

## Bildverbesserung und Kantenextraktion

### Grundlagen

Unter Bildverbesserung versteht man die Optimierung des Bildsignals in Hinblick auf eine Anwendung, z.B. Unterdrückung uninteressanter Grauwertbereiche und Hervorhebung interessierender Strukturen. Im Gegensatz zur Bildrestauration unterliegt sie subjektiven Kriterien.

Leider gibt es keine universelle Methode zur Bildverbesserung. Die bei den einzelnen Anwendungen geforderten Veränderungen lassen sich kaum miteinander vergleichen, da kein einheitliches Qualitätskriterium der Bildverbesserung existiert, nach dem eine Bewertung vorgenommen und auch ein universeller Algorithmus zur Bildverbesserung entwickelt werden könnte. Deshalb werden im folgenden die wichtigsten allgemein verwendeten Methoden zur Bildverbesserung vorgestellt.

Das einfachste Verfahren zur Bildverbesserung besteht in der Änderung von Helligkeit und Kontrast. Dabei wird die Grauwerte Verteilung des Bildes modifiziert. Mit allen üblichen Bildbearbeitungsprogrammen kann man Helligkeit und Kontrast ändern. Unter Unix ist z.B. das Programm `xvview` (Aufruf mit `xv`) sehr verbreitet. Mit `xvview` kann die Modifikation der Grauwerttabelle direkt beobachtet werden.

Weitere Verfahren zur Bildverbesserung sind Filter zur Glättung oder zur Konturbetonung.

### Glättungsfiler

Es gibt eine Vielzahl von Bildtransformationen, die die Nachbarschaft eines Bildpunktes mit verwenden:

Ein Nachbarschaftsfenster der Größe  $(2n+1) \times (2m+1)$ , das sogenannte *Koppelfeld*, wird über das Bild bewegt. Dabei werden die Bildpunkte innerhalb des Fensters durch eine Funktion  $g$  miteinander verknüpft und das Ergebnis dem zentral im Fenster liegenden Pixel zugewiesen (Abbildung 1).

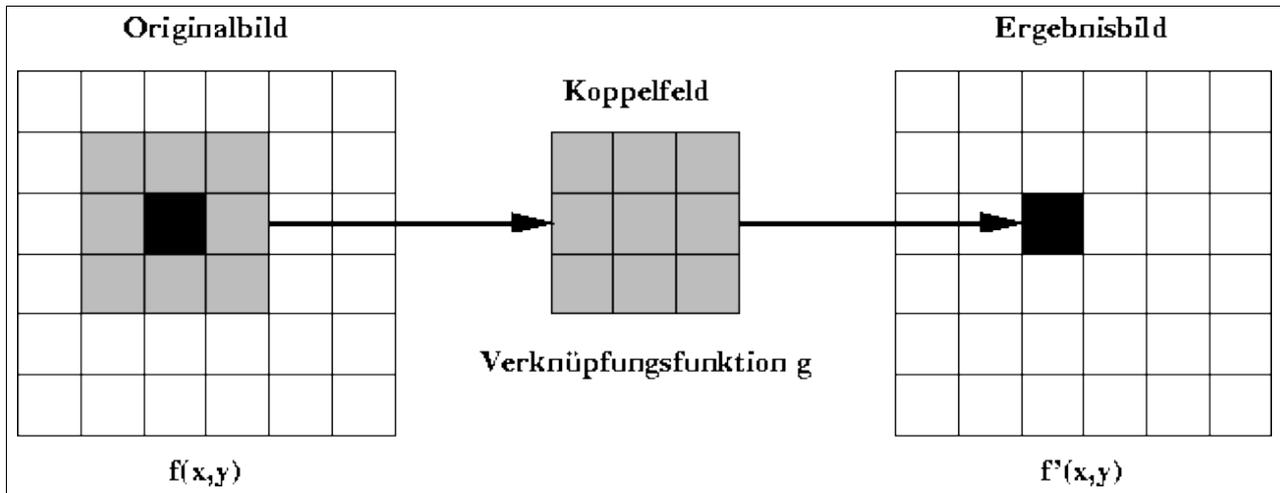


Abbildung 1: Koppelfeldprozess

Das Bild wird punktwise abgetastet. Alle Punkte, die sich in einem Rahmen definierter Größe um den aktuellen Punkt befinden, werden zur Berechnung des neuen Grauwerts herangezogen.

Ein wichtiger Sonderfall ist die lineare Verknüpfung der Koppelfeldpunkte mit  $g$ :

$$f'(x, y) = \sum_{i=-n}^n \sum_{j=-m}^m f(x + i, y + j) g(i, j)$$

$f(x,y)$  bezeichnet den Grauwert des Pixels an der Stelle  $(x,y)$ . Diese Transformationen werden als *Koppelfeldprozesse* bezeichnet. Mit geeignet gewählten Funktionen  $g$  kann man z.B. Rauschstörungen glätten oder Objektkanten hervorheben.

Für den Randbereich eines zu transformierenden Bildes ergeben sich mit obiger Gleichung Probleme, wenn das Koppelfeld über den Bildrand hinausragt. Beispielsweise ist der Grauwert an der Position  $(-1,-1)$  nicht definiert. Abhilfe schafft eine spezielle Randbehandlung. Es bieten sich mehrere Möglichkeiten an:

- Der Randbereich wird im transformierten Bild weggelassen. Dadurch wird dieses jedoch kleiner, was oftmals nachteilig ist.
- Der Randbereich wird gesondert behandelt. Dabei wird durch adaptive Anpassung der Koppelfeldgröße erreicht, dass bei der Transformation auch nur auf tatsächlich vorhandene Punkte zugegriffen wird.
- Das Ausgangsbild wird mit einem  $n$  bzw.  $m$  Pixel breiten Rand versehen. Dieser Rand kann z.B. mit einem konstanten Grauwert gefüllt werden. Es ist auch üblich, das Ausgangsbild periodisch oder konstant in diesen Rand hinein fortzusetzen.

Glättungsfilter dienen der Glättung der Bildfunktion z.B. mit dem Ziel, Signalfuktuationen, die durch Rauschüberlagerungen (zufällige Grauwertschwankungen) bei der Signalaufnahme entstanden sind, zu minimieren.

Eine einfache Transformation zur Glättung ist der **Mittelwertfilter**. Hierbei werden die Grauwerte im Koppelfeld gemittelt:

$$f'(x, y) = \frac{1}{(2m + 1)(2n + 1)} \sum_{i=-n}^n \sum_{j=-m}^m f(x + i, y + j)$$

Für  $n=m=1$  erhält man damit folgende  $3 \times 3$ -Koppelfeldmatrix  $g$ :

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Mittelwertgefilterte Bilder wirken im Vergleich zum Ausgangsbild weicher bzw. unschärfer. Dabei ist der Grad der Unschärfe abhängig von der Größe des Koppelfeldes. Homogene Grauwertbereiche bleiben unverändert, hingegen werden Grauwertübergänge geglättet und dadurch auch Objektkanten verschmiert. Es lässt sich zeigen, dass dem Ausgangsbild überlagertes Bildrauschen verringert wird.

Um den unerwünschten Effekt der Verschmierung von Objektkanten zu minimieren, können nichtlineare Glättungsfilter eingesetzt werden: Beim **Medianfilter** (s. Abbildung 2) wird dem Bildpunkt  $f(x,y)$  der Median der Grauwerte aus dem zugehörigen Koppelfeld zugewiesen. Dazu werden die Grauwerte der Größe nach sortiert. Der Median dieser Folge ist dann der in der Mitte stehende Wert. Die Hälfte der Grauwerte im Koppelfeld sind kleiner oder gleich, die andere Hälfte größer oder gleich im Vergleich mit diesem.

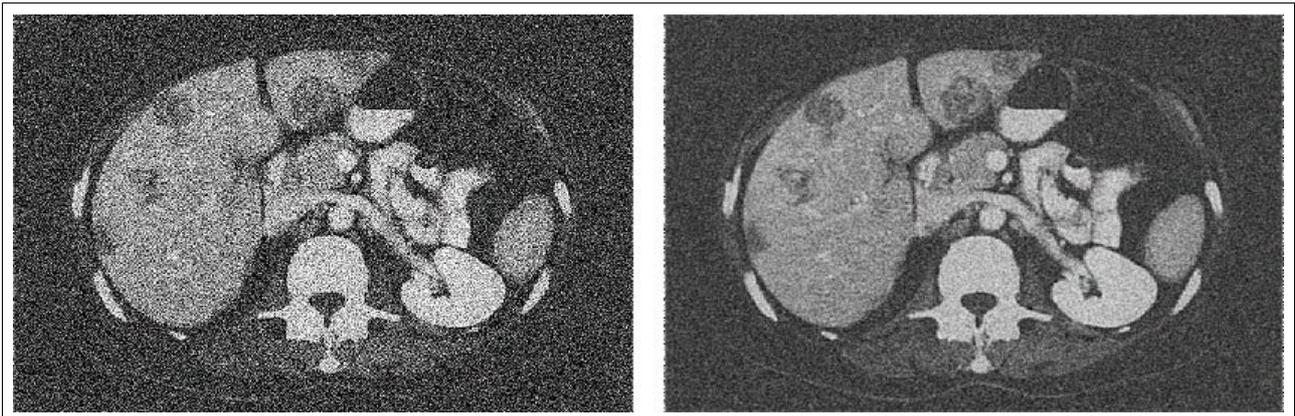


Abbildung 2: Original (links) und mediangefiltertes Bild (rechts).

Aus dieser Rechenvorschrift ergibt sich, dass z.B. bei sporadischen Bildstörungen die isolierten Grauwertspitzen oder -tiefen eliminiert werden. Ebenso ergibt sich bei Objektkanten ein geringer Einfluß der evtl. völlig anders gearteten Grauwerte jenseits der Objektgrenze und somit bleiben die Kanten weitgehend erhalten.

### ***Kantenextraktion***

Objekte haben meist deutlich sichtbare Kanten, anhand derer eine Segmentierung vorgenommen werden kann. In der medizinischen Bildgebung werden die Objekte in den Aufnahmen durch unterschiedliche Grauwerte dargestellt, wobei die zugehörigen Objektkanten als lokale Grauwertübergänge abgebildet werden. Durch Auffinden dieser Grauwertübergänge können die Konturen der Objekte extrahiert und damit die Objekte segmentiert werden.

An einer Objektkante ändert sich der Verlauf der Bildfunktion  $f$  der Pixelgrauwerte idealerweise sehr stark (z.B. Übergang von einem weißen zu einem schwarzen Bereich).

Dieser Anstieg kann durch Berechnung der 1. und 2. Ableitung der Bildfunktion leicht lokalisiert werden. Die erste Ableitung hat an einer Kante ein lokales Maximum, die zweite Ableitung einen Nulldurchgang. Im Zweidimensionalen werden die partiellen Ableitungen  $\delta f(x,y)/\delta x$  und  $\delta f(x,y)/\delta y$  verwendet. Nach ihrer Definition gilt:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{t \rightarrow 0} \frac{f(x+t,y) - f(x,y)}{t}$$

Viele Kantendetektoren berechnen die Kanten mit Hilfe des Gradienten der Bildfunktion  $f$ . Er ist definiert als

$$\text{grad } f = \left( \frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \right) \approx (\Delta_x f(x,y), \Delta_y f(x,y)).$$

Bei den Kantendetektoren wird der Betrag (und seltener auch die Richtung) des Gradienten ausgewertet.

$$|\text{grad } f| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2} \approx \sqrt{(\Delta_x f(x,y))^2 + (\Delta_y f(x,y))^2}.$$

Bei Pixelbildern werden die Ableitungen durch Differenzen approximiert und  $t$  gleich  $-1$  gesetzt (In diesem Fall erhält man den *Rückwärtsgradienten*).

$$\frac{\Delta f(x,y)}{\Delta x} = \Delta_x f(x,y) = \frac{f(x+t,y) - f(x,y)}{t} = f(x,y) - f(x-1,y) \quad \text{für } t = -1.$$

Analog ist  $\Delta_y f(x,y)$  zu berechnen.

In solch einem diskreten Falle kann man die Differenzoperatorberechnung auch als Faltung mit einem Koppelfeld  $g$  auffassen:

$$\Delta f(x,y) = \sum_{i=-n}^n \sum_{j=-m}^m f(x+i,y+j)g(i,j).$$

Für den einfachen Rückwärtsgradienten ergeben sich folgende  $3 \times 3$ -Matrizen:

$$g_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad g_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$g_x$  spricht nur auf vertikale,  $g_y$  auf horizontale Kanten an. Bei diesem einfachen Differenzoperator fehlt jedoch eine Glättung und damit ist er sehr rauschempfindlich. Weiterhin ist er nicht isotrop, d.h. die Filterantwort ist nicht unabhängig von der Orientierung der Kante, sondern am stärksten bei einem Kantenverlauf senkrecht zur Ableitungsrichtung.

Ähnlich wie der Rückwärtsdifferenzoperator arbeitet der Roberts-Operator oder auch **Roberts-Gradient**. Hier werden vor allem diagonal verlaufende Kanten gut erkannt. Für ihn gelten die gleichen Vor- und Nachteile wie für den einfachen Differenzoperator.

$$g_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Der **Sobel-Operator** ist einer der am häufigsten verwendeten Kantendetektoren. Er enthält eine Glättung mit einer approximierten Gaußverteilung. Mit dieser Wichtung soll die Entstehung von Artefakten bei der Filterung verhindert und die Auswirkung im Ortsfrequenzraum regional begrenzt werden.

$$g_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad g_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Ein reines Kantenbild kann durch eine anschließende Schwellwertbildung auf den Betrag des Gradienten erzeugt werden. Übersteigt dieser einen vorgegebenen Schwellwert, wird im Ergebnisbild ein weißes Pixel gesetzt, ansonsten ein schwarzes.



Abbildung 3: Kantendetektion. Von links nach rechts: Originalbild, Prewitt-Operator, Sobel-Operator.

## Vorbereitung

1. Machen Sie sich mit den Rahmenroutinen in der Datei Modify.cpp vertraut.
2. Überlegen Sie, wie Sie bei der Implementierung von Koppelfeldprozessen Rechenoperationen und damit Rechenzeit sparen können.

## Aufgaben

1. Installieren Sie sich die Imcon Programmdateien mithilfe des selbstextrahierenden Zip-Files dbv\_praktikum.exe in ein eigenes Verzeichnis.
2. Hinterlegen Sie in den 4 Routinen Modify2DIB(), ...,Modify5()DIB den Code für folgende Algorithmen: Mittelwertfilter, Medianfilter, Robertsgradient, Sobeloperator.
3. Verwenden Sie beim Median- und Mittelwertfilter 3x3 und auch 5x5 Pixel große Koppelfelder. Vergleichen Sie die erzielten Resultate.
4. Ergänzen Sie die Programme zur Kantendetektion um eine Schwellwertberechnung.
5. Demonstrieren Sie an einem Beispielbild die Wirkungsweise der vier Bildoperatoren. Zeigen Sie die anhand der Schwellwertbildung aufgefundenen Objektkanten.

## Hinweise

Die Programmierung erfolgt auf der Basis des Rahmenprogramms Imcon aus dem Buch Grafikformate von Thomas W. Lipp. Dieses stellt eine grafische Benutzungsoberfläche zur Verfügung. Der Funktionsumfang umfasst das Einlesen/Abspeichern verschiedener Bildformate, die Wandlung eines Farbbildes in ein Grautonbild (mit 8 Bit pro Pixel), die Monitor-Darstellung und den Druck von Bildern. Als interne Bilddatenstruktur verwendet das Programm eine DIB (device independant bitmap). Eine DIB besteht aus einem Informationsteil (z.B. Bildgröße in Anzahl Pixel pro Zeile und Spalte, Lookup-Tabelle, Anzahl Bits pro Pixel) und den eigentlichen Bilddaten (Pixel-Farbwerte).

Im Menüpunkt Bearbeiten stehen 9 Auswahlmöglichkeiten Modify1, ..., Modify9 zur freien Verwendung zur Verfügung. Dahinter verbergen sich die Routinen Modify1DIB(), ..., Modify9(DIB). Diese stellen den Programmierrahmen für die Implementierung eigener Algorithmen bereit und befinden sich in der Datei Modify.cpp. Die Modify\*DIB()-Routinen wandeln das eingelesene Bild zunächst in ein Grautonbild um und stellen es in einem Ausgabefenster dar. Dann wird die DIB des gewandelten Bildes kopiert und ein Mechanismus zum Zugriff auf die Pixeldaten in einer Doppelschleife (über alle Zeilen und Spalten des Bildes) bereitgestellt.

Mit dem Applikations-Programm Paintshop Pro können Bilder angezeigt, verändert und in unterschiedlichen Formaten abgespeichert werden.

Geeignete Beispielbilder finden Sie Beispielbilder-Ordner.