

Praktikumsversuch

Bildvorverarbeitung

R. Klaus, K. Peschke und H. Burkhardt

Inhaltsverzeichnis

1	Bildvorverarbeitung	5
1.1	Einleitung	5
1.2	Bildverbesserung durch Punktoperatoren und lokale Filterung . .	6
1.2.1	Grauwerttransformationen	7
1.2.2	Lineare Lokaloperatoren	12
1.2.3	Medianfilter	16
1.3	Kantenextraktion	21
1.3.1	Gradientenoperator	22
1.3.2	Laplaceoperator	25
1.3.3	Differenzen von Mittelwerten	27
1.3.4	Cannyoperator	28
1.4	Binarisierung und morphologische Binärbildverarbeitung	32
1.4.1	Binarisierung	33
1.4.2	Dilatation	33
1.4.3	Erosion	36
1.4.4	Open und Close	38
1.5	Aufgaben	41
1.5.1	Theoretische Aufgaben	41
1.5.2	Praktische Aufgaben	42

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.

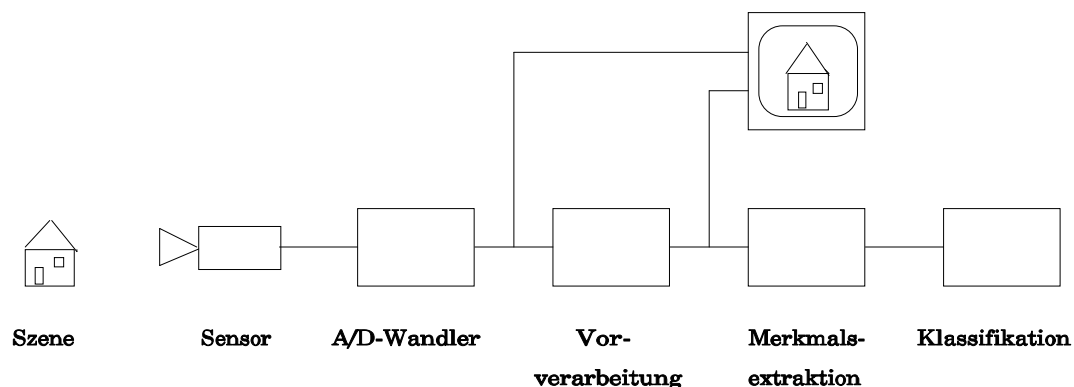


Abbildung 1.1: Allgemeines Schema zur Bildverarbeitung und Mustererkennung

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. emphasis accordingly

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 nichtlineare Variante der Lokaloperatorfilterung kennenlernen, die wir im allgemeinen Fall als *Rangordnungsfilter*, im hier nur behandelten speziellen Fall als *Medianfilter* [12] 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 Rangordnungsfilter als Maxfilter oder Minfilter bezeichnen. Diese Art der Rauschunterdrückung eignet sich besonders für pulsförmige Störungen. Feine Bilddetails bleiben dabei weitestgehend erhalten. Wir werden das Verhalten an entsprechender Stelle noch genauer analysieren.



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

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 $s = r^\gamma$ (Gammakurve) 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.

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.

Ansammlung

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

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

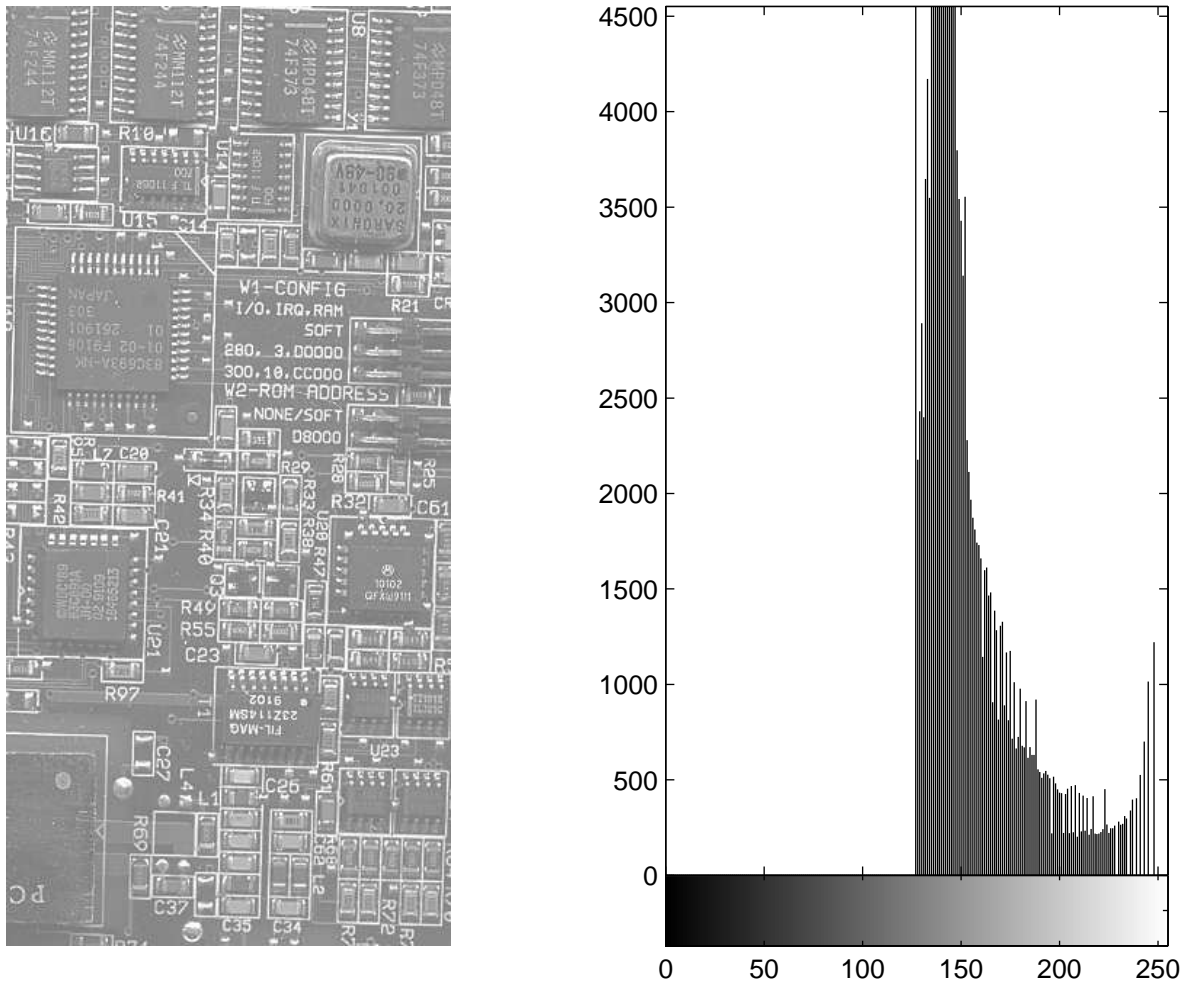


Abbildung 1.3: Histogramm eines Bildes

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 [6] 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:

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

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.

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

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.

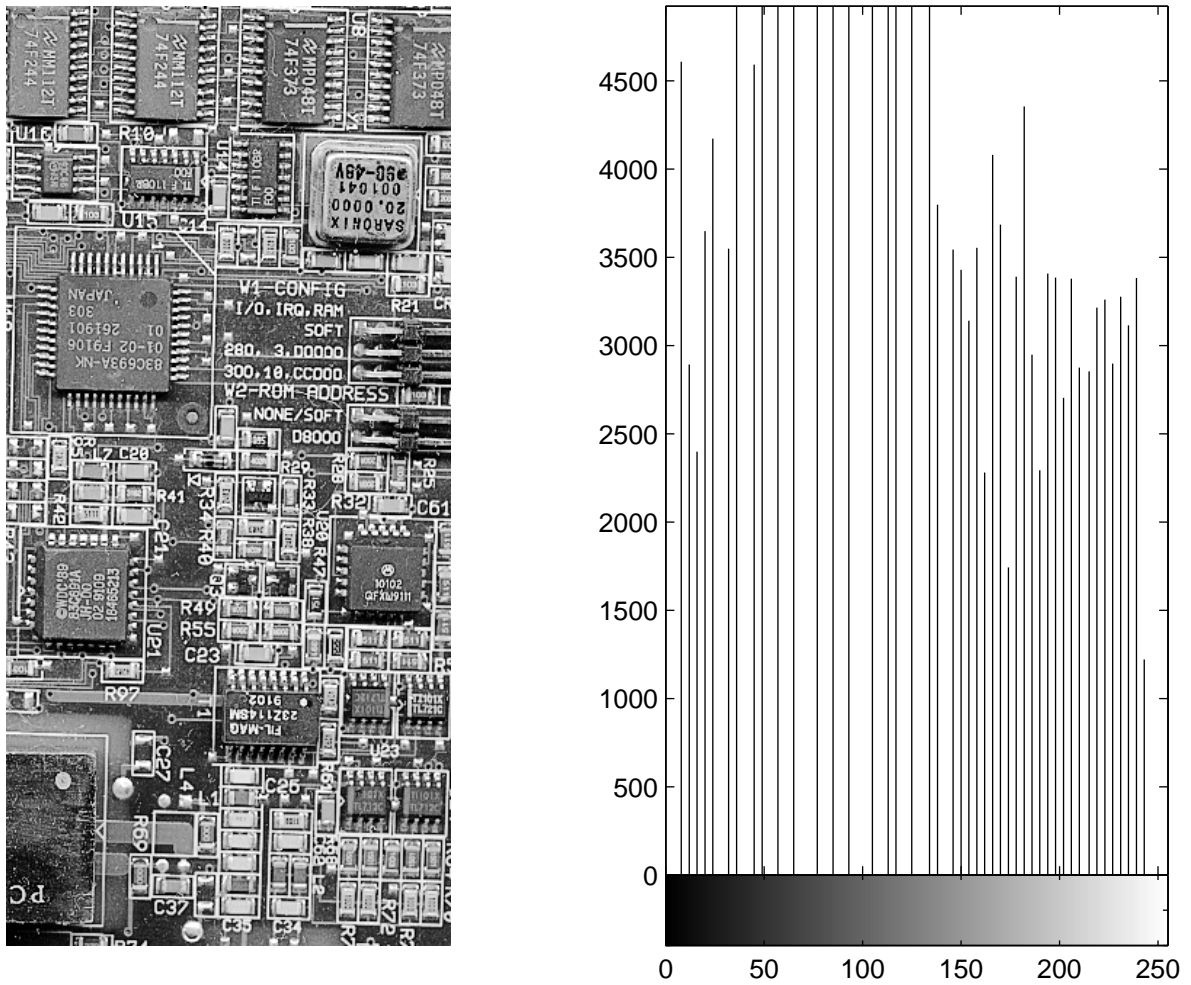


Abbildung 1.4: ausgeglichenes Histogramm eines Bildes

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.

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 Rauschanteile 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.

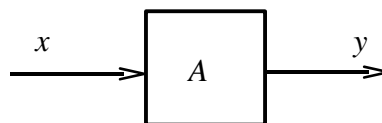


Abbildung 1.5: Prinzipskizze eines linearen Systems

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:

$$y = Ax \quad (1.10)$$

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 [1] müssen.

Definition 1.1 (Superpositionsprinzip)

$$A(x_1 + x_2) = Ax_1 + Ax_2 \quad (1.11)$$

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.

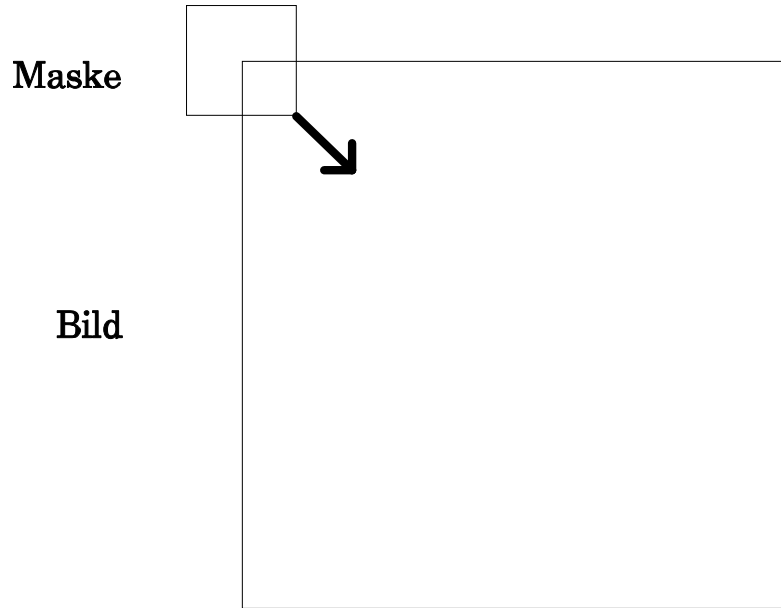


Abbildung 1.6: allgemeine Lokaloperation

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 [1] und [13] gezeigt wird, gelten für die Faltung eines Bildes X mit dem Operator H folgende Zusammenhänge:

Definition 1.5 (**Isomorphie** der FT) 

$$Y = H^{(+, \times)} * X \quad (1.18)$$



$$\tilde{Y} = \tilde{H} \cdot \tilde{X} \quad (1.19)$$

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 Faltung 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:

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

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 [9].

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 nichtlinearen 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)

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

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 med_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:

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

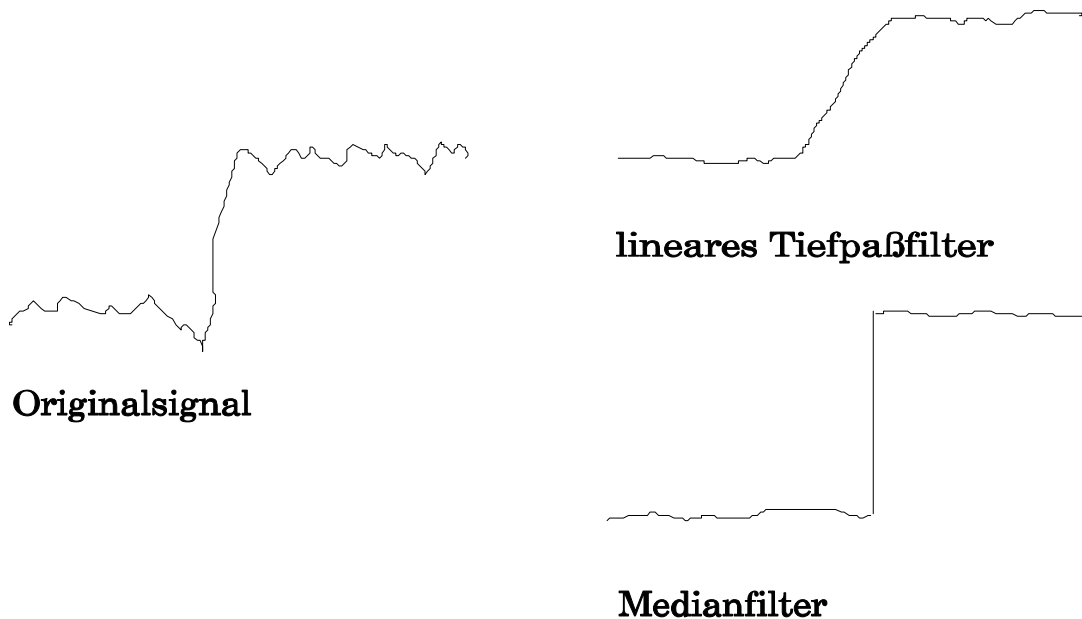


Abbildung 1.7: Vergleich von linearem Tiefpaßfilter und Medianfilter

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 [12].

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

Daraus läßt sich leicht ableiten, daß eine Sequenz auch dann $lomo(m)$ 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 [12].

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 [12].

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 waagerecht verlaufen.

Neben der Fensterform ist der notwendige Sortiervorgang der zweite wichtige Punkt. Wir wollen an dieser Stelle nicht auf spezielle Sortiervverfahren 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 applikationsspezifische Ergebnisse liefern. Bevor wir den inhaltlichen Teil des gesamten Kapitels abstecken, soll zunächst eine terminologische Einordnung vorgenommen werden.

Eigenschaft

Die *Kantenextraktion* kann sowohl als ein Teil der *Bildverbesserung* (image enhancement) als auch der *Bildsegmentierung* (image segmentation) angesehen werden [10, 8]. 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 [13]. 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. Kapitel 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 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)

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

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

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

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

preferential direction

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 [13, 4]. Ausgehend von der Definition im kontinuierlichen Fall kann die Wurzel aus der Quadratsumme gebildet werden.

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

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:*

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

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 Richtungsdivergenzen 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 Wertepegel charakterisiert sind, die durch Differenzbildungen hervorgerufen werden. Es wurden keine mathematisch spezifizierten Anforderungen an

das Verhalten des Operators gestellt. Canny [2] 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 [2].

Spezialisiert man das Optimierungsverfahren auf sprungförmige Kantenverläufe, wie in Abb. 1.8 [2] geschehen, so kann man eine parametrisierte geschlossene Form der Lösung angeben. Bei der Herleitung sieht man, daß die Anforderungen nach geringer Fehlerrate und gutem Lokisierungsverhalten 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:

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

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 Macload [7] angegeben. Der Operator bildet die Differenz zweier gegeneinander versetzter Gaußfunktionen, die wiederum mit einer dritten im Ursprung lokalisierten Gaußfunktion multipliziert werden.

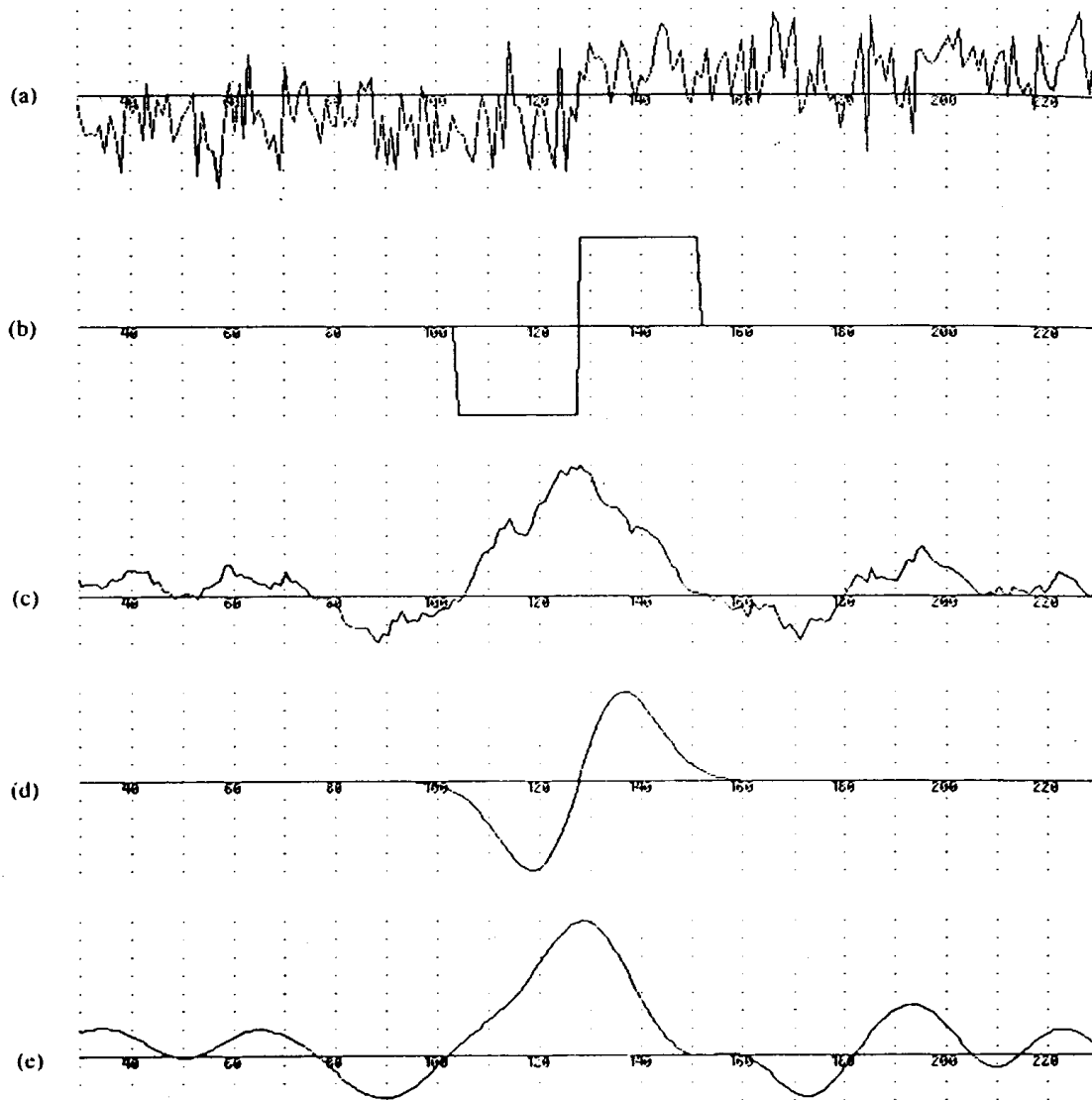


Abbildung 1.8: Extraktion einer verrauschten Sprungkante [8]: (a) verrauschte Sprungkante, (b) Operator für Differenz von Mittelwerten, (c) Operator aus (b) angewandt auf Signalverlauf aus (a), (d) erste Ableitung eines Gaußoperators, (e) Operator aus (d) angewandt auf Signalverlauf aus (a)

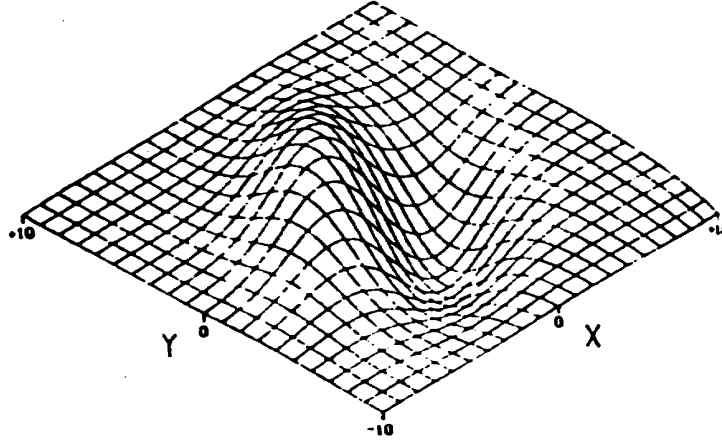


Abbildung 1.9: Zweidimensionale Näherung nach Macleod [9]

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

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 [7] 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 [3, 11, 5]. 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.

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

Diese einfache Methode ist nur für den Fall eines bimodalen Grauwerthistogramms 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 *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

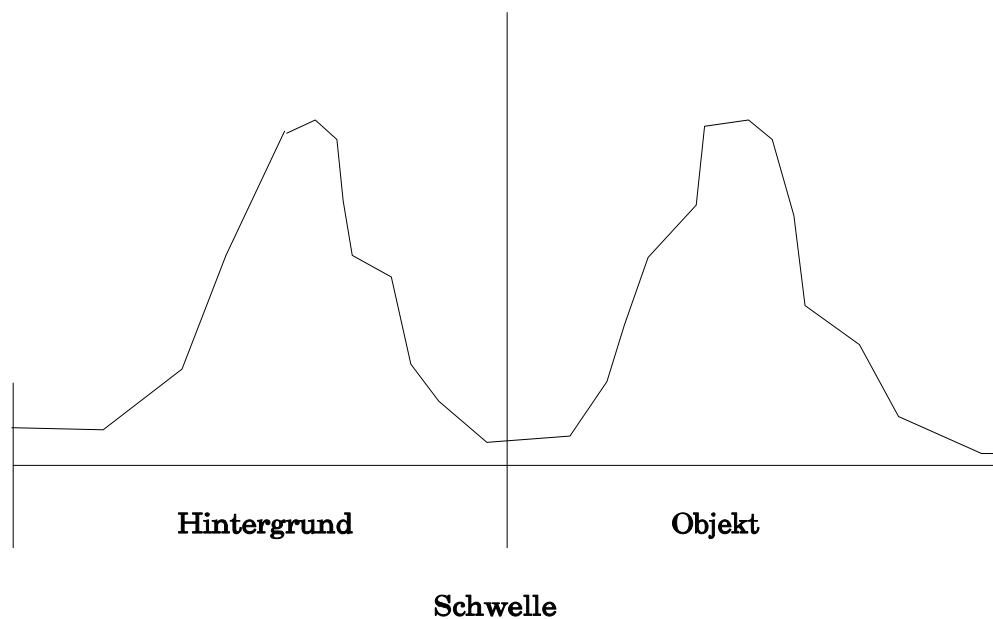


Abbildung 1.10: Binarisierungsschwelle bei bimodalem Grauwerthistogramm

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.

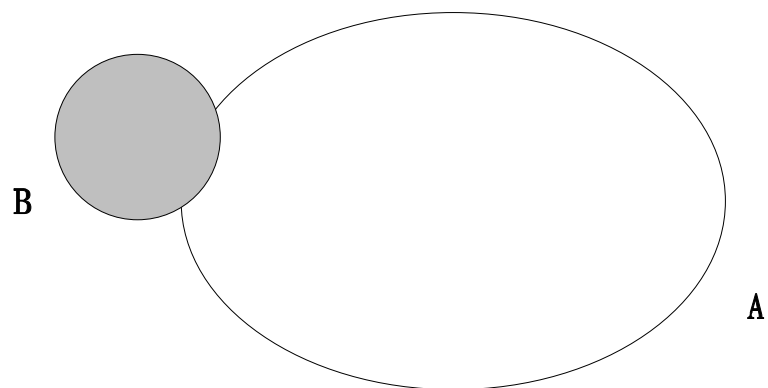


Abbildung 1.11: Strukturelement B und Bild A überlappen sich

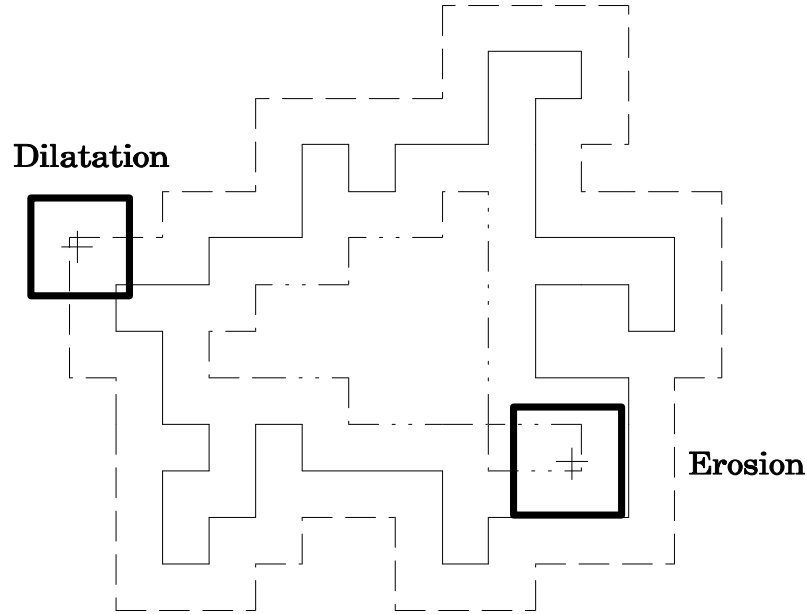


Abbildung 1.12: Dilatation und Erosion an einer Beispielfigur (durchgezogene Linie: Originalbild, Strichlinie: Dilatation, Strichpunktlinie: Erosion)

Definition 1.16 (Dilatation)

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

$$A \oplus B = B(\vee LOK \wedge)A \quad (1.38)$$

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 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

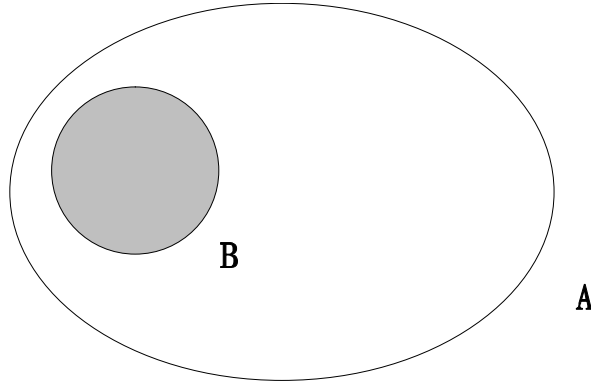


Abbildung 1.13: Bild A majorisiert Strukturelement B

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)

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

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 ortsinvariantes 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 $A1$ im Bild $A2$ enthalten und werden beide mit dem gleichen Strukturelement erodiert, so ist die Erosion von $A1$ in der von $A2$ enthalten. Man bezeichnet in der Literatur ein solches Operatorverhalten als *increasing*. Sind auf der anderen Seite $A1$ und $A2$ Strukturelemente und ist wiederum $A1$ in $A2$ enthalten, so ist die Erosion von B mit $A2$ in der mit $A1$ 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

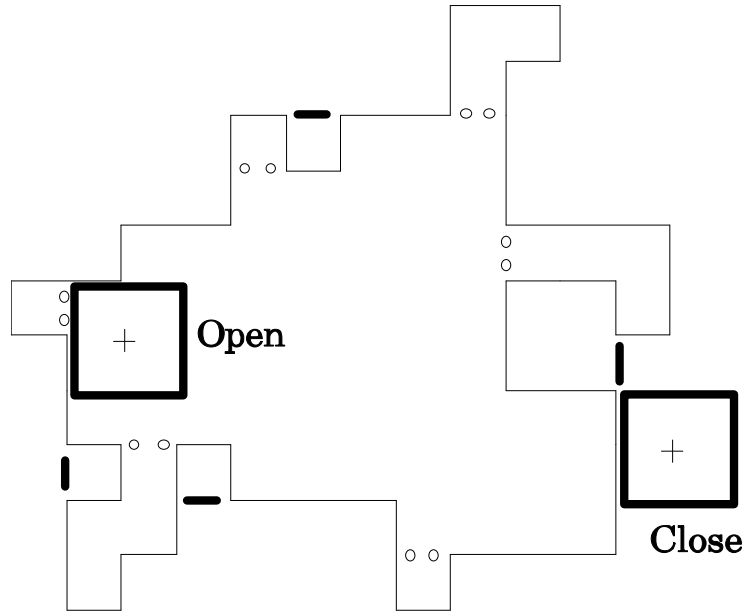


Abbildung 1.14: Open (Punktlinie) und Close (Strichlinie) angewandt auf das Binärbild aus Abb.4.3 mit quadratischem Strukturelement

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 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:

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

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 Pefferrauschen hinzu und filtern Sie das Bild.

Aufgabe 8: (Bildverschärfung) Überlegen Sie wie man unter Benutzung eines Filters und Verwendung einfacher arithmetrischer 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 `special` 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 Quadratsumme (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 iterartiver 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(...)`, `imerode(...)`, `imdilate(...)`,
`imopen(...)`, `imclose(...)`, `strel(...)`

Bilder in Matlab

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

Literaturverzeichnis

- [1] H. Burkhardt. Algorithmen zur Digitalen Bildverarbeitung. Vorlesung an der Universität Freiburg.
- [2] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 8/6:679–698, 1986.
- [3] C. R. Giardina and E. R. Dougherty. *Morphological Methods in Image and Signal Processing*. Prentice Hall, Englewood Cliffs, 1988.
- [4] R. C. Gonzales and P. Wintz. *Digital Image Processing*. Addison Wesley, Reading, 1987.
- [5] 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.
- [6] F. Beichelt (Hrsg.). *Teubner-Taschenbuch der Stochastik : Wahrscheinlichkeitstheorie, stochastische Prozesse, mathematische Statistik*. Teubner, Stuttgart, 1. edition, 2003.
- [7] I. D. G. Macleod. Comments on techniques for edge detection. In *Proceedings of the IEEE*, volume 60, page 344, 1972.
- [8] T. Peli and D. Malah. A study of edge detection algorithms. *Computer Graphics and Image Processing*, 20:1–21, 1982.
- [9] W.K. Pratt. *Digital Image Processing*. Wiley, New York, 1978.
- [10] A. Rosenfeld and A.C. Kak. *Digital Picture Processing 2nd ed.* Academic Press, New York, 1982.
- [11] J. Serra. Introduction to mathematical morphology. In *Computer, Vision, Graphics and Image Processing*, volume 35, pages 283–305, 1986.
- [12] S. G. Tyan. Median filtering: Deterministic properties. *Two-Dimensional Digital Signal Processing II (T.S.Huang ed.)*, 1981.
- [13] F. M. Wahl. *Digitale Bildsignalverarbeitung*. Springer-Verlag, Berlin, 1984.