

Nachrichtenübertragung - Softwarepraktikum

Termin 14:00 Uhr - 15:45 Uhr

Protokoll Nr. 4

Gruppe "sw06c"

Tilman Knebel (Matr. Nr: 195298)

<tilman@cs.tu-berlin.de>

Christian Richter (Matr. Nr: 192548)

<richterc@cs.tu-berlin.de>

17. Juni 2003

Inhaltsverzeichnis

1	Übungsblatt 7 – Bildsignalverarbeitung I	1
1.1	Vorbereitungsaufgaben	2
1.1.1	Periodizität	2
1.1.2	Konjugiert komplexe Zonen im zweidimensionalen Frequenzbereich	2
1.1.3	Transformationspaare	3
1.1.4	Skalierung bei Rücktransformation	3
	Beispiel a)	3
	Beispiel b)	3
	Beispiel c)	4
1.2	Übungsaufgaben	4
1.2.1	Erzeugung eines Spektrums aus Deltapulsen	4
	Beispiel a)	5
	Beispiel b)	6
	Beispiel c)	6
1.2.2	Transformation der Spektren in den Ortsbereich	6
	Beispiel a)	7
	Beispiel b)	8
	Beispiel c)	8
1.2.3	Manipulationen von Signalen im Ortsbereich	9
	Erzeugung eines Quadrates	9
	Erzeugung eines Kreises	10
1.2.4	Transformation der Bilder in den Frequenzbereich	10
	Quadrat im Frequenzbereich	10
	Kreis im Frequenzbereich	12

2	Übungsblatt 8 – Bildsignalverarbeitung II	14
2.1	Vorbereitungsaufgaben	14
2.1.1	Skalierungseigenschaften der zweidimensionalen FFT	14
2.1.2	Spektrum eines verschobenem Deltapulses	15
2.1.3	Zweidimensionales Filter	15
2.2	Übungsaufgaben	17
2.2.1	Bildausschnittverschiebung	17
	Bild einlesen	17
	Bildausschnitt kopieren und verschieben	17
2.2.2	Bild-Tiefpassfilterung	18
	Bild einlesen	18
	Zweidimensionale Tiefpaßfilterung	19
2.2.3	Histogramm von Bildsignalen	20

Zusammenfassung

Bei der Bildverarbeitung ist – ähnlich wie im Audibereich – eine Abbildung in den Frequenzbereich sinnvoll. Dafür es wird die zweidimensionale diskrete Fouriertransformation (2D-DFT) genutzt. Im Gegensatz zu Audiosignalen bezieht sich die Frequenz im Spektrum hier auf die Struktur der Farbwechsel. Eine hohe Frequenz entspricht einem fein strukturierte Muster, eine niedrige Frequenz einem groben Muster. In den Versuchen wird der Einfluß von Bandbereichsfilterungen im Frequenzbereich auf die entsprechenden Bildstrukturen im Ortsbereich gezeigt.

Kapitel 1

Übungsblatt 7 – Bildsignalverarbeitung I

Die zweidimensionale Fouriertransformation wandelt eine diskrete, reelle Matrix, die z.B. Helligkeitswerte eines Bildes enthält, in eine komplexe Matrix mit den Frequenzanteilen dieses Bildes um.

Diese Matrix ist ein Periodenausschnitt aus dem diskreten, periodischen Spektrum, wobei der erste Eintrag oben links den konstanten Signalanteil repräsentiert, und die anderen Werte wegen des reellen Eingangssignals die üblichen Symmetrieeigenschaften der Fouriertransformation besitzen.

Um die Lesbarkeit dieser Frequenzmatrix zu erhöhen, wird sie vor der Darstellung derart umgestellt, daß der Mittelwert von oben links in die Bildmitte wandert, die niedrigen Frequenzen um die Mitte herum herum, und die hohen Frequenzen am Rand erscheinen.

Definition der zweidimensionalen FFT:

$$X(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \cdot e^{-j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)} \quad (1.1)$$

Die Größen M und N entsprechen dabei den Kantenlängen der Matrizen.

Für die Rücktransformation vom Frequenz- in den Ortsbereich benutzt man die inverse diskrete Fourier-Transformation, die durch

$$x(m, n) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} X(u, v) \cdot e^{j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)} \quad (1.2)$$

gegeben ist.

1.1 Vorbereitungsaufgaben

1.1.1 Periodizität

Bei der digitalen Signalverarbeitung liegen die Daten zeitdiskret vor. Es werden Signalausschnitte betrachtet, die periodisch bis ins Unendliche erweitert werden, um nach einer Fouriertransformation wiederum ein zeitdiskretes Signal zu erhalten. Setzt man einen Wert aus einem beliebigen Periodenausschnitt des Ortsbereiches in die Transformationsgleichung ein, erhält man

$$X(u + aM, v + bM) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \cdot e^{-j2\pi\left(\frac{um+amM}{M} + \frac{vn+bnN}{N}\right)} \quad (1.3)$$

Wegen $e^{j2\pi(\varphi+k)} = e^{j2\pi\varphi} \forall k \in \mathbb{Z}$ kann man Gleichung 1.3 vereinfachen zu

$$X(u + aM, v + bM) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \cdot e^{-j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)} = X(u, v)$$

1.1.2 Konjugiert komplexe Zonen im zweidimensionalen Frequenzbereich

Wie für die eindimensionale Fouriertransformation wird aus einem reellen Eingangssignal ein komplexes Spektrum, welches für jeden Wert einen konjugiert komplexen, symmetrisch angeordneten Gegenwert besitzt. Dies gilt auch im zweidimensionalen Fall. Unter der Voraussetzung, dass der Imaginärteil des Signals im Ortsbereich immer gleich Null sein muß (es gibt keine komplexen Bildsignale) ergibt sich:

$$\Im[x(m, n)] = 0$$

$$\Im \left[\frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} X(u, v) \cdot e^{j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)} \right] = 0$$

Diese Gleichung ist erfüllt, solange für jedes $X(u, v)$ ein konjugiert komplexer Wert $X^*(-u + aM, -v + bN)$ existiert. In der Summe entstehen dann Terme der Form

$$X(u, v) \cdot e^{j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)} + X^*(-u, -v) \cdot e^{j2\pi\left(\frac{-um}{M} + \frac{-vn}{N}\right)}$$
$$2\Re[X(u, v)] \cdot \cos\left(2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)\right) + 2\Im[X(u, v)] \cdot \sin\left(2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)\right)$$

Durch das Einfügen der konjugiert komplexen Werte entstehen also – entsprechend der gegebenen Bedingung – Signale ohne Imaginärteil.

1.1.3 Transformationspaare

Für einen Deltapuls im Ursprung ergibt sich nach Einsetzen von $X = \delta_{u,v}$ in die Rücktransformationsgleichung (Gleichung 1.2) ein konstanter Wert im Bildbereich. Dabei wird die Ausblendeigenschaft zur Auflösung der Summe genutzt und die Kantenlänge von $M = N = 64$ eingesetzt:

$$x(m, n) = \frac{2}{MN} \cdot e^{-2\pi \cdot 0} = \frac{1}{64 \cdot 64} = \frac{1}{4096} \quad (1.4)$$

Wird hier für x ein Deltapulspar $x = \delta_{u,v} + \delta_{-u,-v}$ eingesetzt, ergibt sich

$$x(m, n) = \frac{2}{MN} \cdot \cos\left(2\pi \frac{um}{M} + \frac{vn}{N}\right)$$

Ein Deltapulspar im Frequenzbereich führt also zu einer Cosinusschwingung im Bildbereich.

1.1.4 Skalierung bei Rücktransformation

Beispiel a)

Es soll im Bildbereich ein Mittelwert von 128 eingestellt werden, indem ein Deltapuls im Ursprung eingestellt wird. In der Definition der Rücktransformation wird eine Skalierung mit $\frac{2}{MN}$ vorgenommen, so daß ein Deltapuls einer Intensität von $X_a = 128 \cdot 64 \cdot 64 = 524288$ für alle Beispiele **a**, **b** und **c** eingestellt werden muss.

Beispiel b)

Das gegebene Deltapulspar bei $(0, 4)$ und $(0, -4)$ entspricht nach Gleichung 1.4 einer Cosinusschwingung. Sie soll den Dynamikbereich von ± 127 um den Mittelwert ausnutzen, so daß sie zusammen mit dem Mittelwert einen Bereich von 1 bis 255 ausfüllt. Der Wert 256 kann dann nicht ausgenutzt werden. Es muss also gelten:

$$127 = \max \left[\frac{2}{N^2} \cdot X \cdot \cos\left(\frac{2\pi(um + vn)}{N}\right) \right]$$

wobei X die Intensität des Deltapuls paares ist. Es ergibt sich eine Intensität von $X_b = \frac{127 \cdot N^2}{2} = 260096$.

Beispiel c)

Hier wird ein weiterer Deltapuls bei $(6, 4)$ und $(-6, -4)$ eingestellt. Da nun zwei Deltapulspaare überlagert werden, ergeben sich zwei Cosinusschwingungen und die Intensität der Schwingung aus Teil **b** muß halbiert werden. Alle vier Deltapulse bekommen also eine Intensität von $X_c = \frac{X_b}{2} = 130048$.

1.2 Übungsaufgaben

1.2.1 Erzeugung eines Spektrums aus Deltapulsen

Es werden quadratische Bilder mit einer Kantenlänge von 64 Pixel untersucht. Im Frequenzbereich ergibt sich eine Matrix der gleichen Größe, allerdings mit komplexen Werten.

Um Delta-Pulse im Frequenzbereich zu erzeugen, muß man das zweidimensionale Signal, also die oben beschriebene Matrix, zunächst mit Nullen initialisieren. In einem nächsten Schritt, werden dann gezielt Delta-Pulse an den gewünschten Positionen eingefügt. Da wir uns im optischen Frequenzbereich bewegen, wird der Ursprung zu Anschaulichkeit in der Mitte der Matrix auf der Position $n_0 = m_0 = \frac{n}{2} + 1 = 33$ festgelegt. Zur Verarbeitung mit der IFFT müßte er wiederum in die obere linke Ecke zurückgeschoben werden.

Um die Zuordnung der Matrizenwerte zu einzelnen Farben festzulegen, wird mit dem Befehl *colormap* eine bestimmte Farbpalette generiert. In allen folgenden Beispielen wird eine Grauwert-Tabelle mit 256 verschiedenen Graustufen (*gray(256)*) verwendet. Je größer der Wert in der Matrix, desto heller ist auch der zugehörige Bildpunkt. Eins entspricht also Schwarz und 256 entspricht Weiß. Diese Zuordnung gilt allerdings nur im Bildbereich. Im Frequenzbereich beschränkt sich der Wertebereich der Matrizenwerte dagegen nicht auf $[1...256]$ weshalb wir hier die Matrizen mit Nullen initialisieren.

```
colormap( gray(256) ); % Farbpalette erzeugen
n = 64; % Seitenbreite
m_null = n/2+1; % Mittelpunkt (opt.
                Koordinaten)
% Frequenzbilder erzeugen (Beispiel a)
iaf = zeros(n,n);
iaf(m_null,m_null)=n*n*128;
% Frequenzbilder erzeugen (Beispiel b)
ibf = iaf;
ibf(m_null , m_null+4) = n*n*63.5;
```



```

ibf(m_null , m_null-4) = n*n*63.5;
% Frequenzbilder erzeugen (Beispiel c)
icf = iaf;
icf(m_null , m_null+4) = n*n*31.75;
icf(m_null , m_null-4) = n*n*31.75;
icf(m_null+6 , m_null+4) = n*n*31.75;
icf(m_null-6 , m_null-4) = n*n*31.75;

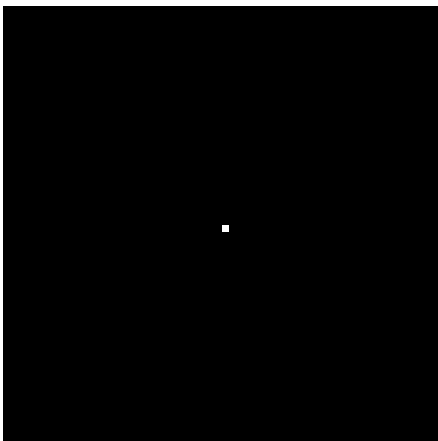
```

Für die bildliche Darstellung solcher Spektren können zwei Befehle genutzt werden. Mit *image* werden die Matrixwerte als Farbpaletteneintrag zwischen 1 und 256 (siehe oben) interpretiert, wobei dieser Wertebereich für eine korrekte Darstellung auch unbedingt eingehalten werden muß. Dabei wird automatisch eine entsprechende Diskretisierung, also in diesem Fall eine Rundung vorgenommen.

Mit *imagesc* dagegen wird automatisch anhand der auftretenden Extremwerte eine Normalisierung des Wertebereiches der Matritzenwerte auf den Bereich der Farbpalette vorgenommen. Dabei geht die Information des absoluten Mittelwertes und der Dynamikbreite verloren. Für Spektren ist dieser Befehl daher der geeignetere, da der Wertebereich der Farbpalette normalerweise nicht eingehalten wird, und sich die verlorengehenden Informationen leicht dazudenken lassen.

Beispiel a)

Abbildung 1.1: Darstellung des Orts- und Frequenzbereiches, Beispiel a)



(a) Frequenzbereich



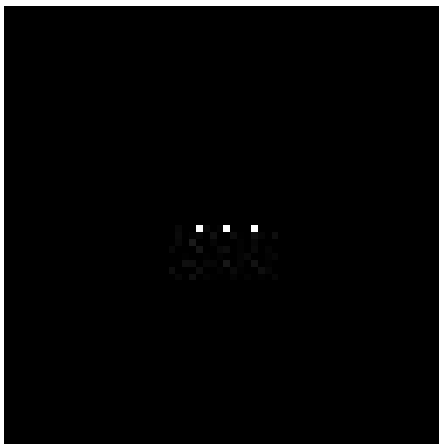
(b) Ortsbereich

Im Bild 1.1a (Frequenzbereich) erkennt man an der großen schwarzen Fläche, daß die Matrix im wesentlichen mit Einsen gefüllt ist. Lediglich im Ursprung (Bildmitte) ist

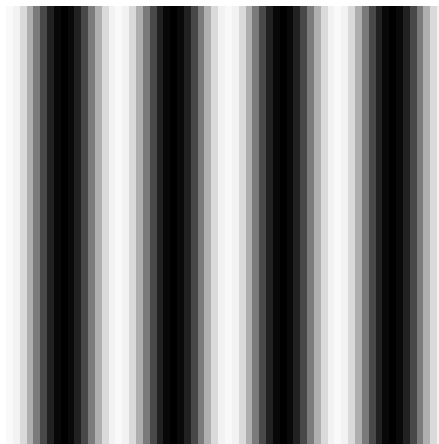
ein Punkt mit abweichendem Grauwert zu erkennen, der den eingestellten Deltapuls darstellt.

Beispiel b)

Abbildung 1.2: Darstellung des Orts- und Frequenzbereiches, Beispiel b)



(a) Frequenzbereich



(b) Ortsbereich

Der Frequenzbereich für Beispiel **b)** (zu sehen in Bild 1.2a) zeigt den Deltapuls aus a), außerdem aber noch ein zusätzliches Deltapulspaar, welches waagrecht um den Ursprung angeordnet ist.

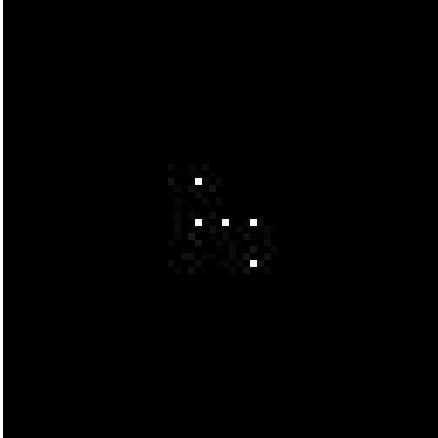
Beispiel c)

In Beispiel **c)** (Bild 1.3a) sind alle Pulse aus a) bzw b) enthalten. Hinzu kommt außerdem ein Deltapulspaar, welches in etwas größerem Abstand diagonal um den Ursprung angeordnet ist.

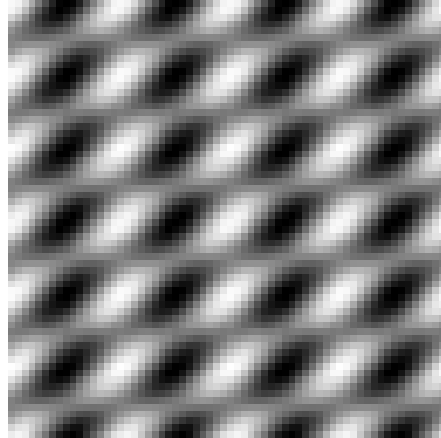
1.2.2 Transformation der Spektren in den Ortsbereich

Vor einer Rücktransformation der in Abschnitt 1.2.1 erzeugten Spektren in den Ortsbereich muß zunächst die dort beschriebene Koordinatentransformation rückgängig gemacht werden. Dabei hilft der Befehl *fftshift*. Er teilt die Matrix (für gerade Kantelängen) in vier gleichgroße Quadranten und tauscht die diagonal gegenüberliegenden Abschnitte aus.

Abbildung 1.3: Darstellung des Orts- und Frequenzbereiches, Beispiel c)



(a) Frequenzbereich



(b) Ortsbereich

Die eigentliche Rücktransformation der Spektren, die nun wieder in der normalen Darstellung vorliegen, erledigt die zweidimensionale inverse Fouriertransformation (*ifft2*). Die IFFT ist in Gleichung 1.2 definiert.

Zur Darstellung kann nun der Befehl *image* genutzt werden, da der Wertebereich im Normalfall eingehalten wird und der korrekte Mittelwert wichtig ist (siehe Erklärung in Abschnitt 1.2.1).

```
% Beispiel a)  
iaf_shift = fftshift(iaf);  
iab = ifft2(iaf_shift);  
image(iab);  
% Beispiel b)  
ibf_shift = fftshift(ibf);  
ibb = ifft2(ibf_shift);  
image(ibb);  
% Beispiel c)  
icf_shift = fftshift(icf);  
icb = ifft2(icf_shift);  
image(icb);
```

Beispiel a)

Das Bild 1.1b zeigt das nach der Rücktransformation des Spektrums entstehende Bild, den Ortsbereich. Zu erkennen ist eine homogene graue Fläche, der eingestellten Helligkeit von 128.

An diesem Beispiel ist sehr schön die Analogie zur eindimensionalen Fouriertransformation zu erkennen. So wie der eindimensionale δ -Puls nach der Fouriertransformation einen konstanten Wert – im Eindimensionalen also eine Linie – ergibt, so resultiert aus dem zweidimensionalen δ -Puls nach der Transformation ebenfalls ein konstanter Wert, der im Zweidimensionalen als Fläche dargestellt wird.

Beispiel b)

Im Ortsbereich des zweiten Beispiels (dargestellt in Bild 1.2b) ist eine Cosinusschwingung in X-Richtung zu erkennen, die genau den Wertebereich von 0 (schwarz) bis 255 (weiß) ausnutzt. Dieser Fall ließe sich wegen den konstanten Werte in Y-Richtung auch auf eine Dimension abbilden. Das entspricht dann dem Transformationspaar

$$\frac{1}{2} [\delta(f - 4) + \delta(f + 4)] \rightarrow \frac{1}{N} \cdot \cos \left(4t \cdot \frac{2\pi}{N} \right)$$

Die komplette von m und u unabhängige Gleichung für das Beispiel sieht so aus:

$$X_a \delta(u, v) + X_b \delta(u, v - 4) + X_b \delta(u, v + 4) \rightarrow \frac{1}{N^2} \cdot \left[X_a + 2 \cdot X_b \cdot \cos \left(4n \cdot \frac{2\pi}{N} \right) \right]$$

Beispiel c)

Bild 1.3b schließlich stellt die Rücktransformation des dritten Spektrums dar. Die Schwingung aus Beispiel **b)** ist noch enthalten, jedoch in der Intensität halbiert. Zusätzlich kommt eine Schwingung in diagonaler Richtung hinzu, wobei nach der Mischung die gleichmäßigen Wellenfronten durch die zweite Welle ebenfalls gewellt sind.

Auch diese Schwingung ließe sich auf eine Dimension abbilden, jedoch müßte vorher eine Drehung des Koordinatensystems vorgenommen werden. Das komplette Transformationspaar wäre:

$$X_a \delta(u, v) + X_c [\delta(u, v - 4) + \delta(u, v + 4) + \delta(u - 6, v - 4) + \delta(u + 6, v + 4)]$$

$$\rightarrow \frac{1}{N^2} \cdot \left[X_a + 2 \cdot X_c \cdot \left(\cos \left(\frac{4n \cdot 2\pi}{N} \right) + \cos \left(\frac{(6m + 4n) \cdot 2\pi}{N} \right) \right) \right]$$

1.2.3 Manipulationen von Signalen im Ortsbereich

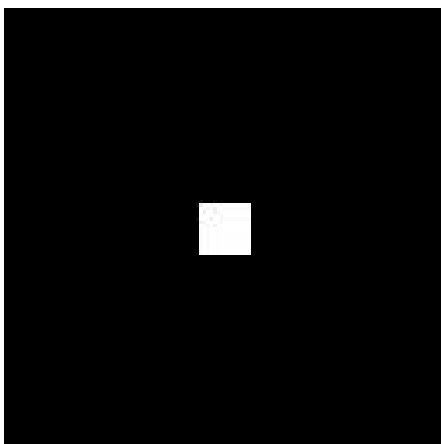
Erzeugung eines Quadrates

Es wird ein Bild erzeugt, welches ein weisses Quadrat vor schwarzem Hintergrund zeigt. Dieses Bild wird wie gehabt durch eine Matrix dargestellt. Da wir uns hier im Bildbereich befinden, muß diese Matrix die Farbwerte zwischen 1 und 256 gemäß der 256-stufigen Graupalette enthalten. Nachdem man mittels *ones* eine entsprechend große Matrix von schwarzen Bildpunkten (Schwarz entspricht einem Wert von 1) erzeugt hat, kann man weiße Bereiche einfach dadurch erzeugen, daß man die Matrixwerte der entsprechenden Pixel gezielt auf 256 (entspricht Weiß) setzt.

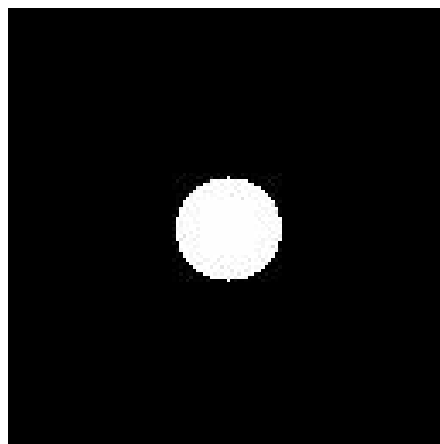
Das Bild hat eine Größe von 128×128 Pixeln. Das weiße Quadrat liegt in der Mitte des Bildes und hat eine Kantenlänge von 15 Pixeln. Der Bildmittelpunkt liegt (bezüglich des optischen Koordinatensystems) bei $(64.5, 64.5)$, da das nullte Pixel fehlt. Da das Quadrat jedoch einen relativen Mittelpunkt bei $(8, 8)$ hat, kann es nicht exakt in der Mitte liegen, sondern z.B. bei $(57, 57)$ bis $(71, 71)$. Dann liegt der absolute Mittelpunkt des Quadrates bei $(64, 64)$.

```
n2 = 128
n2_null = n2/2+1
% Bildbereich (Quadrat)
sab = ones(n2, n2);
sab(n2_null-7:n2_null+7 , n2_null-7:n2_null+7) = 256;
image(sab);
```

Abbildung 1.4: Direkt erzeugte Ortsbereiche



(a) Quadrat



(b) Kreis

Das Ergebnis dieser Zeilen ist in Bild 1.4a zu sehen.

Erzeugung eines Kreises

Nun wird (exakt wie im vorhergehenden Abschnitt) ein schwarzes Bild erzeugt. In die Mitte soll nun aber ein weißer Kreis (Radius 15 Pixel) gesetzt werden. Die weißen Punkte des Kreises (Farbwert: 256) lassen sich am einfachsten mit zwei geschachtelten **for**-Schleifen zuweisen.

```
% Bildbereich (Kreis)  
cab = ones(n2, n2);  
for x = n2_null-15:n2_null+15  
    y_delta=round(sqrt(15*15-(x-n2_null)*(x-n2_null)));  
    for y = n2_null-y_delta:n2_null+y_delta;  
        cab(x,y) = 256;  
    end;  
end;  
image(cab);
```

Das Ergebnis ist in Bild 1.4b zu sehen.

1.2.4 Transformation der Bilder in den Frequenzbereich

Die in Abschnitt 1.2.3 erstellten Bilder werden nun mit der zweidimensionalen Fouriertransformation (*fft2*) in den Spektralbereich übertragen. Die entstehende Matrix ist wie oben erwähnt, komplex und enthält in der oberen linken Ecke den Ursprung, also den Mittelwert. Er wird mit der Funktion *fftshift* in die Mitte verlegt (optisches Koordinatensystem), so dass die Symetrieigenschaften um den Ursprung deutlich werden.

Quadrat im Frequenzbereich

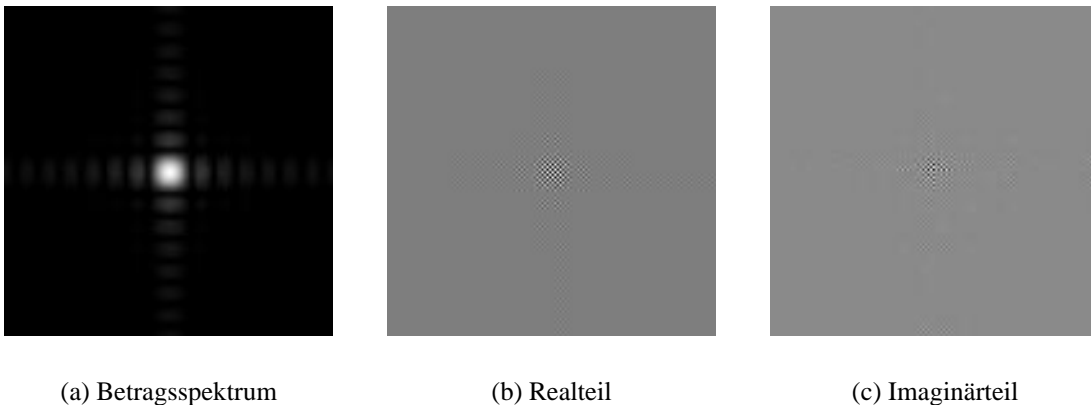
```
% Rücktrafo  
saf = fft2(sab);  
saf_shift = fftshift(saf);  
% Betragsspektrum (Quadrat)  
imagesc(abs(saf_shift));  
min(min(abs(saf_shift)))  
max(max(abs(saf_shift)))  
% Imaginärteil (Quadrat)  
imagesc(imag(saf_shift));  
min(min(imag(saf_shift)))
```

```

max(max(imag(saf_shift)))
% Realteil (Quadrat)
imagesc(real(saf_shift));
min(min(real(saf_shift)))
max(max(real(saf_shift)))

```

Abbildung 1.5: Quadrat im Frequenzbereich



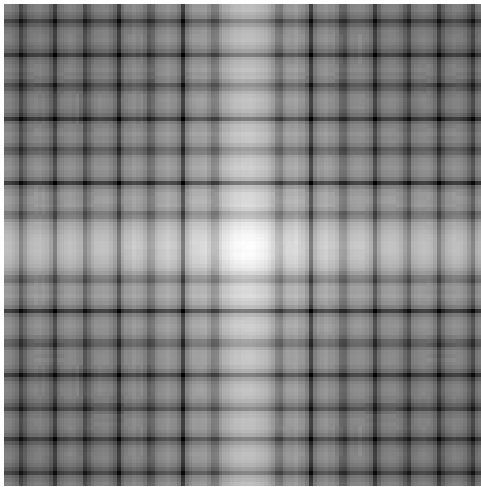
Das Betragsspektrum (Bild 1.5a) ist im wesentlichen schwarz, nur die Bereiche, wo das Quadrat war, sowie die horizontal und vertikal daneben liegenden Bereiche sind mit einer vom Ursprung aus abnehmenden Welligkeit versehen.

Zur Interpretation kann man das Ergebnis mit der Fouriertransformierten einer Rechteckfunktion vergleichen. Es ergibt sich eine Si-Funktion, deren Frequenz von der Breite des Rechtecks abhängt. Je breiter das Rechteck ist, desto schmaler (höherfrequenter) ist die Si-Funktion. Jeder horizontale oder vertikale eindimensionale Ausschnitt aus dem Bild enthält entweder genau so eine Rechteckfunktion oder ist gleich Null. Das Ergebnis ist dann eine Aneinanderreihung und Mischung vieler Si-Funktionen. Die oben erwähnten Symmetrieeigenschaften gelten auch hier, das Spektralbild hat eine mittlere horizontale und vertikale Spiegelachse.

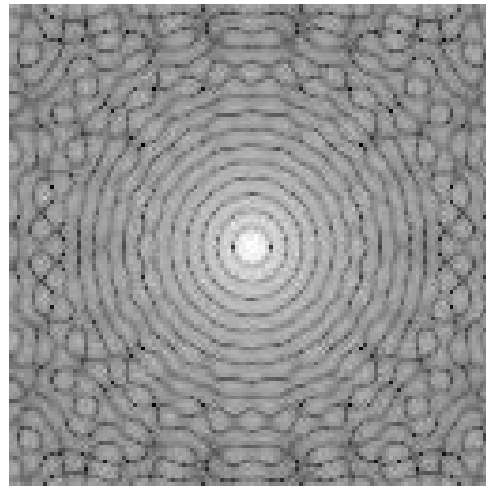
In obigem Code werden außerdem die Extremwerte der einzelnen Spektren, jeweils aufgeschlüsselt nach Real- und Imaginärteil, berechnet. Es zeigt sich, daß der Imaginärteil der Spektren (bis auf Rundungsfehler) Null ist. Im Bild 1.5c ist der Imaginärteil des Spektrums dargestellt. Hier sind zwar Werte ungleich Null zu erkennen, das liegt jedoch daran, daß diese Spektren mit *imagesc* gezeichnet wurden und so die äußerst geringen Rundungsfehler im Bereich von 10^{-14} auf den kompletten Helligkeitsbereich hochgerechnet wurden.

Die Spektren sind also rein reell was impliziert, daß sich die Signale vollständig durch Cosinus-Funktionen beschreiben lassen und es sich daher um *gerade* Funktionen (bezüglich des Bildmittelpunktes) handelt.

Abbildung 1.6: Betragsspektren (Logarithmisch)



(a) Quadrat



(b) Kreis

Ein logarithmisch aufgetragenes Betragsspektrum des Quadrates (Bild 1.6a) wird mit folgender Zeile erzeugt:

```
imagesc(log(abs(saf_shift)));
```

Man erkennt deutlich, daß gegenüber dem normalen Spektrum (Bild 1.61.5a) hier auch in den Bereichen mit schwachen Komponenten deutliche Strukturen auszumachen sind.

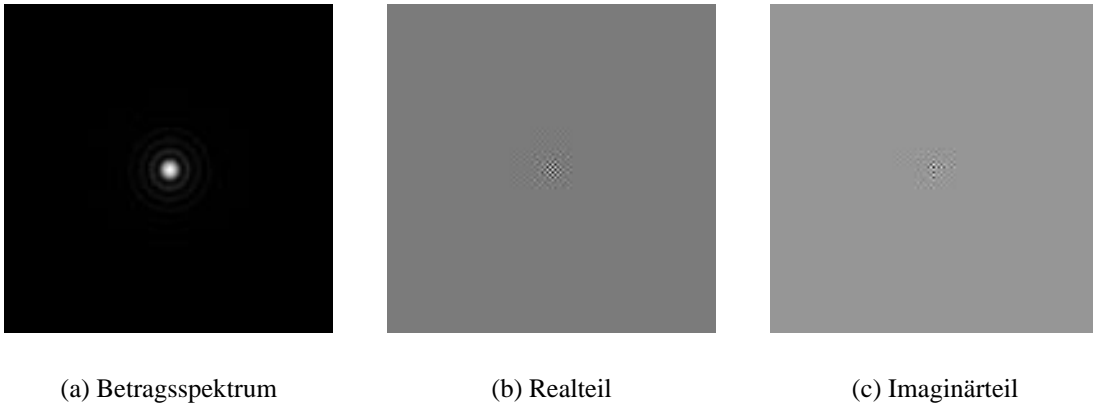
Kreis im Frequenzbereich

Für den Kreis ergibt sich ein ähnliches Spektrum (Bild 1.7a), jedoch ist das Ursprungsbild (Ortsbereich) als Kreis natürlich punktsymmetrisch, was auch ein punktsymmetrisches Spektrum zur Folge hat.

Was die Extremwerte des Imaginärteils betrifft, so gilt hier das oben für das Quadrat gesagte ebenfalls. Der Imaginärteil ist (im Rahmen der Rechengenauigkeit) Null und das Signal ist somit gerade zum Bildmittelpunkt.

```
% Rücktrafo  
caf = fft2(cab);  
caf_shift = fftshift(caf);  
% Betragsspektrum (Kreis)  
imagesc(abs(caf_shift));
```


Abbildung 1.7: Kreis im Frequenzbereich



```
min(min(abs(caf_shift)))  
max(max(abs(caf_shift)))  
% Imaginärteil (Kreis)  
imagesc(imag(caf_shift));  
min(min(imag(caf_shift)))  
max(max(imag(caf_shift)))  
% Realteil (Kreis)  
imagesc(real(caf_shift));  
min(min(real(caf_shift)))  
max(max(real(caf_shift)))
```

Eine lokale Verschiebung des Bildinhaltes (Ortsbereich) wirkt sich nicht auf das Betragsspektrum, sondern nur auf die Phasenlage der Daten im Frequenzbereich aus.

Eine logarithmische Darstellung des Betragsspektrums ist in Bild 1.6b gegeben. Bezüglich der Analyse gilt hier das selbe wie für das Quadrat. Die schwachen Strukturen erscheinen deutlich hervorgehoben.

Kapitel 2

Übungsblatt 8 – Bildsignalverarbeitung II

2.1 Vorbereitungsaufgaben

2.1.1 Skalierungseigenschaften der zweidimensionalen FFT

Wendet man auf ein Bildsignal die zweidimensionale FFT und gleich anschliessend die Rücktransformation an, so erhält man das Ursprungssignal identisch zurück. Wenn ein Signal im Frequenzbereich manipuliert werden soll, kann dies durch elementweise Multiplikation der Frequenzmatrix mit einer Filtermatrix geschehen. Ist diese in allen Elementen gleich 1, wird keine Änderung durchgeführt. Diese Filtermatrix mit dem Mittelwert 1 wäre dann im Ortsbereich ein Einheitspuls der Intensität 1 im Ursprung. Mit Hilfe der Faltungsregel ergibt sich mathematisch ausgedrückt:

$$IFFT(FFT(x)) = x$$

$$IFFT(FFT(x) \cdot FFT(y)) = x * y$$

$$IFFT(FFT(x) \cdot 1) = x * \delta(n)$$

Wendet man die Faltungsgesetze rückwärts an, ergibt sich im Prinzip derselbe Zusammenhang, jedoch muß die elementweise Multiplikation mit der Inverstransformierten eines Deltapulses der Stärke MN geschehen.

$$FFT(IFFT(X)) = X$$

$$FFT(IFFT(X) \cdot IFFT(Y)) = X * Y$$

$$FFT(IFFT(X) \cdot 1) = X * (MN \cdot \delta(m))$$

2.1.2 Spektrum eines verschobenem Deltapulses

Das Spektrum eines einzelnen realen Deltapulses $\delta_{m,n}$ ergibt eine periodische Cosinusschwingung, gemischt mit einer imaginären Sinusschwingung, wobei sich die Frequenzen in X- und Y-Richtung aus der genauen Position des Deltapulses ergeben.

$$X(u, v) = e^{-j2\pi \frac{um+vn}{N}} = \cos\left(2\pi \frac{um+vn}{N}\right) - j \cdot \sin\left(2\pi \frac{um+vn}{N}\right)$$

Wird das Spektrum von x mit dem Spektrum dieses Deltapulses multipliziert und rücktransformiert, entspricht das der Faltung des Signals x mit dem verschobenen Deltapuls und das Ergebnis ist eine Verschiebung des Originalbildes um (m, n) , wobei man sich das Bild periodisch denken muß. Verschwindende Bildteile tauchen auf der anderen Seite wieder auf, so daß keine Informationen verloren gehen.

2.1.3 Zweidimensionales Filter

Ein eindimensionales Filter hat eine Impulsantwort h . Signale die man filtern möchte, werden mit dieser Impulsantwort gefaltet, bzw. mit der Fouriertransformierten der Impulsantwort, dem Frequenzgang H multipliziert. Ein Filter mit einem rechteckigen Frequenzgang hat beispielsweise also eine Si-förmige Impulsantwort.

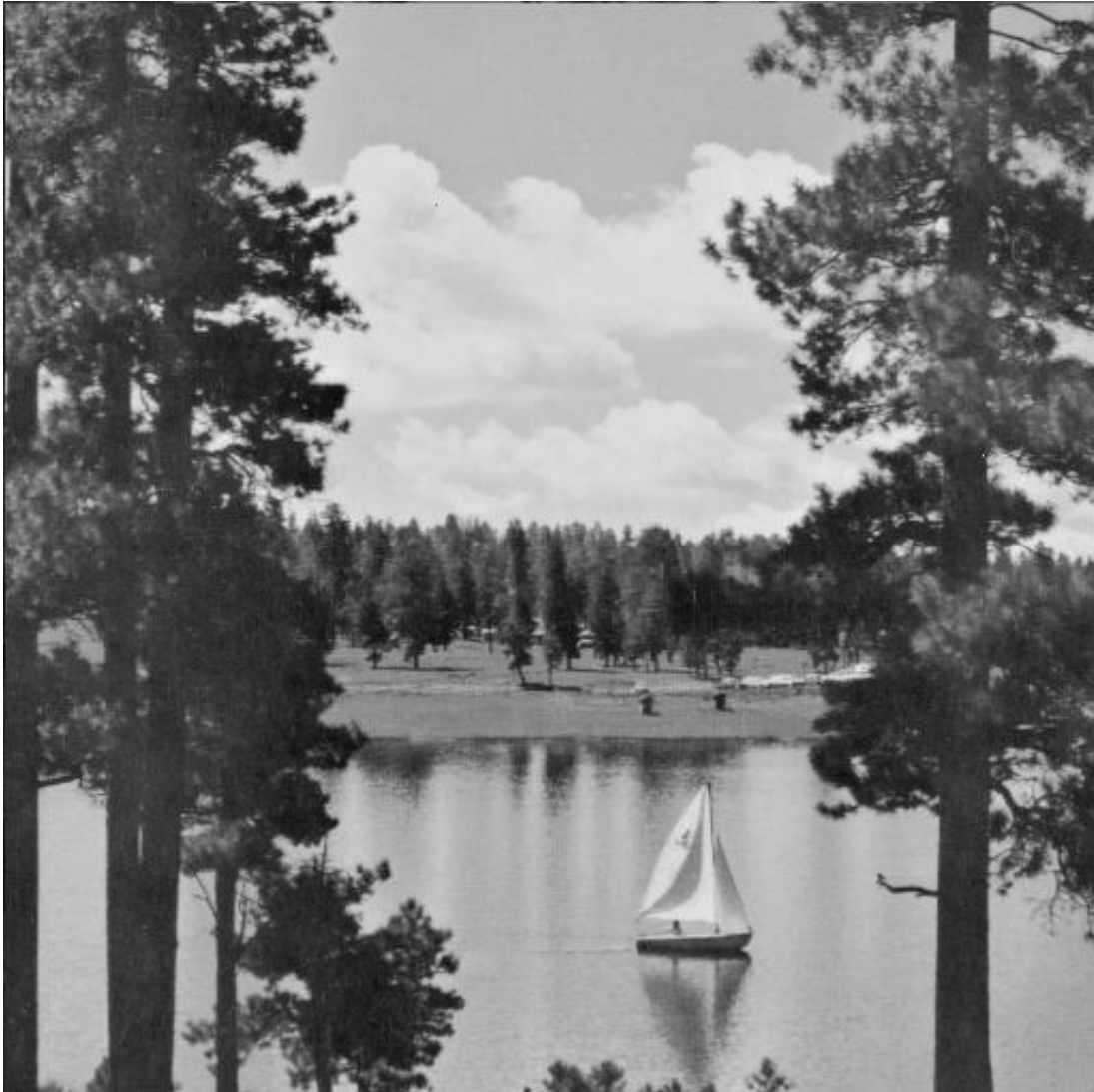
Erweitert man diese Vorstellung jetzt auf den zweidimensionalen Fall, indem beide Koordinaten auf ihren Abstand zum Ursprung abgebildet werden, so ergibt sich für ein zweidimensionales Rechteckfilter im Idealfall als Impulsantwort ein Signal wie das in Bild 1.7a. Ein heller, runder Bereich mit schnell abfallender Intensität (entspricht dem Hauptmaximum der eindimensionalen Si-Funktion) in der Mitte des Bildes und konzentrisch darum herum angeordnete Wellen mit langsam abnehmender Amplitude, die den Nebenmaximas der normalen Si-Funktion entsprechen.

$$r_{1dim} = \sqrt{m^2 + n^2}$$

$$h(m, n) = \frac{\Omega}{\pi} Si(\Omega \cdot \sqrt{m^2 + n^2})$$

Alternativ kann auch statt der Faltung mit der Impulsantwort auch direkt im Frequenzbereich mit dem gewünschten Frequenzgang multipliziert werden.

Abbildung 2.1: Datei *'lake.by'*



2.2 Übungsaufgaben

2.2.1 Bildausschnittverschiebung

Bild einlesen

Zunächst wird das Bildsignal 'Lake' geladen und dargestellt. Es handelt sich hierbei um ein 256-Graustufen-Bild mit 512×512 Pixeln. Zunächst wird dafür mit *fopen* ein File-Descriptor für die entsprechende Datei erzeugt und über diesen dann mittels *fread* der Inhalt in eine Matrix geladen. Dabei muß jedes Element der Matrix um 1 inkrementiert werden, um die unterschiedlichen Wertebereiche anzupassen. In der Datei liegen die Werte im Bereich von 0 bis 255, während Octave eine Grauwert-Palette mit Werten zwischen 1 und 256 benutzt (siehe Erklärung in Abschnitt 1.2.1). Das importierte Bild ist in Abbildung 2.1 dargestellt.

```
% see laden
seeFile = fopen("lake.by", "r");
see = fread( seeFile , [512,512] , "uchar" )' + 1;
image(see);
```

Bildausschnitt kopieren und verschieben

Nun wurden durch Probieren die Koordinaten eines Quadrats mit 128×128 Pixeln um das Segelboot bestimmt. Die obere linke Ecke dieses Quadrates liegt bei (350, 270), die rechte untere dementsprechend bei (350 + 127, 270 + 127)liegen.

```
% Boot ausschneiden
boot = see(350:(350+127) , 270:(270+127));
image(boot);
```

Dieser Bildausschnitt wird wie in Abschnitt 2.1.1 beschrieben transformiert, mit einer Transformierten eines Deltapulses elementweise multipliziert, und Rücktransformiert. Dabei wird das Segelboot um ein viertel seiner Größe nach rechts unten geschoben. Der Deltapuls muss dafür die Intensität 1 haben und bei $(\frac{128}{4}, \frac{128}{4})$ liegen.

```
% Bildausschnitt transformieren
boot_fb = fftshift( fft2(boot) );
% 2D-Deltapuls für Bildverschiebung erzeugen
delta = zeros(128, 128);
delta(128/4+1, 128/4+1) = 1;
delta_fb = fftshift( fft2(delta) );
```

```

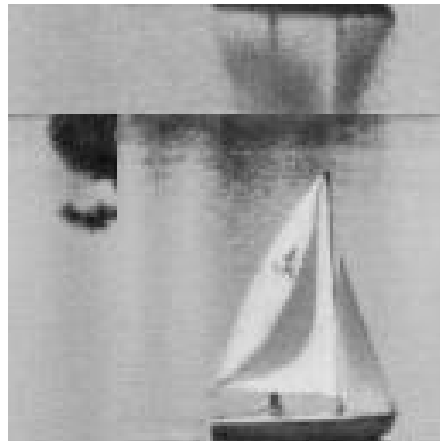
% im Frequenzbereich Multiplizieren (Ortsbereich => Faltung)
vboot_fb = boot_fb .* delta_fb;
vboot = ifft2( fftshift(vboot_fb));
image(vboot);

```

Abbildung 2.2: Segelboot Bildausschnitt



(a) Original-Ausschnitt



(b) Um 32 Pixel verschoben

Den originalen, sowie den verschobenen Bildausschnitt kann man in Bild 2.2 sehen.

2.2.2 Bild-Tiefpassfilterung

Bild einlesen

Als erstes wird – analog zur vorhergehenden Aufgabe – das Bildsignal 'Lena128' geladen, der Farbraum konvertiert und das Bild dargestellt. Es handelt sich wieder um ein 256-Graustufen-Bild, diesmal mit 128×128 Pixeln. Das Ergebnis – eine wunderschöne Frau – ist in Bild 2.3a dargestellt.

```

% lena (128x128 pixel)
lena128File = fopen("lena128.bytest", "r");
lena128 = fread(lena128File, [128,128], "uchar")' + 1;
image(lena128);
% Betragsspektrum
imagesc(fftshift(log(abs(fft2(lena128)))));

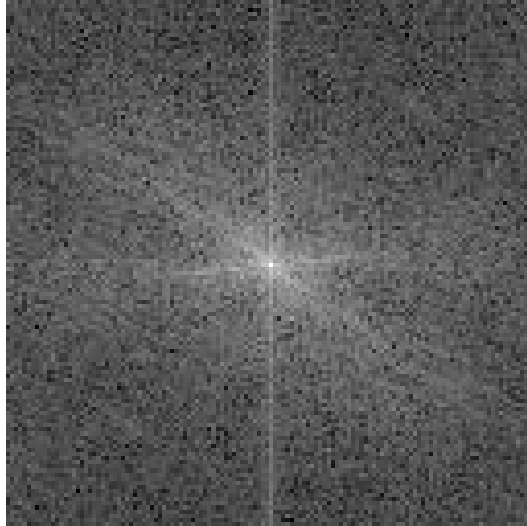
```

Das logarithmische Betragsspektrum ist in Bild 2.3b zu sehen. Das Bild hat Komponenten in nahezu allen Frequenzen.

Abbildung 2.3: Datei 'lena128.byf'



(a) Ortsbereich



(b) Logarithmiertes Betragsspektrum

Zweidimensionale Tiefpaßfilterung

Nun wird ein Frequenzgang zur Tiefpassfilterung generiert. Dafür kann die Octave-Funktion aus Abschnitt 1.2.3 genutzt werden, die eine Matrix kreisförmig mit Werten füllt. Die gewünschte Grenzfrequenz ist $\Omega = \frac{\pi}{4}$, was einer Periodendauer von $T = 8$ entspricht. Bezogen auf die FFT-Blockgröße ergibt das eine Grenzfrequenz mit dem Radius von $\frac{128}{8} = 16$ Pixeln.

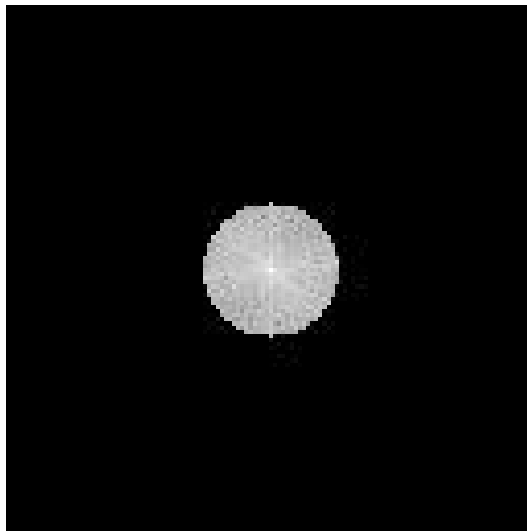
```
%Erstellung des Tiefpass-Frequenzganges
cab = zeros(128, 128);
rd=128/8;
for x = 65-rd:65+rd
    y_delta=round(sqrt(rd*rd-(x-65)*(x-65)));
    for y = 65-y_delta:65+y_delta;
        cab(x,y) = 1;
    end;
end;
%Die Filterung geschieht im Frequenzbereich, also
%Multiplikation der Spektren
lena_tp=ifft2(fft2(lena128).*fftshift(cab));
% Betragsspektren (logarithmisch)
imagesc(fftshift(log(abs(fft2(real(lena_tp)))+0.001)));
```

Die eigentliche Filterung geschieht durch elementweise Multiplikation des Spektrums des Originalbildes mit der Kreismatrix. Das entstehende Spektrum hat nach Rücktransformation keinen Imaginärteil. Im gefilterten Bild (Abbildung 2.4a) ist im Wesentlichen das Originalbild zu sehen, jedoch ist es stark weichgezeichnet, die Konturen sind stark verwaschen. Auffällig dabei sind unerwartete Flecken ähnlicher Helligkeit, die offensichtlich von den zahlreichen Nullstellen in der Impulsantwort herrühren. Das liesse sich evtl. durch Anwendung einer Fensterfunktion vermeiden.

Abbildung 2.4: Datei 'lena128.by' tiefpaßgefiltert



(a) Ortsbereich (Weichgezeichnet)



(b) Logarithmiertes Betragsspektrum

Im Spektrum (Bild 2.4b) ist deutlich die Wirkung des Filters zu sehen, die hohen Frequenzanteile, die am Rande des Spektralbildes liegen müssten, sind nach der Filterung nicht mehr vorhanden.

2.2.3 Histogramm von Bildsignalen

Die Bilder des Sees (Bild 2.1) und des Segelbootes (Bild 2.2a) aus den vorhergehenden Aufgaben sollen auf ihre relativen Häufigkeitsverteilungen der Farbwerte von 1 bis 256 untersucht werden.

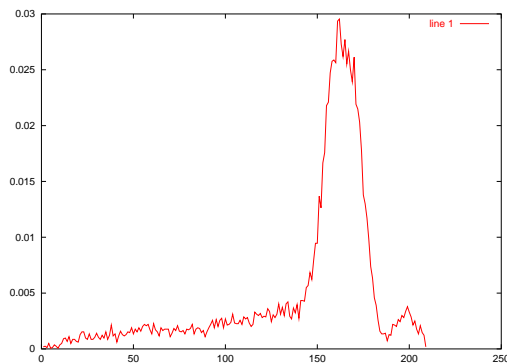
Die Darstellung der Helligkeitsverteilung (Histogramm genannt) ist im Prinzip nichts anderes als die Amplitudendichteverteilung, welche sich mittels *empirical_pdf* leicht berechnen läßt. Dazu ist die örtliche Struktur der Daten unwichtig, also werden die Werte aus der Bildmatrix vorher mit *reshape* in einen eindimensionalen Vektor umverteilt, von dem dann die Amplitudendichteverteilung wie gehabt berechnet wird.


```

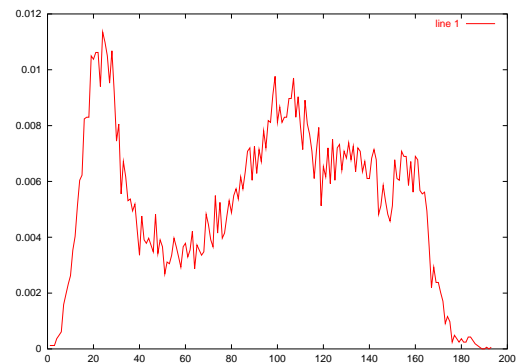
% Boot
r_boot = reshape(boot, 1, 128*128);
hist_boot = empirical_pdf(min(r_boot):max(r_boot), r_boot);
plot(hist_boot);
% Lena
r_lena = reshape(lena128, 1, 128*128);
hist_lena = empirical_pdf(min(r_lena):max(r_lena), r_lena);
plot(hist_lena);

```

Abbildung 2.5: Histogramme



(a) Bootschnitt



(b) Lena128

Die Histogramme der Bilder sind beide in Abbildung 2.5 dargestellt. Die Unterschiede bezüglich Kontrast und Helligkeitsbereich sind hier quantitativ abzulesen.

Offensichtlich ist in Bild 2.5a (Segelboot) ein bestimmter Amplitudenwert relativ häufig vertreten. Aller Wahrscheinlichkeit nach handelt es sich dabei um das Grau der Wasseroberfläche. In Bild 2.5b (Lena) sind die Amplitudenwerte (und damit die Häufigkeit der Grautöne) wesentlich gleichmäßiger verteilt.

Abbildungsverzeichnis

1.1	Darstellung des Orts- und Frequenzbereiches, Beispiel a)	5
1.2	Darstellung des Orts- und Frequenzbereiches, Beispiel b)	6
1.3	Darstellung des Orts- und Frequenzbereiches, Beispiel c)	7
1.4	Direkt erzeugte Ortsbereiche	9
1.5	Quadrat im Frequenzbereich	11
1.6	Betragsspektren (Logarithmisch)	12
1.7	Kreis im Frequenzbereich	13
2.1	Datei ' <i>lake.by</i> '	16
2.2	Segelboot Bildausschnitt	18
2.3	Datei ' <i>lena128.by</i> '	19
2.4	Datei ' <i>lena128.by</i> ' tiefpaßgefiltert	20
2.5	Histogramme	21