten auch der größte oder kleinste Wert ausgewählt werden. Dementsprechend müßten wir das entstehende Rangordnungsfiler als Maxfiter oder Minfiter bezeichnen. Diese Art der Rauschunterdrückung eignet sich besonders für pulsformige Störungen. Feine Bilddetails bleiben dabei weitestgehend erhalten. Wir werden das Verhalten an entsprechender Stelle noch genauer analysieren.

1.2.1 Grauwerttransformationen

Grauwerttransformationen sind eine einfache, aber häufig sehr effektive Methode zur Vorverarbeitung von Bildern. Jeder Grauwert r eines Bildes wird mit Hilfe einer Transformation T in einen neuen Grauwert s übertragen, so dass gilt: $s = T(r)$. Diese Transformationen können als Punktoperatoren aufgefasst werden, die zur besseren Darstellung der Bilder am Bildschirm dienen. Die Art der Transformation kann zum Beispiel vom Benutzer vorgegeben werden oder unter Verwendung von Nebenbedingungen auch aus dem Bild selbst ermittelt werden.

Gammakorrektur

Viele Systeme besitzen physikalisch bedingt einen nichtlinearen Kennlinienverlauf zur Erfassung oder Wiedergabe von Bildern. Bei der Darstellung von Bildern an Bildschirmen mit Kathodenstrahlröhre führt diese Nichtlinearität dazu, dass die Bilder überwiegend zu dunkel dargestellt werden. Durch eine Kurve der Form $s = r^\gamma$ (Gamma-kurve) lässt sich die Kennlinie annähern und durch eine entsprechende Korrektur die verfälschte Darstellung beheben. Ähnliches gilt für die Ausgabe an Druckern, bei denen Bilder ohne Vorverarbeitung zu dunkel ausgedruckt werden. Eine Korrektur erfolgt hier jedoch meist automatisch. Allgemein kann eine Gammakorrektur dann angewendet werden, wenn dunkle Bereiche heller dargestellt werden sollen. Dies sieht man in der Abbildung 1.2. Die dunklen Bereiche im linken Bild sind nur schlecht sichtbar. Rechts daneben befindet sich das Bild bei dem die Grauwerte durch eine Gammakorrektur transformiert wurden. Die Darstellung lässt nun mehr Details erkennen.

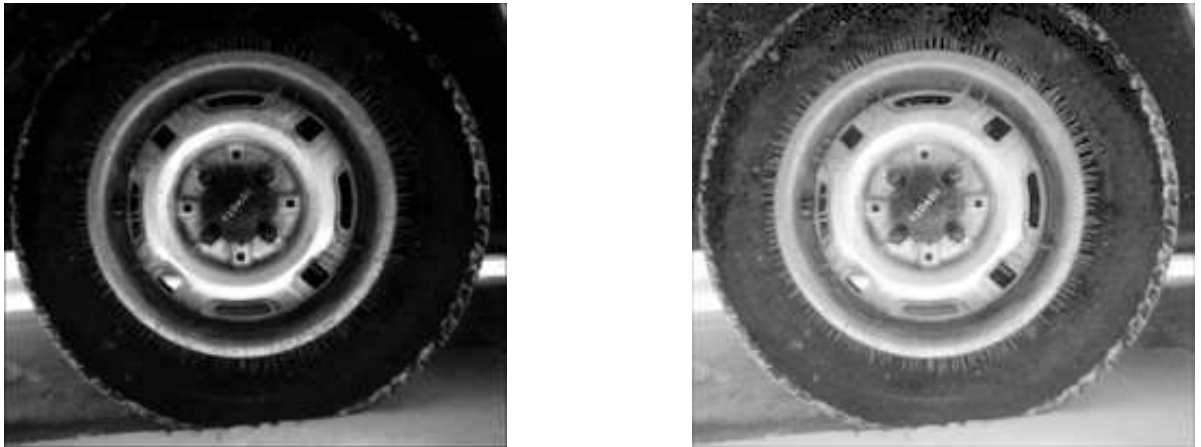


Abbildung 1.2: Bild ohne Gammakorrektur(links) und mit Gammakorrektur (rechts)

Histogrammbasierte Transformationen

Bilder lassen sich durch statistische Kenngrößen beschreiben. Dies hat den Vorteil, dass Verfahren und Methoden der Stochastik in der Bildverarbeitung angewendet werden können. Dazu wird die Intensität des Bildes als Zufallsvariable aufgefasst, deren unterschiedliche Werte (Grauwerte) jeweils von den verschiedenen Positionen im Bild abhängen. Eine globale Beschreibung des Bildes ist dann durch die Häufigkeitsverteilung (Histogramm) der Grauwerte $p_r(r_k)$ gegeben. Der Kurvenverlauf des Histogramms kann bei diskreten Werten ermittelt werden, indem man die Anzahl n_k der Pixel bestimmt, die einen bestimmten Grauwert r_k besitzen. Diese Zahl wird entsprechend dem jeweiligen Grauwert in einem Diagramm aufgetragen. Die folgende Formel beschreibt diese Beziehung, wobei N die gesamte Anzahl der Pixel des Bildes ist.

$$p_r(r_k) = \frac{n_k}{N} \quad (1.1)$$

Während eine Zufallsvariable auch kontinuierliche Werte annehmen kann, handelt es sich in der Bildverarbeitung immer um diskrete Werte. Formeln können jedoch analog hergeleitet werden. So gilt für die kumulierte Verteilungsfunktion (cdf) im kontinuierlichen Fall:

$$P_r(r) = \int_{r_{\min}}^{r_{\max}} p_r(w)dw \quad (1.2)$$

Während diese Funktion im Diskreten in Summenschreibweise angegeben wird und als kumuliertes Histogramm bezeichnet wird.

$$P_r(r_{\max}) = \sum_{j=r_{\min}}^{r_{\max}} p_r(r_j) \quad (1.3)$$

In der Abbildung 1.3 sind ein Bild und dessen Histogramm dargestellt.

Histogrammausgleich

Der Kurvenverlauf des Histogramms macht Eigenschaften des Bildes deutlich, wie im Bild 1.3 dargestellt ist. Im dargestellten Histogramm liegen die meisten Grauwerte in einem schmalen Bereich, wodurch das Bild kontrastarm wird. Umgekehrt kann man sagen, dass bei Bildern für die sich die Häufigkeit der Grauwerte gleichmäßig auf den gesamten Grauwertbereich verteilt, der Kontrast größer ist. Ziel ist deshalb eine Transformation zu finden, die das Histogramm eines kontrastarmen Bildes in ein Histogramm mit (annähernd) gleichverteilten Werten überführt. Diese Transformation wird Histogrammausgleich genannt und lässt sich über die Transformation von Zufallsvariablen [23] herleiten.

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| \quad (1.4)$$

mit

$$s_k = T(r_k) \quad (1.5)$$

Dabei wird vorausgesetzt, dass die Transformation T monoton wachsend ist, so dass Grauwerte nicht invertiert werden. Außerdem soll eine Rücktransformation stets möglich sein, das heißt die Transformation soll immer eine Inverse besitzen. Weiterhin soll der Dynamikbereich des Bildes nach der Transformation gleich dem Ursprungsbild sein und wird hier zur Herleitung im Bereich $[0, 1]$ angenommen.

Für das ausgeglichene Histogramm des Ergebnisbildes soll gelten:

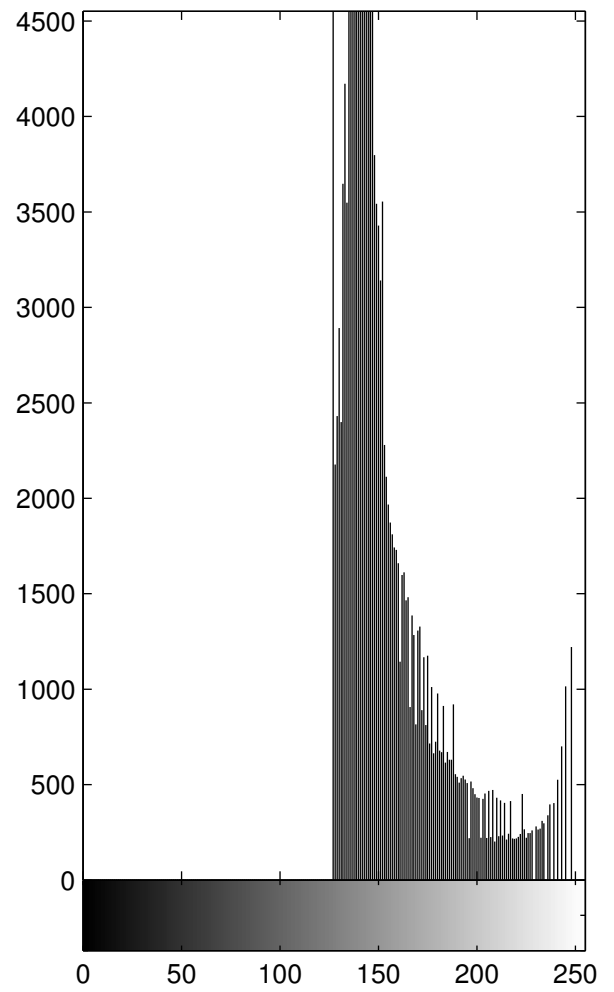
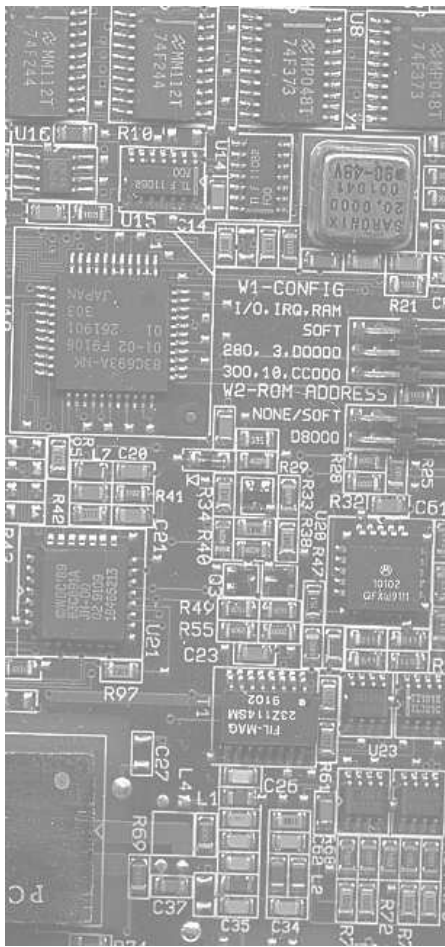


Abbildung 1.3: Histogramm eines Bildes

$$p_s(s) = \text{const} \quad (1.6)$$

Fasst man die Formeln 1.4 und 1.6 zusammen, so lässt sich schreiben:

$$\int_0^g 1 ds = \int_0^k p_r(r) dr \quad (1.7)$$

$$g = \int_0^k p_r(r) dr \quad (1.8)$$

Wobei wie oben erwähnt im Diskreten die rechte Seite das kumulierte Histogramm darstellt.

$$g = \sum_{j=0}^k p_r(r_j) dr \quad (1.9)$$

Wendet man diesen Histogrammausgleich auf das Histogramm des obigen Bildes an, so ergibt sich ein Histogramm wie es im Bild 1.4 zu sehen ist. Aufgrund der Diskretisierung der Grauwerte, ist der Verlauf des Ergebnishistogramms nicht gleichmäßig konstant.

Neben diesem globalen Histogrammausgleich lässt sich ein Ausgleich auch lokal in einer bestimmten Nachbarschaft erreichen. Dazu wird das Bild in Bereiche unterteilt, auf denen jeweils getrennt ein Histogrammausgleich durchgeführt wird. Dieses Verfahren wird als adaptiver Histogrammausgleich bezeichnet und ermöglicht eine noch kontrastreichere Darstellung.

Histogrammspreizung

Neben dem Histogrammausgleich lässt sich eine kontrastreichere Darstellung durch eine Histogrammspreizung erreichen. Dabei wird aus dem Histogramm des Ursprungsbildes ein Bereich ausgeschnitten und auf einen größeren Bereich übertragen, indem die Abstände zwischen den einzelnen Grauwerten vergrößert werden. Die Auswahl des Bereiches, sowie die Größe des Zielbereiches werden durch den Anwender bestimmt, um eine subjektiv bessere Darstellung zu bekommen.

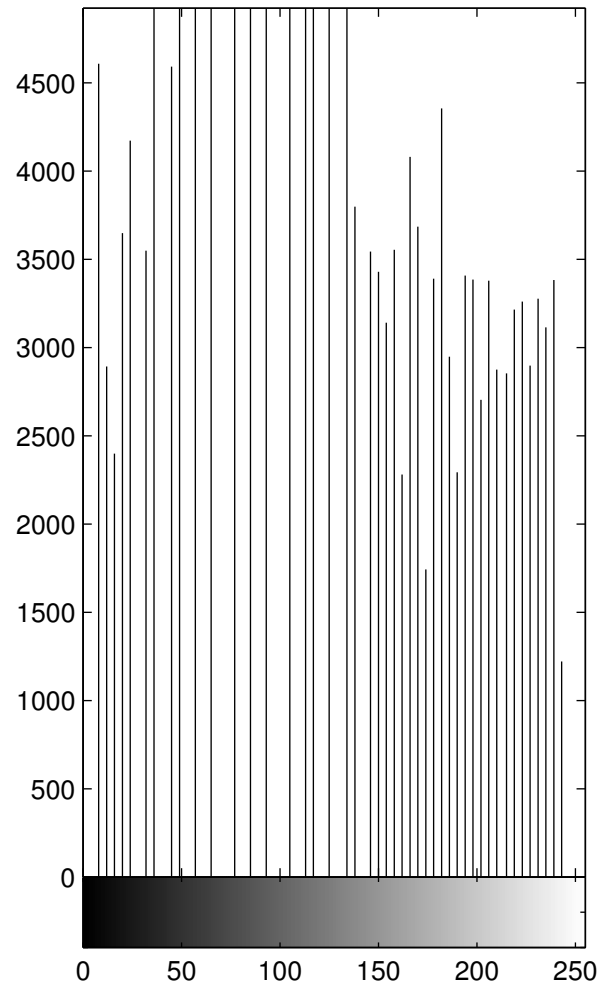
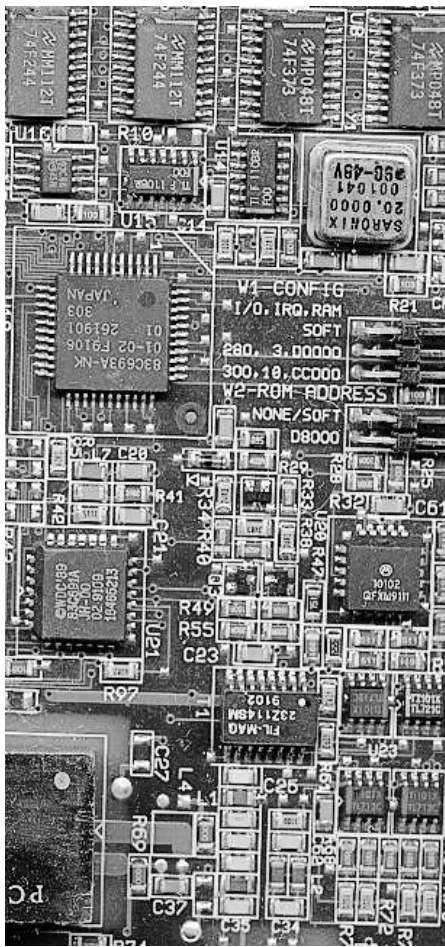


Abbildung 1.4: ausgeglichenes Histogramm eines Bildes

1.2.2 Lineare Lokaloperatoren

In der Einleitung zu diesem Kapitel waren schon einige Gedanken zur Motivation der linearen Lokaloperatoren angesprochen worden. Wir haben ein Bild vorliegen, das durch Rauschüberlagerung gestört ist. Die Rauschteile benachbarter Pixel sind bei weißen Rauschprozessen unkorreliert, die Signalanteile hingegen in der Regel stark korreliert. Summieren und skalieren wir folglich die gestörten Pixel in einer lokalen Umgebung auf, so heben sich die Störanteile zu einem gewissen Maße auf, die Signalanteile aber bleiben wenig verändert. Diese heuristische Überlegung bildet die wesentliche Grundlage der linearen Filterung mit Lokaloperatoren.

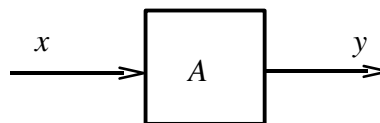


Abbildung 1.5: Prinzipskizze eines linearen Systems

Die Eigenschaft der Linearität von Filtern lässt sich aus der Systemtheorie, genauer gesagt dem Durchgang von Signalen durch Systeme, übertragen. Liegt an einem linearen System A das Eingangssignal x an, so lässt sich am Ausgang des Systems das Signal y abgreifen. Dieser Vorgang ist schematisch in der Abbildung 1.5 dargestellt. Geht man dabei von der Annahme aus, dass \mathcal{X} und \mathcal{Y} endlichdimensionale Vektorräume sind, wobei der Originalbereich den Raum der Eingangssignale $x \in \mathcal{X}$ und der Bildbereich den Raum der Ausgangssignale $y \in \mathcal{Y}$ umfasst, so kann man diese Beziehung mathematisch sehr anschaulich wie folgt formulieren:

$$y = Ax \tag{1.10}$$

Wesentlich ist dabei, dass bei einem linearen System, für zwei unterschiedliche Eingangssignale x_1 und x_2 sowohl das Superpositionsprinzip, als auch das Verstärkungsprinzip erfüllt sein [6] müssen.

Definition 1.1 (Superpositionsprinzip)

$$A(x_1 + x_2) = Ax_1 + Ax_2 \tag{1.11}$$

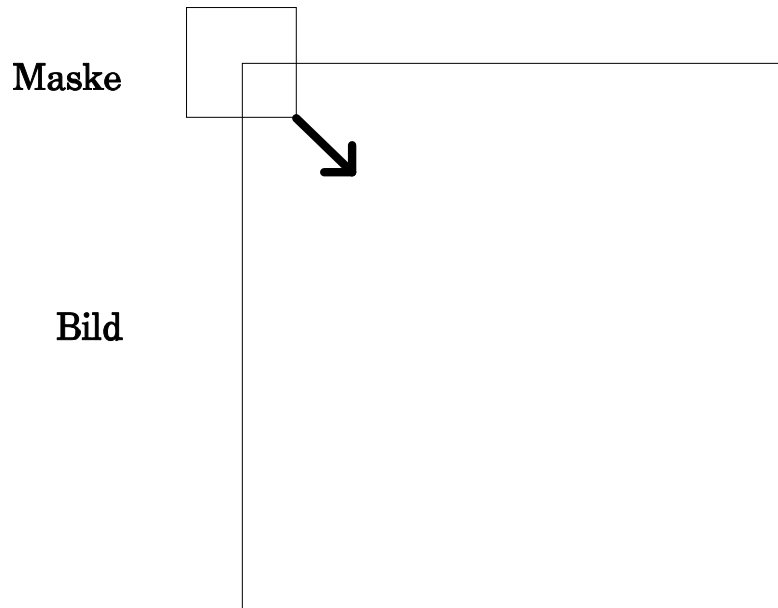


Abbildung 1.6: allgemeine Lokaloperation

Definition 1.2 (Verstärkungsprinzip)

$$A(\alpha \cdot x_1) = \alpha \cdot Ax_1 \quad (1.12)$$

Ein Vorteil linearer Systeme liegt in der soliden mathematischen Beschreibbarkeit durch operatorentheoretische Begriffe der linearen Algebra. Auf diese Art lassen sich unterschiedliche Systeme übergeordneten Modellen zuordnen und gemeinsam beschreiben. Filter, die den obigen Prinzipien entsprechen, werden somit als lineare Filter bezeichnet.

Bevor wir weiter auf die lineare Filterung eingehen wollen, soll die in der Einleitung bereits angesprochene *allgemeine Lokaloperation* definiert werden. Die gewählte sehr allgemeine Notation gestattet es, alle weiteren speziellen Lokaloperationen in diesem Skript auf diese Definition zurückzuführen. Die Abb.1.6 soll diesen Vorgang verdeutlichen.

Eine Maske H wird über das Bild X geschoben und die entsprechenden Elementen aus H und X werden einem Innenprodukt unterzogen. Im Fall der linearen Filterung

treten dann als Operatoren die Addition und Multiplikation auf. Die Form der Maske H ist typischerweise quadratisch, kann aber auch rechteckig, kreuzförmig oder rund sein. Bleiben die Maskenelemente während der Verschiebung unverändert, so handelt es sich um eine lineare ortsinvariante Filterung, die man auch als *Faltung* bezeichnet. Bleibt die Stoßantwort auf endliche Dimension beschränkt, so spricht man von einem FIR-Filter (*finite impulse response*). Die Maskenelemente selbst beinhalten die Elemente der Stoßantwort eines entsprechenden Übertragungssystems in punktgespiegelter Form. Dies kommt in der Definition 1.3 zum Ausdruck, die die eigentliche Summendarstellung und eine symbolische Darstellung mit dem Faltungsoperator angibt.

Definition 1.3 (Faltung)

$$y(m,n) = \frac{1}{MN} \sum_M \sum_N h(m-m', n-n') x(m', n') \quad (1.13)$$

$$Y = H \overset{(+, \times)}{*} X \quad (1.14)$$

Eine entsprechende Formulierung für die Korrelation sowohl in Summendarstellung als auch mit dem Korrelationsoperator gibt die Definition 1.4 an.

Definition 1.4 (Korrelation)

$$y(m,n) = \frac{1}{MN} \sum_M \sum_N h(m+m', n+n') x(m', n') \quad (1.15)$$

$$Y = H \overset{(+, \times)}{\#} X \quad (1.16)$$

Die Operationen in den folgenden Kapiteln werden ausschließlich von der Korrelation Gebrauch machen. Dabei werden lediglich die Operatoren im Innenprodukt geeignet ausgetauscht. Symbolisch können Faltung und Korrelation auch folgendermaßen dargestellt werden:

$$Y = H(+\text{LOK}\times)X \quad (1.17)$$

Dabei wird ersichtlich, daß eine Korrelation mit punktgespiegelter Maske der Faltung entspricht.

An dieser Stelle soll auf eine wichtige Eigenschaft der Fourier Transformation (FT) in Verbindung mit linearen Operatoren hingewiesen werden. Wie in [6] und [53] gezeigt wird, gelten für die Faltung eines Bildes X mit dem Operator H folgende Zusammenhänge:

Definition 1.5 (Isomorphie der FT)

$$Y = H \overset{(+, \times)}{*} X \tag{1.18}$$



$$\tilde{Y} = \tilde{H} \cdot \tilde{X} \tag{1.19}$$

Wobei \tilde{Y} , \tilde{H} und \tilde{X} die Fouriertransformierten des gefilterten Bildes, des Lokaloperators und des Ausgangsbildes sind. Ähnliches gilt für die Korrelation eines Bildes. Anschaulich heißt das, dass man sämtliche Operationen, die auf einer Faltungen beruhen, auch im Fourierraum berechnen kann. Aus der sehr rechenaufwendigen Faltung wird dann aufgrund der Isomorphie der FT eine einfache Multiplikation. Dies wird außerdem dadurch begünstigt, dass mit der schnellen Fourier Transformation (FFT) eine Möglichkeit gegeben ist, mit vergleichbar geringem Rechenaufwand die Transformation, sowie die Rücktransformation durchzuführen. Es gilt also:

$$Y = FFT^{-1}(FFT(H) \cdot FFT(X)) \tag{1.20}$$

Die Größe der Maske ist eine Frage des Aufwands. Größere Masken gestatten es, den überlagerten Störanteil besser zu unterdrücken. Gefragt ist demnach ein für den gerade vorliegenden Fall geeigneter Kompromiß. In Beispiel 1.1 werden drei gebräuchliche lineare 3×3 -Lokaloperatoren angegeben [42].

Beispiel 1.1 (3×3 -Lokaloperatoren)

$$\mathbf{LOK1} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{LOK2} = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{LOK3} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Alle drei Operatoren sind normalisiert, um im gefilterten Bild den gleichen Dynamikbereich wie im Originalbild zu erhalten. Faltung und Korrelation können auch nach zweidimensionaler diskreter Fouriertransformation im Ortsfrequenzbereich charakterisiert werden. Wir wollen aber an dieser Stelle nicht weiter darauf eingehen.

1.2.3 Medianfilter

Alle bisher besprochenen Bildfilter hatten zumindest lineare Eigenschaften. Als zusätzliche Bedingung kam noch die Ortsinvarianz hinzu. Ortsinvariante lineare zweidimensionale Filter lassen sich mit den üblichen Methoden der Signal- und Systemtheorie beschreiben. Unter dem Aspekt der Filterung verrauschter Bilder kommt es jedoch entscheidend auf die Eigenschaften des Rauschprozesses an, um die Wirksamkeit des Filterentwurfs zu beurteilen. Die weithin übliche Annahme geht von weißem gaußschen Rauschen aus. Dieses Vorgehen deckt jedoch nicht alle in der Praxis vorkommenden Fälle ab. Eine weitere wichtige Sonderform geht von sogenanntem *salt and pepper noise* aus. Damit sind auf Pixelausmaße beschränkte impulsartige Störungen gemeint. Diese Störungen lassen sich weit besser mit einer nichtlinearen Lokaloperation bekannt unter dem Namen *Medianfilter* beseitigen.

Die linearen Filter nehmen beträchtlichen Einfluß auf den determinierten Signalanteil. Diese unerwünschte Eigenschaft führte zu verschwommenen Bilddetails. Das Medianfilter vermeidet diesen Nachteil weitgehend. Ein Operatorfenster selektiert p Elemente aus dem Originalbild. Die Größe p sei dabei einfachheit halber ganzzahlig ungerade. Alle p Werte werden der Größe nach sortiert. Das resultierende Ausgangsbildelement ist dann jeweils der $(p+1)/2$ - größte Wert dieser Reihe, der Median. Einfache Varianten dieses Verfahrens sind *Min-* und *Max-Filter*, die anstelle des Median den minimalen oder maximalen Wert auswählen. Allgemein werden all diese Verfahren unter dem Begriff *Rangordnungsfiler* zusammengefaßt.

Das Medianfilter beseitigt alle Strukturen vollständig, deren Ausdehnung innerhalb des Operatorfensters nicht mehr als $(p-1)/2$ Bildpunkte betragen. Das Verhalten der nicht-linearen Medianfilteroperation läßt sich mit den Methoden der linearen Systemtheorie nicht beschreiben. Hier sind spezielle deterministische und stochastische Methoden notwendig. Im weiteren Verlauf werden wir auf den relativ einfachen Fall der *Fixpunkte* bei eindimensionalen Medianfiltern eingehen. Zunächst wollen wir aber eine Definition des Medianfilters angeben. Sie zeigt die uns schon bekannte Notation für die Korrelation, bei der aber die Operationen für das Innenprodukt durch *take* und *med* ersetzt wurden. Die Originalbildpixel unter der Maske H werden selektiert (*take*) und der Medianwert bestimmt (*med*).

Definition 1.6 (Medianfilter)

$$G = M \stackrel{(med,take)}{\#} B \quad (1.21)$$

Die Gleichung 1.22 zeigt die eigentliche Medianbildung.

$$x(i) = med(x(i-v), \dots, x(i), \dots, x(i+v)) \quad (1.22)$$

Die Pixel $x(i-v), \dots, x(i), \dots, x(i+v)$ seien bereits der Größe nach sortiert und entstammen einem ein- oder zweidimensionalen Operatorfenster. Ein einfaches eindimensionales Beispiel soll den Vorgang verdeutlichen. Dabei bedeutet $median_3$ die laufende Medianbildung mit einem Fenster der Größe 3.

Beispiel 1.2 (eindimensionales Medianfilter)

$$med_3(2, 10, 2, 2, 2, 10, 10, 1, 10) = (2, 2, 2, 2, 10, 10, 10)$$

An Beispiel 1.2 sehen wir die zwei wesentlichen Eigenschaften eindimensionaler Medianfilter, die auch durch Abb.1.7 verdeutlicht werden:

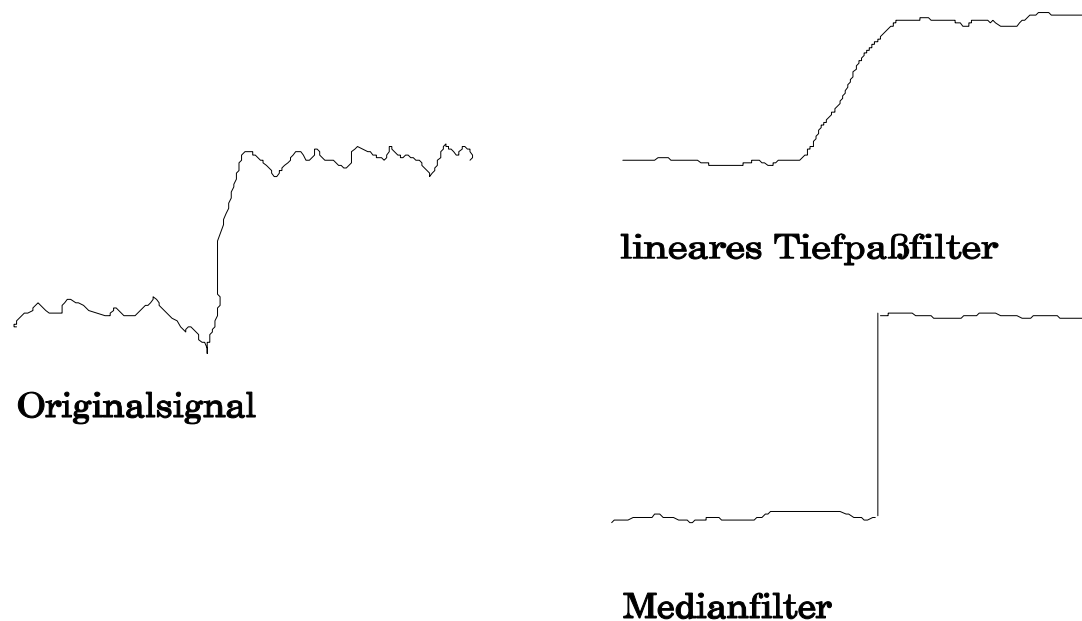


Abbildung 1.7: Vergleich von linearem Tiefpaßfilter und Medianfilter

- Medianfilter bewahren scharfe Kanten, lineare Tiefpaßfilter hingegen verschleifen Kanten
- Medianfilter können sehr effektiv Impulsstörungen beseitigen

Die erste der genannten Eigenschaften soll hier anhand des eindimensionalen Falls näher erläutert werden. Wir suchen nach einer hinreichenden Bedingung für eine Zahlensequenz, damit sie von einer Medianfilterung unbeeinflusst bleibt. Anders ausgedrückt suchen wir nach *Fixpunkten* (root signals) des Medianfilters. Dadurch erhalten wir einen ersten Eindruck der grundlegenden determinierten Eigenschaften besagter Operatoren. Der Fixpunkt einer beliebigen Transformation C ist definiert als:

Definition 1.7 (Fixpunkt) $C(x)=x$

Eine große Kategorie an Fixpunkten besteht aus verallgemeinerten monotonen Sequenzen. Wir wollen zunächst Monotonie für eine beidseitig unendliche Sequenz annehmen. Dann ergibt sich sofort die Eigenschaft 1.1.

Eigenschaft 1.1 (Median einer beidseitig unendlichen monotonen Sequenz)

$$x(-k) \leq \dots \leq x(0) \leq \dots \leq x(+k) \Rightarrow \text{med}(x(-k), \dots, x(0), \dots, x(+k)) = x(0) \quad (1.23)$$

Die obige Beziehung gilt für beliebig große Lokaloperatorfenster. Eine Sprungfunktion ist eine monotone Sequenz und somit invariant gegenüber Medianfiltern. Monotone Sequenzen sind nur die einfachsten Fixpunkte. Wir werden eine davon ausgehende Verallgemeinerung betrachten.

Die geforderte Monotonie ist eine nicht notwendige Restriktion, denn wir haben nicht beliebig große Operatorfenster sondern solche von fester und endlicher Größe. Von dieser Überlegung ausgehend definieren wir eine *lokal monotone Sequenz* der Länge m [52].

Definition 1.8 (lokal monotone Sequenz) Eine Sequenz $\{x\}$ ist lokal monoton der Länge m ($\text{lomo}(m)$), wenn $(x(n), \dots, x(n+m-1))$ ist monoton für alle n .

Daraus läßt sich leicht ableiten, daß eine Sequenz auch dann $\text{lomo}(m)$ ist, falls bei einem Trendwechsel von steigendem zu fallendem Verhalten oder umgekehrt die betrachtete Sequenz für mindestens $m-1$ Werte konstant bleibt. Dazu betrachten wir ein Beispiel.

Beispiel 1.3 (lokale Monotonie bei Trendwechsel)

$$(0, 1, 3, 3, 4, 4, 3, 2, 2, 1) \rightarrow \begin{pmatrix} 0, 1, 3 \\ 1, 3, 3 \\ 3, 3, 4 \\ 3, 4, 4 \\ 4, 4, 3 \\ 4, 3, 2 \\ 3, 2, 2 \\ 2, 2, 1 \end{pmatrix}$$

Das Beispiel zeigt, daß die Sequenz $lomo(3)$ ist, weil sie für 2 aufeinanderfolgende Werte (in diesem Fall die Zahl 4) konstant ist. Alle 8 Teilsequenzen auf der rechten Seite sind demnach für sich genommen monoton.

Die strenge Bedingung der Monotonie einer Sequenz schlechthin ergab einen Fixpunkt eines Medianfilters beliebiger Fensterlänge. Die lokale Monotonie der Länge m ergibt einen Fixpunkt eines Medianfilters nur bis hin zu einer bestimmten Fensterlänge. Dieses Verhalten beschreibt das folgende Theorem [52].

Theorem 1.1 *Eine $lomo(m)$ Sequenz ist Fixpunkt eines Medianfilters bis hin zur Fensterlänge $2m - 3$.*

Wir betrachten dazu erneut die Sequenz aus Beispiel 1.3. Die Originalsequenz war $lomo(3)$, also $m = 3$. Die maximale Fensterlänge ist demnach $2m - 3 = 3$. Für die entsprechende mediangefilterte Sequenz ergibt sich ein Ergebnis entsprechend Beispiel 1.4.

Beispiel 1.4 (mediangefilterte Testsequenz aus Bsp 1.3)

$$med_3(0, 1, 3, 3, 4, 4, 3, 2, 2, 1) = (1, 3, 3, 4, 4, 3, 2, 2)$$

Aus der ursprünglichen Sequenz der Länge 10 wird bei der Lokaloperatorlänge von 3 eine Ausgangssequenz der Länge 8. Das erste und letzte Element geht in diesem Fall verloren. Der eigentliche Kern der Sequenz bleibt aber unverändert, ist demnach Kern des Medianfilters der Länge 3. Zu Theorem 1.1 existiert eine Umkehrung die an dieser Stelle aber nicht besprochen werden soll [52].

Die Fixpunktanalyse zweidimensionaler Medianfilter ist aufgrund des größeren Freiheitsgrads bei der Wahl von Operatorfenstern schwieriger. Im eindimensionalen Fall wird bei jeder Fensterbewegung nur ein Bildpunkt fallengelassen und dafür nur ein neuer Bildpunkt hinzugenommen. Auf der anderen Seite werden bei zweidimensionalen Fenstern bei jeder Verschiebung mehr als ein Pixel ausgetauscht. Intuitiv ist leicht einzusehen, daß der Freiheitsgrad gewaltig anwachsen kann. Die Größe und Form des

Operatorfensters spielt dabei eine entscheidende Rolle. Eine analytische und anschaulich einfache Behandlung zweidimensionaler Fixpunkte ist an dieser Stelle nicht möglich. Es werden deshalb Fensterformen definiert und deren Fixpunktverhalten anhand einfacher Binärbildstrukturen erläutert.

Zur Erläuterung des generellen Verhaltens sind zwei Fensterformen besonders interessant. Dabei handelt es sich um ein Quadrat und ein Kreuz mit senkrecht und waagrecht verlaufenden Balken. Damit sollen drei charakteristische Modellsignale gefiltert werden: dünne Linien, Kanten und Ecken. Beispiel 1.5 zeigt die Quellsignale in der linken Spalte, die mit dem 3×3 -Quadratfenster erhaltenen Filterergebnisse in der mittleren Spalte und die mit dem 3×3 -Kreuzfenster rechts.

Beispiel 1.5 (Verhalten zweidimensionaler Medianfilter)

0 0 1 0 0	0 0 0 0 0	0 0 1 0 0
0 0 1 0 0	0 0 0 0 0	0 0 1 0 0
0 0 1 0 0	0 0 0 0 0	0 0 1 0 0
0 0 1 0 0	0 0 0 0 0	0 0 1 0 0
0 0 1 1 1	0 0 1 1 1	0 0 1 1 1
0 0 1 1 1	0 0 1 1 1	0 0 1 1 1
0 0 1 1 1	0 0 1 1 1	0 0 1 1 1
0 0 1 1 1	0 0 1 1 1	0 0 1 1 1
0 0 1 1 1	0 0 1 1 1	0 0 1 1 1
0 0 1 1 1	0 0 0 1 1	0 0 1 1 1
0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
0 0 0 0 0	0 0 0 0 0	0 0 0 0 0

Dem obigen Beispiel ist leicht das unterschiedliche Verhalten verschiedener Fensterfunktionen zu entnehmen. Bei quadratischer Form gehen dünne Linien verloren und Ecken werden verschliffen. Bei der senkrechten und waagerechten Kreuzform hingegen bleiben alle drei Originalbilder unverfälscht. Bei beiden Fensterformen ist es unerheblich ob die dünnen Linien senkrecht oder waagrecht verlaufen.

Neben der Fensterform ist der notwendige Sortiervorgang der zweite wichtige Punkt. Wir wollen an dieser Stelle nicht auf spezielle Sortierverfahren eingehen. Ein Aspekt, der zu schnellen Algorithmen führen kann soll aber kurz erwähnt werden. Da das Fenster

jeweils nur um eine Pixelposition verschoben wird, müssen die selektierten Zahlen nicht jedesmal komplett neu sortiert werden. Die ausscheidenden Pixel werden entfernt und die dafür neu hinzukommenden Pixel gleich einsortiert. Eine komplette Neusortierung ist dann nur zu Beginn einer neuen Zeile notwendig.

1.3 Kantenextraktion

Die Kanten von Objekten sind ein für die rechnergestützte Bildanalyse, aber auch für das Auge des menschlichen Betrachters wichtiges Attribut. Die Definition des Problems Kantenextraktion und deren Lösung ist weder trivial noch eindeutig und hat eine ganze Reihe von Extraktionsverfahren hervorgebracht, die jedoch in der Regel applikations-spezifische Ergebnisse liefern. Bevor wir den inhaltlichen Teil des gesamten Kapitels abstecken, soll zunächst eine terminologische Einordnung vorgenommen werden.

Die *Kantenextraktion* kann sowohl als ein Teil der *Bildverbesserung* (image enhancement) als auch der *Bildsegmentierung* (image segmentation) angesehen werden [44, 39]. Unter dem Aspekt der Bildverbesserung dient sie als Verfahren zur Verschärfung (sharpening) von Bildobjekten, zunächst gleichgültig ob sie zur Aufbereitung für den Mensch oder Computer gedacht ist. Das Gegenstück dazu bildet die Störunterdrückung (smoothing).

Aus der Sicht der Bildsegmentierung dient die Kantenextraktion als erster Schritt zur eigentlichen Konturdetektion von Objekten. Die weiteren Schritte bestehen aus einer Schwellwertbildung, die entscheidet ob eine Kante vorliegt, und einem Konturverfolgungsalgorithmus, der zu einer für Objekte notwendigen geschlossenen Konturlinie führt. Die zweite Gruppe der Segmentierung sind bereichsorientierte Verfahren, die auf Schwellwertbildung oder Bereichswachstum basieren. Dabei wird anhand eines Einheitlichkeitskriteriums entschieden welche Bildflächen einem Objekt zuzuschreiben sind. Die bereichsorientierten Verfahren werden hier nicht behandelt.

Alle kantenorientierten Verfahren versuchen die Trennungslinien zwischen Bildbereichen mit unterschiedlichen Eigenschaften zu berechnen, indem ein Eigenschaftsgradient für jeden Bildpunkt bestimmt wird [53]. Wir wollen uns hier auf Graubilder beschränken. Die zu betrachtende Eigenschaft ist dann einfacherweise der Grauwertpegel.

In Kapitel 1.3.1 werden zunächst einfache Gradientenoperatoren behandelt. Wir werden sehen wie aus der kontinuierlichen Definition des zweidimensionalen Gradienten mit Betrag und Richtung einfache diskrete Lokaloperatoren abgeleitet werden. Kapi-

tel 1.3.2 wird den entsprechenden Laplaceoperator behandeln. In Kapitel 1.3.3 werden Differenzen von ungewichteten und gewichteten Mittelwerten (Prewitt- und Sobeloperator) zur Konturextraktion herangezogen.

Alle drei bisherigen Verfahren gehen von der alleinigen Vorstellung aus, Kantenverläufe durch Differenzbildungen zu verstärken. Das genaue Profil der zu detektierenden Kante, die Art der Störung, die Detektionsempfindlichkeit oder die Lokalisierungsgenauigkeit gehen entweder überhaupt nicht oder nur sehr vage in die Herleitung eines Operators ein. Canny schlug deshalb ein Verfahren vor, das aus der Optimierung mehrerer der genannten Größen hervorgeht. Wir werden uns hier auf sprungförmige Kantenverläufe beschränken und eine Approximation an die Optimallösung angeben.

1.3.1 Gradientenoperator

Der *Gradient* einer kontinuierlichen, differenzierbaren, zweidimensionalen Funktion $f(m, n)$ ist definiert als:

Definition 1.9

$$\Delta f(m, n) = \frac{d}{dm}[f(m, n)]e_m + \frac{d}{dn}[f(m, n)]e_n \quad (1.24)$$

m und n sind die kontinuierlichen Ortskoordinaten und e_m bzw. e_n die entsprechenden Einheitsvektoren. Wir erhalten einen Vektor bestehend aus Betrag und Richtung, der die maximale erste Ableitung und deren Richtung in der mn -Ebene angibt. Ausgehend von dieser Definition suchen wir nun Approximationen, die die Ableitung der kontinuierlichen Funktion $f(m, n)$ durch lokale Differenzbildungen der entsprechenden ortsdiskreten Funktion angeben. Dabei werden die Pixel des Quellbildes innerhalb eines Lokalfensters gewichtet und dann zur Gradientenbildung herangezogen. Das Ergebnis wird, abhängig von der gewählten Operation einem bestimmten Koordinatenpunkt des Lokalfensters zugeordnet. Insofern handelt es sich um eine Operation, die pixelweise das Ergebnisbild erzeugt.

Wählen wir zunächst das kleinstmögliche Fenster der Größe 2×2 , so gibt es eine einfache Möglichkeit der Verknüpfung von Bildpixeln für Δ_m und Δ_n , die *linksseitigen Differenzen*.

Definition 1.10

$$\Delta_m[f(m,n)] = f(m,n) - f(m-1,n) \quad (1.25)$$

$$\Delta_n[f(m,n)] = f(m,n) - f(m,n-1) \quad (1.26)$$

$$\mathbf{GRAD1M} = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix} \quad \mathbf{GRAD1N} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}$$

Wir sehen zunächst, daß das Pixel $f(m-1, n-1)$ unseres Operatorfensters nicht zum Tragen kommt. Bei beiden Differenzen erhalten wir ein Ergebnis, das den Koordinatenpunkt (m,n) und jeweils einen Nachbarn umfaßt, letztendlich aber nicht einem dazwischenliegenden Punkt, sondern wieder (m,n) zugeordnet wird. Das Ergebnis Δ_m müßte eigentlich auf einen Gitterpunkt $(m-0.5, n)$ führen, Δ_n hingegen auf $(m, n-0.5)$ aber nicht wie oben definiert beide wieder auf (m,n) . Für die Gradientenbilder Δ_m bzw. Δ_n für sich genommen erhalten wir um 0.5 Koordinatenpunkte in m - bzw. n -Richtung verschobene Ergebnisse. Dieser Effekt stört zunächst nicht, wird aber bei der Zusammenführung von Δ_m und Δ_n zu einem Betragsgradienten als fehlerhaft empfunden. Gleichgültig wie die Betragsbildung aussieht, es werden nicht deckungsgleiche Gradientenbilder miteinander verknüpft.

Um diesen Effekt zu vermeiden kann man einen Differenzenoperator in ebenfalls horizontaler und vertikaler Richtung, aber in mit einer 3×3 -Maske über den Punkt (m,n) hinweg definieren, der die *zentralen Differenzen* enthält.

Definition 1.11

$$\Delta_{2m}[f(m,n)] = f(m+1,n) - f(m-1,n) \quad (1.27)$$

$$\Delta_{2n}[f(m,n)] = f(m,n+1) - f(m,n-1) \quad (1.28)$$

$$\mathbf{GRAD2M} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{GRAD2N} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Dieser Operator beinhaltet nicht den Quellpunkt (m, n) . Er umfaßt im Gegensatz zur ersten Definition in jeder Richtung eine Distanz von zwei Einheiten. Um amplitudennormierte Gradientenbilder zu erhalten, müßten wir beim obigen Operator durch 2 teilen. Die Antwort der Operatoren Δ_{2m} bzw. Δ_{2n} auf eine ideale Sprungkante hat außerdem eine Strichbreite von zwei Pixeln. Die Lokalisierungsgenauigkeit ist deshalb eingeschränkt.

Eine dritte Möglichkeit einen Gradientenoperator anzugeben stammt von Roberts und ist als *Roberts-Cross-Operator* bekannt.

Definition 1.12 (Roberts-Cross-Operator)

$$\Delta_+[f(m, n)] = f(m + 1, n + 1) - f(m, n) \quad (1.29)$$

$$\Delta_-[f(m, n)] = f(m, n + 1) - f(m + 1, n) \quad (1.30)$$

$$\mathbf{RCOP} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \mathbf{RCO} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Der Operator bildet die beiden *diagonalen Differenzen*, die aber im Gegensatz zur ersten Definition dem gleichen Koordinatenpunkt $(m + 0.5, n + 0.5)$ zugeordnet werden können.

Alle drei bisher definierten Gradientenoperatoren liefern zunächst zwei getrennte Richtungsgradienten. Bei den beiden ersten Operatoren zeigten sie in horizontaler und vertikaler Richtung, beim Roberts-Cross-Operator in die beiden diagonalen Richtungen. In der Regel ist man jedoch an einem Gradientenbild schlechthin unabhängig von einer Vorzugsrichtung interessiert. Aus den zwei Komponenten ist deshalb ein Betrag zu ermitteln, der der größten örtlichen Differenz unabhängig von der Richtung gleichkommt. Die Richtungsunabhängigkeit selbst wird ein wichtiges Beurteilungskriterium sein.

Ein Betragsgradient kann auf unterschiedliche Art und Weise gebildet werden [53, 18]. Ausgehend von der Definition im kontinuierlichen Fall kann die Wurzel aus der Quadratsumme gebildet werden.

$$\Delta_g = \sqrt{\Delta_1^2 + \Delta_2^2} \quad (1.31)$$

Eine zweite weit weniger rechenintensive Verknüpfung bildet die Summe der Beträge.

$$\Delta_b = |\Delta_1| + |\Delta_2| \quad (1.32)$$

Die dritte Methode nimmt das Maximum der beiden Beträge.

$$\Delta_m = \max\{|\Delta_1|, |\Delta_2|\} \quad (1.33)$$

Im Versuchsteil dieses Praktikums wird die Wirksamkeit der obigen Operatoren anhand synthetischer und natürlicher Bildvorlagen untersucht werden.

1.3.2 Laplaceoperator

Im vorherigen Kapitel hatten wir diskrete Lokaloperatoren basierend auf der Definition des kontinuierlichen Gradienten kennengelernt. Dabei konnten wir aus den zwei Komponenten der Richtungsableitung erster Ordnung unterschiedliche Betragsgradienten ableiten. In diesem Kapitel gehen wir vom kontinuierlichen *Laplaceoperator* aus, der die Krümmung der Bildfläche bestimmt.

Definition 1.13 (kontinuierlicher Laplaceoperator)

$$\nabla^2 f(m, n) = \frac{d^2}{dm^2} f(m, n) + \frac{d^2}{dn^2} f(m, n) \quad (1.34)$$

Während bei den Gradientenmasken die Reaktion auf eine Objektkante große Amplitudenwerte liefert, ist die Reaktion bei Maskenoperatoren, die den Laplaceoperator approximieren durch einen Nulldurchgang gekennzeichnet.

Durch zweimalige einfache Differenzbildung benachbarter Pixel und anschließender Aufsummierung wird folgender einfacher Lokaloperator generiert:

$$\mathbf{LAP1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Alternativ kann ein zweiter Operator angegeben werden, der eine weitere einfache Approximation darstellt:

$$\mathbf{LAP2} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Wie leicht gezeigt werden kann, ist der kontinuierliche Laplaceoperator rotationsinvariant. Wir erhalten also ein Ergebnis, das unabhängig von der Lage der Kante in der Ebene ist. Entscheidend für die Amplitude der Reaktion ist allein die Krümmung des Originalbildes. Dieses Verhalten bleibt bei den angegebenen diskreten Approximationen selbst annäherungsweise nicht erhalten. Zunächst kann anhand der obigen Lokaloperatoren leicht festgestellt werden, daß die Ergebnisse für 0 und 90 bzw. 45 und 135 Grad jeweils gleich sind. Diese Ergebnisse sind jedoch paarweise ungleich. Das folgende Beispiel verdeutlicht diesen Sachverhalt.

Beispiel 1.6 *Laplaceoperator $L1$ angewandt auf eine 90 und eine 45 Grad Kante ergibt:*

$$\begin{array}{cccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & \longrightarrow & 0 & -1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{array}$$

$$\begin{array}{cccccc}
 1 & 1 & 1 & 1 & 0 & & 0 & 0 & -2 \\
 1 & 1 & 1 & 0 & 0 & & 0 & -2 & 2 \\
 1 & 1 & 0 & 0 & 0 & \longrightarrow & -2 & 2 & 0 \\
 1 & 0 & 0 & 0 & 0 & & 2 & 0 & 0
 \end{array}$$

Die Reaktionen unterscheiden sich um den Faktor 2, der Operator ist demnach nicht rotationsinvariant. Analoge Verhaltensweisen liegen bei dem Operator L2 vor. Die diskreten Laplaceoperatoren reagieren sehr heftig auf Ecken, Linien, Linienenden und isolierte Punkte. Das Verhalten in einem rauschbehafteten Bild kann daher unbefriedigend sein. Wir werden dies im Versuchsteil des Praktikums sehen.

1.3.3 Differenzen von Mittelwerten

Die Kantenoperatoren der beiden letzten Abschnitte haben wir aus dem kontinuierlichen Gradienten- bzw. Laplaceoperator abgeleitet. Als Nachteil dieser Verfahren tritt die hohe Störempfindlichkeit auf. Die Ableitung eines Signalverlaufs bevorzugt hohe Frequenzanteile. Da der Nutzanteil des gestörten Signals in der Regel Tiefpaßcharakter hat, der Störanteil dagegen unter idealisierten Voraussetzungen (weißes Rauschen) gleichmäßig über dem Frequenzbereich verteilt ist, werden außer den gesuchten Kanten die Rauschanteile besonders hervorgehoben.

In diesem Kapitel wollen wir Kantenoperatoren vorstellen, die grundsätzlich auf der Differenz von Mittelwerten zweier Umgebungen beruhen. Die Reihenfolge zuerst Mittelwertbildung (Tiefpaß) und dann Differenzbildung (Hochpaß) verdeutlicht das zu erwartende, gegenüber Rauschen verbesserte Verhalten durch den resultierenden Bandpaß. Wir werden zunächst die Operatoren vorstellen, um im Versuchsteil die Wirksamkeit im Vergleich zu den in den vorherigen Kapiteln angegebenen Operatoren zu untersuchen.

Grundsätzlich gesucht sind Operatoren, die Differenzen von Mittelwerten bilden. Die beiden zunächst gesuchten Mittelwerte sollten auf nichtüberlappenden Gebieten beruhen. Bei überlappenden Gebieten hätten die entsprechenden Pixel keine Wirkung, denn sie würden sich bei der Differenzbildung egalalisieren. Analog zu den Gradientenoperatoren müssen zwei Lokaloperatoren gebildet werden, die einfacherweise in horizontaler bzw. vertikaler Richtung besonders empfindlich reagieren. Um bei der Mittelwertbildung nicht von vornherein die eigentlich gesuchten Kanten zu verschleifen, werden die Mittelungen jeweils nur über Pixel entlang der Kanten gebildet. Bei einem horizontal

sensiblen Operator sind das zwei horizontale Pixelreihen über die zunächst gemittelt wird. Zusätzlich herangezogene horizontale Pixelreihen würden von vornherein die Konturen verschlechtern. Für den vertikalen Fall gilt das eben Gesagte natürlich analog.

Das denkbar einfachste Operatorpaar bildet den Mittelwert jeweils zweier benachbarter Pixel und anschließend deren Differenz. Beispiel 1.7 verdeutlicht diesen Sachverhalt.

Beispiel 1.7

$$\mathbf{DVM1M} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad \mathbf{DVM1N} = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Ein von Prewitt angegebenes Operatorpaar, das symmetrisch zu (m, n) liegende Ergebnisse liefert, zeigt Def. 1.14.

Definition 1.14 (Prewittoperator)

$$\mathbf{PREWM} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \mathbf{PREWN} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Alle bis dato besprochenen Operatoren basieren auf Differenzen von ungewichteten Mittelwerten. Sobel hat einen auf gewichteten Mittelwerten beruhenden Operator angegeben, der mit dem Prewittoperator fast identisch ist. Def. 1.15 zeigt den Sobeloperator.

Definition 1.15 (Sobeloperator)

$$\mathbf{SOBM} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \mathbf{SOBN} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Der Sobeloperator gewichtet die Pixel, die näher zum Operatormittelpunkt liegen stärker. Das bewirkt, daß diagonale Kanten im Gegensatz zum ungewichteten Fall sensibler detektiert werden.

Analog zu den Gradientenoperatoren gibt es auch hier unterschiedliche Verfahren, um die beiden Richtungs-differenzen zu einer Betragsdifferenz zusammenzufügen. Dieser Punkt soll an dieser Stelle nicht weiter erläutert werden, denn er kann analog zu Kapitel 1.3.2 behandelt werden. Im Versuchsteil werden wir einige Fälle exemplarisch untersuchen.

1.3.4 Cannyoperator

Die Kantenextraktion in den bisherigen Kapiteln basierte auf der Überlegung, daß Grauwertkanten, und nur solche sollen hier betrachtet werden, durch unterschiedliche Wertepiegel charakterisiert sind, die durch Differenzbildungen hervorgehoben werden. Es wurden keine mathematisch spezifizierten Anforderungen an das Verhalten des Operators gestellt. Canny [10] hat nun abweichend davon Kriterien formuliert, die eng genug sind um das gewünschte Verhalten des Detektors festzulegen, aber nur minimale Annahmen über den eigentlichen Kantenverlauf machen. Die gestellten Forderungen sind:

- geringe Fehlerrate, d.h. alle bestehenden Kanten sollten möglichst detektiert werden, aber auch nicht mehr
- gute Lokalisierungseigenschaften, d.h. geringe Distanz zwischen detektierter und wahrer Lage der Kante
- nur eine Antwort auf eine Kante

Für den eindimensionalen Fall kann das zu lösende *Optimierungsproblem* folgendermaßen formuliert werden: Wir gehen aus von einer gegebenen Kantenform die mit weißem gaußischem Rauschen gestört ist. Dieser Signalverlauf wird linear gefiltert, so daß dessen maximale Ausgangswerte die Kanten wiedergeben. Das Optimierungsproblem besteht nun in der Suche nach dem Filter, das die beste Lösung im Sinne der oben schon angegebenen Kriterien aufweist. Der genaue Gang der Herleitung soll an dieser Stelle nicht angegeben werden [10].

Spezialisiert man das Optimierungsverfahren auf sprungförmige Kantenverläufe, wie in Abb. 1.8 [10] geschehen, so kann man eine parametrisierte geschlossene Form der Lösung angeben. Bei der Herleitung sieht man, daß die Anforderungen nach geringer Fehlerrate und gutem Lokalisierungsverhalten gegensinnig verlaufen. Wir erhalten ein nach dem Ortsbereich skaliertes Filter mit einer einzigen Einhüllenden. Dabei hat ein schmales Filter bessere Lokalisierungseigenschaften, ein breites Filter aber ein besseres Störverhalten.

Eine effektive Approximation für sprungförmige Kantenverläufe ergibt sich aus der ersten Ableitung der Gaußfunktion. Für den eindimensionalen Fall erhalten wir dann:

$$f(m) = -\frac{m}{\sigma^2} e^{-\frac{m^2}{2\sigma^2}} \quad (1.35)$$

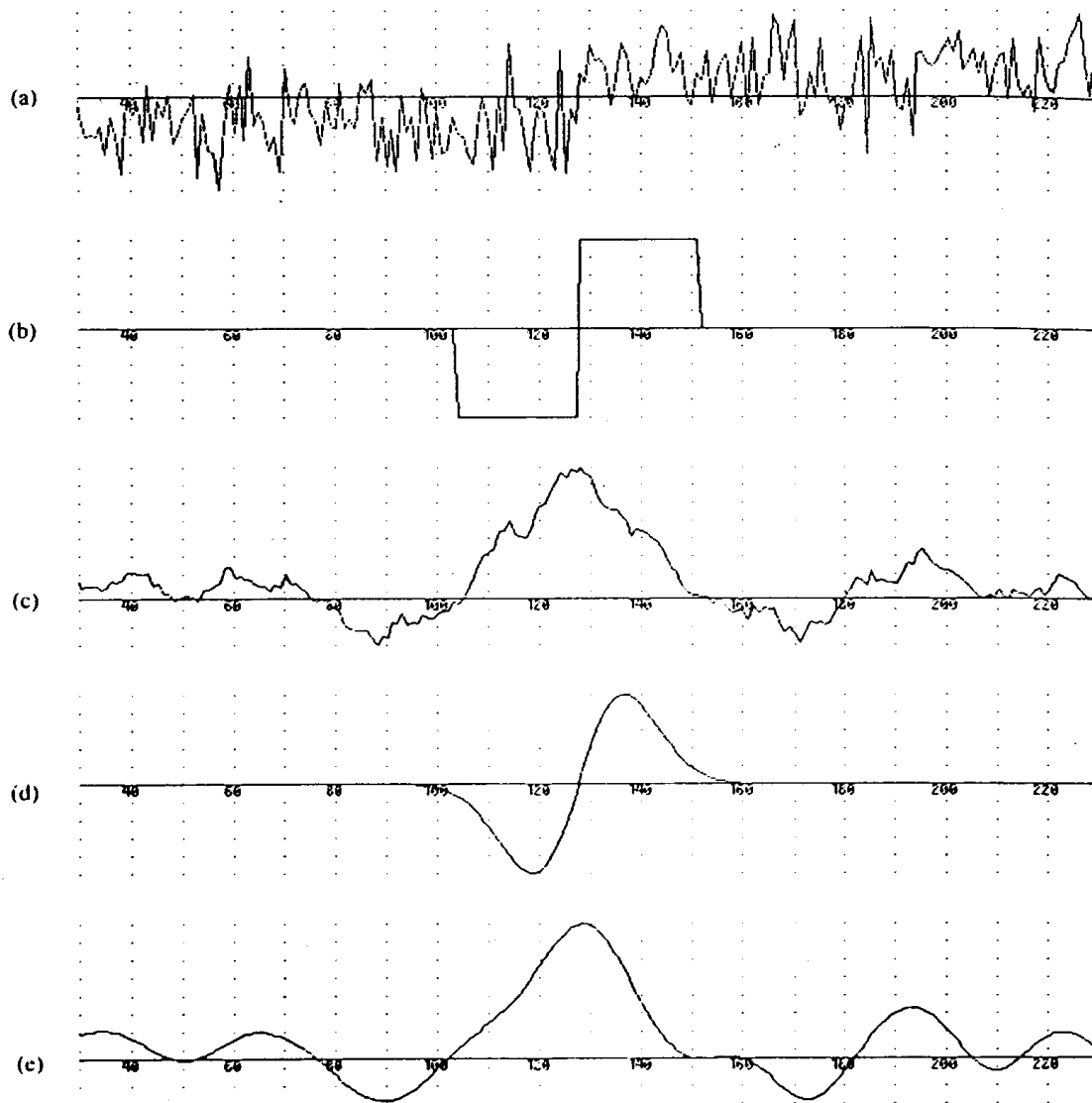


Abbildung 1.8: Extraktion einer verrauschten Sprungkante [8]: (a) verrauschte Sprungkante, (b) Operator für Differenz von Mittelwerten, (c) Operator aus (b) angewandt auf Signalverlauf aus (a), (d) erste Ableitung eines Gaußoperators, (e) Operator aus (d) angewandt auf Signalverlauf aus (a)

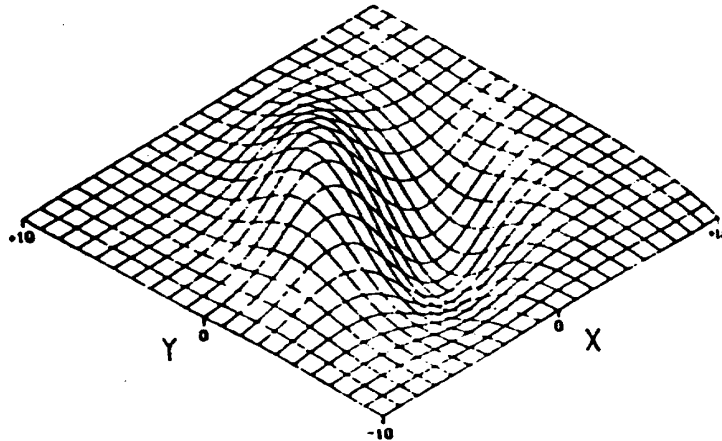


Abbildung 1.9: Zweidimensionale Näherung nach Macleod [9]

Bei realen Bildern ist der Unterschied zwischen optimalem Operator und obiger Approximation kaum merkbar. Für obiges Filter existieren sehr effektive zweidimensionale Erweiterungen, die in der Literatur beschrieben werden. Eine gute Annäherung an den obigen Operator wurde von Macleod [31] angegeben. Der Operator bildet die Differenz zweier gegeneinander versetzter Gaußfunktionen, die wiederum mit einer dritten im Ursprung lokalisierten Gaußfunktion multipliziert werden.

$$f(m, n) = e^{-\frac{m^2+n^2}{d_r^2}} \left[e^{-\left(\frac{d_{mn}+d_{pk}}{d_{pk}}\right)^2} - e^{-\left(\frac{d_{mn}-d_{pk}}{d_{pk}}\right)^2} \right] \quad (1.36)$$

d_r legt die örtliche Ausdehnung fest, d_{pk} den Abstand der Maxima vom Ursprung und $d_{mn} = m \sin \theta - y \cos \theta$ beinhaltet die Orientierung des Operators. Die Abb. 1.9 [31] zeigt ein Gitterdiagramm der zweidimensionalen Näherung von Macleod.

Auffallend ist zunächst, daß der Operator keine Diskontinuitäten beinhaltet. Seine Empfindlichkeit gegenüber Störeinflüssen wird geringer sein als bei Operatoren mit Diskontinuitäten, denn diese heben außer den gewünschten Kanten Rauschanteile ebenfalls

hervor. Auch hier haben wir ähnlich zu Kap.3.1 und 3.3 richtungsabhängiges Verhalten. Beispielhaft seien zwei Operatorpaare angegeben.

Beispiel 1.8 (Macleodoperator ($d_r = 1, d_{pk} = 1$ und $\theta = 90^0$ bzw. 180^0))

$$\mathbf{MLO1M} = \begin{bmatrix} -0.133 & 0 & 0.133 \\ -0.361 & 0 & 0.361 \\ -0.133 & 0 & 0.133 \end{bmatrix} \quad \mathbf{MLO1N} = \begin{bmatrix} 0.133 & 0.361 & 0.133 \\ 0 & 0 & 0 \\ -0.133 & -0.361 & -0.133 \end{bmatrix}$$

Beispiel 1.9 (Macleodoperator ($d_r = 1.5, d_{pk} = 2$ und $\theta = 90^0$ bzw. 180^0))

$$\mathbf{MLO2M} = \begin{bmatrix} -0.028 & -0.073 & 0 & 0.073 & 0.028 \\ -0.106 & -0.277 & 0 & 0.277 & 0.106 \\ -0.166 & -0.432 & 0 & 0.432 & 0.166 \\ -0.106 & -0.277 & 0 & 0.277 & 0.106 \\ -0.028 & -0.073 & 0 & 0.073 & 0.028 \end{bmatrix}$$

$$\mathbf{MLO2N} = \begin{bmatrix} 0.028 & 0.106 & 0.166 & 0.106 & 0.028 \\ 0.073 & 0.277 & 0.432 & 0.277 & 0.073 \\ 0 & 0 & 0 & 0 & 0 \\ -0.073 & -0.277 & -0.432 & -0.277 & -0.073 \\ -0.028 & -0.106 & -0.166 & -0.106 & -0.028 \end{bmatrix}$$

Das Operatorpaar **MLO1M** / **MLO1N** zeigt ähnliches Verhalten wie das Paar **SOBM** / **SOBN**. Lediglich das Verhältnis zwischen den Elementen weicht hier vom Sobeloperator ab. Dieser geringe Unterschied wird kaum voneinander abweichende Verhaltensweisen hervorbringen. Das Operatorpaar **MLO2M** / **MLO2N** hingegen hat schon deutlich den in Abb. 1.9 charakterisierten Verlauf und die damit verbundenen Eigenschaften. Wir werden im Versuchsteil die angestellten Vermutungen bestätigt finden.

1.4 Binarisierung und morphologische Binärbildverarbeitung

Im Kapitel Binärbildverarbeitung sollen Verfahren der *mathematischen Morphologie* behandelt werden [17, 47, 20]. Dabei handelt es sich um die Lokaloperationen Dilatation, Erosion, Open und Close. Vorangestellt wird in Kap.4.1 kurz auf die Binarisierung von Grauwertbildern eingegangen.

Die mathematische Morphologie basiert auf fundamentalen Mengenoperationen. Die Punkte der Ursprungsmenge werden in die Menge der aktiven bzw. selektierten und die der inaktiven bzw. deselektierten unterteilt. Dabei seien definitionsgemäß schwarze Pixel eines Bildes selektiert, weiße hingegen deselektiert. Wir behandeln in den folgenden Kapiteln nacheinander die folgenden Operationen der binären Morphologie: Dilatation in Kap. 4.2, Erosion in Kap. 4.3 und Open sowie Close in Kap. 4.4.

1.4.1 Binarisierung

Viele Anwendungen der digitalen Bildverarbeitung können als binäre Probleme betrachtet werden. Dabei treten nur zwei Grauwertstufen auf, eine für das oder die vorhandenen Objekte und eine für den Hintergrund. Beispielsweise kann es sich um Schrift, Zeichnungen oder auch flache Werkstücke handeln, für die alle ein idealisiertes zweipegeliges Bildmodell angenommen wird. Die Aufnahme der Szene mit einer Kamera führt aber in der Regel auf ein Bild mit einem Histogrammverlauf über weite Bereiche der Grauwertskala. Die Gründe für das Abweichen vom idealen binären Verhalten des aufgenommenen Bildes bei an sich zweipegeliger Vorlage sind von vielfältiger systembedingter Natur. Die Effekte reichen von Kamerafehlern bis hin zu inhomogener Szenenbeleuchtung.

Somit stellt sich die Aufgabe der *Binarisierung* eines Grauwertbildes, so daß der enthaltene figürliche Bildinhalt erhalten bleibt. Die einfachste Lösung sieht die Anwendung einer konstanten Grauwertschwelle S vor, die die folgende Grauwerttransformation vornimmt.

$$y(x) = \begin{cases} 0 & \text{für } x < S \\ 255 & \text{sonst} \end{cases}$$

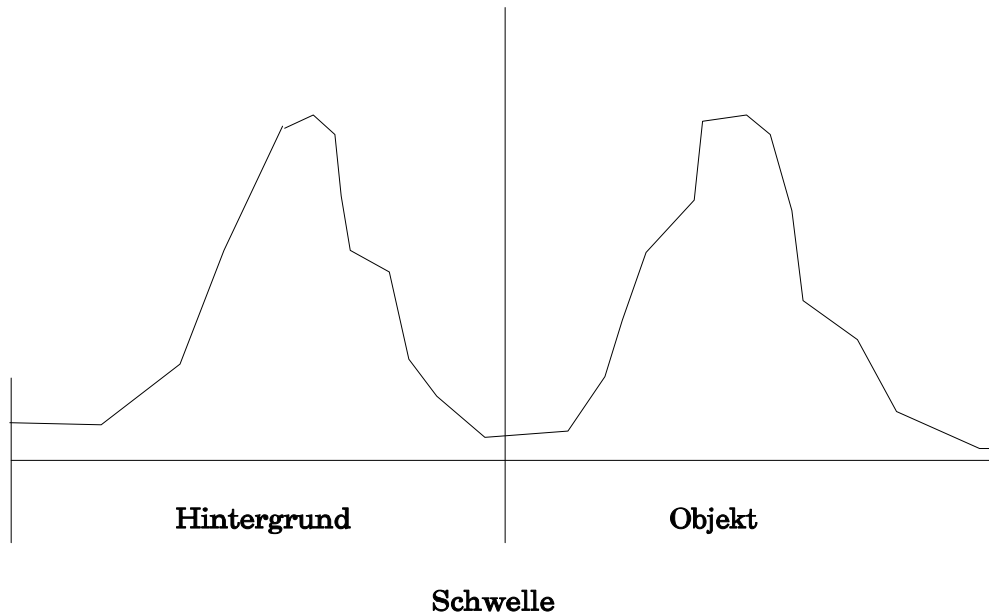


Abbildung 1.10: Binarisierungsschwelle bei bimodalem Grauwert-Histogramm

Diese einfache Methode ist nur für den Fall eines bimodalen Grauwert-Histogramms erfolgversprechend. Nur in diesem Fall kommt der Histogrammverlauf dem Idealmodell eines Zweipegelbildes nahe genug.

Die Abb. 1.10 zeigt einen solchen typischen Verlauf. Wir sehen zwei scharfe Anhäufungen, eine für das Objekt und eine für den Hintergrund. Die Binarisierungsschwelle wird genau an die tiefste Stelle zwischen den beiden Anhäufungen gelegt.

1.4.2 Dilatation

Die *Dilatation* ist eine morphologische Operation, die zwei Mengen mit Hilfe der schon besprochenen allgemeinen Lokaloperation und nachfolgenden logischen Operationen miteinander verknüpft. A und B seien zwei Mengen im zweidimensionalen Raum mit Elementen $a = (a_m, a_n)$ und $b = (b_m, b_n)$. A repräsentiert die Menge der selektierten Pixel des Originalbildes und B die Menge der selektierten Pixel eines Lokaloperatorfensters. Die Anordnung der Pixel b im besagten Fenster bezeichnet man auch als

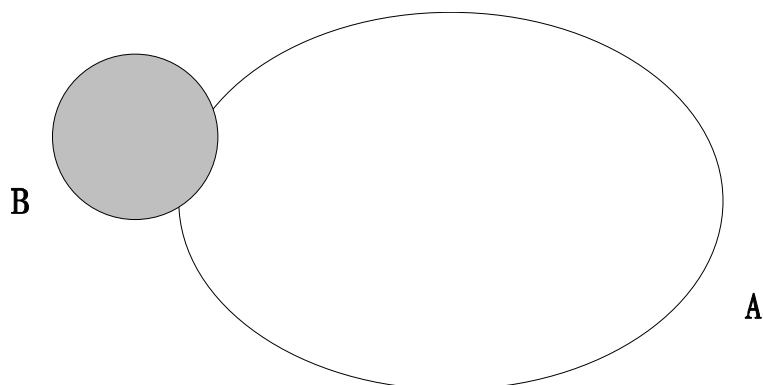


Abbildung 1.11: Strukturelement B und Bild A überlappen sich

Strukturelement, weil dessen Form und die Lage des darin enthaltenen Ursprungs die Filtereigenschaften entscheidend beeinflussen. Die Dilatation von A mittels B ist die Menge aller möglichen Punkte p , die mindestens einen selektierten Objektbildpunkt und einen selektierten Strukturelementbildpunkt an der Verschiebungsposition p der Maske gemeinsam haben. Die Abb.1.11 verdeutlicht diesen Vorgang der Überlappung.

Die Def. 1.16 gibt die Notation mit Hilfe der Korrelation und veränderten Operatoren im Innenprodukt und die (APL2 nahe) Notation mit dem allgemeinen Lokaloperator an. Wir sehen an dieser Stelle sehr deutlich, wie übergreifend die Definition des allgemeinen Lokaloperators ist.

Definition 1.16 (Dilatation)

$$A \oplus B = B \overset{(\vee, \wedge)}{\#} A \tag{1.37}$$

$$A \oplus B = B(\vee LOK \wedge)A \tag{1.38}$$

Die Definition 1.16 soll anhand eines Beispiels erläutert werden. Wie die Abb.1.12 zeigt, kommt es neben der Vergrößerung von Objekten zur Auffüllung von Konkavitäten. Die zusätzlich bestehende Möglichkeit der Verschmelzung zweier Objekte ist hier nicht gezeigt. Die in der Abbildung bereits enthaltene Erosion ist der Vollständigkeit halber schon mit aufgeführt, wird aber erst in Kap.4.3 besprochen.

Eine alternative Betrachtungsweise für die morphologische Bildfilterung beruht auf der *Translation* der selektierten Pixel des Originalbildes. Diese Betrachtungsweise hat große

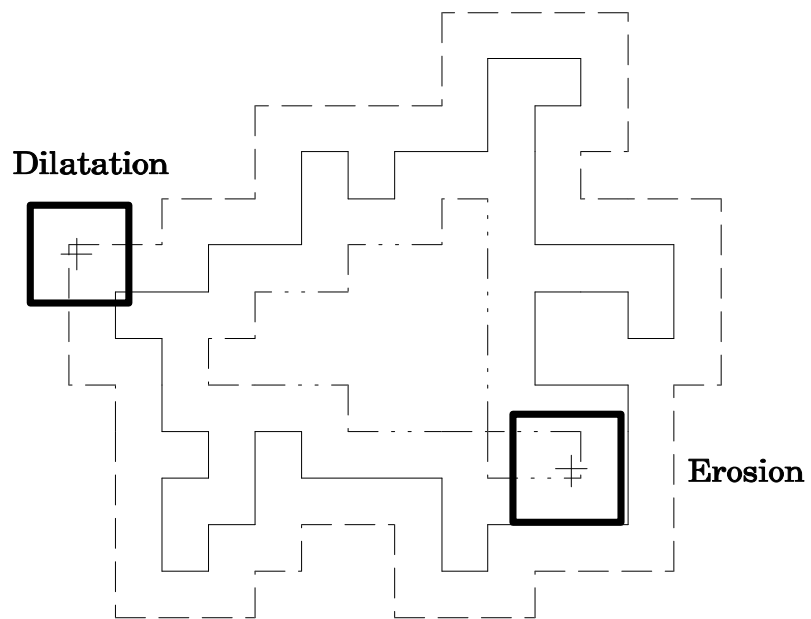


Abbildung 1.12: Dilatation und Erosion an einer Beispielfigur (durchgezogene Linie: Originalbild, Strichlinie: Dilatation, Strichpunktlinie: Erosion)

praktische Bedeutung für die Realisierung schneller morphologischer Operationen in kommerziellen Bildverarbeitungssystemen. Dazu ist zunächst die Formulierung einer Eigenschaft der Dilatation notwendig. Ein Strukturelement B kann in eine Vereinigung zweier Mengen B_1 und B_2 zerlegt werden.

Eigenschaft 1.2 (Distributivität der Dilatation)

$$A \oplus (B_1 \cup B_2) = (A \oplus B_1) \cup (A \oplus B_2) \quad (1.39)$$

Die Zerlegung von B kann soweit fortgesetzt werden, bis jedes B_i nur noch aus einem einzigen Element besteht, für das die Dilatation in eine reine Translation des Originalbildes übergeht. Die Dilatation $A \oplus B$ kann somit als die Vereinigungsmenge aller Translationen von A um alle Vektoren $-b$ der Strukturelementmenge B angesehen werden. Wir formulieren diesen Sachverhalt hier als eine weitere Eigenschaft der Dilatation.

Eigenschaft 1.3

$$A \oplus B = \bigcup_{b \in B} (A)_{-b} \quad (1.40)$$

Die Eigenschaft 1.3 in Worten ausgedrückt läßt sich folgendermaßen formulieren: Verschiebe das Quellbild A um alle in der Strukturelementmenge enthaltenen Vektoren $-b$ und bilde darüber die Vereinigungsmenge. Daraus läßt sich einfach ablesen, daß das Quellbild A nur dann mit Sicherheit in $A \oplus B$ enthalten ist, wenn die Menge B den Ursprung enthält, also den Verschiebungsvektor $(0, 0)$. Aus Eigenschaft 1.3 ergibt sich eine weitere wichtige Eigenschaft der Dilatation, die Ortsinvarianz.

Eigenschaft 1.4 (Ortsinvarianz der Dilatation)

$$(A)_m \oplus B = (A \oplus B)_m \quad A \oplus (B)_m = (A \oplus B)_{-m} \quad (1.41)$$

Die Dilatation eines verschobenen Bildes ergibt die Dilatation des Originalbildes mit anschließender Verschiebung um den gleichen Vektor. Eine entsprechende Beziehung gilt auch für die Verschiebung des Strukturelements, wobei zusätzlich ein negatives Vorzeichen auftritt.

Wir hatten schon im Anschluß an Eigenschaft 1.3 gesehen, daß das Dilatationsergebnis das Originalbild mit Sicherheit enthält, wenn im Strukturelement der Ursprung $(0, 0)$,

also die Verschiebung um den Nullvektor enthalten ist. Man bezeichnet dieses Verhalten als *extensiv*, denn mindestens das Quellbild selbst ist im Ergebnis enthalten. Wählt man jedoch ein Strukturelement B mit nur zwei voneinander weit genug entfernten Elementen aus, so kann es zu einer reinen Kopie der Originalbilder an die Positionen der in B enthaltenen Elemente kommen. Die Dilatation wäre dann *nichtextensiv*. Die Wahl eines geeigneten Strukturoperators in Abhängigkeit von den zu bearbeitenden Objekten im Quellbild ist also von entscheidender Bedeutung.

1.4.3 Erosion

Die zur Dilatation duale morphologische Operation heißt *Erosion*. Im Sprachgebrauch der Bildverarbeitung wird sie auch *shrink* oder *reduce* genannt. Die Definition 1.17 gibt eine Form anhand der Korrelation mit veränderten Operatoren im Innenprodukt und eine (APL2 nahe) Notation mit dem allgemeinen Lokaloperator an.

Definition 1.17 (Erosion)

$$A \ominus B = B \overset{(\wedge, \leq)}{\#} A \quad (1.42)$$

$$A \ominus B = B(\wedge LOK \leq)A \quad (1.43)$$

Die Erosion von A mittels B besteht also aus der Menge aller Positionen p von B , für die das Strukturelement B vollständig im Originalbild A enthalten ist, also A muß B majorisieren. Die Abb.4.4 zeigt diesen Sachverhalt. Zusätzlich enthält die Abb.4.3 in Kap.4.2 über die Dilatation auch das entsprechende Ergebnis für die Erosion.

Definitionsgemäß ist das Erosionsergebnis im Quellbild enthalten. Von daher verstehen sich die Synonyme *shrink* und *reduce* von selbst. Die Eigenschaften der Erosion sind dual zur Dilatation. Sie verkleinert die Objektgröße, entfernt Konvexitäten und kann zur Auftrennung in Teilobjekte führen. Auf diese elementaren Eigenschaften wird im weiteren Verlauf noch eingegangen. Zunächst soll jedoch eine zur Dilatation äquivalente Eigenschaft formuliert werden, die auf der Translation der Objekte im Bild A selbst basiert. Dazu ist die Eigenschaft 1.5 voraussetzend.

Eigenschaft 1.5 (Distributivität der Erosion)

$$A \ominus (B_1 \cup B_2) = (A \ominus B_1) \cap (A \ominus B_2) \quad (1.44)$$

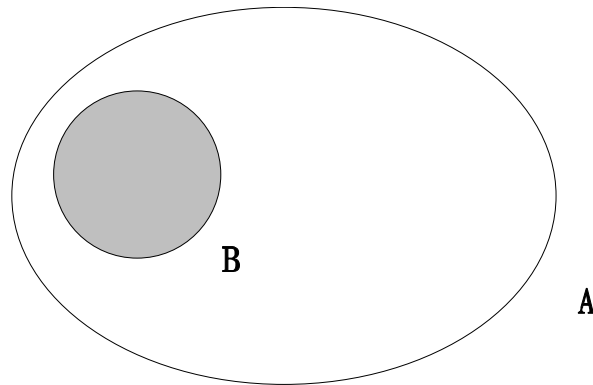


Abbildung 1.13: Bild A majorisiert Strukturelement B

Man kann also das Strukturelement B in die Vereinigungsmenge von zwei Teilmengen B_1 und B_2 zerlegen und damit die Erosion in eine Schnittmenge über zwei Teilerosionen überführen. Dies Vorgehen kann weiter fortgesetzt werden, bis die Teilmengen aus einzelnen Bildpunkten bestehen. Die Erosion mit einem einzelnen Punkt geht aber analog zur Dilatation in eine Verschiebung über. Die Erosion kann also als eine Verkettung von Verschiebungen von A um alle Elemente von B und anschließender Schnittmengenbildung formuliert werden.

Eigenschaft 1.6

$$A \ominus B = \bigcap_{b \in B} (A)_{-b} \quad (1.45)$$

Die Erosion von A mittels B ist also die Schnittmenge über alle A verschoben um $-b$, mit allen $b \in B$. Die Erosion zeichnet sich als *antiextensiv* aus, das Ergebnis ist also im Quellbild enthalten, wenn das Strukturelement den Ursprung enthält.

Analog zur Dilatation haben wir bei der Erosion ortsvariantes Verhalten.

Eigenschaft 1.7 (Ortsinvarianz der Erosion)

$$(A)_m \ominus B = (A \ominus B)_m \quad (1.46)$$

$$A \ominus (B)_m = (A \ominus B)_{-m} \quad (1.47)$$

Weil die Erosion nicht kommutativ ist, müssen beide Formeln angegeben werden. Es ist bedeutungslos ob Quellbild oder Strukturelement verschoben werden, wir erhalten gegeneinander verschobene aber ansonsten gleiche Ergebnisse. Die Relation unterschiedlicher Quellbildgrößen bei gleichen Strukturelementen und umgekehrt wird durch das größenverhältnisbewahrende Verhalten festgelegt. Ist ein Bild A_1 im Bild A_2 enthalten und werden beide mit dem gleichen Strukturelement erodiert, so ist die Erosion von A_1 in der von A_2 enthalten. Man bezeichnet in der Literatur ein solches Operatorverhalten als *increasing*. Sind auf der anderen Seite A_1 und A_2 Strukturelemente und ist wiederum A_1 in A_2 enthalten, so ist die Erosion von B mit A_2 in der mit A_1 enthalten. Nach der letzten Aussage zufolge haben somit größere Strukturelemente größeren Einfluß auf das zu filternde Bild. Besonders erwähnt werden sollen Operatoren mit gleicher Form aber unterschiedlicher Größe.

1.4.4 Open und Close

In den beiden vorherigen Kapiteln wurden mit der Dilatation und der Erosion die beiden grundlegenden Operationen der morphologischen Bildverarbeitung vorgestellt. In der Praxis werden beide Operationen in der Regel nacheinander mit dem gleichen Strukturelement ausgeführt, entweder zuerst Erosion und dann Dilatation, was wir als *Open* bezeichnen wollen, oder in umgekehrter Reihenfolge, was wir als *Close* bezeichnen wollen. Der Vorteil der kaskadierten Ausführung liegt darin, daß feine Details, kleiner als das Strukturelement, unterdrückt werden können ohne, wie bei isolierter Ausführung der Erosion üblich, zu global signifikanten geometrischen Verfälschungen der Bildobjekte zu führen.

Definition 1.18 (Open) Die Operation *Open* eines Bildes A durch ein Strukturelement B ist definiert als:

$$A \circ B = (A \ominus B) \oplus B \quad (1.48)$$

Definition 1.19 (Close) Die Operation *Close* eines Bildes A durch ein Strukturelement B ist definiert als:

$$A \bullet B = (A \oplus B) \ominus B \quad (1.49)$$

Dabei sind die Operationen \oplus und \ominus die in den vorherigen Kapiteln eingeführten Operationen Dilatation und Erosion. Beide Operationen sollen anhand des schon bekannten

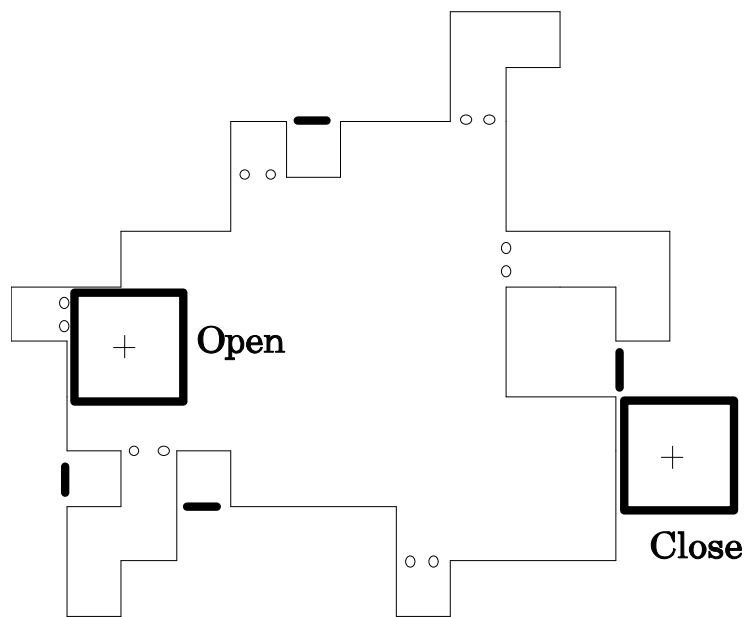


Abbildung 1.14: Open (Punktlinie) und Close (Strichlinie) angewandt auf das Binärbild aus Abb.4.3 mit quadratischem Strukturelement

Bildbeispiels verdeutlicht werden. Ausgangspunkt ist ein gemeinsames Quellbild A , um die unterschiedlichen Filtereigenschaften zu verdeutlichen.

Die Abb.4.3 zeigt die Operationen Open und Close angewandt auf das bereits bekannte Binärbild. Das Ergebnis der Operation Open entspricht der Fläche, die man beim überstreichen des Quellbildes mit dem Strukturelement entlang des Inneren der Kontur erhält. Analog liefert die Operation Close ein Ergebnis, das dem Entlangfahren am äußeren der Kontur entspricht. Feinstrukturierte Details, kleiner als das Strukturelement werden demnach beim Open herausgesiebt, dünne Verbindungen zwischen größeren Teilen aufgelöst. Deshalb die Bezeichnung Open. Beim Close dagegen bleiben zunächst alle Anteile unverändert erhalten. Kleinere Risse, Lücken und feine Details werden dagegen aufgefüllt und mit den großen Teilen zusammengeschlossen.

Viele Eigenschaften der Operationen Open und Close lassen sich aus denjenigen der Dilatation und Erosion ableiten. Es kommen jedoch zusätzliche, für unsere Problemstellung der Bildvorverarbeitung nützliche Faktoren hinzu. Aus Abb.4.3 läßt sich ablesen, daß die Operation Open *antiextensiv*, die Operation Close hingegen *extensiv* ist.

Eigenschaft 1.8

$$A \circ B \subseteq A \quad (\text{Open ist antiextensiv}) \quad (1.50)$$

$$A \subseteq A \bullet B \quad (\text{Close ist extensiv}) \quad (1.51)$$

$A \circ B$ ist also in A enthalten oder maximal gleich A . A ist dementsprechend in $A \bullet B$ enthalten oder gleich $A \bullet B$. Beides gilt unabhängig davon, ob B den Ursprung enthält oder nicht. Zur Erinnerung: Dies war bei der Dilatation und der Erosion nicht der Fall. Die Verallgemeinerung geht sogar soweit, daß die Ergebnisse von Open und Close ganz unabhängig von der Lage von B sind, das Ergebnis also völlig unberührt davon bleibt, wo das Strukturelement liegt. Zunächst folgt aus der Ortsinvarianz von Dilatation und Erosion, so wie sie in den Kapiteln 4.2 und 4.3 angegeben wurden, die Ortsinvarianz von Open und Close. Zusätzlich dazu sind Open und Close unabhängig gegenüber Verschiebungen des Strukturelements B selbst.

Eigenschaft 1.9 (Ortsinvarianz von Open und Close)

$$A \circ (B)_m = A \circ B \quad (1.52)$$

$$A \bullet (B)_m = A \bullet B \quad (1.53)$$

Die wohl wichtigste Eigenschaft der kaskadierten morphologischen Operationen Open und Close ist deren *Idempotenz*. Die iterative Anwendung eines der Operatoren mit dem gleichen Strukturelement führt auf ein unverändertes Ergebnis. Es gibt sogar Quellbilder, die bezüglich bestimmter Strukturelemente B bei Anwendung einer der beiden Operationen von vornherein unverändert bleiben.

Eigenschaft 1.10

$$(A \circ B) \circ B = A \circ B \quad (1.54)$$

$$(A \bullet B) \bullet B = A \bullet B \quad (1.55)$$

Man bezeichnet ein Bild als B -open, wenn es bezüglich der Operation Open B unverändert bleibt. Analog bezeichnet man ein Bild als B -close, wenn es bezüglich der Operation Close B unverändert bleibt. Diese Eigenschaft eines Bildes kann von vornherein bestehen oder durch die entsprechende Operation herbeigeführt werden. Ein nachrichtentechnisches Analogon zur hier festgestellten Idempotenz sind ideal tiefpaßbegrenzte Signale. Eine erneute ideale Tiefpaßfilterung mit einer Grenzfrequenz, die größer oder gleich der bereits festgestellten Frequenzgrenze liegt, hat keinerlei Einfluß auf das Signal, die Operation ist mithin idempotent.

1.5 Aufgaben

1.5.1 Theoretische Aufgaben

Aufgabe 1: Was bewirkt eine Transformation mit monoton fallender Kurvenform? Wie sieht das Ergebnis histogramm aus?

Aufgabe 2: Wie sieht die Transformationskurve für eine Histogrammtransformation aus, bei der nur ein bestimmter Grauwertbereich (z.B. alle Grauwerte von Objekten) heller dargestellt werden soll und der Rest (z.B. der Hintergrund) mit einem konstanten Wert.

Aufgabe 3: Wie sieht die Transformationskurve zur Erzeugung eines Binärbildes mit Hilfe eines Thresholds aus.

Aufgabe 4: Worin unterscheidet sich das Histogramm / akkumulierte Histogramm eines Grauwertbildes?

Aufgabe 5: Berechnen Sie die Kennlinie für den Fall der Histogrammexponentialisierung. Die Zielverteilung ist dabei definiert als:

$$p_s(s) = ae^{-as} \quad (1.56)$$

Aufgabe 6: Zeigen Sie, dass eine Medianfilterung eine nichtlineare Transformation ist.

Aufgabe 7: Ergänzen Sie die in Beispiel 1.5 gezeigten Verhalten der beiden Fensterformen durch das Verhalten bei diagonalen Linien- und Kantenverläufen.

Aufgabe 8: Erstellen Sie eine Tabelle, aus der die Reaktion der drei Operatorpaare Δ_m und Δ_n , Δ_{2m} und Δ_{2n} bzw. Δ_+ und Δ_- auf die zwei optimalen Sprungkanten unter 0 und 45 Grad ersichtlich wird.

Aufgabe 9: Beurteilen Sie die Richtungsabhängigkeit der drei Betragsoperationen anhand der in Aufgabe 1.5.1 erstellten Tabelle. Worin besteht der Vorteil der Gl. 1.33 ?

Aufgabe 10: Zeigen Sie, daß durch zweimalige einfache Differenzbildung benachbarter Pixel und anschließender Aufsummierung folgender einfacher Lokaloperator generiert wird:

$$\mathbf{LAP1} = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

1.5.2 Praktische Aufgaben

Aufgabe 1: Machen Sie sich mit den zu verwendenden Matlab Funktionen vertraut.

Aufgabe 2: (Histogram) Laden Sie das Bild 'pout.tif' in Matlab und berechnen Sie ein Histogramm von diesem Bild (imhist) . Was lässt sich anhand des Histogrammverlaufs über das Bild aussagen?

Aufgabe 3: (Histogrammausgleich) Schreiben Sie ein Skript mit dessen Hilfe Sie einen Histogrammausgleich durchführen können und wenden sie dieses auf das Bild aus der vorherigen Aufgabe an.

- Aufgabe 4:** (Histogrammausgleich) Laden Sie das Bild 'circuit.tif' und berechnen Sie einen Histogrammausgleich mit der von Matlab zur Verfügung gestellten Funktion 'histeq'. Inwieweit kann man diese Ergebnisse auch mit der Funktion 'imadjust' erreichen.
- Aufgabe 5:** (Lineare Lokaloperatoren) Wenden Sie die drei Lokaloperatoren von Beispiel 1.1 auf ein Bild mit Salz- und Pfefferrauschen, sowie ein Bild mit Gaußschen Rauschen an und vergleichen Sie die Ergebnisse.
- Aufgabe 6:** (Lineare Lokaloperatoren) Implementieren Sie einen Gaußfilter und wenden sie diesen auf die beiden Bilder der vorherigen Aufgabe an. Welche Unterschiede ergeben sich.
- Aufgabe 7:** (Medianfilter) Erzeugen Sie eine 1D-Testsequenz, in der Rampen und Maxima mit scharfe Kanten vorhanden sind. Wählen Sie zwei benachbarte Fenstergrößen so aus, dass das kleinere Fenster genau die Länge der größten Rampe bzw. des größten Maxima besitzt und führen Sie eine Medianfilterung durch. Laden Sie anschließend ein Bild, fügen Sie Salz- und Pfefferrauschen hinzu und filtern Sie das Bild.
- Aufgabe 8:** (Bildverschärfung) Überlegen Sie wie man unter Benutzung eines Filters und Verwendung einfacher arithmetischer Operationen mit dem Originalbild eine Verschärfung des Bildes erreichen kann. Implementieren Sie diese Bildverschärfung.
- Aufgabe 9:** (Unschärfefilter) Unschärfefilter werden verwendet, um den Kontrast in einem Bild zu erhöhen. In Matlab ist ein solches Filter als ein negierter Laplacianfilter implementiert. Erzeugen Sie dieses Filter mit der Funktion `fspecial` und dem Parameter 'unsharp' und wenden Sie diese auf ein Bild ihrer Wahl an.
- Aufgabe 10:** (Kantenextraktion) Laden Sie das Bild `rice.tif` und filtern Sie dieses mit den Filtern `GRAD1M`, `GRAD1N`, `GRAD2M`, `GRAD2N`, und `RCO`. Verknüpfen Sie die Ergebnisse mit der Formel $Q = \sqrt{I_x^2 + I_y^2}$ (1.31) (alternativ Betragssumme (1.32) oder Maximumsbildung (1.33)) und vergleichen Sie die Ergebnisse.
- Aufgabe 11:** (Kantenextraktion) Wenden Sie nun die Laplaceoperatoren `LAP1` und `LAP2` auf das Bild an und vergleichen Sie die Ergebnisse mit denen aus der vorherigen Aufgabe. Wie stark beeinflusst gaußssches Rauschen die Ergebnisse?
- Aufgabe 12:** (Kantenextraktion) Führen Sie eine Kantendetektion mit der Matlabfunktion `edge` durch. Vergleichen Sie die Ergebnisse für den Prewitt-, Sobel-, und

Cannyoperator auf rauschfreien und verrauschten Bildern (Salz- und Pfefferrauschen und Gaußsches Rauschen).

Aufgabe 13: (Dilatation) Laden Sie das Bild 'rice.tif' und wandeln Sie dieses in ein Binärbild um, sodass die Objekte durch 1 und der Hintergrund durch 0 gegeben sind. Dilatieren Sie nun das Bild mit einem quadratischen Strukturelement der Größe 3×3 . Zerlegen Sie das Strukturelement und berechnen Sie die Dilatation mit iterativer Anwendung der Operatorelemente. Bringen Sie eine Störung in das Bild ein, indem Sie gezielt einige Pixel von 0 auf 1 setzen. Berechnen Sie die Dilatation und vergleichen Sie die Ergebnisse.

Aufgabe 14: (Erosion) Berechnen Sie eine Erosion des Bildes aus der vorherigen Aufgabe mit dem quadratischen Strukturelement. Bringen Sie eine Störung ein und bilden Sie je nach den Ausmaßen der eingestreuten Störungen zwei abweichende Operatoren, die gerade noch und gerade nicht mehr zum Verschwinden der Störungen führen.

Aufgabe 15: (Open und Close) Bilden Sie die Ergebnisse der Operationen Open und Close anhand des Original- und gestörten Bildes. Vergleichen Sie die vier morphologischen Operationen und das Original daraufhin, welche Mengenverhältnisse zwischen den aktivierten Pixeln bestehen. Welchen Vorteil bieten die Operationen Open und Close gegenüber Dilatation und Erosion?

Verwendete Matlabefehle

imhist(...), histeq(...), cumsum(...), filter2(...), imnoise(...), fspecial(...), edge(...), median(...), medfilt2(...), freqz2(...), imopen(...), imclose(...), strel(...)

Bilder in Matlab

pout.tif, circuit.tif, blood1.tif, rice.tif, saturn.tif, alumgrns.tif, cell.tif, eight.tif

Praktikumsversuch

Kamerakalibrierung

S. Rahmann und H. Burkhardt

2 Kamerakalibrierung

2.1 Einleitung

Eine Kamera bildet die dreidimensionale Welt auf ein zweidimensionales Bild ab; 3D Welt- oder Szenenpunkte werden auf 2D Bildpunkte projiziert. Dieser Praktikumsversuch beschäftigt sich mit der Modellierung von realen Kameras und geeigneten Methoden zur Bestimmung der Modellgrößen.

Ein zentraler Bereich des Computer Sehens ist die dreidimensionale Interpretation von Bildern. Basierend auf den Bilddaten können Objekte vermessen oder ihre Lage im Raum bestimmt werden. Auch vollständig unbekannte Szene können rekonstruiert werden. Für all diese Aufgaben ist es notwendig, daß in irgendeiner Form die Parameter des Kameramodells bestimmt werden. Die Bestimmung der Kameraparameter nennt man Kamerakalibrierung. Kamerakalibrierung ist ein wichtiges Thema, da die Genauigkeit der Rekonstruktion natürlich von der Genauigkeit der Kalibrierung abhängt. Grundsätzlich gibt es dazu zwei Möglichkeiten: Eine Kalibrierung auf Grund von Bildern eines speziellen Kalibrieremusters und die sogenannte Selbstkalibrierung, welche allein mehrere Aufnahmen einer unbekanntem Szene benutzt. Ansätze zur traditionellen Kamerakalibrierung sind schon seit langem bekannt und liefern eine präzise Bestimmung der Kameragrößen. Nachteilig ist hierbei, daß ein spezielles und genaues dreidimensionales Kalibrieremuster aufgebaut werden muß. In den 90-er Jahren wurden dann Ansätze zur Selbstkalibrierung entwickelt. Diese benötigen kein spezielles Kalibrieremuster, doch sind sie auch nicht so präzise. In diesem Praktikumsversuch wird eine Methode vorgestellt, welche die Vorzüge beider Ansätze vereint. Nur ein einfaches ebenes Kalibrieremuster, welches durch einen Standarddrucker erzeugt werden kann, ist dazu nötig. Dies wird von mehreren beliebigen Blickwinkeln aus aufgenommen. Da das Kalibrieremuster bekannt ist, die Blickwinkel aber nicht, ist das Verfahren ein Hybrid aus Standardkalibrierung und Selbstkalibrierung. Es ist einfach anzuwenden und liefert eine präzise Kalibrierung.

Es wird gezeigt, daß die projektive Geometrie eine geeignete Basis für die Beschreibung von geometrischen 2D/2D und 3D/2D Transformationen darstellt. 2D/2D Transformationen, wie z.B. die Abbildung eines ebenen Kalibrierungsmusters auf die Bildebene, werden ausführlich besprochen. 3D/2D Transformationen, also Abbildungen von Weltpunkten auf Bildpunkte, werden durch Kameras geleistet. Die dazu notwendigen Grundlagen der projektiven Geometrie werden im zweiten Kapitel präsentiert. Im folgenden Kapitel wird die Modellierung von Kameras besprochen. Im vierten Kapitel werden dann die eigentlichen Kalibriermethoden vorgestellt. Die 3D Kalibrierung wird der Vollständigkeit halber vorgestellt, ist sonst aber nicht Gegenstand des praktischen Teils. Die 2D Kalibrierung ist das Verfahren, welches im praktischen Teil umgesetzt wird. Der kurze Abschnitt der Selbstkalibrierung soll nur die grundsätzliche Idee vermitteln; ansonsten ist das Thema weitaus komplexer. Das fünfte Kapitel behandelt Aspekte der numerischen Berechnung. Die Beschreibung schließt mit einer Darstellung der relevanten Literatur. Die theoretischen Fragen im Aufgabenteil sollten vor dem Praktikumstermin bearbeitet werden, während die praktischen Aufgaben zum Termin selbst umgesetzt werden.

2.2 Projektive Geometrie

2.2.1 Einleitung

Euklidische und projektive Geometrie

Für uns Menschen ist in erster Linie die euklidische Geometrie von Bedeutung, da sie unserer natürlichen Erfahrung des uns umgebenden Raums entspricht. Im euklidischen Raum sind zum Beispiel Größen wie Längen und Winkel definiert. Punkte in der Ebene oder im Raum werden durch zwei- beziehungsweise dreidimensionale Vektoren dargestellt; dies sind ihre euklidischen (oder auch inhomogenen) Koordinaten. Punkte sind identisch, wenn ihre Koordinaten identisch sind. Auch können Abstände zwischen zwei Punkten berechnet werden. Geometrische Gebilde können im euklidischen Raum verschoben und gedreht werden.

Gewisse Sachverhalte sind aber in einer euklidischen Betrachtungsweise nur unschön zu beschreiben. So wird in der Geometrie der Ebene zwischen Punkten, Linien und Richtungen unterschieden. Zum Beispiel haben Linien mit unterschiedlichen Richtungen einen eindeutigen Schnittpunkt, während Linien mit identischer Richtung keinen

oder alle Punkte gemeinsam haben. Weiterhin möchte man eine allgemeinere Klasse von Transformationen zur Verfügung haben, welche nicht nur Verschiebung und Drehung enthält. Betrachten wir zu Veranschaulichung folgendes Beispiel: Der Mensch, oder eine Kamera, betrachten ein Gebäude, welches typischerweise aus rechtwinkligen Strukturen aufgebaut ist. Das Bild, welches dabei entsteht, ist durch eine perspektivische Projektion entstanden, so daß das wohlbekannte Phänomen von perspektivischen Verzerrungen zu beobachten ist: In der Welt parallele Linien treffen sich im Bild in den sogenannten Fluchtpunkten; ursprünglich senkrecht aufeinandertreffende Geraden bilden im Bild keine rechten Winkel mehr.

Die beschriebenen Phänomene sind mit Hilfe der projektiven Geometrie sehr elegant und einfach zu beschreiben. Der Schlüsselaspekt dabei ist, daß die projektive Geometrie die Erweiterung der euklidischen Geometrie um Punkte im Unendlichen ist. In der Ebene können Linien und Punkte nun einheitlich beschrieben werden. Auch parallele Linien besitzen einen Schnittpunkt, welcher dann nämlich ein Punkt im Unendlichen ist. Unter einer projektiven Transformation können unendlich weit entfernte Punkte auf endlich weit entfernte Punkte abgebildet werden und umgekehrt: Die parallelen Linien des Schachbretts schneiden sich im Unendlichen, während auf dem Bild (oder in den Augen des Betrachters) der Fluchtpunkt dieser Linien in dem Bild selbst, also im Endlichen, liegt. Mit Hilfe der projektiven Geometrie ergeben sich auch für die Modellierung von Kameras wesentlich einfachere Zusammenhänge, da die Abbildung von 3D Weltpunkten auf 2D Bildpunkte eine lineare Form erhält.

Euklidische und projektive Koordinaten

Ein Punkt im n -dimensionalen euklidischen Raum \mathcal{R}^n wird durch einen n -dimensionalen Vektor $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathcal{R}^n$ beschrieben. Dies ist die Repräsentation durch euklidische oder inhomogene Koordinaten.

Ein Punkt im n -dimensionalen projektiven Raum \mathcal{P}^n wird durch einen $(n+1)$ -dimensionalen Vektor $\tilde{\mathbf{x}} = (x_1, \dots, x_{n+1})^T$ beschrieben. Dies ist die Repräsentation durch projektive oder homogene Koordinaten. Homogene Koordinaten werden auch Verhältniskoordinaten genannt. Zwei projektive Punkte $\tilde{\mathbf{x}}$ und $\tilde{\mathbf{y}}$ sind identisch, genau dann wenn ihre homogenen Koordinatenvektoren äquivalent sind. Die Äquivalenz, beschrieben durch das Formelzeichen \sim , bedeutet Gleichheit bis auf einen Skalierungsfaktor:

$$\tilde{\mathbf{x}} \sim \tilde{\mathbf{y}} \quad \Leftrightarrow \quad \tilde{\mathbf{x}} = \lambda \tilde{\mathbf{y}}, \quad \text{mit } \lambda \in \mathcal{R} \setminus \{0\}.$$

Der Nullvektor $\mathbf{0}_{n+1}$ ist kein Element von \mathcal{P}^n , da dieser keiner geometrischen Größe im

projektiven Raum entspricht. Der projektive Raum \mathcal{P}^n ist also eine Menge von äquivalenten Vektoren in $\mathbb{R}^{n+1} \setminus \{\mathbf{0}_{n+1}\}$. Euklidische Koordinaten können einfach in projektive konvertiert werden:

$$(x_1, \dots, x_n) = \mathbf{x} \rightarrow \tilde{\mathbf{x}} = (\lambda x_1, \dots, \lambda x_n, \lambda) \quad .$$

Da der euklidische Raum einen Teilraum des projektiven Raums darstellt, können umgekehrt nicht alle Punkte des projektiven Raums in euklidische Koordinaten umgewandelt werden. Nur solche Punkte, für welche die letzte Komponente x_{n+1} nicht Null ist, können auch in inhomogenen Koordinaten umgeschrieben werden:

$$(x_1, \dots, x_n, x_{n+1}) = \tilde{\mathbf{x}} \rightarrow \mathbf{x} = \left(\frac{x_1}{x_{n+1}}, \dots, \frac{x_n}{x_{n+1}} \right) \quad .$$

Punkte der Form $\tilde{\mathbf{x}}_\infty = (x_1, \dots, x_n, 0)^T$ sind Punkte im Unendlichen. Der projektive Raum ist der euklidische Raum erweitert durch Punkte im Unendlichen.

2.2.2 Die projektive Ebene

Punkte und Geraden

In der projektiven Ebene werden Punkte und Geraden gleichermaßen als dreidimensionale Vektoren dargestellt. Die Bedingung dafür, daß ein Punkt $\mathbf{x} = (x, y)^T$ auf einer Linie liegt, ist durch die allgemeine Form der Geradengleichung gegeben: $ax + by + c = 0$. Sei die Linie \mathbf{l} definiert durch den Vektor $\mathbf{l} = (a, b, c)^T$, so ist die linke Seite der Geradengleichung nichts anderes als das Skalarprodukt zwischen dem Linienvektor \mathbf{l} und dem Punktvektor: $\tilde{\mathbf{x}} = (\mathbf{x}^T, 1)^T$:

$$\mathbf{l}^T \tilde{\mathbf{x}} = 0 \quad . \tag{2.1}$$

Eine Gerade ist durch zwei Punkte eindeutig definiert. Würde man den herkömmlichen Weg gehen, so müßte man dazu ein Gleichungssystem bestehend aus zwei Geradengleichungen lösen. Mit projektiven Koordinaten geht dies jedoch einfacher. Sei \mathbf{l} die Gerade, welche durch die Punkte $\tilde{\mathbf{x}}$ und $\tilde{\mathbf{y}}$ geht. Es gilt sowohl $\mathbf{l}^T \tilde{\mathbf{x}} = 0$ als auch $\mathbf{l}^T \tilde{\mathbf{y}} = 0$. Der Vektor \mathbf{l} steht also sowohl auf $\tilde{\mathbf{x}}$ als auch auf $\tilde{\mathbf{y}}$ senkrecht. Der Vektor, welches dieses leistet ist das Vektorprodukt aus $\tilde{\mathbf{x}}$ und $\tilde{\mathbf{y}}$:

$$\mathbf{l} \sim \tilde{\mathbf{x}} \times \tilde{\mathbf{y}} \quad . \tag{2.2}$$

Gleichermaßen ist ein Punkt $\tilde{\mathbf{x}}$ eindeutig durch den Schnitt zweier Linien \mathbf{l}_1 und \mathbf{l}_2 definiert, und es ergibt sich:

$$\tilde{\mathbf{x}} \sim \mathbf{l}_1 \times \mathbf{l}_2 \quad . \tag{2.3}$$

Man sieht, daß die Rolle von Punkten und Linien vertauschbar ist. Dies nennt man die Dualität von Punkten und Linien in der projektiven Ebene.

Werfen wir noch einen Blick auf die geometrische Bedeutung der einzelnen Elemente eines Linienvektors. Die Geradengleichung wird allgemeiner auch als Hesse Normalenform bezeichnet, wobei der Vektor (a, b) die Normale der Geraden ist. Der Abstand der Geraden zum Ursprung ist dann: $\frac{|c|}{\sqrt{a^2+b^2}}$.

Die Linie im Unendlichen

Punkte im Unendlichen, auch ideale Punkte genannt, haben die Form: $\tilde{x}_\infty = (x, y, 0)^T$. Alle idealen Punkte liegen auf einer einzigen Linie, der Linie im Unendlichen: $\mathbf{l}_\infty = (0, 0, 1)^T$. Ideale Punkte spielen die Rolle von Richtungen. Schneidet man eine Linie mit der Linie im Unendlichen, so ergibt sich ein idealer Schnittpunkt: $\mathbf{l} \times \mathbf{l}_\infty = (b, -a, 0)^T$. Die ersten beiden Komponenten $(b, -a)$ sind der Richtungsvektor der Geraden.

2.2.3 Transformationen und ihre Hierarchie

Die im folgenden beschriebenen Transformationen werden für die projektive Ebene erläutert. Alle Eigenschaften gelten aber natürlich ebenso für den höherdimensionalen Fall, d.h. den projektiven Raum. Der Reihe nach werden die projektive, affine, ähnliche und euklidische Transformation behandelt und bilden so eine Hierarchie von Transformationen. Die projektive ist die allgemeinste und die euklidische die spezielleste Transformation. Die in dieser Reihe folgende Transformation ist eine Untergruppe der vorangehenden, das heißt eine euklidische ist auch eine ähnliche Transformation, eine ähnliche ist auch eine affine Transformation und eine affine ist auch eine projektive Transformation. Umgekehrt ist aber zum Beispiel nicht jede projektive auch eine affine Transformation. Für jede Gruppe von Transformationen ist es wichtig diejenigen geometrischen Größen zu kennen, welche durch die jeweilige Transformation nicht verändert werden. Man spricht von sogenannten Invarianten. Bezogen auf die Hierarchie der Transformationen nimmt die Anzahl der Invarianten von der projektiven zur euklidischen Transformation zu. Jede Invariante der allgemeineren Gruppe ist auch gleichzeitig Invariante der Untergruppe.

Projektive Transformation

Eine projektive Transformation bildet Punkte auf Punkte und Linien auf Linien ab; sie ist invertierbar. Daher ist eine projektive Transformation als lineare Abbildung in den homogenen Koordinaten zu schreiben:

$$\tilde{\mathbf{x}}' \sim \mathbf{H}\tilde{\mathbf{x}} \quad . \quad (2.4)$$

\mathbf{H} ist eine reguläre 3×3 Matrix, so daß die Abbildung invertierbar ist. Eine projektive Transformation wird auch Homographie genannt (daher auch die Bezeichnung \mathbf{H}). Die obige Gleichung ist homogen, das heißt alle Matrizen, welche sich nur durch einen Skalierungsfaktor unterscheiden, sind äquivalent. Die Matrix enthält 9 Elemente. Abzüglich des Skalierungsfaktors ergeben sich 8 Freiheitsgrade.

Die einzige Invariante einer projektiven Transformation ist das Doppelverhältnis. Seien A, B, C, D Punkte auf einer Geraden, so ist ein Verhältnis von Längen zum Beispiel $\frac{|AB|}{|AC|}$, wobei $|AB|$ die Länge der Strecke von Punkt A zu Punkt B ist. Das Doppelverhältnis der vier Punkte ist dann: $\frac{|AB||CD|}{|AC||BD|}$. Durch eine Homographie können unendlich weit entfernte Punkte auf endliche Punkte abgebildet werden. In der folgenden Abbildung sieht man, daß der Schnittpunkt von vormaligen parallelen Geraden ein endlicher Punkt ist (Fluchtpunkt).

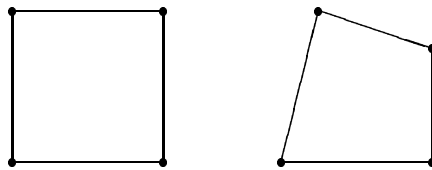


Abbildung 2.1: Variation eines Quadrats unter einer projektiven Transformation \mathbf{H}_p .

Affine Transformation

In inhomogenen Koordinaten schreibt sich eine affine Transformation wie folgt: $\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{t}$. Die beiden Teile, Multiplikation mit \mathbf{A} und die Addition von \mathbf{t} , vereinfachen sich in der homogenen Schreibweise zu einer einzigen Multiplikation:

$$\tilde{\mathbf{x}}' \sim \mathbf{H}_A \tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \tilde{\mathbf{x}} \quad . \quad (2.5)$$

Die Matrix \mathbf{A} ist hierbei regulär. Die affine Transformation bildet endliche auf endliche Punkte und unendliche auf unendliche Punkte ab. Schreiben wir obige Gleichung noch einmal explizit in ihren Komponenten auf:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} \sim \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} . \quad (2.6)$$

Es ist ersichtlich, daß die affine Transformation 6 Freiheitsgrade (FG) hat: zwei FG für die Translation, ein FG für die Rotation, zwei FG für die Skalierung in x - und y -Richtung und ein FG für die Scherung. Die Werte für Rotation, Skalierung und Scherung stecken implizit in den Elementen von \mathbf{A} . Die affine Transformation weist zusätzlich zum Doppelverhältnis der projektiven Transformation noch folgende Invarianten auf: Parallelität von Geraden bleibt erhalten, Längenverhältnisse auf Geraden und Flächenverhältnisse bleiben konstant.

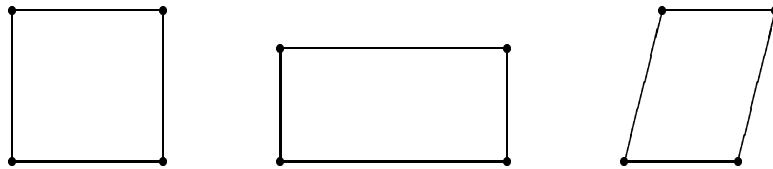


Abbildung 2.2: Variation eines Quadrats unter einer affinen Transformation \mathbf{H}_A .

Ähnlichkeitstransformation

Die Ähnlichkeitstransformation setzt sich aus Translation, Drehung und Skalierung zusammen und enthält somit 4 Freiheitsgrade:

$$\tilde{x}' \sim \mathbf{H}_S \tilde{x} = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \tilde{x} . \quad (2.7)$$

\mathbf{R} ist eine Rotationsmatrix und in den Komponenten geschrieben ergibt sich:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} \sim \begin{pmatrix} s\cos(\alpha) & s\sin(\alpha) & t_x \\ -s\sin(\alpha) & s\cos(\alpha) & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} , \quad (2.8)$$

wobei α der Drehwinkel ist. Unter der Ähnlichkeitstransformation sind Winkel und beliebige Längenverhältnisse invariant.

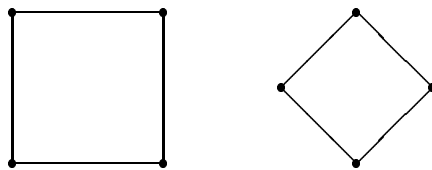


Abbildung 2.3: Variation eines Quadrats unter einer Ähnlichkeitstransformation \mathbf{H}_S .

Euklidische Transformation

Die euklidische Transformation besteht nur aus einer Translation und einer Rotation und besitzt daher 3 Freiheitsgrade:

$$\tilde{x}' \sim \mathbf{H}_S \tilde{x} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \tilde{x} \quad . \quad (2.9)$$

Längen und Flächen sind unter der euklidischen Transformation invariant.

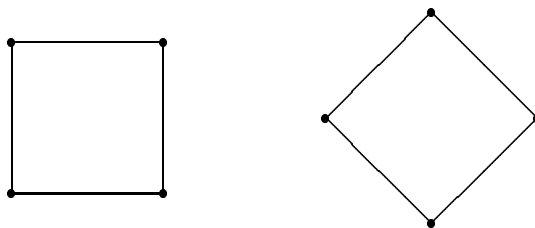


Abbildung 2.4: Variation eines Quadrats unter einer euklidischen Transformation \mathbf{H}_E .

2.2.4 Berechnung einer Homographie

Eine Homographie besitzt 8 Freiheitsgrade. Das heißt, zur Bestimmung einer Homographie müssen 8 Gleichungen bereitgestellt werden. Eine Punktkorrespondenz wie in Gleichung 2.4 liefert 2 Gleichungen (3 Gleichungen minus des Skalierungsfaktors). Folglich sind 4 Punktkorrespondenzen ausreichend, um die Homographie zu berechnen.

Schreiben wir die Homographie \mathbf{H} in ihren drei Zeilenvektoren, so ergibt sich:

$$\tilde{x}' \sim \begin{pmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{pmatrix} \tilde{x} \quad . \quad (2.10)$$

Inhomogene Koordinaten bedeuten auch Verhältniskoordinaten. Bildet man das Verhältnis $\frac{x'}{z'}$ und $\frac{y'}{z'}$:

$$\begin{aligned} \frac{x'}{z'} &= \frac{\mathbf{h}_1^T \tilde{x}}{\mathbf{h}_3^T \tilde{x}} \quad \wedge \quad \frac{y'}{z'} = \frac{\mathbf{h}_2^T \tilde{x}}{\mathbf{h}_3^T \tilde{x}} \quad \Leftrightarrow \\ z' \mathbf{h}_1^T \tilde{x} - x' \mathbf{h}_3^T \tilde{x} &= 0 \quad \wedge \quad z' \mathbf{h}_2^T \tilde{x} - y' \mathbf{h}_3^T \tilde{x} = 0 \quad \Leftrightarrow \\ \begin{pmatrix} z' \tilde{x}^T & \mathbf{0} & -x' \tilde{x}^T \\ \mathbf{0} & z' \tilde{x}^T & -y' \tilde{x}^T \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} &= \mathbf{0} \quad . \end{aligned} \quad (2.11)$$

Kompakt schreibt sich dieses Gleichungssystem als $\mathbf{A}_i \mathbf{h} = 0$, wobei \mathbf{A}_i die bekannten Koordinaten der i -ten Korrespondenz enthält. Der Lösungsvektor $\mathbf{h} = (\mathbf{h}_1^T, \mathbf{h}_2^T, \mathbf{h}_3^T)^T$ enthält die gesuchten Elemente der Homographie. Schreibt man für 4 Korrespondenzen die Gleichung $\mathbf{A}_i \mathbf{h} = 0$ untereinander, so erhält man $\mathbf{A} \mathbf{h} = 0$, wobei \mathbf{A} eine 8×9 Matrix ist. \mathbf{A} hat den Rang 8 und somit gibt es eine nicht triviale Lösung für \mathbf{h} (die triviale Lösung ist $\mathbf{h} = \mathbf{0}$). Für ideale, das heißt rauschfreie Messungen in den Punktkoordinaten gibt es auch für mehr als 4 Korrespondenzpaare eine Lösung für $\mathbf{A} \mathbf{h} = 0$, da der Rang von \mathbf{A} immer noch 8 beträgt. Da jede Messung aber rauschbehaftet ist, gilt dies für reale Punkte nicht mehr. Vielmehr wird nach einer bestmöglichen Lösung gesucht, welcher der ursprünglichen Gleichung am nächsten kommt. Die bestmögliche Lösung für \mathbf{h} ist dann diejenige, welche den quadratischen Fehler $\|\mathbf{A} \mathbf{h}\|$, also den Abstand zwischen $\mathbf{A} \mathbf{h}$ und dem Nullvektor, minimiert. Die Lösung für \mathbf{h} kann formal folgendermaßen aufgeschrieben werden:

$$\mathbf{h} = \arg \min_{\mathbf{h}} \|\mathbf{A} \mathbf{h}\| \quad \text{unter der Bedingung} \quad \|\mathbf{h}\| = 1 \quad . \quad (2.12)$$

Die Bedingung $\|\mathbf{h}\| = 1$ ist notwendig um die triviale Lösung auszuschließen. Diese Methode zur Berechnung von \mathbf{H} wird *Direct-Linear-Transformation* genannt. Das rein numerische Verfahren zur Lösung ist als kleinste Fehlerquadrat Methode bekannt (siehe auch Abschnitt 2.5.1).

2.3 Kameramodell

Dieser Praktikumsversuch beschränkt sich auf die Modellierung von normalen Video- oder Industriekameras. Das einfachste Modell ist die sogenannte Lochkamera oder perspektivische Kamera. Auch der Begriff Zentralprojektion ist hierfür gebräuchlich. Das Modell der Lochkamera wird durch die Modellierung von Linsenverzerrungen erweitert, welche bei einer großen Anzahl von Kameras auftreten.

Weitere Kameratypen, welche hier nicht behandelt werden, sind die Panorama- oder omnidirektionalen Kameras. Auch die Bildsensoren von Satelliten sind anders geartet und unter dem Begriff Pushbroom-Kameras bekannt.

2.3.1 Die Lochkamera

In einer Lochkamera werden Weltpunkte mittels Zentralprojektion auf die Bildpunkte abgebildet. Wie in der Abbildung 2.5 zu sehen ist, liegen Weltpunkt und Bildpunkt auf einem Strahl, welcher vom Kamerazentrum ausgeht. Das Kamerazentrum liegt im Ursprung des Koordinatensystems der Kamera. Punkte in diesem Koordinatensystem werden mit $\mathbf{X}_C = (X_C, Y_C, Z_C)^T$ bezeichnet. Die Bildebene ist die Ebene $Z_C = f$, wobei f die Brennweite ist. Die optische Achse ist durch den Richtungsvektor $(0, 0, 1)$ gegeben, und der Punkt, wo die optische Achse die Bildebene durchsticht, ist der Hauptpunkt mit den Koordinaten $(0, 0, f)$. Der Bildpunkt $\mathbf{x} = (x, y)^T$ hat die 3D Koordinaten $(x, y, f)^T$. Gemäß des Strahlensatzes gelten dann folgende Verhältnisse:

$$\frac{x}{X_C} = \frac{y}{Y_C} = \frac{f}{Z_C}$$

Für die Abbildung von Weltpunkten auf Bildpunkte gilt dann:

$$(X_C, Y_C, Z_C) \rightarrow \left(\frac{f}{Z_C} X_C, \frac{f}{Z_C} Y_C \right) .$$

Durch Überführung in homogene Koordinaten erhalten wir: $(\frac{f}{Z_C} X_C, \frac{f}{Z_C} Y_C, 1) \sim (f X_C, f Y_C, Z_C)$. Die obige nichtlineare Gleichung läßt sich dann als einfache lineare Gleichung schreiben:

$$\begin{pmatrix} f X_C \\ f Y_C \\ Z_C \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{pmatrix} \Leftrightarrow \tilde{\mathbf{x}} \sim \mathbf{P} \tilde{\mathbf{X}}_C . \quad (2.13)$$

Die Matrix \mathbf{P} kann in das Produkt einer Diagonalmatrix \mathbf{K} und einer Einheitsprojektionsmatrix \mathbf{P}_p aufgespalten werden:

$$\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \Leftrightarrow \mathbf{P} = \mathbf{K}\mathbf{P}_p .$$

Die Einheitsprojektionsmatrix \mathbf{P}_p kann auch als Aneinanderreihung der 3×3 Einheitsmatrix \mathbf{I} und des Nullvektors $\mathbf{0}$ geschrieben werden: $\mathbf{P}_p = (\mathbf{I}|\mathbf{0})$.

Schlußendlich werden Punkte nicht in dem Kamerakoordinatensystem, sondern in dem Bildkoordinatensystem angegeben. Dort ist die physikalisch Einheit ein Pixel. Ist der Bildvektor \mathbf{x} eine Größe in Pixel, so hat auch die Brennweite f die Einheit Pixel. Normalerweise ist es so, daß die optische Achse nicht auf den Ursprung des Bildkoordinatensystems abgebildet wird. Die ursprünglichen Bildkoordinaten (x, y) erhalten einen Offset (x_0, y_0) durch den Bildhauptpunkt: $(x, y) \rightarrow (x, y) + (x_0, y_0)$. Die Matrix \mathbf{K} wird dann zu:

$$\mathbf{K} = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} . \quad (2.14)$$

Im Falle einer realen Kamera kann es vorkommen, daß der Bildsensor aus nicht quadratischen Pixeln aufgebaut ist. Das bedeutet, daß das Bild in vertikaler und horizontaler Richtung unterschiedlich skaliert ist. Ein Quadrat wird dann auf ein Rechteck abgebildet. Folglich ergibt sich:

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} . \quad (2.15)$$

Manchmal kommt es vor, daß die optische Achse des Objektivs nicht senkrecht auf dem Bildsensor steht. In diesem Fall tritt eine Scherung auf und rechte Winkel erscheinen leicht schiefwinkelig im Bild. Dieser Einfluß wird durch den Scherungsparameter s ausgedrückt:

$$\mathbf{K} = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} . \quad (2.16)$$

Normalerweise werden Weltpunkte nicht relative zu dem Kamerakoordinatensystem beschrieben, sondern relative zu einem eigenen Weltkoordinatensystem. Kamera- und

Weltkoordinatensystem sind durch eine euklidische Transformation miteinander verbunden: $\mathbf{X}_C = \mathbf{R}\mathbf{X} + \mathbf{t}$. Die Projektionsmatrix \mathbf{P} läßt sich dann schreiben als:

$$\mathbf{P} = \mathbf{K}(\mathbf{I}|\mathbf{0}) \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} = \mathbf{K}(\mathbf{R}|\mathbf{t}) \quad . \quad (2.17)$$

Die *internen Kameraparameter* werden durch die Kalibriermatrix \mathbf{K} beschrieben. Hier sind alle Größen über die Kamera als solche enthalten. Die *externen Kameraparameter* beschreiben die Transformation von Welt- in Kamerakoordinaten in Form der Rotationsmatrix \mathbf{R} und des Translationsvektors \mathbf{t} .

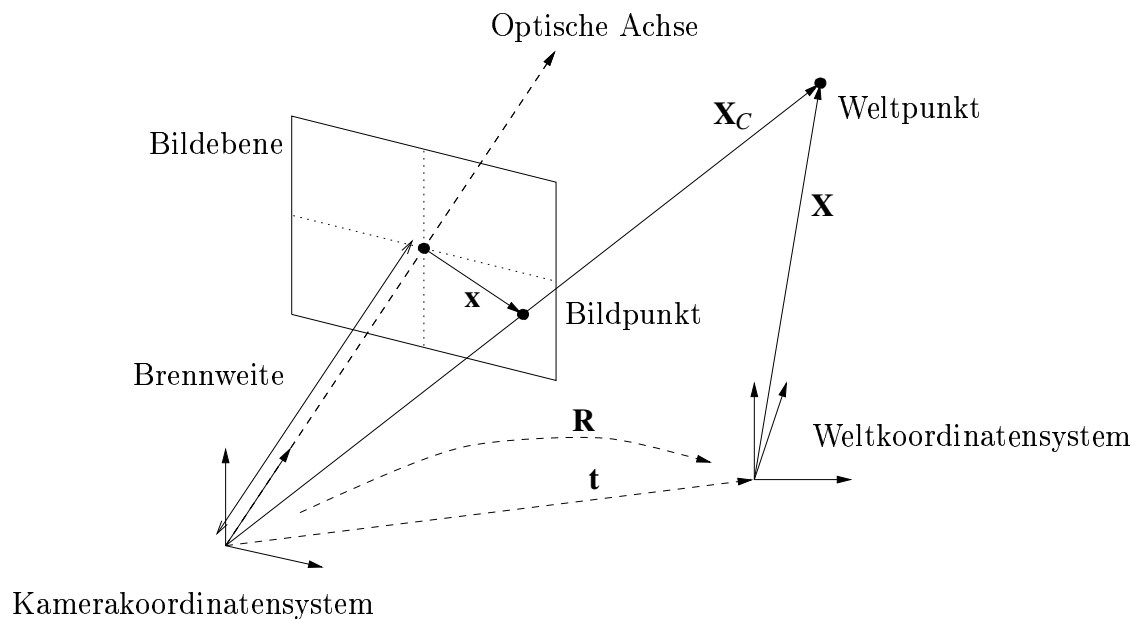


Abbildung 2.5: Modell der Lochkamera

2.3.2 Linsenverzerrungen

Bisher wurde angenommen, daß ein lineares Modell geeignet ist um reale Kameras nachzubilden. In einem linearen Modell, also einer projektive 3D zu 2D Transformation, liegen Weltpunkt, Bildpunkt und optisches Zentrum auf demselben optischen Strahl.

Weiterhin werden Geraden als Geraden abgebildet. Bei vielen realen Kameras trifft dies allerdings nicht zu. Besonders dann, wenn die Brennweite klein ist, das heißt das Blickfeld ist groß (Weitwinkelobjektiv), werden Geraden auf krumme Linien abgebildet. Oft treten Linsenverzerrungen bei eher billigen Objektiven auf. In der Regel sind die auftretenden Linsenverzerrung radiale Linsenverzerrungen, das heißt, das Maß der Verzerrung ist abhängig von dem Abstand des Bildpunktes zum Bildmittelpunkt. Der Effekt der Linsenverzerrung ist in der Abbildung 2.6 dargestellt. In diesem Fall erfährt das Quadrat eine konvexe Deformation; eine konkave Deformation kann aber ebenfalls auftreten.

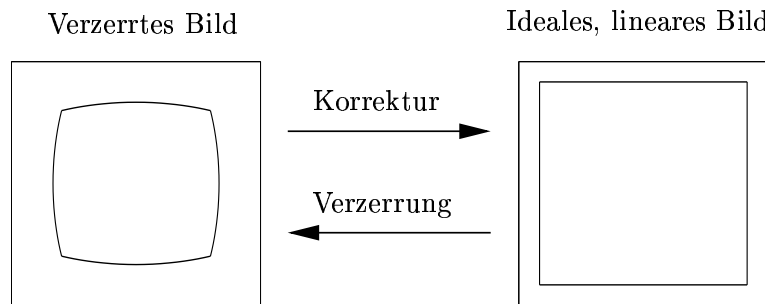


Abbildung 2.6: Effekt einer radialen Linsenverzerrung bei der Abbildung eines Quadrats.

Seien $\mathbf{x} = (x, y)^T$ die idealen, linearen Bildkoordinaten und $\mathbf{x}_d = (x_d, y_d)^T$ die realen, verzerrten Bildkoordinaten. Das Zentrum der Verzerrung sei der Bildhauptpunkt \mathbf{x}_0 . Die Linsenverzerrung kann dann folgendermaßen modelliert werden:

$$(\mathbf{x}_d - \mathbf{x}_0) = L(r)(\mathbf{x} - \mathbf{x}_0) \quad , \quad (2.18)$$

wobei r der Abstand von \mathbf{x} zum Bildhauptpunkt ist: $r = \|\mathbf{x} - \mathbf{x}_0\|$. Die Linsenfunktion L nimmt für den Bildhauptpunkt den Wert eins an: $L(0) = 1$. Ansonsten ist L unbekannt und wird daher durch einer Taylorreihenentwicklung um $r = 0$ approximiert: $L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \dots$. Angenommen eine quadratische Approximation ist ausreichend genau, obige Formel schreibt sich dann wie folgt:

$$\mathbf{x}_d = \mathbf{x} + (\mathbf{x} - \mathbf{x}_0) (\kappa_1 r + \kappa_2 r^2) \quad . \quad (2.19)$$

Kamerakalibrierung bedeutet also nicht nur die Bestimmung der Elemente der Kalibriermatrix \mathbf{K} , sondern auch der Verzerrungsparameter $\{\kappa_1, \kappa_2, \kappa_3, \dots\}$.

Anmerkung

Sei \mathbf{x}' der virtuelle Bildpunkt, welcher bei der Abbildung mit einer Einheitskalibriermatrix $\mathbf{K} = \mathbf{I}$ entsteht: $\tilde{x}' \sim (\mathbf{R} | \mathbf{t}) \tilde{X}$. Der tatsächliche Bildpunkt \tilde{x} entsteht dann durch: $\tilde{x} \sim \mathbf{K} \tilde{x}'$. Die oben beschriebene Modellierung der Linsenverzerrung kann auf die Punkte \mathbf{x} oder \mathbf{x}' angewendet werden. Rein theoretisch betrachtet ist dies äquivalent. Unter numerischen Gesichtspunkten aber ist die Modellierung der Verzerrung bezüglich der Punkte \mathbf{x}' günstiger als die der Punkte \mathbf{x} . Die Begründung dafür ist analog zu den Überlegungen bei der Normalisierung von Daten, welche im Abschnitt 2.5.2 behandelt wird. Für eine Modellierung mit n Verzerrungsparametern und einem repräsentativen Bildpunkt von $\mathbf{x} = (100, 100)$, ist $r^n = \|\mathbf{x} - \mathbf{x}_0\|^n$ in der Größenordnung von 1 bis 10^{2n} . Benutzt man stattdessen den virtuellen Bildpunkt mit $r' = \|\mathbf{x}'\|$, so ist r^n typischerweise kleiner Eins. Numerische Instabilitäten bei der nicht linearen Optimierung von mehr als zwei Verzerrungsparametern werden dadurch vermieden.

2.3.3 Projektion einer Ebene

Von besonderem Interesse ist die Projektion einer Ebene in die Bildebene, da dies Teil des 2D Kalibriervorgangs ist, welcher im nächsten Kapitel behandelt wird. Sei die Ebene die X/Y -Ebene in Weltkoordinaten. Die Gleichung der Ebene ist dann: $Z = 0$. Die Rotationsmatrix \mathbf{R} läßt sich in ihren drei Spaltenvektoren auch als $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ schreiben. Für die Abbildung von Punkten einer Ebene ergibt sich dann:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \mathbf{K}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{t}) \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}) \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} . \quad (2.20)$$

Bezeichnen wir mit $\tilde{X}_E = (X, Y, 1)^T$ den 2D projektiven Punkt in der besagten Weltebene. Man erkennt, daß die projektive Abbildung der Kamera von 3D Weltpunkten nach 2D Bildpunkten nun die Form einer 2D zu 2D Abbildung erhält:

$$\tilde{x} \sim \mathbf{H} \tilde{X}_E \quad \text{mit} \quad \mathbf{H} = \mathbf{K}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}) . \quad (2.21)$$

Auf der Basis von mindestens vier Punktkorrespondenzen kann die Homographie \mathbf{H} berechnet werden. Ausgehend davon können Bedingungen für die Kalibriermatrix \mathbf{K} abgeleitet werden und zur Bestimmung dieser genutzt werden.

Kennt man bereits die Kamera \mathbf{K} so kann mittels der Homographie die Position und Orientierung der Ebene berechnet werden. Mit $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3)$ gilt:

$$(\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3) = \lambda \mathbf{K}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}) \quad . \quad (2.22)$$

Die Rotationsmatrix $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ ist eine orthogonale Matrix mit $\|\mathbf{r}_i\| = 1$ und $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$. \mathbf{K} ist definitionsgemäß invertierbar und mit $\lambda = \|\mathbf{K}^{-1}\mathbf{h}_1\|$ folgt:

$$(\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}) = \mathbf{K}^{-1}\mathbf{H}/\|\mathbf{K}^{-1}\mathbf{h}_1\| \quad . \quad (2.23)$$

Das heißt, daß auf der Basis von vier bekannten Punkten auf einer Ebene und einer kalibrierten Kamera die Lage dieser Ebene im Raum berechnet werden kann.

2.4 Kalibrierung

2.4.1 Einleitung und Überblick

Dieses Kapitel gliedert sich in drei Abschnitte. Im ersten Abschnitt wird das Standardverfahren für dreidimensionale Kalibriermuster vorgestellt. Eine lineare und eine nicht lineare Methode wird beschrieben. Selbst wenn diese Methoden im Rahmen dieses Versuchs keine praktische Umsetzung finden, sind sie von Bedeutung und wichtig für das allgemeine Verständnis des Themenkomplexes. Im zweiten Abschnitt wird dann eine Methode vorgestellt, welche den Vorteil einer einfachen Umsetzung aufweist: Nur ein einziges ebenes Kalibriermuster muß erstellt werden. Auf Grund dieses enormen praktischen Vorteils wird diese Methode auch im praktischen Teil implementiert werden. Zur theoretischen Untermauerung des 2D Kalibrierverfahrens wird im letzten Abschnitt das Thema der Selbstkalibrierung kurz angeschnitten.

Das Verfahren nach Tsai wird noch häufig in der Literatur zitiert und in der Praxis auch verwendet. Es wurde im Jahr 1987 veröffentlicht und hat seitdem eine weite Verbreitung erfahren, auch weil lauffähiger Programmcode dafür im Netz zu finden ist. Vom Stand der Forschung aus betrachtet ist dieses Verfahren jedoch veraltet. Für die 3D Kalibrierung ist das hier vorgestellte Verfahren wesentlich übersichtlicher und sollte daher verwendet werden. Oft ist zu lesen, daß das Tsai-Verfahren auch mit nur einer Kalibrierebene angewendet werden kann. Dies ist nur teilweise richtig, da als Voraussetzung die Pixelgröße des Sensors bekannt sein muß. Ohne diese Vorkenntnis ist es nicht möglich eine Kalibrierung mit nur einer Ebene durchzuführen, was in diesem Kapitel auch gezeigt wird.

2.4.2 Kalibriermuster

Ein Kalibriermuster kann aus einer Kalibrierebene (2D Kalibrierung), oder mehreren Kalibrierebenen (3D Kalibrierung) bestehen. Die Kalibriermarkierungen selbst bestehen aus schwarz-weiß Mustern, wobei exakt bekannte Punkte auf den Mustern gut zu lokalisierende Punktmerkmale im Bild erzeugen. Im einfachsten Fall wird das Muster auf Papier gedruckt und dann auf einer soliden Platte aufgezogen. Um hochpräzise Kalibriermuster zu erhalten, können die Muster auch mit Hilfe von CNC Maschinen in Metall gefräßt werden.

Die einzelnen Markierungen können unterschiedliche Formen haben. Wie in der Abbildung 2.7(a) zu sehen, besteht hier das Muster aus regelmäßig angeordneten Quadraten. Typischerweise liegen die Muster auf einem rechtwinkligen Gitternetz mit identischen Abständen in den Gitterpunkten. Mit Hilfe eines Kantendetektors werden zuerst die Kanten der Quadrate bestimmt. An diese Kanten werden Linien angepaßt. Die Schnittpunkte der Linien bilden die Eckpunkte der Quadrate. Dies hat den Vorteil, daß pro Quadrat vier Meßpunkte zur Verfügung stehen. Bei starker Linsenverzerrung ist allerdings zu beachten, daß die Bestimmung der Eckpunkte als Schnitt der Linien ungenau wird.

Eine andere Möglichkeit, welche auch im Rahmen dieses Versuches verwendet wird, ist ein regelmäßiges Muster bestehend aus Kreisen oder Kreisscheiben, siehe Abbildung 2.7(b). Als Nachteil ist die geringere Anzahl an Markierungspunkten zu nennen. Der Vorteil ist die einfache und robuste Extraktion der Markierungsmerkmale: Das Markierungsmerkmal ist der Kreismittelpunkt. Mit Hilfe einer einfachen Segmentierung werden die Kreisscheiben vom Hintergrund getrennt und deren Schwerpunkte berechnet (Siehe auch Abschnitt 2.4.4).

Zum Schluß noch eine Überlegung bezüglich der notwendigen Anzahl an Kalibriermarkierungen. Einmal ist es einsichtig, daß das Kalibriermuster möglichst den Großteil des Bildes abdecken sollte. Wird zum Beispiel das Kalibriermuster so weit entfernt von der Kamera aufgestellt, daß es nur den zentralen Bildteil abdeckt, so wird die Kalibrierung bezüglich der Verzerrungsparameter schlecht ausfallen, weil die Linsenverzerrung sich zum Rand des Bildes hin stärker ausprägt. Weiterhin gibt es eine Daumenregel, welche besagt, daß die Anzahl an notwendigen Korrespondenzen die Anzahl der zu bestimmenden Parameter um den Faktor fünf übersteigen sollte. Das heißt, das Kalibriermuster sollte circa 30 bis 40 Markierungen aufweisen.

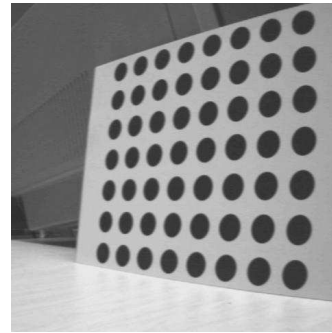
2.4.3 3D Kalibrierung

3D Kalibriermuster

Für eine komplette Kalibrierung ist es notwendig, daß die Punkte des Kalibrierungsmusters nicht alle auf einer Ebene liegen. Ein Kalibrierungsmuster besteht daher typischerweise aus zwei oder drei ebenen Flächen, auf die Kalibrierungsmarkierungen aufgebracht sind. In der folgenden Abbildung ist der Fall von zwei Ebenen zu sehen. Der Winkel zwischen den beiden Ebenen muß dabei genau vermessen werden.



(a) 3D Kalibriermuster



(b) 2D Kalibriermuster

Abbildung 2.7: Dreidimensionales Kalibrierungsmuster bestehend aus zwei einzelnen schiefwinkligen Ebenen (a) und ein einfaches 2D Kalibrierungsmuster (b).

Eine weitere Möglichkeit nicht koplanare 3D Punkte zu erhalten ist die Verwendung einer einzelnen Kalibrierfläche, welche mit Hilfe einer mechanischen Einrichtung exakt in Richtung der Normalen verschoben wird. Der Nachteil hierbei ist der erhöhte Aufwand, welcher durch die notwendige Verschiebeeinrichtung entsteht.

Ein linearer Algorithmus

Nehmen wir an, daß eine Anzahl von Korrespondenzen $\tilde{X}_j \leftrightarrow \tilde{x}_j$ zwischen Weltpunkten \tilde{X}_j und Bildpunkten \tilde{x}_j gegeben ist. Das Modell einer projektiven Kamera ist die 3×4 Projektionsmatrix \mathbf{P} . Seien $\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T$ die drei Zeilenvektoren von \mathbf{P} , so läßt sich

schreiben:

$$\tilde{x}_j \sim \mathbf{P}\tilde{X}_j = \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{pmatrix} \tilde{X}_j \quad . \quad (2.24)$$

In dem Kontext der Kamerakalibrierung sind die Punkte \tilde{x}_j tatsächliche Messungen in dem Bild und liegen somit im Endlichen: \tilde{x}_j läßt sich schreiben als $\tilde{x}_j = (x_j, y_j, 1)^T$. Schreibt man die Äquivalenz $\tilde{x}_j \sim \mathbf{P}\tilde{X}_j$ als Gleichung $(\lambda x_j, \lambda y_j, \lambda)^T = \mathbf{P}\tilde{X}_j$, so folgt:

$$x_j = \frac{\mathbf{p}_1^T \tilde{X}_j}{\mathbf{p}_3^T \tilde{X}_j} \quad , \quad y_j = \frac{\mathbf{p}_2^T \tilde{X}_j}{\mathbf{p}_3^T \tilde{X}_j} \quad . \quad (2.25)$$

Diese Gleichungen lassen sich in folgendes Gleichungssystem umformen:

$$\begin{pmatrix} -\tilde{X}_j^T & \mathbf{0}^T & x_j \tilde{X}_j^T \\ \mathbf{0}^T & -\tilde{X}_j^T & y_j \tilde{X}_j^T \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{A}_j \mathbf{p} = \mathbf{0} \quad . \quad (2.26)$$

Beachten Sie die Ähnlichkeit dieser Gleichung mit der Gleichung 2.11. Die Elemente der 2×12 Matrix \mathbf{A}_j bestehen nur aus den Koordinaten der Bildmessung und der bekannten Koordinaten des Kalibrierungsmusters. Der Lösungsvektor $\mathbf{p} = (\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T)^T$ besteht aus den 12 Elementen von \mathbf{P} , welche zu bestimmen sind. Stehen n Korrespondenzen zur Verfügung, so kann man n Gleichungssysteme dieser Form übereinander stapeln und erhält das Gleichungssystem $\mathbf{A}\mathbf{p} = \mathbf{0}$, wobei \mathbf{A} eine $2n \times 12$ Matrix ist. Da die Matrix \mathbf{P} bis auf einen Skalierungsfaktor definiert ist, enthält sie $12 - 1 = 11$ Unbekannte. 11 Gleichungen sind für eine eindeutige Lösung in \mathbf{p} notwendig. Jede Korrespondenz liefert zwei Gleichungen. Somit sind 6 Korrespondenzen notwendig, wobei eine Gleichung der insgesamt 12 bereitgestellten Gleichung ungenutzt bleiben kann.

Jede reale Messung ist mit Rauschen behaftet, und somit gilt nicht mehr die Projektionsgleichung $\tilde{x}_j = \lambda \mathbf{P}\tilde{X}_j$. Daraus folgt, daß auch das Gleichungssystem $\mathbf{A}\mathbf{p} = \mathbf{0}$ für $n \geq 6$ Punkt Korrespondenzen keine nicht triviale Lösung mehr besitzt (Die triviale Lösung ist $\mathbf{p} = \mathbf{0}$). Stattdessen wird im Sinne des kleinsten Fehlerquadrates (siehe auch Abschnitt 2.5.1) nach einer Lösung gesucht, welche den Fehler $\|\mathbf{A}\mathbf{p}\|$ minimiert. Da die triviale Lösung auszuschließen ist, und da \mathbf{P} und somit auch \mathbf{p} bis auf einen Skalierungsfaktor definiert ist, wird \mathbf{p} auf eins normiert:

$$\mathbf{p} = \arg \min_{\mathbf{p}} \|\mathbf{A}\mathbf{p}\| \quad \text{mit} \quad \|\mathbf{p}\| = 1 \quad . \quad (2.27)$$

Dies ist der *Direct-Linear-Transformation* Algorithmus zur Berechnung der projektiven Kamera \mathbf{P} . Ausgehend von \mathbf{P} können dann die internen Kameraparameter \mathbf{K} und

die externen Kameraparameter \mathbf{R}, \mathbf{t} berechnet werden. Für reale Kameras mit Linsenverzerrungen ist nur die nichtlineare Methode geeignet, welche im nächsten Abschnitt beschrieben wird.

Ein nichtlinearer Algorithmus

Lineare Methoden können immer dann eingesetzt werden, wenn die bestimmenden Gleichungen linear in den gesuchten Größen sind. Sind die bestimmenden Gleichungen nicht linear in den gesuchten Größen, kann zur Lösung auch nur ein nichtlinearer Algorithmus benutzt werden. Vergleichbar mit dem oben beschriebenen linearen Ansatz wird auch der nichtlineare Ansatz in Form eines Minimierungsproblems formuliert. Basis ist daher die geeignete Wahl einer Fehlerfunktion. Für das Problem der Kamerakalibrierung wird der euklidische Abstand d_j zwischen dem gemessenem Bildpunkt \mathbf{x}_j und dem Bild $\hat{\mathbf{x}}_j$ des Weltpunktes \mathbf{X}_j gewählt: $d_j = \|\mathbf{x}_j - \hat{\mathbf{x}}_j\| = \sqrt{(x_j - \hat{x}_j)^2 + (y_j - \hat{y}_j)^2}$, wobei zum Beispiel für die Lochkamera $\hat{\mathbf{x}}_j \sim \mathbf{P}\tilde{\mathbf{X}}_j$ gilt. Das Fehlermaß d_j wird auch als geometrischer Fehler bezeichnet.

Sei \mathbf{p} der Satz an Parametern, welche ein gegebenes Kameramodell beschreiben, so ist der projizierte Punkt \hat{x} eine Funktion dieser Parameter: $\hat{x} = \hat{x}(\mathbf{p}; \mathbf{X})$. Kamerakalibrierung kann dann als Optimierungsproblem über dem Parameterraum \mathbf{p} formuliert werden:

$$\mathbf{p} = \arg \min_{\mathbf{p}} \sum_j d_j^2 = \arg \min_{\mathbf{p}} \sum_j d(\mathbf{x}_j, \hat{x}(\mathbf{p}; \mathbf{X}_j))^2 = \arg \min_{\mathbf{p}} \sum_j \|\mathbf{x}_j - \hat{x}(\mathbf{p}; \mathbf{X}_j)\|^2 \quad . \quad (2.28)$$

Der Parameterraum \mathbf{p} kann zum Beispiel aus den Elementen der projektiven Kamera \mathbf{P} bestehen. Wird die Kamera einschließlich Linsenverzerrung kalibriert, so gilt $\mathbf{p} = (\boldsymbol{\kappa}, \mathbf{K}, \mathbf{R}, \mathbf{t})$. Das Optimierungsproblem selbst wird dann iterativ gelöst, zum Beispiel durch den Levenberg-Marquardt Algorithmus. Jeder iterative Algorithmus bedarf einer Startlösung. Diese wird typischerweise durch eine lineare Lösung bereitgestellt. Da für Kameras mit Linsenverzerrung kein linearer Ansatz hergeleitet werden kann, wird als Startlösung die lineare Lösung der projektiven Kamera benutzt. Ausgehend von \mathbf{P} werden dann $\mathbf{K}, \mathbf{R}, \mathbf{t}$ berechnet. Die erste Iteration startet dann mit Werten von $\boldsymbol{\kappa} = \mathbf{0}$. Im Gegensatz zu einem linearen Ansatz garantiert eine iterative Optimierungsmethode im allgemeinen nicht das Finden der global richtigen Lösung. Abhängig von der verwendeten Startlösung wird nur lokale Optimalität erreicht. In dem hier vorliegenden Fall der Kamerakalibrierung kann allerdings davon ausgegangen werden, daß die lineare Startlösung, welche ja nur die Linsenverzerrung nicht in der Modellierung mit einschließt, globale Optimalität gewährleistet.

2.4.4 2D Kalibrierung

Kalibriervorgang

Wie in Gleichung 2.20 gezeigt wurde, ist die Abbildung von koplanaren Punkten (alle Punkte liegen in einer Ebene) ein besonderer Fall. Die Projektionsmatrix \mathbf{P} kann dann auf eine Homographie \mathbf{H} reduziert werden. Schreibt man die Projektionsmatrix in ihren Spaltenvektoren $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4)$ so ergibt sich daraus für die Homographie: $\mathbf{H} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4)$. Betrachtet man bekannte Kalibrierpunkte auf einer Ebene so kann mit Hilfe dieser 2D-2D Korrespondenzen die Homographie berechnet werden. Eine komplette Kalibrierung von \mathbf{P} ist nicht möglich, da nur 8 der insgesamt 11 Parameter bestimmt werden können. Was passiert nun, wenn man mehrere ebene Kalibriermuster der Kamera präsentiert, wobei die Position und Orientierung der einzelnen Ebenen unbekannt ist? In diesem Abschnitt wird gezeigt, daß drei Ebenen zur Berechnung der internen Kameraparameter ausreichend sind.

Der Einfachheit halber wird für den Kalibriervorgang ein einziges ebenes Kalibriermuster aufgebaut. Dieses wird von drei unterschiedlichen Richtungen aus aufgenommen, wie in folgender Abbildung zu sehen ist.

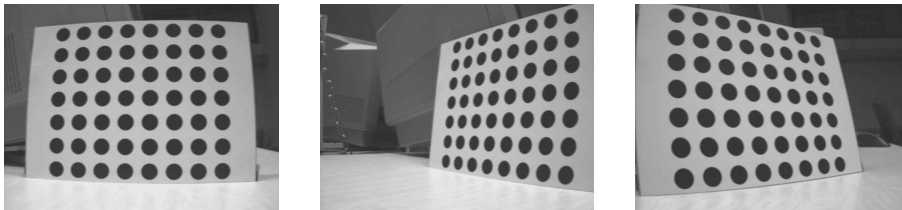


Abbildung 2.8: Drei Aufnahmen eines ebenen Kalibrierusters unter verschiedenen Orientierungen.

Lineare Lösung

Jede Aufnahme der Kalibrierfläche stellt eine Homographie \mathbf{H} zur Verfügung. Schaut man sich noch einmal Gleichung 2.22 an, so ergibt sich für die beiden Vektoren $\mathbf{r}_1, \mathbf{r}_2$:

$$\mathbf{r}_1 = \lambda^{-1} \mathbf{K}^{-1} \mathbf{h}_1 \quad , \quad \mathbf{r}_2 = \lambda^{-1} \mathbf{K}^{-1} \mathbf{h}_2 \quad . \quad (2.29)$$

Die beiden Vektoren $\mathbf{r}_1, \mathbf{r}_2$ sind orthonormal mit $\mathbf{r}_1^T \mathbf{r}_2 = 0$ und $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1$. Eingesetzt in diese beiden Bedingungen ergibt sich:

$$\mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 = 0 \quad (2.30)$$

$$\mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 \quad . \quad (2.31)$$

Dies sind zwei Bedingungen für die internen Kameraparameter in \mathbf{K} . Da die Homographie 8 Freiheitsgrade besitzt und 6 externe Kameraparameter existieren, kann es auch nur zwei Bedingungen für die internen Parameter geben. Im allgemeinen Fall sind 5 interne Kameraparameter zu bestimmen. Dafür sind dann drei Homographien nötig. Ist die Kamera zum Beispiel frei von Scherung, dann reichen zwei Homographien zur Berechnung von \mathbf{K} aus. Die Matrix $\mathbf{K}^{-T} \mathbf{K}^{-1}$ beschreibt das Bild des absoluten Kegelschnitts; mehr dazu ist im Abschnitt zur Selbstkalibrierung (2.4.5) zu finden.

Da die Matrix $\mathbf{B} = \mathbf{K}^{-T} \mathbf{K}^{-1}$ symmetrisch ist, enthält sie 6 unabhängige homogene Elemente:

$$\mathbf{K}^{-T} \mathbf{K}^{-1} = \mathbf{B} = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{pmatrix} \quad , \quad (2.32)$$

welche in dem Vektor $\mathbf{b} = (B_{11}, B_{12}, B_{13}, B_{22}, B_{23}, B_{33})$ zusammengefaßt werden können. Sei der Vektor \mathbf{v}_{ij} folgendermaßen definiert:

$$\mathbf{v}_{ij} = (h_{i,1}h_{j,1}, h_{i,1}h_{j,2} + h_{i,2}h_{j,1}, h_{i,1}h_{j,3} + h_{i,3}h_{j,1}, h_{i,2}h_{j,2}, h_{i,2}h_{j,3} + h_{i,3}h_{j,2}, h_{i,3}h_{j,3})^T \quad .$$

Die Bedingungen 2.30 und 2.31 lassen sich dann schreiben:

$$\begin{pmatrix} \mathbf{v}_{12}^T \\ \mathbf{v}_{11}^T - \mathbf{v}_{22}^T \end{pmatrix} \mathbf{b} = \mathbf{0} \quad . \quad (2.33)$$

Stehen n Homographien zur Verfügung, so erhält man durch Übereinanderstapelung eine $2n \times 6$ Matrix \mathbf{V} , so daß gilt:

$$\mathbf{V} \mathbf{b} = \mathbf{0} \quad . \quad (2.34)$$

Für ideale rauschfreie Messungen ist dieses System eindeutig lösbar. Für rauschbehaftete Messungen ist die Lösung im Sinne des kleinsten Fehlerquadrats (siehe Abschnitt 2.5.1) zu berechnen.

Die Inverse der Kalibriermatrix \mathbf{K} ist ebenfalls eine obere Dreiecksmatrix (alle Elemente unterhalb der Diagonalen sind Null), und sei mit \mathbf{G} bezeichnet. Es gilt, daß das Produkt $\mathbf{G}^T \mathbf{G}$ einer oberen Dreiecksmatrix eine symmetrische und positive definite

Matrix ergibt (Pos. Definitheit bedeutet, daß alle Eigenwerte positiv sind.). Umgekehrt gilt, daß jede symmetrische, positiv definite Matrix sich als Produkt $\mathbf{G}^T \mathbf{G}$ einer oberen Dreiecksmatrix schreiben läßt. Dies ist die sogenannte Cholesky Faktorisierung. Haben wir \mathbf{G} aus \mathbf{B} berechnet, so folgt durch Invertierung die Kalibriermatrix $\mathbf{K} = \mathbf{G}^{-1}$.

Wichtig bei dieser Art der Kalibrierung ist, daß die Kalibrierflächen nicht parallel zueinander sind, da sonst die Rotationsmatrizen \mathbf{R} identisch sind und die Gleichungen 2.30 und 2.31 keine zusätzlichen Bedingungen liefern.

Nichtlineare Lösung

Ausgehend von der oben gewonnenen linearen Lösung kann eine nichtlineare Optimierung angehängt werden, welche die Bestimmung der Linsenverzerrung mit einschließt. Man verwendet wieder den geometrischen Fehler wie in Gleichung 2.28 definiert. Für jede Aufnahme i der Kalibrierfläche gibt es einen Satz von externen Kameraparametern $\mathbf{R}_i, \mathbf{t}_i$. Der Kalibrierpunkt \mathbf{X}_j wird in dem Bild i auf den Bildpunkt $\hat{x}_{ij} = \hat{x}(\mathbf{K}, \mathbf{k}, \mathbf{R}_i, \mathbf{t}_i; \mathbf{X}_j)$ abgebildet. \mathbf{x}_{ij} ist der im Bild gemessene Punkt. Der zu minimierende Gesamtfehler ist dann die Summation über alle Bilder und über alle Bildpunkte:

$$\min_{(\mathbf{K}, \mathbf{k}, \mathbf{R}_i, \mathbf{t}_i)} \sum_i \sum_j \|\mathbf{x}_{ij} - \hat{x}(\mathbf{K}, \mathbf{k}, \mathbf{R}_i, \mathbf{t}_i; \mathbf{X}_j)\|^2 \quad . \quad (2.35)$$

Dieses Problem wird mit einem nichtlinearen Optimierungsalgorithmus, zum Beispiel vom Typ Levenberg-Marquardt, gelöst.

Genauigkeit der Markierungen

Um bewerten zu können, wie gut die berechnete Lösung ist, wird der geometrische Fehler mit der Genauigkeit verglichen, mit welcher die einzelnen Markierungspunkte gemessen wurden. Der Meßfehler ist die untere Grenze für die Genauigkeit der berechneten Lösung.

Sei im Bild eine schwarze Kreisscheibe vor weißem Hintergrund mit dem Radius r gegeben, so gibt es $2\pi r$ Randpixel. Mittels einer Schwellwertbildung wird die Kreisscheibe segmentiert. Diese Diskretisierung verursacht einen Fehler von ± 0.5 Pixeln in den Randpixeln. Die Standardabweichung als Maß für diesen Fehler beträgt dann circa 0.3 Pixel. Berechnen wir nun den Schwerpunkt aller Randpixel, so ist die Standardabweichung des Schwerpunktes $0.3/\sqrt{2\pi r}$. Da die inneren Pixel der Kreisscheibe fehlerfrei segmentiert

werden, ist der Fehler des Flächenschwerpunkts der Kreisscheibe $0.3/\sqrt{2\pi r}(2\pi r)/(\pi r^2)$. Bei einem Radius von 10 Pixeln ergibt sich, daß der Schwerpunkt fast sicher mit einer Genauigkeit von einigen 10^{-2} Pixeln berechnet werden kann.

Weiterhin ist noch der systematische Fehler zu untersuchen, welcher entsteht, wenn Kreise projektiv verzerrt werden. Unter einer allgemeinen projektiven Transformation wird ein Kreis auf eine Ellipse abgebildet. Dabei gilt, daß der Schwerpunkt des transformierten Kreises nicht identisch mit der Transformierten des ursprünglichen Schwerpunkt ist. Das heißt, der berechnete Schwerpunkt ist nicht das exakte Bild des Schwerpunkts auf dem Kalibriermuster. Dieser so entstandene Fehler ist abhängig von dem Abstand Z des Musters zur Kamera und dem Radius R des Kreises. Eine obere Schranke für diesen (relativen) Fehler ist $\frac{R^2}{Z}$. Dies gilt für den Flächenschwerpunkt; die obere Fehlerschranke für den Linienschwerpunkt, das ist der Schwerpunkt der Randpunkte, liegt nur bei $\frac{R}{Z}$. Man sieht auch hier wieder, daß als Markierung der Flächenschwerpunkt dem Linienschwerpunkt vorzuziehen ist. Beträgt zum Beispiel der mittlere Abstand des Kalibrierusters 30cm und der Radius des Kreises 1cm, so liegt der Fehler bei 10^{-3} . Bei der hier eingesetzten Webcam ist der Abbildungsmaßstab circa 1cm zu 10 Pixel. Das heißt, der systematische Fehler durch die projektive Verzerrung beträgt größenordnungsmäßig 10^{-2} Pixel.

2.4.5 Selbstkalibrierung

Die hier vorgestellten Kalibrierverfahren beruhen auf der Kenntnis der Position von 3D Weltpunkten. Jetzt stellt sich natürlich folgende Frage: Ist eine Kalibrierung der Kamera auch dann noch möglich, wenn keine Kenntnis über die Weltpunkte vorliegt? Die Antwort ist ja! Der Vorgang der Berechnung von Kameraparametern ohne Wissen über die 3D Geometrie wird Selbstkalibrierung genannt. Selbstkalibrierung wird immer dann eingesetzt, wenn kein Kalibriermuster zur Verfügung steht. Steht ein Kalibriermuster zur Verfügung, dann sollte dies zur Kalibrierung auch verwendet werden, da die Ausnutzung der zusätzlichen Information in Form der bekannten Weltkoordinaten die Genauigkeit der Kalibrierung erhöht. Auch ist die Implementierung der Kalibrierung dann einfacher.

Nehmen wir an, daß eine Menge von Weltpunkten \tilde{X}_j mittels mehrerer Kameras \mathbf{P}_i auf Bildpunkte \tilde{x}_{ij} abgebildet wird: $\tilde{x}_{ij} \sim \mathbf{P}_i \tilde{X}_j$. Ist die Anzahl der Kameras mindestens zwei, so ist es möglich nur auf Grund der Bildpunkte \tilde{x}_{ij} die Kameras \mathbf{P}_i zu berechnen. Mit bekannten Kameras können dann die Weltpunkte rekonstruiert werden. Dies bezeichnet man dann als Tiefen- oder Stereosehen. Allerdings ist es nun so, daß bei nur

zwei Aufnahmen sowohl die Kameras \mathbf{P}_i wie auch die rekonstruierten Punkte \tilde{X}_j nur bis auf eine projektive Transformation bestimmt werden können. Eine metrische oder euklidische Rekonstruktion ist nicht möglich.

Werden nun alle Aufnahmen von derselben Kamera gemacht, so ist \mathbf{K} konstant und es gilt: $\mathbf{P}_i = \mathbf{K}(\mathbf{R}_i | \mathbf{t}_i)$. Die internen Kameraparameter sind für alle Aufnahmen konstant, während die externen Kameraparameter die euklidische Bewegung zwischen den Kameras beschreiben. Glücklicherweise gibt es nun eine geometrische Größe im projektiven 3D Raum, welche unabhängig von den externen Parametern konstant bleibt und daher zur Kalibrierung der Kamera genutzt werden kann. Diese geometrische Größe ist der sogenannte *absolute Kegelschnitt* $\mathbf{\Omega}_\infty$, abgekürzt *AC* (engl. *Absolute Conic*). Ein Kegelschnitt ist eine quadratische Form für Punkte der projektiven Ebene. Sei zum Beispiel \mathbf{C} ein Kegelschnitt, so erfüllen die Punkte \tilde{x} folgende Gleichung: $\tilde{x}^T \mathbf{C} \tilde{x} = 0$. Der absolute Kegelschnitt ist ein Kegelschnitt, welcher in der unendlich fernen Ebene liegt. Das heißt, daß Punkte, welche auf dem Kegelschnitt liegen, die Form $\tilde{X} = (X, Y, Z, 0)^T = (\mathbf{X}^T, 0)^T$ aufweisen. Der absolute Kegelschnitt ist definiert als: $\mathbf{\Omega}_\infty = \mathbf{I}$ (\mathbf{I} ist die 3×3 Einheitsmatrix.). Die Kegelschnittgleichung lautet dann:

$$\mathbf{X}^T \mathbf{\Omega}_\infty \mathbf{X} = 0 \quad \text{mit} \quad \mathbf{\Omega}_\infty = \mathbf{I} \quad .$$

Punkte $\tilde{X} = (\mathbf{X}^T, 0)^T$ werden durch eine euklidische Transformation bestehend aus Rotation \mathbf{R}_i und Translation \mathbf{t}_i auf Punkte $\tilde{X}' = (\mathbf{X}'^T, 0)^T$ abgebildet, für die gilt: $\mathbf{X}' \sim \mathbf{R}_i \mathbf{X}$. Setzt man dies in obige Gleichung ein, so erhält man: $\mathbf{X}'^T \mathbf{\Omega}_\infty \mathbf{X}' = (\mathbf{R}_i^{-1} \mathbf{X}')^T \mathbf{I} (\mathbf{R}_i^{-1} \mathbf{X}') = \mathbf{X}^T \mathbf{R}_i^{-T} \mathbf{R}_i^{-1} \mathbf{X} = \mathbf{X}^T \mathbf{X} = \mathbf{X}^T \mathbf{\Omega}_\infty \mathbf{X} = 0$. Man sieht, der absolute Kegelschnitt ist invariant unter euklidischen Transformationen.

Weltpunkte der Form $\tilde{X} = (\mathbf{X}^T, 0)^T$ werden auf Bildpunkte \tilde{x}_i gemäß $\tilde{x}_i \sim \mathbf{K} \mathbf{R}_i \mathbf{X}$ abgebildet. Da \mathbf{K} und \mathbf{R}_i invertierbar sind gilt: $\mathbf{X} \sim (\mathbf{K} \mathbf{R}_i)^{-1} \tilde{x}_i$. Setzt man dies oben ein, ergibt sich: $((\mathbf{K} \mathbf{R}_i)^{-1} \tilde{x}_i)^T \mathbf{I} (\mathbf{K} \mathbf{R}_i)^{-1} \tilde{x}_i = \tilde{x}_i^T \mathbf{K}^{-T} \mathbf{R}_i \mathbf{R}_i^{-1} \mathbf{K}^{-1} \tilde{x}_i = \tilde{x}_i^T \mathbf{K}^{-T} \mathbf{K}^{-1} \tilde{x}_i = 0$. Die Matrix $\mathbf{\omega} = \mathbf{K}^{-T} \mathbf{K}^{-1}$ ist das *Bild des absoluten Kegelschnitts*, abgekürzt *IAC* (engl. *Image of the Absolute Conic*). Da $\mathbf{\Omega}_\infty$ invariant bezüglich der Kamerabewegung ist, gilt dies auch für dessen Bild $\mathbf{\omega}$:

$$\mathbf{\omega} = \mathbf{K}^{-T} \mathbf{K}^{-1} = \text{constant} \quad \text{für beliebige Kamerabewegungen} (\mathbf{R}_i, \mathbf{t}_i) \quad .$$

Jede Kamera außer der Ersten \mathbf{P}_i , $i = 2, 3, \dots$ liefert Bedingungen bezüglich des IACs; wie diese Bedingungen im Detail aussehen, soll hier nicht weiter erörtert werden. Wichtig ist nur die Tatsache, daß bereits drei Bilder mit einer identischen Kamera \mathbf{K} aufgenommen, ausreichend zur Bestimmung des IACs und somit zur Kalibrierung sind.

2.5 Numerisches Handwerkszeug

2.5.1 Fehlerquadrat Methode

An bisher zwei Stellen war eine lineares Problem im Sinne der kleinsten Fehlerquadratmethode zu lösen: Zur Berchnung der Homographie basierend auf 2D/2D Punktkorrespondenzen und zur Berechnung der projektiven Kamera durch 3D/2D Punktkorrespondenzen. Hier beschreiben wir jetzt die numerische Lösung für dieses Problem.

Sei folgendes homogene lineare Gleichungssystem gegeben:

$$\mathbf{Ax} = \mathbf{0} \quad ,$$

wobei \mathbf{A} eine $m \times n$ Matrix und \mathbf{x} ein n -dimensionaler Vektor ist. Besitzt die Matrix den Rang r , so hat der Lösungsraum die Dimension $n - r$. Daraus folgt, daß für den Fall, daß der Rang der Dimension des Vektors entspricht $n - r = 0$, die Lösung nur die triviale Lösung $\mathbf{x} = \mathbf{0}$ ist. Eine nicht triviale Lösung gibt es nur für $n > r$. Gibt es mehr Gleichungen m als Unbekannte n und hat die Matrix vollen Rang $r = n$, so ist das System überbestimmt. Dies ist der Fall, wenn mehr Meßpunkte oder Korrespondenzen als Unbekannte vorliegen und die Messungen nicht exakt, also rauschbehaftet sind.

Ist das System überbestimmt, so gibt es keine Lösung im eigentlichen Sinne. Stattdessen wird nach einer Lösung \mathbf{p} gesucht, welche den Fehlervektor $\mathbf{e} = \mathbf{Ap}$ minimiert. Der Fehler \mathbf{e} wird algebraischer Fehler genannt, weil er im Gegensatz zum sogenannten geometrischen Fehler (siehe auch Abschnitt 2.4.3) keiner geometrischen Größe entspricht. Der Abstand des Fehlers zum Ursprung ist seine euklidische Norm $e = \|\mathbf{e}\|$. Die kleinste Fehlerquadratmethode ist formal definiert als:

$$\mathbf{x} = \arg \min_{\mathbf{x}} \|\mathbf{Ax}\| \quad \text{mit} \quad \|\mathbf{x}\| = 1 \quad .$$

Die Normierung mit $\|\mathbf{x}\| = 1$ ist notwendig, um die triviale Lösung auszuschließen. Dieses Optimierungsproblem kann zum Beispiel mit Hilfe der Lagrangeschen Multiplikatormethode gelöst werden. Obiges Minimierungsproblem ist äquivalent zur Minimierung der Lagrangefunktion L :

$$L(\mathbf{x}, \lambda) = \|\mathbf{Ax}\|^2 + \lambda(1 - \|\mathbf{x}\|^2) \quad .$$

Eine notwendige Bedingung für ein Minimum ist:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{x}} &= \mathbf{0} \quad \Leftrightarrow \\ \frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} + \lambda(1 - \mathbf{x}^T \mathbf{x})) &= \mathbf{0} \quad \Leftrightarrow \\ 2(\mathbf{A}^T \mathbf{A}) \mathbf{x} - 2\lambda \mathbf{x} &= \mathbf{0} \quad \Leftrightarrow \\ (\mathbf{A}^T \mathbf{A}) \mathbf{x} &= \lambda \mathbf{x} \quad . \end{aligned}$$

Die letzte Zeile ist die Eigenwert- bzw. Eigenvektorgleichung. Alle Eigenvektoren \mathbf{x} erfüllen die notwendige Bedingung für ein Minimum. Der Wert von $\mathbf{x}_i^T (\mathbf{A}^T \mathbf{A}) \mathbf{x}_i$ für die Eigenvektoren \mathbf{x}_i ist $e_i^2 = \mathbf{x}_i^T (\mathbf{A}^T \mathbf{A}) \mathbf{x}_i = \mathbf{x}_i^T \lambda_i \mathbf{x}_i = \lambda_i$. Das Minimum wird also für den Eigenvektor angenommen, welcher zu dem kleinsten Eigenwert λ_i gehört. Die numerische Berechnung des kleinsten Fehlerquadrates ist folglich ganz einfach durch die Eigenwertberechnung der Matrix $\mathbf{A}^T \mathbf{A}$ gegeben: $\mathbf{A}^T \mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$, wobei die Spaltenvektoren von \mathbf{V} die Eigenvektoren sind und $\mathbf{\Lambda}$ eine Diagonalmatrix ist, welche die Eigenwerte enthält.

2.5.2 Normalisierung

Der Direct-Linear-Transformation Algorithmus besteht numerisch betrachtet aus der Eigenwertzerlegung einer Matrix, deren Elemente sich aus Koordinaten korrespondierender Punkte zusammensetzen. Nehmen wir als Beispiel einen typischen Bildpunkt mit den Koordinaten (100, 100). Da die Einträge in der Matrix \mathbf{A} in Gl. 2.11 quadratisch in den Bildkoordinaten sind, gibt es Einträge der typischen Größenordnung von 1, 10^2 und 10^4 ; in der Matrix $\mathbf{A}^T \mathbf{A}$ reicht die Größenordnung von 1 bis 10^8 . Man sieht, die Größenordnungen der einzelnen Einträge sind sehr verschieden. Dies resultiert in einer schlechten Konditionierung der Matrix $\mathbf{A}^T \mathbf{A}$. Die Kondition ist das Verhältnis von größtem zum kleinsten Eigenwert (Da bei unverrauschten Daten der kleinste Eigenwert Null ist, wird hier nicht der kleinste, sondern der zweitkleinste Eigenwert gewählt.). Bei den oben gewählten typischen Koordinaten kann die Kondition in der Größenordnung von 10^8 liegen. Eine schlechte Konditionierung führt zu ungenauen Ergebnissen in der Eigenwertberechnung. Im Extremfall wird die Rechengenauigkeit der Maschine erreicht, und eine Berechnung ist gar unmöglich. Daher ist es notwendig, die Daten geeignet zu normalisieren, sodaß die Konditionierung verbessert wird. Die Daten werden derart normalisiert, daß der Mittelwert Null ist, und die Standardabweichung $\sqrt{2}$ beträgt (Bei dreidimensionalen Daten ist $\sqrt{3}$ zu wählen.). Die Einträge der Matrix \mathbf{A} werden dann alle die gleiche Größenordnung aufweisen und die Kondition ist oft kleiner als

10. In dem praktischen Teil wird gezeigt werden, daß der DLT Algorithmus einschließlich Normalisierung deutlich bessere Ergebnisse bei verrauschten Daten liefert als der DLT Algorithmus ohne Normalisierung. In Formeln schreibt sich die Normalisierung wie folgt:

$$\mathbf{x}_{\text{norm}} = \frac{\sqrt{2}}{\sigma}(\mathbf{x} - \boldsymbol{\mu}) \quad \text{mit} \quad \boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad , \quad \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \boldsymbol{\mu}\|^2} \quad .$$

2.6 Literatur

Das in diesem Bericht beschriebene Kalibrierungsverfahren geht zurück auf Zhang und wurde 1998 bzw. 2000 veröffentlicht [56, 57]. Für Windows lauffähiger Code ist dazu im WWW zu finden [55]. Diese Methode ist von höherem praktischen Wert als das Verfahren nach Tsai [51], welches in der Vergangenheit häufig eingesetzt wurde. Der Direct-Linear-Transformation Algorithmus geht zurück auf Abdel-Aziz und Karara [1]. Ein ausgesprochen gut geschriebenes und verständliches Buch über das Computersehen ist das Buch von Hartley und Zisserman [21]. Dort werden angefangen mit den geometrische Grundlagen, Kameramodelle und Kalibrierung, einschließlich Selbstkalibrierung, bis hin zum Stereosehen alle wichtigen Themen in einer abgerundeten Form präsentiert.

2.7 Aufgaben

Die mit einem Stern * gekennzeichneten Aufgaben sind optional.

2.7.1 Theoretischer Teil

Projektive Geometrie

Aufgabe 16: Seien $\mathbf{m}_{11} = (0,0)^T$ und $\mathbf{m}_{12} = (1,1)^T$ zwei Punkte auf der Geraden \mathbf{l}_1 . Berechnen Sie die homogenen Koordinaten von \mathbf{l}_1 . Wie lautet der Normalen- und der Richtungsvektor von \mathbf{l}_1 ? Sei eine weitere Gerade \mathbf{l}_2 durch die Punkte $\mathbf{m}_{21} = (1,0)^T$ und $\mathbf{m}_{22} = (2,1)^T$ gegeben. Wie steht die Gerade \mathbf{l}_2 zur Geraden \mathbf{l}_1 ? Berechnen Sie den Schnittpunkt der beiden Geraden. Was ist die geometrische Bedeutung der Lösung?

Aufgabe 17: Wieviele Freiheitsgrade haben die projektive, affine, ähnliche und euklidische Transformation und wieviele Punktkorrespondenzen sind zu ihrer Berechnung notwendig?

Aufgabe 18: Zeigen Sie, daß die affine Transformation Punkte im Unendlichen bzw. Endlichen auf Punkte im Unendlichen respektive Endlichen abbildet.

Kameramodell

Aufgabe 19: Wie äußert sich die Eigenschaft von quadratischen Pixeln in der Kalibriermatrix?

Aufgabe 20: Welche Bedeutung haben die beiden Elemente x_0, y_0 der Kalibriermatrix?

Aufgabe 21: Was modelliert der Scherungsparameter s ?

Aufgabe 22: Was versteht man unter radialer Linsenverzerrung?

Kalibrierung

Aufgabe 23: Wie muß ein Kalibriermuster aufgebaut sein, damit eine Kalibrierung mit nur einer Aufnahme möglich ist.

Aufgabe 24: Ein ebenes Kalibriermuster wird in Richtung der Normalen verschoben. Muß die Länge der Verschiebung bekannt sein?

Aufgabe 25: Wieso sind 3D Kalibrierverfahren unhandlich?

Aufgabe 26: Wieviele Aufnahmen einer Kalibrierebene sind für eine komplette 2D Kalibrierung notwendig?

Aufgabe 27: Welche geometrische Größe wird bei der nicht linearen Optimierung minimiert?

Aufgabe 28: Beschreiben Sie das 2D Kalibrierverfahren.

2.7.2 Praktischer Teil

Verweise auf zu benutzende MATLAB Funktion sind in `typewriter` formatiert.

Eine Sequenz bestehend aus fünf Ansichten eines Kalibrieremusters wird in `calibrationSequence.mat` zur Verfügung gestellt. Auch die dazugehörigen Punktkorrespondenzen sind dort zu finden.

Labelling

Aufgabe 1: Anhand eines Bildes aus der Sequenz soll der Vorgang des sogenannten Labellings behandelt werden. Ein Satz von gelabelten Punktkorrespondenzen ist der erste Schritt bei der Kamerakalibrierung.

An erster Stelle steht die Detektion der Kreise. Dazu wird zuerst die gewünschte Region des Kalibrieremusters ausgeschnitten (`R = roipoly(Image)`). Dann werden die Kantepunkte der Kreise gefunden (`E = edge(Image)&R`). Damit die Kreise auch wirklich zusammenhängend sind, werden die Kanten dilatiert (`E = imdilate(E,ones(3,3))`), und dann die Kreise gefüllt (`E = bwfill(E,'holes')`). Manchmal kommt es vor, daß beim Ausschneiden Kreise angeschnitten werden. Diese Kreise sollten erkannt und aussortiert werden, da sie keine korrekte Markierung mehr darstellen (`[m,n] = find(E&~R); E(bwselect(E,m,n)) = 0`). Schließlich werden alle zusammenhängenden Gebiete durchnumeriert (`L = bwlabel(E)`), wobei anzumerken ist, daß diese Numerierung quasi willkürlich ist, d.h. die Numerierung noch keinen Rückschluß auf die Position der Markierung zuläßt.

Aufgabe 2: Der eigentliche Labelling Prozeß besteht nun daraus, den Markierungspunkten ihre Weltkoordinaten zuzuordnen. Dieser Vorgang soll automatisch geschehen und ist in der Funktion `labelling` implementiert. Dort werden zuerst die Schwerpunkte der Kreise berechnet. Die linke, obere Markierung wird als Koordinatenursprung gewählt. Die beiden nächstgelegenen (Schwer-) Punkte bestimmen willkürlich die x - und y -Achse des ebenen Koordinatensystems. Ausgehend davon werden sukzessiv alle Punkte in ihren x - und y -Koordinaten durchnumeriert. Der Abstand der Punkte ist dann auf Eins normiert. Soll die tatsächliche metrische Größe als Abstand in die Kalibrierung mit eingehen, so erreicht man dies durch eine einfache Multiplikation der Koordinaten mit dieser Größe.

Berechnung der Homographien

Aufgabe 3: Programmieren Sie die fehlenden Teile in `computeHomography`; die Vorlage dazu ist in `computeHomograph.template.m` zu finden.

Normalisierung

Anhand der Berechnung einer Homographie soll gezeigt werden, wie wichtig die Normalisierung der Daten ist. In der Datei `dataNormalizationData.mat` befindet sich ein Testdatensatz mit 9 Korrespondenzen. Die unverrauschten Punktkoordinaten sind für beide Bilder X_0 . Dazu wurde Rauschen, modelliert durch eine Gaußsche Normalverteilung mit einer Standardabweichung von 5, addiert, so daß sich die Korrespondenzen X_1, X_2 ergeben. Berechnen sie die Homographie mit und ohne Normalisierung (`help computeHomography`). Die beiden Ergebnisse sollen mit der tatsächlichen Homographie, nämlich der homogenen 3×3 Einheitsmatrix, verglichen werden.

Aufgabe 4*: Programmieren Sie die fehlenden Teile in `dataNormalization`; die Vorlage dazu ist in `dataNormalization.template.m` zu finden.

Aufgabe 5*: Berechnen Sie den geometrischen Fehler, welcher der euklidische Abstand zwischen den Punkten und ihren projizierten Korrespondenzen ist: $d_i = \|\mathbf{x}_i - \mathbf{H}^1 \mathbf{x}_i\|$. Der Gesamtfehler ist dann $\sqrt{\frac{1}{n} \sum_{i=1}^n d_i}$, welcher auch als Root-Mean-Square (RMS) Fehler bezeichnet wird.

Kameramodell

Aufgabe 6: Die Funktion `camera` bildet Weltpunkte einer Ebene auf Bildpunkte ab. Programmieren sie die Lücken in `camera.template.m`.

Kalibrierung

Aufgabe 7: Das Skript `calibration` führt die komplette Kamerakalibrierung durch. Schauen Sie sich das Skript an und vollziehen die einzelnen Schritte nach.

Praktikumsversuch

Farbe

S. Siggelkow und H. Burkhardt

3 Farbe

3.1 Einleitung

Farbe ist ein wichtiger Bestandteil in der Bildverarbeitung und Bildanalyse. Es ist jedoch keineswegs trivial, das menschliche Farbempfinden auf dem Computer nachzubilden. Am Anfang müssen daher Untersuchungen der menschlichen Farbwahrnehmung stehen. Zum einen darf bei der Abbildung der realen Szene auf den Computer keine für den Menschen relevante Information verlorengehen, zum anderen braucht aber auch diejenige Information, welche der Mensch nicht wahrnimmt, nicht abgebildet werden. In Kapitel 3.2 wird daher der Ablauf der Farbwahrnehmung im menschlichen Auge betrachtet.

In Kapitel 3.3 werden dann unterschiedliche Möglichkeiten der Darstellung von Farbe auf dem Computer vorgestellt. Diesen liegen unterschiedliche Motive zugrunde: Manche Systeme basieren auf der Farberzeugung im Monitor und sind somit auf diese bestens abgestimmt, wieder andere sind für eine Übertragung von Bildinformationen bestimmt, zielen also auf eine einfache verlustbehaftete, aber visuell kaum wahrnehmbare Komprimierung ab. Für das Messen von Farbe wiederum sind Systeme gefragt, in denen das menschliche Farbunterscheidungsvermögen durch einfache Metriken dargestellt werden kann. Und schließlich sind für den interaktiven Gebrauch Farbsysteme gefragt, die Farbe in einer dem Menschen geläufigen Form beschreiben.

Durch Beleuchtungsänderungen können sich die Farben ein und derselben Szene entscheidend verändern. Der Mensch ist aber trotzdem in der Lage, eine Szene bei Tageslicht und bei Sonnenuntergang als unverändert zu erkennen. Für den Computer ist dies keine leichte Aufgabe. Es gibt aber Möglichkeiten den Einfluß der Beleuchtung unter einigen idealisierten Bedingungen herauszurechnen. Ein solches Verfahren wird in Kapitel 3.4 behandelt werden.

Im letzten Kapitel sind schließlich die zu bearbeitenden Aufgaben angegeben.

3.2 Menschliche Farbwahrnehmung

Die menschliche Farbwahrnehmung ist von fundamentaler Bedeutung für die Betrachtung von Farbe im Rahmen der digitalen Bildverarbeitung. Denn nur, was der Mensch sieht, braucht auf dem Computer nachgebildet zu werden. Daher soll zunächst geklärt werden: *Was ist Farbe* überhaupt? Wie kommt das Farbsehen zustande?

Physikalisch gesehen ist Farbe nichts anderes als Licht mit unterschiedlichen Wellenlängen. Der sichtbare Bereich beginnt bei Blauviolett (ca. 410 nm) und setzt sich fort über Blau (470nm), Grün (520nm), Gelb (590nm), Orange (620nm) bis zu Rot (700nm).

Die Farbe eines Gegenstandes wird bestimmt durch die Zusammensetzung des Lichts, das der Gegenstand reflektiert bzw. hindurchläßt. Farben, die absorbiert werden, verschwinden im Spektrum des reflektierten Lichts und verändern somit seine Farbe. Wie wird diese nun vom menschlichen Auge aufgenommen?

Die Netzhaut des Auges (Abbildung 3.1) enthält lichtempfindliche Sensoren. Man unterscheidet zwischen zwei Arten:

Stäbchen für reines Helligkeitsempfinden bei wenig Lichtintensität (ca. 120 Mio.) und

Zapfen für das Farbsehen bei höherer Lichtintensität (ca. 5..10 Mio.).

Dabei sind die Stäbchen ungefähr um den Faktor 1000 empfindlicher als die Zapfen und somit für das Nachtsehen zuständig. Da sie aber nur zum Helligkeitsempfinden beitragen, erscheinen Objekte in der Dunkelheit farblos. Während die Zapfen vor allem in der Netzhautgrube (gelber Fleck) liegen, auf die das einfallende Licht fokussiert wird, befinden sich die Stäbchen im Umfeld. Daher werden Objekte nachts eher unscharf wahrgenommen.

Das farbige Sehen beruht nach Young und Helmholtz darauf, daß 3 Arten von Zapfen vorhanden sind, und zwar rot-, grün-, und blauempfindliche. Diese Zäpfchen tragen zur Farbwahrnehmung blau-, grün- und gelbsensible Fotopigmente. Die Benennung der roten Zäpfchen ist etwas verwirrend und wohl historisch dadurch zu begründen, daß sie die empfindlichsten für das rechte Ende des sichtbaren Spektrums, also den roten Bereich sind. Tatsächlich liegt ihre maximale Empfindlichkeit aber im gelben Bereich. Bedingt durch diese Fotopigmente, die ungleiche anzahlmäßige Verteilung der einzelnen

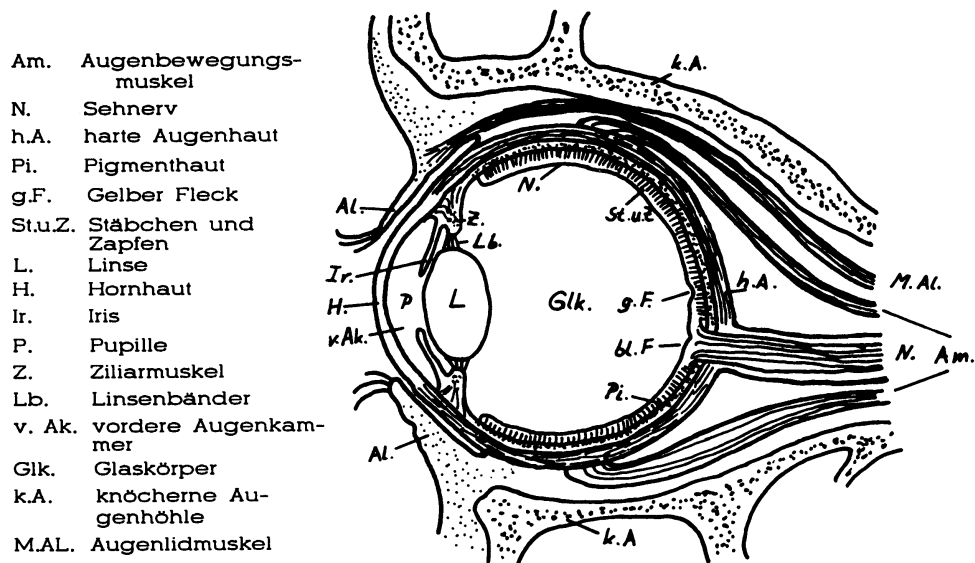


Abbildung 3.1: Vertikaler Schnitt durch das menschliche Auge [36].

Fotopigmente (64% gelbe, 32% grüne und der Rest blaue) und eine ungleiche räumliche Verteilung auf der Netzhaut (im Zentrum grün, dann gelb, dann blau), ergeben sich für die einzelnen Zäpfchensorten sehr unterschiedliche Farbempfindlichkeiten wie in Abbildung 3.2 dargestellt (die Farbempfindlichkeit der blauempfindlichen Zapfen ist sehr gering und daher 20-fach überhöht dargestellt).

Signaltechnisch betrachtet, findet im Auge also eine Abbildung des kompletten Spektralraums $C(\lambda)$ auf 3 Farbsignale $\alpha_i, i = 1, 2, 3$ (Tristimulus) statt, das Auge entspricht einem Filter mit folgender Funktion:

$$\alpha_i(C) = \int_{\lambda_{\min}}^{\lambda_{\max}} S_i(\lambda) C(\lambda) d\lambda, \quad i = 1, 2, 3, \quad (3.1)$$

wobei $S_i(\lambda)$ die in Abbildung 3.2 dargestellten Sensorfunktionen darstellen.

Dies hat wichtige Folgen für die Behandlung von Farbe auf dem Computer, denn es ist in Anwendungen, in denen der nur visuelle Eindruck einer Szene nachgebildet werden soll (Fernsehen, Computergrafik etc.), nicht nötig, das gesamte Farbspektrum darzustellen, sondern es reichen 3 geeignet gewählte Farbreize aus, um einen Bildpunkt zu beschreiben. Dies ist in der Dreifarben Theorie (s. Kapitel 3.3.1) festgehalten.

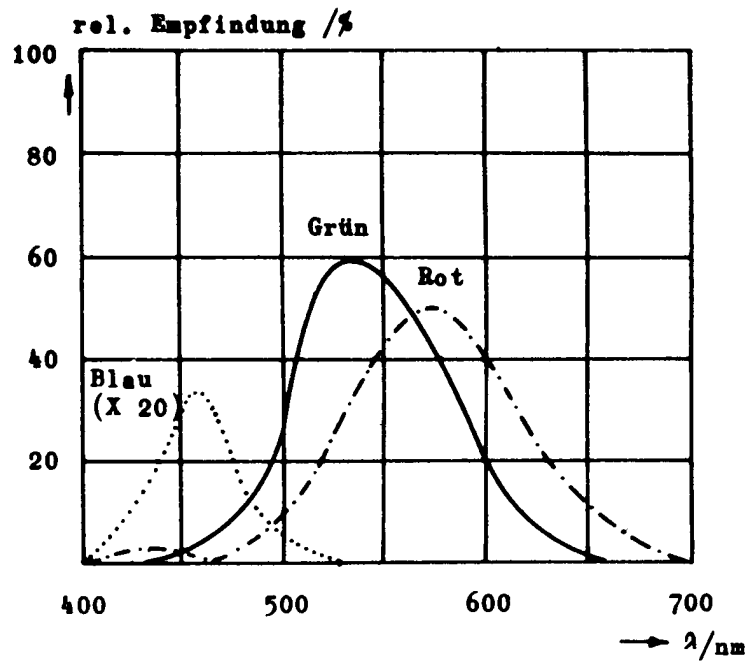


Abbildung 3.2: Relative Farbempfindlichkeit der Zapfen [36].

3.3 Farbdarstellung

3.3.1 Dreifarbentheorie

Wir hatten im vorigen Kapitel gesehen, daß das menschliche Auge das Farbspektrum in einen dreidimensionalen Farbraum abbildet. Somit sollte die wahrgenommene Farbinformation auch lediglich mit Hilfe von drei Basisvektoren darstellbar sein.

Die Dreifarbentheorie von Thomas Young (1802) besagt genau dies:

Jede sichtbare Farbe läßt sich aus drei Primärfarben durch additive Farbmischung herstellen

Mit den Primärfarben $\mathcal{R}, \mathcal{G}, \mathcal{B}$ ergibt sich die additive Farbmischung also zu

$$X = R\mathcal{R} + G\mathcal{G} + B\mathcal{B}. \quad (3.2)$$

Die Primärfarben müssen dabei natürlich linear unabhängig sein, d.h. es darf sich keine durch Mischen der anderen beiden erzeugen lassen können.

Als ein Satz möglicher Primärfarben wurden z.B. 1931 von der CIE (Commission Internationale de l'Eclairage) die folgenden monochromen Strahler festgelegt (d.h. Strahler, deren Spektrum aus einer einzigen Wellenlänge besteht): Rot mit 700nm, Grün mit 546.1nm und Blau mit 435.8nm. Mit diesen ergeben sich die in Abbildung 3.3 dargestellten Kurven für R, G, B zur Darstellung monochromatischen Lichts. Um z.B. Licht zu erzeugen, das einem monochromen Strahler mit Wellenlänge 580nm entspricht, müßten also die monochromen Strahler Rot und Grün ungefähr zu gleichen Teilen gemischt werden.

Es wird unmittelbar deutlich, daß der Satz zur Dreifarbentheorie nur eingeschränkt gilt, da in der Praxis keine Strahler mit negativer Intensität existieren. Stattdessen können nur die Farben dargestellt werden, die sich durch Linearkombination der Primärfarben mit positiven Vorfaktoren R, G, B erzeugen lassen.

Zur Verdeutlichung dieses Sachverhaltes soll der 3D-Farbraum genauer betrachtet werden: Eine Komponente darin ist die Intensität. Wird ein Farbvektor mit einem Skalar multipliziert, ändert sich zwar seine Intensität, jedoch nicht seine Farbe. Daher sollen für diese Betrachtung alle Farben über die Intensität normiert werden. Diese Operation

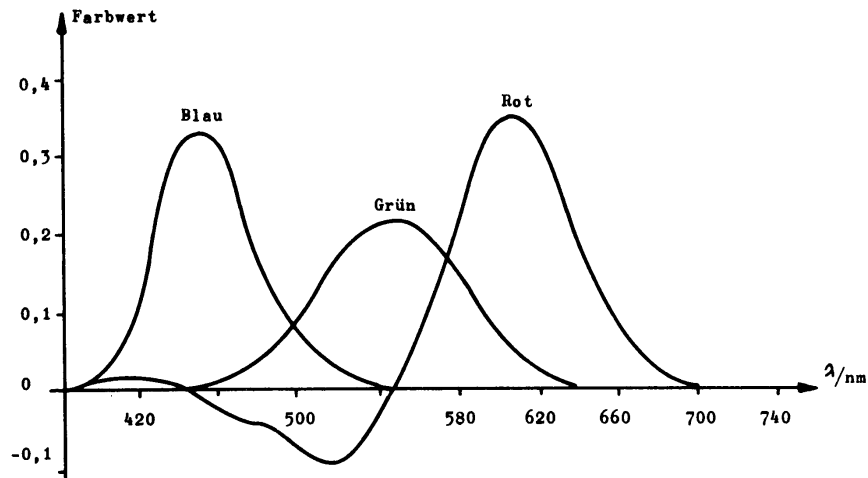


Abbildung 3.3: CIE-Farbmischkurven mit den Primärfarben Rot (700nm), Grün (546.1nm) und Blau (435.8nm) [36].

reduziert den 3D-Farbraum auf eine 2D-Farbebene:

$$r = \frac{R}{R+G+B}, \quad (3.3)$$

$$g = \frac{G}{R+G+B}, \quad (3.4)$$

$$b = \frac{B}{R+G+B}. \quad (3.5)$$

Es ist ausreichend, nur die Komponenten r und g zu betrachten, da offensichtlich $b = 1 - r - g$. Die Grenze der sichtbaren Farben wird durch alle sichtbaren monochromatischen Farben von blau bis rot bestimmt und ist als hufeisenförmige Kurve (Spektralkurve) gegeben. Da nur die Farben innerhalb dieser Form durch Mischung monochromatischer Farben erzeugt werden können, ergibt die von der Spektralkurve und der Verbindung ihrer beiden Enden (Purpurlinie) eingeschlossene Fläche den Raum aller möglichen Farben. Es ist sofort offensichtlich, daß durch die von der CIE vorgegebenen Primärfarben nicht der gesamte sichtbare Farbbereich, sondern nur ein Teilbereich abgedeckt werden kann.

Der Vollständigkeit halber soll auch noch eine zweite Art der Farbmischung erwähnt werden. Die bisher dargestellte Form der additiven Farbmischung trifft für sogenannte

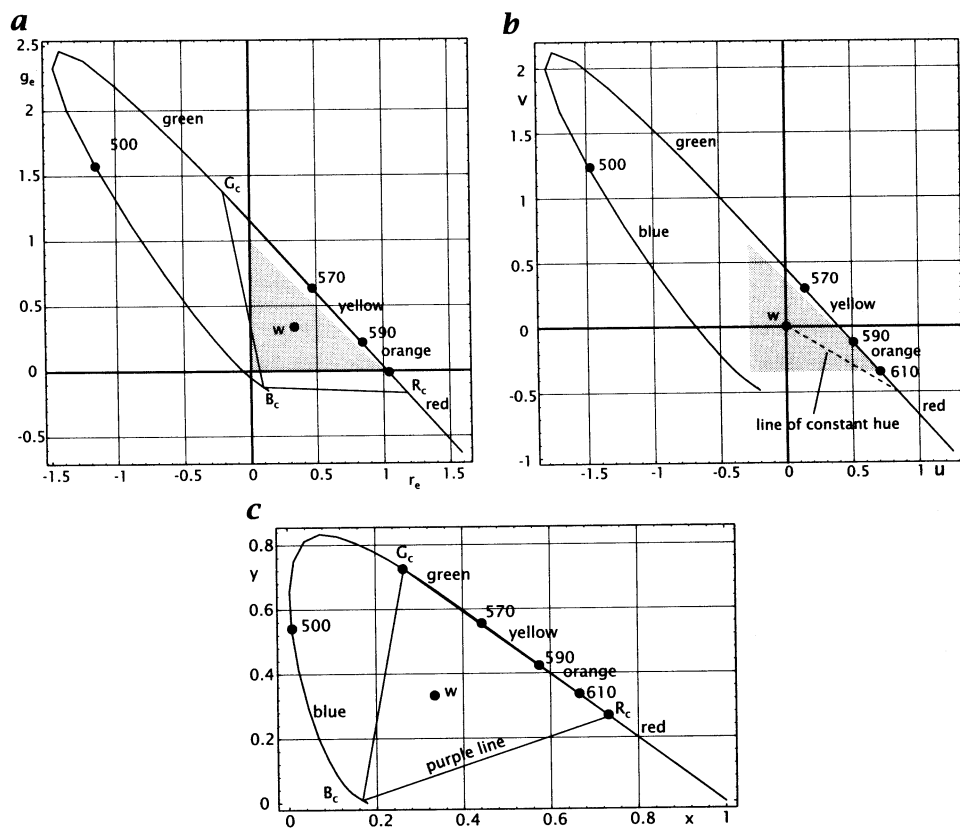


Abbildung 3.4: Farbebene, dargestellt in unterschiedlichen Farbsystemen [27].

Lichtfarben zu, d.h. Farben, die dadurch entstehen, daß eine Lichtquelle selbst Licht abstrahlt. Z.B. entstehen Bilder auf dem Computermonitor oder Fernseher durch Lichtfarben. Die Anzeige erfolgt hier durch laufendes Aussenden von Elektronen, die auf die Bildschirmoberfläche treffen und dort ein farbiges Aufleuchten des Phosphors bewirken.

In der natürlichen Umwelt begegnet man dagegen überwiegend der subtraktiven Farbmischung. Der Farbeindruck, den der Betrachter von einem Gegenstand erhält, entsteht dadurch, daß ein Teil des Lichts, das auf die Oberfläche des Gegenstandes trifft, von dieser reflektiert wird, während der Rest absorbiert wird. Daher heißen diese Farben auch Körperfarben. Körperfarben werden wegen der Absorption (Subtraktion) auch als Subtraktionsfarben bezeichnet.

Alle Farbnuancen der Körperfarben lassen sich auf drei Primärfarben reduzieren: Gelb, Magentarot und Zyanblau. In der Computertechnik ist diese Art der Farbmischung vor allem bei Farbdruckern von Bedeutung.

Abschließend sollen noch einmal die wichtigsten Eigenschaften zusammengefaßt werden:

1. Jede Farbe kann - zumindest theoretisch - aus 3 farbigen Lichtquellen gemischt werden (in der Praxis gibt es Beschränkungen durch das Fehlen negativer Lichtintensitäten).
2. Das menschliche Auge kann nicht die Spektralkomponenten einer Farbe auflösen.
3. Farbaddition: Sei \sim eine Äquivalenzrelation: $C_1 \sim C_2 \iff C_1$ und C_2 werden farblich gleich wahrgenommen. Dann gilt:

$$C_1 \sim C_2, C'_1 \sim C'_2 \implies \alpha_1 C_1 + \alpha_2 C_2 \sim \alpha_1 C'_1 + \alpha_2 C'_2. \quad (3.6)$$

4. Farbsubtraktion:

$$C_1 + C_2 \sim C'_1 + C'_2, C_1 \sim C'_1 \implies C_2 \sim C'_2. \quad (3.7)$$

5. Transitivität:

$$C_1 \sim C_2, C_2 \sim C_3 \implies C_1 \sim C_3. \quad (3.8)$$

3.3.2 Farbsysteme

Im Bereich der Bildverarbeitung und Übertragung existieren zahlreiche unterschiedliche Farbsysteme. Da fragt man sich, ob es denn kein optimales Farbsystem gibt und alle

Bilddaten in diesem dargestellt werden sollten. Die Tauglichkeit von Farbsystemen läßt sich allerdings unter vielen verschiedenen Aspekten betrachten, so daß man keines als universell optimal bezeichnen würde.

Ein Aspekt kann die Anschaulichkeit der Farbdarstellung sein. Diese ist wie wir am Beispiel einer konkreten Aufgabe noch sehen werden z.B. im HSV-Farbraum viel besser gegeben als im RGB-Farbraum.

In anderen Applikationen ist die Gleichabständigkeit von größerer Bedeutung, d.h. der Abstand gerade noch zu unterscheidender Farben sollte im Farbraum eine ortsunabhängige Konstante sein.

Ein weiterer Gesichtspunkt, der sowohl bei der bandbreitensparenden Übertragung im Fernsehen als auch bei der Komprimierbarkeit digitaler Bilddaten zum Tragen kommt, ist die Unterscheidung von Farb- und Helligkeitsinformation. Die Physiologie unserer Augen erlaubt es, im Farbkanal mit der Hälfte oder gar nur einem Viertel der Informationsmenge auszukommen, die im Helligkeitskanal enthalten ist. Deshalb sind sowohl beim Farbfernsehen als auch in der digitalen Bildverarbeitung Farbsysteme gebräuchlich, die aus einem Helligkeitskanal L oder Y für die Luminanz- oder Schwarzweiß-Information und zwei sog. Farbdifferenzkanälen für die Farbinformation bestehen. Es ist aber sofort klar, daß der Verlust von Farbinformation z.B. im Bereich der Qualitätskontrolle zu Problemen führen kann. Die unterschiedlichen Farbsysteme haben also alle ihre Berechtigung.

Im folgenden sollen die gebräuchlichsten und wichtigsten Farbsysteme dargestellt und gemäß ihren Eigenschaften klassifiziert werden.

Das RGB-Farbsystem

Es existieren verwirrenderweise unterschiedliche RGB-Farbsysteme. Die von der CIE vorgeschlagenen Primärfarben haben - zumindest derzeit - technisch wenig Bedeutung, da in Monitoren die Farberzeugung üblicherweise dadurch geschieht, daß Phosphore durch Elektronenbeschuß zum Leuchten angeregt werden. Diese Phosphore geben aber kein monochromatisches Licht ab, wie es bei den CIE-Primärfarben angenommen wird, sondern ein ganzes Lichtspektrum. Somit liegen diese Primärfarben im Inneren der Spektralkurve statt auf ihr. Außerdem wurden bei den Fernsehsystemen PAL/SECAM einerseits und bei NTSC andererseits unterschiedliche Phosphore verwendet. Also gibt es auch hier noch geringe Unterschiede. Meist ist allerdings die NTSC-Empfängerbildröhrensystem-Spezifikation gemeint, wenn vom RGB-System die

Rede ist.

Das Normalfarbsystem XYZ

Um negative Farbkoordinaten wie in Abbildung 3.3 zu vermeiden, wurde ein neues Farbkoordinatensystem mit virtuellen Primärfarben eingeführt, d.h. mit Primärfarben, die keiner physikalischen Farbe entsprechen. Dieses XYZ-System ist so konstruiert, daß es die Spektralkurve umschließt. Es wird durch die folgende lineare Koordinatentransformation vom RGB-NTSC-Empfänger-System beschrieben

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.201 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.117 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \quad (3.9)$$

Die inverse Transformation ergibt sich direkt durch invertieren der Transformationsmatrix.

Die Fernsehfarbsysteme YIQ und YUV

In der Fernsehtechnik wird Nutzen aus der Unvollkommenheit des menschlichen visuellen Wahrnehmungssystems gezogen: Es reagiert sehr viel empfindlicher auf Fehler in der Helligkeit als auf Fehler in der Farbe. Aus diesem Grund ist es vorteilhaft, den Farbraum durch 3 Basisvektoren darzustellen, von denen der eine im Wesentlichen die Helligkeit beschreibt (Luminanz Y) und der daher ziemlich unverfälscht bleiben sollte, und zwei weitere Basisvektoren, die weniger genau übertragen werden müssen. Bei der analogen Fernsehübertragung konnte daher die Bandbreite der beiden Farbsignale gegenüber dem Luminanzsignal reduziert werden. Dies ist in Abbildung 3.5 beispielhaft für das NTSC-System dargestellt. In der Praxis bedeutet dies, daß die Farbart mit wesentlich geringerer örtlicher Auflösung übertragen und damit auch dargestellt wird als die Helligkeit. Bei der ein Europa stärker verbreiteten Fernsehnorm PAL sind die Komponenten U,V um 33° gegenüber I,Q gedreht. Das liegt darin begründet, daß beim YIQ-System eine weitergehende Anpassung an das menschliche visuelle Wahrnehmungsvermögen vorgenommen wurde, die die Richtung besonders hoher Empfindlichkeit für Farbdifferenzen berücksichtigt (s. Abbildung 3.6). Auch diese Koordinatensysteme ergeben sich durch eine lineare Transformation aus dem RGB-NTSC-Empfänger-System. Für das

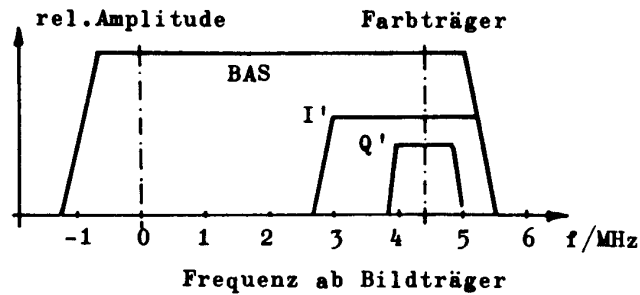


Abbildung 3.5: Übertragungsbänder beim NTSC-System [36].

YIQ-System ergibt sich

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (3.10)$$

und für das YUV-System:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \quad (3.11)$$

Aber auch digitale Bild- und Videoformate nutzen die Unempfindlichkeit des menschlichen visuellen Wahrnehmungsvermögens gegenüber Farbfehlern aus. So liest man im Zusammenhang mit dem Digitalfernsehen bzw. Digitalkameras oft von der Bezeichnung 4:2:2 oder 4:1:1. Diese bedeuten, daß die Farbkomponenten in verringerter örtlicher Auflösung dargestellt werden, so bedeutet z.B. 4:1:1, daß ein 2x2-Bildausschnitt mit 4 Helligkeitswerten, aber nur 2 Farbwerten beschrieben wird (statt 2x4 Farbwerten). Tatsächlich wird auch noch bei der Quantisierung der Signalwerte häufig eine gröbere Unterteilung bei den Farbwerten als bei dem Helligkeitswert vorgenommen.

Die farbmtrischen Räume UVW, L*a*b*

Wie man Abbildung 3.6 bereits gesehen hatte, ist das Farbunterscheidungsvermögen in den bisher betrachteten Räumen äußerst ungleichmäßig verteilt. In vielen Anwendungen ist es aber gewünscht, direkt aus der relativen Lage von zwei Farben abzuleiten,

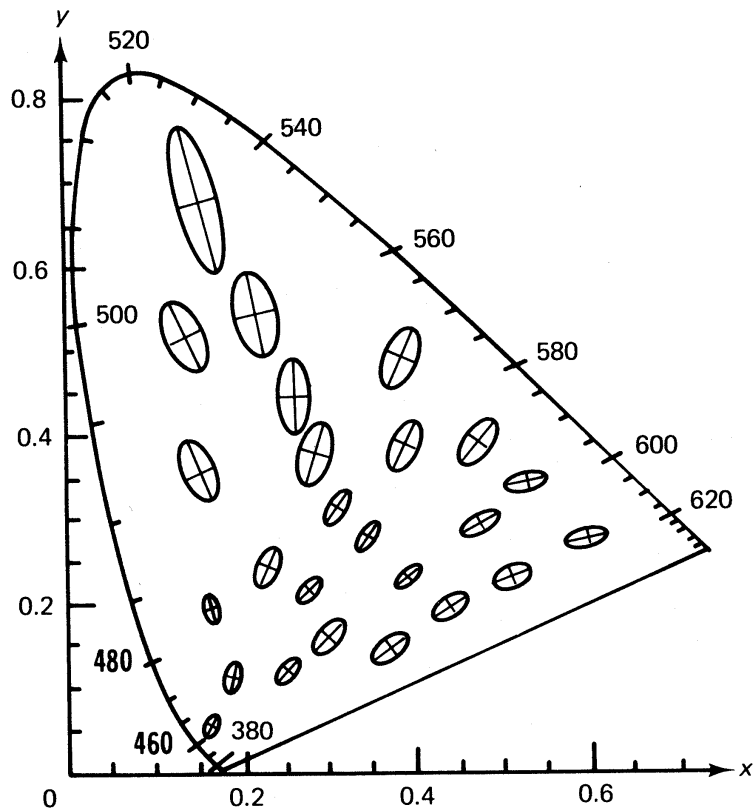


Abbildung 3.6: Relatives Farbunterscheidungsvermögen nach Mc Adams (Mc Adams-Ellipsen) [28]. Die Ellipsen deuten die Farbbereiche an, die vom Auge gerade noch unterschieden werden können, d.h. ein Farbpunkt am Rande einer Ellipse kann gerade eben von einem Farbpunkt im Zentrum der Ellipse unterschieden werden (Ellipsen zehnfach vergrößert dargestellt).

wie unterschiedlich der Mensch sie auflöst. Zu diesem Zweck wurden eine Reihe von farbmtrischen Räumen geschaffen. In diesen entspricht der visuelle Unterschied näherungsweise der euklidischen Distanz der Farben. Im Wesentlichen wurde also versucht, die Mc Adams-Ellipsen (Achsenverhältnis bis 20:1) durch eine Koordinatentransformation näherungsweise auf Kreise (Achsenverhältnis bis 2:1) abzubilden.

Die Transformation gegenüber dem XYZ-System ist gegeben durch:

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 0.405 & 0.116 & 0.133 \\ 0.299 & 0.587 & 0.114 \\ 0.145 & 0.827 & 0.627 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (3.12)$$

Neben dem UVW-Raum existiert auch noch ein modifizierter Raum $U^*V^*W^*$ mit verschobenem Ursprung. In diesem System kann der Farbunterschied zweier ungesättigter (also pastellartiger) Farben sehr gut durch die Länge der Verbindungslinie angenähert werden.

$$d = \sqrt{(\Delta U^*)^2 + (\Delta V^*)^2 + (\Delta W^*)^2} \quad (3.13)$$

Das $L^*a^*b^*$ -System ist durch folgende nichtlineare Transformation des XYZ-Systems gegeben:

$$L^* = 25 \left(\frac{100Y}{Y_0} \right)^{\frac{1}{3}} - 16, \quad 1 \leq 100Y \leq 100 \quad (3.14)$$

$$a^* = 500 \left[\left(\frac{X}{X_0} \right)^{\frac{1}{3}} - \left(\frac{Y}{Y_0} \right)^{\frac{1}{3}} \right] \quad (3.15)$$

$$b^* = 200 \left[\left(\frac{Y}{Y_0} \right)^{\frac{1}{3}} - \left(\frac{Z}{Z_0} \right)^{\frac{1}{3}} \right]. \quad (3.16)$$

Hierin geben X_0, Y_0, Z_0 die Tristimulus-Werte des Referenz-Weißpunktes an. Anschaulich gibt L^* die Helligkeit an, a^* unterscheidet zwischen rot und grün, b^* zwischen gelb und blau.

Auch mit diesem System lassen sich Farbunterschiede einfach als Abstand im Farbraum berechnen.

$$d = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad (3.17)$$

Grundlage dieses Systems bildet das Munsell-System zur Farbklassifikation. Nach diesem System findet die Klassifikation durch Vergleich mit 1200 dem visuellen Empfinden nach geordneten Farbmustern statt. Um diese Methode auch rechnerisch zugänglich zu machen, wurde das $L^*a^*b^*$ -System entwickelt.

Die Farbräume HSV und IHS

Die bisher betrachteten Systeme sind kaum für eine anschauliche Beschreibung von Farbe geeignet. Es würde z.B. kaum jemand auf die Idee kommen, die Farbe einer Sonnenblume als viel Rot und Grün im RGB-System zu beschreiben. Stattdessen würde man sie als helles, sattes Gelb beschreiben. Für Benutzer-Interfaces sollte daher eine andere Beschreibung gewählt werden. Dies ist mit Hilfe der Systeme HSV oder IHS möglich.

In diesen wird eine Farbe durch die 3 Komponenten Helligkeit, Farbton und Sättigung angegeben.

In Abbildung 3.7 ist der HSV-Farbraum dargestellt (Hue – Farbton, Saturation – Sättigung, Value – Helligkeit)¹. Er besitzt die Form eines Zylinders. Von unten nach oben steigt die Helligkeit der Farbe, von innen nach außen die Sättigung. Der Farbton wird näherungsweise durch den auf $[0 \dots 1]$ normierten Winkel repräsentiert (rot $\hat{=}$ 0, grün $\hat{=}$ $1/3$, blau $\hat{=}$ $2/3$). Die Umrechnung aus dem RGB-Farbsystem geschieht nach folgenden Formeln [48]:

$$V = \max(R, G, B) \quad (3.18)$$

$$S = \frac{V - \min(R, G, B)}{V} \quad (3.19)$$

$$R' = \frac{V - R}{V - \min(R, G, B)} \quad (3.20)$$

$$G' = \frac{V - G}{V - \min(R, G, B)} \quad (3.21)$$

$$B' = \frac{V - B}{V - \min(R, G, B)} \quad (3.22)$$

$$6H = \begin{cases} 5 + B' & \text{wenn } R = \max(R, G, B) \text{ und } G = \min(R, G, B) \\ 1 - G' & \text{wenn } R = \max(R, G, B) \text{ und } G \neq \min(R, G, B) \\ 1 + R' & \text{wenn } G = \max(R, G, B) \text{ und } B = \min(R, G, B) \\ 3 - B' & \text{wenn } G = \max(R, G, B) \text{ und } B \neq \min(R, G, B) \\ 3 + G' & \text{wenn } B = \max(R, G, B) \text{ und } R = \min(R, G, B) \\ 5 - R' & \text{sonst} \end{cases} \quad (3.23)$$

Im HSV-Farbraum kann die Ähnlichkeit zweier Farbpunkte $(H_1, S_1, V_1), (H_2, S_2, V_2)$ eben-

¹genaugenommen ist die Grundfläche nicht kreisförmig, sondern sechseckig

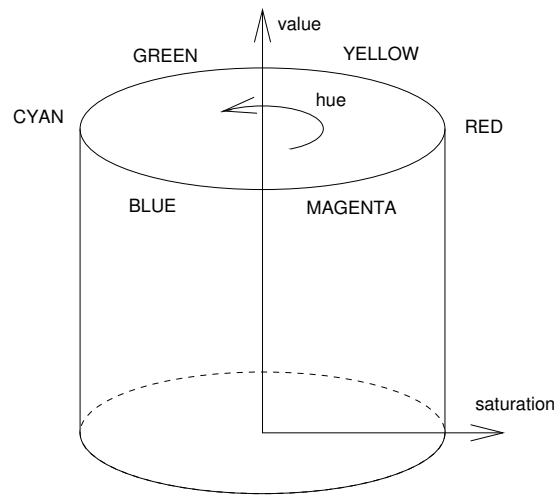


Abbildung 3.7: HSV-Farbraum

falls recht einfach durch die euklidische Distanz zwischen ihnen ausgedrückt werden:

$$d = 1 - \frac{1}{\sqrt{5}} \sqrt{(V_1 - V_2)^2 + (S_1 \cos H_1 - S_2 \cos H_2)^2 + (S_1 \sin H_1 - S_2 \sin H_2)^2}. \quad (3.24)$$

Durch die Normierung mit $1/\sqrt{5}$ wird erreicht, daß $d = 1$ für identische Farben und $d = 0$ für Farben mit für das HSV-System maximal möglichem Abstand.

Ein weiteres System aus dieser Gruppe ist das IHS-Farbsystem. Dieses berücksichtigt, daß die visuelle Farbvielfalt im sehr dunklen und hellen Bereich gemindert ist. Außerdem berücksichtigt es folgenden Konflikt im HSV-System: Bei maximaler Helligkeit wird eine Farbe weiß, d.h. sie wird ungesättigt. Das gleiche gilt für sehr dunkle Farben. Aus diesem Grund ist es sinnvoll, den Zylinder zu einem Doppelkegel umzuformen wie in Abbildung 3.8 dargestellt. Das dort angegebene System ist das IHS-System. Es entsteht durch folgende nichtlineare Transformation aus dem RGB-System [43]:

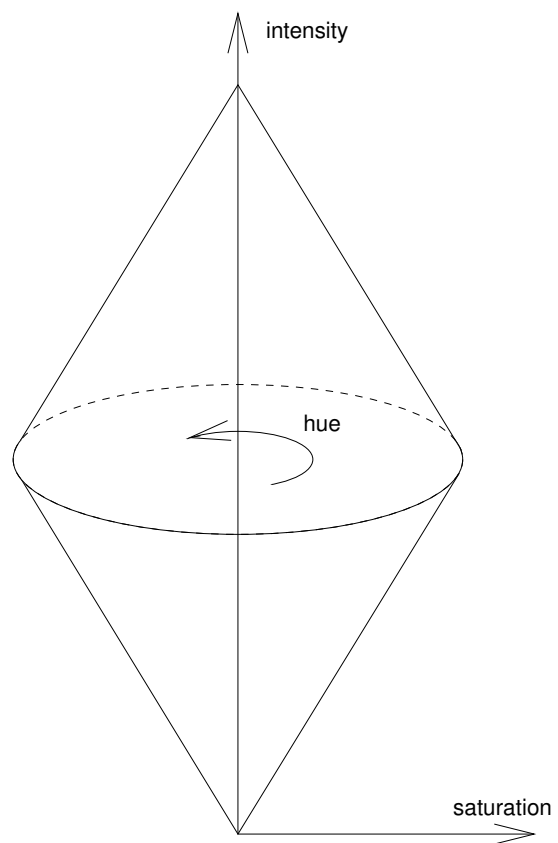


Abbildung 3.8: IHS-Farbraum

$$\begin{pmatrix} I \\ V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{6}} & \frac{1}{3} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & 0 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (3.25)$$

$$H = \tan^{-1} \frac{V_2}{V_1} \quad (3.26)$$

$$S = \sqrt{V_1^2 + V_2^2} \quad (3.27)$$

Auch hier gilt die vorige anschauliche Zuordnung: Intensity – Helligkeit, Hue – Farbton, Saturation – Sättigung.

Der PhotoCD-Farbraum YCC

PhotoYCC ist ein von Kodak und Philips 1991 im Rahmen des Kodak Photo CD Systems entwickelter Farbraum, der sich sehr stark an La^*b^* anlehnt, aber für die digitale Bildverarbeitung durch eine nichtlineare Transformation optimiert wurde. Die beiden Farbdifferenzkanäle weisen zudem eine unterschiedliche Stufenskalierung auf, bei der speziell die Farben, die in fotografischen Farbbildern vorkommen berücksichtigt wurden. Deshalb bietet YCC bei gleicher Quantisierung in der Praxis eine bessere Farbauflösung als La^*b^* . Zudem ist YCC vom Rechenaufwand her geeignet in Hinblick auf eine schnelle Bildverarbeitung und Darstellung auf einem Monitor.

Ein weiterer Aspekt ist die Ausweitung des darstellbaren Farbraums. Wie wir gesehen hatten, ist durch die vorgegebenen Primärstrahler der Bereich der darstellbaren Farben erheblich kleiner als der vom Menschen wahrnehmbare Farbraum. Dies ist zur Zeit eine unbedeutende Einschränkung, da Monitore ohnehin nur diesen Farbbereich darstellen können. Aber es ist ja durchaus zu erwarten, daß künftige Displaysysteme diesen Restriktionen nicht mehr unterliegen (z.B. Laserprojektionssysteme). YCC läßt daher RGB-Werte im Bereich $[-0.2, 2.0]$ statt $[0, 1]$ zu und kann somit auch Farben repräsentieren, die außerhalb des Farbraumdreiecks der Primärvalenzen liegen. Der Weißpunkt (100% Reflexion) liegt zum Beispiel bei $Y=182$, so daß noch Raum (bis $Y=255$) für extreme Glanzlichter ist. Diese kommen in realen Szenen durchaus vor und können von fotografischen Filmen im Gegensatz zu herkömmlichen Farbsystemen gut dargestellt werden.

Auch in YCC sind Luminanz- und Farbartkomponenten getrennt. Somit können auch hier ohne sichtbaren Schärfeverlust die Chroma-Komponenten mit der halben Abtastfrequenz gesampelt werden. Dies spart zum einen Speicherplatz auf der Photo CD, zum anderen beschleunigt es auch die Verarbeitung der Bilddaten.

Was dieses System im Vergleich zu herkömmlichen Farbformaten aber insbesondere auszeichnet ist, daß die Aufnahmesituation standardisiert ist. So ist z.B. die spektrale Empfindlichkeit der Sensoren genau vordefiniert. Es können also keine geräteabhängigen Verfälschungen entstehen. Insbesondere entfällt sogar die Abhängigkeit der Farbe vom Film (so haben z.B. Kodak-Farbnegativfilme einen eher bräunlichen Ton, während Fuji eher die grünen Farben betont). Die Eigenschaften des Filmes werden beim Scannen der Bilder herausgerechnet und somit die realen Farben der Szene rekonstruiert.

3.4 Farbnormalisierung

Ein im Bereich der automatischen Bildanalyse auftretendes Problem ist die Variation der Beleuchtung. So führt etwa eine Änderung der Beleuchtungsfarbe (Tageslicht oder Sonnenuntergang) zu völlig unterschiedlichen Bildern des gleichen Objekts. Oder bei Lageänderungen der Lichtquelle gegenüber dem Objekt oder umgekehrt treten Schatten an unterschiedlichen Stellen auf. Während der Mensch mit solchen Situation meist spielend fertig wird, erschweren diese Effekte die automatische Bildanalyse sehr, weshalb man in kritischen Anwendungen sorgfältigst darauf bedacht ist, konstante Aufnahmeverhältnisse zu schaffen. Dies bedeutet jedoch eine starke Einschränkung, die nicht immer in Kauf genommen werden kann. Daher wird versucht, die oben genannten Effekte in Modellen nachzubilden und mit deren Hilfe Transformationen zu finden, die ein konstantes Bild unter den oben beschriebenen Veränderungen liefern, also gegenüber solchen Variationen des Lichts invariant sind.

3.4.1 Modellbildung

In der Natur kommen zahlreiche unterschiedliche Effekte bei der Beleuchtung von Objekten zum Tragen. Hier sollen vereinfachend nur sogenannte Lambertsche Oberflächen betrachtet werden, d.h. Oberflächen mit perfekter Diffusion. Zudem soll zunächst nur der Fall einer einzigen Lichtquelle betrachtet werden.

Dann ergibt sich nach dem in Abbildung 3.9 dargestellten Modell die Sensorantwort am Ort $\hat{\mathbf{x}}$ des Sensors einer Kamera (oder des Auges) durch

$$p_i^{\hat{\mathbf{x}},E} = \langle \mathbf{e}^{\mathbf{x}}, \mathbf{n}^{\mathbf{x}} \rangle \int_{\lambda_{\min}}^{\lambda_{\max}} S^{\mathbf{x}}(\lambda) E(\lambda) F_i(\lambda) d\lambda, \quad i = 1, 2, 3. \quad (3.28)$$

Darin bezeichnet $\mathbf{e}^{\mathbf{x}}$ die Richtung der Lichtquelle von \mathbf{x} aus, $\mathbf{n}^{\mathbf{x}}$ die Oberflächennormale

in \mathbf{x} , $S^{\mathbf{x}}(\lambda)$ den prozentuales Anteil des reflektierten Lichts bei Wellenlänge λ , $E(\lambda)$ das Beleuchtungsspektrum und $F_i(\lambda)$ die Sensorcurve des Sensors i . Durch Substitution von

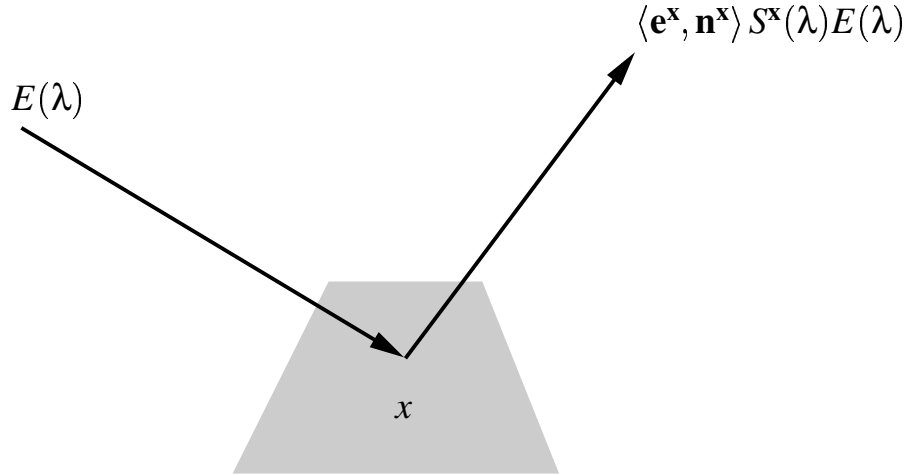


Abbildung 3.9: Beleuchtungsmodell

$\int_{\lambda_{\min}}^{\lambda_{\max}} S^{\mathbf{x}}(\lambda) E(\lambda) F_i(\lambda) d\lambda$ durch $q_i^{\mathbf{x},E}$ erhält man folgende Vereinfachung:

$$p_i^{\hat{\mathbf{x}},E} = q_i^{\mathbf{x},E} \langle \mathbf{e}^{\mathbf{x}}, \mathbf{n}^{\mathbf{x}} \rangle \quad (3.29)$$

Es ist leicht einzusehen, daß $q_i^{\mathbf{x},E}$ nicht durch eine veränderte Lage von Objekt und Lichtquelle beeinflusst wird, wohl aber durch Veränderung des Beleuchtungsspektrums.

Das obige Modell berücksichtigte nur eine punktförmige Lichtquelle. Für m Lichtquellen erhält man

$$p_i^{\hat{\mathbf{x}},E} = q_i^{\mathbf{x},E} \sum_{j=1}^m \langle \mathbf{e}_m^{\mathbf{x}}, \mathbf{n}^{\mathbf{x}} \rangle, \quad (3.30)$$

was rechnerisch durch $\mathbf{e}^{\mathbf{x}} = \sum_{j=1}^m \mathbf{e}_m^{\mathbf{x}}$ wieder zu einer effektiven Lichtquelle zusammengefaßt werden kann.

3.4.2 Normalisierung bzgl. der Beleuchtungsgeometrie

Mit dieser Modellierung läßt sich leicht intuitiv eine Normierung finden, die Invarianz gegenüber Änderungen der Beleuchtungsgeometrie schafft. Da alle Sensoren gleich

beeinflusst werden, ergibt sich folgende Normalisierung:

$$\frac{p_i^{\hat{x},E}}{\sum_{j=1,2,3} p_j^{\hat{x},E}} = \frac{q_i^{\mathbf{x},E} \langle \mathbf{e}^{\mathbf{x}}, \mathbf{n}^{\mathbf{x}} \rangle}{\sum_{j=1,2,3} q_j^{\mathbf{x},E} \langle \mathbf{e}^{\mathbf{x}}, \mathbf{n}^{\mathbf{x}} \rangle} = \frac{q_i^{\mathbf{x},E}}{\sum_{j=1,2,3} q_j^{\mathbf{x},E}}, \quad i = 1, 2, 3. \quad (3.31)$$

Diese Normierung haben wir schon in Kapitel 3.3.1 kennengelernt. Beim Übergang vom RGB- ins rgb-Farbsystem war folgende Normierung verwendet worden, die in der Literatur zur Computer Vision weitverbreitet ist, z.B. [11, 32, 22]:

$$r = \frac{R}{R + G + B} \quad (3.32)$$

$$g = \frac{G}{R + G + B} \quad (3.33)$$

$$b = \frac{B}{R + G + B}, \quad (3.34)$$

$$(3.35)$$

Für die spätere einheitliche Darstellung soll diese Normierung folgendermaßen formuliert werden: Die n Bildpunkte werden durch Aneinanderreihung der Zeilen in einem Vektor dargestellt, so daß man unter Berücksichtigung der 3 Farbkomponenten eine $n \times 3$ -Matrix erhält (Abbildung 3.10), im folgenden Farbbildmatrix genannt. Bei dieser Betrachtung bedeutet die eben gefundene Normalisierung also eine Zeilen-Normierung der Farbbildmatrix \mathbf{I} :

$$Z(\mathbf{I})_{i,j} = \frac{I_{i,j}}{\sum_{k=1,2,3} I_{i,k}}. \quad (3.36)$$

3.4.3 Normalisierung bzgl. des Beleuchtungsspektrums

Als nächstes soll der Einfluß der Verteilung des Beleuchtungsspektrums auf die Sensorantworten untersucht werden. Die Lage von Objektfläche und Lichtquelle soll also konstant gehalten werden, die Farbe der Lichtquelle hingegen darf sich ändern.

Zur Vereinfachung werden δ -förmige Empfindlichkeiten der Kamerasensoren angenommen, was natürlich nur eine grobe Näherung darstellt. Mit $F_i(\lambda) = \delta(\lambda - \lambda_i)$, $i = 1, 2, 3$ vereinfacht sich Gleichung (3.28) zu:

$$p_i^{\hat{x},E} = \langle \mathbf{e}^{\mathbf{x}}, \mathbf{n}^{\mathbf{x}} \rangle \int_{\lambda_{\min}}^{\lambda_{\max}} S^{\mathbf{x}}(\lambda) E(\lambda) \delta(\lambda - \lambda_i) d\lambda = \langle \mathbf{e}^{\mathbf{x}}, \mathbf{n}^{\mathbf{x}} \rangle S^{\mathbf{x}}(\lambda_i) E(\lambda_i) \quad (3.37)$$

R_1	G_1	B_1
R_2	G_2	B_2
R_3	G_3	B_3
R_4	G_4	B_4
R_5	G_5	B_5
R_6	G_6	B_6
\vdots	\vdots	\vdots
R_{n-1}	G_{n-1}	B_{n-1}
R_n	G_n	B_n

Abbildung 3.10: Matrixdarstellung **I** eines Farbbildes.

Für ein anderes Beleuchtungsspektrum E_1 erhält man also

$$p_i^{\hat{\mathbf{x}}, E_1} = \langle \mathbf{e}^{\mathbf{x}}, \mathbf{n}^{\mathbf{x}} \rangle S^{\mathbf{x}}(\lambda_i) E_1(\lambda_i). \quad (3.38)$$

Kombiniert man beides, so sieht man, daß

$$p_i^{\hat{\mathbf{x}}, E_1} = \frac{E_1(\lambda_i)}{E(\lambda_i)} p_i^{\hat{\mathbf{x}}, E}, \quad (3.39)$$

d.h. innerhalb eines Sensors werden alle Werte um den gleichen Faktor skaliert. Bezeichne \mathbf{R} (bzw. \mathbf{G} oder \mathbf{B}) die erste (bzw. zweite oder dritte) Zeile der Matrix \mathbf{I} . Dann wird also durch eine Änderung des Beleuchtungsspektrums eine Transformation der Form $(\mathbf{R}, \mathbf{G}, \mathbf{B}) \rightarrow (\alpha\mathbf{R}, \beta\mathbf{G}, \gamma\mathbf{B})$ durchgeführt. Somit liegt es nahe, eine Normierung innerhalb jeden Farbkanals durchzuführen.

$$\frac{n}{3} \frac{\alpha\mathbf{R}}{\sum_{i=1}^n R_i} = \frac{n}{3} \frac{\mathbf{R}}{\sum_{i=1}^n R_i} \quad (3.40)$$

$$\frac{n}{3} \frac{\beta\mathbf{G}}{\sum_{i=1}^n G_i} = \frac{n}{3} \frac{\mathbf{G}}{\sum_{i=1}^n G_i} \quad (3.41)$$

$$\frac{n}{3} \frac{\gamma\mathbf{B}}{\sum_{i=1}^n B_i} = \frac{n}{3} \frac{\mathbf{B}}{\sum_{i=1}^n B_i} \quad (3.42)$$

Der Vorfaktor $n/3$ ist hier so gewählt, daß die Summe aller Pixel nach Normierung n ergibt, wie auch schon im Fall der Beleuchtungsgeometrie-Normierung.

Wenn wir uns die Darstellung aus Abbildung 3.10 ins Gedächtnis rufen, so bedeutet diese Normalisierung also eine Normierung der Spalten der Farbbildmatrix \mathbf{I} :

$$S(\mathbf{I})_{i,j} = \frac{n}{3} \frac{I_{i,j}}{\sum_{k=1}^n I_{k,j}}. \quad (3.43)$$

3.4.4 Normalisierung bzgl. beider Effekte

Die Normalisierung bzgl. beider Effekte kann durch folgende Schritte erfolgen:

1. Initialisierung: $\mathbf{I}_0 = \mathbf{I}$
2. Iterationsschritt: $\mathbf{I}_{i+1} = S(Z(\mathbf{I}_i))$
3. Terminierungsbedingung $\mathbf{I}_{i+1} \cong (\mathbf{I}_i)$

Es erfolgt also eine fortgesetzte Hintereinanderausführung beider Normalisierungsprozeduren bis das Resultat sich nicht mehr ändert.

Offensichtlich macht dieses Vorgehen nur Sinn, wenn zwei Bedingungen erfüllt sind:

- Das Verfahren konvergiert und
- das Ergebnis ist eindeutig

In [15] werden diese beiden Eigenschaften nachgewiesen, so daß obige Iterationsvorschrift tatsächlich eine Normalisierung bzgl. beider Effekte durchführt.

3.5 Versuchsteil

In diesem Kapitel soll der Versuchsteil beschrieben werden. In Kapitel 3.5.1 wird zunächst der Versuchsaufbau dargestellt. Kapitel 3.5.2 gibt eine grundlegende Einführung in das Arbeiten mit Matlab. Danach folgen dann schließlich die Aufgabenstellungen für die Versuche.

3.5.1 Versuchsaufbau

Der Versuchsaufbau besteht aus einem PC, der mit einer MatroxTM Meteor II Bildaufnahmeplatte (Abbildung 3.11) zur Aufnahme eines Kamerabildes mit bis zu 768×576 Pixeln in 24 Bit Farbtiefe ausgestattet ist, sowie einer 3-Chip-Videokamera. 3-Chip-Kameras besitzen im Gegensatz zu herkömmlichen 1-Chip-Kameras je Farbebene einen lichtsensitiven Chip. Auf diese wird das einfallende Licht mit Hilfe eines Prismas aufgeteilt. Bei herkömmlichen Kameras hingegen sind auf einem Chip nebeneinander rot-, grün- und blau-empfindliche Sensoren plaziert. Dies bedeutet aber, daß die Farben mit einem örtlichen Versatz aufgenommen werden, der im Bildsignal vernachlässigt wird.

Außerdem steht ein Repräsentativ mit verschiedenfarbigen Leuchten zur Verfügung, die zudem leicht in ihrer Position zu verstellen sind.

Nach dem Starten von Matlab werden mit `setup_farbe` alle benötigten Funktionen und Variablen geladen. Zur Aufnahme eines Bildes existiert dann die Funktion `grab_palrgb` (Verwendung z.B. mit `bild=grab_palrgb`).

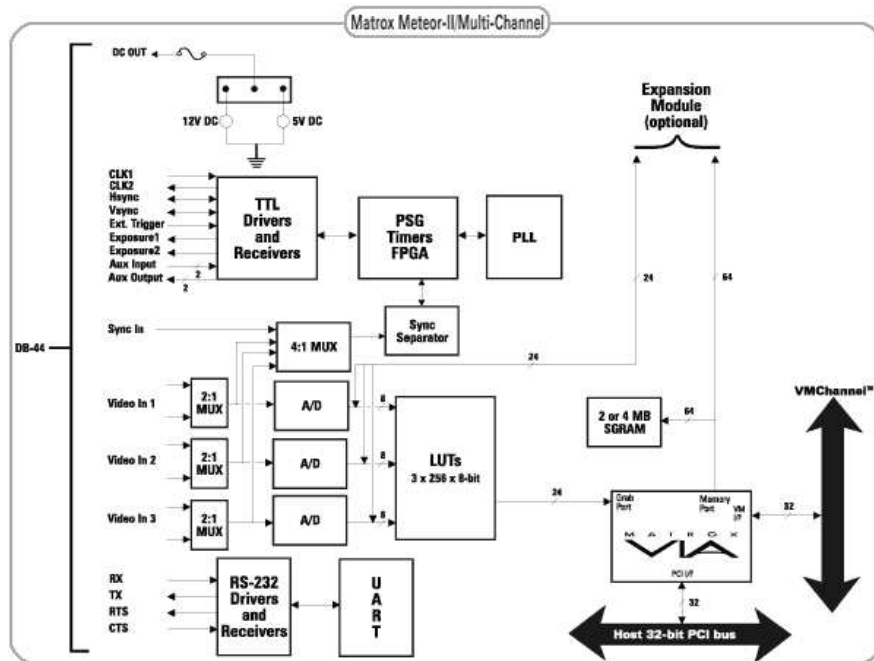


Abbildung 3.11: Blockschaltbild der Bildaufnahmekarte Matrox Meteor II.

3.5.2 Grundlegende Einführung in Matlab

Matlab ist eine interaktive Programmierumgebung zur numerischen Berechnung mathematischer Probleme. Der Name Matlab ist eine Kurzform für *Matrix Laboratory*, was anzeigt, daß Matrizenrechnung eine wichtige Stellung in Matlab einnimmt. In Matlab sind viele wichtige Verfahren der Numerik bereits als fertige Funktionen angeboten, wie z.B. die Inversion von Matrizen. Hier sollen nur die wichtigsten Funktionen beschrieben werden, für eine ausführliche Beschreibung der Software sei auf [34, 33] verwiesen.

Eine Variable kann definiert werden, indem man ihr einen Wert zuweist, z.B.

```
g=9.81;  
a=2e4;
```

Vektoren und Matrizen werden durch eckige Klammern markiert und Zeilen durch Semikolon voneinander getrennt, z.B.

```
vec=[1 2 3 4];  
mat=[1 2 3; 4 5 6; 7 8 9]
```

Wird der Abschluß durch ein ';' weggelassen, wird zudem das Ergebnis angezeigt, im zuletzt genannten Beispiel also

```
mat =  
  
     1     2     3  
     4     5     6  
     7     8     9
```

ausgegeben. Achtung, dies kann bei sehr großen Matrizen äußerst unerwünscht sein! Durch Einfabe des Variablennamens ohne Semikolon kann man sich also den Inhalt der Variablen anzeigen lassen. Die Befehle `who` and `whos` liefern eine Liste aller vom Benutzer verwendeten Variablen.

Matlab stellt zahlreiche Operatoren (+ - * / etc.) und Funktionen (sin, cos, exp, sqrt etc.) bereit.

Neben diesen einfachen elementaren Funktionen, kann man selbst auch eigene Funktionen definieren. Dies geschieht durch Anlegen eines Files mit dem Namen der Funktion und der Funktionsbeschreibung. Eine Funktion zur Berechnung einer Grauwertbildinwertierung (Wertebereich [0, 1]) könnte z.B. so aussehen:


```
function out=grauinv(in)
out=1-in;
```

Für die Durchführung dieses Versuchs könnten vor allem folgende Befehle von Nutzen sein: `size(bild)` ergibt die Dimension der Variablen `Bild`, in diesem Versuch wird man typischerweise eine $m \times n \times 3$ -Matrix vorliegen haben. Leider sind viele Matrixoperationen, wie z.B. die Matrizenmultiplikation `.*`, nur für $m \times n$ -Matrizen vordefiniert. Es lohnt sich daher teilweise, diese Einzelmatrizen nach folgendem Beispiel zu extrahieren, zu bearbeiten und anschließend wieder in eine höherdimensionale Matrix zurückzuschreiben.

```
r=bild(:,:,1);
erg(:,:,1)=matrix.*r;
g=bild(:,:,2);
erg(:,:,2)=matrix.*g;
b=bild(:,:,3);
erg(:,:,3)=matrix.*b;
```

Darin sind die `':'` Platzhalter für den gesamten Indexbereich der jeweiligen Dimension.

Für die Ausgabe der Resultate werden folgende Befehle benötigt: Mit `figure` wird ein neues Fenster geöffnet, in dem Graphen, Bilder etc. angezeigt werden können.

`plot(r)` zeichnet `r` über dem Intervall `[1...size(r)]`. Möchte man den Definitionsbereich selbst vorgeben, kann man dies durch Angabe einer weiteren Variable `a` gleicher Dimension tun: `plot(a,r)`. Dabei geben jeweils korrespondierende Elemente aus `a` und `r` eine Koordinate.

Oftmals ist es vorteilhaft mehrere Verläufe in einem Graphen darzustellen. Mittels `hold on` bleiben alte Verläufe stehen, während ein neuer eingezeichnet wird. Sinnvollerweise sollte man dann durch Übergabe noch weiterer Elemente in `plot` Farbe oder Strichelung variieren (näheres mit `help plot`).

`imshow(bild)` stellt Bilder dar. Diese sollten von der Dimension $m \times n \times 3$ und im Wertebereich `[0...1]` sein. Achtung: Aufgrund numerischer Ungenauigkeiten bei den Berechnungen kann dieser Wertebereich schon mal verlassen werden, obwohl theoretisch z.B. klar ist, daß das Ergebnis maximal 1 ergeben kann. Darum ist es ggf. notwendig, vor der Darstellung Werte > 1 auf 1 zu setzen. Dies geschieht besonders einfach mit

```
index=find(bild>1);
```

```
bild(index)=1;
```

Falls weitere Funktionen benötigt werden sollten oder Unklarheiten bestehen, erhält man mit `help` oder komfortabler mit `helpwin` eine umfassende Hilfe zu den Matlab-Funktionen.

3.5.3 Theoretische Aufgaben

Aufgabe 1: Was ist der Unterschied zwischen additiver und subtraktiver Farbmischung?

Aufgabe 2: Was bedeutet perzeptuelle Uniformität?

Aufgabe 3: Beschreiben sie den Begriff Helligkeit anhand des L*a*b* Farbsystems.

Aufgabe 4: Erklären sie anhand des XYZ Farbraumes was Chromazitätskoordinaten sind.

Aufgabe 5: Kann man Farbdifferenzen in beliebigen Farbsystemen durch den euklidischen Abstand bestimmen? Begründen sie ihre Antwort.

3.5.4 Aufgaben zur menschlichen Farbwahrnehmung

Aufgabe 1: Die Abbildung 3.2 werde in Matlab näherungsweise durch 3 Normalverteilungen mit den in Tabelle 3.1 angegebenen Parametern dargestellt.

Sensorkurve	μ	σ	Maximum
Blau	445	20	0.02
Grün	535	37	0.6
Rot	575	41	0.5

Tabelle 3.1: Normalverteilungsparameter der 3 Sensoren

Aufgabe 2: Berechnen Sie die Summe der drei Kurven und stellen Sie sie dar. Was repräsentiert diese Kurve anschaulich?

Aufgabe 3: Verwenden Sie wieder die Sensorcurven aus Aufgabe 3.5.4. Schauen Sie sich die in den Variablen C1, C2 und C3 vorgegebenen Farbspektren an. Was repräsentiert das Farbspektrum C1? Beurteilen Sie mit Hilfe der Sensorcurven, welche Farbspektren das menschliche Auge unterscheiden kann.

3.5.5 Aufgaben zur Farbdarstellung

Aufgabe 4: Gibt es einen anderen Satz von 3 Primärfarben als die RGB-Primärfarben, mit dem der gesamte Farbraum dargestellt werden könnte?

Aufgabe 5: Schreiben Sie ein Funktion zur Variation der Bildhelligkeit, indem Sie das Bild in das HSV-System konvertieren, es dort geeignet bearbeiten und anschließend zurückwandeln. Hinweise: In Matlab existieren bereits die Funktionen `rgb2hsv` und `hsv2rgb`, verwenden Sie das vorhandene Bild `farbe.tif`

Aufgabe 6: Schreiben Sie eine Funktionen, die bei Eingabe der RGB- bzw. HSV-Werte die zugehörige Farbe anzeigen. Versuchen Sie damit, die auf dem beim Versuch ausliegenden Farbmuster nachzubilden. Welche Möglichkeit der Farbdefinition würden Sie bevorzugen?

Aufgabe 7: Schreiben Sie eine Funktion zur Konvertierung eines Bildes von RGB nach YUV.

Aufgabe 8: In diesem Versuch soll der Vorteil des YUV-Systems gegenüber dem RGB-System bei der Farbbildübertragung demonstriert werden. Es werde angenommen, daß das Signal aufgrund begrenzter Kanalbandbreite nicht in voller Auflösung übertragen werden kann. Das Signal muß also mit verringerter Bandbreite übertragen werden, was sich als eine Tiefpaßfilterung äußert. Um den Effekt überdeutlich zu machen nehmen Sie einen – übertriebenen – Tiefpaß mit einer 9x9-Maske und konstantem Wert $1/81$ an. Schreiben Sie eine Routine, die alle Farbkomponenten des Bildes filtert (Tip: `conv2` führt eine zweidimensionale Faltung durch). Die Kanalbandbreite soll nun zulassen, daß einer der drei Farbkanäle original übertragen werden könnte. Schreiben Sie also eine Routine, die nur die 2 letzten Farbkomponenten filtert. Probieren Sie diese anhand des Bildes `farbe.tif` nun einmal im RGB-Raum und einmal im YUV-Raum aus (Konvertieren nach YUV, Filtern, Konvertieren nach RGB). Vergleichen Sie die Ergebnisse.

3.5.6 Aufgaben zur Farbnormalisierung

Aufgabe 9: Schreiben Sie Funktionen zur Normalisierung bzgl. der Beleuchtungsgeometrie, bzgl. des Beleuchtungsspektrums sowie bzgl. beider Effekte.

Aufgabe 10: Bauen Sie mit ein paar Objekten eine Bildszene unter der Kamera auf.

Aufgabe 11: Variieren Sie die Lage der Lichtquelle, nehmen Sie die Szene unter diesen veränderten Bedingungen auf und normieren Sie sie geeignet.

Aufgabe 12: Variieren Sie die Farbe der Lichtquelle, nehmen Sie die Szene unter diesen veränderten Bedingungen auf und normieren Sie sie geeignet.

Aufgabe 13: Variieren Sie Lage und Farbe der Lichtquelle, nehmen Sie die Szene unter diesen veränderten Bedingungen auf und normieren Sie sie geeignet.

Aufgabe 14: Vergleichen Sie die Ergebnisse. Wodurch könnten Abweichungen zwischen den einzelnen normierten Bildern erklärt werden?

Praktikumsversuch

Autofokus

S. Utcke und H. Burkhardt

4 Autofokus

4.1 Einleitung

Autofokus-Systeme sind aus der modernen Fotografie nicht mehr wegzudenken. Seit ihrer Einführung um das Jahr 1976 [46] haben sie sich quasi zum Industriestandard entwickelt, viele der führenden Kamerahersteller bieten mittlerweile Kleinbild-Spiegelreflex-Kameras ausschließlich mit Autofokus an¹. Gleiches gilt für Diaprojektoren, für die erste Autofokus-Lösungen noch früher auf den Markt kamen, sowie für nahezu alle am Markt erhältlichen Videokameras. Selbst in relativ günstigen Kompakt-Kameras gehört Autofokus mittlerweile zum Standard. Zusehends hält die Autofokus-Technik auch in andere Bereiche, wie zum Beispiel Ferngläser, Einzug.

Dieses Praktikum versucht, den Teilnehmern verschiedenen Möglichkeiten für Autofokus-Systeme vorzustellen, wie sie in ähnlicher Form in kommerziell erhältlichen Systemen verwendet werden oder Verwendung finden könnten. Die vorgestellten Verfahren lassen sich grob in zwei unterschiedliche Kategorien einteilen. Kontrastbasierte Verfahren werden in Abschnitt 4.2 vorgestellt, während Abschnitt 4.3 zwei unterschiedliche Familien von Triangulationsverfahren vorstellt; zum einen passive Verfahren in Abschnitt 4.3.1, zum anderen aktive Verfahren in Abschnitt 4.3.2. Zusätzlich stellt Abschnitt 4.4 weitere kommerziell eingesetzte Verfahren vor, die jedoch im Rahmen dieses Praktikums nicht näher untersucht werden. An diese Einführung in die Theorie schließt sich der Fragenteil in Abschnitt 4.5 an.

Achtung: Die Fragen in Abschnitt 4.5.1 sollten vorbereitend bereits vor Praktikumsbeginn bearbeitet werden!

¹Eine der wenigen Ausnahmen ist Leica, deren Kameras mit Wechselobjektiven ausschließlich ohne Autofokus angeboten werden.

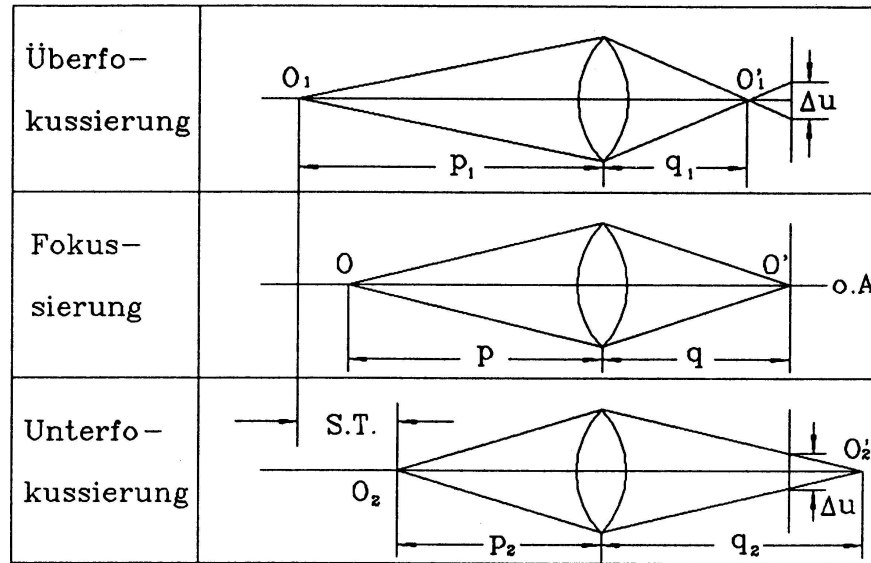


Abbildung 4.1: Schärfentiefe [29].

4.2 Kontrastbasierte Verfahren

Um die Funktionsweise Kontrast-basierter Verfahren zu verstehen ist es wichtig, eine Vorstellung über den Abbildungsvorgang und die Ursachen von Unschärfe zu besitzen. Aus diesem Grund stellt Abschnitt 4.2.1 noch einmal die wesentlichsten Gleichungen vor. Die Abschnitte 4.2.2 und 4.2.3 beschreiben mögliche Kontrast-basierte Verfahren.

4.2.1 Ursachen der Unschärfe

Normalsichtige Menschen sind es gewohnt, ihre Umwelt stets scharf wahrzunehmen. Dies ist ein automatischer Prozeß, es bedarf einer bewußten Anstrengung, um einen Gegenstand unscharf zu sehen. Aus diesem Grund vergessen wir Menschen leicht, daß bei der Abbildung mit Hilfe einer Linse eine unscharfe Abbildung die Regel ist, während eine scharfe Abbildung typischerweise nur für wenige Punkte gleichzeitig möglich ist. Warum ist das so? Für dünne Linsen — und näherungsweise auch für viele Objektive — gilt die Gaußsche Linsengleichung, nach der der Objektabstand p über die Brennweite

der Linse f mit dem Bildabstand q verknüpft ist

$$\frac{1}{p} + \frac{1}{q} = \frac{1}{f} \quad (4.1)$$

Damit ist klar, daß bei gegebener Brennweite f und Bildabstand q (Einstellung am Objektiv) nur Punkte scharf abgebildet werden, die sich genau im Abstand p vor dem Objektiv befinden. Dies ist in Abbildung 4.1 noch einmal dargestellt. Wie man dort ebenfalls sieht, wird ein Punkt im Abstand $p_1 \neq p$ auf einen Kreis abgebildet, dessen Durchmesser sich mit Hilfe des zweiten Strahlensatzes berechnen läßt, es ist

$$\Delta u = d \frac{q - q_1}{q_1} = df \frac{p_1 - p}{p_1(p - f)} \quad (4.2)$$

Es ist offensichtlich, daß Δu proportional zum effektiven Linsendurchmesser d ist². Macht man die durchaus realistischen Näherungen $p \gg f$ und $p \gg \Delta p = p_1 - p$, so erhält man aus (4.2) die vereinfachte Gleichung

$$\Delta u = df \frac{\Delta p}{p^2} \quad (4.3)$$

Das heißt, der Durchmesser des Unschärfe-Kreises (und damit die Unschärfe) ist zusätzlich zum effektiven Linsendurchmesser d auch proportional zur Brennweite f — je größer die Brennweite, desto schneller erscheinen Objekte unscharf — und dem Abstand zwischen p und p_1 . Zusätzlich sieht man, daß das Problem um so irrelevanter wird, je größer der Objektabstand wird — bei Objekten in einigen hundert Metern Entfernung macht eine Fehlfokussierung um einige Meter kaum etwas aus, bei näher gelegenen Objekten kann sie jedoch wesentlich sein. Die Gleichung (4.3) beschreibt alle dem Amateurfotografen bekannten Phänomene.

Gleichung (4.2) beschreibt noch nicht die Helligkeitsverteilung über dem Unschärfe-Kreis. Diese ist gemäß der geometrischen Optik innerhalb des Unschärfebereichs annähernd konstant [29, 3], für einen Punkt der Helligkeit 1 gilt

$$h(x,y) = \begin{cases} \frac{4}{\pi \Delta u^2} & \text{für } x^2 + y^2 \leq \frac{\Delta u^2}{4} \\ 0 & \text{sonst} \end{cases} \quad (4.4)$$

²Der effektive Linsendurchmesser entspricht in guter Näherung dem Durchmesser der Blende bei normalen Objektiven.

Um verschiedene Linsenfehler wie Diffraction³, Aberration⁴ und andere Effekte besser modellieren zu können hat es sich jedoch bewährt, statt dessen die Helligkeitsverteilung durch eine zweidimensionale Gauß-Verteilung zu approximieren [29]

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2} \frac{x^2+y^2}{\sigma^2}} \quad (4.5)$$

Der Parameter σ beschreibt die optischen Eigenschaften der Linse. Er ist proportional zum Durchmesser des Unschärfekreises Δu . Die Proportionalitätskonstante muß typischerweise experimentell bestimmt werden. Zu beachten ist, daß Δu und damit auch σ und $h(x, y)$ natürlich Funktionen von f , q und dem aktuellen Objektstand $p(x, y)$ sind.

In der Systemtheorie wird (4.5) als die Punktantwort des Systems (des Objektivs) bezeichnet. Will man wissen, welchen Einfluß das Linsensystem auf den Abbildungsvorgang hat, so reicht es, die Punktantwort mit dem Eingangssignal zu falten. Dies entspricht einer Tiefpaß-Filterung, da die Fourier-Transformierte eines Gauß-Signals wieder ein Gauß ist:

$$h(r) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2} \frac{r^2}{\sigma^2}} \quad \circ \rightarrow \bullet \quad \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\sigma^2 \rho^2} = H(\rho) \quad (4.6)$$

In Abbildung 4.2 sind schematisch die Ortsübertragungsfunktionen $H(\rho)$ (der Frequenzgang) für verschiedene Werte von σ aufgetragen. Je größer σ wird, desto stärker wirkt die Linse wie ein Tiefpaß. Umgekehrt wirkt für $\sigma = 0$, d. h. ein perfekt fokussiertes Bild, die Linse wie ein Allpaß — sie läßt alle Ortsfrequenzen ungehindert durch.

Hieraus kann man ersehen, daß sich der Frequenzgehalt eines Bildes prinzipiell zur Beurteilung seiner Schärfe eignet. Ein scharfes Bild wird über einen maximalen Anteil hoher Frequenzen verfügen (man spricht auch vom Energiegehalt eines Bildes), während in einem unscharfen Bild die tiefen Frequenzen dominieren werden. Bei der Beurteilung der Bildschärfe wird im wesentlichen zwischen zwei unterschiedlichen Verfahren unterschieden; den relativen Verfahren in Abschnitt 4.2.2 (in der Literatur oft als "Depth from Focus" bezeichnet) und den absoluten Verfahren in Abschnitt 4.2.3 ("Depth from Defocus" in der Literatur).

³Lichtbeugung [24].

⁴Wellenlängen- oder Lage-abhängige Brennweitenänderung [24].

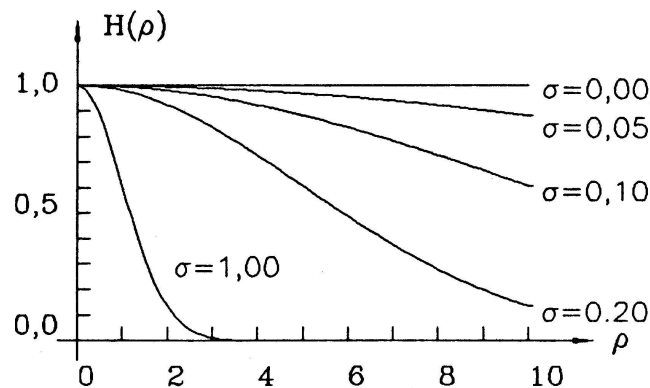


Abbildung 4.2: Schematische Darstellung der Ortsübertragungsfunktion [29].

4.2.2 Relative Verfahren

Im Laufe der Jahre wurden eine Reihe von Funktionen zur Beurteilung der Bildschärfe näher untersucht. So finden sich z. B. in [29] acht mögliche Gütefunktionen für die Beurteilung der Bildschärfe. Allen acht Verfahren ist gemein, daß es sich um Verfahren zur Beurteilung der relativen Bildschärfe handelt. Das bedeutet, daß in allen Fällen mehrere Aufnahmen derselben Szene untersucht werden müssen, von denen dann die Aufnahme mit dem relativ höchsten Kontrast gewählt wird. Im folgenden werden zwei der Gütefunktionen aus [29, Seite 45 ff] näher vorgestellt, viele andere sind jedoch denkbar (vergleiche auch Aufgabe 4.5.1). Wie bereits in Abschnitt 4.2.1 auf Seite 122 erwähnt ist die generelle Idee all dieser Verfahren, ein Maß für die hochfrequenten Anteile (oftmals bezogen auf die tieffrequenten Anteile) zu berechnen, da eine Defokussierung wie ein Tiefpaß wirkt. Aus einer Reihe unterschiedlich fokussierter Aufnahmen des gleichen Objektes wird mithin die am besten fokussierte Aufnahme auch den größten Anteil hoher Frequenzen enthalten.

Die SMD der Grauwerte als Fokussierungsfunktion

SMD ist eine Abkürzung für "Sum Modulus Difference", die ein Maß für den absoluten Gradienten des Bildes darstellt. Betrachten wir ein einfaches Grauwertbild $g(x,y)$, so

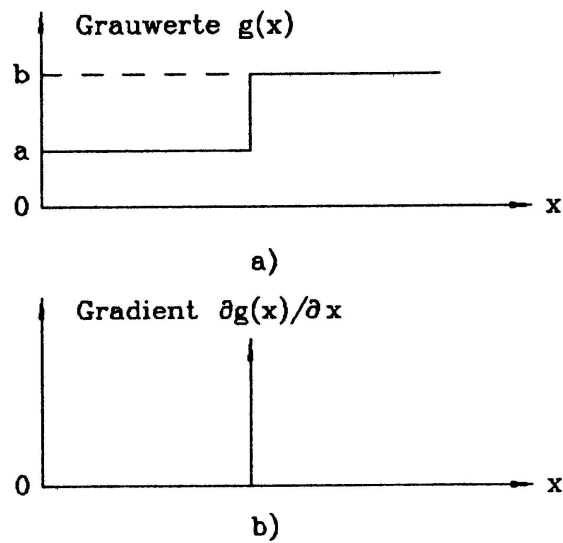


Abbildung 4.3: Eindimensionale Grauwertkante und ihr Gradient [29].

ist der Gradient eine stetige Funktion zweier Variablen

$$\nabla g(x, y) = \left(\frac{\partial g(x, y)}{\partial x}, \frac{\partial g(x, y)}{\partial y} \right)^T \quad (4.7)$$

Der Betrag des Gradienten ist demnach

$$|\nabla g(x, y)| = \sqrt{\left(\frac{\partial g(x, y)}{\partial x} \right)^2 + \left(\frac{\partial g(x, y)}{\partial y} \right)^2} \quad (4.8)$$

und die Richtung des Gradienten berechnet sich aus

$$\theta(\nabla g(x, y)) = \arctan \left(\frac{\partial g(x, y)}{\partial x} \cdot \left(\frac{\partial g(x, y)}{\partial y} \right)^{-1} \right) \quad (4.9)$$

Durch den Gradientenoperator werden kleine Strukturen im Bild hervorgehoben. Bild 4.2.2 zeigt, daß beispielsweise der Gradientenoperator für die Kantendetektion geeignet ist. Der Gradient hat an einer Kante ein Maximum. Im homogenen Bereich ist der Gradient Null.

Im Diskreten kann der Gradient nur approximiert werden. Die Herstellung eines Gradienten erfordert in jedem Bildpunkt die Differenzierung in zwei orthogonale Richtungen der Ortskoordinaten. Weil der Grauwert diskret ist, müssen die Differentialquotienten durch endliche Differenzenquotienten ersetzt werden

$$\frac{\partial g(x,y)}{\partial x} \approx \frac{g(x,y) - g(x-1,y)}{x - (x-1)} = g(x,y) - g(x-1,y) \quad (4.10)$$

$$\frac{\partial g(x,y)}{\partial y} \approx \frac{g(x,y) - g(x,y-1)}{y - (y-1)} = g(x,y) - g(x,y-1) \quad (4.11)$$

$\partial g(x,y)/\partial x$ und $\partial g(x,y)/\partial y$ sprechen auf horizontal und vertikal verlaufende Grauwertkanten entsprechend an.

Die Fokussierungsfunktion SMD hat drei Formen. Für ein Bild der Dimension $N \times M$ ist die erste Form

$$\text{SMD1} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} \left| \frac{\partial g(x,y)}{\partial x} \right| \approx \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} |g(x,y) - g(x-1,y)| \quad (4.12)$$

Sie extrahiert den Kantengradienten entlang der x -Richtung. Gleichfalls extrahiert die zweite Form

$$\text{SMD2} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \left| \frac{\partial g(x,y)}{\partial y} \right| \approx \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} |g(x,y) - g(x,y-1)| \quad (4.13)$$

den Kantengradienten entlang der y -Richtung. Die beiden Fokussierungsfunktionen SMD1 und SMD2 haben einen Nachteil. Für einige Bilder gibt es Kanten nur entlang der x -Richtung oder y -Richtung (Bild 4.4). In dieser Situation sind die Gradienten entlang der y - oder der x -Richtung gleich 0.

Zur Vermeidung dieses Problems kann der Betrag des Gradienten als Fokussierungsfunktion benutzt werden

$$\begin{aligned} \text{SMD3} &= \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sqrt{\left(\frac{\partial g(x,y)}{\partial x}\right)^2 + \left(\frac{\partial g(x,y)}{\partial y}\right)^2} \\ &\approx \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sqrt{(g(x,y) - g(x-1,y))^2 + (g(x,y) - g(x,y-1))^2} \end{aligned} \quad (4.14)$$

Daß die Ableitung tatsächlich ein geeigneter Operator zur Beurteilung des Energiegehalts eines Bildes ist, sieht man, wenn man die Fourier-Transformation idealer Ablei-

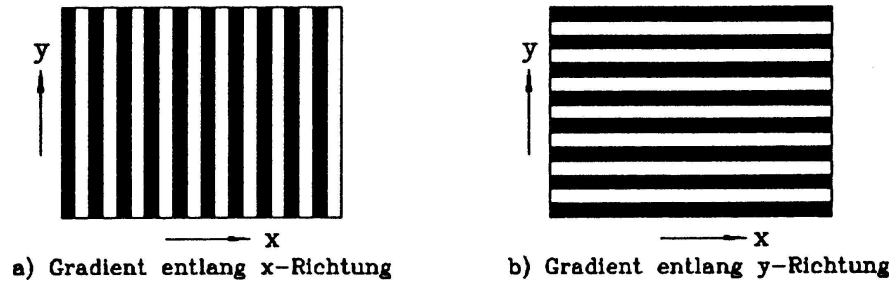


Abbildung 4.4: Zwei spezielle Kantenrichtungen [29].

tungsoperatoren betrachtet [26]:

$$\frac{\partial}{\partial x} g(x, y) \quad \circ \rightarrow \bullet \quad ju \cdot G(u, v) \quad (4.15)$$

$$\frac{\partial}{\partial y} g(x, y) \quad \circ \rightarrow \bullet \quad jv \cdot G(u, v) \quad (4.16)$$

Im Ortsfrequenzbereich bedeutet die Differentiation eine Multiplikation des Frequenzspektrums mit der Ortsfrequenz, d. h. es erfolgt eine Anhebung der hohen Frequenzspektren. Es wird deutlich, daß für ein defokales Bild die Leistungsspektren im hohen Frequenzbereich wegen der Defokussierung gefiltert sind, deswegen sind die Grauwertkanten nicht mehr so deutlich ausgeprägt wie die im fokalen Bild.

Die Leistungsspektren des Bildes als Fokussierungsfunktion

Die Leistungsspektren im hohen Ortsfrequenzbereich vom defokalen Bild werden unterdrückt. Man kann deshalb die Leistungsspektren direkt benutzen, um die Fokussierungsfunktion zu berechnen. Dazu muß eine zweidimensionale diskrete Fouriertransformation (DFT) durchgeführt werden. Man erhält so die diskrete Fouriertransformierte

$$G(u, v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} g(x, y) \cdot \exp \left(-2\pi j \left(\frac{xu}{N} + \frac{yv}{M} \right) \right) \quad (4.17)$$

Das Leistungsspektrum in der Ortsfrequenz (u, v) ist

$$G_\ell(u, v) = |G(u, v)|^2 \quad (4.18)$$

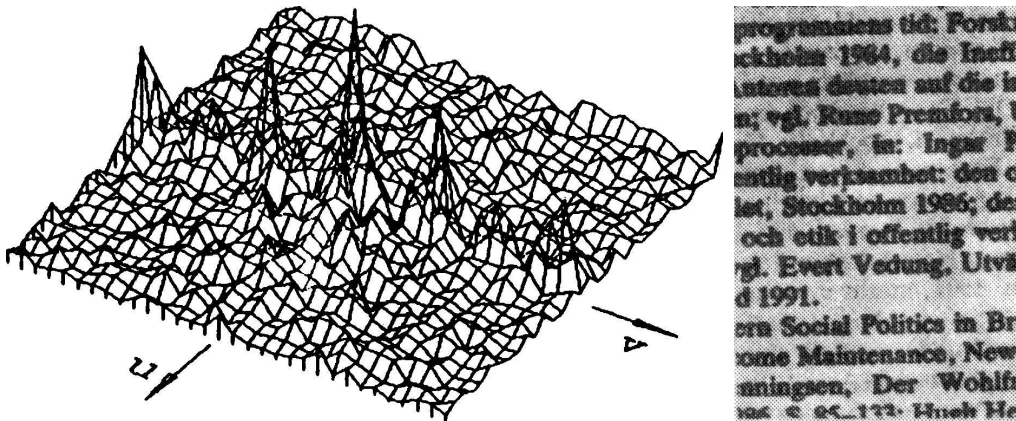


Abbildung 4.5: Leistungsspektrum vom fokalen Bild [29].

Die Fokussierungsfunktion kann durch die Summation der Leistungsspektren im hohen Frequenzbereich erzielt werden. Es ist

$$LS = \sum_{u \in \Psi} \sum_{v \in \Psi} G_{\ell}(u, v) \quad (4.19)$$

Ψ ist die Fläche des hohen Frequenzbereiches. In [29] besteht sie aus allen Ortsfrequenzen außerhalb des Ursprungs $(u, v) = (0, 0)$.

Die zweidimensionale diskrete Fouriertransformation ist ein zeitraubendes Verfahren. Wenn man eine zweidimensionale DFT eines $N \times N$ Bildes direkt nach Gleichung (4.17) berechnet, erfordert jeder Bildpunkt des transformierten Bildes, das ja durch die Multiplikation der entsprechenden Basisfunktion $\exp(-2\pi j(xu/N + yv/N))$ mit allen Bildpunkten des Bildes entsteht, N^2 komplexe Multiplikationen und $N^2 - 1$ komplexe Additionen. Die Durchführung der gesamten DFT erfordert daher insgesamt N^4 komplexe Multiplikationen und $N^2(N^2 - 1)$ komplexe Additionen. Mit der schnellen Fouriertransformation (FFT) kann die Rechenzeit enorm reduziert werden. Dazu wird die zweidimensionale DFT in zwei Schritten berechnet: es erfolgt nämlich die Zerlegung in eine Aufeinanderfolge von eindimensionalen DFT. Wegen der Separierbarkeit der Basisfunktion kann die Gleichung (4.17) entsprechend umgeformt werden

$$G(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \left(\frac{1}{N} \sum_{y=0}^{N-1} g(x, y) \exp(-2\pi j \frac{yv}{N}) \right) \exp(-2\pi j \frac{xu}{N}) \quad (4.20)$$

Die innere Summe stellt eine eindimensionale DFT der N Spalten dar, die äußere die

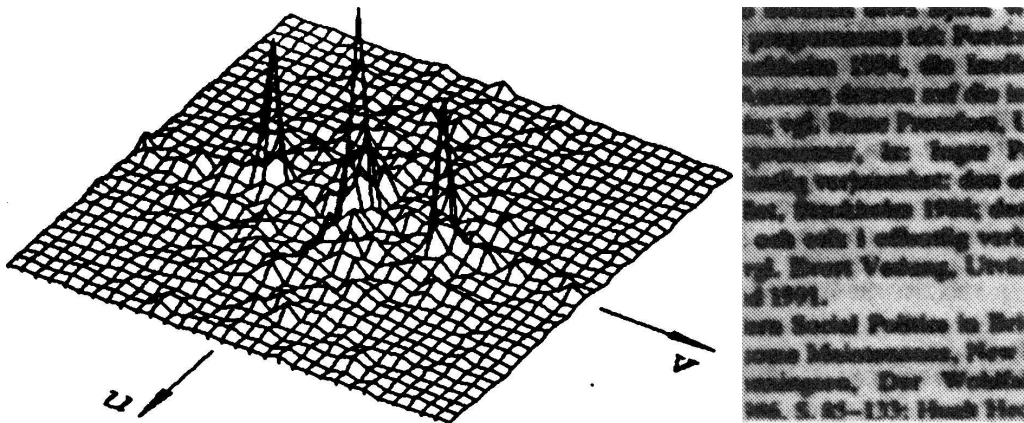


Abbildung 4.6: Leistungsspektrum vom defokalen Bild [29].

anschließende Transformation der N Zeilen. Die 2D-DFT läßt sich daher als Aufeinanderfolge zweier 1D-DFT ausführen, N Zeilentransformationen folgen N Spaltentransformationen auf dem $N \times N$ Bild:

$$\text{Spalten: } G(x, v) = \frac{1}{N} \sum_{y=0}^{N-1} g(x, y) \exp\left(-2\pi j \frac{yv}{N}\right) \quad (4.21)$$

$$\text{Zeilen: } G(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} G(x, v) \exp\left(-2\pi j \frac{xu}{N}\right) \quad (4.22)$$

Für jede der Teiltransformationen kann der 1D-FFT Algorithmus eingesetzt werden.

Für die qualitative Beurteilung der in einem Bild vorhandenen Ortsfrequenzen wird für das Leistungsspektrum $|G(u, v)|^2$ häufig eine dreidimensionale Darstellung angestrebt. Sollen in der dreidimensionalen Darstellung Spektralkomponenten mit kleinem Betrag stärker betont werden als solche mit großem Betrag, so kann die Funktion $G'(u, v)$ einer dreidimensionalen Darstellung von $|G(u, v)|^2$ berechnet werden gemäß

$$G'(u, v) = c \log(1 + |G(u, v)|^2) \quad (4.23)$$

wobei c ein Normierungsfaktor ist. Rechts in Bild 4.5 ist ein fokales Bild gezeigt und links die dreidimensionale Darstellung des Leistungsspektrums.

In diesem Fall sind die Leistungsspektren im hohen Frequenzbereich nicht unterdrückt. Im Gegensatz dazu zeigt das Bild 4.6 ein defokales Bild und die entsprechend dreidimensionale Darstellung des Spektrums. In dieser Situation sind die Leistungsspektren im hohen Frequenzbereich verschwunden oder stark unterdrückt.

4.2.3 Absolute Verfahren

Im Gegensatz zu den in der Literatur häufig "Depth from Focus" genannten relativen Verfahren in Abschnitt 4.2.2 erlauben es die absoluten Verfahren ("Depth from Defocus") prinzipiell, die Entfernung eines jeden Bildpunktes aus einer oder zwei Aufnahmen zu bestimmen. Maßgeblich entwickelt wurden diese Verfahren von Pentland [41, 40], der in [41] zwei Verfahren vorstellte, um die Tiefenverteilung in einer Szene entweder aus zwei oder aber auch aus nur einer Aufnahme der Szene zu bestimmen. Obwohl diese Verfahren nicht Gegenstand dieses Praktikums sind, soll der Vollständigkeit halber kurz auf sie eingegangen werden.

Der grundlegende Ansatz ist in beiden Fällen die Umkehrung von Gleichung (4.2). Es ist offensichtlich, daß man bei bekanntem Durchmesser Δu des Unschärfekreises auf den Objektstand p_1 zurückschließen kann⁵, es ist

$$p_1 = \frac{dfp}{df + \Delta u(f - p)} \quad (4.24)$$

Problematisch ist hierbei insbesondere die Bestimmung von Δu . Hierzu ist entweder Vorwissen über das Motiv nötig (z. B. das Vorhandensein steiler Kanten), oder aber eine zweite Aufnahme mit anderer Entfernungseinstellung⁶. In jedem Fall sind die erforderlichen Berechnungen — typischerweise iterative Verfahren oder die gleichzeitige Berechnung einiger Hundert Filterfunktionen — derart aufwendig, daß diesen Verfahren in der Praxis keinerlei Bedeutung zukommen. Dieses erklärt, warum diese Verfahren trotz ihrer langen Geschichte — Pentlands erste Arbeit auf diesem Gebiet ist von 1982 [40, Vergleiche [41]] — nach wie vor Gegenstand der Forschung sind. Aktuelle Arbeiten⁷ sind z. B. [12, 54].

4.3 Triangulationsverfahren

Die für kommerzielle Systeme mit weitem Abstand bedeutendste Gruppe der Autofokus-Verfahren bilden die Triangulationsverfahren. Der grundsätzliche Aufbau ist hier immer

⁵Tatsächlich ist das Ergebnis zweideutig, kann aber z. B. eindeutig gemacht werden, indem das Objektiv auf die nominelle Entfernung $p = \infty$ eingestellt wird.

⁶Tatsächlich kann auch ein anderer Parameter wie Brennweite oder effektiver Linsendurchmesser variiert werden.

⁷1998.

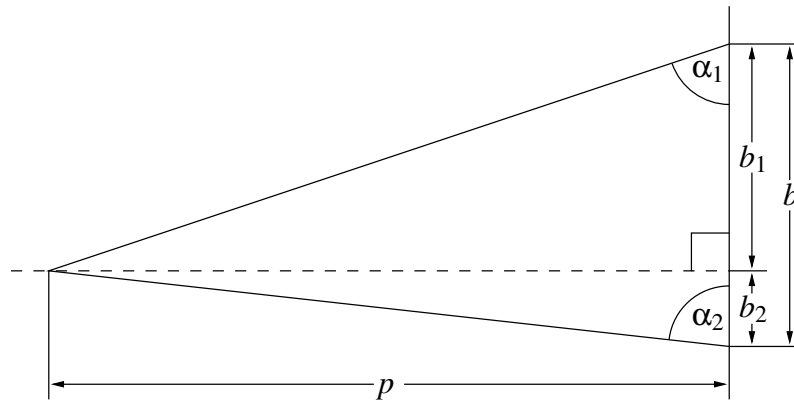


Abbildung 4.7: Typischer Grundaufbau von Triangulationsverfahren.

der gleiche (vergleiche Abbildung 4.7). Der Objektpunkt, auf den scharf gestellt werden soll, wird von zwei Sensoren mit dem sogenannten Basisabstand b unter zwei unterschiedlichen Winkeln α_1 und α_2 betrachtet. Aus dem Basisabstand und den beiden Winkeln lässt sich der Objektabstand p berechnen (siehe Abbildung 4.7), es ist

$$\begin{aligned} \tan(\alpha_1) &= \frac{p}{b_1} \\ \tan(\alpha_2) &= \frac{p}{b_2} \\ b &= b_1 + b_2 \\ \Rightarrow p &= b \frac{\tan(\alpha_1) \tan(\alpha_2)}{\tan(\alpha_1) + \tan(\alpha_2)} \end{aligned} \quad (4.25)$$

Derartige Aufbauten mit zwei passiven (empfangenden) Sensoren werden entsprechend als passive Verfahren bezeichnet. Der bekannteste Vertreter ist das klassische Stereoverfahren, wie es auch der Mensch selber zur Tiefenschätzung einsetzt. Diese Verfahren werden in Abschnitt 4.3.1 näher beschrieben. Ersetzt man hingegen einen der beiden Sensoren durch ein aktives Element (z. B. eine Laserdiode), so erhält man ein sogenanntes aktives Verfahren, vergleiche Abschnitt 4.3.2.

4.3.1 Passive Verfahren

In diesem Abschnitt werden verschiedene passive Triangulationsverfahren vorgestellt. Dieses ist zum einen das klassische Stereoverfahren in Abschnitt 4.3.1, wie es der

Mensch zur Tiefenschätzung einsetzt. Eine Variante dieses Verfahrens, das sogenannte Halbbildverfahren, wird im Praktikum eingesetzt. Daran anschließend wird in Abschnitt 4.3.1 das Verfahren vorgestellt, wie es in den meisten modernen Autofokus-Spiegelreflexkameras zum Einsatz kommt. Wesentlicher Unterschied ist hier, daß beide Stereobilder gleichzeitig durch das gleiche Objektiv aufgenommen werden, was den Einsatz in monokularen Kameras überhaupt erst möglich macht.

Klassische Stereoverfahren

Beim klassischen Stereoverfahren wird ein und dieselbe Szene von zwei unterschiedlichen Positionen aus betrachtet. Statt wie in Gleichung (4.25) die Betrachtungswinkel zu verwenden reicht es, die relative Verschiebung eines Bildpunktes bezüglich der jeweils anderen Aufnahme zu betrachten (Disparität, Parallaxe). Aus den Disparitäten der beiden Ansichten kann theoretisch eine fast komplette Tiefenkarte der Szene aus der Blickrichtung der beiden Sensorpositionen bestimmt werden. Die einzige Ausnahme bilden Regionen, die jeweils nur in einer der beiden Ansichten vorkommen, weil sie in der zweiten Ansicht durch weiter vorne liegende Szenenelemente verdeckt werden (sogenannte Selbstverdeckung)⁸.

Prinzipiell ist die Anordnung der beiden Sensoren zueinander — die sogenannte Epipolar-Geometrie — ohne Bedeutung (vergleiche die in weitem Bereich frei beweglichen Augen des Menschen oder [14, Seite 165 ff]). Traditionell wird jedoch ein Aufbau gewählt, in dem zwei Kameras derart angeordnet werden, daß sich die beiden Bildebenen in der gleichen Ebene befinden und — im Falle einer CCD-Kamera — die n -te Zeile der ersten Kamera auf einer Linie mit der n -ten Zeile der zweiten Kamera liegt. In diesem Fall kann ein Punkt auf der n -ten Zeile der ersten Kamera nur irgendwo auf der n -ten Zeile der zweiten Kamera liegen, so daß es prinzipiell ausreichend ist, jede Zeile getrennt zu betrachten⁹. Der Aufbau entspricht der Abbildung 4.8; d_1 und $-d_2$ sind die jeweiligen Abstände der Bilder eines Punktes vom Ursprung, so daß in dieser Anordnung für die

⁸Daraus ergibt sich sofort eines der Probleme bei der Stereobildauswertung. Um möglichst exakt den Abstand eines Szenenelements von den Sensoren zu bestimmen ist es geraten, den Basisabstand möglichst groß zu wählen. Andererseits steigt aber die Größe und Anzahl selbstverdeckter Regionen mit wachsendem Basisabstand.

⁹Tatsächlich können gerade in Zweifelsfällen die benachbarte Zeilen wertvolle zusätzliche Informationen liefern, vergleiche [30].

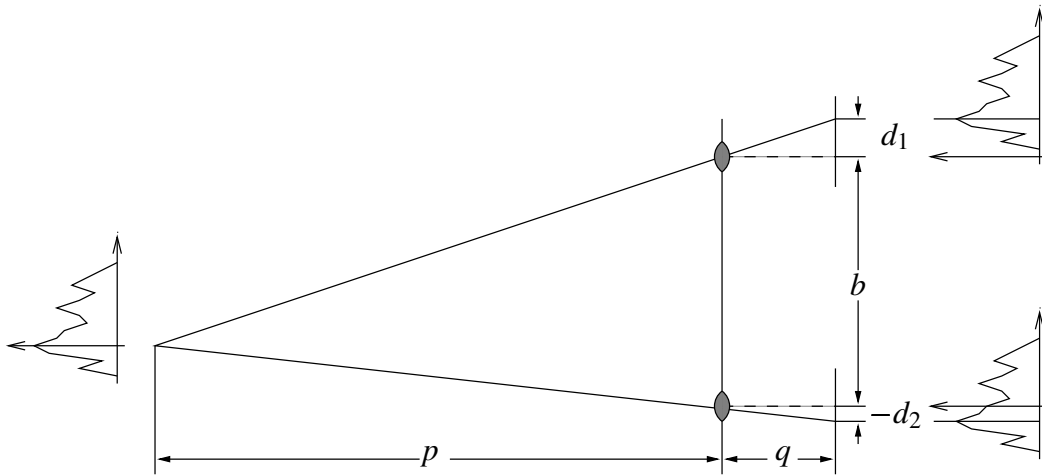


Abbildung 4.8: Typischer Stereoaufbau mit parallelen Kameras.

Disparität d gilt: $d = d_1 + d_2$. Mit Hilfe des zweiten Strahlensatzes ergibt sich

$$\begin{aligned} \frac{p}{p+q} &= \frac{b}{b+d_1+d_2} \\ \Rightarrow p &= b \frac{q}{d_1+d_2} \end{aligned} \quad (4.26)$$

Prinzipiell gibt es die unterschiedlichsten Methoden, um die Disparitäten d_1 und d_2 (oder einfacher d) zu berechnen. In kommerziell erhältlichen Spiegelreflexkameras kommen in der Regel Korrelations-basierte Verfahren zur Anwendung. Die Korrelation entspricht hierbei dem Vergleich der beiden Sensorsignale für alle möglichen Disparitäten d . Dieser könnte gegebenenfalls durch schnelle iterative Verfahren wie den in der Vorlesung *Digitale Bildverarbeitung II* [7] vorgestellten modifizierten Newton-Raphson-Algorithmus ersetzt werden, vergleiche [9]. Darüber hinaus sind im Bereich des Stereosehens eine Vielzahl weiterer Algorithmen entwickelt worden, die versuchen, solchen Phänomenen wie Selbstverdeckung und Unterschieden in der Perspektive der beiden Kameras gerecht zu werden. Diese werden jedoch bisher in Autofokussystemen noch nicht eingesetzt, da der Aufwand dafür noch zu hoch ist.

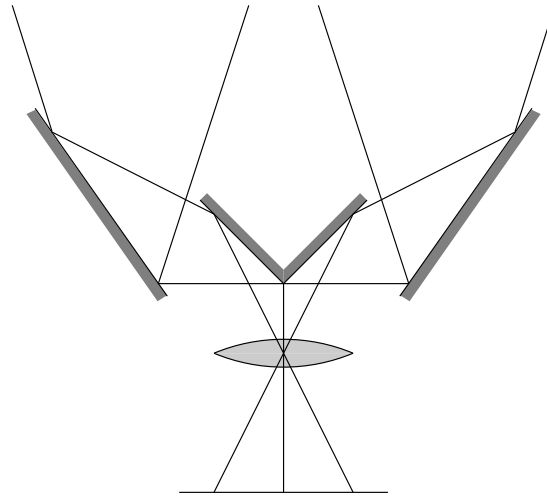


Abbildung 4.9: Typischer Aufbau eines Stereovorsatzes.

Halbbildverfahren

Beim Halbbildverfahren wird statt zweier Kameras nur eine Kamera mit einem Stereovorsatz (Abbildung 4.9) eingesetzt. Über Spiegel werden die Strahlengänge der beiden Sensoren soweit zusammengedrückt, daß sie auf einem Sensor Platz haben. Hierzu steht jedem Sensor je eine Hälfte der Sensorfläche zur Verfügung, deswegen auch die Bezeichnung Halbbildverfahren. Die gesamte übrige Theorie bleibt gleich.

Stereo mit monokularer Kamera

Wesentliche Voraussetzung für den Einsatz von Stereo-basierten Verfahren in industriell gefertigten Autofokus-Kameras ist die Möglichkeit, Stereo-Bilder durch das normale monokulare Objektiv der Kamera aufzunehmen. Tatsächlich ist es mit einem Aufbau, wie er in Abbildung 4.10 skizziert ist, möglich, Stereo-Aufnahmen durch ein normales Objektiv zu tätigen¹⁰. Hierbei werden die beiden Stereobilder durch den gegenüberliegenden Rand des Objektivs aufgenommen. Der Durchmesser des Objektivs bestimmt also den maximal möglichen Basisabstand. Obwohl der Aufbau in Abbildung 4.10 relativ komplex aussieht, kann man sich leicht davon überzeugen, daß es sich im wesentlichen um ein ganz normales Stereoverfahren handelt. Er soll im folgenden kurz

¹⁰Mit entsprechend stark eingeschränktem Öffnungswinkel.

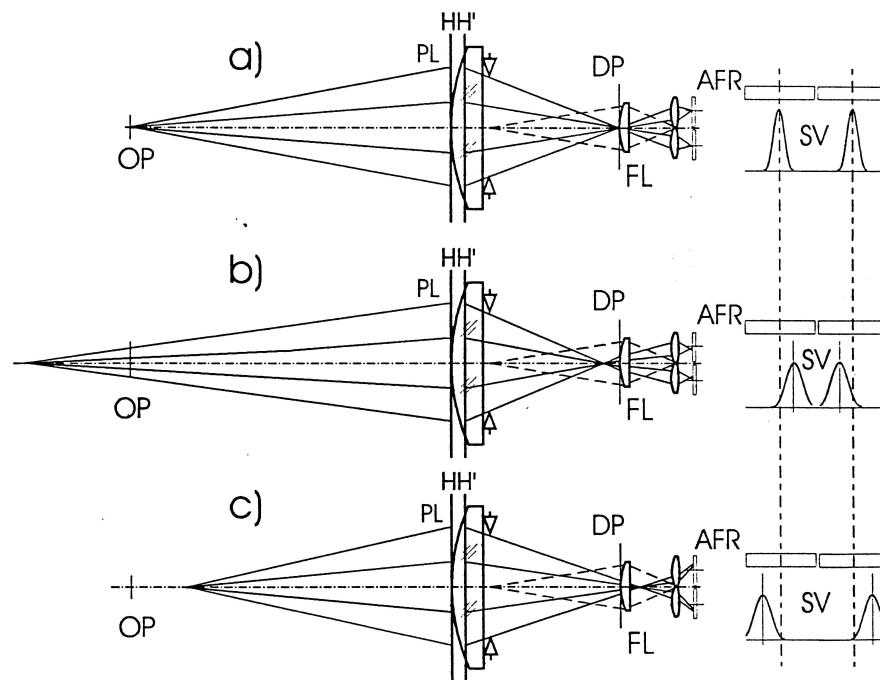


Abbildung 4.10: Typischer Stereo-Aufbau in kommerziell erhältlichen AF-Kameras (Rollei).

beschrieben werden.

Von einem Punkt auf der *object plane* OP aus gehen Lichtstrahlen durch die obere und untere Hälfte des Objektivs (*photographic lens*, PL). Genau an der Position des Filmmaterials DP, *detector plane*, befindet sich ein Prisma oder eine Linse FL, *field lens*. Ist die Kamera korrekt fokussiert, so gehen die Lichtstrahlen quasi ungebrochen durch das Zentrum dieser Linse hindurch und werden von einer weiteren Doppellinse in Normalposition auf die beiden Detektoren AFR, *autofocus detector rows*, projiziert. Ist die Kamera jedoch fehlfokussiert, so werden die Lichtstrahlen je nach Art der Fehlfokussierung entweder zur Mitte oder aber nach Außen aus ihrer Position abgelenkt. Man erhält ein Bild des Objekt-Punktes SV, *signal voltage*, das zum einen seitlich verschoben, zum anderen durch die Fehlfokussierung Tiefpaßgefiltert ist — in Abbildung 4.10 ist dies durch ein breiteres aber niedrigeres Signal angedeutet, das Signal ist sozusagen verschmiert.

Nahezu alle in kommerziellen SLR-Kameras eingesetzten Autofokus-Verfahren arbeiten nach dem in Abbildung 4.10 skizzierten oder sehr ähnlichen Prinzipien. Derartige Verfahren werden häufig unter Namen wie Visiotronic, Correfot, Kontrastvergleich, Phasenvergleich oder Phasenkorrelation gehandelt. Statt direkt die Signale zu vergleichen, werden bei letzteren Verfahren oft durch ein bewegliches Prisma oder Gitter zwei Wechselströme erzeugt, deren Phase miteinander verglichen wird.

4.3.2 Aktive Verfahren

Beim aktiven Verfahren, wie es in den meisten AF-Kleinbild- oder Videokameras Verwendung findet, handelt es sich um ein modifiziertes Stereo-Verfahren. Hier wird einer der Sensoren durch eine Lichtquelle (Laserdiode) ersetzt. Diese projiziert einen Lichtpunkt auf das vor der Kamera befindliche Objekt. Anhand der Lage dieses Lichtpunktes im Bild ist es möglich, den Abstand zwischen Kamera und Objekt zu berechnen.

4.4 Weitere Verfahren

Zusätzlich zu den oben genannten Verfahren werden noch eine Reihe weiterer Autofokus-Verfahren in kommerziell erhältlichen Geräten eingesetzt. Hierbei handelt es sich jedoch in der Regel um Sonderlösungen einzelner Hersteller oder für einzelne Anwendungsgebiete, so daß ihnen keine allzu große Bedeutung zukommt. Zusätzlich ist all diesen

Verfahren gemein, daß sie nicht direkt die Bildschärfe auswerten, sondern stattdessen eine Zwischengröße regeln. Es handelt sich hier also bloß um *Fokus-Steuerungen*, eine Regelung der Bildschärfe findet nicht statt. Dadurch ergeben sich auch einige der Probleme mit diesen Verfahren, insbesondere der Bildlagemessung in Abschnitt 4.4.3.

4.4.1 Mechanische Kopplung

In Vergrößerungsgeräten wird häufig der Abstand zwischen Objektiv und Fotopapier mechanisch abgetastet und der Fokus entsprechend diesem Abstand und dem Abbildungsmaßstab automatisch verändert. Es ist jedoch leicht klar, daß es sich hierbei nur im weitesten Sinne um Autofokus handelt.

4.4.2 Laufzeitmessungen

Insbesondere Polaroid setzt in seinen Sofortbildkameras die Laufzeitmessung zur Entfernungsbestimmung ein. Hierbei gibt die Kamera einen kurzen Ultraschall-Impuls¹¹ ab. Dieser wird an festen Objekten reflektiert und zur Kamera zurückgeworfen. Aus der Laufzeit läßt sich sehr genau die Objektentfernung bestimmen. Prinzipiell läßt sich dieses Verfahren auch mit Lichtimpulsen anwenden, allerdings ist deren Laufzeit viel geringer und die Auswertung deshalb fehleranfälliger.

4.4.3 Bildlagemessung

In Diaprojektoren wird üblicherweise die Bildlagenmessung zur Steuerung des Autofokus benutzt. Da sich typischerweise der Projektionsabstand nicht ändert, können sich Unschärfen nur dadurch ergeben, daß entweder das gesamte Dia Spiel in der Führung hat (z.B. bei unterschiedlichen Rahmendicken) oder dadurch, daß sich glaslos gerahmte Dias unter Hitzeeinfluß verziehen. Um eine gute Fokussierung sicherzustellen reicht es also, diese Lageänderungen des Dias auszugleichen. Das Vorgehen hierzu ist denkbar einfach [50]. Ein Lichtstrahl wird unter feststehendem Winkel auf das Dia gerichtet. An der Diaoberfläche, d. h. der dem Objektiv am nächsten gelegenen Fläche wird der Strahl zu einem Lichtempfänger in Form von zwei dicht nebeneinander angeordneten Fotowiderständen reflektiert (Abbildung 4.11). Liegt diese Fläche nicht in der exakten

¹¹Tatsächlich verwendet Polaroid drei Impulse unterschiedlicher Frequenzen.

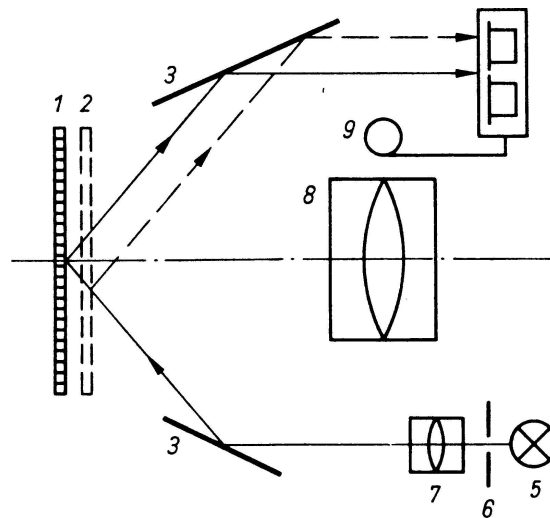


Abbildung 4.11: Prinzip der Bildlagenmessung nach [50].

Bezugsebene, so trifft der Lichtstrahl (gestrichelt dargestellt) einen der beiden Fotowiederstände stärker und erzeugt einen Stromfluß, der verstärkt wird und die motorische Schärferegulierung in Gang setzt.

Probleme hat dieses Verfahren, wenn Rahmen unterschiedlicher Glasstärke oder Rahmen mit und ohne Glas in einem Magazin kombiniert werden. In diesem Fall muß die Schärfe zwischendurch von Hand korrigiert werden. Kontrastbasierte Verfahren (Abschnitt 4.2) vermeiden dieses Problem und stellen deshalb eine gute (wengleich aufwendigere) Alternative dar.

4.5 Aufgaben

Die Aufgaben zerfallen in zwei Blöcke: vorbereitende Aufgaben, die von jedem Studenten vor Beginn des Praktikums zu lösen sind, in Abschnitt 4.5.1; die eigentlichen Praktikumsaufgaben, während des Praktikums zu bearbeiten, in Abschnitt 4.5.2.

Hinweis: Lesen Sie alle Aufgaben sorgfältig durch, bevor Sie mit der Lösung der ersten Aufgabe beginnen. Unter Umständen lassen sich mit ein wenig Vorausschau mehrere Aufgaben in einem Durchgang bearbeiten.

Zur Durchführung des Praktikums stehen unter anderem die folgenden Geräte zur Verfügung:

- Optische Bank
- 1/2" Progressive Scan CCD Firewire-Farbkamera, 658pxl × 494pxl
- Laser
- Stereovorsatz

4.5.1 Vorbereitende Aufgaben

Kontrastbasierte Verfahren

Aufgabe 4.5.1 Überlegen Sie Sich mindestens eine weitere Gütefunktion, mit deren Hilfe der Kontrast eines Bildes beurteilt werden kann (siehe Abschnitt 4.2). Warum ist die von Ihnen gewählte Funktion geeignet?

Aufgabe 4.5.2 In [29, Seite 104] wird die Kamera Pentax MEF erwähnt¹². Diese Kamera verfügt über zwei Sensoren (ähnlich Aufgabe 4.5.1), die im gleichen Abstand d vor und hinter der Bildebene positioniert sind. Überlegen Sie, welche Vorteile die Verwendung von zwei Sensoren bringen könnte. Überlegen Sie, wie diese Kamera im Praktikumsversuch mit nur einer CCD-Kamera simuliert werden kann.

Triangulationsverfahren

Aufgabe 4.5.3 Bestimmen Sie analytisch die zu erwartende Genauigkeit der Triangulationsverfahren in Abschnitt 4.3 in Abhängigkeit von der Auflösung der Sensoren. Geben sie ein realistisches Zahlenbeispiel.

Hinweis: Eine einfache Einführung in die Fehlerfortpflanzung findet sich in [4, Seiten 99f.] oder [25, Seiten 656f.].

Aufgabe 4.5.4 Überlegen Sie Sich ein Verfahren, um automatisch die Disparität zwischen zwei eindimensionalen Signalen (je eine Zeile aus dem Linken und rechten Stereobild) zu bestimmen.

¹²Nicht jedoch die Lösung für diese Aufgabe!

Aufgabe 4.5.5 *Wie würden Sie, unter Berücksichtigung der Ergebnisse aus Aufgabe 4.5.3, die Laserdiode für das in Abschnitt 4.3.2 beschriebene aktive Verfahren relativ zum Objektiv (Sensor) befestigen? Warum? Eine qualitative Begründung reicht.*

Hinweis: *Berücksichtigen Sie, daß von einer Autofokuskamera üblicherweise erwartet wird, auf Objekte im Bildzentrum scharf zu stellen.*

4.5.2 Versuchsaufgaben

Kontrastbasierte Verfahren

Aufgabe 4.5.6 Führen Sie Kontrastmessungen durch. Wählen Sie eine der beiden in Abschnitt 4.2 beschriebenen Gütefunktionen, und berechnen Sie deren Wert in Abhängigkeit von der am Objektiv eingestellten Entfernung. Nehmen Sie mindestens 10 Meßwerte, verteilt über den gesamten Meßbereich. Stellen Sie den Verlauf graphisch dar. Ergibt sich der von Ihnen erwartete Verlauf? Reichen 10 Meßwerte?

Aufgabe 4.5.7 Führen Sie die gleichen Schritte durch wie in Aufgabe 4.5.6, jedoch diesmal unter Verwendung der Gütefunktion aus Aufgabe 4.5.1. Vergleichen Sie die Ergebnisse.

Aufgabe 4.5.8 Führen Sie analog zu den Aufgaben 4.5.6 und 4.5.7 Messungen für die zur Verfügung gestellten Vorlagen mit unterschiedlichem Kontrast durch.

Aufgabe 4.5.9 In Aufgabe 4.5.2 wurde die Kamera Pentax MEF erwähnt, die über zwei Sensoren — im gleichen Abstand d vor und hinter der Bildebene positioniert — verfügt. Führen Sie Messungen entsprechend den Aufgaben 4.5.6, 4.5.7 und 4.5.8 durch. Welche Vorteile der von Pentax verwendeten Methode lassen sich erkennen?

Triangulationsverfahren

Aufgabe 4.5.10 Bringen Sie zwei eindimensionale Signale (je eine Zeile aus dem linken und rechten Stereobild) manuell zur Deckung. Wie würden Sie diesen Vorgang automatisieren (vergleiche Aufgabe 4.5.4)? Implementieren Sie einen entsprechenden Algorithmus.

Aufgabe 4.5.11 Führen Sie analog zu den Aufgaben 4.5.7 und 4.5.8 Messungen unter Verwendung des Stereovorsatzes durch. Verwenden Sie eine feste Entfernungseinstellung an der Kamera, und nehmen Sie Werte bei unterschiedlichen Entfernungen und mit unterschiedlich kontrastreichen Vorlagen auf. Stellen Sie das Ergebnis graphisch dar.

Aufgabe 4.5.12 Führen sie die gleichen Schritte wie in Aufgabe 4.5.11 unter Verwendung der Kamera und einer Laserdiode durch.

Achtung: Schauen Sie niemals direkt in den Laserstrahl!

Fehlerbetrachtungen

Aufgabe 4.5.13 Welche Aussagen lassen sich über die relative Genauigkeit der einzelnen Verfahren machen?

Hinweis: Berechnen Sie die Ableitung der jeweiligen Meßkurve. Die Größe des von einem konstanten Störsignal verdeckten Bereichs gibt einen Hinweis auf die Störanfälligkeit des Verfahrens. Vergleichen Sie mit den Ergebnissen aus Aufgabe 4.5.3.

Praktikumsversuch

Mustererkennung

A. Fenske und H. Burkhardt

5 Mustererkennung

5.1 Einleitung

Der Mustererkennung kommt in der Bildverarbeitung eine große Bedeutung zu, die motiviert wird vom dem Wunsch, das sehr komplexe optische Wahrnehmungsvermögen des Menschen mit Hilfe von Digitalrechnern nachzuahmen. Der Mensch ist in der Lage, aus der ungeheuren Datenflut, die das Auge dem Gehirn liefert, die für ihn wichtigen Informationen zu extrahieren und auch so zu speichern, daß ein Wiedererkennungsprozeß möglich ist.

Die meisten Bildverarbeitungssysteme arbeiten nach dem in der Abbildung 5.1 gezeigten Prinzip. Eine Bildszene wird mit einem Sensor aufgenommen und digitalisiert. Um bessere Daten zu erhalten, können Bildverbesserungsmaßnahmen oder eine Bildrestaurierung vorgenommen werden. Je nach Anwendung und auch im Hinblick auf eine später zu erfolgende Klassifikation können sehr unterschiedliche Vorverarbeitungsschritte sinnvoll sein. Maßnahmen wie Filterung und Histogrammskalierung werden im Praktikumsversuch **Bildvorverarbeitung** behandelt. Liefert das Bild die zu interessierenden Objekte innerhalb einer komplexen Szene, ist eine Segmentierung des Bildes erforderlich.

Ist das interessante Objekt vom Hintergrund getrennt, kann die eigentliche **Mustererkennung** beginnen. Sie setzt sich aus zwei Schritten zusammen, die **Merkmalsextraktion** und die **Klassifikation**. Je nach Anforderungsgrad erhalten die beiden Stufen ihre Gewichtung. Bei einer bildunterstützten Qualitätskontrolle beispielsweise, bei der das Objekt fest positioniert ist, kommt dem Klassifikator eine größere Bedeutung zu als der Merkmalsextraktion, die sich z.B. auf rein heuristische Messgrößen stützen kann. Sollen Objekte unabhängig von ihrer Lage (z.B. Translation und Rotation) erkannt werden, was das Ziel in diesem Versuch sein soll, ist eine aufwendigere **lageinvariante Merkmalsextraktion** erforderlich.

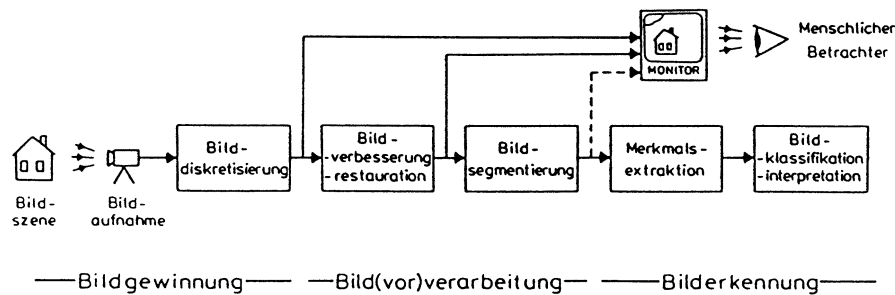


Abbildung 5.1: Stufen der Bildverarbeitung [8]

Dieser Praktikumsversuch beschränkt sich auf die Mustererkennung von einfachen, zusammenhängenden Objekten in Binärbildern, in denen nur zwei Helligkeitsstufen, weiß (1) und schwarz (0), vorhanden sind. Eine sinnvolle Unterscheidung zwischen den Objekten und dem Hintergrund erfolgt durch Binarisieren des Grauwertbildes, welches die Kamera liefert, mit Hilfe einer Schwellwertoperation aufgrund des Grauwertistogramms, wie sie bereits in dem Praktikumsversuch Bildvorverarbeitung behandelt wurde.

Da die Objekte als einfach zusammenhängend vorgegeben sind, sind sie durch ihre Konturlinie vollständig bestimmt. Daher beschäftigt sich Kapitel 5.2 mit der **Konturfindung**. Wichtig ist die Konturfindung auch im Hinblick auf die **Datenreduktion**, die wegen der großen Datenmengen oftmals unerlässlich ist.

In Kapitel 5.3 werden ausgehend von der Konturlinie primitive **geometrische Merkmale** der Objekte besprochen. Neben den Grundlagen der lageinvarianten Mustererkennung werden einige lageinvariante Merkmale vorgestellt, die aber nicht der Forderung der Vollständigkeit genügen, d.h. die nicht das Muster eindeutig kennzeichnen.

Durch einen direkten Vergleich des Testmusters mit allen Referenzmustern (matching) läßt sich dieser eindeutige Bezug herstellen. Wegen der zugelassenen Lagevarianz der Testobjekte ist dieses Verfahren allerdings zu rechenintensiv. Eine verfeinerte Methode, die charakteristische Merkmale der Muster und deren Beziehungen zueinander ausnutzt, ist das **Graph-matching**, das in Kapitel 5.4 vorgestellt wird. Durch geschickte Wahl der Merkmale und deren Relationen wird ein **Graph** definiert, der das Objekt eindeutig darstellt. Zur Klassifikation wird dann der Testgraph mit den Graphen der Referenzmuster verglichen.

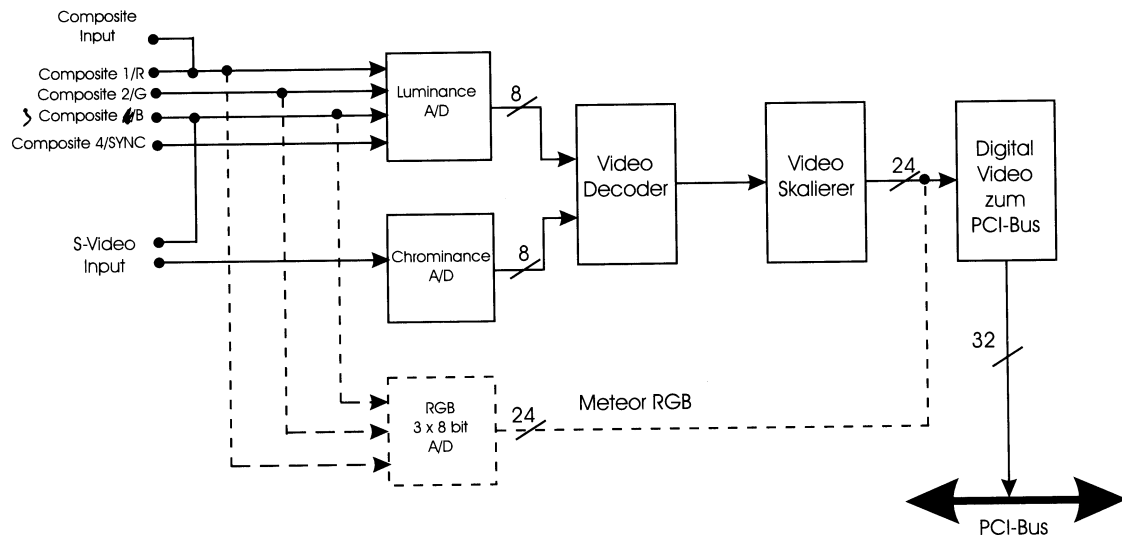


Abbildung 5.2: Bildverarbeitungskarte MatroxTM Meteor

Die in den bisherigen Kapiteln besprochenen Mustererkennungsverfahren versagen, falls die Kontur des Testmusters wesentlich gestört ist, z.B. ein Teil des Objektes fehlt, oder das Objekt teilweise von einem anderen überlappt ist. Das Problem des Segment-Vergleichs und von überlappenden Mustern wird in Kapitel 5.5 durch **Subgraph-matching** gelöst. Die Grunddefinitionen der Graphentheorie werden dabei anwendungsbezogen eingeführt.

Der Versuchsaufbau wird prinzipiell in Abbildung 5.1 deutlich. Analog aufgenommene Bilder werden diskretisiert dem Computer zugänglich gemacht. Dies geschieht durch eine spezielle Bildverarbeitungskarte (frame grabber board), die in Abbildung 5.2 schematisch erklärt ist. Ansonsten ist der Aufbau des Praktikumversuches unmittelbar einsehbar.

5.2 Konturfindung

Die Bestimmung der Kontur eines Objektes in einem Binärbild bereitet keine großen Schwierigkeiten, da eine Kante eines ausgedehnten Objektes einfach durch den Über-

gang von schwarz auf weiß, bzw. von 0 auf 1, definiert wird. In Abschnitt 5.2.1 wird die **Konturverfolgung** besprochen. Ein Algorithmus liefert die Konturpunkte in einer nach der Bogenlänge geordneten Folge. Der Abschnitt 5.2.2 behandelt die **Konturglättung**, die hochfrequente Störeinflüsse der Konturlinie beseitigt, und durch die eine Datenreduktion erreicht wird. Das hier beschriebene Verfahren approximiert die Objektkontur durch einen Polygonzug, also ausschließlich durch Geradenstücke. Ausgehend von dieser Polygonapproximation lassen sich in Kapitel 5.3 einfache geometrische Merkmale des Objektes berechnen. Eine weitere Möglichkeit zur Datenreduktion, ist die Kettenkodierung, die in Abschnitt 5.2.3 eingeführt wird.

5.2.1 Konturverfolgung

Die Kontur eines Objektes in einem Binärbild läßt sich mit einem Algorithmus, der die Übergänge von 0 zu 1 sucht, finden. Der Konturfinde-Algorithmus hat die Aufgabe, eine Folge von Zahlenpaaren zu ermitteln, welche die Positionen der Hell-Dunkel-Übergänge am Objektrand in der Reihenfolge eines Umlaufs angibt.

Als Startpunkt wird dem Algorithmus ein bekannter Konturpunkt übergeben, von dem er den Objektrand so verfolgt, daß rechts bzw. links der Suchrichtung dunkle oder helle Bildbereiche liegen. Wird der Startwert im Laufe der Suche wieder erreicht, so ist die Kontur geschlossen und die Aufgabe dieses Algorithmus vollendet. Eine einfache Prozedur (Kontur) ist [2] :

1. Fahre zeilenweise über das Bild, bis ein Objektpunkt als Startpunkt gefunden ist.
2. Man stelle sich vor, daß man auf den zuletzt erreichten Punkt zugegangen ist. Wenn er ein Objektpunkt ist, trage man ihn in die Liste ein und biege nach links ab, sonst biege man nur nach rechts ab.
3. Man wiederhole Schritt 2, bis der Startpunkt erreicht ist.

Die Abbildung 5.3 zeigt das Verfahren an einem Beispiel. Die Prozedur Kontur beschreibt die Objektregion nur konsistent bei einer Vierer-Nachbarschaft (s. Abbildung 5.4). Wie aus Aufgabe 5.1 deutlich wird, ist eine Abfrage nach doppelt gezählten Konturpunkten notwendig.

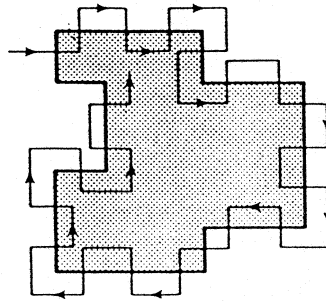


Abbildung 5.3: Konturverfolgung (Prozedur Kontur) [2]

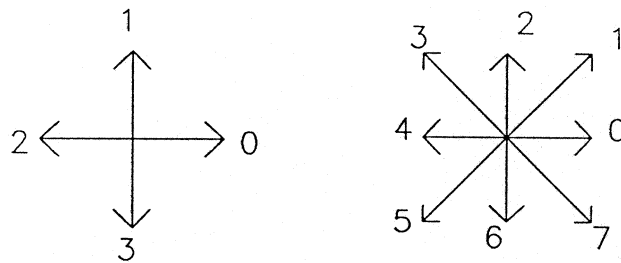
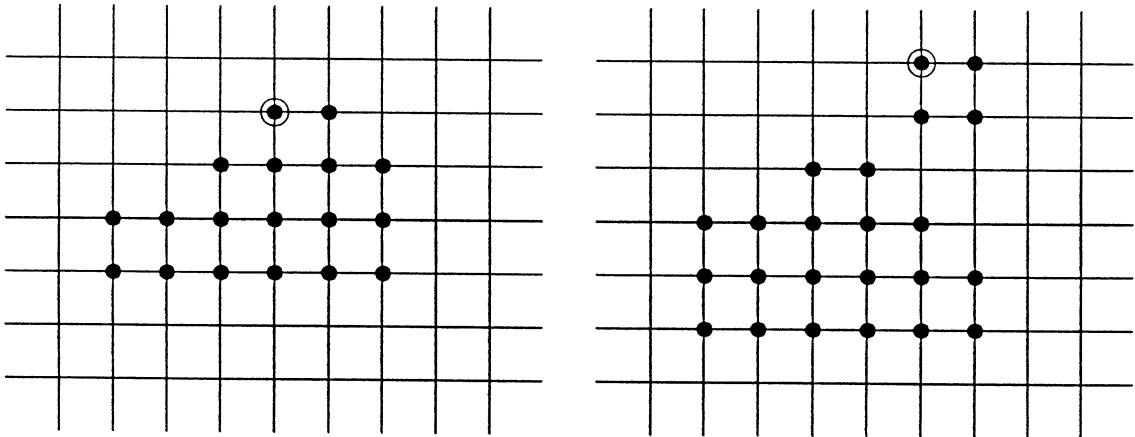


Abbildung 5.4: (a) Vierer- (b) Achternachbarschaft

Aufgabe 5.1 *Simulieren Sie den Algorithmus Kontur in folgenden zwei Beispielen mit \odot als Startpunkt, tragen Sie den Pfad ein und markieren Sie doppelt gezählte Konturpunkte.*



5.2.2 Konturglättung

Eine Objektkontur, die aus einem Binärbild extrahiert wird, unterliegt im allgemeinen Störungseinflüssen, die dazu führen daß die Kontur sich mit der tatsächlichen Randlinie nicht an jeder Stelle deckt. Die Störung kommt zustande durch ortsvariante Objektausleuchtung, durch Randpositionen, die sich nicht mit dem Abtastraster decken, und durch additive Rauschstörungen in der Videokamera. Diese Störeinflüsse sind im wesentlichen hochfrequenter Natur. Ein Glättungsverfahren soll derartige Fehler beseitigen. Gleichzeitig wird eine Datenreduktion angestrebt durch Zerlegung der Kontur in vorgegebene Randsegmente wie z.B. Kreisbögen, Winkelstücke, Geradenstücke usw. Es ist dann nicht mehr notwendig, jeden Randpunkt abzuspeichern, sondern nur noch die Art und Lage der einzelnen Segmente.

Das hier beschriebene Konturglättungsverfahren [45] approximiert die Kontur, die durch eine Zahlenfolge repräsentiert wird, durch einen Polygonzug, also ausschließlich durch Geradenstücke. Die Eckpunkte des Polygonzugs stellen den reduzierten Datensatz dar. Der Abbruch der Polygonzugadaptation tritt ein, wenn alle Konturpunkte zwischen zwei benachbarten Stützstellen des Polygonzugs innerhalb eines vorgegebenen Abstandmaßes der Breite E um das entsprechende Geradenstück liegen. Für Geradenstücke, bei denen das Kriterium nicht erfüllt ist, wird der Punkt mit maximalem Abstand von dem Geradenstück als zusätzliche Stützstelle des Polygonzugs eingesetzt. Der Algorithmus Polapprox der Polygonapproximation, der die in Kontur ermittelten Konturpunkte benutzt, sieht folgendermaßen aus :

1. Fahre zeilenweise über das Konturlinienbild, bis ein Konturpunkt als Startpunkt r_1 gefunden ist.
2. Bestimme den Konturpunkt r_2 mit dem größten Abstand davon, lege eine Gerade von r_1 , nach r_2 und führe die nächsten Schritte für die beiden Zweige des Polygons ($r_1 \rightarrow r_2$) und ($r_2 \rightarrow r_1$) getrennt aus.
3. Wenn der Abstand der Konturpunkte von allen Geraden kleiner ist als ε , dann ist die Approximation beendet, sonst fahre fort mit Schritt 4.
4. Für jede der bisher gefundenen Geraden mit zu großem Approximationsfehler führe Schritt 5 und 6 aus.
5. Ermittle den Punkt P mit dem größten Abstand von der Geraden.
6. Ersetze diese alte Gerade durch zwei neue Geraden mit P als mittlerem Eckpunkt.
7. Gehe zurück nach Schritt 3.
8. Weise alle Polygonzugpunkte zurück, die einen nicht genügend spitzen Winkel mit ihren zwei Nachbarpunkten bilden.

In Abbildung 5.5 ist ein Beispiel für die sukzessive Approximation durch die Polygonzugpunkte P_i dargestellt. Für die nachfolgenden Matching-Verfahren ist es wichtig, nur die signifikanten Punkte der Kontur zu finden, was bei dem in der Literatur unter *Split+Merge* bezeichnetem Algorithmus [5] nicht ausschließlich der Fall ist. Durch eine Abfrage nach dem Winkel an den Eckpunkten im Schritt 8 läßt sich das erreichen [49].

5.2.3 Kettenkodierung

Eine weitere Repräsentation der Konturlinie ist die Kettenkodierung (chain coding) [37]. Von einem Startpunkt beginnend wird jeweils die Richtung zum nächsten Punkt der Linie angegeben. Dabei werden vier oder acht Richtungen unterschieden, so daß zwei bzw. drei bit zur Kodierung einer Richtungsstrecke (link) ausreichen. Die vier oder acht Richtungen ergeben die in Abbildung 5.4 definierte 4- bzw. 8-Nachbarschaft.

Aufgabe 5.2 *Bilden Sie bei folgenden zwei Beispielen den 8-Kettencode mit \odot als Startpunkt (s. Abbildung 5.4), wobei die Laufrichtung im Uhrzeigersinn sein soll. Ist der Kettencode invariant gegenüber Translationen, Rotationen oder Skalierungen (Vergrößerung/Verkleinerung) des Musters?*

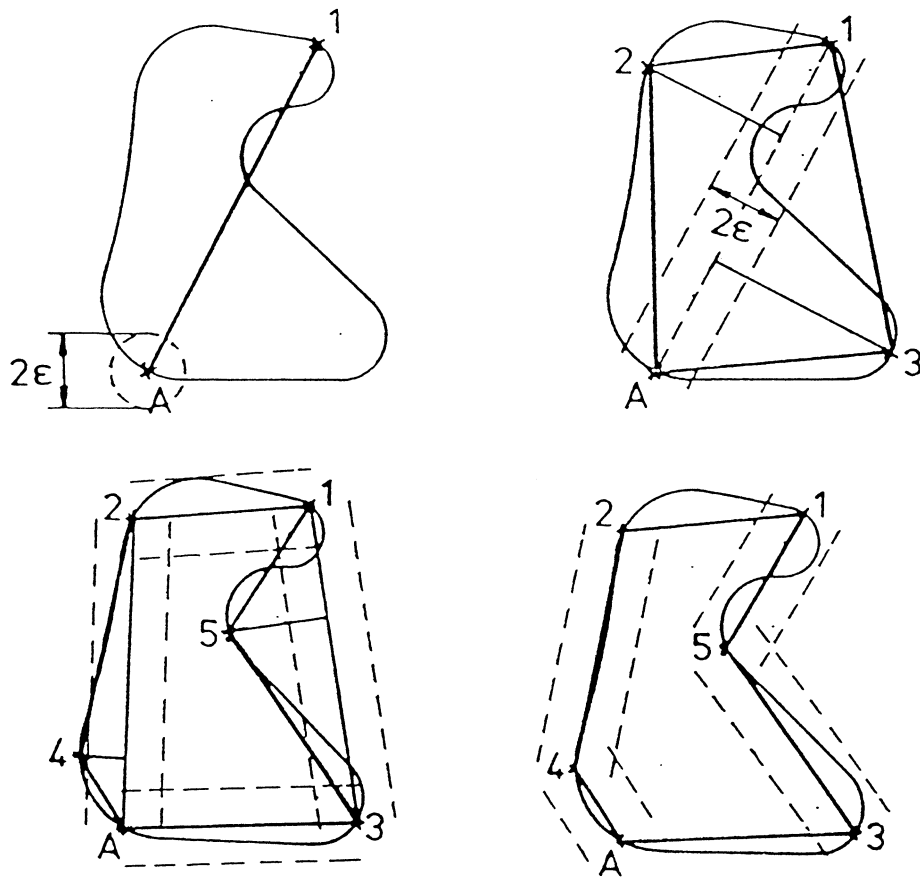
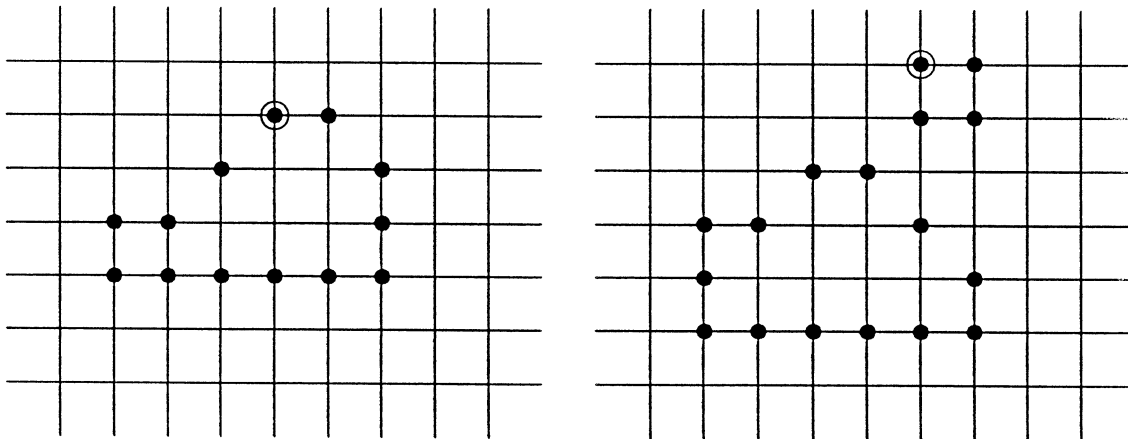


Abbildung 5.5: Polygonapproximation der Kontur [45]



Einige Merkmale des Musters wie Umfang und Fläche lassen sich direkt aus dem Kettencode berechnen. Bedeutsam ist die Kettenkodierung wegen der Datenreduktion von Linienbildern. Da hier aber nicht auf das Gebiet der Kodierung eingegangen werden soll, wird in den anschließenden Kapiteln auf die anschaulichere Polygonapproximation der Konturlinie zur Merkmalsgewinnung der Binärobjekte zurückgegriffen.

5.3 Geometrische Merkmale

Für die Lösung einfacher Bildanalyseprobleme wie die Mustererkennung einzelner Objekte in Binärbildern genügen häufig einfach berechenbare geometrische Merkmale zur Objektbeschreibung. Hierzu gehören beispielsweise die **Fläche** und **Umfang** eines Binärmusters, die in Abschnitt 5.3.2 behandelt werden. Einfache geometrische Merkmale sind **Momente** durch die sich auch die Lage des Objektes bestimmen läßt. Die Momente werden in Abschnitt 5.3.3 besprochen. Die häufig in der industriellen Anwendung benutzte Methode des **Polar-check** wird in Abschnitt 5.3.4 vorgestellt. Da der Praktikumsversuch die lageinvariante Mustererkennung behandelt und die dazugehörige Vorlesung **Mustererkennung** eventuell noch nicht gehört ist, werden die Grundlagen der lageinvarianten Klassifikation im Abschnitt 5.3.1 behandelt, genauso wie sie in der Vorlesung eingeführt werden [8].

5.3.1 Grundlagen der lageinvarianten Mustererkennung

Mustererkennung ist die Theorie der Zuordnung eines unbekanntes Musters X_i zu einer Bedeutungs- oder **Äquivalenzklasse** χ_j . Auf der Menge von Mustern $M = \{X, Y, Z, \dots\}$ ist im allgemeinen eine Äquivalenzrelation \sim mit folgenden drei Eigenschaften definiert :

1. Für jedes $X \in M$ gilt $X \sim X$ (Reflexivität).
2. Aus $X \sim Y$ folgt $Y \sim X$ (Symmetrie).
3. Aus $X \sim Y$ und $Y \sim Z$ folgt $X \sim Z$ (Transitivität).

Für jedes $X \in M$ ist die dazugehörige Äquivalenzklasse χ_x :

$$\chi_x = \{W | W \in M \quad \text{und} \quad X \sim W\}. \quad (5.1)$$

Äquivalenzklassen sind definiert durch alle Repräsentanten von χ oder auch parametrisch durch ein erzeugendes Element X^0 und eine Abbildungsklasse $\Gamma(X)$, die z.B. durch alle Bewegungen in der Ebene gebildet wird.

Bei der lageinvarianten Musterklassifikation muß ein unbekanntes Muster X , welches innerhalb des Gesichtsfeldes der Kamera eine Lagetransformation erfährt, einer Bedeutungsklasse zugeordnet werden. Dabei soll angenommen werden, daß das Muster X durch eine zulässige, jedoch unbekannte Lagetransformation $\gamma_j \in \Gamma$ aus einem der möglichen Referenzmuster X_i^0 erzeugt werden kann.

$$X = \gamma_j(X_i^0) \quad (5.2)$$

Es ist $\gamma_j \in \Gamma$ unbekannt, wobei Γ die Menge der zulässigen Lagetransformationen sei. Als Menge der zulässigen Lagetransformationen des Musters soll im folgenden die Menge der Bewegungen in der Ebene (Translation und Rotation) und die Menge der Skalierung (Vergrößerung/ Verkleinerung) gelten. In (5.2) ist i auch unbekannt mit $i \in 1, \dots, n$, wobei n die Anzahl der (bekannten) Referenzmuster ist.

Sind x, y die Koordinaten des Referenzmusters X^0 , $X^0 = X^0(x, y)$, und x', y' die des transformierten Musters X , wird Γ beschrieben durch

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = R \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_t \\ y_t \end{pmatrix} \quad (5.3)$$

Für den Skalierungsfaktor gilt $R > 0$. Die Translation des Musters wird beschrieben durch x_t, y_t . Der Drehwinkels des Objekts ist ϕ . (5.3) beschreibt die Ähnlichkeitsgruppe.

Die einfachste Möglichkeit der Klassifikation ist der direkte Vergleich des unbekanntes Musters X in allen möglichen Lagen mit jedem Referenzmuster X_i^0 (matching). Ein weniger rechenintensives Verfahren, insbesondere bei der zugelassenen Lagevarianz der Muster, ist das **Graph-matching**, welches in den nächsten beiden Kapiteln besprochen wird.

Eine weitere Möglichkeit zur Musterklassifikation ergibt sich durch die Verwendung von **lageinvarianten Transformationen**, deren einfachste Beispiele hier als geometrische Merkmale behandelt werden. Anstatt ein gemessenes Muster in jeder Lage mit allen Referenzmustern zu vergleichen, wendet man zunächst eine Transformation τ auf das Muster X an, um die von der Ähnlichkeitsgruppe unabhängigen, strukturspezifischen Merkmale zu extrahieren. Das transformierte Muster, bzw. der gewonnene Merkmalsvektor, \tilde{X} ergibt sich also aus

$$\tau : X \rightarrow \tilde{X}. \quad (5.4)$$

Die Merkmale \tilde{X} werden nun mit den Transformaten der Referenzmuster $\{\tilde{X}_i^0\}$ verglichen, wobei häufig eine starke Reduktion des Merkmalsraums auf wenige, dominante Merkmale möglich ist.

Eine unter der Gruppe Γ , der Menge der zulässigen Lagetransformationen, invariante Transformation τ erfüllt notwendigerweise die Bedingung, daß alle Muster, die durch eine Lagetransformation γ_j hervorgehen, in einen Punkt des Merkmalraums abgebildet werden.

$$X = \gamma_j(X_i^0) \Rightarrow \tau(X_i^0) = \tau(X), \quad \forall \gamma_j \in \Gamma \quad (5.5)$$

Wenn gilt zusätzlich gilt, daß aus

$$\tau(X) = \tau(X_i^0) \Rightarrow X = \gamma_j(X_i^0), \quad \gamma_j \in \Gamma \quad (5.6)$$

folgt, ist τ eine **vollständige**, lageinvariante Transformation. Die Vollständigkeit gewährleistet, daß Muster verschiedener Äquivalenzklassen nicht in den selben Punkt im

Merkmalraum abgebildet werden. Die strenge Forderung der vollständigen Separierbarkeit aller möglichen Muster ist häufig jedoch nicht erforderlich, und zwar dann, wenn aus Kenntnis der Muster die Separierbarkeit unter der Transformation gesichert ist.

Nach der Merkmalsextraktion, der Bildung des Merkmalsvektors \tilde{X} , erfolgt die Klassifikation der Muster anhand der lageunabhängigen, strukturspezifischen Merkmale. Als Maß für die Ähnlichkeit zweier Muster X, Y wird eine Metrik $D(\tilde{X}, \tilde{Y})$ definiert. Die Zuordnung eines Testmusters zu einer Bedeutungsklasse geschieht auf der Basis des kleinsten Abstands gemäß der Metrik D , also

$$X_i \sim X_j^0 \quad \text{falls} \quad D(\tilde{X}_i, \tilde{X}_j^0) = \min. \quad (5.7)$$

Dieser einfache Klassifikator heißt **Minimumdistanz-Klassifikator**. Eine Zurückweisungsklasse läßt sich einfach durch einen Schwellwert, der der gewünschte maximale Fehlerabstand zwischen einem Testmuster und seinem dazugehörigen Referenzmuster ist, bilden. Für die Robustheit des Klassifikationsverfahrens ist es wünschenswert, daß die Abbildung der gewählten Transformation τ im Sinne der gewählten Metrik stetig ist, d.h. es gibt zu jedem $\eta > 0$ ein $\varepsilon > 0$, so daß

$$D(X, Y) < \varepsilon \quad \Rightarrow \quad D(\tilde{X}, \tilde{Y}) < \eta. \quad (5.8)$$

5.3.2 Umfang, Fläche und Kompaktheit

Nach der Polygonapproximation liegt die Konturlinie als ein geschlossener Polygonzug mit N Eckpunkten $z(i) = (x(i), y(i)), i = 0, \dots, N$ vor, wobei $z(0) = z(N)$ ist. Der Umfang U und die Fläche F des Objektes lassen sich daraus sehr einfach bestimmen.

$$U = \sum_{i=0}^{N-1} |z(i+1) - z(i)| \quad (5.9)$$

$$F = \frac{1}{2} \sum_{i=0}^{N-1} (x(i+1)y(i) - x(i)y(i+1)) \quad (5.10)$$

Aufgabe 5.3 *Leiten Sie die Gl. (5.10) für die Fläche her. Erinnern Sie sich dabei an das Kreuzprodukt zweier Vektoren. Wie transformieren sich der Umfang U und die Fläche F unter der zugelassenen Lagetransformation in Gl. (5.3)? Wird die notwendige Bedingung der Lageinvarianz (3-5) erfüllt?*

Als Ergebnis der Aufgabe 5.3 wird deutlich, daß der Umfang und die Fläche nicht skalierungsinvariant sind. Es kann allerdings ein größeninvarianter Formfaktor K , der Kompaktheit genannt wird [2], definiert werden durch

$$K = \frac{U^2}{4\pi F} \quad . \quad (5.11)$$

Dieser wird näherungsweise gleich 1 für Kreise und nimmt große Werte für linienhafte Objekte an.

5.3.3 Momente

Eine weitere lageinvariante Lagetransformation ist die Methode der **Momente** [53]. Sei das Objekt im Binärbild durch $f(x,y) = 1$ gekennzeichnet, dann sind die Momente von $f(x,y)$ definiert durch

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y), \quad p, q = 0, 1, 2, \dots \quad (5.12)$$

$(p+q)$ bezeichnet man als Ordnung des Momentes m_{pq} . Mit den beiden Flächenschwerpunkten $\bar{x} = m_{10}/m_{00}$ und $\bar{y} = m_{01}/m_{00}$ sind die **Zentralmomente** gegeben zu

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x,y), \quad p, q = 0, 1, 2, \dots \quad . \quad (5.13)$$

Im Gegensatz zu den in (5.12) gegebenen Momenten sind die Zentralmomente invariant gegenüber Translationen des Bildsignals. Normiert man die Zentralmomente gemäß

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^r}, \quad r = \frac{p+q}{2} + 1 \quad (5.14)$$

mit $p+q = 2, 3, \dots$, so läßt sich ein Satz von Momenten angeben, der invariant ist gegenüber Translationen, Rotationen und Größenveränderungen, also die Bedingung (5.5) erfüllt :

$$\begin{aligned}
 I_1 &= \eta_{20} + \eta_{02} \\
 I_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
 I_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2 \\
 I_4 &= (\eta_{30} - \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 .
 \end{aligned} \tag{5.15}$$

Damit sind neben der Kompaktheit K weitere einfache geometrische Merkmale gegeben, Objekte zu klassifizieren. Außerdem läßt sich die Lage und die Orientierung eines Objektes bezüglich des Referenzmusters durch Momente angeben. Der Schwerpunkt des Objektes z_s wird durch $z_s = (\bar{x}, \bar{y})$ bestimmt, die Orientierung θ ergibt sich aus

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2m_{11}}{m_{20} - m_{02}} \right) . \tag{5.16}$$

Aufgabe 5.4 Erfüllen die Invarianten I_1, \dots, I_4 die Bedingung der Vollständigkeit einer lageinvarianten Transformation (5.6) ?

5.3.4 Polar-check

Die Methode des Polar-check zur Mustererkennung ist ein in der industriellen Anwendung häufig benutztes Verfahren [16]. Es besteht darin, einen oder mehrere Kreise mit bestimmten Radien um den Schwerpunkt des Objekts zu ziehen und die Schnittpunkte der Kreise mit der Kontur zu bestimmen (s. Abb. 5.6). Je nach Anforderungsgrad an die Klassifikationsfähigkeit lassen sich Merkmale auf zwei Weisen finden:

1. als Anzahl der Schnittpunkte zu den jeweiligen Radien,
2. als Folge von Winkeldifferenzen, die man erhält, wenn man die Schnittpunkte mit dem Schwerpunkt des Objekts verbindet.

Beim 1. Verfahren wird die Anzahl der Schnittpunkte mit denen der Referenzmuster verglichen. Beim 2. Verfahren wird die maximale Korrelation der Winkeldifferenzfolge des Testobjektes mit denen der Referenzobjekte gesucht. Sind $f_i, i = 1, \dots, n$ die Listenelemente des Testmusters, $g_i, i = 1, \dots, n$ die des Referenzmusters, so wird das Testobjekt der Referenzklasse zugeordnet, für die

$$\frac{\sum_{i=1}^n f_i g_{i+\tau}}{\sqrt{\sum_{i=1}^n f_i^2} \sqrt{\sum_{i=1}^n g_i^2}} \quad (5.17)$$

bei einer zyklischen Permutation τ maximal ist.

Aufgabe 5.5 *Ist bei beiden Verfahren des Polar-check die notwendige Bedingung oder die Vollständigkeit der lageinvarianten Transformation erfüllt ?*

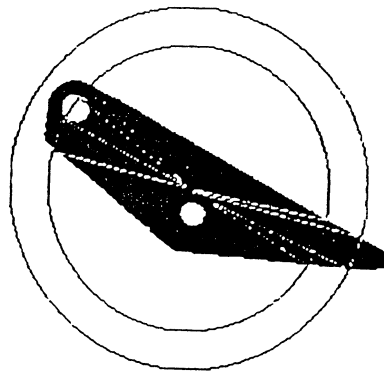


Abbildung 5.6: Polar-check [16]

5.4 Graph-matching

In Abschnitt 5.4.1 werden die Grundlagen der Darstellung eines Binärmusters als Graph besprochen [2, 35, 38]. Der zu definierende **Attributgraph** soll das Muster lageinvariant beschreiben. Die Klassifizierungsaufgabe, zu dem Testgraphen den identischen Referenzgraphen zu finden, ist der **Graph- Isomorphismus**. Die Definition wird in Abschnitt 5.4.2 gegeben. In Abschnitt 5.4.3 wird die Praxis des Graph- matching beschrieben. Die durch die Aufnahmeeinheit verursachten Fehler machen eine Vergleichsmetrik notwendig, da eine 1:1 Abbildung nur näherungsweise gefunden werden kann. Außerdem ist der Einfluß von Bildvorverarbeitungsschritten, hier die Kantenapproximation, zu beachten.

5.4.1 Grundlagen der Darstellung eines Musters als Graph

Definition 5.1 Ein gerichteter Graph $G = (V, E)$ besteht aus der Menge $V = \{1, 2, \dots, |V|\}$, deren Elemente die Ecken (vertex, node) von G heißen, und einer Teilmenge $E \subseteq V \times V$, deren Elemente Kanten (edge) von G genannt werden. Ein Paar $(v, w) \in E$ ist eine Kante von v zu w . Kanten der Form (v, v) heißen Schlingen.

Für einen ungerichteten Graphen, hier immer Graph genannt, gilt: $(v, w) \in E \Leftrightarrow (w, v) \in E$. Im folgenden sei $n = |V|$ und $e = |E|$. Die Abbildung 5.7 zeigt ein Beispiel eines gerichtet Graphen.

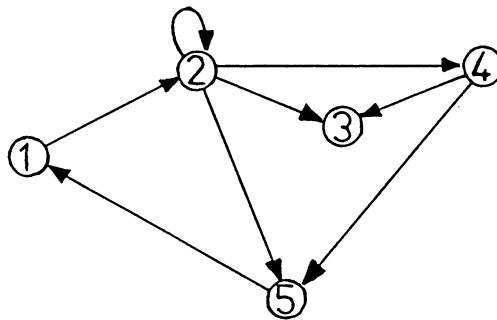


Abbildung 5.7: Gerichteter Graph [35]

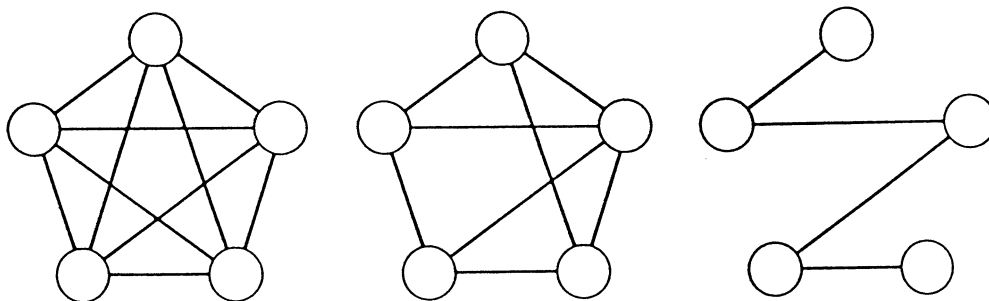


Abbildung 5.8: Graph : (a) 4-fach (b) 3-fach (c) 1-fach-zusammenhängend [2]

Zwei Methoden, einen Graphen zu speichern, sind üblich:

1. Adjazenzmatrix : $G = (V, E)$ wird repräsentiert durch eine $n \times n$ Matrix $A = (a_{ij})$ mit

$$a_{ij} = \begin{cases} 1, & (i, j) \in E \\ 0, & (i, j) \notin E \end{cases} \quad (5.18)$$

2. Liste : Die Speicherung von $G = (V, E)$ erfolgt in folgender Form :

$$n, e, i_1, k_1, i_2, k_2, \dots, i_e, k_e \quad (5.19)$$

Der Index der Anfangsecke des j -ten Pfeils ist i_j , der Index der Endecke ist k_j .

Die Gl. (5.20) zeigt die Adjazenzmatrix des Beispiels. Ein Graph heißt r -fachzusammenhängend, wenn je zwei Ecken durch mindestens r kreuzungsfreie Wege verbunden sind. Wege zwischen v und w sind kreuzungsfrei, wenn sie außer v und w keine gemeinsamen Ecken haben. Die Abbildung 5.8 zeigt einige Beispiele.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.20)$$

Aufgabe 5.6 *Wie sieht die Adjazenzmatrix des in Abbildung 5.7 gezeigten Graphen aus, falls die Kanten ungerichtet sind.*

Bisher sind den Kanten eines Graphen nur zweiwertig repräsentiert, also nur durch wahr oder falsch, ob eine Kante zwischen den Ecken existiert. Eine Verallgemeinerung ist möglich, indem man den Kanten Attribute zuordnet. Diese können z.B. zur Darstellung einer komplexen Szene Relationen wie UNTER, ÜBER, RECHTS-VON usw. oder geometrische Merkmale sein. Auch den Knoten lassen sich über die Schlingen Attribute zuordnen. Ein solcher Graph wird ein **bewerteter Graph** oder **Attributgraph** genannt [19]. Mit ihm lassen sich beliebige relationale Strukturen darstellen. Im folgenden soll der Attributgraph $G_A = (V, E, M)$ definiert sein durch:

Definition 5.2 *Sei M eine nichtleere Menge, $G = (V, E)$ ein Graph und eine Abbildung von G in M ; $\gamma : G \rightarrow M$. Dann ist $G_A = (V, E, M)$ ein M -bewerteter Graph.*

Besteht über den Wertebereich M kein Zweifel, so schreiben wir auch kurz $G_A = (V, E,)$. Ähnlich wie ein unbewerteter Graph läßt sich ein Attributgraph als Matrix oder Liste darstellen und speichern. Ziel ist es nun, die Elemente der Mengen V, E und die Abbildung so zu wählen, daß sich ein Binärmuster eindeutig beschreiben läßt und die Freiheitsgrade bezüglich der Lagetransformation eingeschränkt sind.

Aufgabe 5.7 Versuchen Sie, ein Rechteck mit Hilfe eines Attributgraphen zu beschreiben.

5.4.2 Graph-Isomorphismus

Durch die Repräsentation eines Testmusters als Attributgraph ist es zur Klassifikation nur noch notwendig, den Graphen des Testmusters mit denen der Referenzmuster zu vergleichen und nicht das Muster selbst. Ist ein Graph mit einem anderen identisch, gibt es eine bijektive Abbildung zwischen den Ecken, Kanten und Attributen der beiden Graphen, ein **Graph-Isomorphismus**

Definition 5.3 Ein Isomorphismus des Graphen $G_A = (V, E, , M)$ auf den Graphen $G'_A = (V', E', ', M)$ ist eine bijektive Abbildung $f: V \rightarrow V'$, so daß gilt:

$$\begin{aligned}(v_1, v_2) \in E &\iff (f(v_1), f(v_2)) \in E' && , v_1, v_2 \in V \\ (v) \in M &\iff (f(v)) \in M && , v \in V \cup E\end{aligned}$$

D.h., G ist isomorph zu G' genau dann, wenn es eine 1:1 Korrespondenz zwischen der Menge der Knoten von G und G' gibt, und die Verbindungen der Knoten mit den dazugehörigen Attributen erhalten werden.

Algorithmen zum Graph-Isomorphismus sind NP-Algorithmen, das sind *nichtdeterministische* polynomial lösbare Algorithmen. D.h. nur nichtdeterministische Algorithmen lösen das Problem in polynomialer Zeit. Deterministische Algorithmen zum Graph-Isomorphismus besitzen einen *exponentiellen* Aufwand. Manche Spezialfälle lassen sich aber mit einem P-Algorithmus, einem deterministischem Algorithmus mit *polynomialer* Zeit, lösen. Sind die Graphen planar, so kann das Problem des Graph-Isomorphismus mit einem P-Algorithmus gelöst werden. Ein Graph heißt planar, wenn er sich kreuzungsfrei in der Zeichenebene darstellen läßt [2, 38].

5.4.3 Attributierung

An dieser Stelle wird eine Darstellung eines einfach zusammenhängenden Binärmusters als Attributgraph vorgestellt, die im Praktikumsversuch realisiert ist. Der Graph ist unter den zugelassenen Lagetransformationen *teilinvariant* so daß sich die Anzahl der Freiheitsgrade reduziert. Im vorliegenden Fall ist der Attributgraph invariant unter der Gruppe der Translation und Skalierung, aber variant unter der Gruppe der Rotation. Diese Varianz kommt in der Abhängigkeit des Graphen von dem Aufpunkt des Konturverfolgers und damit in der Indizierung des Polygonzugs zum Ausdruck. Der Attributgraph ist dadurch variant unter der Gruppe der zyklischen Permutationen der Knoten, was bei der Auswertung berücksichtigt werden muß.

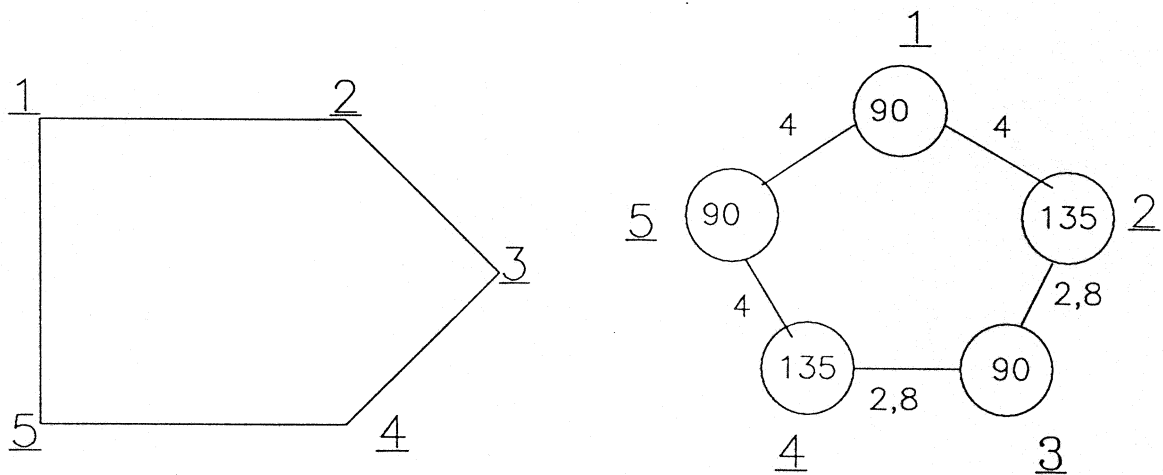


Abbildung 5.9: (a) Polygonzug (b) Attributgraph

Ausgangspunkt ist der Polygonzug mit den Eckpunkten $z_i = (x_i, y_i)$, $i = 0, 1, 2, \dots, N$ mit $z_0 = z_N$, durch den die Objektkontur approximiert ist. Die Umlaufrichtung ist entgegengesetzt dem Uhrzeigersinn, so wie der Algorithmus *Polapprox* arbeitet. Aus Abbildung 5.9 wird deutlich, daß der Polygonzug topologisch einer Darstellung als Graph entspricht, faßt man die Eckpunkte als Knoten auf. Die Folge der Eckpunkte läßt sich darstellen als 2-fachzusammenhängender Graph mit Ringstruktur. So, wie ein Eckpunkt des Polygonzugs z_i nur mit den nächsten Nachbarn z_{i+1}, z_{i-1} , verbunden ist, ist der dazugehörige Vertex v_j nur mit den jeweiligen Knoten v_{j+1}, v_{j-1} , verbunden. Die Struktur eines Polygonzugs bleibt also erhalten.

Die zu den Knoten gehörige Attributmenge $|V$ seien die Winkel ϕ_i an den Eckpunkten z_i . Es ist $\phi_0 = \phi_N$. Werden die Kanten zwischen zwei benachbarten Vertices mit den Längen der Geradenstücke $t_{i,i+1}$, zwischen den abgebildeten Eckpunkten attribuiert, erhält man eine Graphdarstellung des Musters. Normiert man die Längen $t_{i,i+1}$, mit dem dem Umfang U , so wird die relationale Struktur des Musters lage- und größeninvariant abgebildet, wohingegen sich die Lagevarianz des Musters gegenüber dem Aufpunkt des Konturverfolgers als Varianz der Knoten unter der Gruppe der zyklischen Permutationen ausdrückt. Die Abbildung der relationalen Struktur des Musters auf den Attributgraphen $G_A = (V, E, R)$ mit $\phi_i = (z_i), t_{i,i+1} = (z_i, z_{i+1})$ ist eindeutig, da jede Deformation des Musters auf eine andere Attributierung führt. Deformationen sind alle möglichen Veränderungen des Musters bis auf die zugelassenen Lage- und Größentransformationen. In Abbildung 5.9 ist ein Muster und der dazugehörige Attributgraph gezeigt. Genau genommen liegt durch den Umlaufsinn der Kontur ein gerichteter Graph vor. Durch den Konturfindalgorithmus ist die Umlaufrichtung festgelegt, so daß sie nicht beachtet werden muß.

Bei der Definition 4.3 eines Graph-Isomorphismus wird davon ausgegangen, daß die Identität der Attribute geprüft wird. In der Praxis ist dies nicht sinnvoll, da sich aufgrund von Fehlern, die von der Aufnahmeeinheit oder äußeren Störeinflüssen herrühren, im allgemeinen die Attributgraphen G_A, G'_A desselben Musters in zwei verschiedenen Aufnahmen unterscheiden. Ein Vergleich der Attribute soll nicht nur bei Identität wahr sein, sondern auch bei Unterschreiten einer Fehlertoleranzgrenze δ im Sinne einer zu definierenden Metrik $D(\phi, \phi')$ und $D(t, t')$.

Weiterhin ist zu beachten, daß die Polygonapproximation nur wirklich markante Punkte der Kontur berechnet, da kleine Fehler in der Kontur nicht zu fehlenden oder zusätzlichen Eckpunkten führen dürfen, da die Graph-Isomorphie als erste Klassifikationsbedingung die gleiche Anzahl von Knoten beim Test- und Referenzgraph fordert. Das Genauigkeitsintervall ε der Polygonapproximation (s. Abschnitt 5.2.2) darf daher nicht zu klein (wegen Rauschen) oder zu groß (wegen fehlender Punkte) gewählt werden (s. Abbildung 5.10), was die Klassifikationsfähigkeit des Graph-matching einschränkt. Größere Störungen der Objektkontur oder überlappende Muster führen folglich zu Fehlklassifikationen, die die abgewandelte Methode des Subgraphmatching erforderlich machen.

Aufgabe 5.8 Welche Objekte lassen sich aufgrund der Polygonapproximation schlecht durch dieses Verfahren klassifizieren ?

Ein Binärobjekt wird durch die dargestellte Attributierung als folgende Liste dargestellt

:

$$n, \phi_0, t_{0,1}, \phi_1, t_{1,2}, \dots, \phi_{n-1}, t_{n-1,n} \quad . \quad (5.21)$$

Die notwendige Bedingung für einen Isomorphismus zwischen dem Graphen G_A des Referenzmusters O_{ref} und dem Graphen G'_A des Testmusters O_{test} ist

$$n = n'. \quad (5.22)$$

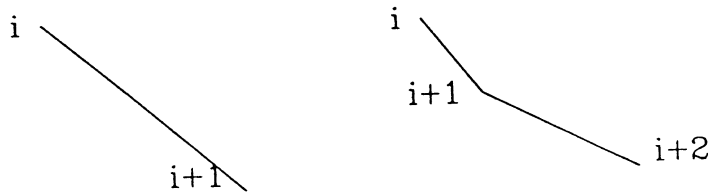


Abbildung 5.10: Fehlermöglichkeit

Ist Gl. (5.22) erfüllt, werden die übrigen Listenelemente des Testmusters mit denen der Referenzmuster verglichen. Da der Startpunkt des Konturfindalgorithmus unbekannt ist, was die Varianz der Transformation des Musters auf den Attributgraphen unter der Gruppe der zyklischen Permutationen der Knoten zur Folge hat, muß der Vergleich mit allen zyklischen Permutationen der Listenpaare (ϕ, t) von O_{test} stattfinden. Stimmen sämtliche Listenelemente innerhalb eines Fehlerintervalls δ_ϕ bzw. δ_t überein, wird das Testobjekt der Klasse des Referenzobjektes zugeordnet.

Aufgabe 5.9 Welche Ordnung hat der aufgeführte Algorithmus? Was ist der Grund für die Reduktion der Zeitkomplexität gegenüber einem gewöhnlichen NP- Algorithmus zur Lösung eines Graph-Isomorphismus?

5.5 Subgraph-matching

Das im vorigen Kapitel behandelte Verfahren des Graph-matching versagt bei rundlich geformten Konturen oder bei deformierten Objekten, bei denen z.B. ein Teil fehlt oder

überdeckt wird, da der Graph- Isomorphismus eine 1:1 Korrespondenz zwischen Test- und Referenzgraph fordert. Für diese Fälle ist ein verfeinertes Matching-Verfahren, das **Subgraph-matching**, notwendig, bei dem Teile des Testgraphen mit den Referenzgraphen verglichen werden und die größte Übereinstimmung (best match) gesucht wird. In Abschnitt 5.5.1 wird der dazu benötigte **Assoziationsgraph** definiert, in dem die maximale **Clique** gesucht wird [2]. Als Attribute zum Aufbau des Assoziationsgraphen werden nicht die in Kapitel 5.4 genannten verwendet, sondern Parameter der Koordinatentransformation von Test- zu Referenzobjekt [13]. Die Parameter werden in Abschnitt 5.5.2 hergeleitet. Der Algorithmus zum Subgraph-matching wird in Abschnitt 5.5.3 behandelt.

5.5.1 Subgraph-Isomorphismus und Assoziationsgraph

Die Abbildung 5.11 illustriert zwei Beispiele, in denen ein Objekt deformiert oder teilweise überlappt ist. Wie sofort ersichtlich ist, kann kein Graph-Isomorphismus zwischen Referenz- und Testgraph existieren, falls der Graph wie in Kapitel 5.4 aus dem Binärbild extrahiert wird. Statt dessen wird versucht, Teile des Referenzgraphen Teilen des Testgraphen zuzuordnen, um so die größte Übereinstimmung festzustellen. Das Problem ist das Finden eines "doppelten" Subgraph-Isomorphismus, das sich durch Bildung eines Assoziationsgraphen mit einem einfachen Subgraph-Isomorphismus-Algorithmus lösen läßt.

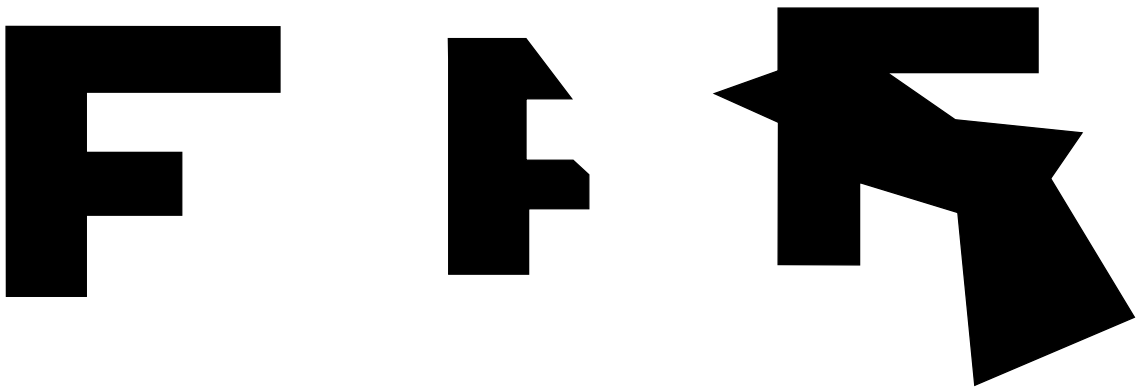


Abbildung 5.11: (a) Referenzmuster (b) deformiertes und (c) überlapptes Testmuster

Ein Teilgraph (subgraph) ist von einem Untergraph (induced subgraph) zu unterschei-

den. Die Definitionen lauten folgendermaßen :

Definition 5.4 G, G' seien Graphen. $G' = (V', E')$ heißt *Teilgraph (subgraph)* von $G = (V, E)$, wenn $V' \subseteq V$ und $E' \subseteq E \cap V' \times V'$ gilt.

Definition 5.5 G' heißt *Untergraph (induced subgraph)* von G , wenn $V' \subseteq V$ gilt und die Menge der Kanten $E' = E \cap V' \times V'$ ist.

Das Problem des Subgraph-Isomorphismus ist, einen Isomorphismus zwischen einem Graphen G und einem Teilgraphen des Graphen G' zu finden.

Definition 5.6 G'_S sei Subgraph von G' . Ein *Subgraph-Isomorphismus* des Graphen $G = (V, E, M)$ auf den Graphen $G'_S = (V'_S, E'_S, M)$ ist eine bijektive Abbildung $f: V \rightarrow V'_S$, so daß gilt :

$$\begin{aligned} (v_1, v_2) \in E &\iff (f(v_1), f(v_2)) \in E'_S && , v_1, v_2 \in V \\ (v) \in M &\iff (f(v)) \in M && , v \in V \cup E \quad . \end{aligned}$$

Aus Abbildung 5.11 ist ersichtlich, daß die Forderung nach einem Subgraph-Isomorphismus zwischen Test- und Referenzgraph nicht allgemein genug ist, um das Testmuster einer Referenzklasse zuzuordnen, da jeweils Teile des Testobjektes nur mit Teilen des Referenzobjektes übereinstimmen. Das Problem ist also, einen Isomorphismus zu finden zwischen Teilgraphen vom Testobjekt O_{test} und Teilgraphen von O_{ref} , und zwar den, der die größte Übereinstimmung (best match) liefert. Der zu definierende Assoziationsgraph [2], auch Korrespondenzgraph genannt, ist das geeignete Instrument, die zwei Graphen miteinander zu vergleichen.

Definition 5.7 $G = (V, E, M)$ und $G' = (V', E', M')$ seien *Attributgraphen*. Der *Assoziationsgraph* $A_{G,G'} = (V_A, E_A, M_A)$ wird folgendermaßen konstruiert. Für jedes $v \in V$ und $v' \in V'$, bilde einen Knoten $v_A \in V_A$ mit der Kennzeichnung (v, v') genau dann, wenn $(v) = (v')$ ist. Bilde eine Kante $e_A \in E_A$ zwischen zwei Knoten $v_A^1 = (v_i, v'_j)^1$ und $v_A^2 = (v_k, v'_l)^2$ des Assoziationsgraphen genau dann, wenn $(v_i, v_k) = (v'_j, v'_l)$ gilt. A sei eine Abbildung $A: (V_A, E_A) \rightarrow M_A$.

Ein Knoten in $A_{G,G'}$ wird gebildet, wenn ein Knoten von G das identische Attribut wie ein Knoten von G' hat. Der Knoten des Assoziationsgraphen drückt also eine Übereinstimmung von G und G' aus, die aber nicht auf die relationale Struktur der Graphen eingeht. Stimmt ein Kantenattribut von G mit einem von G' überein *und* bilden die Anfangs- und Endpunkte der beiden Kanten von G, G' jeweils einen Knoten im Assoziationsgraphen, haben also identische Attribute, so wird eine Kante in $A_{G,G'}$ gebildet. Kanten von $A_{G,G'}$ drücken aus, daß außer den Knoten auch relationale Strukturen beider Graphen identisch sind. Die Abbildung 5.12 zeigt ein Beispiel [2] :

Zwei Muster O, O' seien durch "primitive Elemente "(Kreis \circ , Quadrat \square) dargestellt, die als Knoten in den dazugehörigen Graphen G, G' erscheinen. Die Kanten zwischen zwei Elementen beschreiben die relationale Struktur, die Attribute sind also $\circ - \square$, $\circ - \circ$ oder $\square - \square$. Ein Knoten des Assoziationsgraphen $A_{G,G'}$ korrespondiert zu einem Paar von Knoten von G, G' , deren Eigenschaften gleich sind. Es werden jeweils die Eckpunkte von $A_{G,G'}$ verbunden, die die gleiche relationale Struktur von G und G' ausdrücken.

Gibt es einen Isomorphismus zwischen einem Teilgraphen G_S von G und einem Teilgraphen G'_S von G' , so wird dieser durch eine Menge von vollständig verbundenen Knoten im Assoziationsgraphen repräsentiert. Vollständig verbunden heißt, jeder Knoten ist mit jedem anderen Knoten verbunden.

Definition 5.8 *Eine Clique der Größe N ist ein vollständig verbundener Teilgraph der Größe N .*

Die Abbildung 5.8(a) zeigt eine 5-Clique. Eine Clique der Größe N ist $N - 1$ - fach zusammenhängend. Am Beispiel der Abbildung 5.12 sieht man, daß ein Isomorphismus zwischen einem Subgraph von (a) und einem Subgraph von (b), also eine 1:1 Korrespondenz zwischen Teilen der Muster, eine maximale Clique im Assoziationsgraphen bildet. Eine Clique ist maximal, wenn ein Hinzufügen eines Knoten die Eigenschaft der vollständige Verbundenheit zerstört. Um die Klassifikationsaufgabe, ein deformiertes oder überlapptes Testmuster bestmöglich einer Referenzklasse zuzuordnen, zu erfüllen, muß die größte maximale Clique, d.h. die maximale Clique mit größtem N , in den Assoziationsgraphen, die vom Testmuster mit allen Referenzmustern gebildet werden, gefunden werden.

5.5.2 Koordinatentransformation

Wie in Kapitel 5.4 beim Graph-matching ist es wichtig, geeignete Attribute zur Beschreibung der Test- und Referenzmuster zu finden, um mit einem Assoziationsgraphen die in Abbildung 5.11 dargestellte Aufgabe zu lösen. Die Innenwinkel und die Kantenlängen des Polygonzugs lassen sich nicht wie in Abschnitt 5.4.3 verwenden, da die Strecken zwischen zwei Punkten nicht einheitlich über den Gesamtumfang des Musters normierbar sind, da sich der Umfang wegen einer Deformation oder Überlappung ändert. Daher sind andere Attribute zur Bildung eines Assoziationsgraphen zu verwenden, die die Klassifikationsaufgabe von deformierten oder überlappten Mustern unter der zugelassenen Lagetransformation 3.3 ermöglicht. Angenommen, zwei Punkte (x_1, y_1) und (x_2, y_2) des Referenzobjektes korrespondieren mit (x'_1, y'_1) und (x'_2, y'_2) des Testmusters. Dann gilt gemäß Gl. 5.3

$$\begin{aligned} \begin{pmatrix} x'_1 \\ y'_1 \end{pmatrix} &= R \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} x_t \\ y_t \end{pmatrix}, \\ \begin{pmatrix} x'_2 \\ y'_2 \end{pmatrix} &= R \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + \begin{pmatrix} x_t \\ y_t \end{pmatrix}. \end{aligned} \quad (5.23)$$

Die Parameter $TP = (\phi, R, x_t, y_t)$ der Koordinatentransformation ergeben sich aus Gl. 5.23 zu

$$R = \sqrt{\frac{(x'_1 - x'_2)^2 + (y'_1 - y'_2)^2}{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \quad (5.24)$$

$$\tan \phi = \frac{(y'_1 - y'_2)(x_1 - x_2) - (y_1 - y_2)(x'_1 - x'_2)}{(x'_1 - x'_2)(x_1 - x_2) + (y_1 - y_2)(y'_1 - y'_2)} \quad (5.25)$$

$$x_t = \frac{1}{2}(x'_1 + x'_2 + R(y_1 + y_2) \sin \phi - R(x_1 + x_2) \cos \phi) \quad (5.26)$$

$$y_t = \frac{1}{2}(y'_1 + y'_2 - R(y_1 + y_2) \cos \phi - R(x_1 + x_2) \sin \phi) \quad (5.27)$$

Die Verwendung der Parameter zur Bildung eines Assoziationsgraphen $A_{G,G'}$ wird in Abbildung 5.13 deutlich. Gegeben seien die Eckpunkte $[z_{i-1}, z_i, z_{i+1}]$ des einen (Referenz-) Musters und $[z'_{j-1}, z'_j, z'_{j+1}]$ des (Test-) Musters ($z = (x, y)$). Bei einer Korrespondenz zwischen z_{i-1}, z_i und z'_{j-1}, z'_j ergeben sich $TP_{ij}^- = (R_{ij}^-, \phi_{ij}^-, x_{t,ij}^-, y_{t,ij}^-)$ als Transformationsparameter, bei einer Korrespondenz zwischen z_{i+1}, z_i und z'_{j+1}, z'_j ergeben sich

$TP_{ij}^+ = (R_{ij}^+, \phi_{ij}^+, x_{t,ij}^+, y_{t,ij}^+)$. Soll nun die Ecke um z_i , die mit $[z_{i-1}, z_i, z_{i+1}]$ bezeichnet wurde, ein möglicher Matchkandidat für die Ecke um z'_j sein, so ist die notwendige Bedingung dafür die Gleichheit der Parameter :

$$R_{ij}^- = R_{ij}^+, \quad \phi_{ij}^- = \phi_{ij}^+, \quad x_{t,ij}^- = x_{t,ij}^+, \quad y_{t,ij}^- = y_{t,ij}^+ \quad . \quad (5.28)$$

Ist Gl. (5.28) erfüllt, so wird ein Knoten mit der Kennzeichnung (v_i, v'_j) im Assoziationsgraphen gebildet. Als Attribut erhält er die Parameter der Koordinatentransformation der korrespondierenden Polygonzugecken z_i und z'_j , also TP_{ij} . Werden alle Ecken des Testmusters mit allen Ecken eines Referenzmusters verglichen, so wird nach obiger Konstruktionsregel ein Assoziationsgraph erzeugt, in dem die Knoten übereinstimmende Ecken von G und G' darstellen. Sämtliche Knoten mit übereinstimmenden Parametern werden verbunden, so daß eine Clique gebildet wird. Die Größe der Clique sagt aus, wie viele Ecken des Testmusters mit Ecken des Referenzmusters matchen, d.h. unter der jeweiligen Lagetransformation übereinstimmen. Ein Testmuster wird der Referenzklasse zugeordnet, dessen Referenzmuster mit dem Testmuster den Assoziationsgraphen mit der größten maximalen Clique bildet.

5.5.3 Berechnung

In einem unbewerteten Graph ist das Suchen einer Clique ein NP- vollständiges Problem, d.h. ein deterministischer Algorithmus löst das Problem mit exponentiellem Aufwand. Im vorliegenden Fall vereinfacht sich der Aufwand durch die Attributierung der Knoten des Assoziationsgraphen mit den Parametern der Koordinatentransformation TP_{ij} .

Besitzt das Testobjekt n Polygonpunkte, und wird es mit einem Referenzobjekt mit m Punkten verglichen, so werden $n \times m$ Möglichkeiten für Matchkandidaten durchgerechnet. Ist m ungefähr gleich n , so ist der Aufwand für den Algorithmus quadratisch in n , $O(n^2)$. Die jeweiligen Ecken, die zueinander matchen, bilden den Assoziationsgraph, der sich als Liste ab speichern läßt. Seien z_i ; $i = 1, \dots, N$ die Polygoneckpunkte des einen Musters z'_j ; $j = 1, \dots, M$ die des zu vergleichenden Musters, und gibt es p richtige Eckenzuordnungen, dann repräsentiert folgende Liste den Assoziationsgraph :

$$\begin{aligned}
z_{i_1}, z'_{j_1} & ; R_{i_1, j_1}, \phi_{i_1, j_1}, x_{t_{i_1, j_1}}, y_{t_{i_1, j_1}} \\
z_{i_2}, z'_{j_2} & ; R_{i_2, j_2}, \phi_{i_2, j_2}, x_{t_{i_2, j_2}}, y_{t_{i_2, j_2}} \\
& \dots \\
z_{i_p}, z'_{j_p} & ; R_{i_p, j_p}, \phi_{i_p, j_p}, x_{t_{i_p, j_p}}, y_{t_{i_p, j_p}}
\end{aligned} \tag{5.29}$$

(z_i, z'_j) repräsentieren die korrespondierenden Ecken und R, ϕ, x_t, y_t sind die Parameter der dazugehörigen Koordinatentransformation. Eine Clique wird von den Eckpaaren (z_i, z'_j) gebildet, die die gleichen Transformationsparameter haben. Die Liste 5.7 muß also nach gleichen Parametersätzen abgesucht werden, was im schlechtesten Fall einen Aufwand von $(p-1)p/2$ erfordert. Da im Extremfall p gleich n ist, hat der Algorithmus zum Suchen auch einen quadratischen Aufwand in n . Somit hat der hier vorgestellte Algorithmus zum Subgraph-matching insgesamt die Ordnung $O(n^2)$.

Der Match mit der größten Anzahl an identischen Parametersätzen liefert die Zuordnung des Testmusters zu einer Referenzklasse. Wegen der in Abschnitt 5.4.3 erwähnten Fehlerquellen muß natürlich dabei ein Fehlertoleranzintervall berücksichtigt werden.

Aufgabe 5.10 *Machen Sie sich die Bildung des Assoziationsgraphen an einem Beispiel klar. Überlegen Sie, was das Verfahren kann und welche Nachteile es hat.*

5.6 Aufgaben im Praktikum

Aufgabe 5.11 *Binarisieren Sie eines der bereitgestellten Bilder mit der Funktion `binarisierung`. Wählen Sie ein Bild mit mehreren dunklen Objekten auf weißem Hintergrund. Zeigen Sie das Original und das binarisierte Bild mit der Funktion `view` an. Extrahieren Sie die Konturen mit der Funktion `konturen` und zeigen Sie diese mit der Funktion `zeige Konturen` an.*

Aufgabe 5.12 *Testen Sie den Algorithmus `polapprox` zur Polygonapproximation mit verschiedenen Fehlern ϵ . Bestimmen Sie ϵ so, daß brauchbare Ergebnisse geliefert werden. Dazu wird mit `unpack` eine einzelne Kontur aus dem von `konturen` zurückgelieferten Array extrahiert und der Funktion `polapprox` übergeben. Zum Anzeigen der Approximation gegenüber dem Original kann die Funktion `zeige Konturen` verwendet werden.*

Aufgabe 5.13 Schreiben Sie eine Funktion `umf`, die den Umfang eines Polygonzugs berechnet.

Aufgabe 5.14 Schreiben Sie eine Funktion `flae`, die die Fläche eines Polygonzugs berechnet.

Aufgabe 5.15 Schreiben Sie eine Funktion `kompaktheit`, die die Kompaktheit eines Polygonzugs berechnet.

Aufgabe 5.16 Bestimmen Sie die Kompaktheit K von vier Referenzobjekten. Anschließend klassifizieren Sie unbekannte Muster mit Hilfe eines Minimumdistanz-Klassifikators `klasskom`.

Aufgabe 5.17 Schreiben Sie eine Funktion `schwer`, die den Schwerpunkt eines Binärbildes berechnet.

Aufgabe 5.18 Erlernen Sie die Momenten-Invarianten I_1, \dots, I_4 aus den Binärbildern von vier Mustern durch die Funktion `moment`. Anschließend klassifizieren Sie unbekannte Muster mit Hilfe eines Minimumdistanz-Klassifikators `klassmom`.

Aufgabe 5.19 Extrahieren Sie die Konturen mehrere Objekte und vereinfachen Sie diese (`konturen`, `polapprox`). Zeigen Sie die Knoten mit den dazugehörigen Attributen, also den Winkeln und den Bogenlängen, als Liste auf (`graphlist`).

Aufgabe 5.20 Verwenden Sie vier verschiedene Objekte als Referenzmuster und klassifizieren Sie überlappte oder deformierte Testmuster mit Hilfe des Programms `subgraph`.

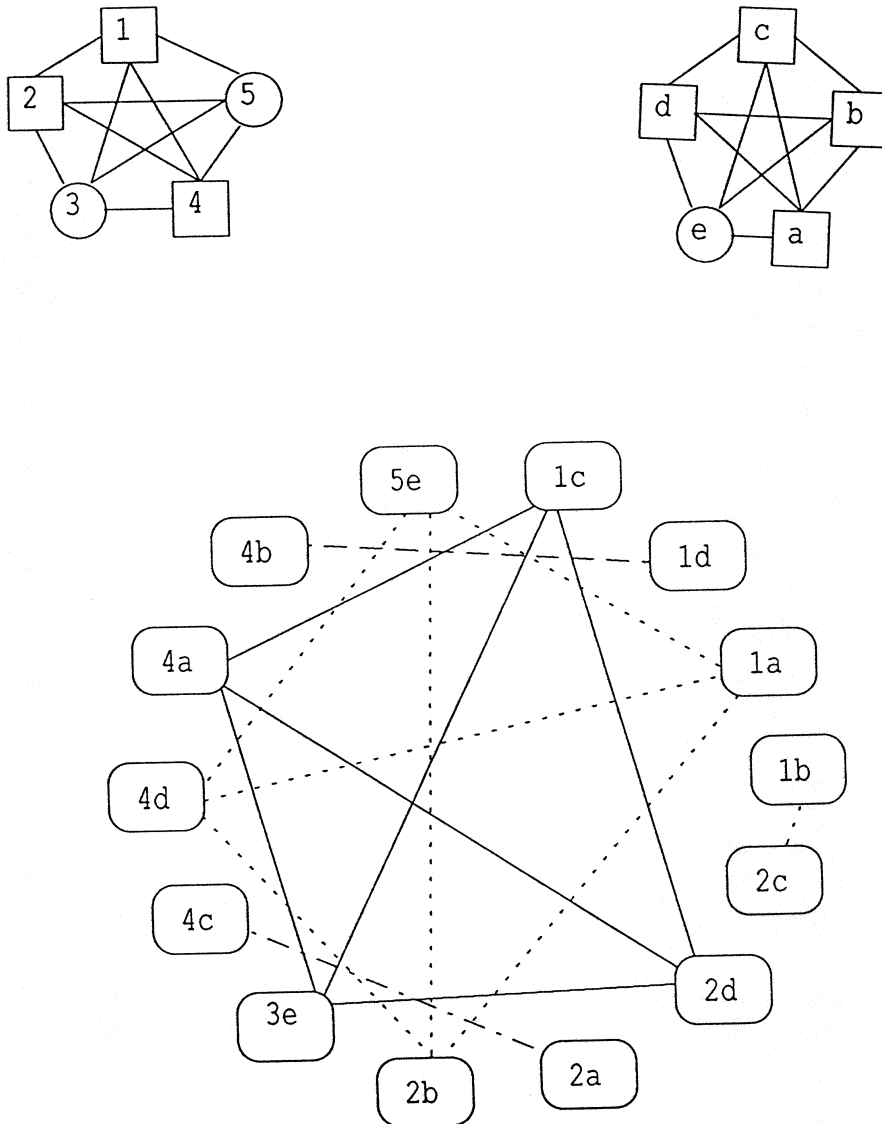


Abbildung 5.12: Beispiel zum Assoziationsgraph. Zwei Graphen (a) und (b) mit \square oder \circ als Knotenmerkmale und $\square - \square$, $\square - \circ$ oder $\circ - \circ$ als Kantenmerkmale sollen verglichen werden. Den Assoziationsgraph zeigt (c). Die verschiedenen Cliques sind durch Strichmarkierungen hervorgehoben.

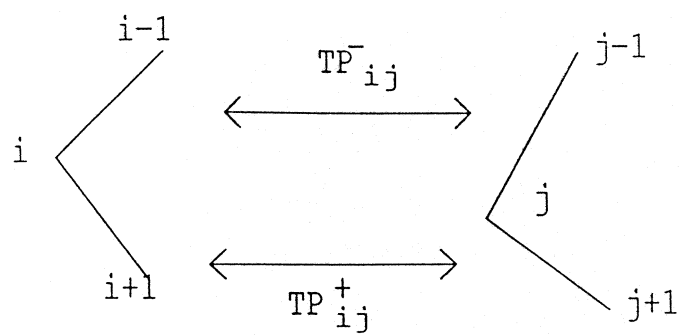


Abbildung 5.13: Vergleich einer Ecke $[z_{i-1}, z_i, z_{i+1}]$ des Testpolygonzugs mit einer Referenzecke $[z'_{j-1}, z'_j, z'_{j+1}]$

Literaturverzeichnis

- [1] Y.I. Abdel-Aziz and H.M. Karara. Direct linear transformation into object space coordinates in close-range photogrammetry. In *Proc. Symposium on Close-Range Photogrammetry*, pages 1–18, Urbana, Illinois, Jan 1971.
- [2] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, 1982.
- [3] L.M. Biebermann. *Perception of Displayed Information*. Plenum Press, NY/London, 1973.
- [4] I.N. Bronstein and K.A. Semendjajew. *Taschenbuch der Mathematik*. G. Grosche and V. Ziegler and D. Ziegler, Harri Deutsch, Thun, Frankfurt (Main), 23. Ausgabe, 1987.
- [5] B. Bässmann and P. W. Besslich. *Konturorientierte Verfahren in der digitalen Bildverarbeitung*. Springer-Verlag, Berlin, 1989.
- [6] H. Burkhardt. Algorithmen zur Digitalen Bildverarbeitung. Vorlesung an der Universität Freiburg.
- [7] H. Burkhardt. Digitale Bildverarbeitung II. Vorlesung an der Albert-Ludwigs-Universität Freiburg.
- [8] H. Burkhardt. Mustererkennung. Vorlesung am Institut für Informatik, Albert-Ludwigs-Universität Freiburg.
- [9] H. Burkhardt and H. Moll. Ein modifiziertes Newton-Raphson-Schema zur model-ladaptiven Identifikation von Laufzeiten. Interner Bericht 1/78, Institut für Meß- und Regelungstechnik, Universität Karlsruhe, 1978.
- [10] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 8/6:679–698, 1986.

- [11] J.L. Crowley and F. Berard. Multi-modal tracking of faces for object colour perception. In *Proceedings of CVPR'97*, pages 640–645, 1997.
- [12] D-M.Tsai and C.T. Lin. A moment-preserving approach for depth from defocus. *Pattern Recognition*, 31(5):551–560, 1998.
- [13] L. S. Davis. Shape matching using relaxation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(1):60–72, January 1979.
- [14] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, Cambridge, Massachusetts, 1993.
- [15] G. Finlayson, B. Schiele, and J.L. Crowley. Comprehensive colour image normalization. In *In H. Burkhardt and B. Neumann, Hrsg., Fifth European Conference on Computer Vision (ECCV) 1998*, Freiburg, jun 1998.
- [16] J. P. Foith. *Intelligente Bildsensoren zum Sichten, Handhaben, Steuern und Regeln*. Springer-Verlag, Berlin, 1982.
- [17] C. R. Giardina and E. R. Dougherty. *Morphological Methods in Image and Signal Processing*. Prentice Hall, Englewood Cliffs, 1988.
- [18] R. C. Gonzales and P. Wintz. *Digital Image Processing*. Addison Wesley, Reading, 1987.
- [19] R. Halin. *Graphentheorie I*. Wissenschaftliche Buchgesellschaft, Darmstadt, 1980.
- [20] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. In *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, volume 9/4, pages 532–550, 1987.
- [21] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.
- [22] G. Healey. Using color for geometry-insensitive segmentation. *Journal of the Optical Society of America, A*, 6:920–937, 1989.
- [23] F.Beichelt (Hrsg.). *Teubner-Taschenbuch der Stochastik : Wahrscheinlichkeitstheorie, stochastische Prozesse, mathematische Statistik*. Teubner, Stuttgart, 1. edition, 2003.
- [24] O.W. Hübsch and T. Maschke. *Das neue Lexikon der Fotopraxis*. information foto. Laterna magica, München, 1989.
- [25] G. Musiol I.N. Bronstein, K.A. Semendjajew and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun, Frankfurt am Main, 2. Ausgabe, 1995.

- [26] B. Jähne. *Digitale Bildverarbeitung*. Springer Verlag, Berlin, 1991.
- [27] B. Jähne. *Digitale Bildverarbeitung*. Springer Berlin Heidelberg, 1997.
- [28] A.K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
- [29] B. Liao. *Ein Beitrag zur Kamerafokussierung bei verschiedenen Anwendungen der Bildverarbeitung*. PhD thesis, Dissertation, Universität der Bundeswehr Hamburg, Juli 1993.
- [30] A. Luo. *Helligkeitsbasiertes Rechnersehen zur direkten Ermittlung räumlicher Eigenschaften*. PhD thesis, Dissertation, Technische Universität Hamburg-Harburg, Dezember, 1993.
- [31] I. D. G. Macleod. Comments on techniques for edge detection. In *Proceedings of the IEEE*, volume 60, page 344, 1972.
- [32] J. Matas, R. Marik, and J. Kittler. On representation and matching of multicoloured objects. In *Proceedings of ICCV'95*, pages 726–732, jun 1995.
- [33] The Math Works Inc. *Image Processing Toolbox for Use with Matlab*, jan 1997.
- [34] The Math Works Inc. *Matlab 5*, jan 1998.
- [35] K. Mehlhorn. *Graph Algorithms and NP-Completeness*. Springer-Verlag, Berlin, 1984.
- [36] B. Morgenstern. *Farbfernsehtechnik*, volume 4. Teubner Stuttgart, 1994.
- [37] H. Niemann. *Klassifikation von Mustern*. Springer-Verlag, Berlin, 1983.
- [38] H. Noltemeier. *Graphentheorie mit Algorithmen und Anwendungen*. Walter de Gruyter, Berlin, 1976.
- [39] T. Peli and D. Malah. A study of edge detection algorithms. *Computer Graphics and Image Processing*, 20:1–21, 1982.
- [40] A.P. Pentland. Depth of scene from depth of field. In *DARPA82*, pages 253–259, 1982.
- [41] A.P. Pentland. A new sense for depth of field. *PAMI*, 9(4):523–531, July 1987.
- [42] W.K. Pratt. *Digital Image Processing*. Wiley, New York, 1978.
- [43] W.K. Pratt. *Digital Image Processing*, volume 2. John Wiley & Sons, 1991.
- [44] A. Rosenfeld and A.C. Kak. *Digital Picture Processing 2nd ed.* Academic Press, New York, 1982.

- [45] H. Schorb. Modellhafter Einsatz von Mustererkennungsverfahren für Grauwertbilder und Binärbilder. Technical report, Institut für Meß- und Regelungstechnik der Universität Karlsruhe, 1983.
- [46] H. Schöttle. *DuMont's Lexikon der Fotografie*, volume 58 of *DuMont-Kunst-Taschenbücher*. DuMont Buchverlag, Köln, 1978.
- [47] J. Serra. Introduction to mathematical morphology. In *Computer, Vision, Graphics and Image Processing*, volume 35, pages 283–305, 1986.
- [48] J. R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression*. PhD thesis, Dissertation, Columbia University, 1997.
- [49] C. Teh and R. T. Chin. On the detection of dominant points on digital curves. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 11-8, pages 859–872, August 1989.
- [50] G. Teicher, editor. *Handbuch der Fototechnik*. VEB Fotokinoverlag, Leipzig, 8. edition, March 1983.
- [51] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.
- [52] S. G. Tyan. Median filtering: Deterministic properties. *Two-Dimensional Digital Signal Processing II (T.S.Huang ed.)*, 1981.
- [53] F. M. Wahl. *Digitale Bildsignalverarbeitung*. Springer-Verlag, Berlin, 1984.
- [54] M. Watanabe and S.K. Nayar. Rational filters for passive depth from defocus. *International Journal of Computer Vision*, 27(3):203–225, 1998.
- [55] Z. Zhang. Microsoft easy camera calibration tool. Microsoft Research.
- [56] Z. Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, 1998.
- [57] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22(11):1330–1334, 2000.