LU GDBV Abgabe 1
Gruppe 25

Peter Holzkorn	Andreas Bretschneider	Martin Tintel
0426262	0327444	0402913
p.holzkorn@gmx.at	bretschneider@gmx.at	mtintel@gmx.at

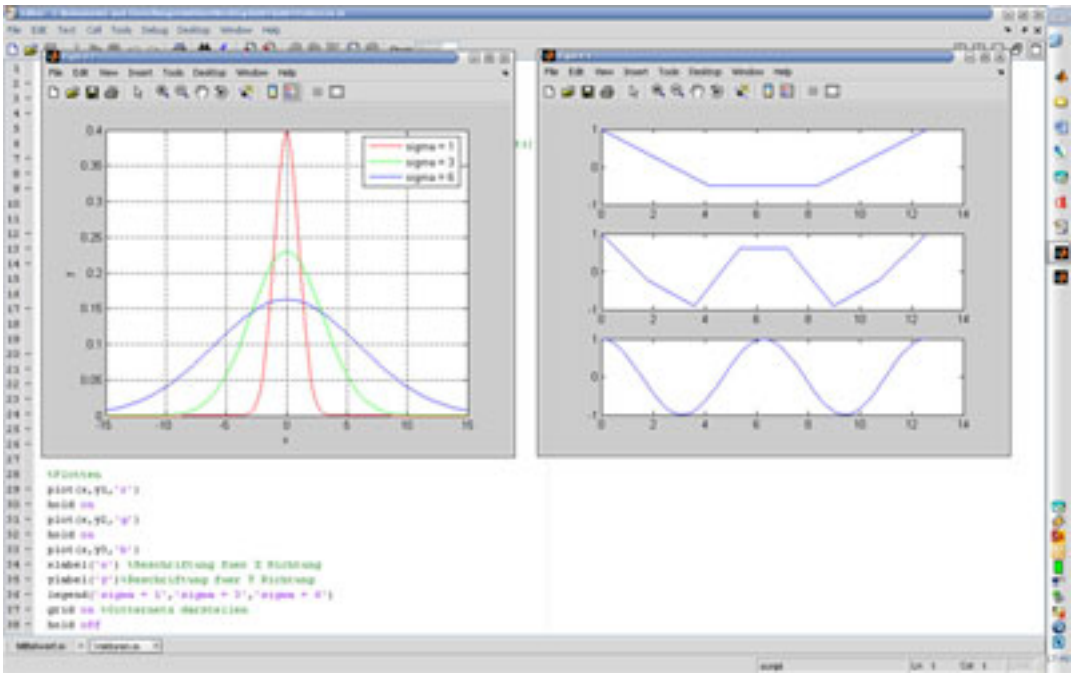
26. November 2006

1 Einführung

1.1 Vektoren

```
%Erzeugen der Vektoren mit 4, 8 und 128 Elementen

t1 = [0:4/3:4].*pi
t2 = [0:4/7:4].*pi
t3 = [0:4/127:4].*pi

%Erzeugen Sie einen Vektor y der gleichen Groesse wie t, mit yi = cos(ti)

y1 = cos(t1(1:end))
y2 = cos(t2(1:end))
y3 = cos(t3(1:end))

%Plotten

subplot(3,1,1); plot(t1,y1)
subplot(3,1,2); plot(t2,y2)
subplot(3,1,3); plot(t3,y3)

%neues Fenster erstellen

figure

%Erzeugen

x = [-15:0.2:15]
sigma = 1
y1 = gauss1d(x, sigma)
sigma = 3
y2 = gauss1d(x, sigma)
sigma = 6
y3 = gauss1d(x, sigma)

%Plotten

plot(x,y1,'r')
hold on
plot(x,y2,'g')
hold on
plot(x,y3,'b')
```

```
xlabel('x') %Beschriftung fuer X Richtung

ylabel('y') %Beschriftung fuer Y Richtung

legend('sigma = 1','sigma = 3','sigma = 6')
grid on %Gitternetz darstellen
hold off
```



1.1.1 teil 1

(Abb. oben: rechts) Die Vektoren repräsentieren die Funktionswerte einer Sinusfunktion, die im Intervall $[0; 4\pi]$ 4mal bzw. 8mal bzw. 128mal abgetastet wird.

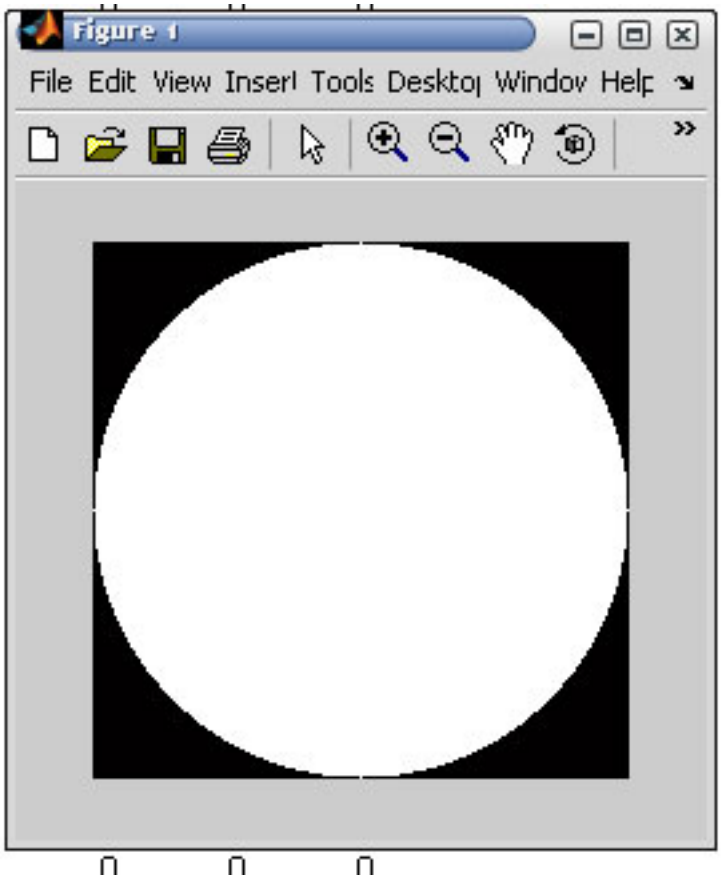
1.1.2 teil 2

(Abb. oben: links) Die Vektoren repräsentieren eindimiensionale Gauss-Kurven, die sich nur in der Standardabweichung σ unterscheiden.

1.2 Matrizen

`Matrizen.m`

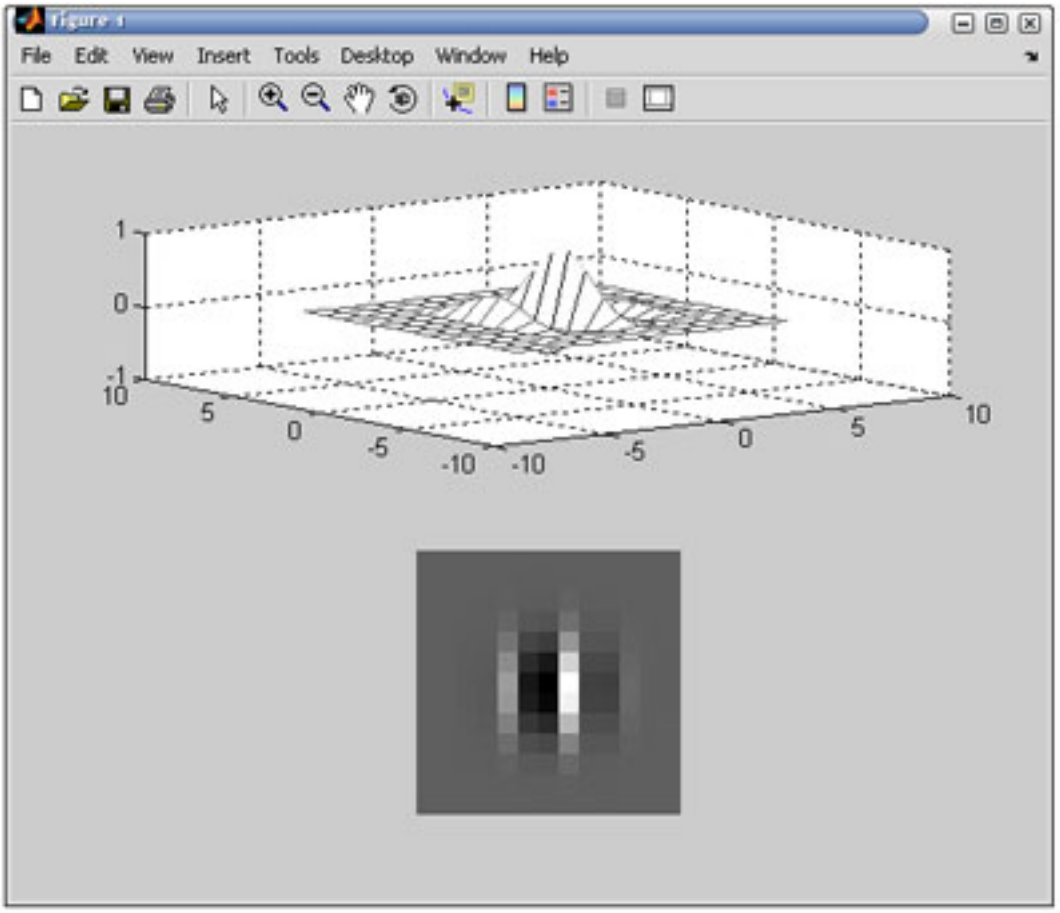
```
[x y] = meshgrid(-2.*pi:2.*pi)

Z = exp(-(x.^2 + y.^2) ./ (2.*(pi./2).^2)) .* sin(2.*x)

subplot(2,1,1); mesh(x,y,Z)

subplot(2,1,2); imshow(Z,[])



C = circle(100);

figure

imshow(C)


Circle.m

function circ = circle(rad)

dist = zeros(2 .* rad + 1, 2 .* rad + 1);
middle = rad + 1;
tend = 2.*rad+1;
total = tend.*tend;
for n=1:tend
    dist(n,:) = sqrt(((1:tend)-middle).^2 + (n-middle).^2);
end

c = ones(tend, tend);

for n=1:total
    if dist(n) > rad
        c(n) = 0;
    end

end

circ = c;
```



1.2.1 Circle

Die Funktion circle.m generiert zunächst die Distanzmatrix, indem sie die Distanzen der einzelnen Elemente einer $n \times n$ Matrix (mit n = Breite bzw. Höhe des gewünschten Bildes in Pixeln) zu ihrem Mittelpunkt berechnet. Dazu wird über die Zeilen mit einer FOR-Schleife iteriert und für jede Zeile eine Vektorrepräsentation errechnet. Die jeweilige Distanz eines Elements zum Mittelpunkt wird berechnet, indem die vertikale und horizontale Entfernung (in Elementen) als Vektorkoordinaten x und y betrachtet werden - die absolute Entfernung erhält man dann mit der Formel zur Berechnung der Länge eines Vektors aus seinen Koordinaten: $\sqrt{x^2 + y^2}$. Um schließlich den eigentlichen Kreis darzustellen, wird der Wert jedes Elements aus der Distanzmatrix mit dem Kreisradius verglichen. Ist die Distanz größer als der Radius, wird der Pixel, der dem Element entspricht, schwarz eingefärbt, andernfalls weiß.

1.2.2 Funktion



$Z = \exp(-\frac{x^2+y^2}{2\sigma^2}) \cdot \sin(2x)$

1.3 Mittelwert und Standardabweichung

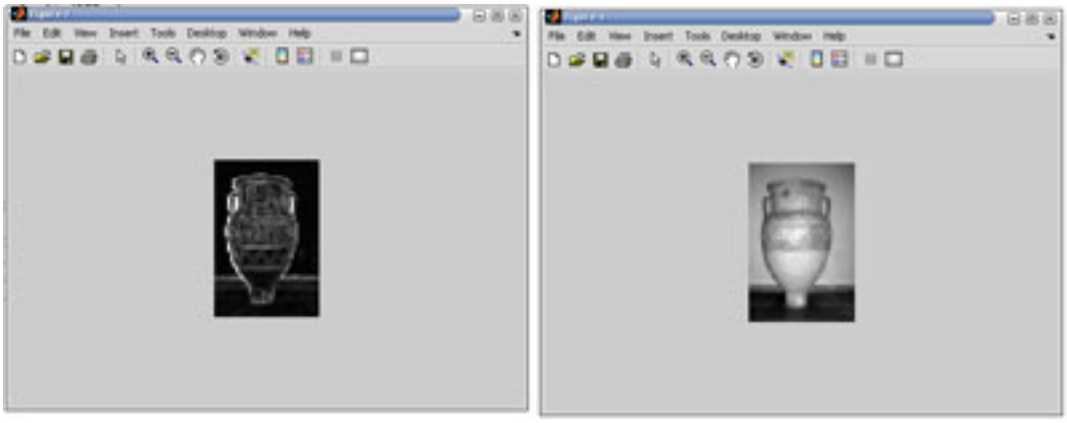
```
chaplin = double(imread('chaplin.tif'));
y1 = chaplin(2,:);
y2 = chaplin(200,:);

m1 = mean(y1)
s1 = std(y1)
m2 = mean(y2)
s2 = std(y2)
```

1.3.1 teil 1

Ergebnisse: Zeile 2 (Hut und Hintergrund) $m1 = 57.6797$ $s1 = 30.1462$ Zeile 200 (Gesicht und Hintergrund) $m2 = 81.5742$ $s2 = 44.8647$ Der zweite Mittelwert ist aufgrund der hellen Farbwerte im Gesicht höher, die zweite Standardabweichung aufgrund des stärkeren Kontrastes zum Hintergrund.

1.3.2 teil 2

```
c = double(imread('ceramic.tif'));
% b = im2col(c, [8,8], 'distinct');
fun = @mean2;
b = blkproc(c, [8,8], fun);
morefun = @std2;
s = blkproc(c, [8,8], morefun);
imshow(b,[])
figure
imshow(s,[])
```



Die Abbildung der Mittelwerte der Blöcke führt zu einer simplen Verkleinerung und Glättung des Bildes (ähnlich der Mittelwert-Faltung). Die Abbildung der Standardabweichungen führt zu einer Verkleinerung und einer Hervorhebung der Kanten, da scharfe Übergänge zwischen Grauwertbereichen stark variierender Intensität eine Hohe Standardabweichung erzeugen.

2 Bildrauschen und Bildglättung

2.1 Mittelwert und Gauss

```
mw3x3 = fspecial('average',3);
mw7x7 = fspecial('average',7);
g3x3 = fspecial('gaussian',3,0.5);
g6x6 = fspecial('gaussian',6,1);

image = imread('lena.tif');

c1 = conv2(image, mw3x3);
c2 = conv2(image, mw7x7);
c3 = conv2(image, g3x3);
c4 = conv2(image, g6x6);
figure
subplot(3,2,1); imshow(image,[]);
subplot(3,2,3); imshow(c1,[])
subplot(3,2,4); imshow(c2,[])
subplot(3,2,5); imshow(c3,[])
subplot(3,2,6); imshow(c4,[])
```

Die Mittelwertfaltungen lassen die Konturen stark verschwimmen, besonders bei größeren Masken (siehe Abb., 2. Reihe). Die Gauss-Faltungen hingegen entfernen Störungen, erzeugen aber bei ähnlichen Maskengrößen wesentlich weniger Unschärfe, weil starke Grauwertschwankungen innerhalb kleiner Bereiche durch die Gewichtung der Gausskurve besser erhalten bleiben.

2.2 Salz und Pfeffer Beseitigung

```
image2 = imnoise(image, 'salt & pepper', 0.05);
image3 = imnoise(image, 'salt & pepper', 0.1);
image4 = imnoise(image, 'salt & pepper', 0.2);

c5 = conv2(image2, mw3x3);
c6 = conv2(image2, mw7x7);
c7 = conv2(image2, g3x3);
c8 = conv2(image2, g6x6);
cMed = medfilt2(image2);
```

```
figure
subplot(1,3,1); imshow(image2,[]);
subplot(1,3,2); imshow(image3,[]);
subplot(1,3,3); imshow(image4,[]);
```

Mit den Mittelwertfaltungen wird das Bild wieder nur unschärfer gemacht. Die Gaussfaltung produziert weniger verschwommene Bilder, kann die Störung aber auch nicht beseitigen, sondern verteilt sie nur stärker und schwächt sie etwas ab. Der Medianfilter hingegen scheint die Störung komplett zu entfernen und vermindert die Qualität des eigentliches Bildes kaum - das liegt daran, dass die *Salz und Pfeffer*-Störung aus vereinzelten komplett von der Umgebung abweichenden Pixeln liegt, die nie als Median ausgewählt werden, weil ihr Wert entweder maximal oder minimal ist.

2.3 Behandlung des Randproblems

4 Möglichkeiten:

- Algorithmische Berücksichtigung.

  Z.B. könnte man überprüfen, ob ein Teil der Maske außerhalb des Bildes liegt und in diesem Falle nur die innerhalb liegenden Pixel für die Faltung berücksichtigen (z.B. den Mittelwert entsprechend anpassen).

- Kleineres Ergebnisbild

  Wenn man die Ränder einfach abschneidet, umgeht man das Problem - je nach Anwendungszweck kann das aber vielleicht unerwünscht sein, schließlich erhält man dann ein Bild mit anderen Dimensionen.

- Pixel außerhalb des Bildes mit 0 annehmen

  Das ist simpel, führt aber normalerweise zu einer Art Rahmen um das restliche Bild.

- Zyklischer Abschluss

  Hier wird das Bild als Ausschnitt aus einem sich wiederholenden Muster betrachtet. Das kann bei Bildern mit gleichmäßiger Farb- bzw. Grauwertsverteilung zu guten Ergebnissen führen, bei starken Unterschieden aber wieder zu einem Rahmen führen.

Die algorithmische Berücksichtigung scheint also die beste Methode zu sein, muss aber je nach Anwendungsfall angepasst werden. Je nach Bildart kann auch der zyklische Abschluss gute Ergebnisse liefern.

3 Fourier-Transformation

3.1 DFT und Glättung

DFT1.m

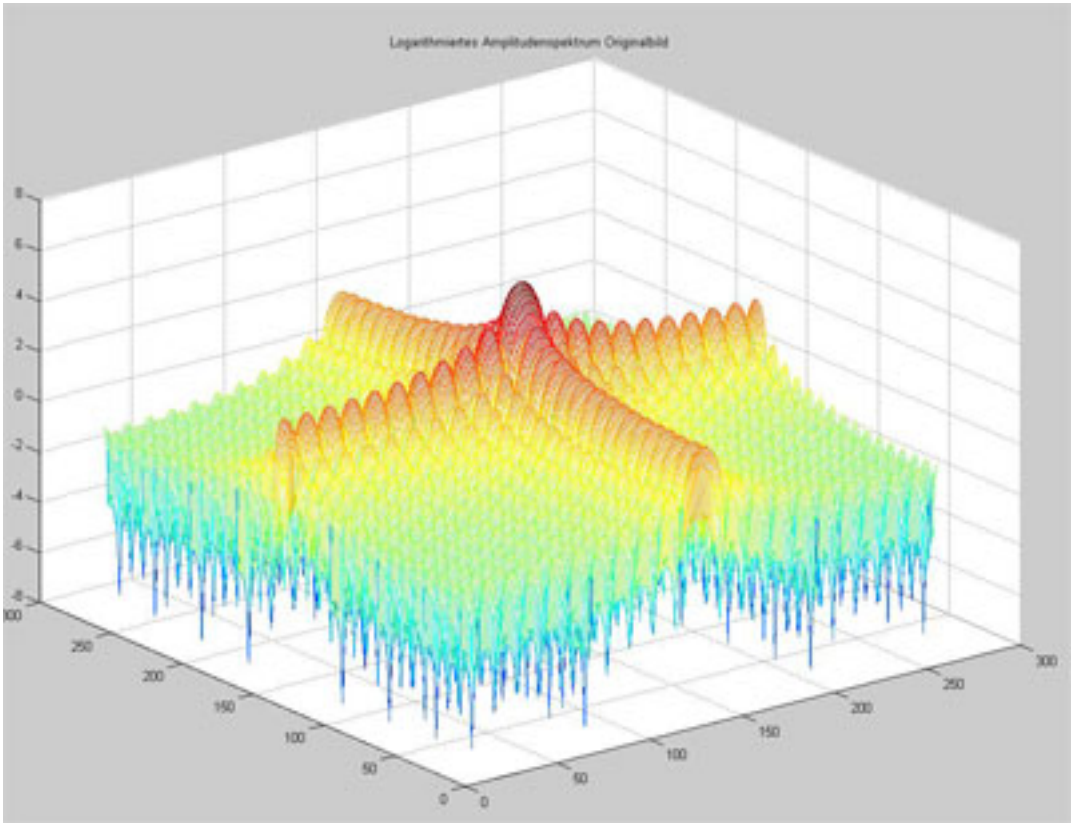
```
% Angabebild erzeugen
I = zeros(256, 256);
I(138:168, 138:158) = 1;

% ------------------------------------------------------------------------
% 3.1
dft = fft2(I); % Two-dimensional discrete Fourier transform
t1 = log(abs(dft));
o1 = fftshift(t1);

mesh(o1)
title ('Logarithmiertes Amplitudenspektrum Originalbild')
figure()

% ------------------------------------------------------------------------
% 3.2
temp1 = imrotate(I,45);
dft2 = fft2(temp1);
t2 = log(abs(dft2));
o2 = fftshift(t2);

mesh(o2)
title ('Logarithmiertes Amplitudenspektrum Originalbild 45° gedreht')
figure()

% ------------------------------------------------------------------------
% 3.3
% medfilt
m3 = fspecial('average', 3);
Im3 = conv2(I,m3, 'same');
dft3 = fft2(Im3);
t3 = log(abs(dft3));
om3 = fftshift(t3);

% gauss
g05 = fspecial('gaussian',3,0.5)
Ig05 = conv2(I,g05, 'same');
dft4 = fft2(Ig05);
```

```
t4 = log(abs(dft4));
og05 = fftshift(t4);

subplot (3,1,1)
mesh(o1)
title ('Logarithmiertes Amplitudenspektrum Originalbild')


subplot (3,1,2)
mesh(om3)
title ('Logarithmiertes Amplitudenspektrum 3x3 Mitterlwertfilter')

subplot (3,1,3)
mesh(og05)
title ('Logarithmiertes Amplitudenspektrum Gauss 0.5')

figure()

% ------------------------------------------------------------------------
% 3.4
m7 = fspecial('average', 7);
m9 = fspecial('average', 9);
g1 = fspecial('gaussian', 7, 1);
g15 = fspecial('gaussian', 9, 1.5);

% medfilt 7
Im7 = conv2(I, m7);
dft5 = fft2(Im7);
t5 = log(abs(dft5));
om7 = fftshift(t5);

% medfilt 9
Im9 = conv2(I, m9, 'same');
dft6 = fft2(Im9);
t6 = log(abs(dft6));
om9 = fftshift(t6);

% gauss 1
Ig1 = conv2(I, g1);
dft7 = fft2(Ig1);
t7 = log(abs(dft7));
og1 = fftshift(t7);

% gauss 15
```

```
Ig15 = conv2(I, g15);
dft8 = fft2(Ig15);
t8 = log(abs(dft8));
og15 = fftshift(t8);

r1 = o1(1:256, 128);
rm3 = om3(1:256, 128);
rg05 = og05(1:256, 128);
rm7 = om7(1:256, 128);
rm9 = om9(1:256, 128);
rg1 = og1(1:256, 128);
rg15 = og15(1:256, 128);

subplot(3,1,1)
plot(r1, 'r')
hold on
plot(rm3, 'g')
hold on
plot(rg05, 'b')
legend ('r1 = original', 'rm3 = 3x3 Mittelwert', 'rg05 = Gaussfilter 0.5')
xlabel ('Pixelindex')
ylabel ('Amplitude')
ylim([-40 10]);


subplot(3,1,2)
plot(r1, 'r')
hold on
plot(rm7, 'g')
hold on
plot(rg1, 'b')
legend ('r1 = original', 'rm7 = 7x7 Mittelwert', 'rg1 = Gaussfilter 1')
xlabel ('Pixelindex')
ylabel ('Amplitude')
ylim([-40 10]);

subplot(3,1,3)
plot(r1, 'r')
hold on
plot(rm9, 'g')
hold on
plot(rg15, 'b')
legend ('r1 = original', 'rm9 = 9x9 Mittelwert', 'rg015 = Gaussfilter 1.5')
xlabel ('Pixelindex')
```

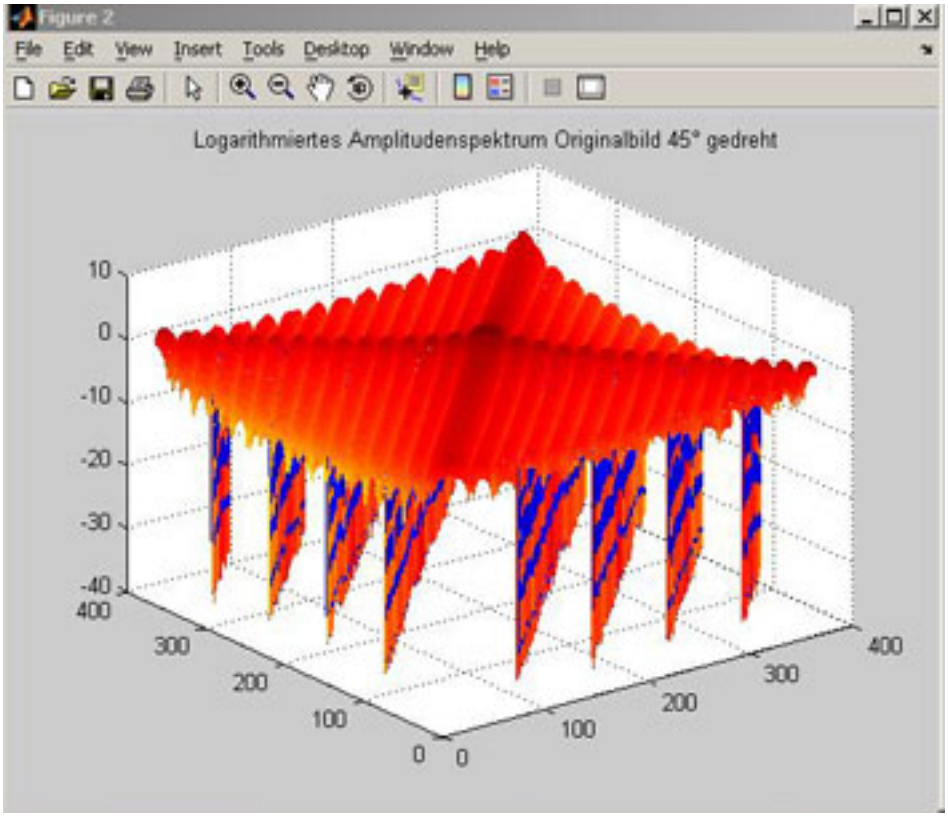
```
ylabel ('Amplitude')
ylim([-40 10]);

figure()
```

3.1.1 teil 1



Die harten Übergänge im Ortsraum resultieren in einer kreuzähnlichen Form im Fourierraum.

3.1.2 teil 2



Die Rotation im Ortsraum schlägt sich direkt in einer Rotation im Frequenzraum nieder.

3.1.3 teil 3

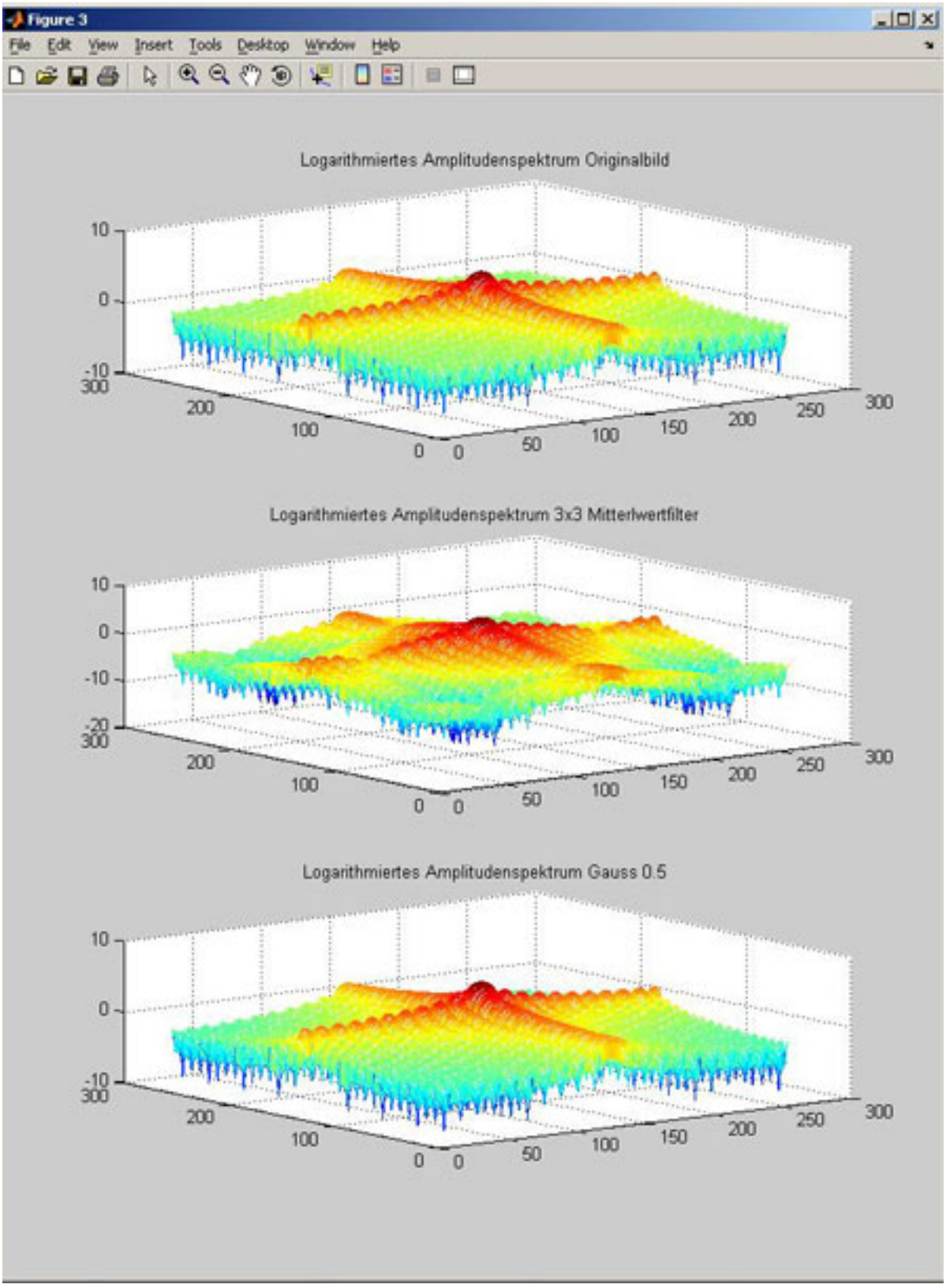
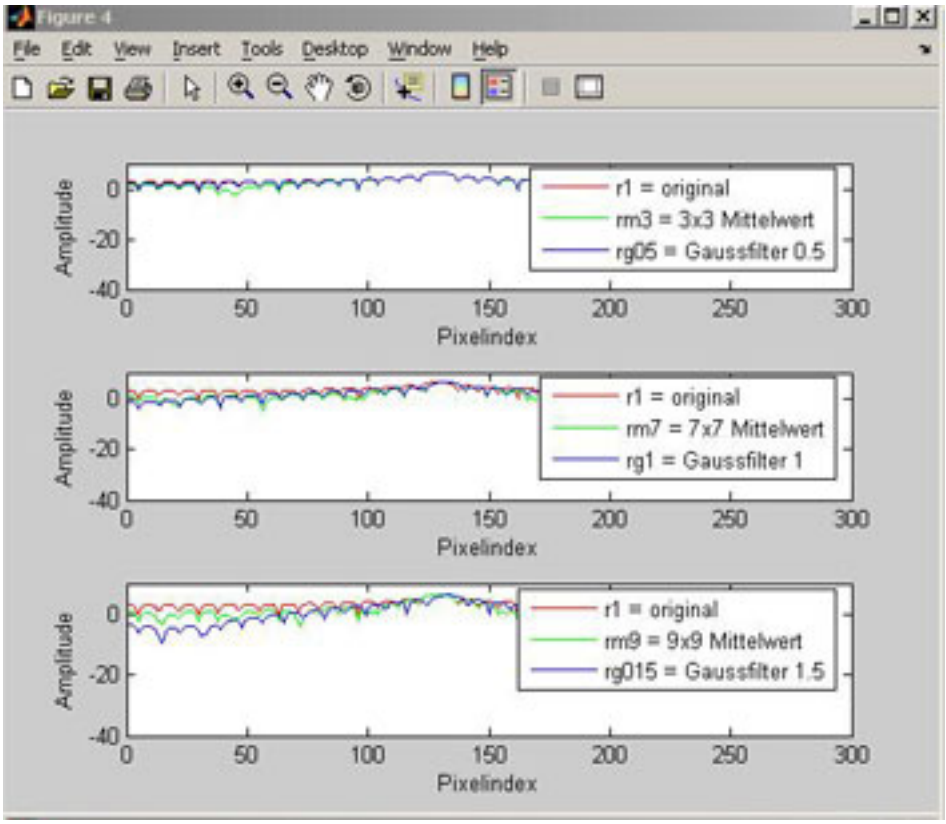


Figure 3
File Edit View Insert Tools Desktop Window Help
Logarithmiertes Amplitudenspektrum Originalbild
10
0
-10
300
200
100
0
0
50
100
150
200
250
300
Logarithmiertes Amplitudenspektrum 3x3 Mitterlwertfilter
10
0
-10
-20
300
200
100
0
0
50
100
150
200
250
300
Logarithmiertes Amplitudenspektrum Gauss 0.5
10
0
-10
300
200
100
0
0
50
100
150
200
250
300

Der Mittelwertfilter verändert die Frequenzdarstellung stärker.

3.1.4 teil 4



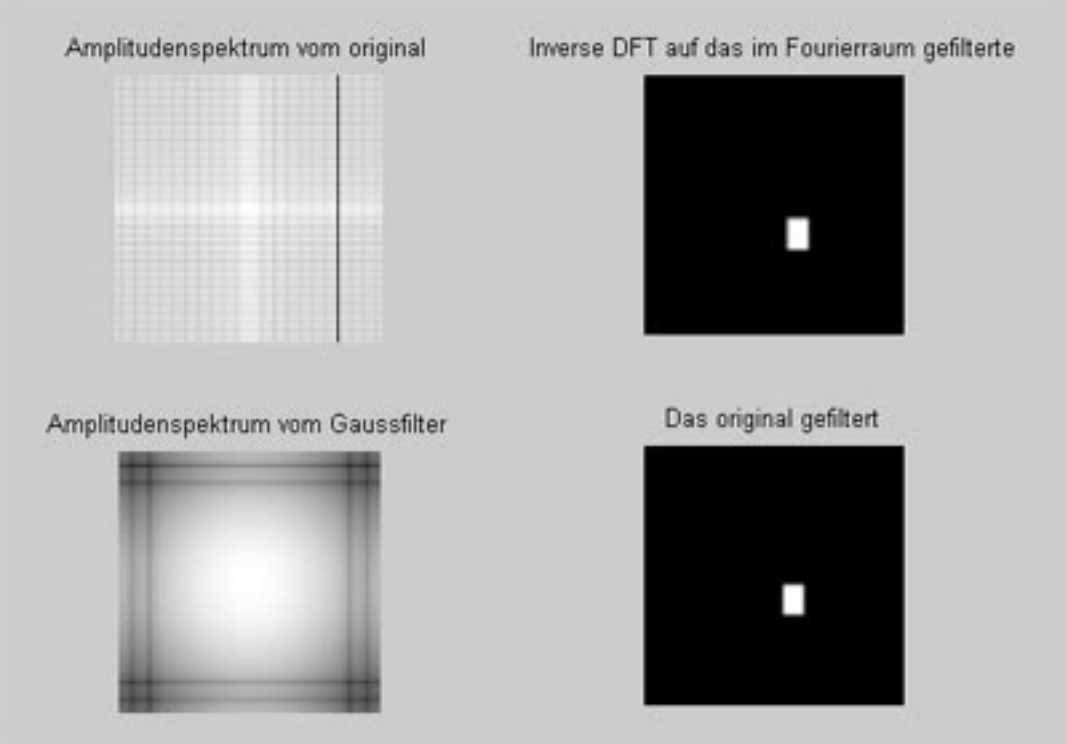
Bei größeren Masken verringert der Gaussfilter die hohen Frequenzen (d.h. die scharfen Übergänge) stärker als der Mittelwertfilter.

3.2 Konvolutionstheorem

DFT2.m

```
% Angabebild erzeugen
I = zeros(256, 256);
I(138:168, 138:158) = 1;

%fourier transform
dft = fft2(I, 264, 264);
dft = fftshift(dft);
dftlog = log(abs(dft));
```

```
%gauss im fourier raum
g15 = fspecial('gaussian', 9, 1.5);
dftg15 = fft2(g15, 264, 264);
dftg15 = fftshift(dftg15);
dftg15log = log(abs(dftg15));

g15filtered = dft .* dftg15;

%rücktransformieren
inverse = ifftshift(g15filtered);
inverse2 = ifft2(inverse);
inverse2 = inverse2(1:256, 1:256);
inverse2 = real(inverse2);

%filter auf original
Ifiltered = conv2(I, g15);


subplot(4,1,1)
imshow(dftlog,[])
title('Amplitudenspektrum vom original')

subplot(4,1,2)
imshow(dftg15log,[])
title('Amplitudenspektrum vom Gauss-Filter')

subplot(4,1,3)
imshow(inverse2,[])
title('Inverse DFT auf das im Fourierraum gefilterte')

subplot(4,1,4)
imshow(Ifiltered,[])
title('Das original gefiltert')
```



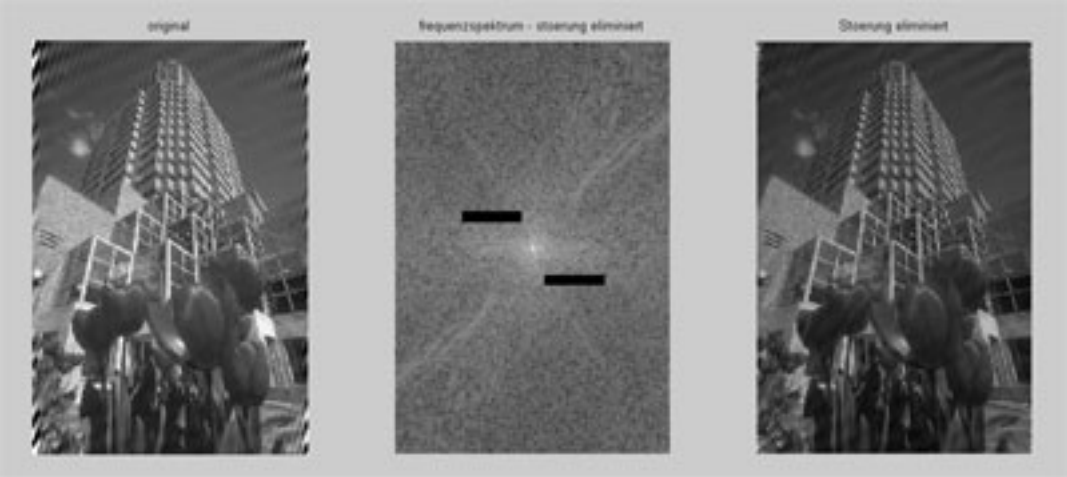
Da eine (elementweise) Multiplikation im Fourier-Raum mit einer Faltung im Ortsraum identisch ist, liefert eine Multiplikation des Amplitudenspektrums des Originalbildes mit dem Amplitudenspektrum eines Gauss-Filters das gleiche Ergebnis wie eine Gauss-Faltung des Originalbildes im Ortsraum. Wichtig ist es dabei, nur den Teil, der die Größe des Originalbildes hat, aus dem Ergebnis zu extrahieren (für die Multiplikation war eine Skalierung bzw. Auffüllung der beiden Spektren-Matrizen notwendig) sowie den bei der Transformation entstandenen Imaginärteil wegzulassen.

3.3 Bildrestaurierung

DFT3.m

```
I = imread('building_noisy.tif');

% fourier transformation
Idft = fft2(I);
Idfts = fftshift(Idft);

% erster index ist Reihen, zweiter ist Spalten
Idfts(137:142,88:122) = 0;
Idfts(100:105,40:74) = 0;
```

```
%inverse transformation
Inv = ifftshift(Idfts);
Inv = ifft2(Inv);

%original
subplot(1,3,1);
imshow(I);
title('original');

%frequenzspektrum - stoerung eliminiert
subplot(1,3,2);
imshow(log(abs(Idfts)),[]);
title('frequenzspektrum - stoerung eliminiert');

subplot(1,3,3);
imshow(Inv,[]);
title('Stoerung eliminiert'); % :-)
```



Eine Reduktion der Störung kann durch ein Eliminieren der blockförmigen Frequenzkonzentrationen erreicht werden. Ein gänzliches Entfernen der Störung ist schwierig, da im Bereich der Störung auch Frequenzen enthalten sind, die zur Bildinformation gehören.

4 Einfacher Kantenoperator

Kantenop.m

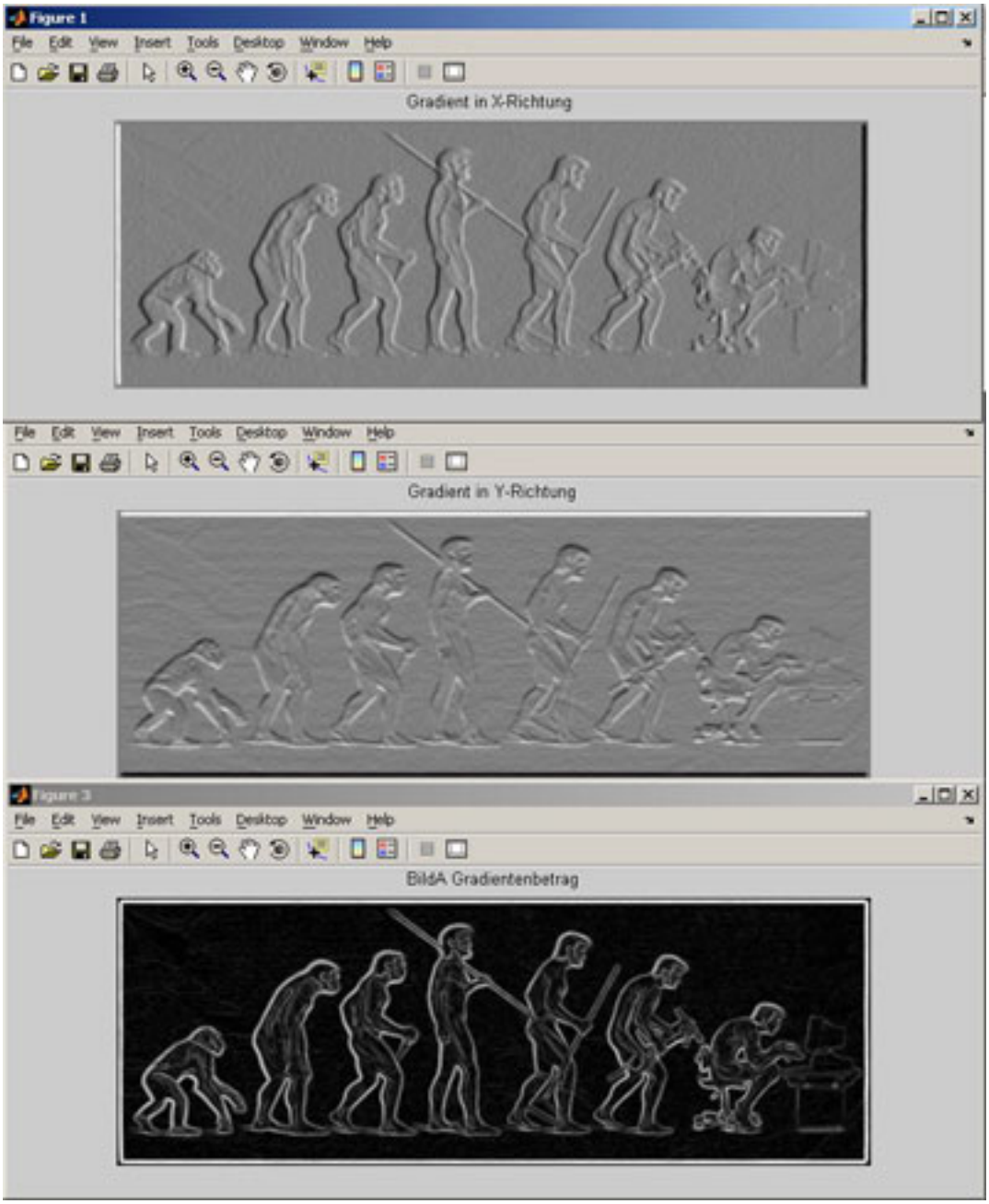
```
% einlesen des Bildes
BildA = imread('BildA.jpg');

% Bild Filtern
BildAgauss = conv2(BildA, fspecial('gaussian', 7, 1));

% Gradienten des gefilterten Bildes berechnen
[BildAgradientX,BildAgradientY] = gradient(BildAgauss);

figure();
imshow(BildAgradientX,[]);
title('Gradient in X-Richtung');
figure();
imshow(BildAgradientY,[]);
title('Gradient in Y-Richtung');
figure();

% Gradientenbetrag berechnen
BildAgradient = sqrt(BildAgradientX.^2+BildAgradientY.^2);


% Gradientenbetrag darstellen
imshow(BildAgradient,[]);
title('BildA Gradientenbetrag');
figure();

% invertierter Gradientenbetrag + Vektorfeld Bild A
imshow(-BildAgradient,[]);
title('BildA Gradient + Vektorfeld');

hold on

quiver(BildAgradientX, BildAgradientY);

% invertierter Gradientenbetrag + Vektorfeld Bild B
imshow(-BildAgradient,[]);
title('BildA Gradient + Vektorfeld');

hold on
```

```
quiver(BildAgradientX, BildAgradientY);

figure();

% Bild BW machen mittel threshold
ABW = BildAgradient / max(max(BildAgradient));

ABW = ABW >= 0.2;

imshow(ABW);
title('BildA BW');

% nr 6
[rows,cols] = size(ABW);
GX = BildAgradientX;
GY = BildAgradientY;
B = BildAgradient;
for i = 1:rows
    for j = 1:cols
        if ABW(i,j) == 1
            dir = [GY(i,j) ./ B(i,j), GX(i,j) ./ B(i,j)] ;
            dirRound = [round(dir(1)), round(dir(2))];
            nbpos = B(i + dirRound(1), j + dirRound(2));
            nbneg = B(i - dirRound(1), j - dirRound(2));
            if (B(i,j) < nbpos) | (B(i,j) < nbneg)
                ABW(i,j) = 0;
            end
        end
    end
end

figure();
imshow(ABW,[]);

figure();
img = imread('BildB.jpg');
imgE = myedge(img, 2, 0.26);
subplot(1,2,1); imshow(img);
subplot(1,2,2); imshow(imgE);
```

myedge.m

```
function e = myedge(I, sigma, threshold)

IGauss = conv2(I, fspecial('gaussian', 2*floor(3*sigma)+1, sigma));

[GX, GY] = gradient(IGauss);

GB = sqrt(GX.^2 + GY.^2);

Bin = GB ./ max(max(GB));
Bin = Bin >= threshold;


[rows,cols] = size(Bin);
for i = 1:rows
    for j = 1:cols
        if Bin(i,j) == 1
            dir = [GY(i,j) ./ GB(i,j), GX(i,j) ./ GB(i,j)] ;
            dirRound = [round(dir(1)), round(dir(2))];
            nbpos = GB(i + dirRound(1), j + dirRound(2));
            nbneg = GB(i - dirRound(1), j - dirRound(2));
            if (GB(i,j) < nbpos) | (GB(i,j) < nbneg)
                Bin(i,j) = 0;
            end
        end
    end
end

e = Bin;
```

4.0.1 teil 1-3

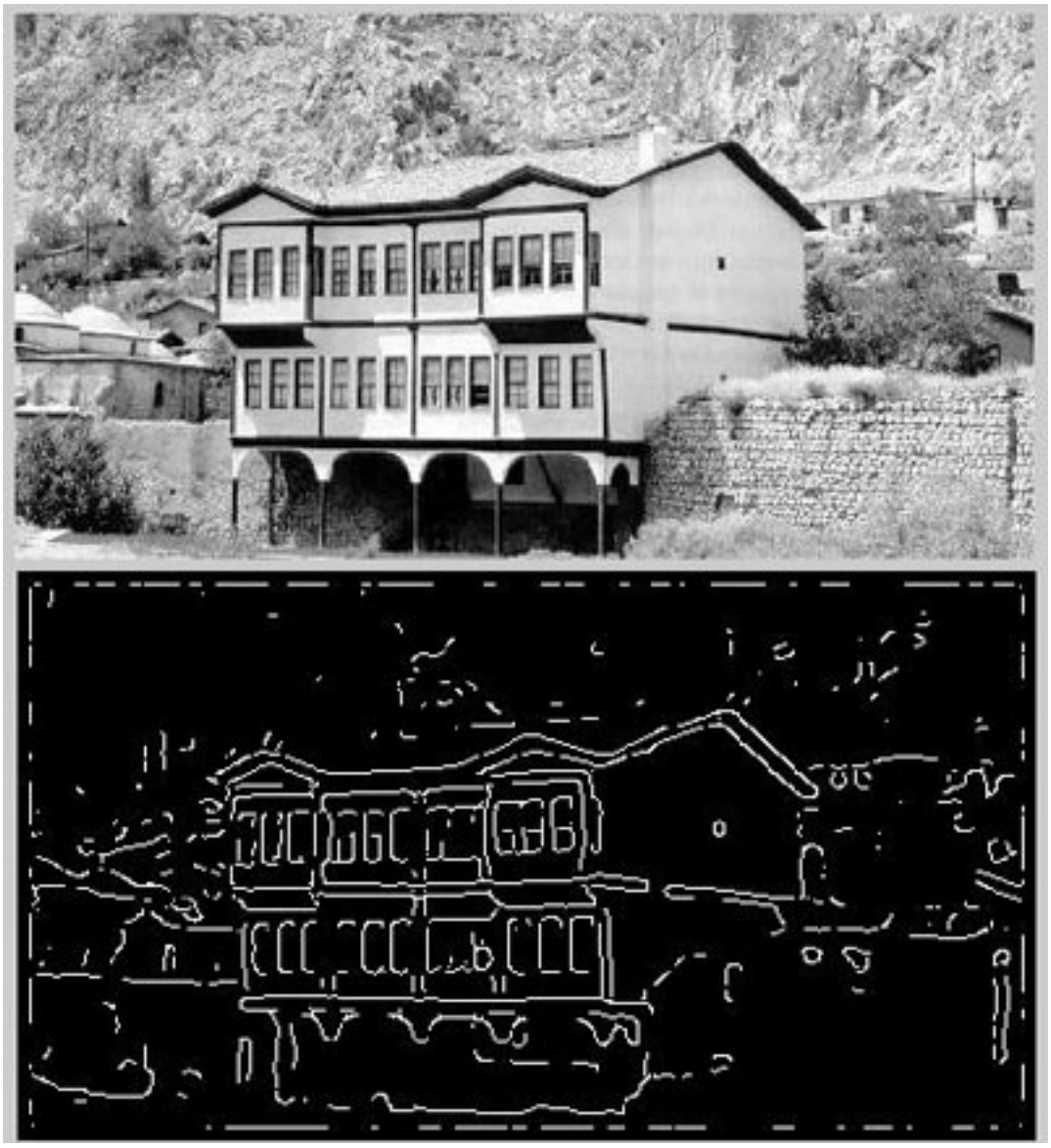


Die Darstellungen der Richtungsableitungen scheinen, als würden die Kanten von rechts bzw. von unten beleuchtet. Der Grund dafür ist, dass hier jeweils die Steigung (d.h. Stärke der Unterschiede in Grauwerten) in eine Richtung berechnet wurde.

4.0.2 teil 4-6

Die als Pfeile dargestellten Vektoren des aus den Richtungsableitungen konstruierten Vektorfeldes indizieren die Richtung und stärke der jeweils stärksten Steigung in Grauwertintensitäten.

4.0.3 teil 7



Aufgrund der scharfen und dichten Kontraste im Hintergrund des Bildes ist es hier schwierig, den richtigen Gauss-Parameter und den richtigen Binär-Threshold zu wählen. Eine relativ starke Glättung kann die Hintergrundkontraste aber ausreichend reduzieren, um ein brauchbares Ergebnis zu liefern.

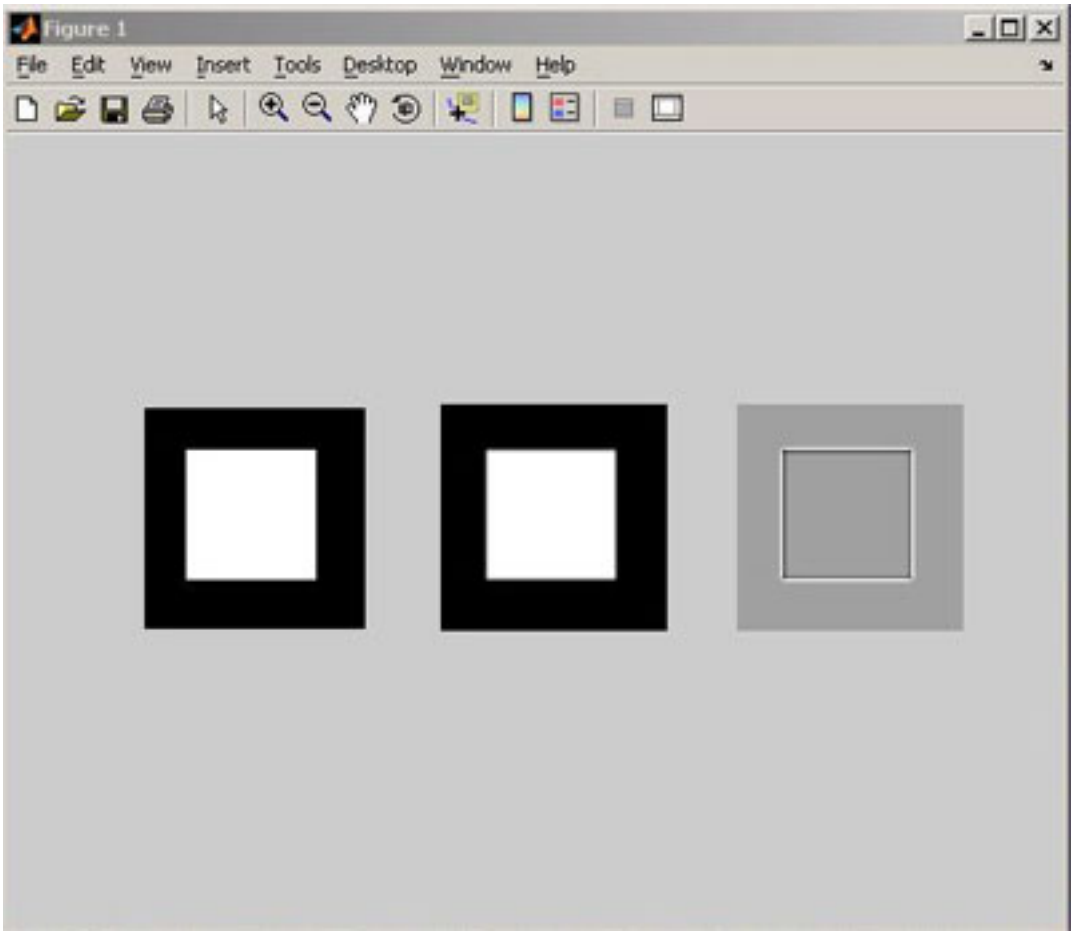
5 Laplace-Operator

Schaerfen.m

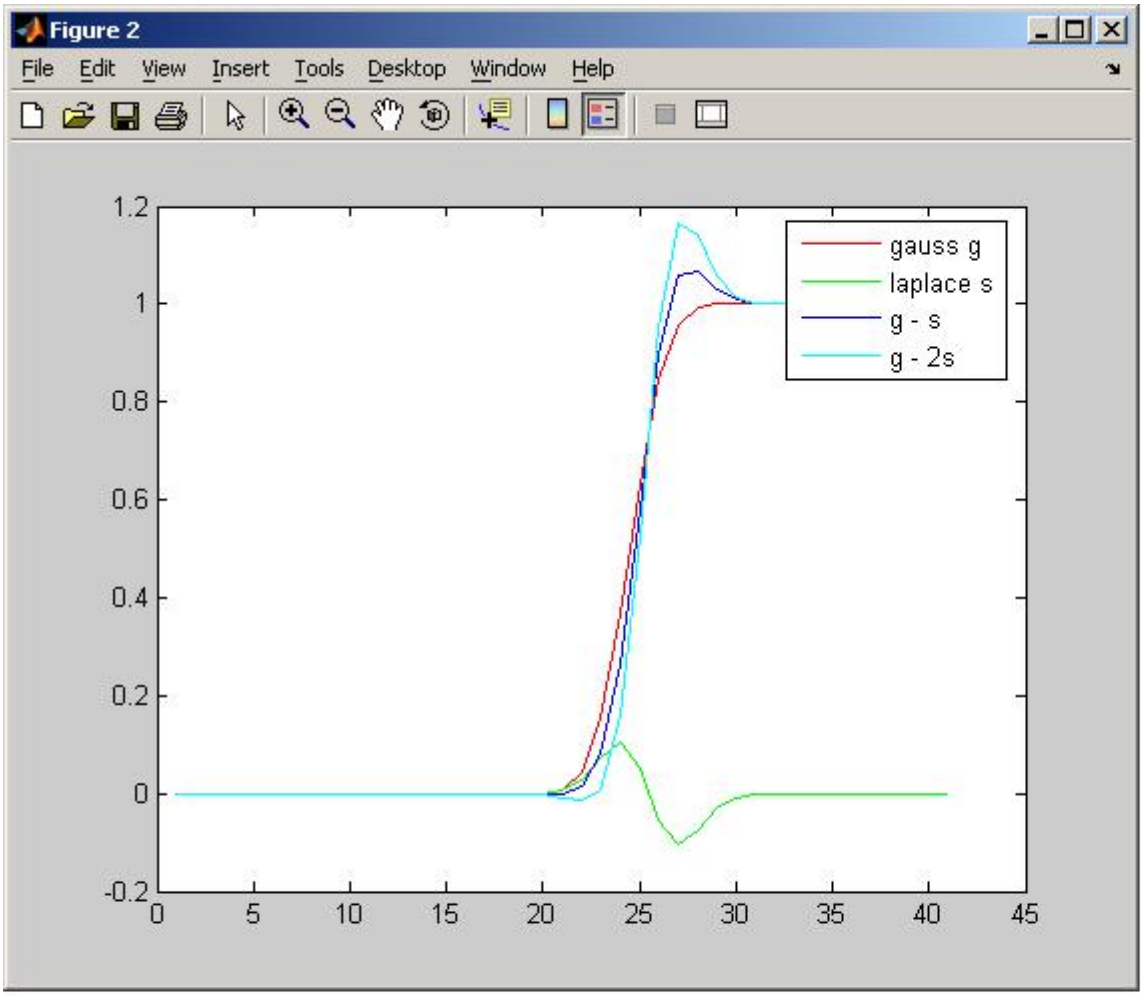
```
I = zeros(256, 256);
I(50:200,50:200) = 1;

%gaussfilter
g15= fspecial('gaussian', 9, 1.5);

B = zeros(3);
B(1,1) = 0;
B(1,2) = 1;
B(1,3) = 0;
B(2,1) = 1;
B(2,2) = -4;
B(2,3) = 1;
B(3,1) = 0;
B(3,2) = 1;
B(3,3) = 0;

Igauss = conv2(I, g15);
Ilap = conv2(Igauss, B);

figure();
subplot(1,3,1); imshow(I,[]);
subplot(1,3,2); imshow(Igauss,[]);
subplot(1,3,3); imshow(Ilap,[]);

g = Igauss(100,30:70);
s = Ilap(100,30:70);
figure();
plot(g, 'r');
hold on;
plot(s, 'g');
hold on;
plot(g-s, 'b');
hold on;
plot(g-2.*s, 'c');
legend('gauss g','laplace s','g - s', 'g - 2s');
```

```
V = B;

moon = imread('Moon.jpg');
retina = imread('retina.png');

% Falten mit Gaussfilter
moongauss = conv2(moon, fspecial('gaussian', 7, 1));
retinagauss = conv2(retina, fspecial('gaussian', 7, 1));

% Laplace-Transformierung des gefilterten Bildes mittels Faltung des Gauss-geglätteten Bil
moonlaplace = conv2(moongauss, V);
retinalaplace = conv2(retinagauss, V);

% Letzte Faltung ergibt Bild, dass größer ist als Igauss daher Extraktion
% eines Subsets aus Ilaplace
moonI = moonlaplace(1:size(moongauss, 1), 1:size(moongauss, 2));
retinaI = retinalaplace(1:size(retinagauss, 1), 1:size(retinagauss, 2));

% Zusammenfügen von den niederfrequenten und den verstärkten hochfrequenten Anteilen;
% Zur Bildschärfung kommt es da die Kanten hochfrequent (und scharf) sind
moonscharf = moongauss - moonI;
retinascharf = retinagauss - retinaI;

subplot(2, 2, 1);
imshow(moon)
title('Original');

subplot(2, 2, 2);
imshow(moongauss, [])
title('Gauss gefiltertes Bild');

subplot(2, 2, 3);
imshow(moonlaplace, [])
title('Bild nach Laplace Transformation');

subplot(2, 2, 4);
imshow(moonscharf, [])
title('Geschärftes Bild');

figure();

subplot(2, 2, 1);
imshow(retina)
title('Original');
```

```
subplot(2, 2, 2);
imshow(retinagauss, [])
title('Gauss gefiltertes Bild');

subplot(2, 2, 3);
imshow(retinalaplace, [])
title('Bild nach Laplace Transformation');

subplot(2, 2, 4);
imshow(retinascharf, [])
title('Geschärftes Bild');

moonrow1 = moonscharf(270,40:70);
moonrow2 = moon(270,40:70);
moonrow3 = moongauss(270,40:70);
moonrow4 = moonlaplace(270,40:70);

retrow1 = retinascharf(154, 140:158);
retrow2 = retina(154,140:158);
retrow3 = retinagauss(154,140:158);
retrow4 = retinalaplace(154,140:158);

figure();
plot(moonrow3, 'r');
hold on
plot(moonrow4, 'b');
hold on
plot(moonrow3 - moonrow4, 'g');
hold on
plot(moonrow3 - 2 .* moonrow4, 'black');
legend('gauss','laplace', 'gauss - laplace', 'gauss - 2* laplace');
title('moon');

figure();
plot(retrow3, 'r');
hold on
plot(retrow4, 'b');
hold on
plot(retrow3 - retrow4, 'g');
hold on
plot(retrow3 - 2 .* retrow4, 'black');
legend('gauss','laplace', 'gauss - laplace', 'gauss - 2* laplace');
title('retina');
```



Im Laplace-transformierten Bild werden Übergänge von niedrigen zu hohen Grauwerten als positive Werte dargestellt (im Bild: weiß), Übergänge von hohen zu niedrigen Grauwerten als negative Werte (im Bild: schwarz).



Durch eine Subtraktion der Laplace-Darstellung eines Bildes von einer geglätteten Darstellung des selben Bildes lassen sich die weichen Übergänge so weit verstärken, dass Werte außerhalb des Darstellungsbereichs entstehen, die daher abgeschnitten werden und somit die Kanten schärfen.

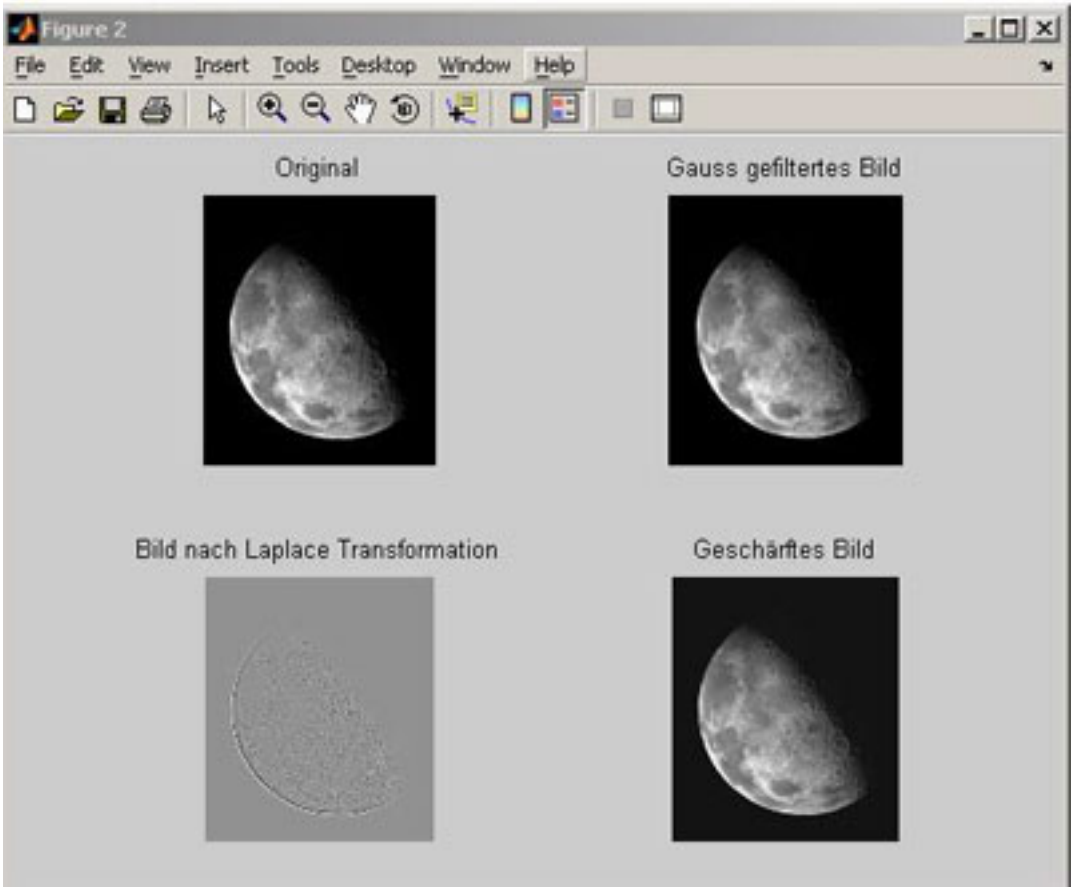


Figure 2
File Edit View Insert Tools Desktop Window Help
Original
Gauss gefiltertes Bild
Bild nach Laplace Transformation
Geschärftes Bild

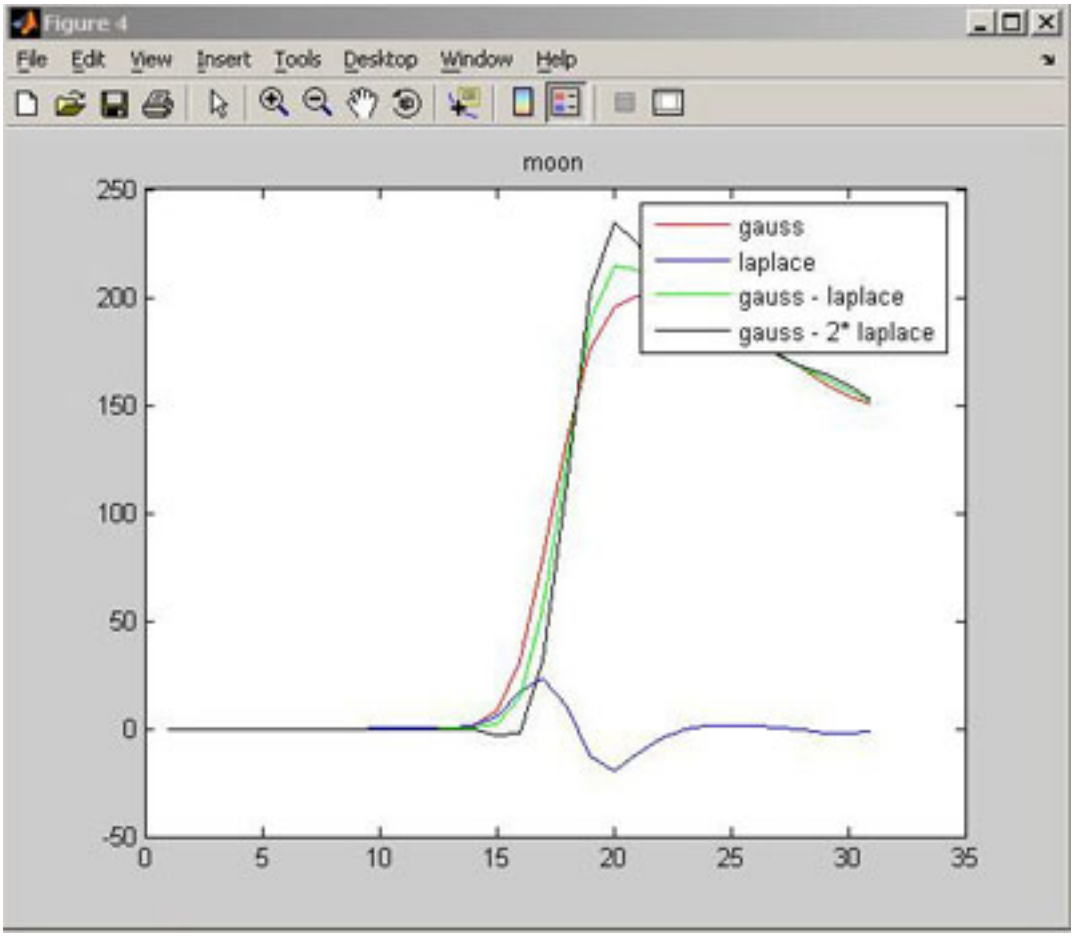


Figure 4
File Edit View Insert Tools Desktop Window Help
moon
gauss
laplace
gauss - laplace
gauss - 2* laplace
250
200
150
100
50
0
-50
0
5
10
15
20
25
30
35

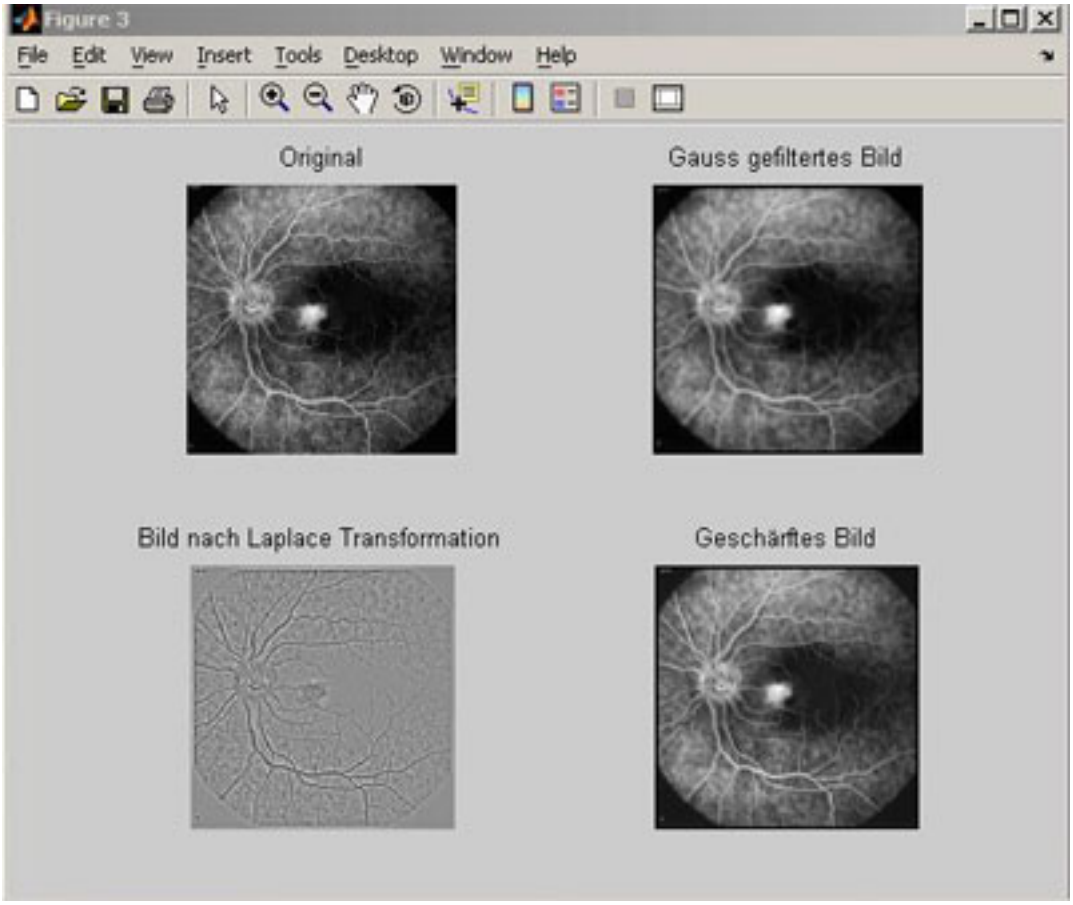
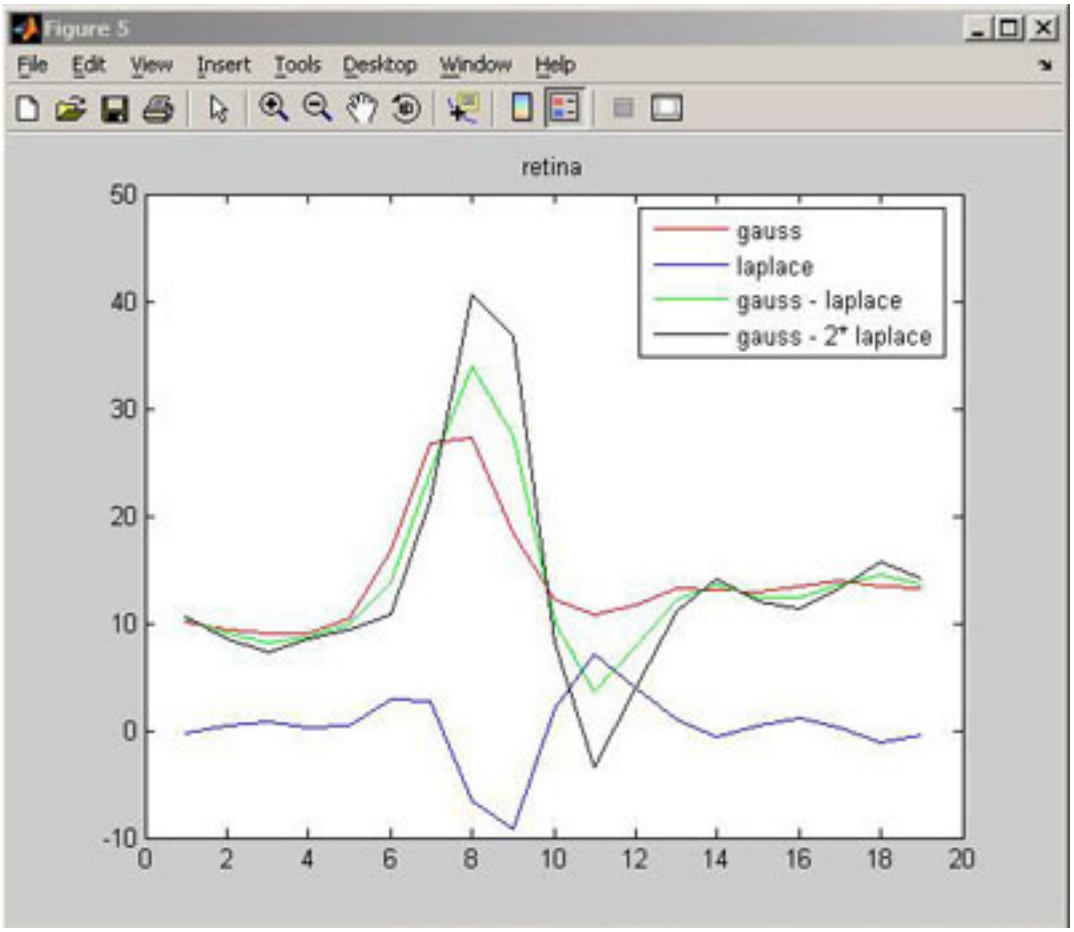


Figure 3
File Edit View Insert Tools Desktop Window Help
Original
Gauss gefiltertes Bild
Bild nach Laplace Transformation
Geschärftes Bild



Bildstörungen werden beim Schärfen ebenfalls verstärkt.