4. Vorlesung Mikroskopische Bildverarbeitung

Wahlpflichtmodul 9521: EI-M im 1. und 3. Fachsemester

Initialisierung

4. Dezember 2015

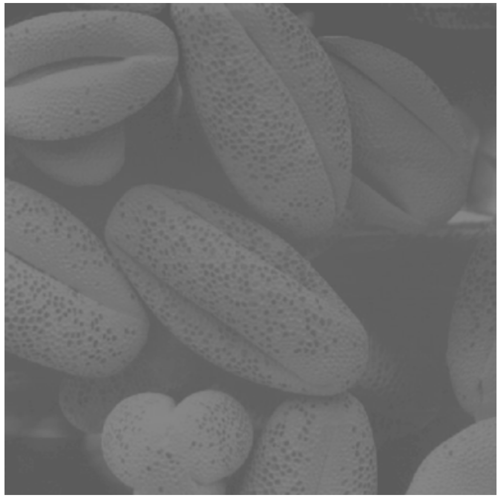
Wiederholung aus der 3. Vorlesung

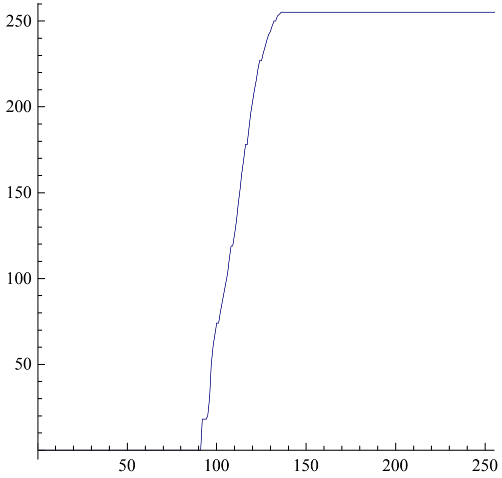
Aus 10. Histogramme, Histogrammausgleich, Histogrammanpassung

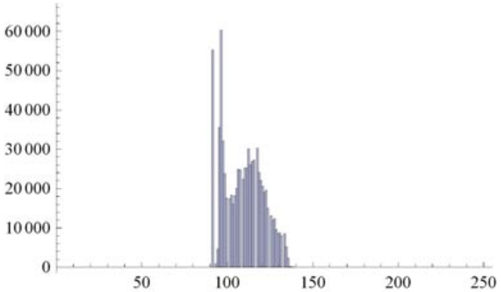
Histogrammausgleich

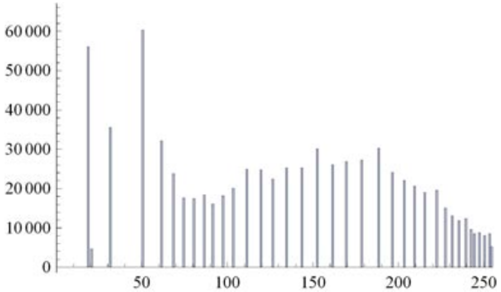
- Ziel ist eine linear ansteigende kumulative Häufigkeitsverteilung (Annahme, daß im Normalfall jede Intensitätsstufe gleichhäufig vorzukommen habe)
- aus der kumulativen Häufigkeitsverteilung wird durch Normierung auf die Bildgröße und den Wertebereich eine Tonwertkurve als Look-up-Table erstellt

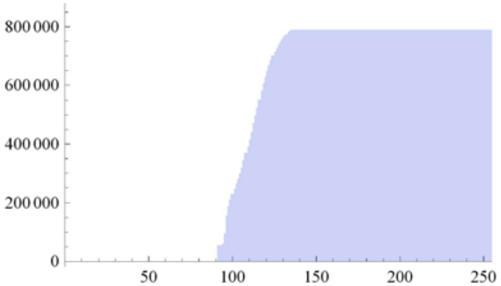
```
Clear[histogrammausgleich]
histogrammausgleich[skalarbildein_Image] := Module[{stufenanzahlen, kumulation, skalarbildaus},
   stufenanzahlen = BinCounts[Flatten[ImageData[skalarbildein, "Byte"]], {0, 255 + 1, 1}];
   kumulation = Round[Accumulate[stufenanzahlen] / (Times @@ ImageDimensions[skalarbildein]) * 255];
   (*
   Print[Grid[{{Show[ListPlot[stufenanzahlen,Filling→Axis,PlotRange→{{0,255},All}],ImageSize→256],
         Show[ListLinePlot[kumulation,AspectRatio→1,PlotRange→{{0,255},All}],ImageSize→256]}}]];
   *)
   skalarbildaus = Image[Map[kumulation[[# + 1]] &, ImageData[skalarbildein, "Byte"], {2}], "Byte"];
   Print[Grid[{{Show[skalarbildein, ImageSize → 256], Show[skalarbildaus, ImageSize → 256],
        Show[ListLinePlot[kumulation, AspectRatio → 1, PlotRange → {{0, 255}, All}], ImageSize → 256]}}]];

   Print[Grid[{{Show[Histogram[Flatten[ImageData[skalarbildein, "Byte"]], {0, 255, 1}, "Count", PlotRange → {{0, 255}, Automatic},
          AxesOrigin → {0, 0}], ImageSize → 256], Show[Histogram[Flatten[ImageData[skalarbildaus, "Byte"]],
          {0, 255, 1}, "Count", PlotRange → {{0, 255}, Automatic}, AxesOrigin → {0, 0}], ImageSize → 256]}}]];

   Print[Grid[{{Show[Histogram[Flatten[ImageData[skalarbildein, "Byte"]], {0, 255, 1}, "CumulativeCount", PlotRange → {{0, 255}, Automatic},
          AxesOrigin → {0, 0}], ImageSize → 256], Show[Histogram[Flatten[ImageData[skalarbildaus, "Byte"]],
          {0, 255, 1}, "CumulativeCount", PlotRange → {{0, 255}, Automatic}], ImageSize → 256]}}]];
   (*
   Print[MapThread[(#1->#2)&,{Range[0,255],kumulation}]];
   *)
   skalarbildaus
  ];

histogrammausgleich[pollen];
```

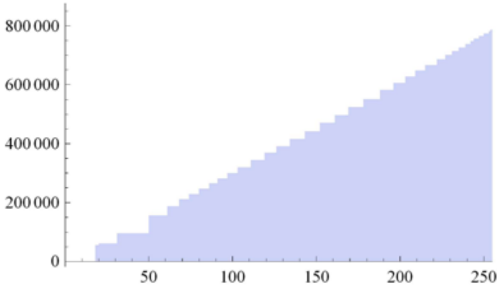




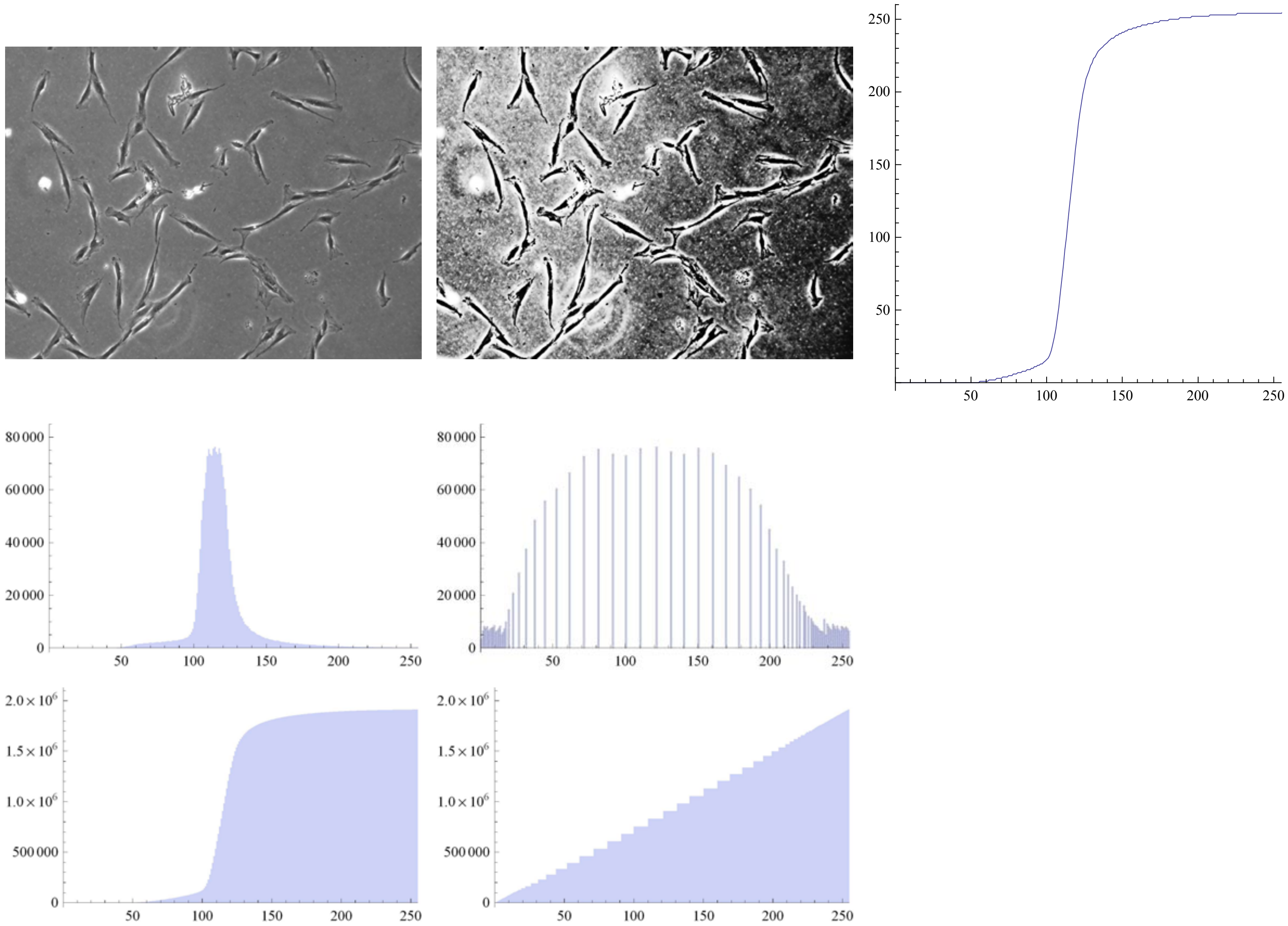




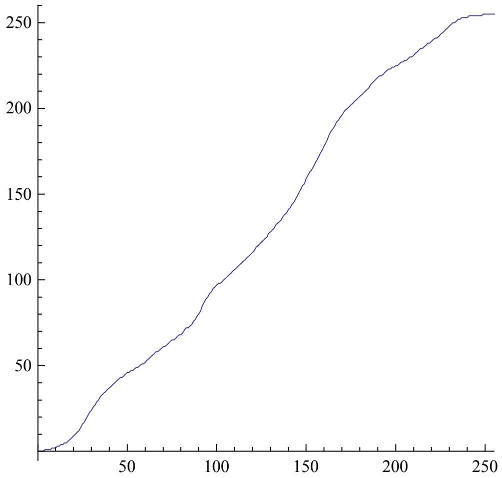


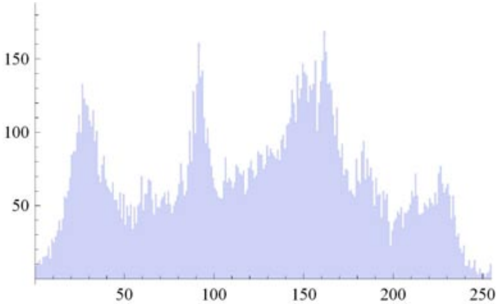


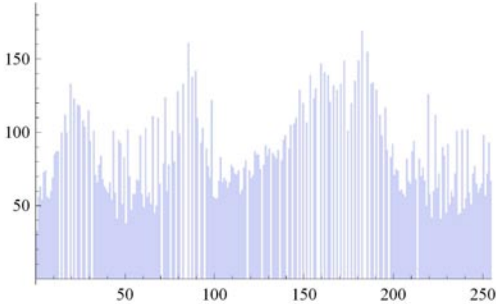

```
histogrammausgleich[phasenkontrast];
```

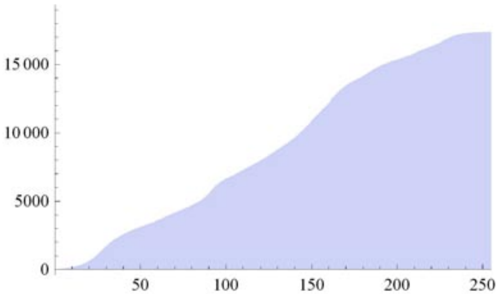


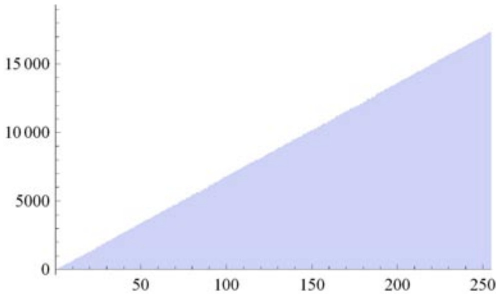
```
histogrammausgleich[lenay];
```



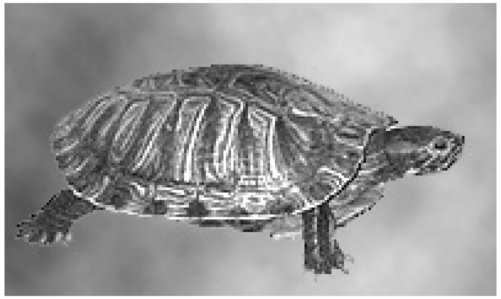


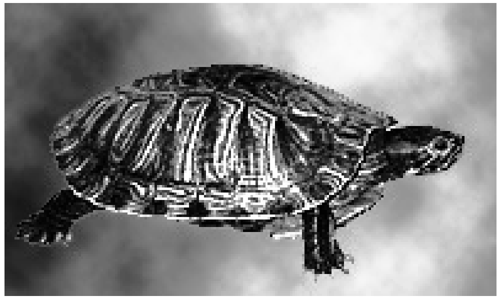


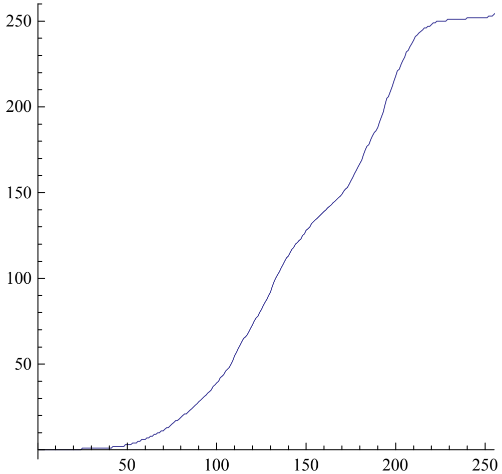


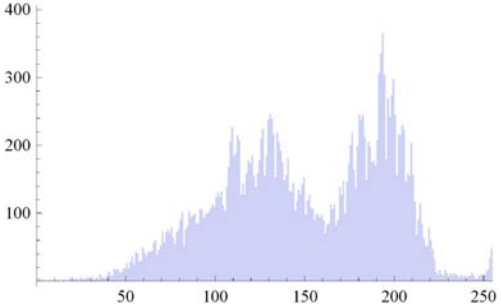


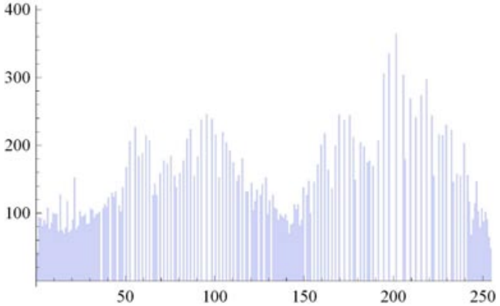

```
histogrammausgleich[turtle];
```

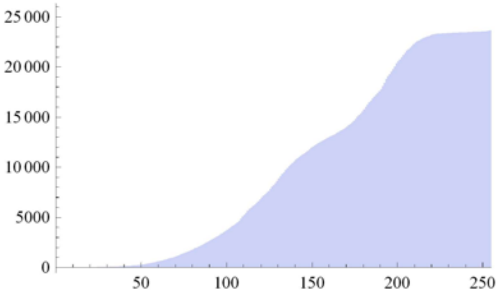


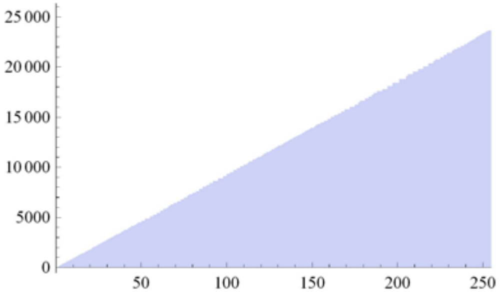




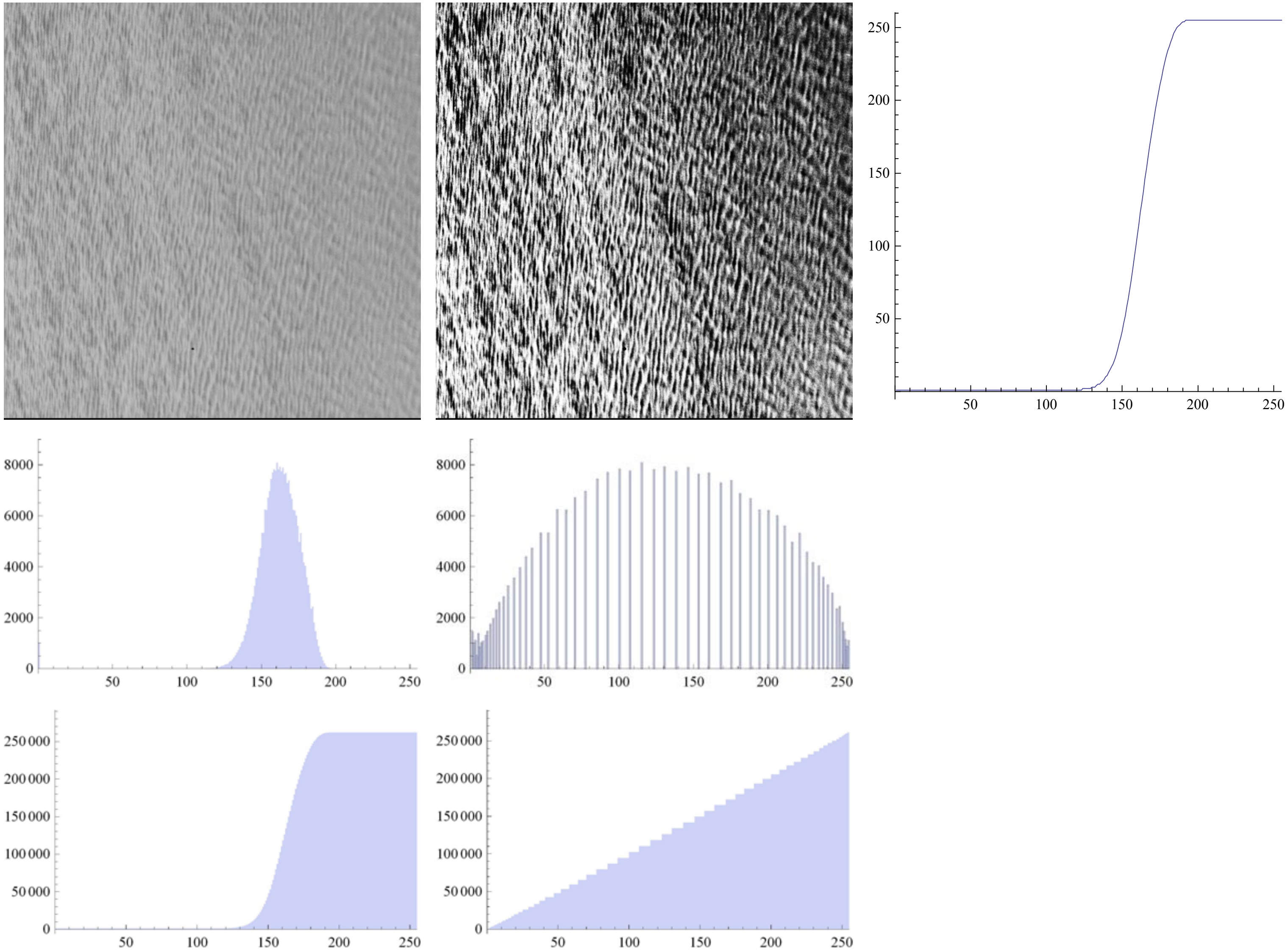




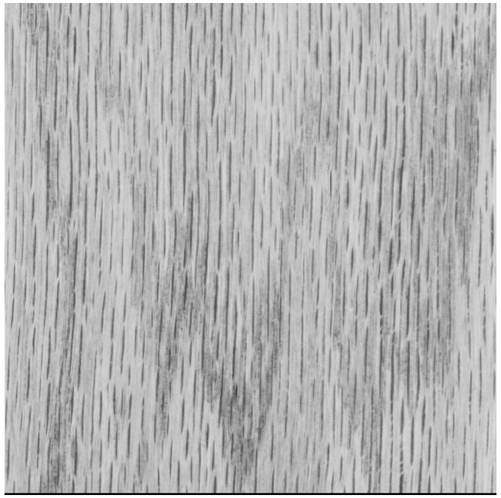


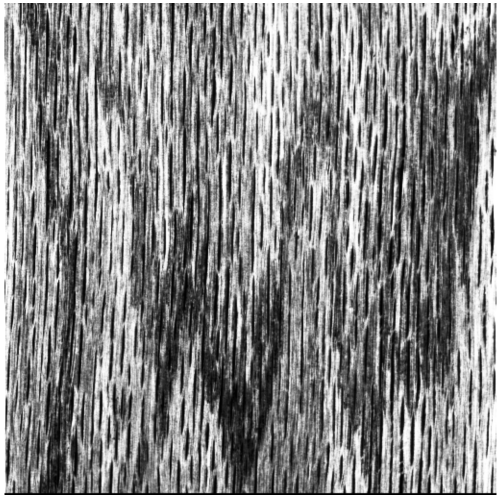


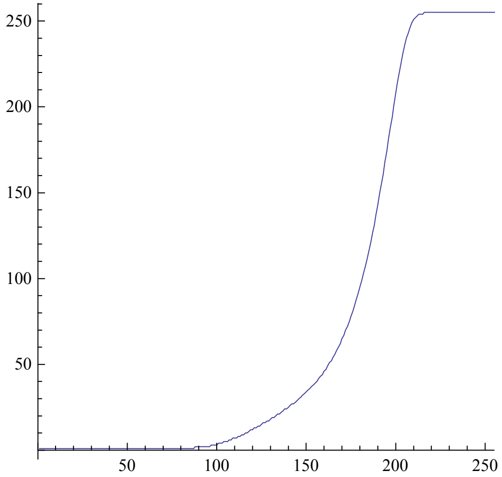

```
histogrammausgleich[maserung1];
```

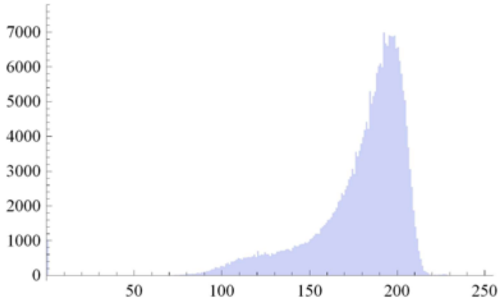


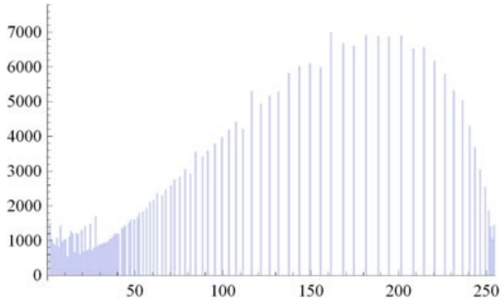
```
histogrammausgleich[maserung2];
```

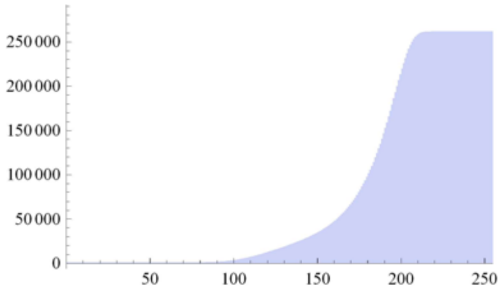


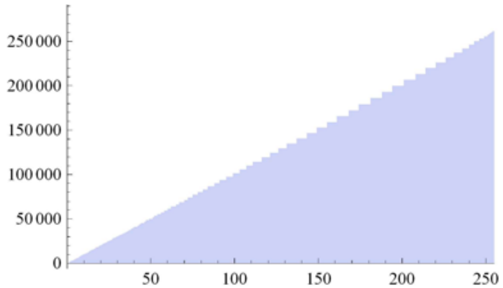




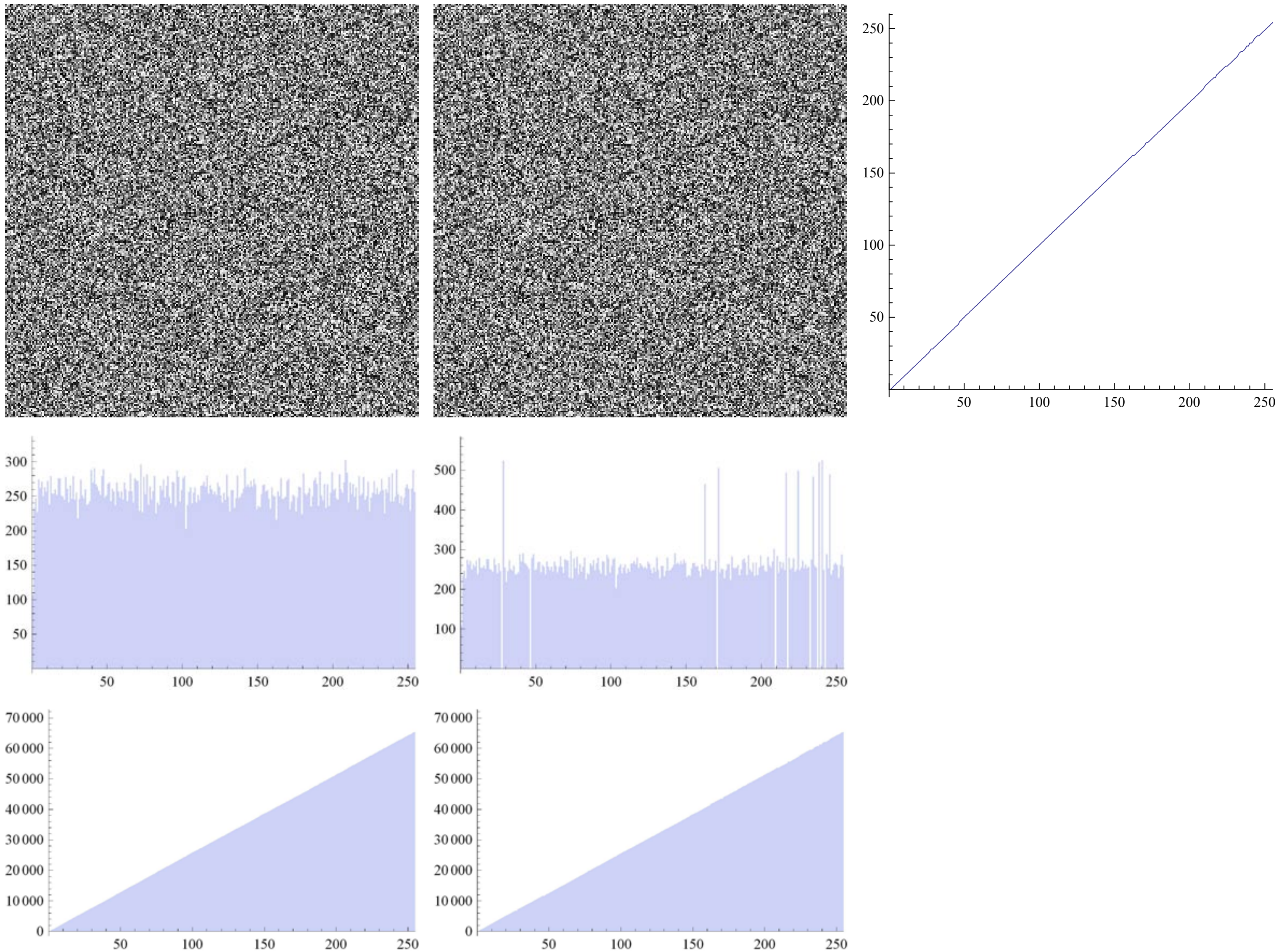




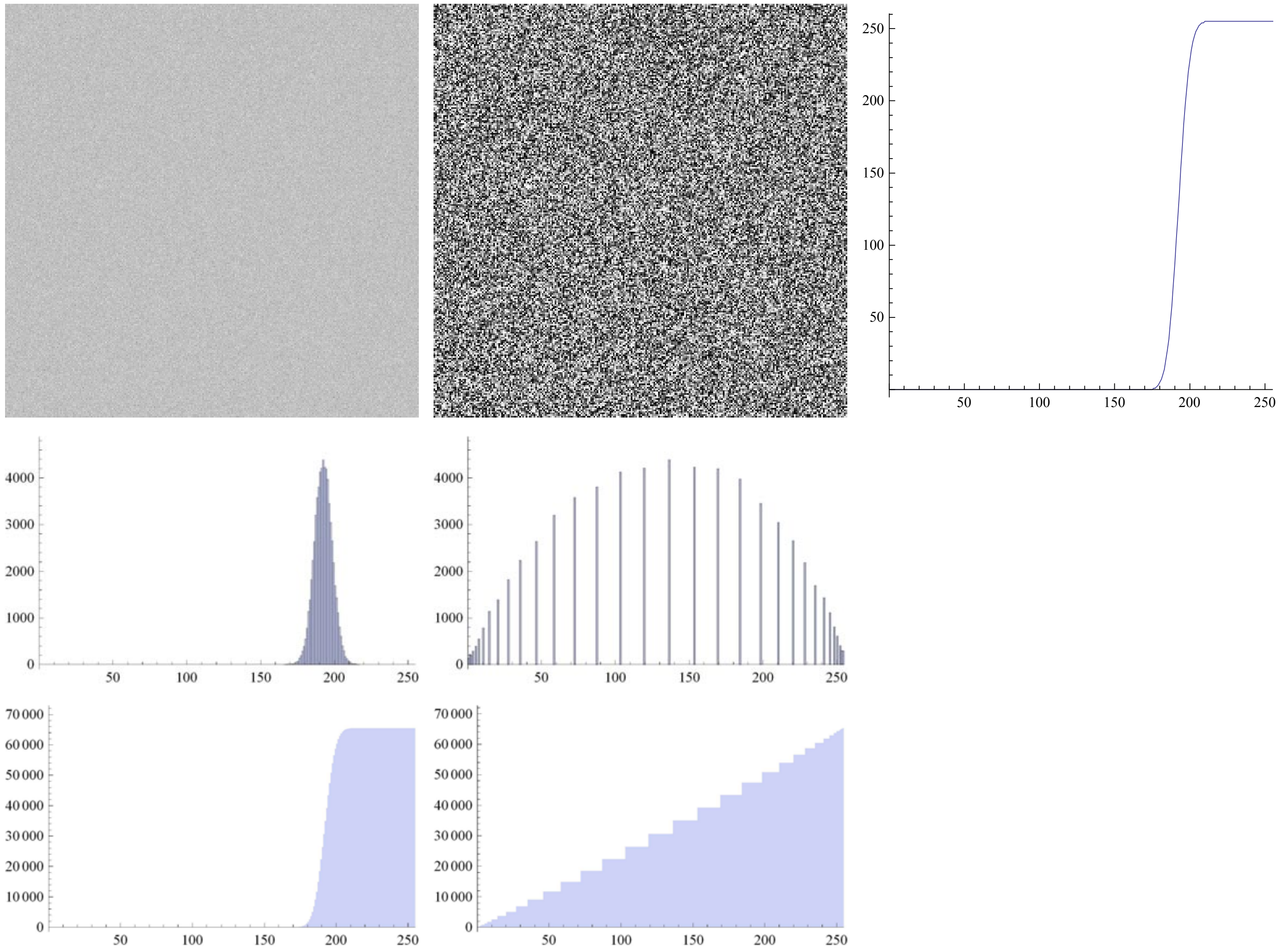





```
histogrammausgleich[randombild];
```




```
histogrammausgleich[rauschbild];
```



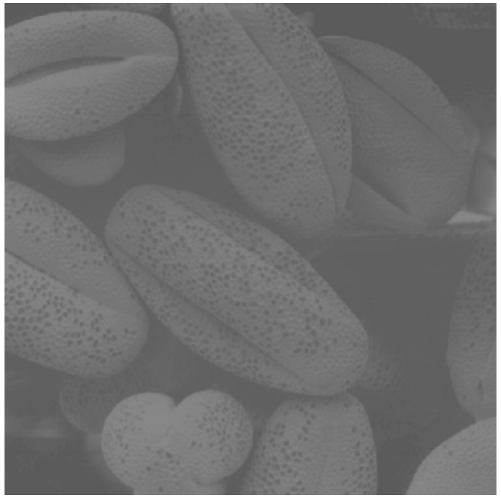
Histogrammanpassung

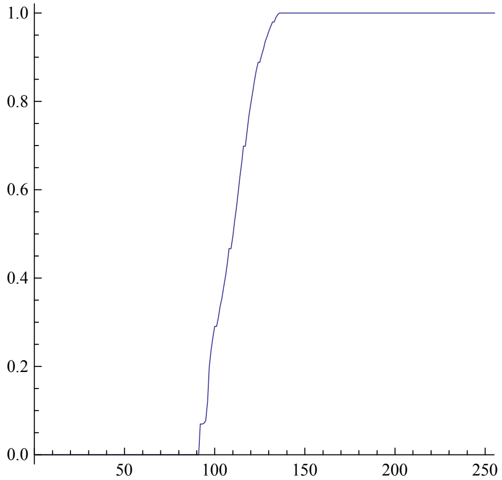
- Ziel ist die möglichst genaue Anpassung der Verteilungsfunktion $p_{in}(i)$ eines Bildes G_{in} an die Verteilungsfunktion $p_{ref}(i)$

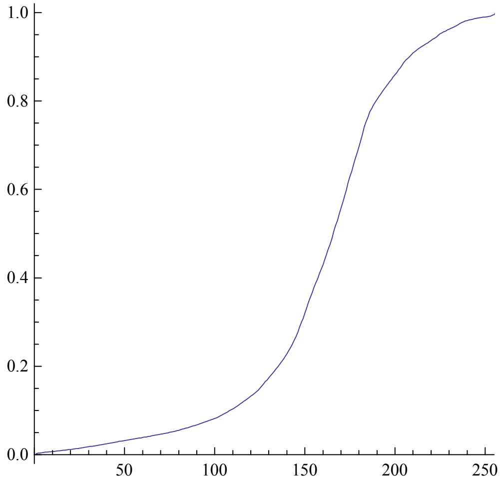
eines Referenzbildes G_{ref} für alle i des Wertebereichs $[0...G_{max}]$
- man liest für jedes Pixel den Wert der kumulativen Wahrscheinlichkeitsdichte des Bildes ab und schaut mit diesem Ablesewert in der kumulativen Wahrscheinlichkeitsdichte des Referenzbildes nach, welcher Referenzbildwert dazu korrespondiert
- es wird eine Tonwertkurve als Look-up-Table erstellt

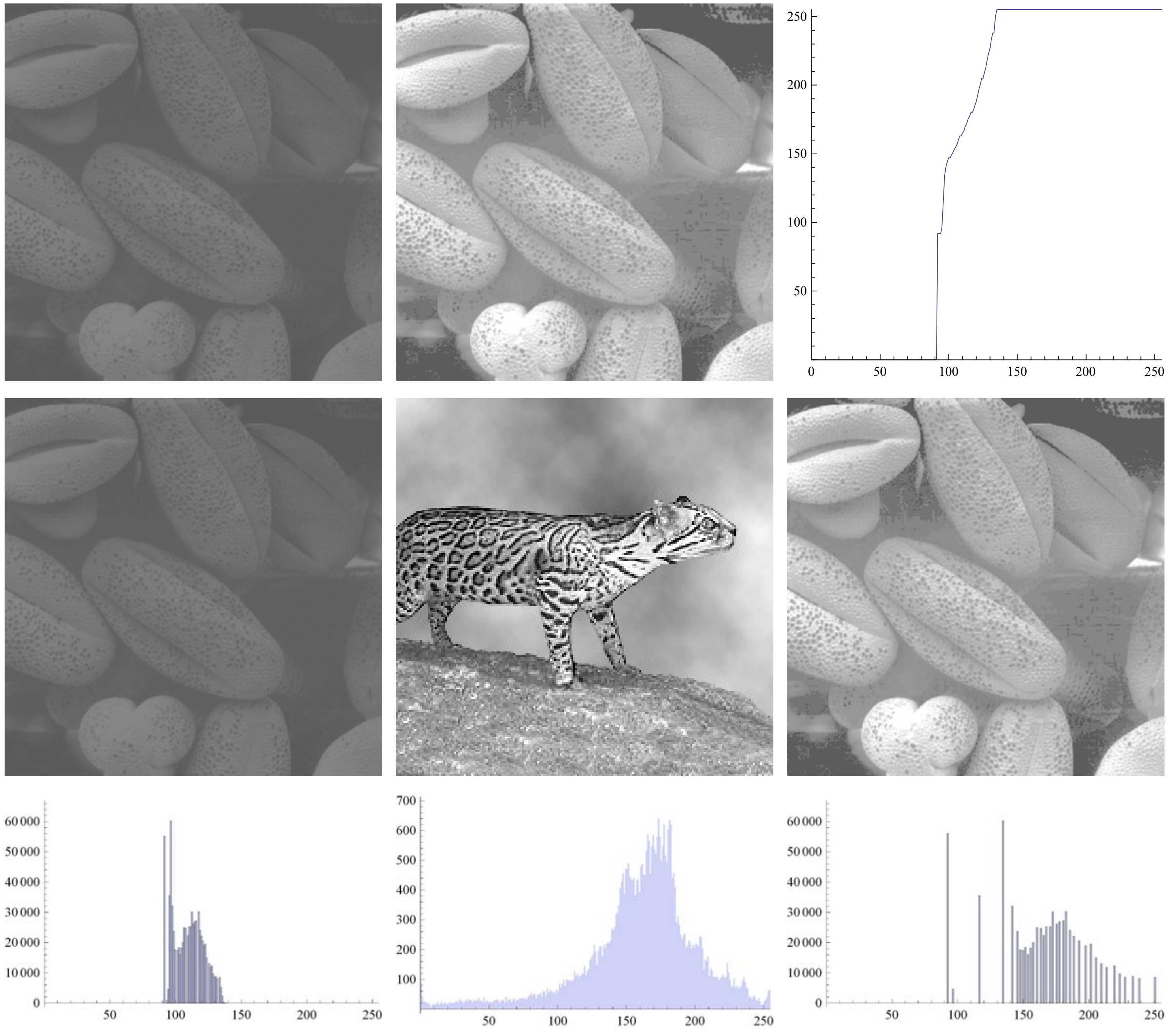
```
Clear[histogrammanpassung]
histogrammanpassung[skalarbildein_Image, skalarbildref_Image] :=
  Module[{stufenanzahlenein, stufenanzahlenref, kumulationein, kumulationref, skalarbildaus, lut},
   stufenanzahlenein = BinCounts[Flatten[ImageData[skalarbildein, "Byte"]], {0, 255 + 1, 1}];
   kumulationein = Accumulate[stufenanzahlenein];
   kumulationein = N[kumulationein / Last[kumulationein]];
   stufenanzahlenref = BinCounts[Flatten[ImageData[skalarbildref, "Byte"]], {0, 255 + 1, 1}];
   kumulationref = Accumulate[stufenanzahlenref];
   kumulationref = N[kumulationref / Last[kumulationref]];


   Print[Grid[{{Show[skalarbildein, ImageSize → 256],
       Show[ListLinePlot[kumulationein, AspectRatio → 1, PlotRange → {{0, 255}, All}], ImageSize → 256]}}]];
   Print[Grid[{{Show[skalarbildref, ImageSize → 256], Show[ListLinePlot[kumulationref,
         AspectRatio → 1, PlotRange → {{0, 255}, All}], ImageSize → 256]}}]];

   (*
   Print[Grid[{{Show[ListPlot[stufenanzahlenein,Filling→Axis,PlotRange→{{0,255},All}],ImageSize→256],
       Show[ListLinePlot[kumulationein,AspectRatio→1,PlotRange→{{0,255},All}],ImageSize→256],Show[skalarbildein,ImageSize→256]}}]];
   Print[Grid[{{Show[ListPlot[stufenanzahlenref,Filling→Axis,PlotRange→{{0,255},All}],ImageSize→256],
       Show[ListLinePlot[kumulationref,AspectRatio→1,PlotRange→{{0,255},All}],ImageSize→256],Show[skalarbildref,ImageSize→256]}}]];
   *)
   lut = Map[Function[mapwert, -1 + First@First@Position[kumulationref, _? (# ≥ kumulationein[[mapwert + 1]] &)]], Range[0, 255, 1], {1}];

   skalarbildaus = Image[Map[lut[[#]] &, ImageData[skalarbildein, "Byte"] + 1, {2}], "Byte"];
   (*Print[Grid[{{Show[skalarbildein,ImageSize→256],Show[skalarbildaus,ImageSize→256]}}]];*)

   (*
   Print[Grid[{{Show[ListLinePlot[kumulationein,AspectRatio→1,PlotRange→{{0,255},All}],ImageSize→256],
        Show[ListLinePlot[kumulationref,AspectRatio→1,PlotRange→{{0,255},All}],ImageSize→256],
        Show[ListLinePlot[lut,AspectRatio→1,PlotRange→{{0,255},All}],ImageSize→256]}}]];
   *)
```

```
    Print[Grid[{{Show[skalarbildein, ImageSize → 256], Show[skalarbildaus, ImageSize → 256],
        Show[ListLinePlot[lut, AspectRatio → 1, PlotRange → {{0, 255}, {0, 255}}], ImageSize → 256]}}]];
    (*
    Print@Show[Histogram[{Flatten[ImageData[skalarbildein,"Byte"]],Flatten[ImageData[skalarbildref,"Byte"]],
         Flatten[ImageData[skalarbildaus,"Byte"]]},{0,255,1},PlotRange→{{0,255},Automatic},AxesOrigin→{0,0}],ImageSize→256];
    *)

    Print[Grid[{{Show[skalarbildein, ImageSize → 256], Show[skalarbildref, ImageSize → 256], Show[skalarbildaus, ImageSize → 256]}}]];


    Print[Grid[{{Show[Histogram[Flatten[ImageData[skalarbildein, "Byte"]], {0, 255, 1}, "Count", PlotRange → {{0, 255}, Automatic},
          AxesOrigin → {0, 0}], ImageSize → 256], Show[Histogram[Flatten[ImageData[skalarbildref, "Byte"]], {0, 255, 1}, "Count",
          PlotRange → {{0, 255}, Automatic}, AxesOrigin → {0, 0}], ImageSize → 256], Show[Histogram[Flatten[ImageData[skalarbildaus, "Byte"]],
          {0, 255, 1}, "Count", PlotRange → {{0, 255}, Automatic}, AxesOrigin → {0, 0}], ImageSize → 256]}}]];

    Print[Grid[{{Show[Histogram[Flatten[ImageData[skalarbildein, "Byte"]], {0, 255, 1}, "CumulativeCount", PlotRange → {{0, 255}, Automatic},
          AxesOrigin → {0, 0}], ImageSize → 256], Show[Histogram[Flatten[ImageData[skalarbildref, "Byte"]], {0, 255, 1}, "CumulativeCount",
          PlotRange → {{0, 255}, Automatic}, AxesOrigin → {0, 0}], ImageSize → 256], Show[Histogram[Flatten[ImageData[skalarbildaus, "Byte"]],
          {0, 255, 1}, "CumulativeCount", PlotRange → {{0, 255}, Automatic}], ImageSize → 256]}}]];

    (*
    Print[MapThread[(#1->#2)&,{Range[0,255],lut}]];
    *)
    skalarbildaus

   ];

histogrammanpassung[pollen, ocelot];
```

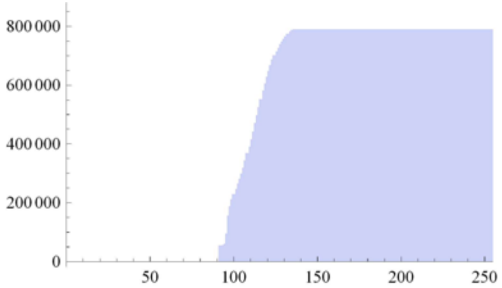


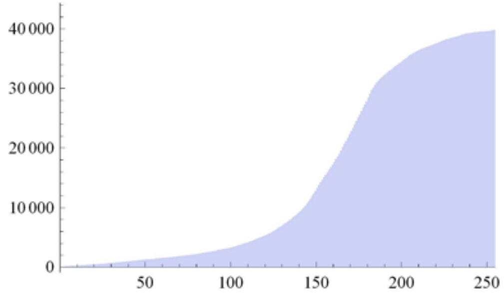

1.0
0.8
0.6
0.4
0.2
0.0
50
100
150
200
250

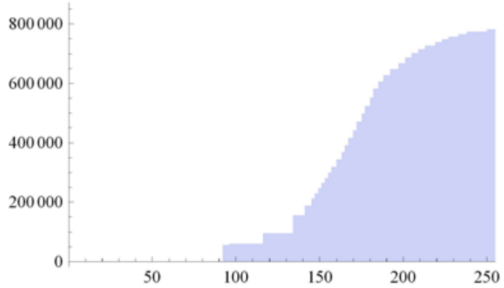

1.0
0.8
0.6
0.4
0.2
0.0
50
100
150
200
250



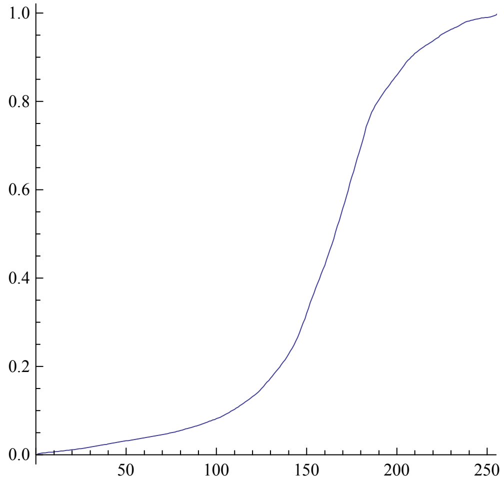
250
200
150
100
50
0
50
100
150
200
250
60000
50000
40000
30000
20000
10000
0
700
600
500
400
300
200
100

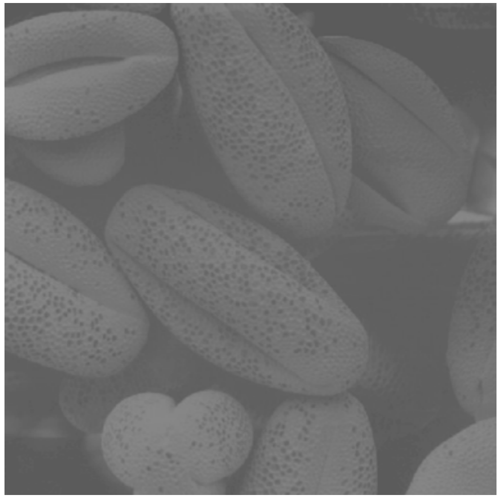


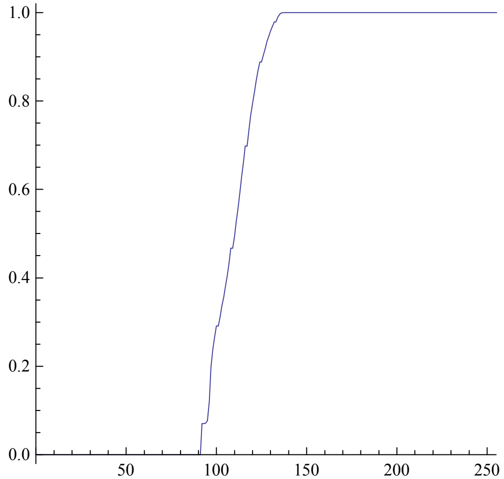


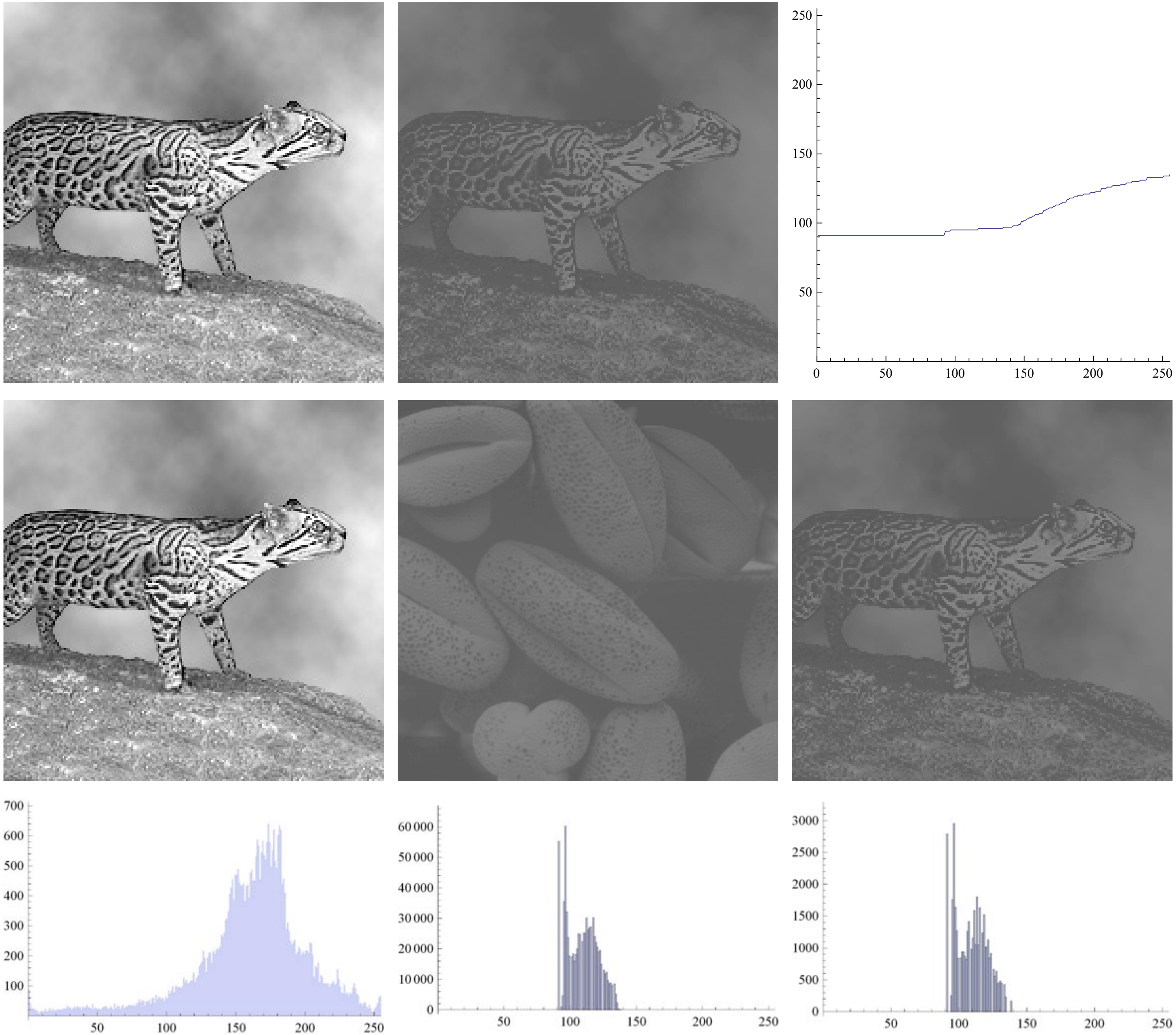


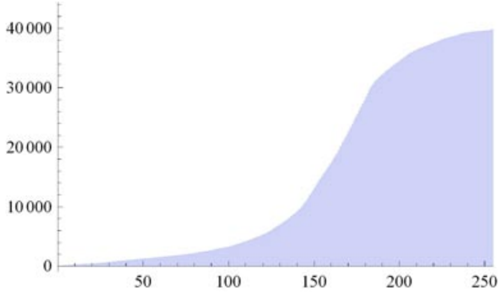
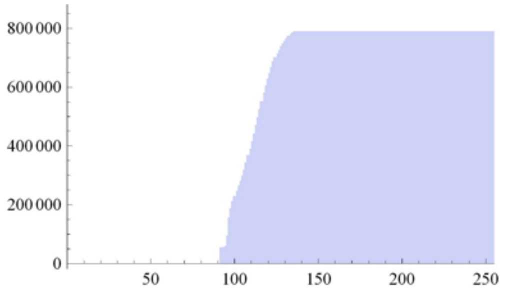
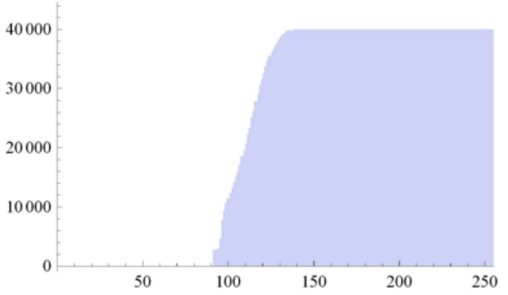

```
histogrammanpassung[ocelot, pollen];
```



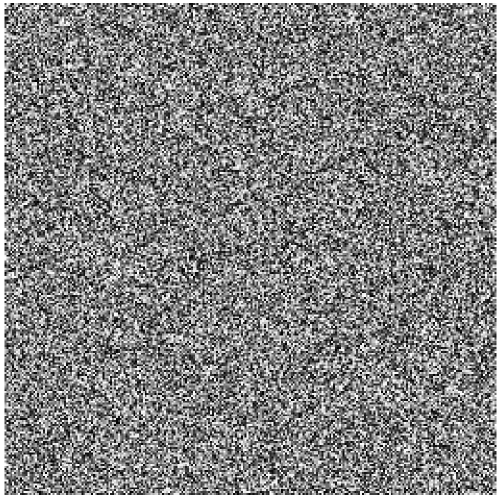
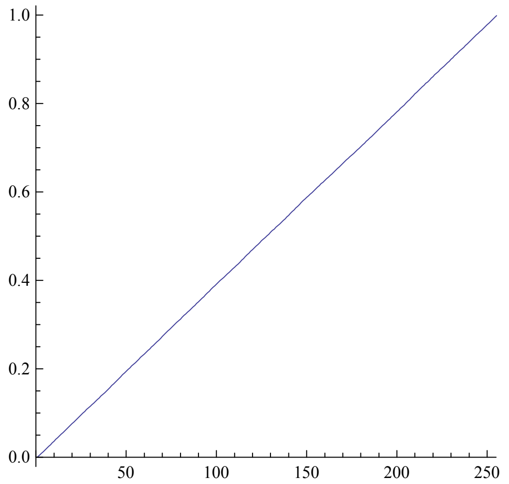


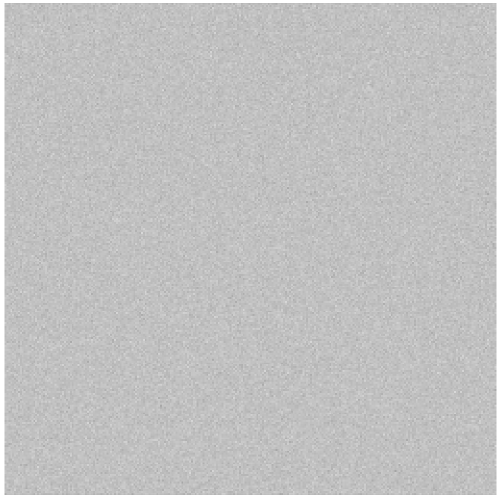
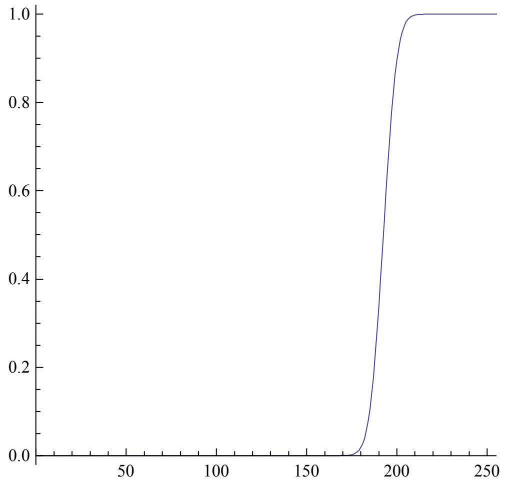


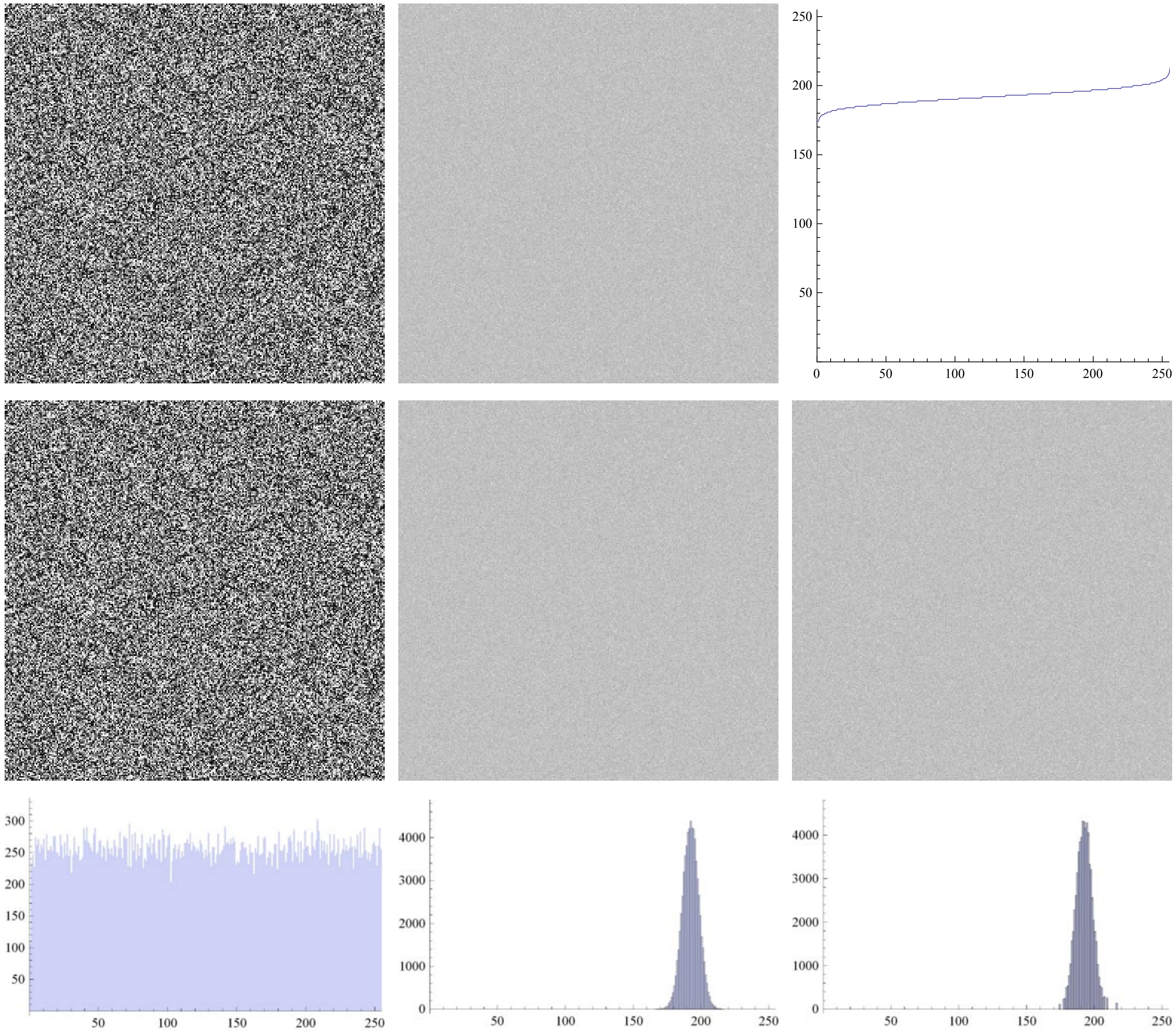

250
200
150
100
50
0 50 100 150 200 250
700 600 500 400 300 200 100
50 100 150 200 250
60 000 50 000 40 000 30 000 20 000 10 000 0
50 100 150 200 250
3000 2500 2000 1500 1000 500 0
50 100 150 200 250

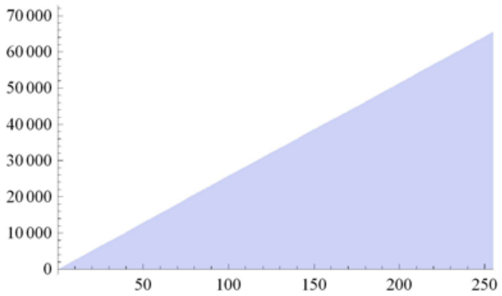
```
histogrammanpassung[randombild, rauschbild];
```

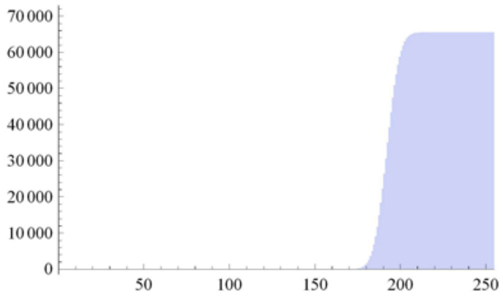



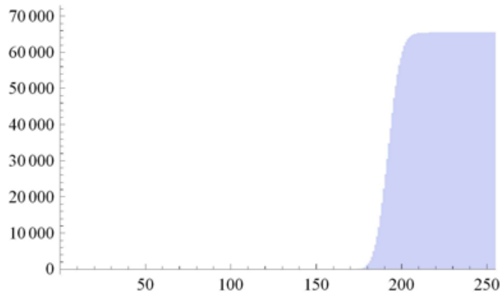





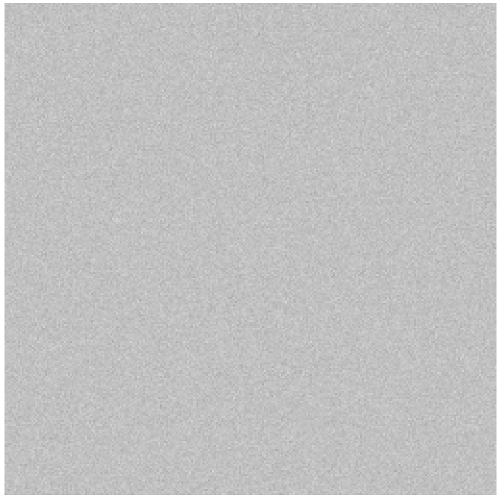
250
200
150
100
50
0
50
100
150
200
250
300
250
200
150
100
50
50
100
150
200
250
4000
3000
2000
1000
0
50
100
150
200
250
4000
3000
2000
1000
0
50
100
150
200
250

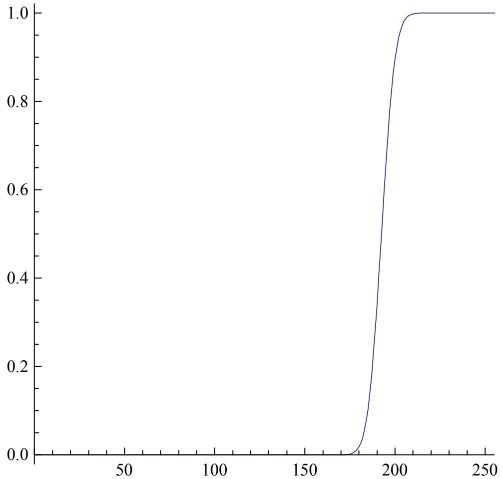


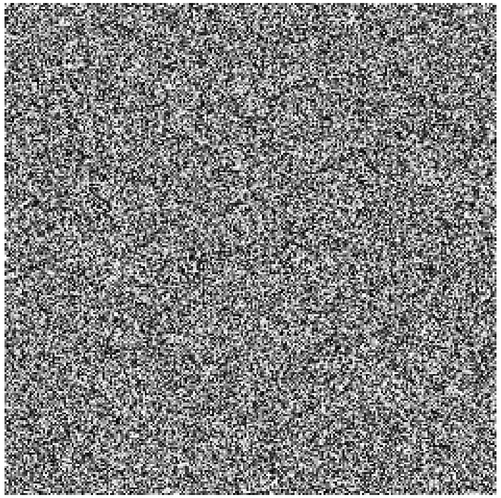


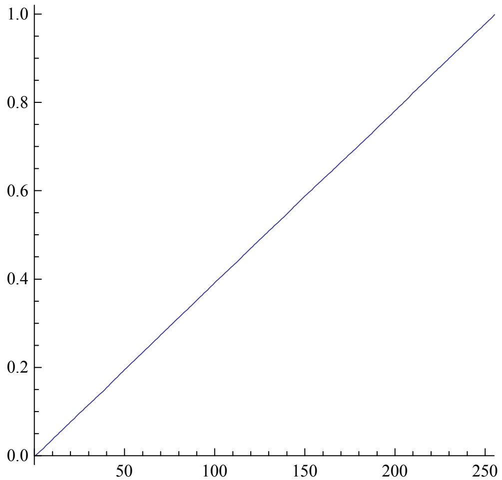


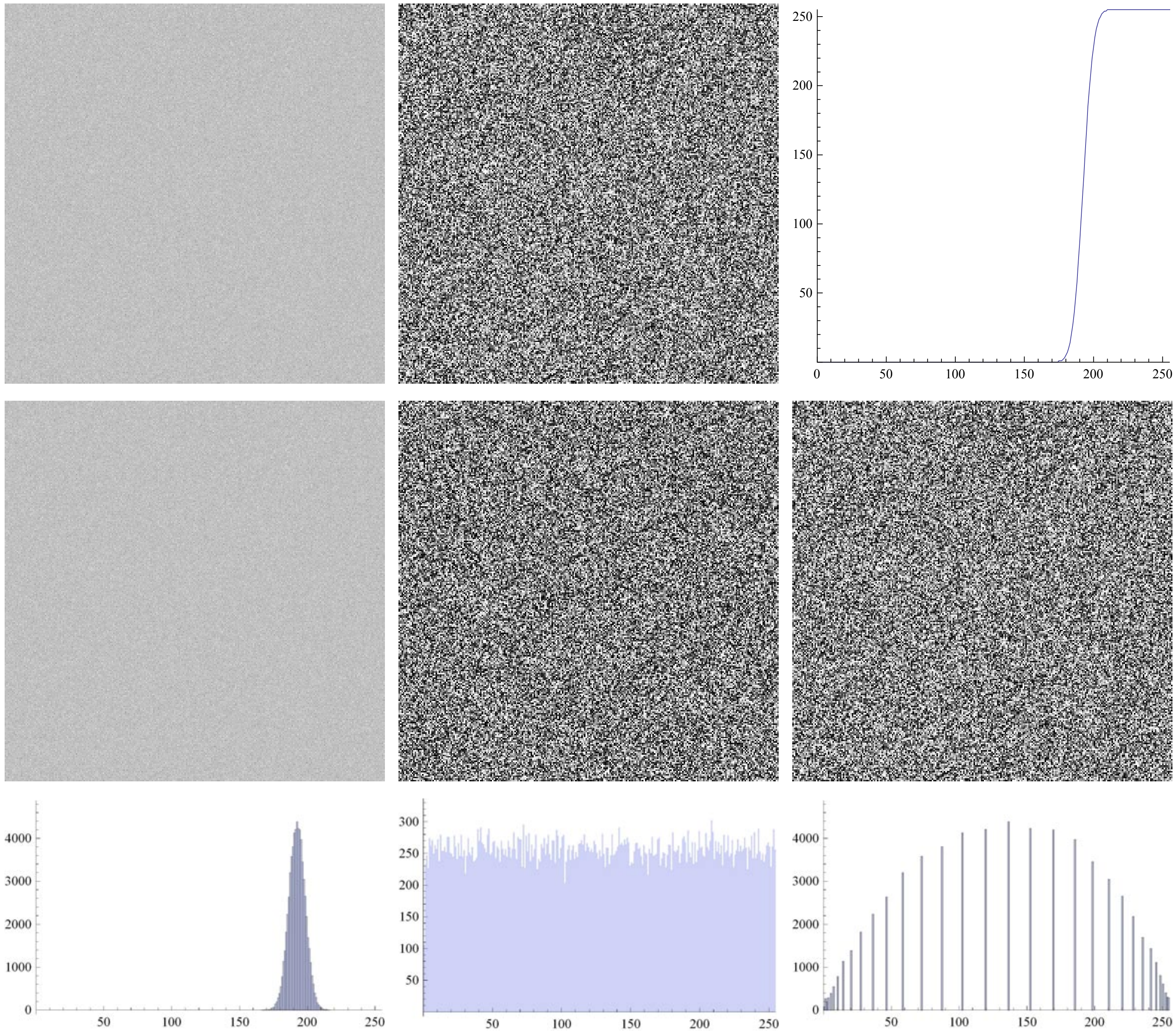

```
histogrammanpassung[rauschbild, randombild];
```

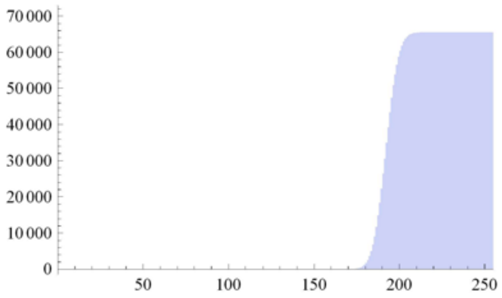


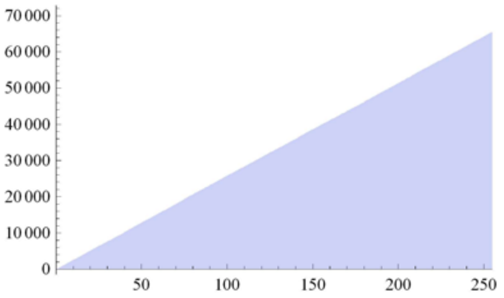


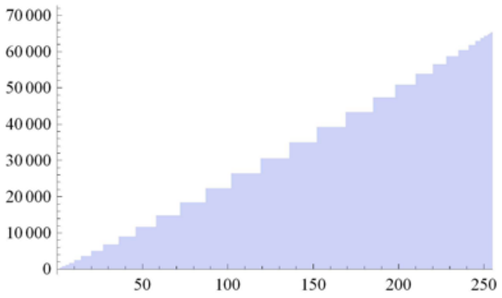




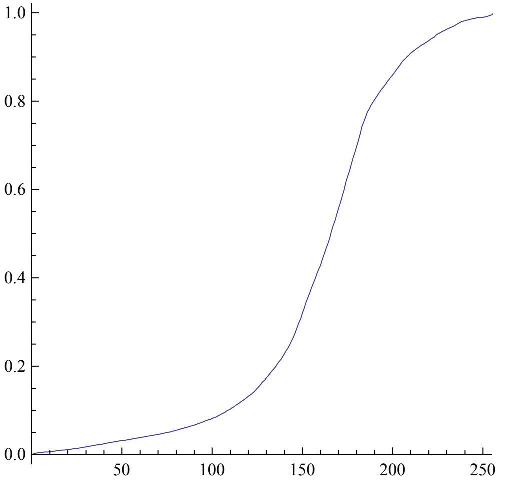

250
200
150
100
50
0 50 100 150 200 250
4000
3000
2000
1000
0
50 100 150 200 250
300
250
200
150
100
50
50 100 150 200 250
4000
3000
2000
1000
0
50 100 150 200 250

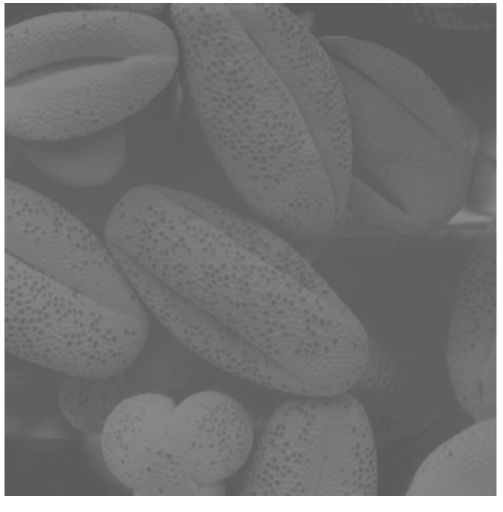


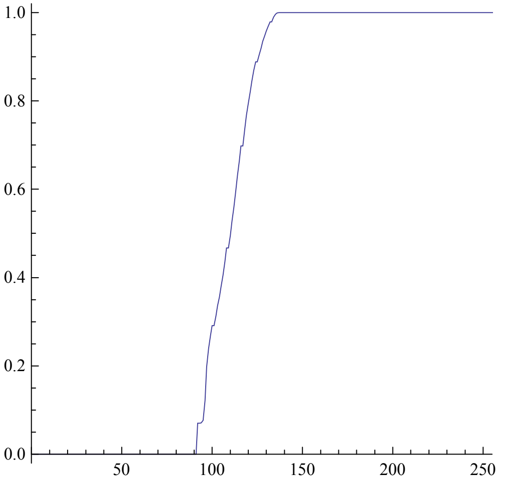


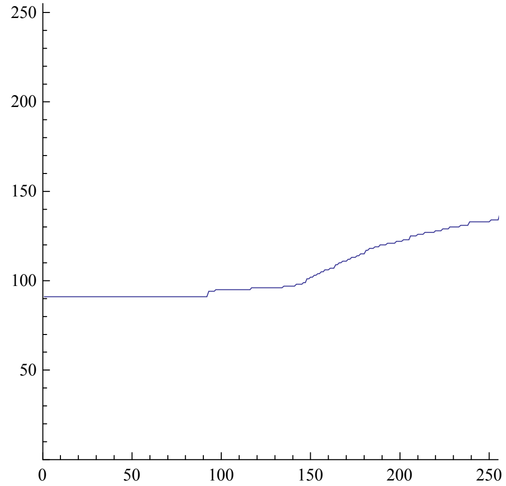


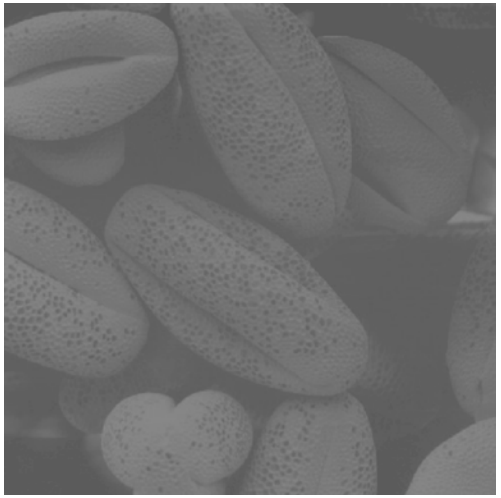

```
histogrammanpassung[histogrammanpassung[ocelot, pollen], ocelot];
```

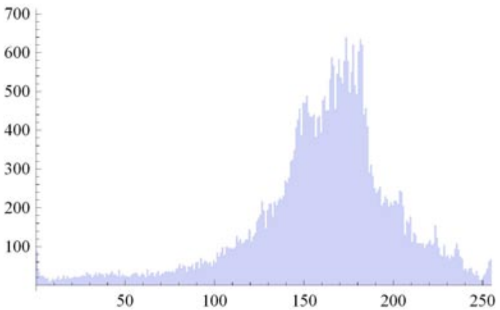


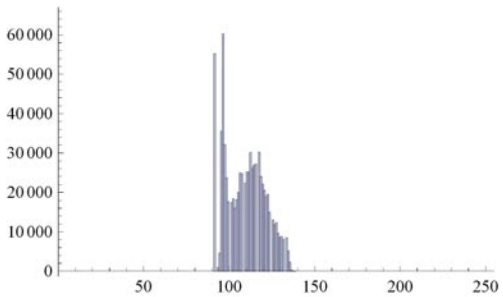


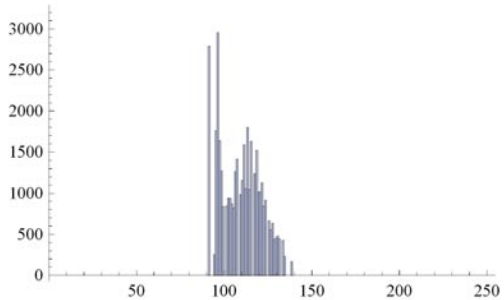


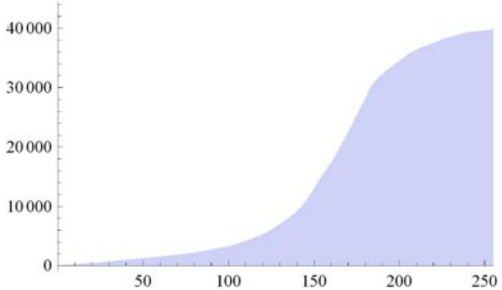

250
200
150
100
50
0
50
100
150
200
250

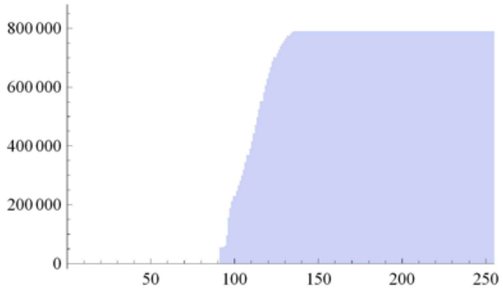


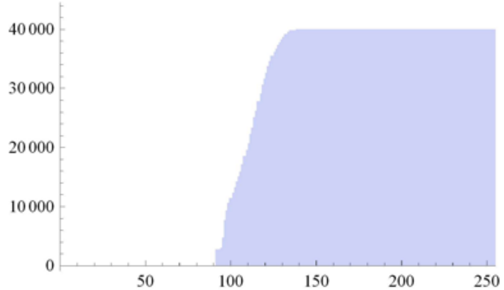

700
600
500
400
300
200
100
50
100
150
200
250

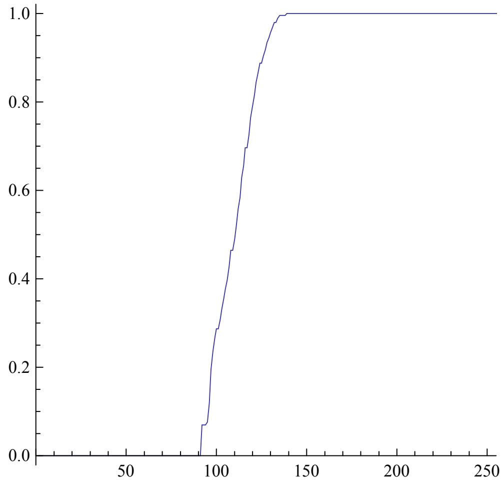

60 000
50 000
40 000
30 000
20 000
10 000
0
50
100
150
200
250

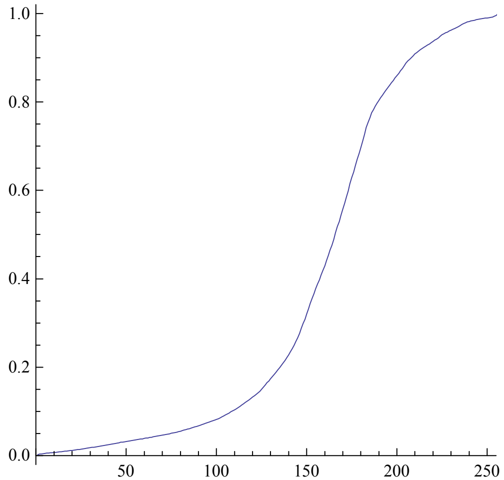

3000
2500
2000
1500
1000
500
0
50
100
150
200
250

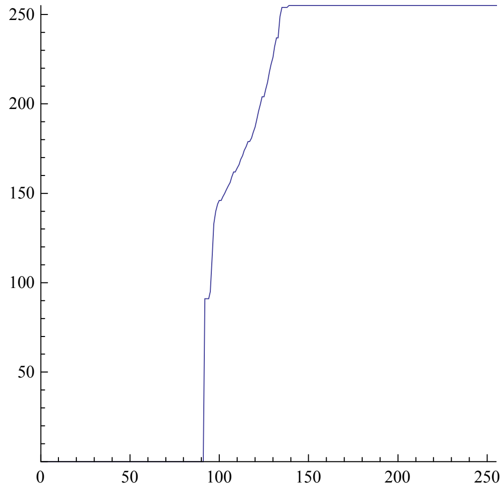

40 000
30 000
20 000
10 000
0
50
100
150
200
250

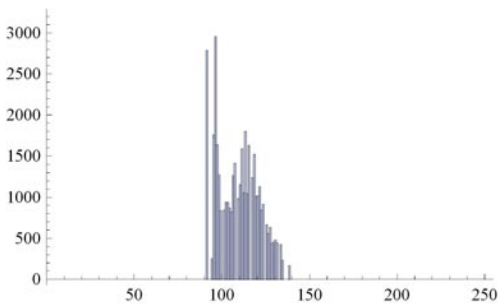

800 000
600 000
400 000
200 000
0
50
100
150
200
250

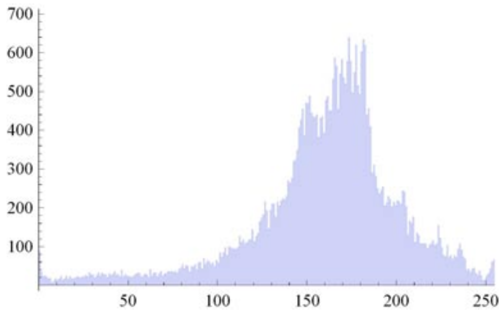

40 000
30 000
20 000
10 000
0
50
100
150
200
250

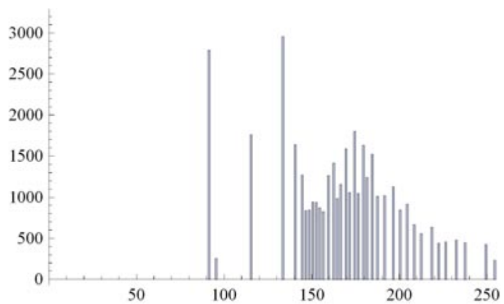

1.0
0.8
0.6
0.4
0.2
0.0
50
100
150
200
250

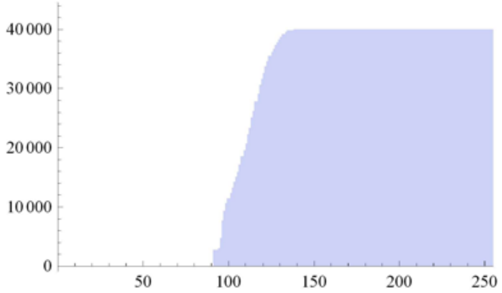

1.0
0.8
0.6
0.4
0.2
0.0
50
100
150
200
250

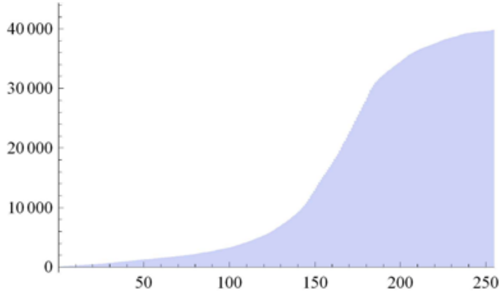

250
200
150
100
50
0
50
100
150
200
250

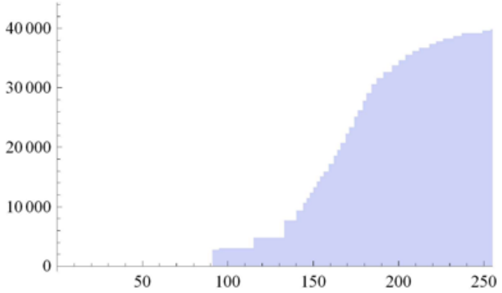

3000
2500
2000
1500
1000
500
0
50
100
150
200
250


700
600
500
400
300
200
100
50
100
150
200
250


3000
2500
2000
1500
1000
500
0
50
100
150
200
250

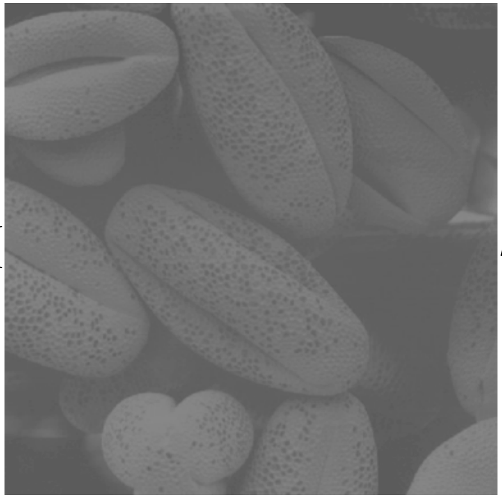




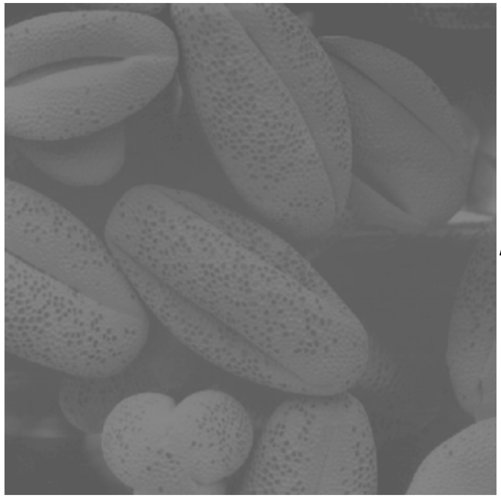


Die Funktionalität der Histogrammanpassung ist in der Funktion HistogramTransform eingebaut:

```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[pollen, ocelot]
```

{, , }

```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[ocelot, pollen]
```

{, , }

HistogramTransform behandelt mehrkanalige Bilder kanalweise:

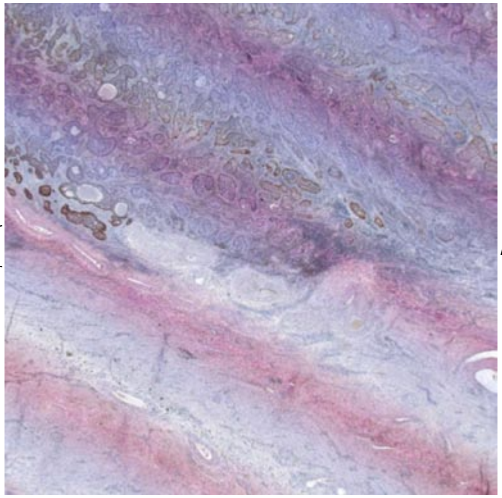
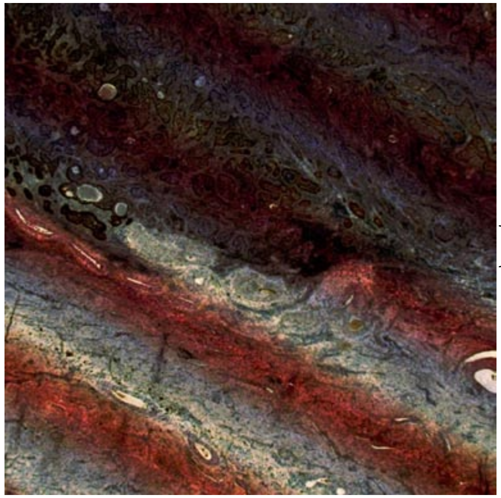
```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[
 ExampleData[{"TestImage", "Girl"}], ExampleData[{"TestImage", "Lena"}]]
```

{, , }

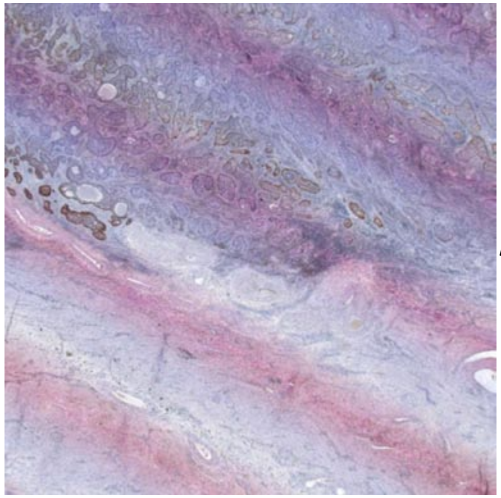
```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[
 ExampleData[{"TestImage", "Lena"}], ExampleData[{"TestImage", "Girl"}]]
```

{, , }

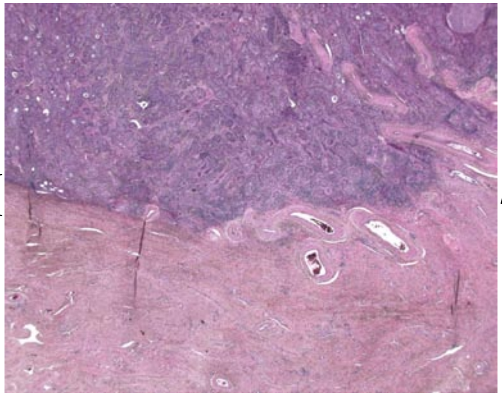
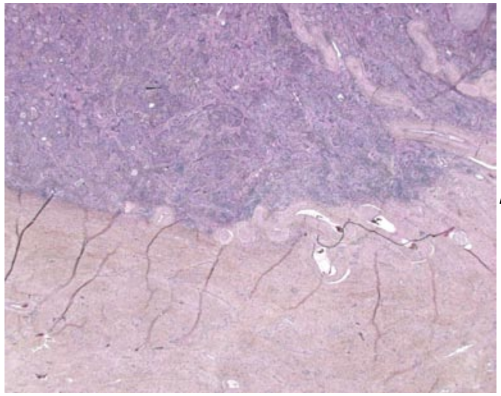
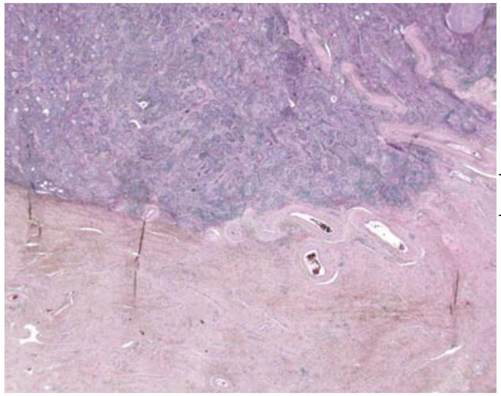
```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[
 testbild, ExampleData[{"TestImage", "Girl"}]]
```

{, , }

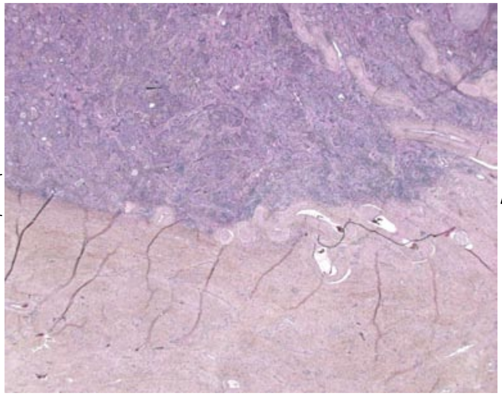
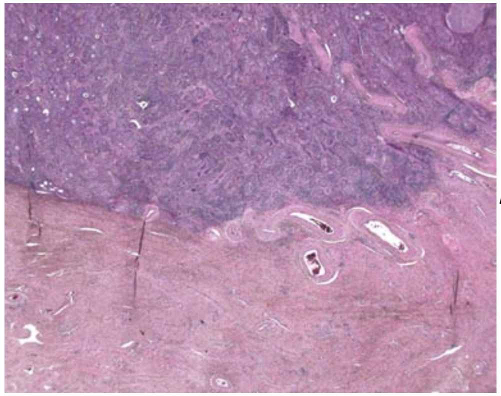
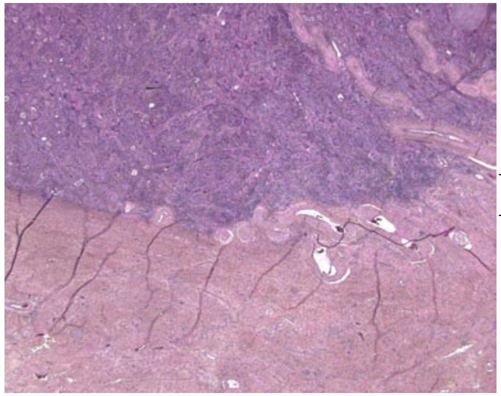
```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[
 ExampleData[{"TestImage", "Girl"}], testbild]
```

{, , }

```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[zervixhe1, zervixhe2]
```

{, , }

```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[zervixhe2, zervixhe1]
```

{, , }

```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[
 ExampleData[{"TestImage", "Girl"}], ExampleData[{"TestImage", "Girl2"}]]
```

{, , }

```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[
 ExampleData[{"TestImage", "Girl2"}], ExampleData[{"TestImage", "Girl"}]]
```

{, , }

```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[
 ExampleData[{"TestImage", "Girl3"}], ExampleData[{"TestImage", "Girl"}]]
```

{, , }

```
ImageMeasurements[ExampleData[{"TestImage", "Girl3"}], "Mean"]
ImageMeasurements[ExampleData[{"TestImage", "Girl"}], "Mean"]
```

{0.506738, 0.389284, 0.490978}

{0.29736, 0.206114, 0.181588}

```
ImageMeasurements[ExampleData[{"TestImage", "Girl3"}], "StandardDeviation"]
ImageMeasurements[ExampleData[{"TestImage", "Girl"}], "StandardDeviation"]
```

{0.245105, 0.210102, 0.15438}

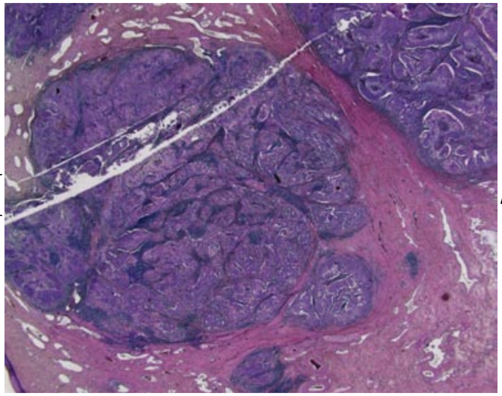
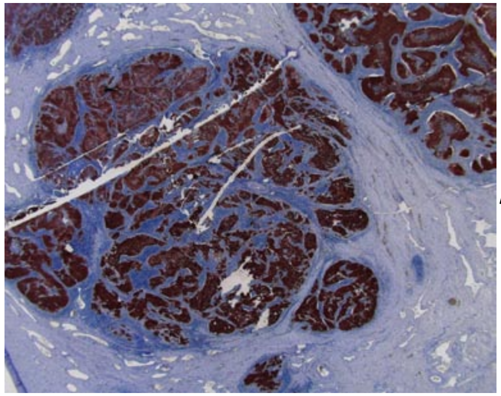
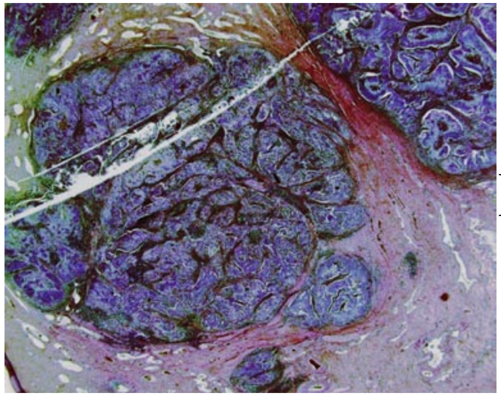
{0.17034, 0.165799, 0.151451}

```
ImageMeasurements[HistogramTransform[ExampleData[{"TestImage", "Girl3"}], ExampleData[{"TestImage", "Girl"}]], "Mean"]
ImageMeasurements[HistogramTransform[ExampleData[{"TestImage", "Girl3"}], ExampleData[{"TestImage", "Girl"}]], "StandardDeviation"]
```

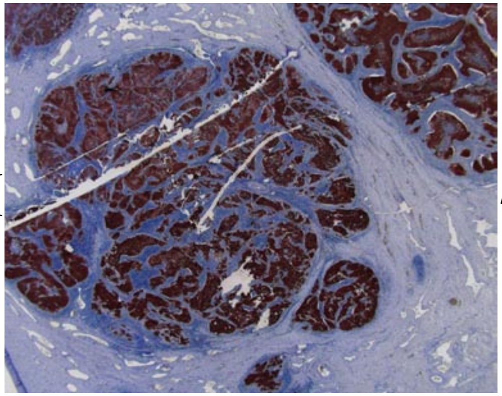
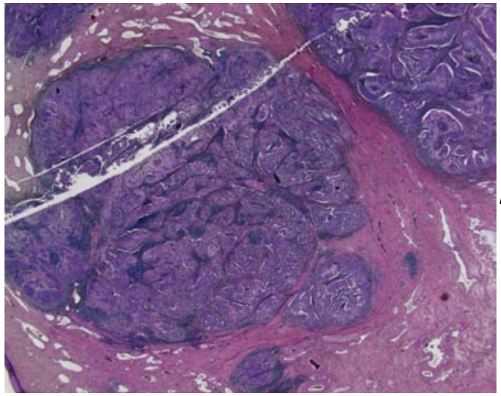
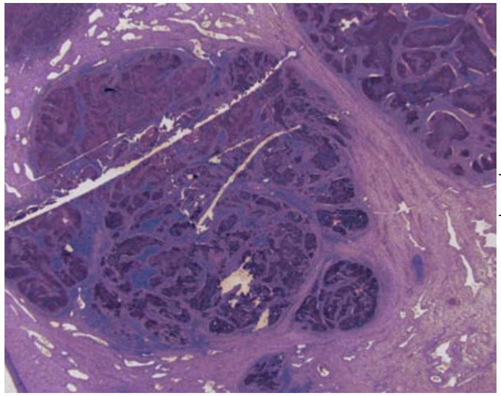
{0.297564, 0.206518, 0.181469}

{0.170103, 0.166532, 0.151109}

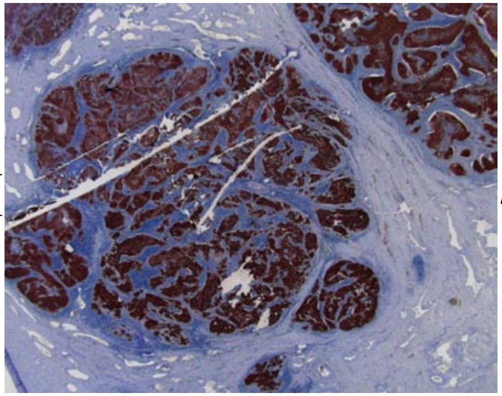
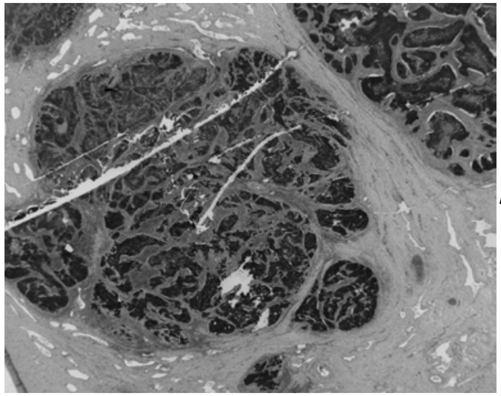
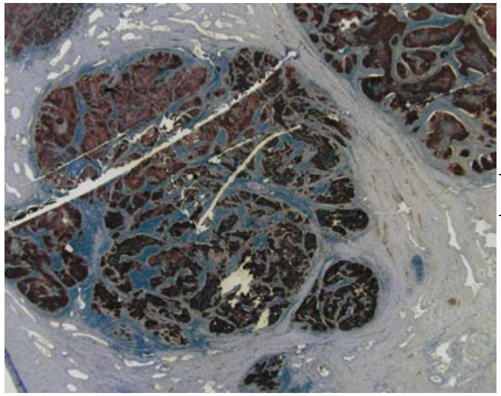
```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[hebild, p16bild]
```

{, , }

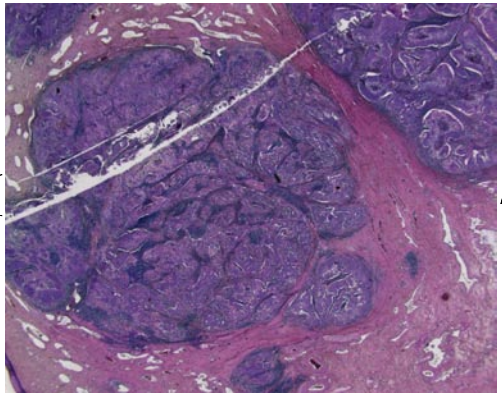
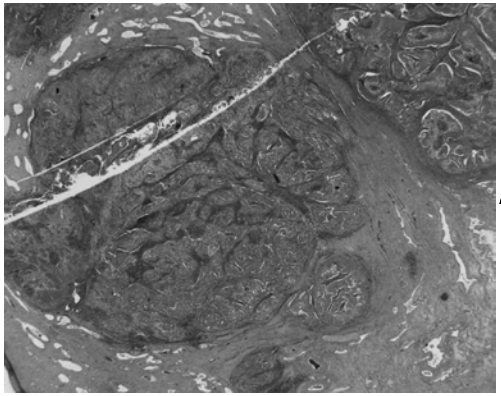
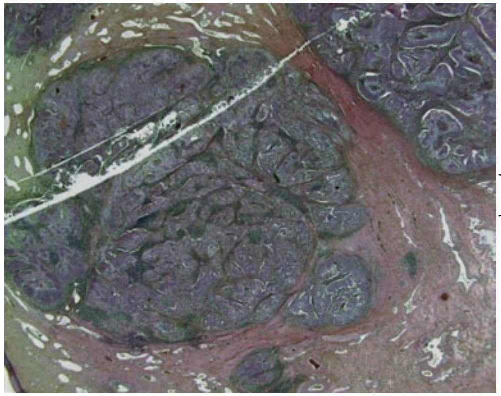
```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[p16bild, hebild]
```

{, , }

```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[
 p16bild, ColorConvert[p16bild, "Graylevel"]]
```

{, , }

```
{Show[#1, ImageSize → 256], Show[#2, ImageSize → 256], Show[HistogramTransform[#1, #2], ImageSize → 256]} &[
 hebild, ColorConvert[hebild, "Graylevel"]]
```

{, , }

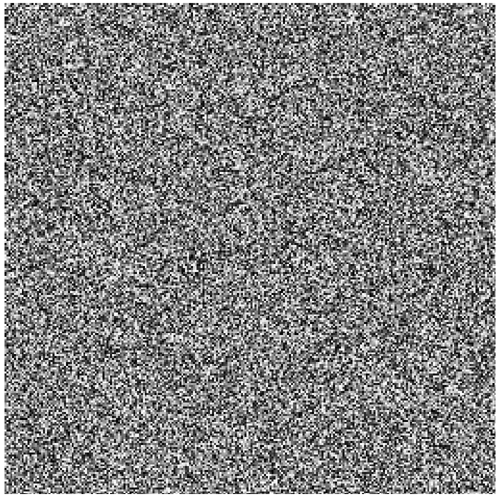
Aus 11. Graylevel-Co-Occurence-Matrix

- Hilfsmittel für die Analyse von texturellen (musterbezogenen) Bildeigenschaften
- geordnete Paare von Intensitätswerten anhand einer definierten binären Relation erfaßt

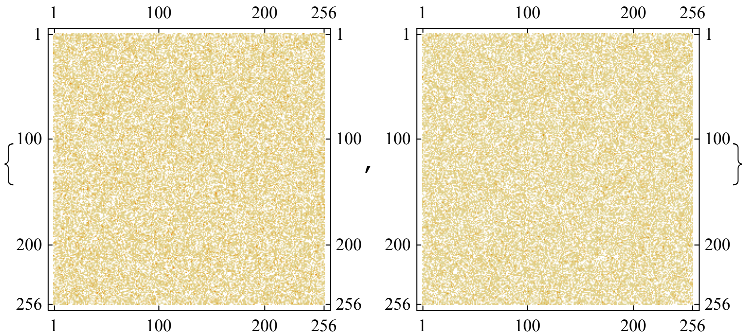
$$C_{\Delta x,\Delta y}(i, j) = \sum_{p=1}^{n} \sum_{q=1}^{m} \begin{cases} 1 & \text{falls } I(p, q) == i \text{ und } I(p + \Delta x, q + \Delta y) == j \\ 0 & \text{sonst} \end{cases}$$

- zusammenhängend besetzte Hauptdiagonalen weisen auf homogene Bildbereiche hin
- Einträge weitab der Hauptdiagonalen deuten auf starke lokale Kontraste hin

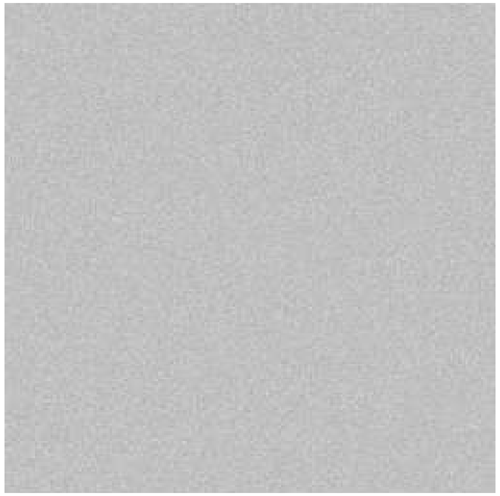
randombild



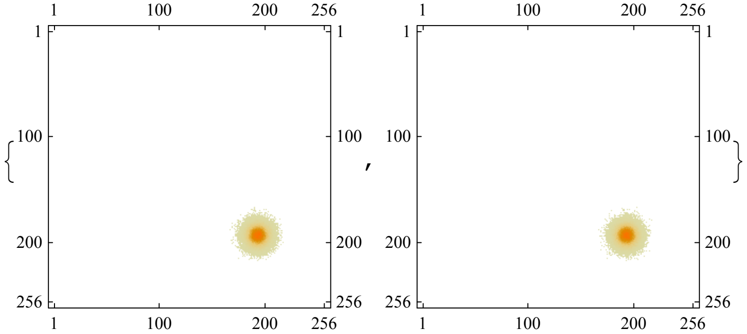
```
{MatrixPlot[ImageCooccurrence[#, 256, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}]],
   MatrixPlot[ImageCooccurrence[#, 256, {{0, 0, 0}, {0, 0, 0}, {0, 1, 0}}]]} &[randombild]
```



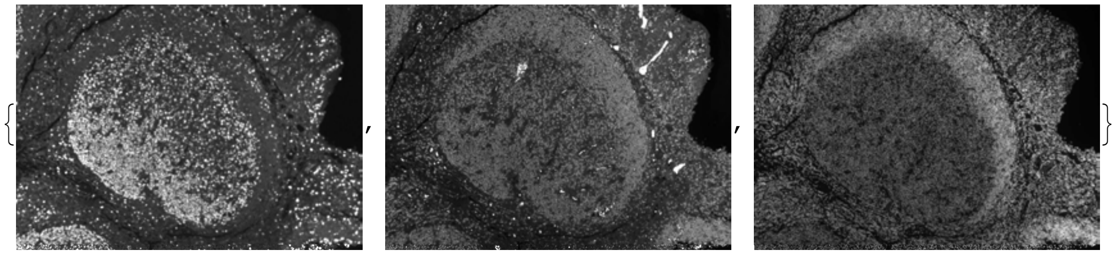
```
rauschbild
```



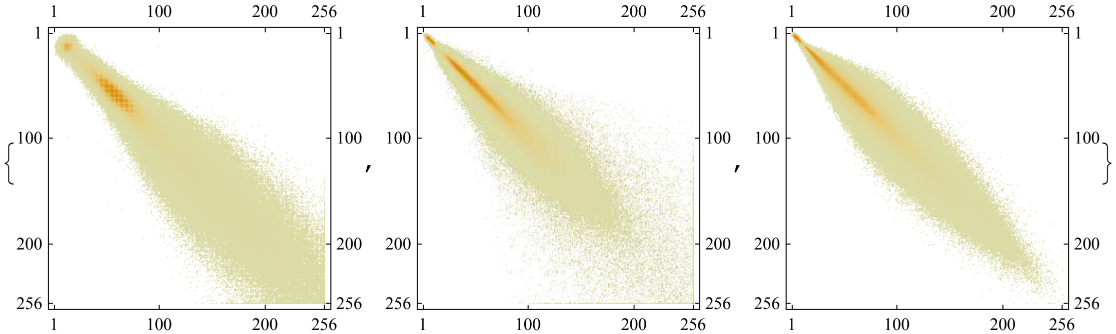
```
{MatrixPlot[ImageCooccurrence[#, 256, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}]],
   MatrixPlot[ImageCooccurrence[#, 256, {{0, 0, 0}, {0, 0, 0}, {0, 1, 0}}]]} &[rauschbild]
```



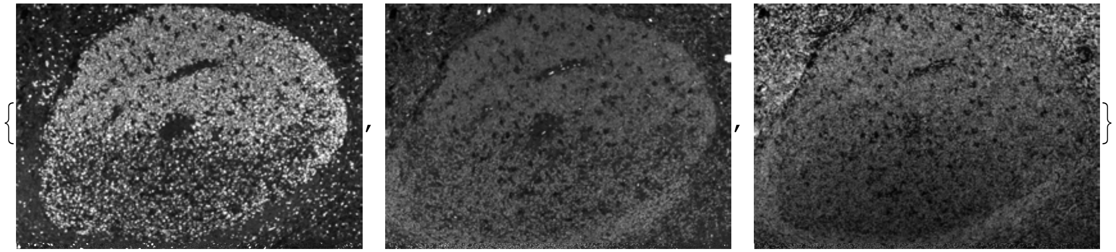

```
ColorSeparate[Ton2CD21dreikanalausgleichF1]
```



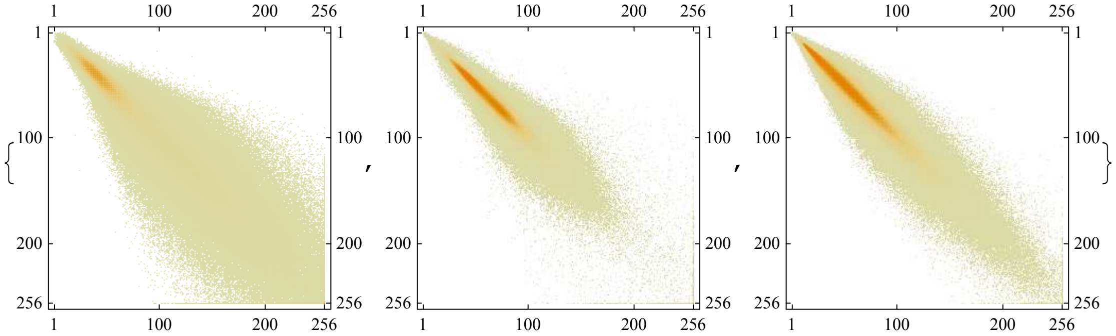
```
MatrixPlot[ImageCooccurrence[#, 256, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}]] & /@ ColorSeparate[Ton2CD21dreikanalausgleichF1]
```



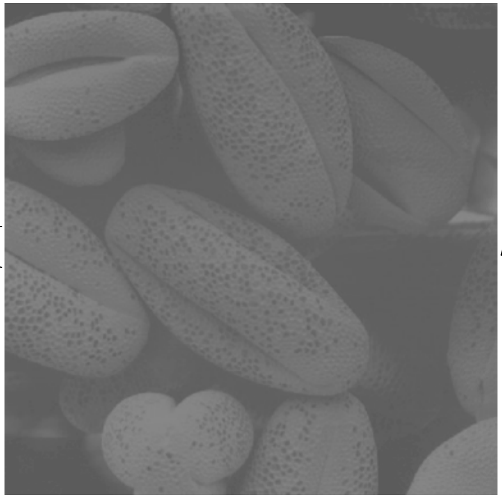
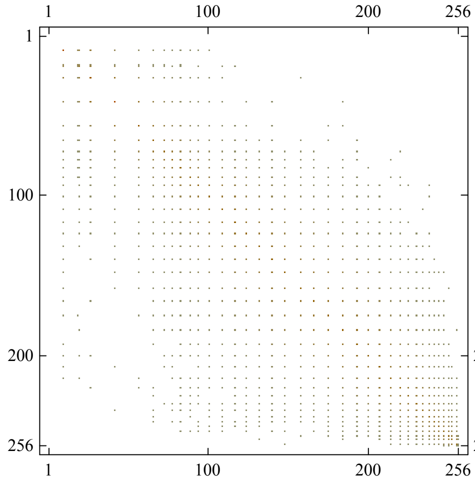
```
ColorSeparate[Ton3CD21dreikanalausgleichF2]
```



```
MatrixPlot[ImageCooccurrence[#, 256, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}]] & /@ ColorSeparate[Ton3CD21dreikanalausgleichF2]
```

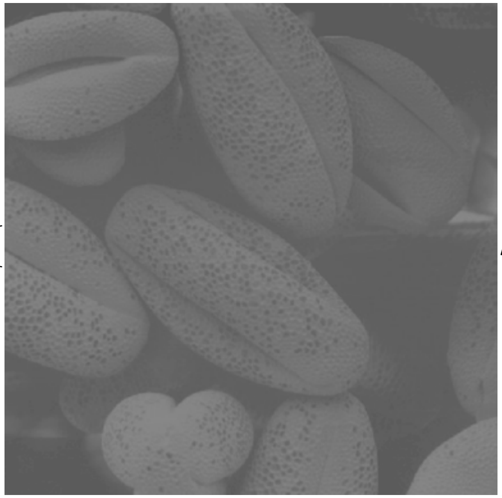
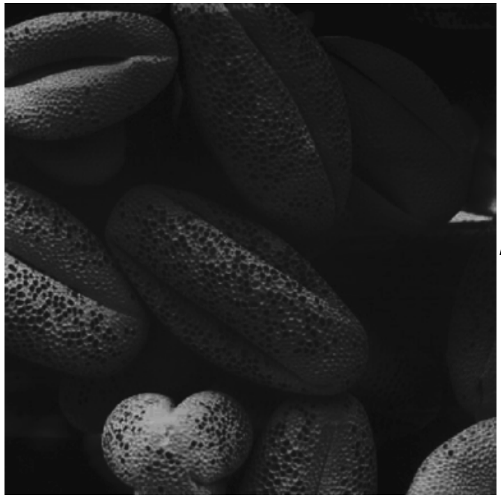
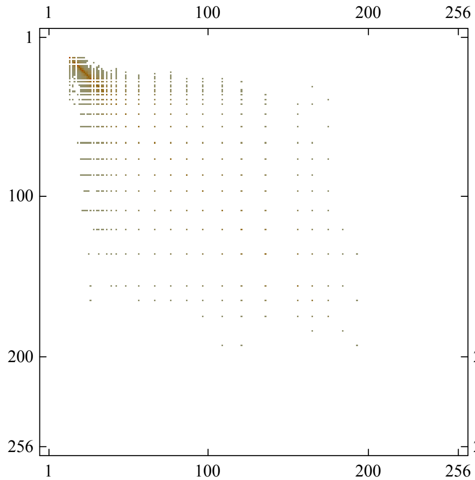


```
Show[#, ImageSize → 256] & /@
 Function[{bild}, Flatten@{bild, {#, MatrixPlot[ImageCooccurrence[#, 256]]} &[HistogramTransform[bild]]}][pollen]
```

{, , 

}

```
Show[#, ImageSize → 256] & /@
 Function[{bild, ref}, Flatten@{bild, {#, MatrixPlot[ImageCooccurrence[#, 256]]} &[HistogramTransform[bild, ref]]}][
  pollen, ExampleData[{"TestImage", "U2"}]]
```

{, , 

}

Analyse der Grauwertübergangsmatrix mittels Haralick'scher Merkmale (insgesamt 14)

Haralick-Features (Implementation von Felix Weiske und Tobias Finze)

```
Manipulate[
 Switch[feat,
  1, {fimg1, fimg2} = {haralick1[img1, graylevelsteps, nbh], haralick1[img2, graylevelsteps, nbh]};
  label = "angular second momentum (energy) = ";
  ,
  2, {fimg1, fimg2} = {haralick2[img1, graylevelsteps, nbh], haralick2[img2, graylevelsteps, nbh]};
  label = "contrast = ";
  ,
  3, {fimg1, fimg2} = {haralick3[img1, graylevelsteps, nbh], haralick3[img2, graylevelsteps, nbh]};
  label = "correlation = ";
  ,
  4, {fimg1, fimg2} = {haralick4[img1, graylevelsteps, nbh], haralick4[img2, graylevelsteps, nbh]};
  label = "variance = ";
  ,
  9, {fimg1, fimg2} = {haralick9[img1, graylevelsteps, nbh], haralick9[img2, graylevelsteps, nbh]};
  label = "entropy = ";
 ];
 Grid[
  {{MatrixPlot[ImageCooccurrence[img1, graylevelsteps, nbh], ImageSize → Medium],
    MatrixPlot[ImageCooccurrence[img2, graylevelsteps, nbh], ImageSize → Medium]},
   {label <> ToString[fimg1], label <> ToString[fimg2]}}
 ],
 {{graylevelsteps, 32, "Grauwertstufen:"}, 1, 256, 1, Appearance → "Labeled"},
 Row[{
   Control[{{img1, rauschbild, "Bild 1:"}, imglist, ImageSize → Tiny, ControlType → PopupMenu}],
   Control[{{img2, randombild, "Bild 2:"}, imglist, ImageSize → Tiny, ControlType → PopupMenu}]
  }, Spacer[20]]
 , {{nbh, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}, "3x3-Nachbarschaft: "},
  {{{0, 0, 0}, {1, 0, 0}, {0, 0, 0}} → "links", {{1, 0, 0}, {0, 0, 0}, {0, 0, 0}} → "links oben", {{0, 1, 0}, {0, 0, 0}, {0, 0, 0}} → "oben",
   {{0, 0, 1}, {0, 0, 0}, {0, 0, 0}} → "rechts oben", {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}} → "rechts", {{0, 0, 0}, {0, 0, 0}, {0, 0, 1}} → "rechts unten",
   {{0, 0, 0}, {0, 0, 0}, {0, 1, 0}} → "unten", {{0, 0, 0}, {0, 0, 0}, {1, 0, 0}} → "links unten"}}, {{feat, 1, "Haralick-Feature: "},
  {1 → "angular second momentum", 2 → "contrast", 3 → "correlation", 4 → "variance", 9 → "entropy"}, ControlType → PopupMenu}]
```

Grauwertstufen: 32

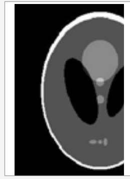
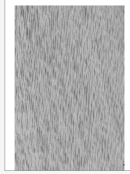
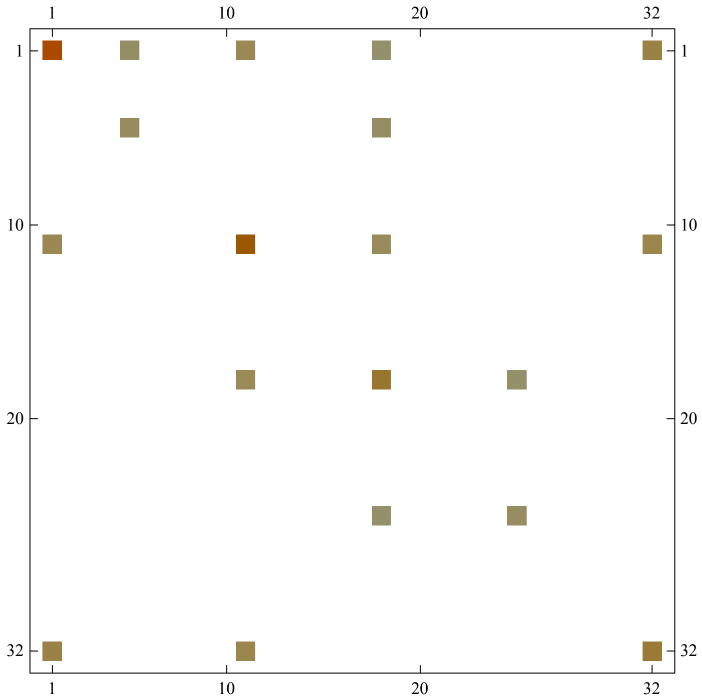
Bild 1: 

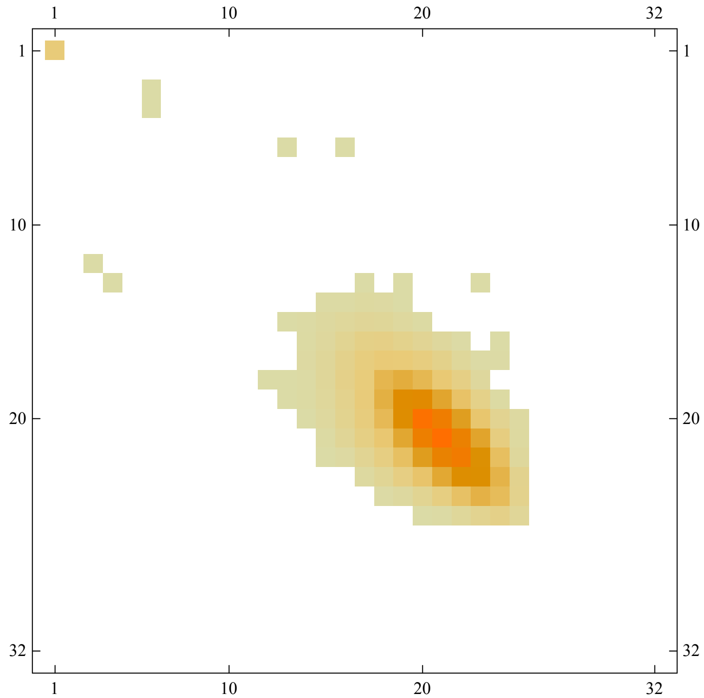
Bild 2: 

3x3–Nachbarschaft: rechts

Haralick–Feature: angular second momentum



angular second momentum (energy) =
haralick1[-Image-, 32, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}]



angular second momentum (energy) =
haralick1[-Image-, 32, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}]

Aus 12. Lineare diskrete verschiebungsinvariante Filter

- als diskrete Faltung: G'=O*G
- $G'_{m,n}=\sum_{i=-r}^{r}\sum_{j=-r}^{r} O_{i,j} G_{m-i,n-j}$

- Verschiebungsinvarianz: wenn Signal verschoben, dann Ergebnis des Operators ebenso verschoben

```
maxwellenzahl = 0.5;
maske = Round[Rescale[Sign[Table[(x^2 + y^2), {x, -256, 256}, {y, -256, 256}] - 256 * (256 + 1)], {1, -1}]];
wellen = Table[Cos[(x^2 + y^2) / 256 * π * maxwellenzahl], {x, -256, 256}, {y, -256, 256}];
wellenbild = Image[Rescale[maske * wellen, {-1, 1}]];

Show[wellenbild, ImageSize → 513]

ListPlot[Table[Cos[x * π * x * maxwellenzahl / 256], {x, 0, 256}], Filling → Axis]

Plot[Cos[x * π * x * maxwellenzahl / 256], {x, 0, 256}]
```

Transferfunktion → Diskrete Fouriertransformation (DFT)

1 D : eindimensionaler Vektor mit (potentiell) komplexen Elementen : $f = (f_0, f_1, \ldots, f_{M-1})^T$

Hintransformation : $\hat{f}_u = \sum_{m=0}^{M-1} f_m \, e^{\left(\frac{-2\pi i m u}{M}\right)}$ mit u = 0 ... M – 1

Rücktransformation : $f_m = \frac{1}{M} \sum_{u=0}^{M-1} \hat{f}_u \, e^{\left(\frac{2\pi i m u}{M}\right)}$ mit m = 0 ... M – 1

Um eine Faltungsmaske in die obige Vektordarstellung zu bringen, ist eine Umsortierung erforderlich :

Maske $(\ldots, f_{-2}, f_{-1}, f_0, f_1, f_2, \ldots)$ wird wegen der negativen Indizes entsprechend der aktuellen Signallänge M zyklisch verschoben in $(f_0, f_1, f_2, \ldots, \ldots, f_{-2}, f_{-1})$ und neu indiziert von 0 ... M – 1 : $(f_0, f_1, f_2, \ldots, f_{M-2}, f_{M-1})$

Beispiel : Binomialfilter

$$b = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)^T$$

$$b_0 = \frac{1}{2}$$

$$b_1 = \frac{1}{4}$$

$$b_{M-1} = \frac{1}{4}$$

$$\hat{b}_u = \frac{1}{2} + \frac{1}{4} e^{\left(\frac{-2\pi i u}{M}\right)} + \frac{1}{4} e^{\left(\frac{-2\pi i (M-1) u}{M}\right)}$$

$$\hat{b}_u = \frac{1}{2} + \frac{1}{4} e^{\left(\frac{-2\pi i u}{M}\right)} + \frac{1}{4} e^{\left(-2 i \pi u + \frac{2 i \pi u}{M}\right)}$$

$$\hat{b}_u = \frac{1}{2} + \frac{1}{4} e^{\left(\frac{-2\pi i u}{M}\right)} + \frac{1}{4} e^{(-2 i \pi u)} e^{\left(\frac{2 i \pi u}{M}\right)}$$

```
Simplify[e^(-2 i π u), Element[u, Integers]]
1
```

```
Solve[e^(-2 i π u) == 1, u]
{{u → ConditionalExpression[-C[1], C[1] ∈ Integers]}}
```

$$\hat{b}_u = \frac{1}{2} + \frac{1}{4} e^{\left(\frac{-2\pi i u}{M}\right)} + \frac{1}{4} e^{\left(\frac{2 i \pi u}{M}\right)}$$

$e^{\left(\frac{-2 \pi i u}{M}\right)}$ // ExpToTrig

$\text{Cos}\left[\frac{2 \pi u}{M}\right] - i \text{Sin}\left[\frac{2 \pi u}{M}\right]$

$e^{\left(\frac{2 i \pi u}{M}\right)}$ // ExpToTrig

$\text{Cos}\left[\frac{2 \pi u}{M}\right] + i \text{Sin}\left[\frac{2 \pi u}{M}\right]$

(*Additionstheorem cosx-isinx+cosx+isinx*)

$e^{\left(\frac{-2 \pi i u}{M}\right)} + e^{\left(\frac{2 i \pi u}{M}\right)}$ // ExpToTrig

$2 \text{Cos}\left[\frac{2 \pi u}{M}\right]$

$e^{-i x} + e^{i x}$ // ExpToTrig

2 Cos[x]

$1/2 \left(1 + 1/2\, e^{\left(\frac{-2 \pi i u}{M}\right)} + 1/2\, e^{\left(\frac{2 i \pi u}{M}\right)}\right)$ // ExpToTrig

$\frac{1}{2} + \frac{1}{2} \text{Cos}\left[\frac{2 \pi u}{M}\right]$

$$\hat{b}_{\text{u}} = \frac{1}{2}\left(\cos\left(\frac{2 \pi u}{M}\right) + 1\right)$$

$$\hat{b}_u = \frac{1}{2}\left(\cos(2 \pi k) + 1\right)$$

$\frac{1}{2} + \frac{1}{2} \text{Cos}\left[\frac{2 \pi u}{M}\right]$ // Simplify

$\text{Cos}\left[\frac{\pi u}{M}\right]^2$

Das Verhältnis u / M wird als Wellenzahl k = 0 ... 0.5 bezeichnet

```
1/2 + 1/2 Cos[2 π k] // Simplify
```

```
Cos[k π]^2
```

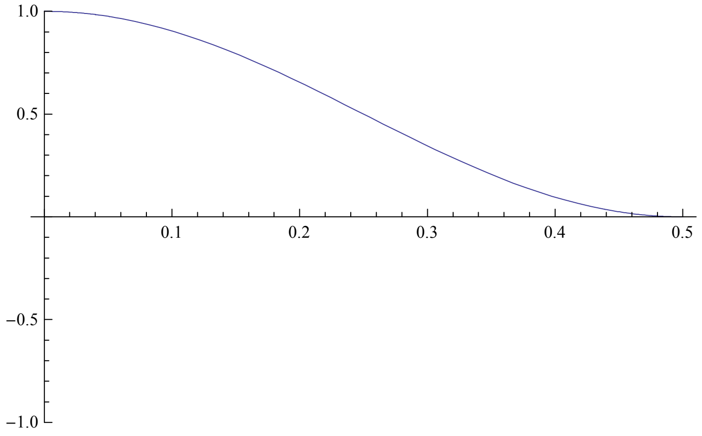
Nullstelle(n) der Transferfunktion $\hat{g}_k$ des Boxfilters $g = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)^T$

```
Solve[1 / 2 * (1 + Cos[2 π k]) == 0 && 0 ≤ k ≤ 1 / 2, k]
```

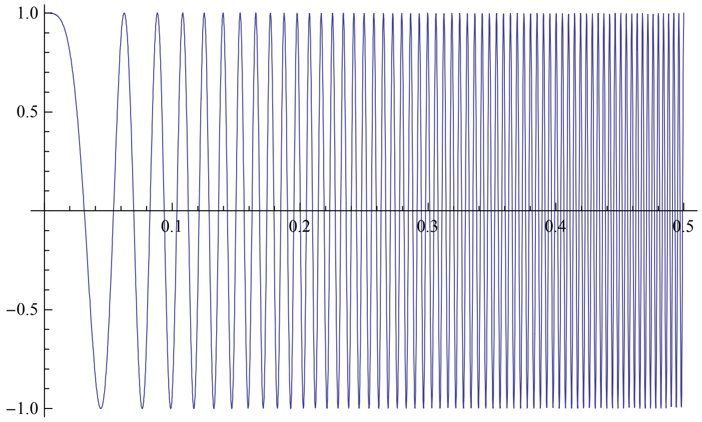
$$\left\{\left\{k \to \frac{1}{2}\right\}, \left\{k \to \frac{1}{2}\right\}\right\}$$

Plot der Transferfunktion $\hat{g}_k$ des Boxfilters $g = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)^T$

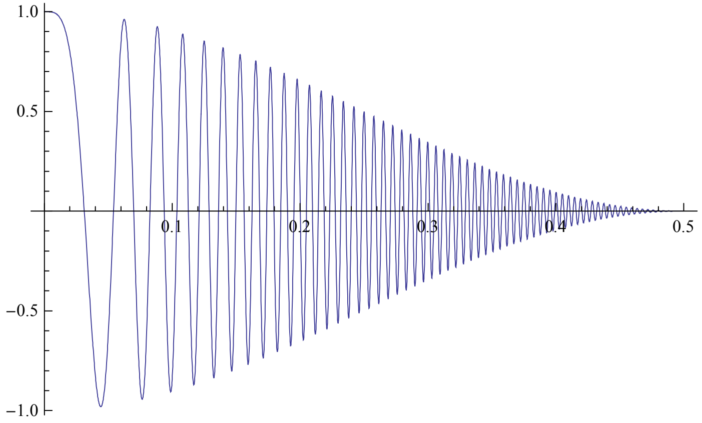
```
Plot[1 / 2 * (1 + Cos[2 π k]), {k, 0, maxwellenzahl}, PlotRange → {Full, {-1, 1}}]
```



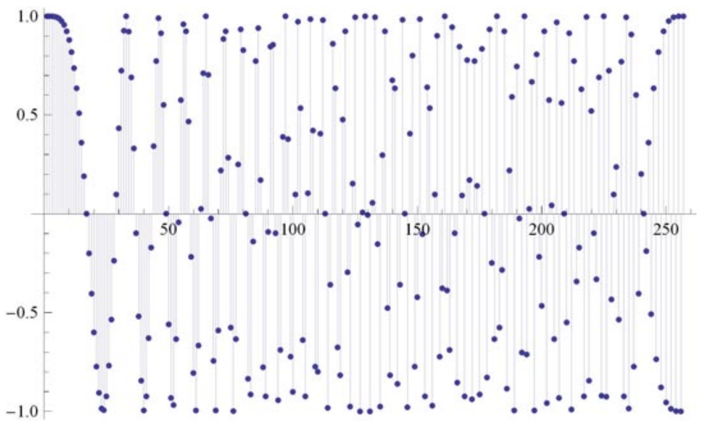
```
Plot[Cos[x * π * x / maxwellenzahl * r], {x, 0, maxwellenzahl}]
```



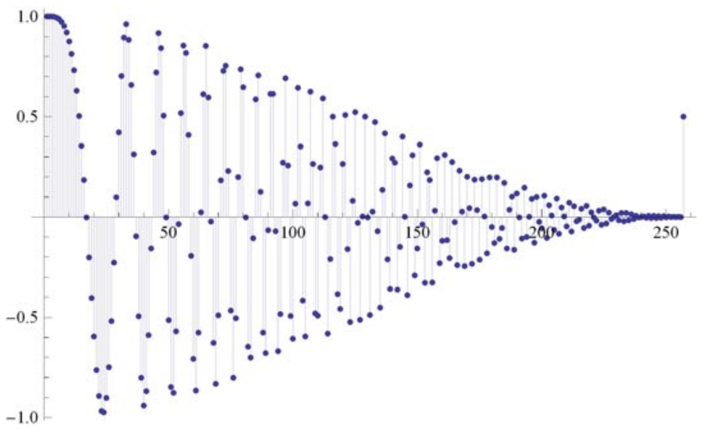

```
Plot[Cos[k * π * k / maxwellenzahl * r] * (1 / 2 * (1 + Cos[2 π k])), {k, 0, maxwellenzahl}]
```



```
ListPlot[Table[Cos[x * π * x * maxwellenzahl / r], {x, 0, r}], Filling → Axis]
```




```
ListPlot[ListConvolve[{1 / 4, 1 / 2, 1 / 4}, Table[Cos[x * π * x * maxwellenzahl / r], {x, 0, r}], 2], Filling → Axis]
```



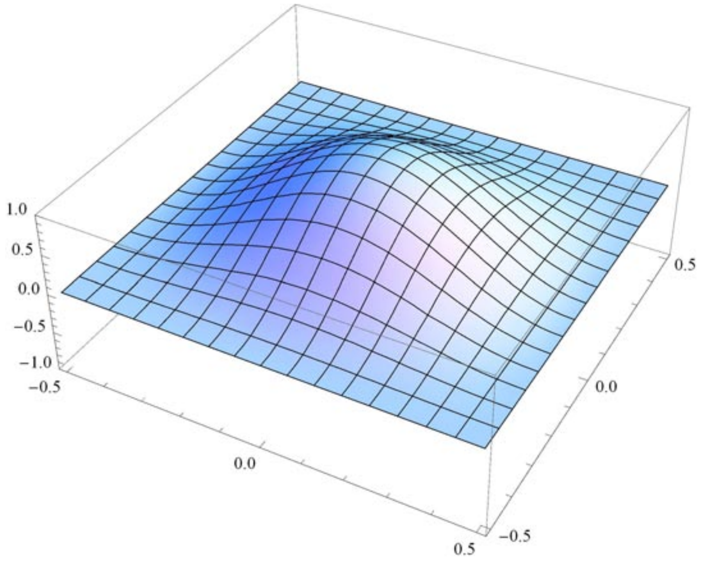
Wie sieht das in 2D für das Ringwellenbild aus?

2 D – Hintransformation : $\hat{f}_{u,v} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n}\, e^{\left(\frac{-2\pi i m u}{M}\right)} e^{\left(\frac{-2\pi i n v}{N}\right)}$ mit u = 0 ... M – 1 und v = 0 ... N – 1

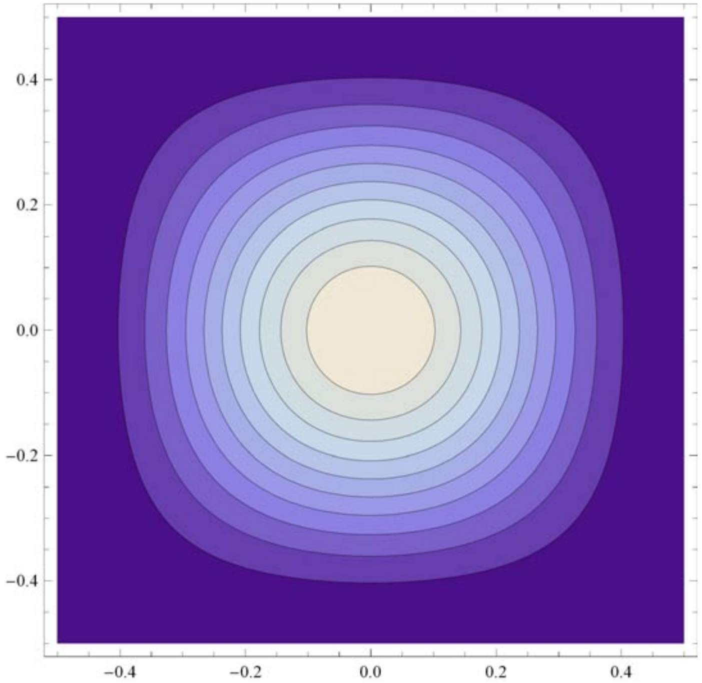
2 D – Rücktransformation : $f_{m,n} = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}_{u,v}\, e^{\left(\frac{2\pi i m u}{M}\right)} e^{\left(\frac{2\pi i n v}{N}\right)}$ mit m = 0 ... M – 1 und n = 0 ... N – 1

Die Verhältnisse u / M bzw. v / N werden als Wellenzahl $k_1 = 0 \ldots \quad 0.5$ bzw. $k_2 = 0 \ldots \quad 0.5$ bezeichnet.

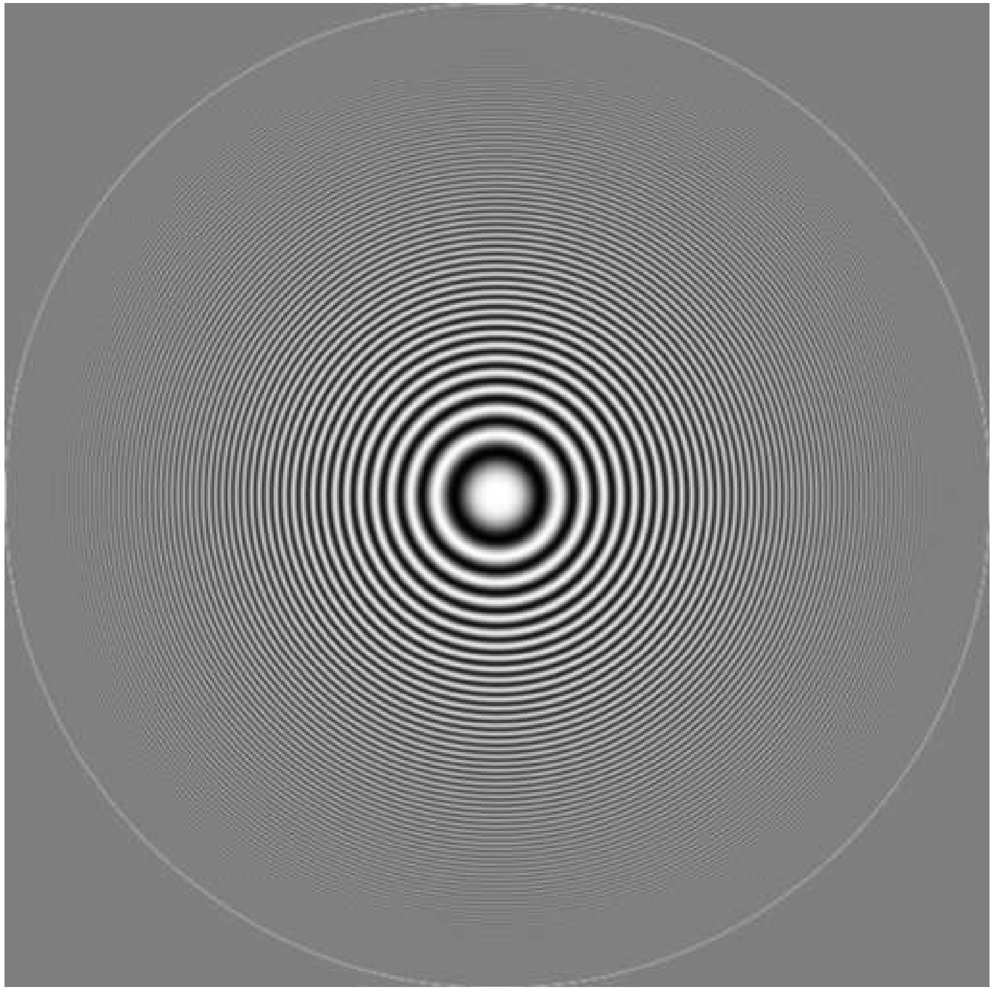
```
Plot3D[1 / 2 (1 + Cos[2 k1 π]) * 1 / 2 (1 + Cos[2 k2 π]), {k1, -maxwellenzahl, maxwellenzahl},
 {k2, -maxwellenzahl, maxwellenzahl}, PlotRange → {Full, Full, {-1, 1}}]
```



```
ContourPlot[1 / 2 (1 + Cos[2 k1 π]) * 1 / 2 (1 + Cos[2 k2 π]),
 {k1, -maxwellenzahl, maxwellenzahl}, {k2, -maxwellenzahl, maxwellenzahl}, Contours → 10]
```



```
Image[ListConvolve[Transpose[{{1 / 4, 1 / 2, 1 / 4}}].{{1 / 4, 1 / 2, 1 / 4}}, ImageData[wellenbild], 2]]
```



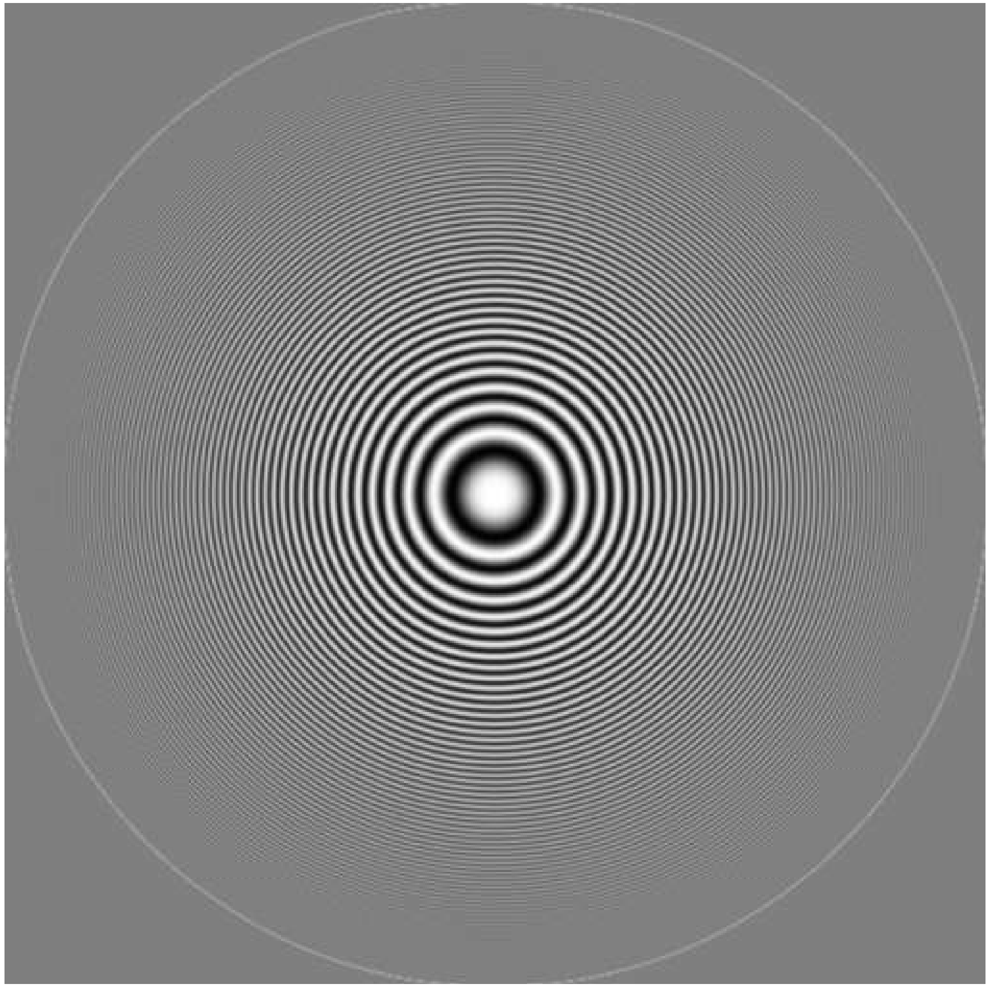
Separierbarkeit

Ein diskretes Filter in 2D, das als äußeres Produkt zweier 1D-Filter gebildet werden kann, kann ebenso durch sequentielle Anwendung beider 1D-Filter angewendet werden. Die Matrix des “Filterkerns” hat den Rang 1. Dadurch kommen anstelle von je N^2 Lesezugriffen und Multiplikationen sowie $N^2 - 1$ Additionen nur $2\,N$ Lesezugriffe und Multiplikationen sowie $2\,(N - 1)$ Additionen zustande.

```
MatrixRank@(Transpose[{{1 / 4, 1 / 2, 1 / 4}}].{{1 / 4, 1 / 2, 1 / 4}})
```

```
1
```

```
Image[ListConvolve[{{1 / 4, 1 / 2, 1 / 4}}, ListConvolve[Transpose@{{1 / 4, 1 / 2, 1 / 4}}, ImageData[wellenbild], {2, 1}], {2, 1}]]
```



```
ClearSystemCache[];
Image[ListConvolve[Transpose[{{1 / 4, 1 / 2, 1 / 4}}].{{1 / 4, 1 / 2, 1 / 4}}, ImageData[wellenbild], {2, 2}]] // AbsoluteTiming // First
```

0.031200

```
ClearSystemCache[];
Image[ListConvolve[N@{{1 / 4, 1 / 2, 1 / 4}}, ListConvolve[Transpose@{{1 / 4, 1 / 2, 1 / 4}}, ImageData[wellenbild], {2, 1}], {2, 1}]] //
  AbsoluteTiming // First
```

0.046800

Leider ist der Rechenzeitgewinn durch Separierbarkeit damit nicht demonstrierbar.

```
ClearSystemCache[];
ImageConvolve[wellenbild, Transpose[{{1 / 4, 1 / 2, 1 / 4}}].{{1 / 4, 1 / 2, 1 / 4}}] // AbsoluteTiming // First
```

```
0.140400
```

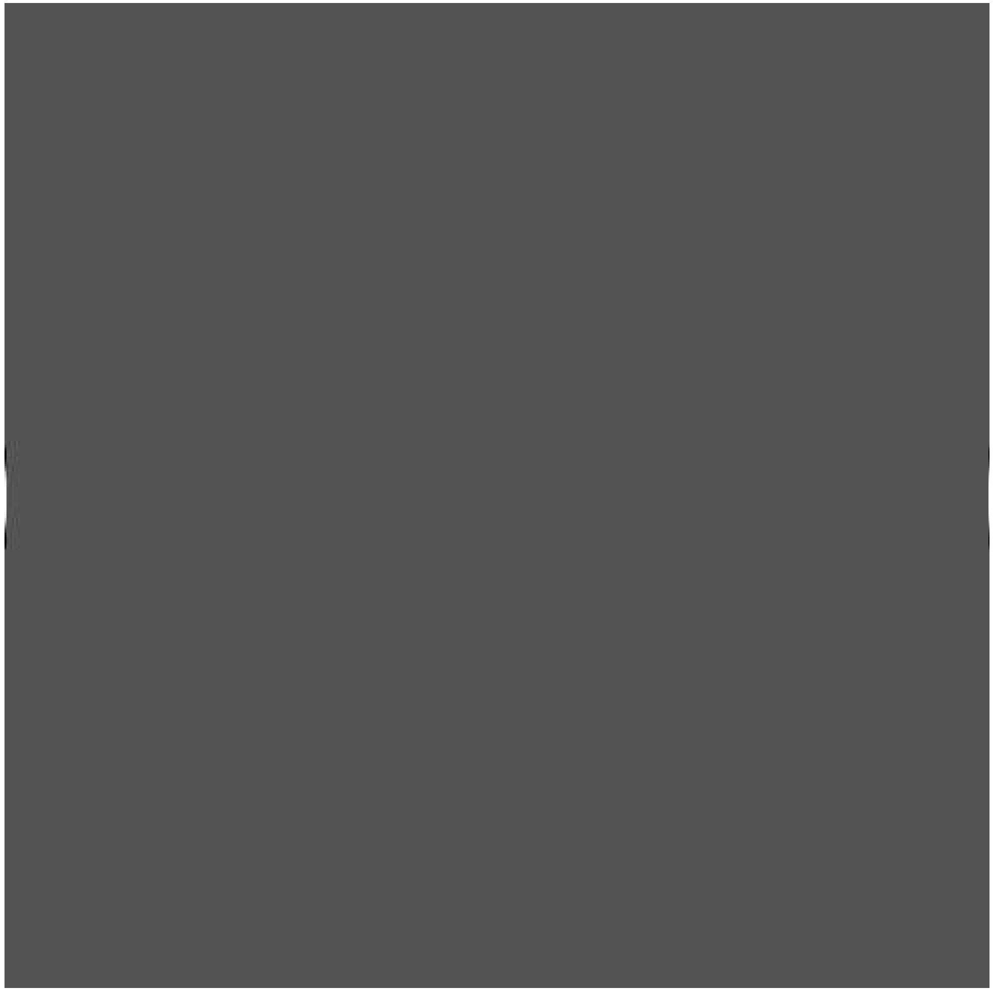
```
ClearSystemCache[];
ImageConvolve[ImageConvolve[wellenbild, Transpose@{{1 / 4, 1 / 2, 1 / 4}}], {{1 / 4, 1 / 2, 1 / 4}}] // AbsoluteTiming // First
```

```
0.062400
```

Jedoch mit ImageConvolve kann man den Gewinn belegen (dennoch langsamer, als oben)

Aber die Gleichheit der Lösung (bis auf Stellen am äußersten Rand):

```
Image@Rescale[ListConvolve[Transpose[{{1 / 4, 1 / 2, 1 / 4}}].{{1 / 4, 1 / 2, 1 / 4}}, ImageData[wellenbild], 2] -
   ListConvolve[{{1 / 4, 1 / 2, 1 / 4}}, ListConvolve[Transpose@{{1 / 4, 1 / 2, 1 / 4}}, ImageData[wellenbild], {2, 1}], {1, 2}]]
```



Automatisierte analytische Berechnung der 1D-DFT

```
Clear[transferfunktion];
transferfunktion[g_List] := Module[{resultat, u, mm, m},
   resultat = (FullSimplify[Sum[g[[Mod[m + (Length[g] - 1) / 2, Length[g]] + 1]] *
         Exp[-2 π ⅈ * Join[Table[i, {i, 0, (Length[g] - 1) / 2, 1}], Table[mm - i, {i, (Length[g] - 1) / 2, 1, -1}]][[m + 1]] u / mm],
        {m, 0, Length[g] - 1, 1}], Assumptions → Element[u, Integers]] /. u / mm → k1);
   Return[resultat];
  ];
```

Automatisierte analytische Berechnung der 2D-DFT

```
Clear[transferfunktion2d];
(*schnellere Variante -(M-1)/2 ... (M-1)/2 und -(N-1)/2 ... (N-1)/2*)
transferfunktion2d[g_List] := Module[{resultat, u, v, mm, nn, m, n},
   resultat =
    ((
       FullSimplify[
         FullSimplify[
          Sum[
           Sum[
            g[[m + (Dimensions[g][[1]] + 1) / 2, n + (Dimensions[g][[2]] + 1) / 2]] *
             Exp[-2 π ⅈ m * u / mm] *
             Exp[-2 π ⅈ n * v / nn],
            {m, - (Dimensions[g][[1]] - 1) / 2, (Dimensions[g][[1]] - 1) / 2, 1}],
           {n, - (Dimensions[g][[2]] - 1) / 2, (Dimensions[g][[2]] - 1) / 2, 1}],
          Assumptions → Element[u, Integers]],
         Assumptions → Element[v, Integers]]
        /. u / mm → k1)
      /. v / nn → k2
    );
   Return[FullSimplify[ComplexExpand[ExpToTrig[resultat]] /. u / mm → k1 /. v / nn → k2]];
  ];
```

Beispiele 1D-DFT:

3er Binomialfilter

```
binom[2]
transferfunktion[binom[2]]
transferfunktion[binom[2]] // TrigReduce
```

$$\left\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right\}$$

$$\mathrm{Cos}[\pi\, k_1]^2$$

$$\frac{1}{2}\left(1 + \mathrm{Cos}[2\,\pi\, k_1]\right)$$

3er Boxfilter

```
box[3]
transferfunktion[box[3]] // TrigReduce
```

$$\left\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right\}$$

$$\frac{1}{3}\left(1 + 2\,\mathrm{Cos}[2\,\pi\, k_1]\right)$$

2er Boxfilter

```
transferfunktion[{1 / 2, 1 / 2, 0}] // ExpToTrig
```

$$\frac{1}{2} + \frac{1}{2}\mathrm{Cos}[2\,\pi\, k_1] + \frac{1}{2}\, i\, \mathrm{Sin}[2\,\pi\, k_1]$$

```
transferfunktion[{0, 1 / 2, 1 / 2}] // ExpToTrig
```

$$\frac{1}{2} + \frac{1}{2}\mathrm{Cos}[2\,\pi\, k_1] - \frac{1}{2}\, i\, \mathrm{Sin}[2\,\pi\, k_1]$$

```
binom[4]
transferfunktion[binom[4]]
transferfunktion[binom[4]] // TrigReduce
```

$\left\{\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right\}$

$\text{Cos}[\pi\, k_1]^4$

$\frac{1}{8}\,(3 + 4\,\text{Cos}[2\,\pi\, k_1] + \text{Cos}[4\,\pi\, k_1])$

```
ableitungsymm
transferfunktion[ableitungsymm] // TrigReduce
```

$\left\{\frac{1}{2}, 0, -\frac{1}{2}\right\}$

$i\,\text{Sin}[2\,\pi\, k_1]$

```
binom[10]
transferfunktion[binom[10]]
transferfunktion[binom[10]] // TrigReduce
```

$\left\{\frac{1}{1024}, \frac{5}{512}, \frac{45}{1024}, \frac{15}{128}, \frac{105}{512}, \frac{63}{256}, \frac{105}{512}, \frac{15}{128}, \frac{45}{1024}, \frac{5}{512}, \frac{1}{1024}\right\}$

$\text{Cos}[\pi\, k_1]^{10}$

$\frac{1}{512}\,(126 + 210\,\text{Cos}[2\,\pi\, k_1] + 120\,\text{Cos}[4\,\pi\, k_1] + 45\,\text{Cos}[6\,\pi\, k_1] + 10\,\text{Cos}[8\,\pi\, k_1] + \text{Cos}[10\,\pi\, k_1])$

1D-Plots von Transferfunktionen :

```
Show[(Print[TableForm@TrigReduce[#[[1]]]];
    Plot[Evaluate[#[[1]]], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}},
     PlotStyle → Thick, PlotLegends → Placed[(MatrixForm[#] & /@ #[[2]]), {{0, 0}, {-0.5, 0.0}}]]) &[
  Transpose[{transferfunktion[#], MatrixForm[{#}]} & /@ {box[1], box[3], box[5], box[7], box[9], box[11]}]], ImageSize → pagewidth]
```

1

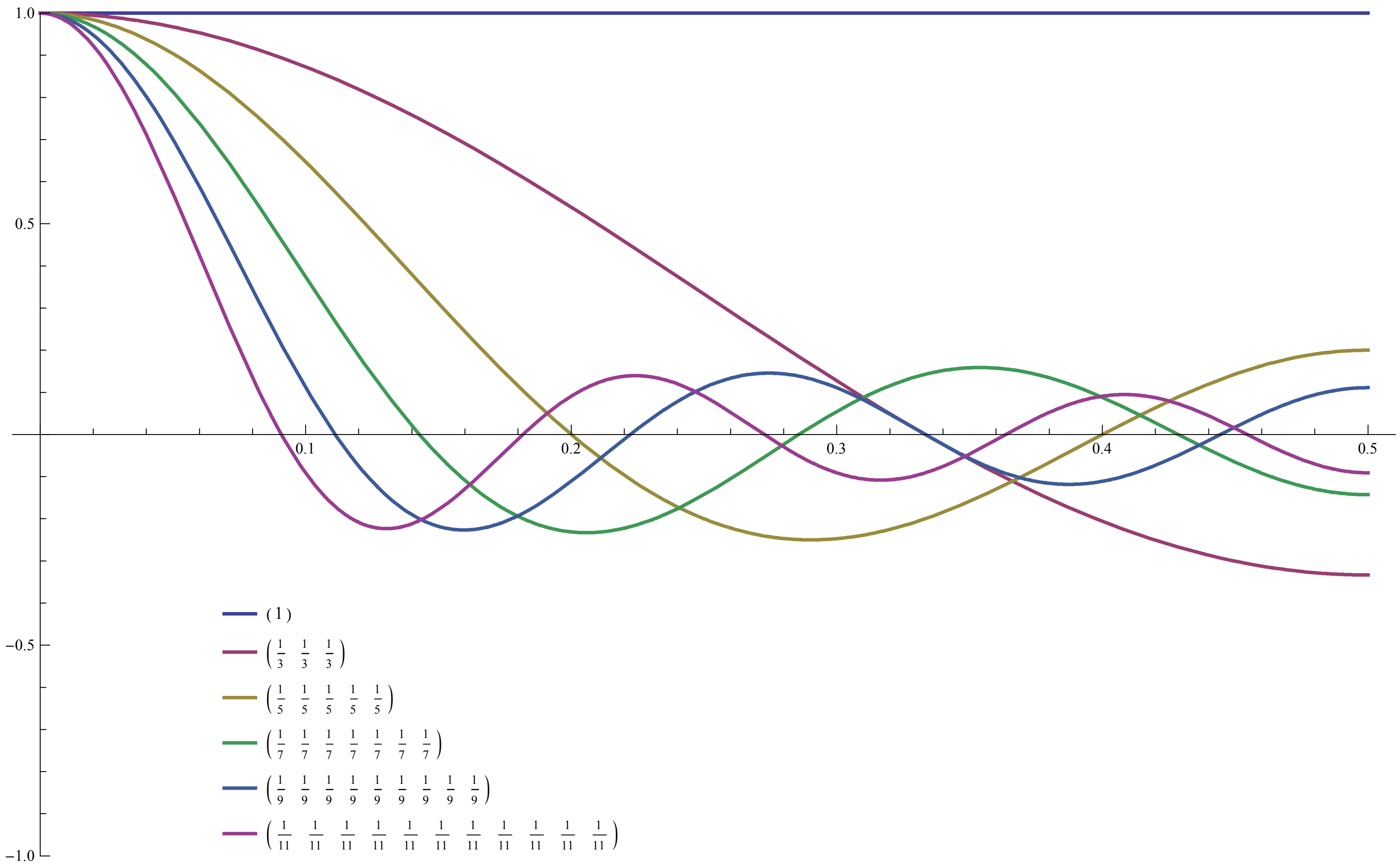
$\frac{1}{3}(1+2\,\mathrm{Cos}[2\,\pi\,k_1])$

$\frac{1}{5}(1+2\,\mathrm{Cos}[2\,\pi\,k_1]+2\,\mathrm{Cos}[4\,\pi\,k_1])$

$\frac{1}{7}(1+2\,\mathrm{Cos}[2\,\pi\,k_1]+2\,\mathrm{Cos}[4\,\pi\,k_1]+2\,\mathrm{Cos}[6\,\pi\,k_1])$

$\frac{1}{9}(1+2\,\mathrm{Cos}[2\,\pi\,k_1]+2\,\mathrm{Cos}[4\,\pi\,k_1]+2\,\mathrm{Cos}[6\,\pi\,k_1]+2\,\mathrm{Cos}[8\,\pi\,k_1])$

$\frac{1}{11}(1+2\,\mathrm{Cos}[2\,\pi\,k_1]+2\,\mathrm{Cos}[4\,\pi\,k_1]+2\,\mathrm{Cos}[6\,\pi\,k_1]+2\,\mathrm{Cos}[8\,\pi\,k_1]+2\,\mathrm{Cos}[10\,\pi\,k_1])$


1.0
0.5
−0.5
−1.0
0.1
0.2
0.3
0.4
0.5
(1)
$\left(\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right)$
$\left(\frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\right)$
$\left(\frac{1}{7}\ \frac{1}{7}\ \frac{1}{7}\ \frac{1}{7}\ \frac{1}{7}\ \frac{1}{7}\ \frac{1}{7}\right)$
$\left(\frac{1}{9}\ \frac{1}{9}\ \frac{1}{9}\ \frac{1}{9}\ \frac{1}{9}\ \frac{1}{9}\ \frac{1}{9}\ \frac{1}{9}\ \frac{1}{9}\right)$
$\left(\frac{1}{11}\ \frac{1}{11}\ \frac{1}{11}\ \frac{1}{11}\ \frac{1}{11}\ \frac{1}{11}\ \frac{1}{11}\ \frac{1}{11}\ \frac{1}{11}\ \frac{1}{11}\ \frac{1}{11}\right)$

```
Show[(Print[TableForm@TrigReduce[#[[1]]]];
    Plot[Evaluate[#[[1]]], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}},
     PlotStyle → Thick, PlotLegends → Placed[(MatrixForm[#] & /@ #[[2]]), {{0, 0}, {-0.5, 0.0}}]]) &[Transpose[
   {transferfunktion[#], MatrixForm[{#}]} & /@ {binom[0], binom[2], binom[4], binom[6], binom[8], binom[10]}]], ImageSize → pagewidth]
```

1

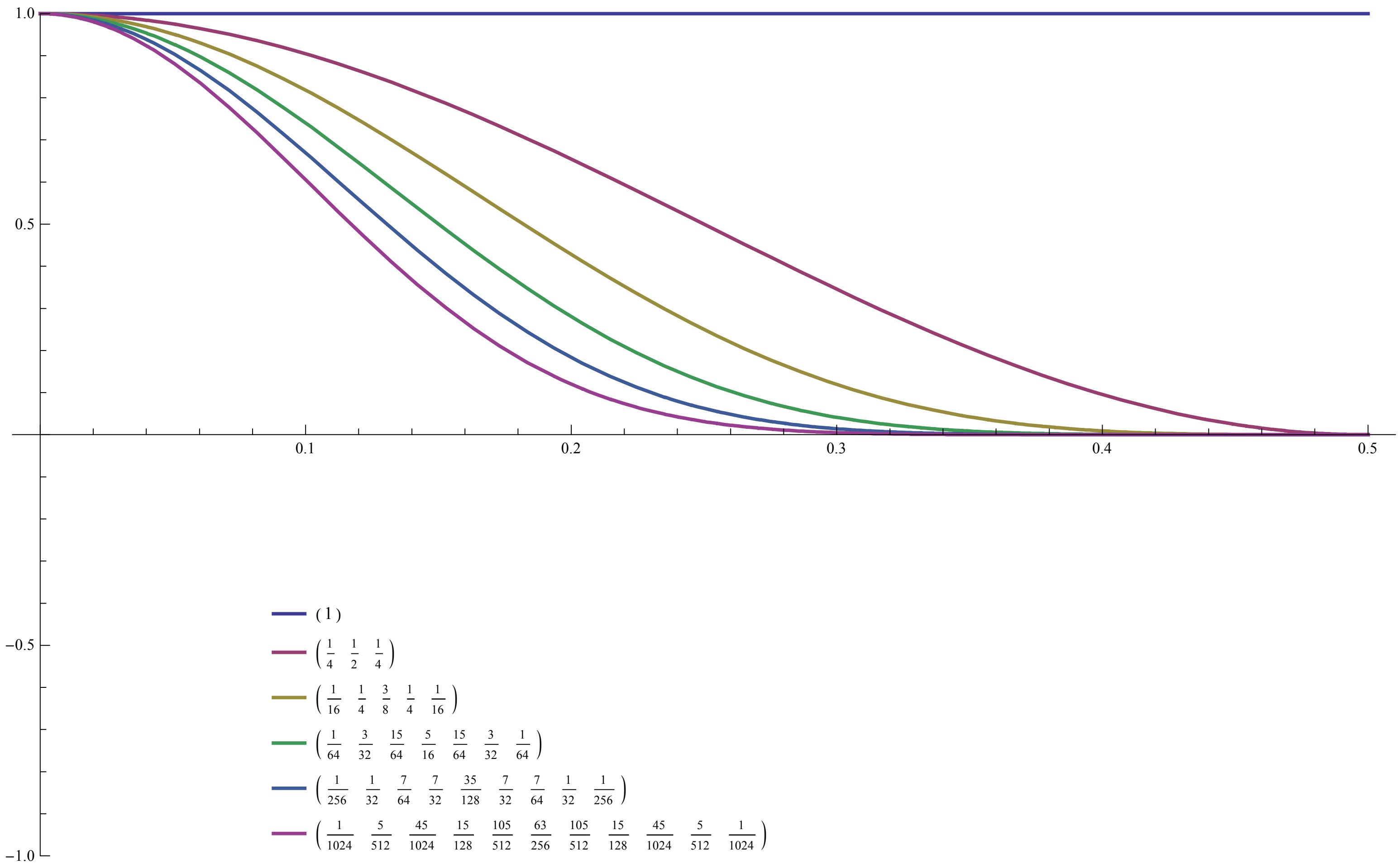
$\frac{1}{2}\left(1+\text{Cos}[2\pi k_1]\right)$

$\frac{1}{8}\left(3+4\,\text{Cos}[2\pi k_1]+\text{Cos}[4\pi k_1]\right)$

$\frac{1}{32}\left(10+15\,\text{Cos}[2\pi k_1]+6\,\text{Cos}[4\pi k_1]+\text{Cos}[6\pi k_1]\right)$

$\frac{1}{128}\left(35+56\,\text{Cos}[2\pi k_1]+28\,\text{Cos}[4\pi k_1]+8\,\text{Cos}[6\pi k_1]+\text{Cos}[8\pi k_1]\right)$

$\frac{1}{512}\left(126+210\,\text{Cos}[2\pi k_1]+120\,\text{Cos}[4\pi k_1]+45\,\text{Cos}[6\pi k_1]+10\,\text{Cos}[8\pi k_1]+\text{Cos}[10\pi k_1]\right)$



Beispiele 2D-DFT:

3x3 Boxfilter

```
Transpose[{box[3]}].{box[3]} // MatrixForm
transferfunktion2d[Transpose[{box[3]}].{box[3]}]
```

$$\begin{pmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{pmatrix}$$

$$\frac{1}{9} (1 + 2 \text{Cos}[2 \pi k_1]) (1 + 2 \text{Cos}[2 \pi k_2])$$

3x3 Binomialfilter

```
Transpose[{binom[2]}].{binom[2]} // MatrixForm
transferfunktion2d[Transpose[{binom[2]}].{binom[2]}]
```

$$\begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix}$$

$$\text{Cos}[\pi k_1]^2 \text{Cos}[\pi k_2]^2$$

5x5 Binomialfilter

```
Transpose[{binom[4]}].{binom[4]} // MatrixForm
transferfunktion2d[Transpose[{binom[4]}].{binom[4]}]
```

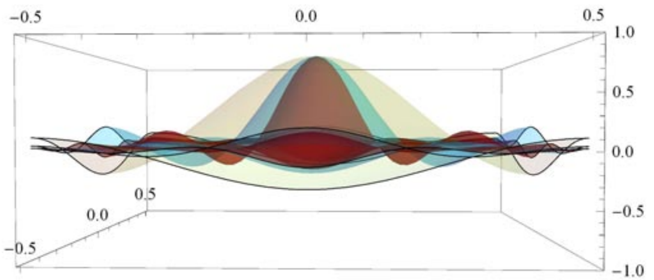
$$\begin{pmatrix} \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{3}{128} & \frac{3}{32} & \frac{9}{64} & \frac{3}{32} & \frac{3}{128} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \end{pmatrix}$$

$\mathrm{Cos}[\pi\, k_1]^4\, \mathrm{Cos}[\pi\, k_2]^4$

2D-Plots von Transferfunktionen :

Boxfilter

```
Plot3D[Evaluate[{transferfunktion2d[Transpose[{box[3]}].{box[3]}],
   transferfunktion2d[Transpose[{box[5]}].{box[5]}], transferfunktion2d[Transpose[{box[7]}].{box[7]}]}], {k1, -kmax, kmax},
 {k2, -kmax, kmax}, PlotStyle → {Directive[Yellow, Opacity[.125]], Directive[Cyan, Opacity[.25]], Directive[Red, Opacity[.5]]},
 PlotLegends → Placed["Expressions", {Left, Bottom}], Mesh → None, PlotRange → {Full, Full, {-1, 1}}, ViewPoint → Front]
```



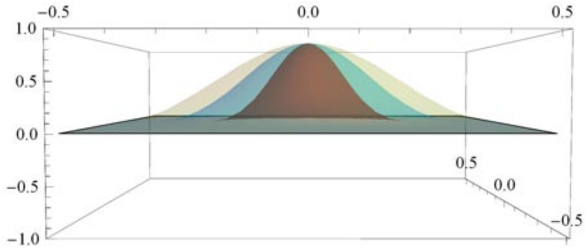
$\frac{1}{9}(1+2\cos(2\pi k_1))(1+2\cos(2\pi k_2))$

$\frac{1}{25}(1+2\cos(2\pi k_1)+2\cos(4\pi k_1))(1+2\cos(2\pi k_2)+2\cos(4\pi k_2))$

$\frac{1}{49}(1+2\cos(2\pi k_1)+2\cos(4\pi k_1)+2\cos(6\pi k_1))(1+2\cos(2\pi k_2)+2\cos(4\pi k_2)+2\cos(6\pi k_2))$

Binomialfilter

```
Plot3D[Evaluate[{transferfunktion2d[Transpose[{binom[2]}].{binom[2]}], transferfunktion2d[Transpose[{binom[4]}].{binom[4]}],
    transferfunktion2d[Transpose[{binom[12]}].{binom[12]}]}], {k1, -kmax, kmax}, {k2, -kmax, kmax},
 PlotStyle → {Directive[Yellow, Opacity[.125]], Directive[Cyan, Opacity[.25]], Directive[Red, Opacity[.5]]},
 PlotLegends → Placed["Expressions", {Left, Bottom}], Mesh → None, PlotRange → {Full, Full, {-1, 1}}, ViewPoint → Front]
```



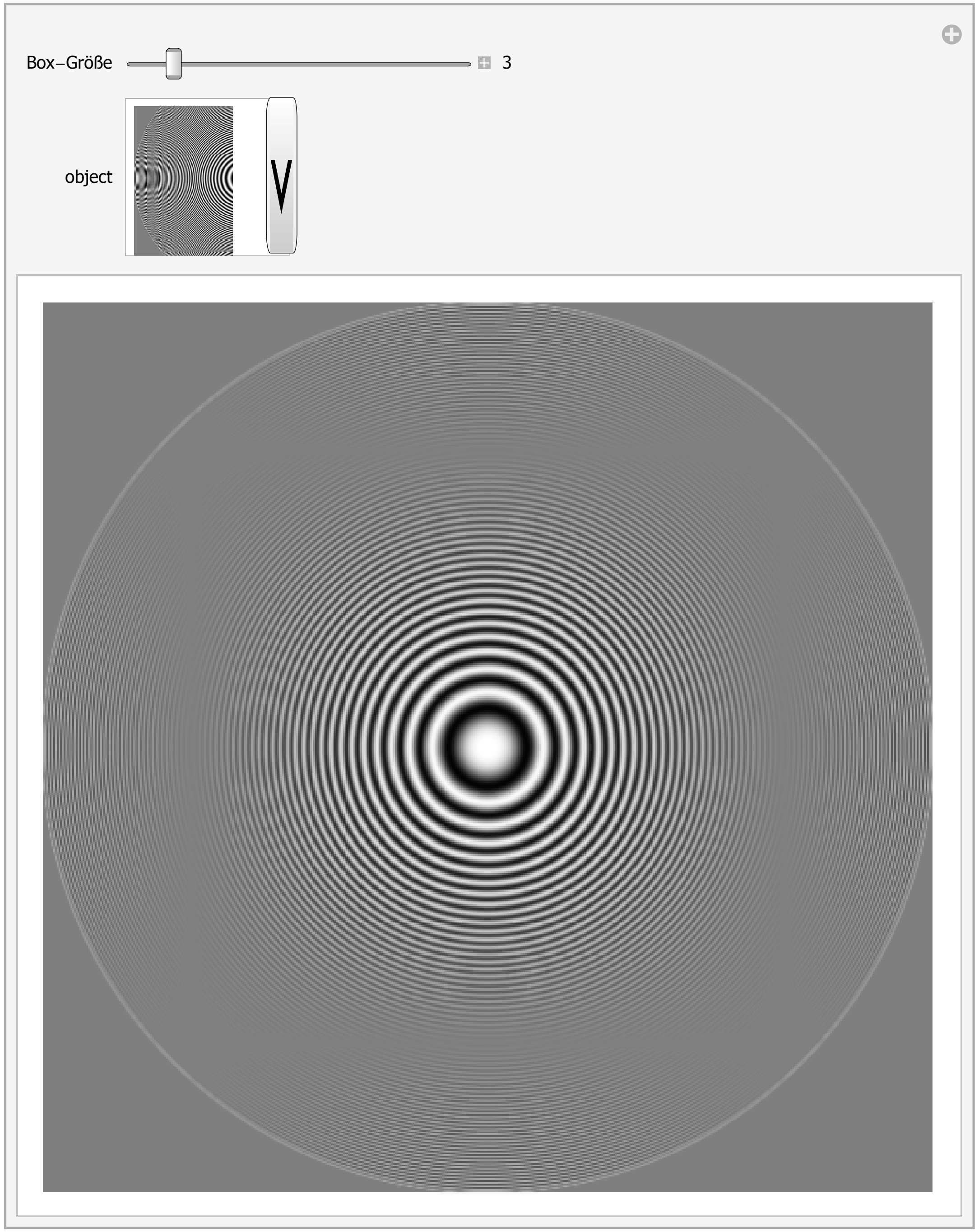
$\cos^2(\pi k_1)\cos^2(\pi k_2)$

$\cos^4(\pi k_1)\cos^4(\pi k_2)$

$\cos^{12}(\pi k_1)\cos^{12}(\pi k_2)$

Interaktive Demo des Boxfilters

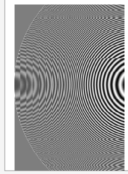
```
Manipulate[Show[Image[ListConvolve[Transpose[{box[größe]}].{box[größe]}, ImageData@bild, (größe + 1) / 2]],
  ImageSize → ImageDimensions@bild], {{größe, 3, "Box-Größe"}, 1, 21, 2, Appearance → "Labeled"},
 {{bild, wellenbild, "object"}, {wellenbild, First@ColorSeparate[Ton3CD21dreikanalausgleichF2], mandrill, phantomhead, turtle, lenay,
   pollen, randombild, rauschbild, ocelot}, ImageSize → Tiny, ControlType → PopupMenu}, ContinuousAction → False, SaveDefinitions → True]
```

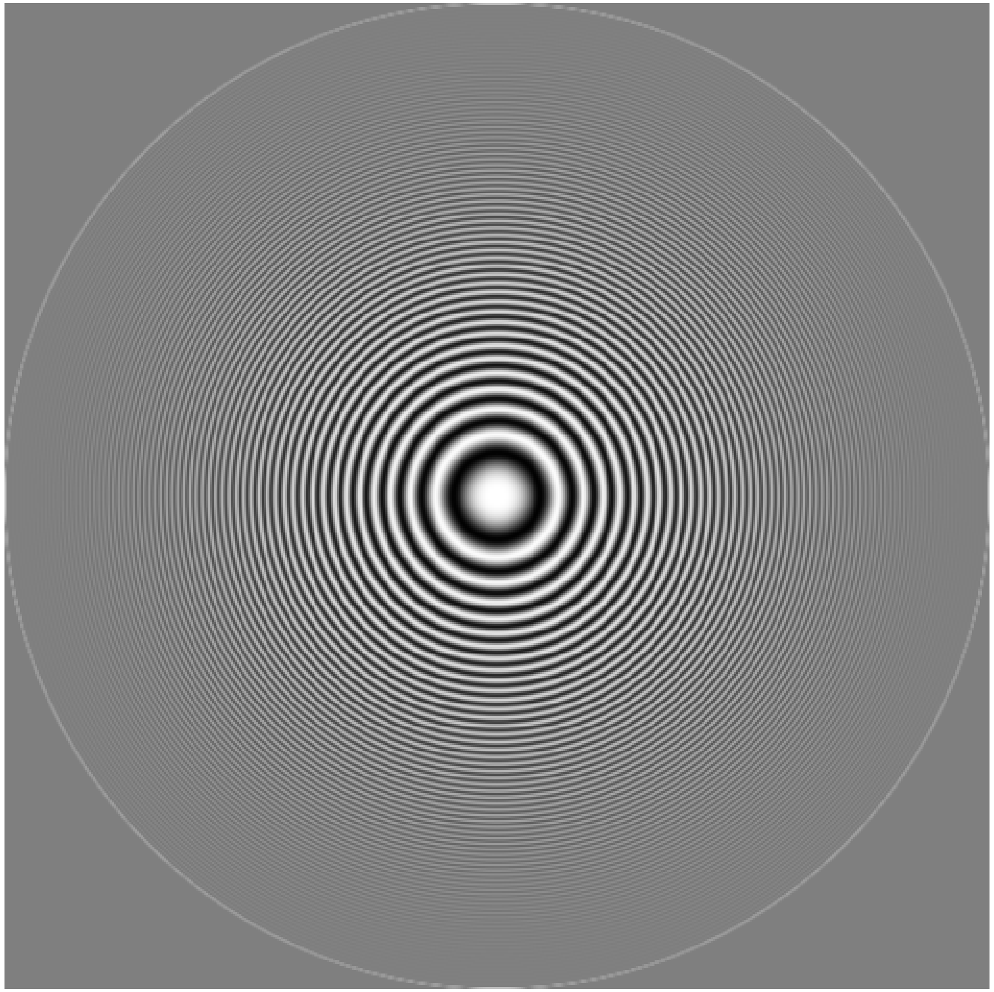

Box–Größe
3
object

Interaktive Demo des Tiefpaßfilter (Binomialfilters)

```
Manipulate[Show[Image[ListConvolve[Transpose[{binom[ordnung]}].{binom[ordnung]}, ImageData@bild, ordnung / 2 + 1]],
  ImageSize → ImageDimensions@bild], {{ordnung, 2, "Binom-Ordnung"}, 0, 30, 2, Appearance → "Labeled"},
 {{bild, wellenbild, "object"}, {wellenbild, First@ColorSeparate[Ton3CD21dreikanalausgleichF2], mandrill, phantomhead, turtle, lenay,
   pollen, randombild, rauschbild, ocelot}, ImageSize → Tiny, ControlType → PopupMenu}, ContinuousAction → False, SaveDefinitions → True]
```

Binom–Ordnung 2

object 



13. Weitere Einsatzfelder linearer diskrete verschiebungsinvariante Filter

Bandpaß als Linearkombination von Filtern, zum Beispiel als Differenz von Binomialfiltern 2. und 4. Ordnung

```
Clear[plottransferfunktion];
plottransferfunktion[formel_] := Module[{},
   GraphicsGrid[{{

      Plot[Evaluate[ComplexExpand[Re[formel]]], {k1, 0, kmax},
       AxesLabel -> {k1, None}, PlotRange → {Full, {-1, 1}}, PlotLabel → "Realteil", PlotPoints → r],

      Plot[Evaluate[ComplexExpand[Im[formel]]], {k1, 0, kmax},
       AxesLabel -> {k1, None}, PlotRange → {Full, {-1, 1}}, PlotLabel → "Imaginärteil", PlotPoints → r],

      Plot[Evaluate[ArcTan[ComplexExpand[Im[formel] / (ComplexExpand[Re[formel]] + $MachineEpsilon)]]],
       {k1, 0, kmax}, AxesLabel -> {k1, None}, PlotRange → {Full, {-π / 2, π / 2}}, PlotLabel → "Phasenwinkel",
       PlotPoints → r, Ticks → {Automatic, (Range[#] - (# + 1) / 2) / ((# - 1) / 2) / 2 * π &[5]}]

     }}]
  ];
```

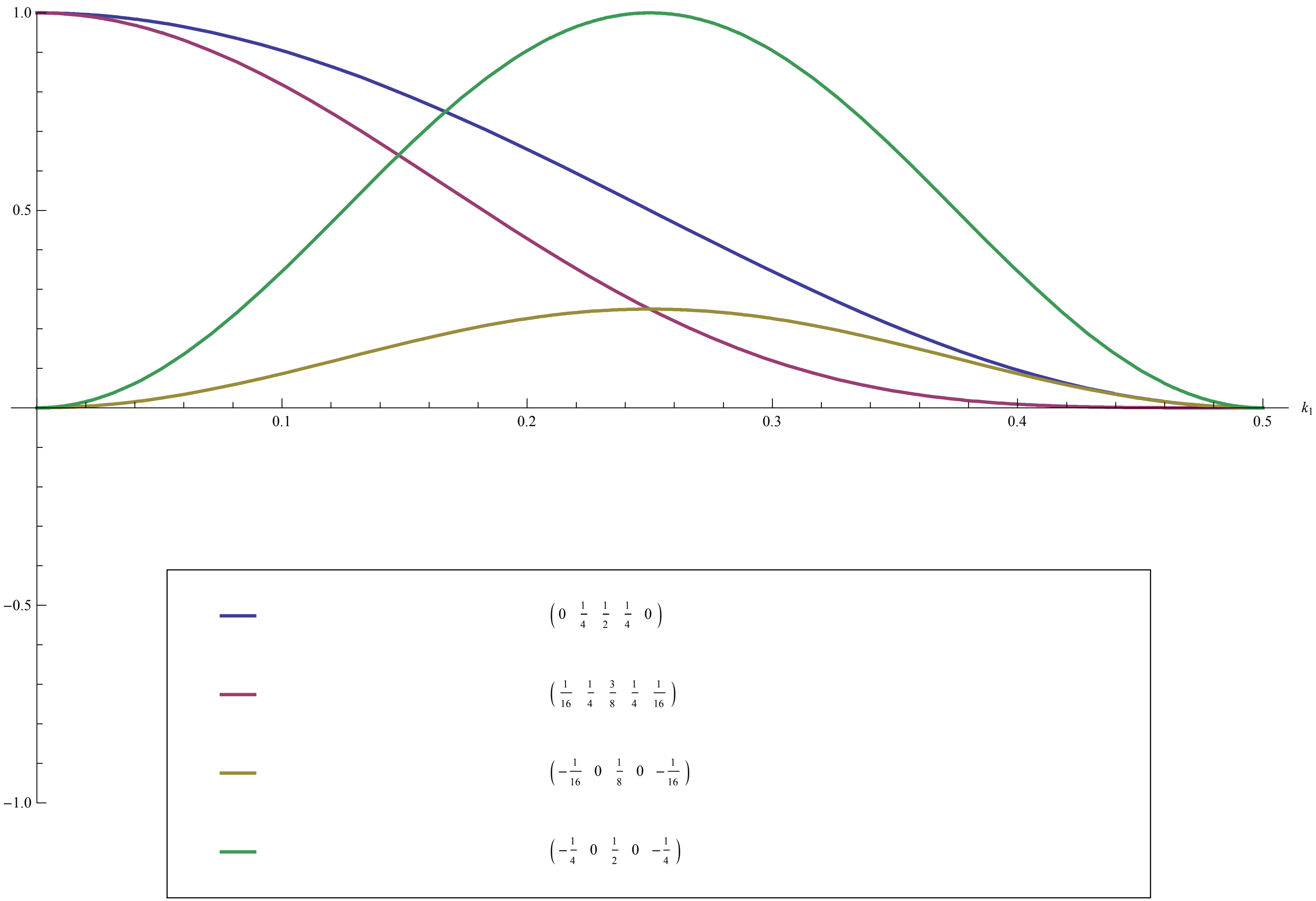
```
Clear[plottransferfunktion2d];
plottransferfunktion2d[formel_] := Module[{},
  GraphicsGrid[{{

     Plot3D[Evaluate[ComplexExpand[Re[formel]]], {k1, -kmax, kmax}, {k2, -kmax, kmax},
      AxesLabel -> {k1, k2, None}, PlotRange → {Full, Full, {-1, 1}}, PlotLabel → "Realteil", Mesh → 19, RotationAction → "Clip"],

     Plot3D[Evaluate[ComplexExpand[Im[formel]]], {k1, -kmax, kmax}, {k2, -kmax, kmax}, AxesLabel -> {k1, k2, None},
      PlotRange → {Full, Full, {-1, 1}}, PlotLabel → "Imaginärteil", Mesh → 19, RotationAction → "Clip"],

     Plot3D[Evaluate[ArcTan[ComplexExpand[Im[formel] / (ComplexExpand[Re[formel]] + $MachineEpsilon)]]], {k1, -kmax, kmax},
      {k2, -kmax, kmax}, AxesLabel -> {k1, k2, None}, PlotRange → {Full, Full, {-π / 2, π / 2}}, PlotLabel → "Phasenwinkel",
      Ticks → {Automatic, Automatic, (Range[#] - (# + 1) / 2) / ((# - 1) / 2) / 2 * π &[3]}, Mesh → 19, RotationAction → "Clip"]

    }}]
  ];

Show[(Print[TableForm@TrigReduce[#[[1]]]];
   Plot[Evaluate[#[[1]]], {k1, 0, kmax}, AxesLabel -> {k1, None},
    PlotRange → {Full, {-1, 1}}, PlotStyle → Thick, PlotLegend → (MatrixForm[#] & /@ #[[2]]), LegendShadow → None,
    LegendPosition → {-0.75, -0.75}, LegendSize → {1.5, .5}, LegendOrientation → Vertical]) &[
  Transpose[{transferfunktion[#], MatrixForm[{#}]} & /@ {ListConvolve[binom[2], identität[5], 2], binom[4],
    ListConvolve[binom[2], identität[5], 2] - binom[4], (ListConvolve[binom[2], identität[5], 2] - binom[4]) /
     First[Maximize[{(transferfunktion[binom[2]] - transferfunktion[binom[4]]), 0 ≤ k1 ≤ 1 / 2}, k1]]}]], ImageSize → 800]
```

$\frac{1}{2}(1 + \text{Cos}[2\pi k_1])$

$\frac{1}{8}(3 + 4\,\text{Cos}[2\pi k_1] + \text{Cos}[4\pi k_1])$

$\frac{1}{8}(1 - \text{Cos}[4\pi k_1])$

$\frac{1}{2}(1 - \text{Cos}[4\pi k_1])$



transferfunktion[binom[2]] - transferfunktion[binom[4]]

$\mathrm{Cos}[\pi\, k_1]^2 - \mathrm{Cos}[\pi\, k_1]^4$

Maximize[{(transferfunktion[binom[2]] - transferfunktion[binom[4]]), 0 ≤ k_1 ≤ 1 / 2}, k_1]

$\left\{\frac{1}{4}, \left\{k_1 \to \frac{1}{4}\right\}\right\}$

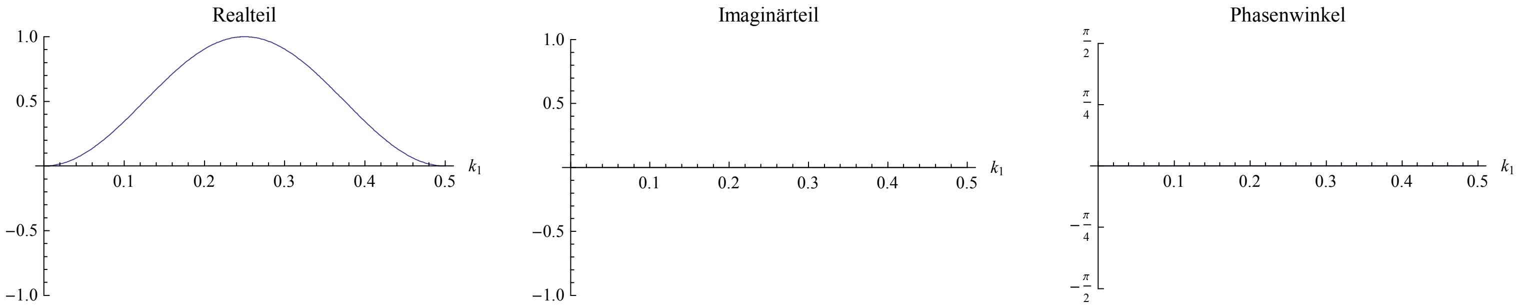
```
(transferfunktion[binom[2]] - transferfunktion[binom[4]]) /
 First[Maximize[{(transferfunktion[binom[2]] - transferfunktion[binom[4]]), 0 ≤ k1 ≤ 1 / 2}, k1]]
```

$4\left(\mathrm{Cos}[\pi k_1]^2 - \mathrm{Cos}[\pi k_1]^4\right)$

Viele Filter weisen keinerlei Phasenversatz auf, da gerade (wie zum Beispiel der eben betrachtete Bandpaß):

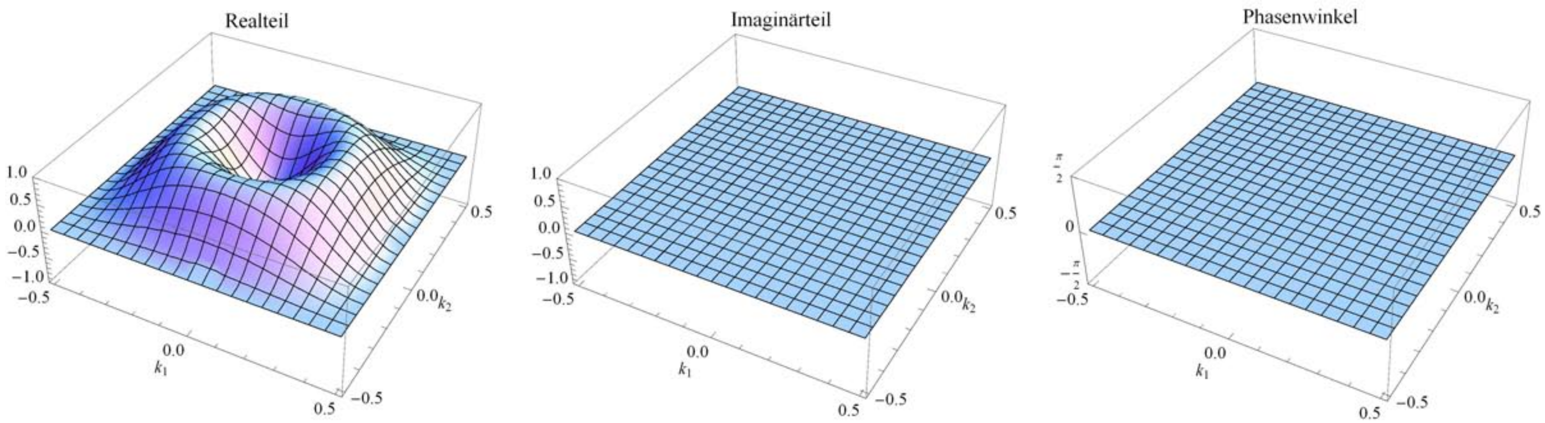
```
4 (transferfunktion[binom[2]] - transferfunktion[binom[4]])
Show[plottransferfunktion[4 (transferfunktion[binom[2]] - transferfunktion[binom[4]])], ImageSize → pagewidth]
```

$4\left(\mathrm{Cos}[\pi k_1]^2 - \mathrm{Cos}[\pi k_1]^4\right)$



```
4 (transferfunktion2d[Transpose[{binom[2]}].{binom[2]}] - transferfunktion2d[Transpose[{binom[4]}].{binom[4]}])
Show[plottransferfunktion2d[
  4 (transferfunktion2d[Transpose[{binom[2]}].{binom[2]}] - transferfunktion2d[Transpose[{binom[4]}].{binom[4]}])], ImageSize → pagewidth]
```

$4\left(\text{Cos}[\pi k_1]^2 \text{Cos}[\pi k_2]^2 - \text{Cos}[\pi k_1]^4 \text{Cos}[\pi k_2]^4\right)$



Das Superpositionsprinzip gilt (man muß nicht die Summe der Transferfunktionen betrachten, man kann auch die Transferfunktion der Summe nehmen):

```
ListConvolve[Transpose[{binom[2]}].{binom[2]}, Transpose[{identität[5]}].{identität[5]}, {{2, 2}}]
```

$\left\{\{0, 0, 0, 0, 0\}, \left\{0, \frac{1}{16}, \frac{1}{8}, \frac{1}{16}, 0\right\}, \left\{0, \frac{1}{8}, \frac{1}{4}, \frac{1}{8}, 0\right\}, \left\{0, \frac{1}{16}, \frac{1}{8}, \frac{1}{16}, 0\right\}, \{0, 0, 0, 0, 0\}\right\}$

```
bandpaßkern = 4 *
   (ListConvolve[Transpose[{binom[2]}].{binom[2]}, Transpose[{identität[5]}].{identität[5]}, {{2, 2}}] - Transpose[{binom[4]}].{binom[4]});
```

```
bandpaßkern // MatrixForm
```

$$\begin{pmatrix} -\frac{1}{64} & -\frac{1}{16} & -\frac{3}{32} & -\frac{1}{16} & -\frac{1}{64} \\ -\frac{1}{16} & 0 & \frac{1}{8} & 0 & -\frac{1}{16} \\ -\frac{3}{32} & \frac{1}{8} & \frac{7}{16} & \frac{1}{8} & -\frac{3}{32} \\ -\frac{1}{16} & 0 & \frac{1}{8} & 0 & -\frac{1}{16} \\ -\frac{1}{64} & -\frac{1}{16} & -\frac{3}{32} & -\frac{1}{16} & -\frac{1}{64} \end{pmatrix}$$

```
MatrixRank[bandpaßkern]
```

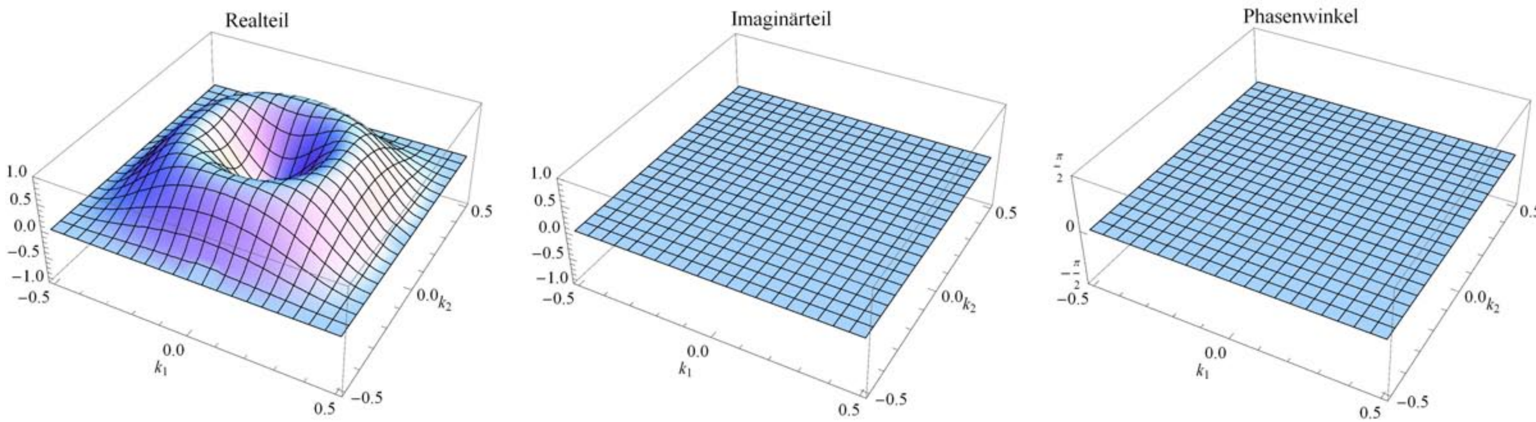
2

```
MatrixRank[Transpose[{binom[4]}].{binom[4]}]
```

1

```
transferfunktion2d[bandpaßkern]
Show[plottransferfunktion2d[transferfunktion2d[bandpaßkern]], ImageSize → pagewidth]
```

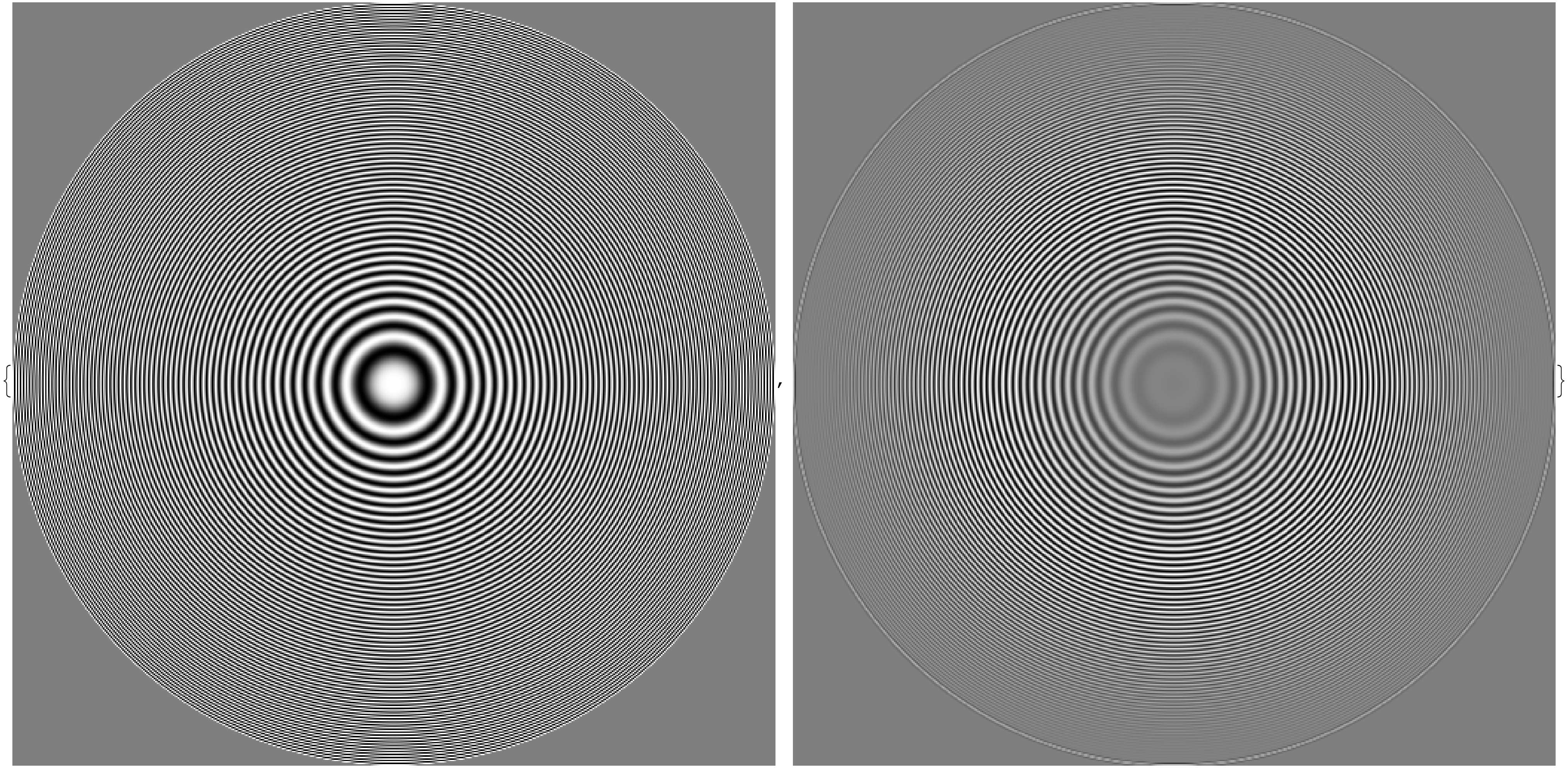
$$-\frac{1}{2} \mathrm{Cos}[\pi k_1]^2 \mathrm{Cos}[\pi k_2]^2 (-6 + 2 \mathrm{Cos}[2 \pi k_1] + \mathrm{Cos}[2 \pi (k_1 - k_2)] + 2 \mathrm{Cos}[2 \pi k_2] + \mathrm{Cos}[2 \pi (k_1 + k_2)])$$



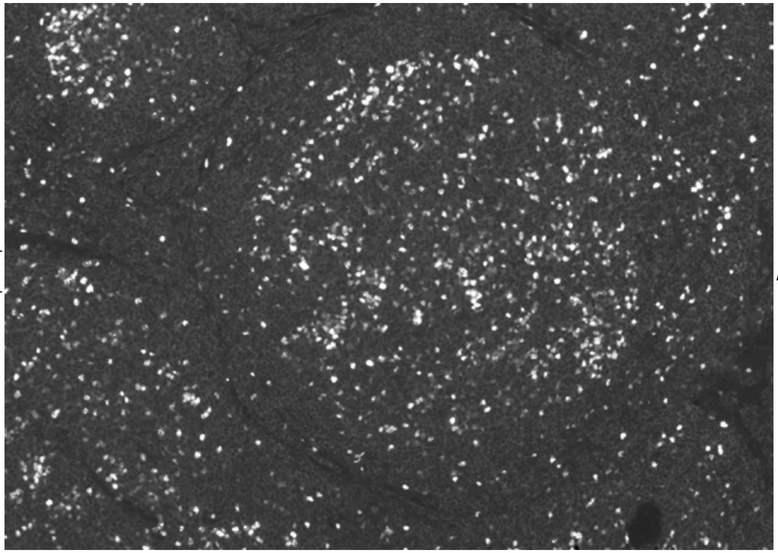
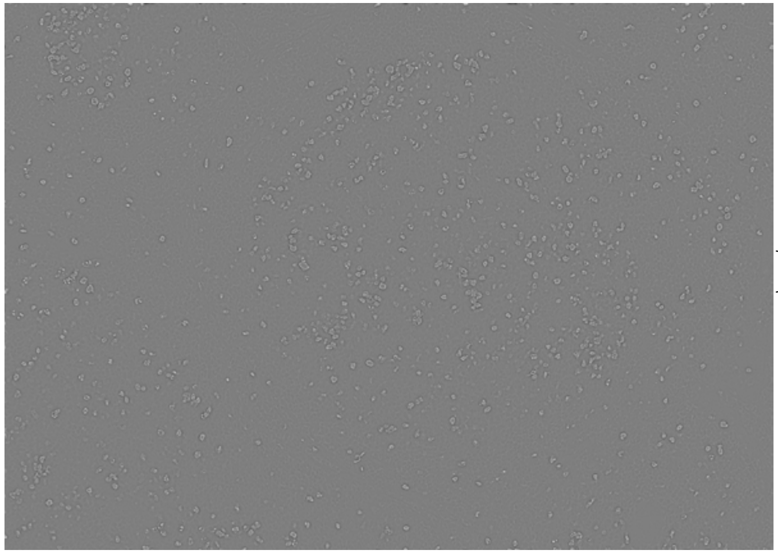
```
FullSimplify[TrigToExp[transferfunktion2d[bandpaßkern]]] ==
 FullSimplify[TrigToExp[4 (transferfunktion2d[Transpose[{binom[2]}].{binom[2]}] - transferfunktion2d[Transpose[{binom[4]}].{binom[4]}])]]
```

True

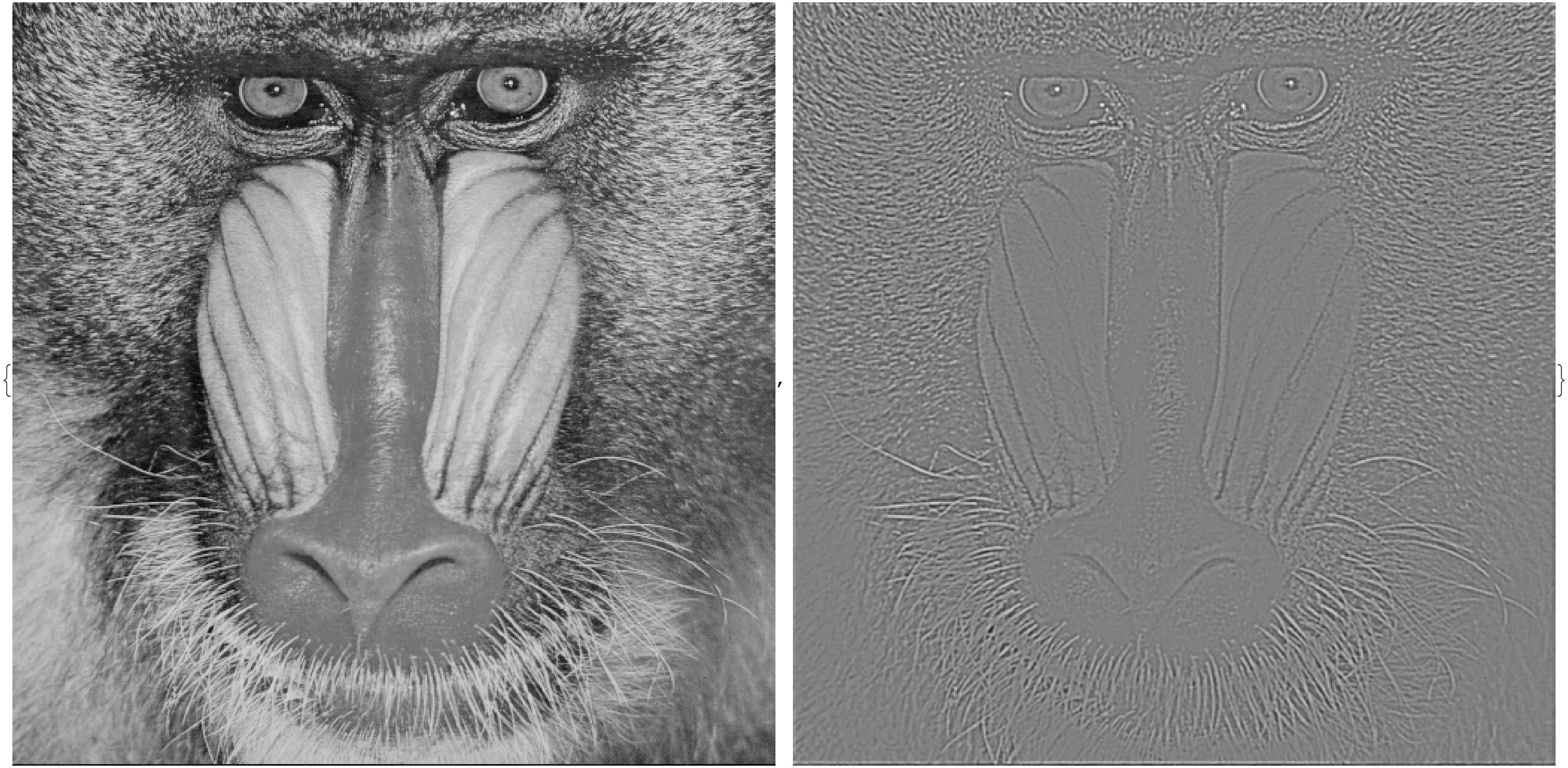
```
{Show[Image[#2]], Show[Image[0.5 + ListConvolve[#1, #2, {(Dimensions[#1] + 1) / 2}]]]} &[bandpaßkern, ImageData[wellenbild]]
```



```
Show[#, ImageSize → pagewidth / 2] & /@ ({Image@#2, Image[0.5 + ListConvolve[#1, #2, {(Dimensions[#1] + 1) / 2}]]} &[
   bandpaßkern, ImageData@First@ColorSeparate@Lym3CD21dreikanalausgleichF2])
```

{, }

```
{Show[Image[#2]], Show[Image[0.5 + ListConvolve[#1, #2, {(Dimensions[#1] + 1) / 2}]]]} &[
 bandpaßkern, ImageData[ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Mandrill"}]]]]
```



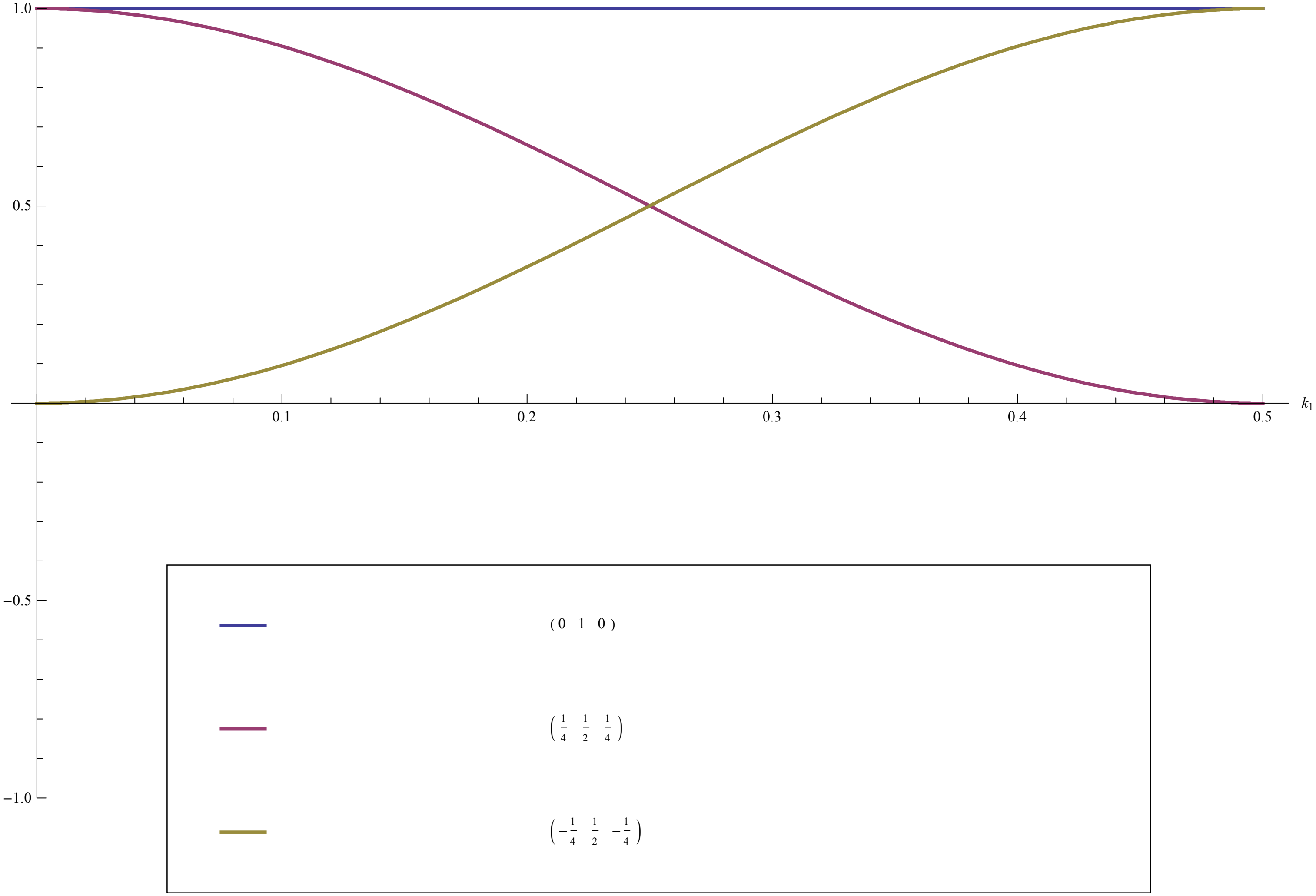
Hochpaß als Linearkombination von Filtern, zum Beispiel als Differenz von Identitätsoperator und Binomialfilter 4. Ordnung

```
Show[(Print[TableForm@TrigReduce[#[[1]]]]; Plot[Evaluate[#[[1]]], {k1, 0, kmax},
      AxesLabel -> {k1, None}, PlotRange → {Full, {-1, 1}}, PlotStyle → Thick, PlotLegend → (MatrixForm[#] & /@ #[[2]]),
      LegendShadow → None, LegendPosition → {-0.75, -0.75}, LegendSize → {1.5, .5}, LegendOrientation → Vertical]) &[
  Transpose[{transferfunktion[#], MatrixForm[{#}]} & /@ {identität[3], binom[2], identität[3] - binom[2]}]], ImageSize → 800]
```

1

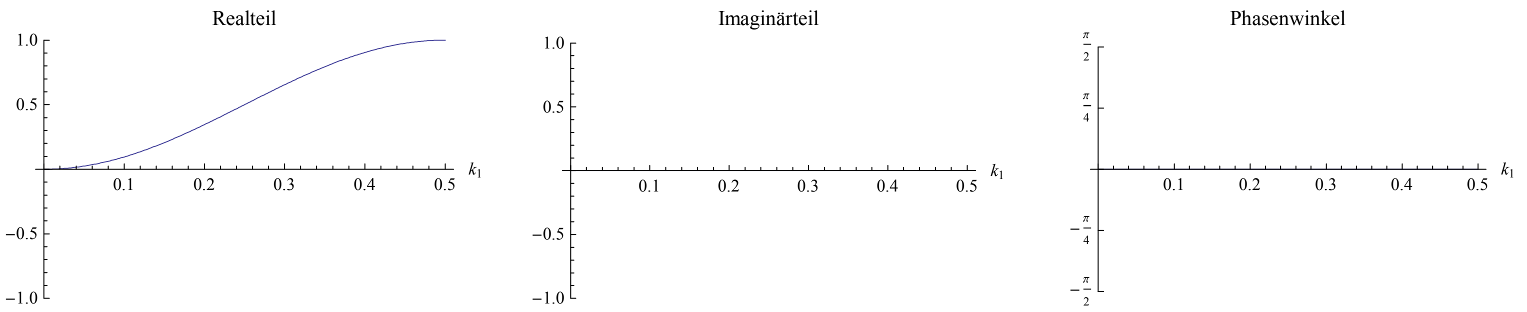
$\frac{1}{2}(1 + \text{Cos}[2\pi k_1])$

$\frac{1}{2}(1 - \text{Cos}[2\pi k_1])$


1.0
0.5
−0.5
−1.0
0.1
0.2
0.3
0.4
0.5
k_1
$(0\ \ 1\ \ 0)$
$\left(\frac{1}{4}\ \ \frac{1}{2}\ \ \frac{1}{4}\right)$
$\left(-\frac{1}{4}\ \ \frac{1}{2}\ \ -\frac{1}{4}\right)$

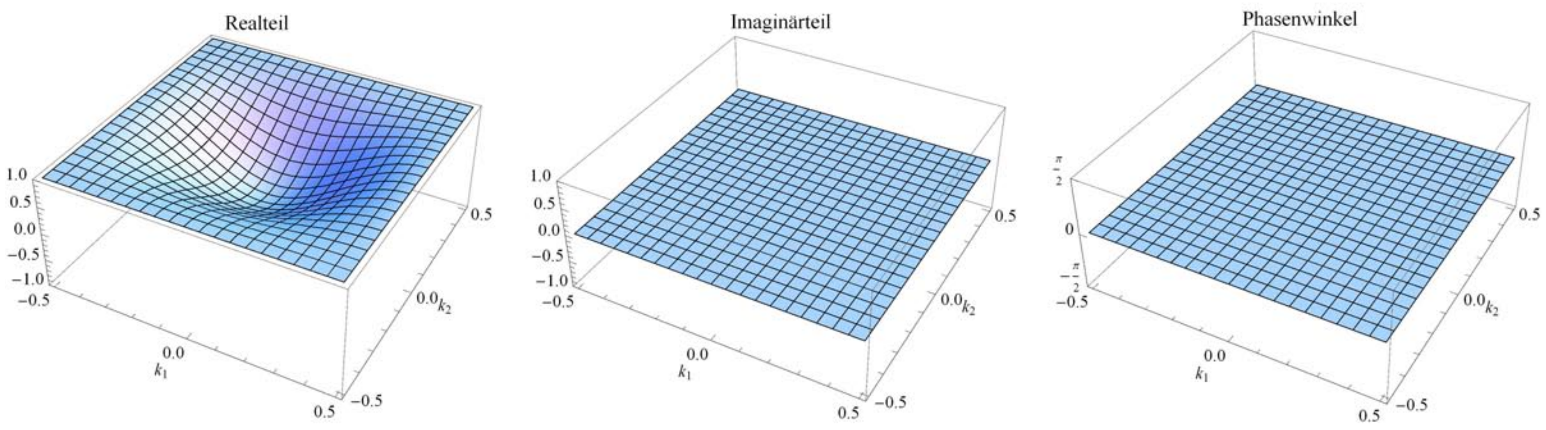
```
transferfunktion[identität[3]] - transferfunktion[binom[2]]
Show[plottransferfunktion[transferfunktion[identität[3]] - transferfunktion[binom[2]]], ImageSize → pagewidth]
```

$1 - \text{Cos}[\pi\, k_1]^2$




```
transferfunktion2d[Transpose[{identität[3]}].{identität[3]}] - transferfunktion2d[Transpose[{binom[2]}].{binom[2]}]
Show[plottransferfunktion2d[transferfunktion2d[Transpose[{identität[3]}].{identität[3]}] -
   transferfunktion2d[Transpose[{binom[2]}].{binom[2]}]], ImageSize → pagewidth]
```

$1 - \text{Cos}[\pi\, k_1]^2\, \text{Cos}[\pi\, k_2]^2$



Das Superpositionsprinzip gilt weiterhin:

```
hochpaßkern = Transpose[{identität[3]}].{identität[3]} - Transpose[{binom[2]}].{binom[2]};
```

```
hochpaßkern // MatrixForm
```

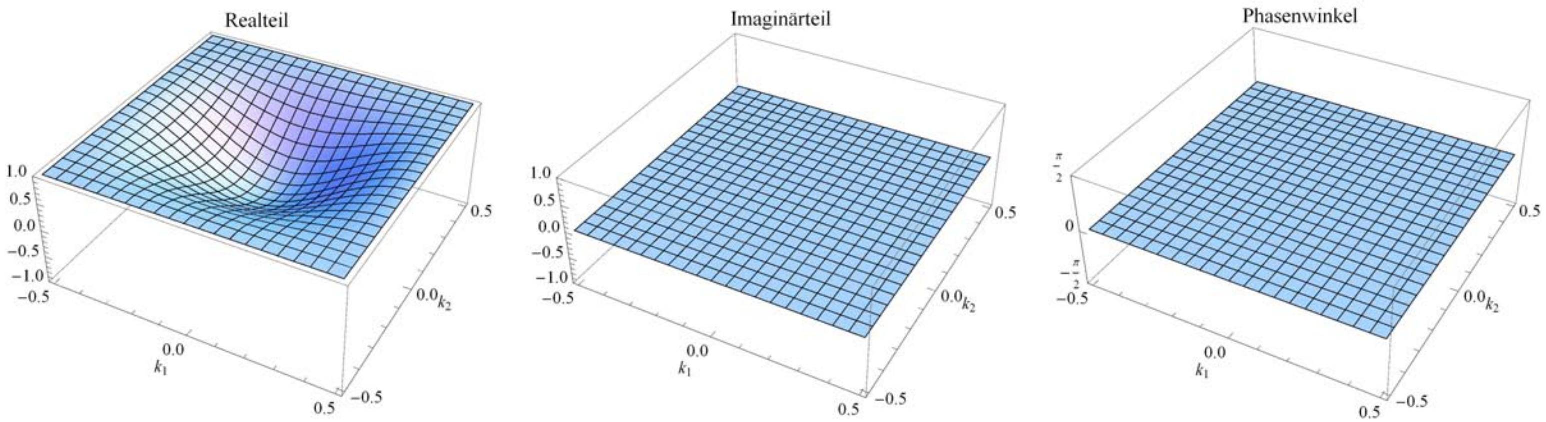
$$\begin{pmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{pmatrix}$$

```
MatrixRank[hochpaßkern]
```

2

```
transferfunktion2d[hochpaßkern]
Show[plottransferfunktion2d[transferfunktion2d[hochpaßkern]], ImageSize → pagewidth]
```

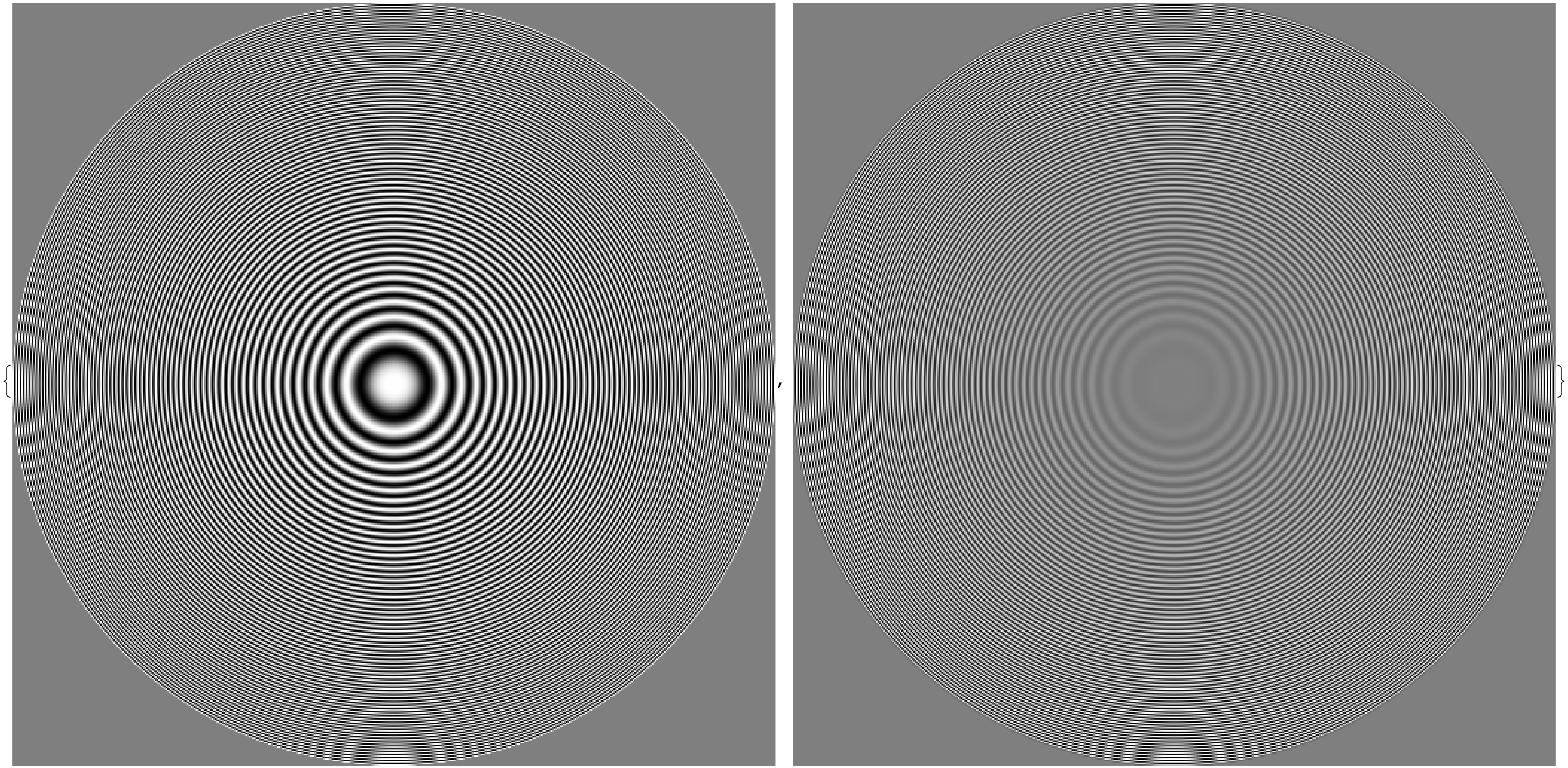
$$\frac{1}{8}\left(6 - 2\,\text{Cos}[2\,\pi\,k_1] - \text{Cos}[2\,\pi\,(k_1 - k_2)] - 2\,\text{Cos}[2\,\pi\,k_2] - \text{Cos}[2\,\pi\,(k_1 + k_2)]\right)$$



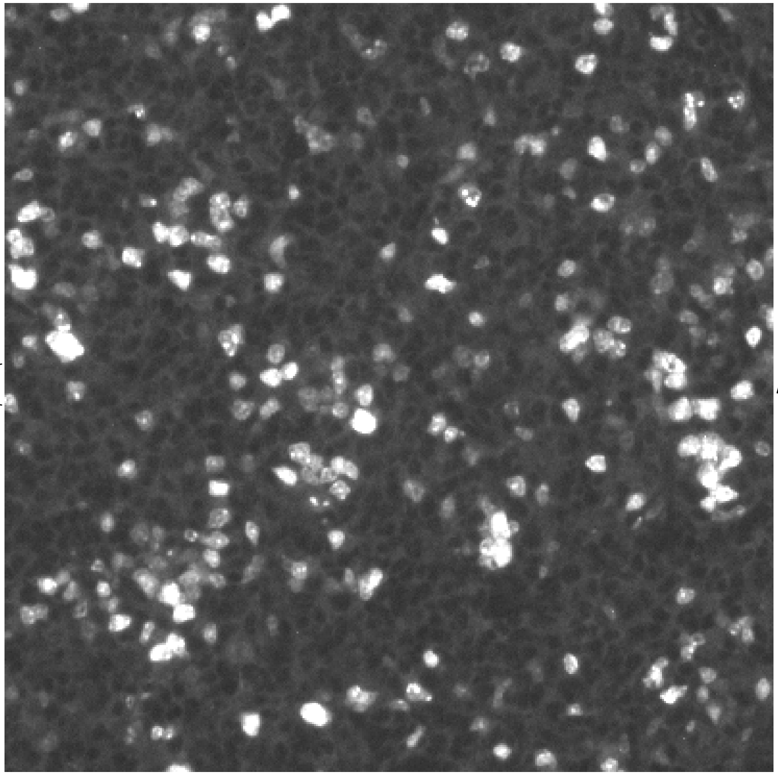
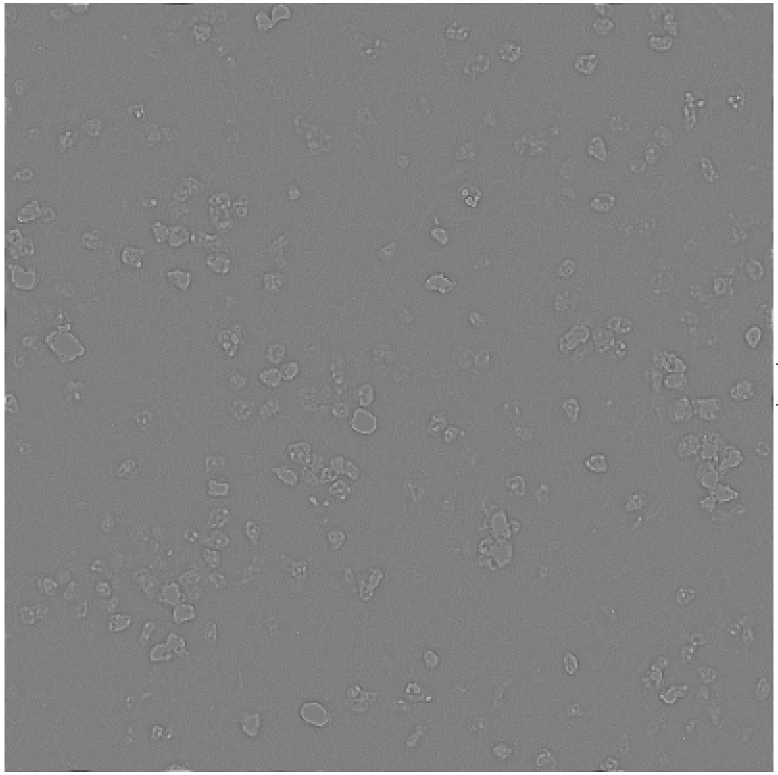

```
FullSimplify[
  TrigReduce[transferfunktion2d[Transpose[{identität[3]}].{identität[3]}] - transferfunktion2d[Transpose[{binom[2]}].{binom[2]}]]] ==
 FullSimplify[TrigReduce[transferfunktion2d[hochpaßkern]]]
```

True

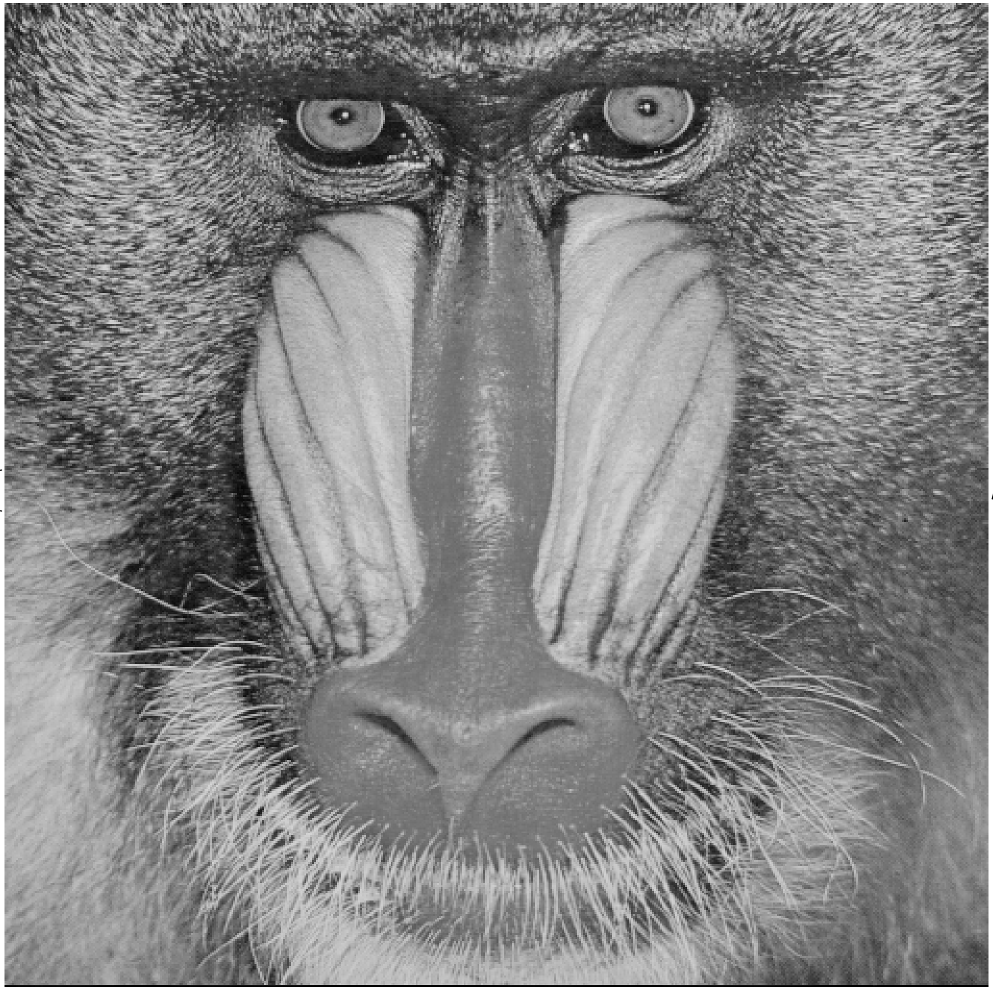
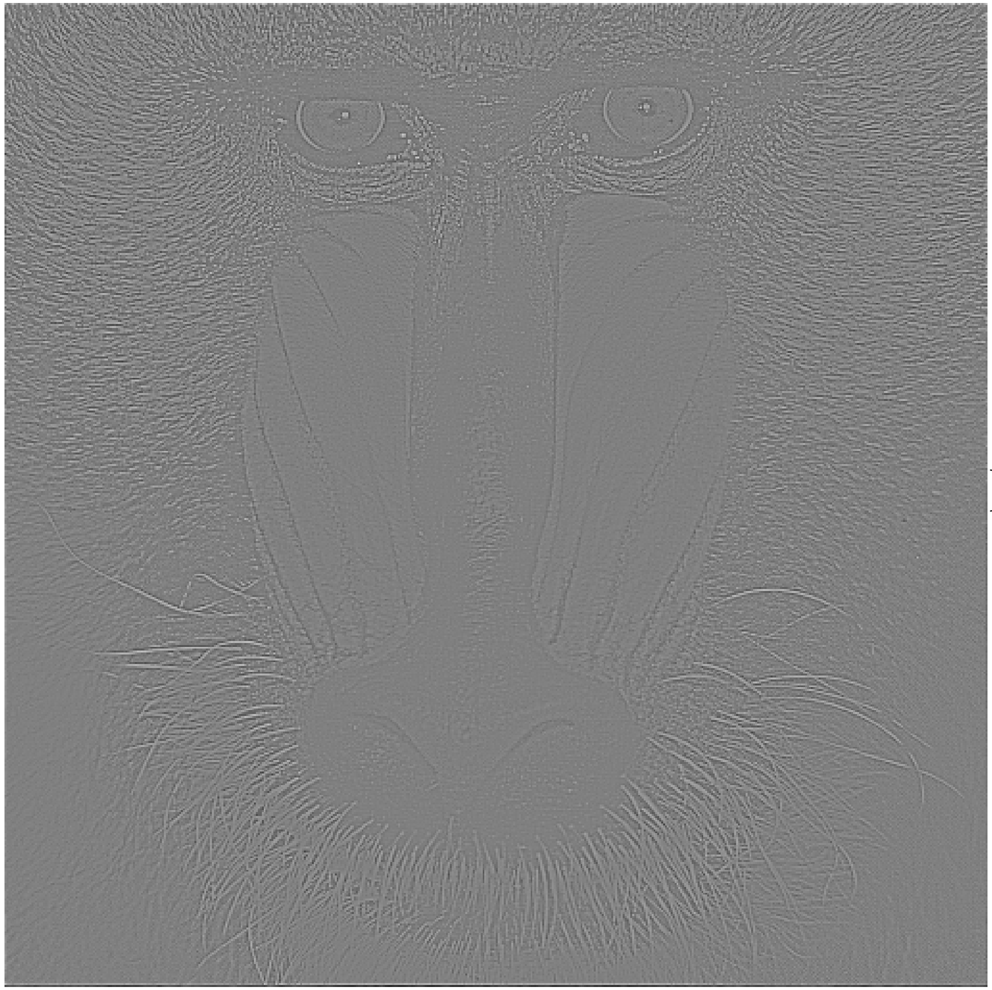
```
{Show[Image[#2]], Show[Image[0.5 + ListConvolve[#1, #2, {(Dimensions[#1] + 1) / 2}]]]} &[hochpaßkern, ImageData[wellenbild]]
```



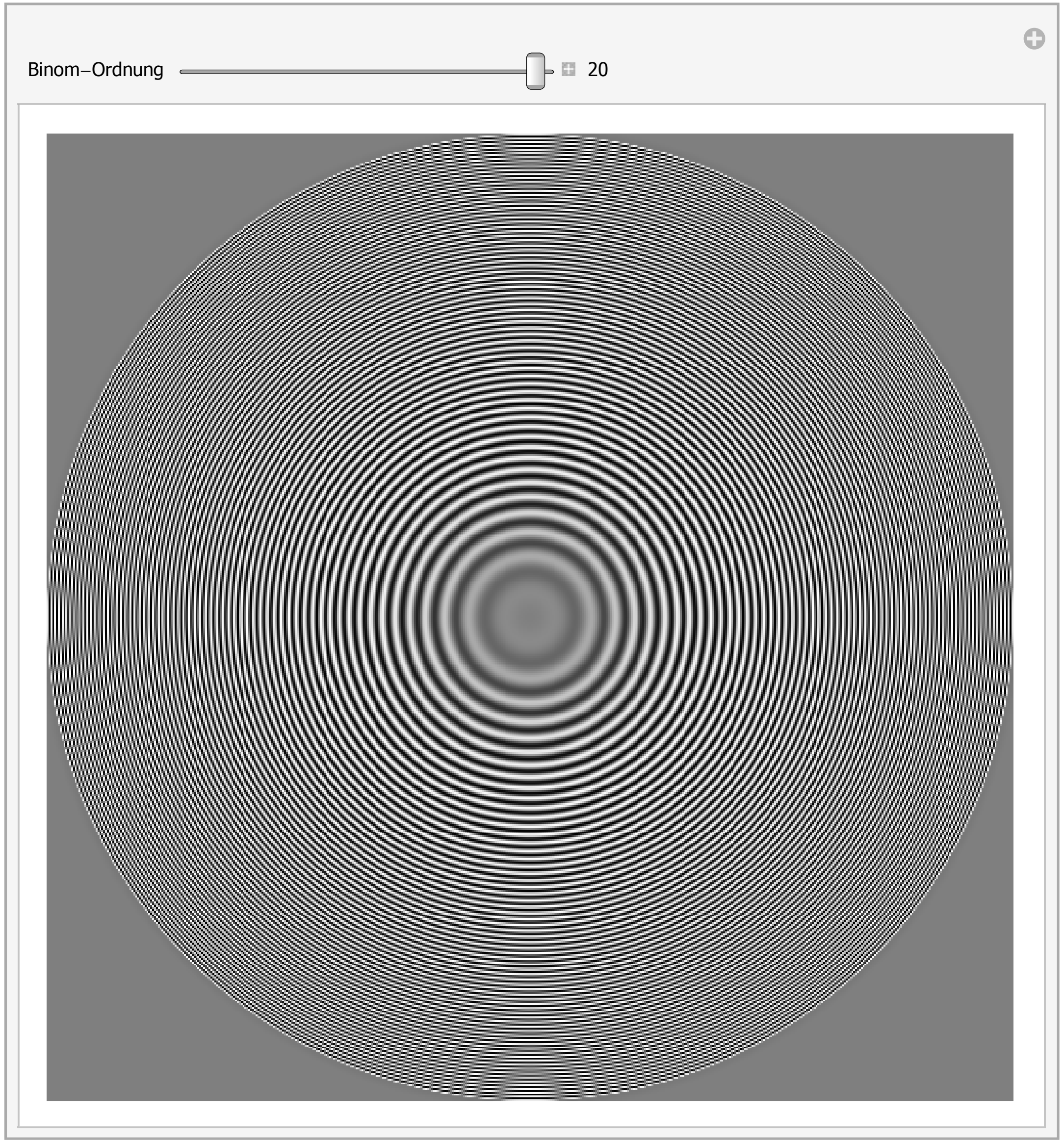
```
Show[#, ImageSize → pagewidth / 2] & /@ ({Image@#2, Image[0.5 + ListConvolve[#1, #2, {(Dimensions[#1] + 1) / 2}]]} &[
   hochpaßkern, ImageData@ImageCrop[First@ColorSeparate@Lym3CD21dreikanalausgleichF2, {pagewidth / 2, pagewidth / 2}]])
```

{, }

```
{Show[Image[#2]], Show[Image[0.5 + ListConvolve[#1, #2, {(Dimensions[#1] + 1) / 2}]]]} &[
 hochpaßkern, ImageData[ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Mandrill"}]]]]
```

{, }

```
Manipulate[
 Show[Image[1 / 2 + ListConvolve[Transpose[{identität[ordnung + 1]}].{identität[ordnung + 1]} - Transpose[{binom[ordnung]}].{binom[ordnung]},
     ImageData@wellenbild, ordnung / 2 + 1]], ImageSize → 2 * r + 1],
 {{ordnung, 2, "Binom-Ordnung"}, 0, 20, 2, Appearance → "Labeled"}, SaveDefinitions → True]
```



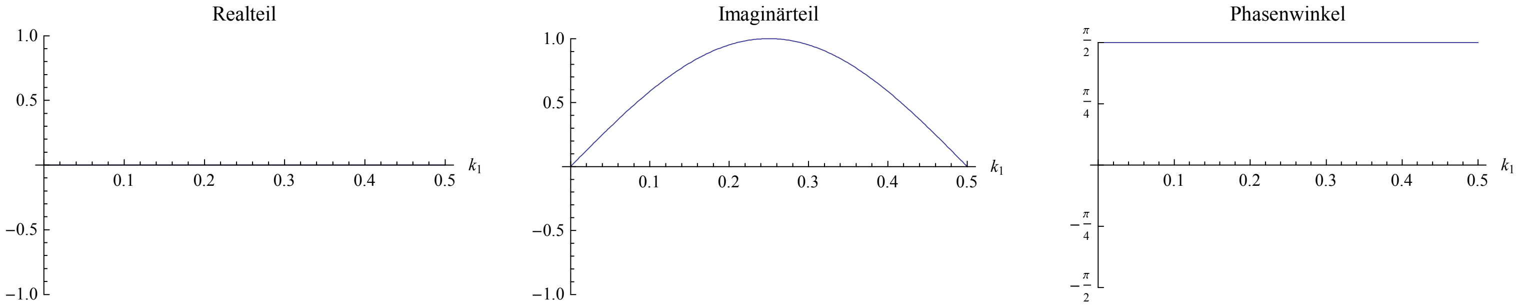
Approximation von Ableitungsoperatoren 1. Ordnung durch lineare Filter

```
ableitungsymm
```

$\left\{\frac{1}{2}, 0, -\frac{1}{2}\right\}$

```
transferfunktion[ableitungsymm]
Show[plottransferfunktion[transferfunktion[ableitungsymm]], ImageSize → 800]
```

$i \, \mathrm{Sin}[2\pi k_1]$

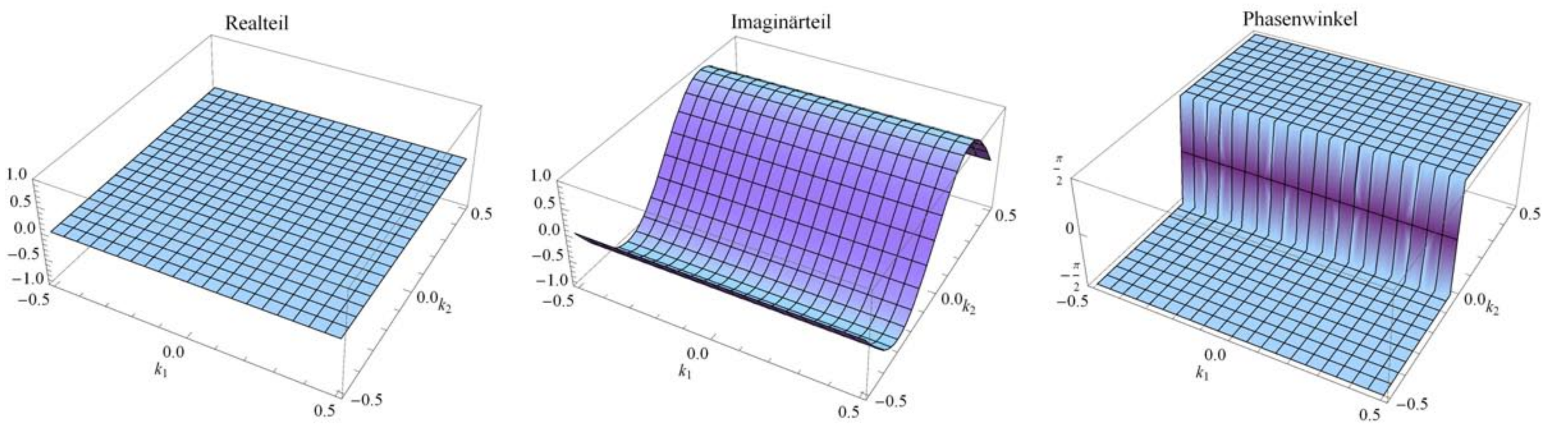



```
Transpose[{identität[3]}].{ableitungsymm} // MatrixForm
```

$$\begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 0 & 0 \end{pmatrix}$$

```
transferfunktion2d[Transpose[{identität[3]}].{ableitungsymm}]
Show[plottransferfunktion2d[transferfunktion2d[Transpose[{identität[3]}].{ableitungsymm}]], ImageSize → 800]
```

$i \, \mathrm{Sin}[2 \pi k_2]$

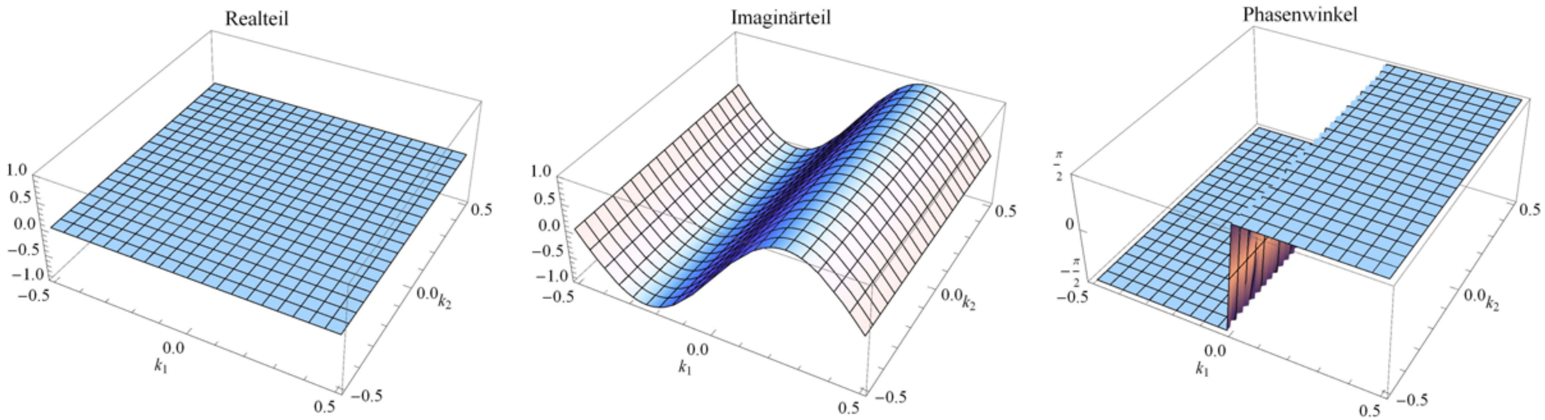



```
Transpose[{ableitungsymm}].{identität[3]} // MatrixForm
```

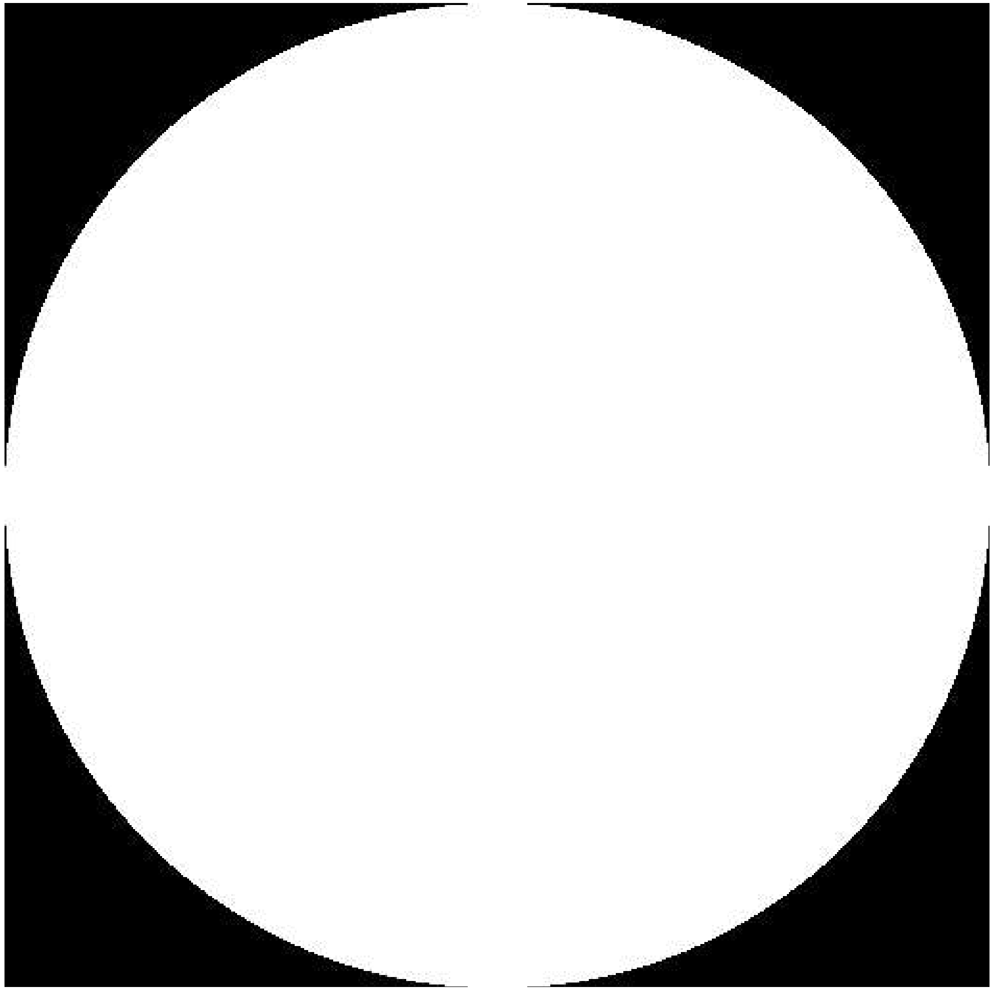
$$\begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 \end{pmatrix}$$

```
transferfunktion2d[Transpose[{ableitungsymm}].{identität[3]}]
Show[plottransferfunktion2d[transferfunktion2d[Transpose[{ableitungsymm}].{identität[3]}]], ImageSize → 800]
```

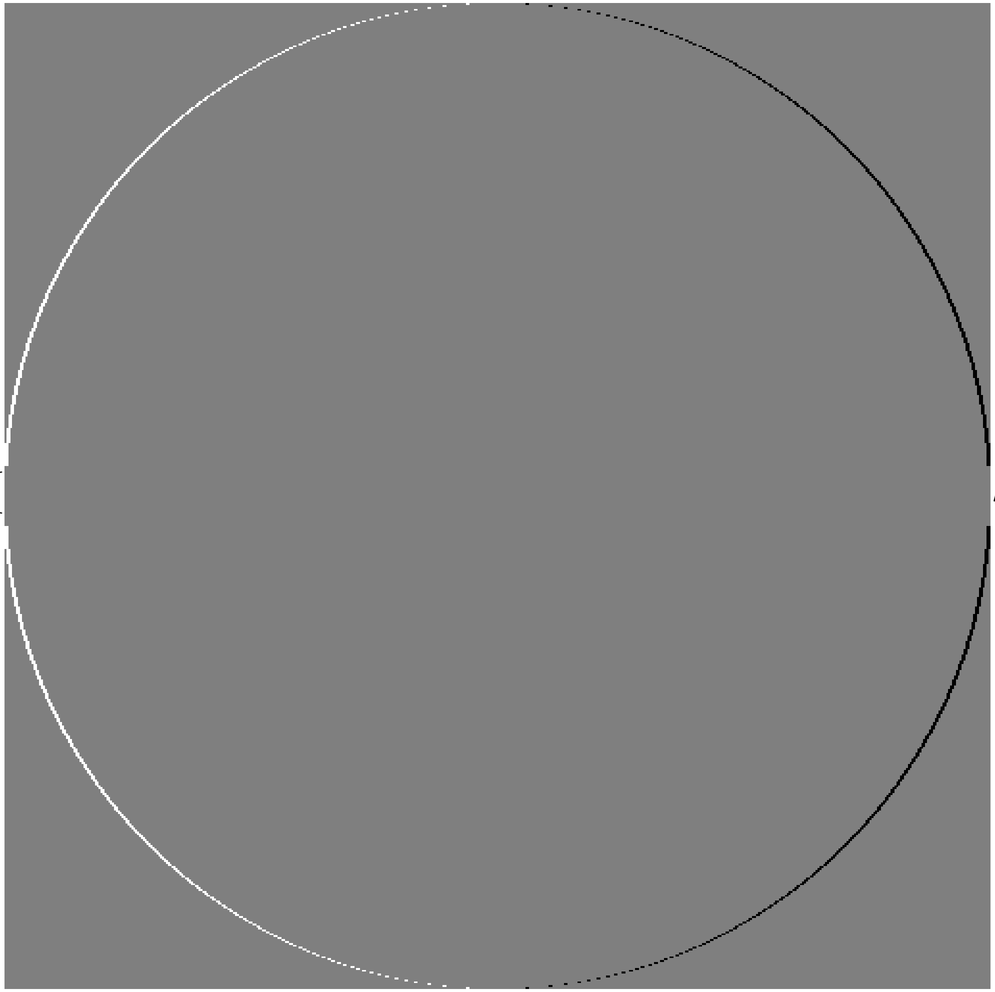
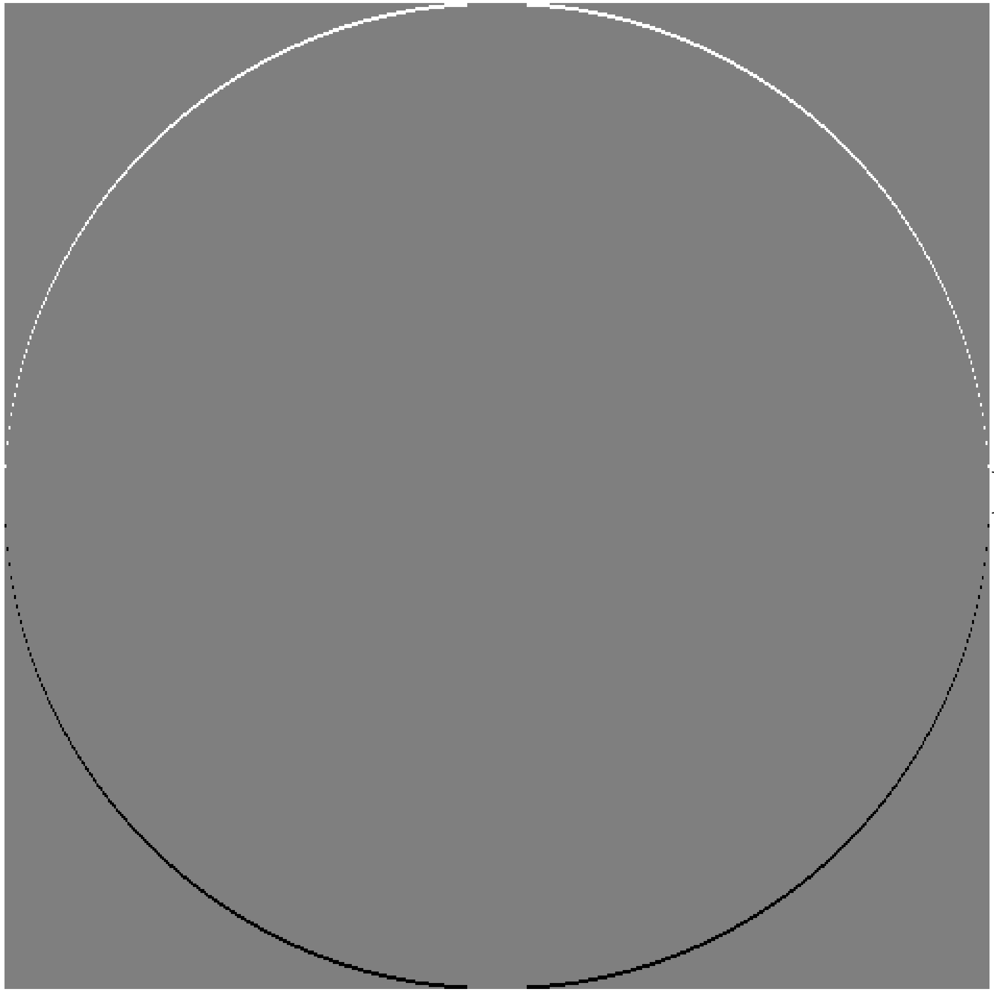
$i\,\mathrm{Sin}[2\,\pi\,k_1]$



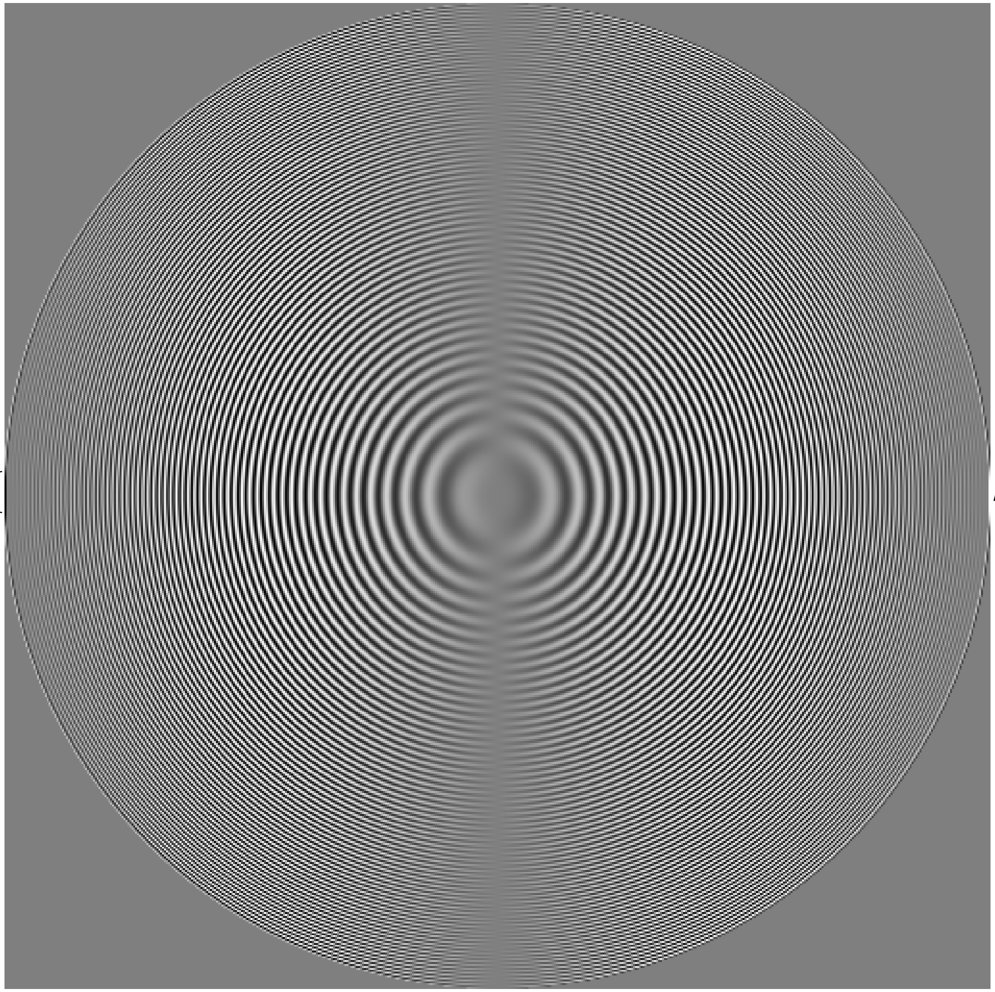
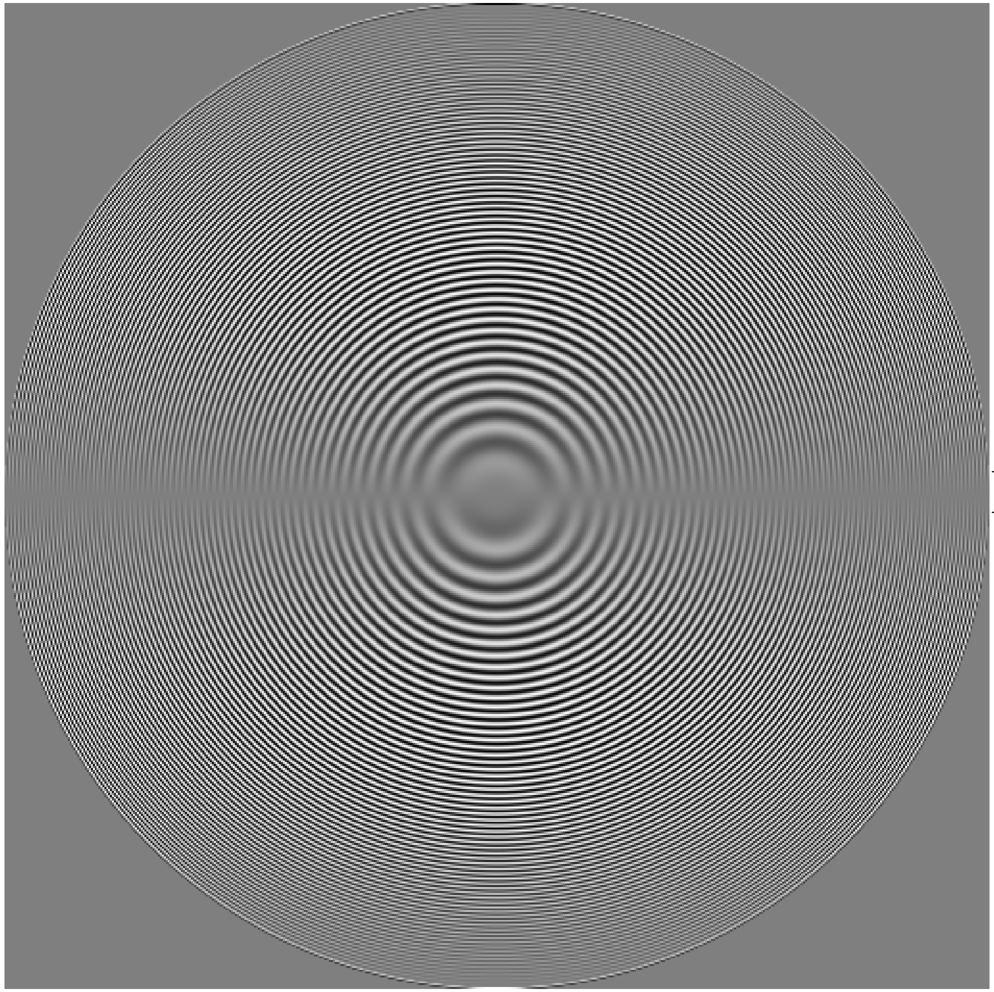
maske // Image



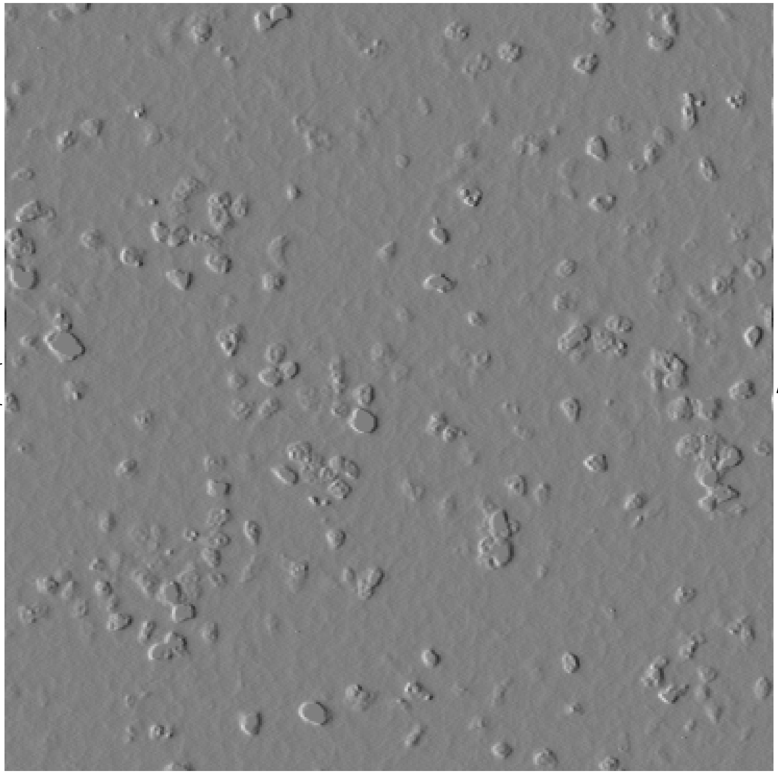
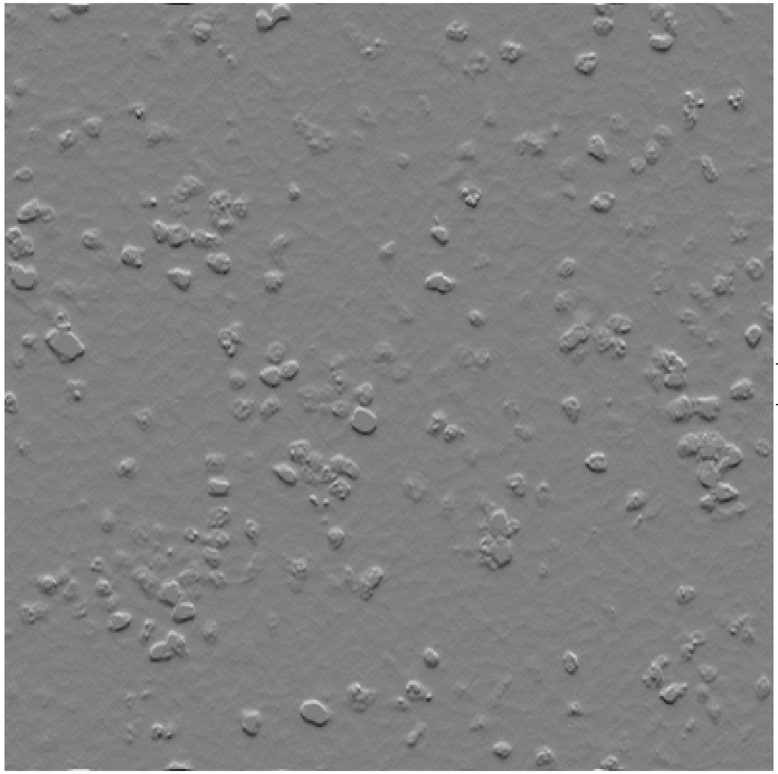
```
{Show[Image[0.5 + ListConvolve[#1, maske, {(Dimensions[#1] + 1) / 2}]]], Show[Image[0.5 + ListConvolve[#2, maske, {(Dimensions[#2] + 1) / 2}]]]} &[
 Transpose[{identität[3]}].{ableitungsymm}, Transpose[{ableitungsymm}].{identität[3]}]
```

{, }

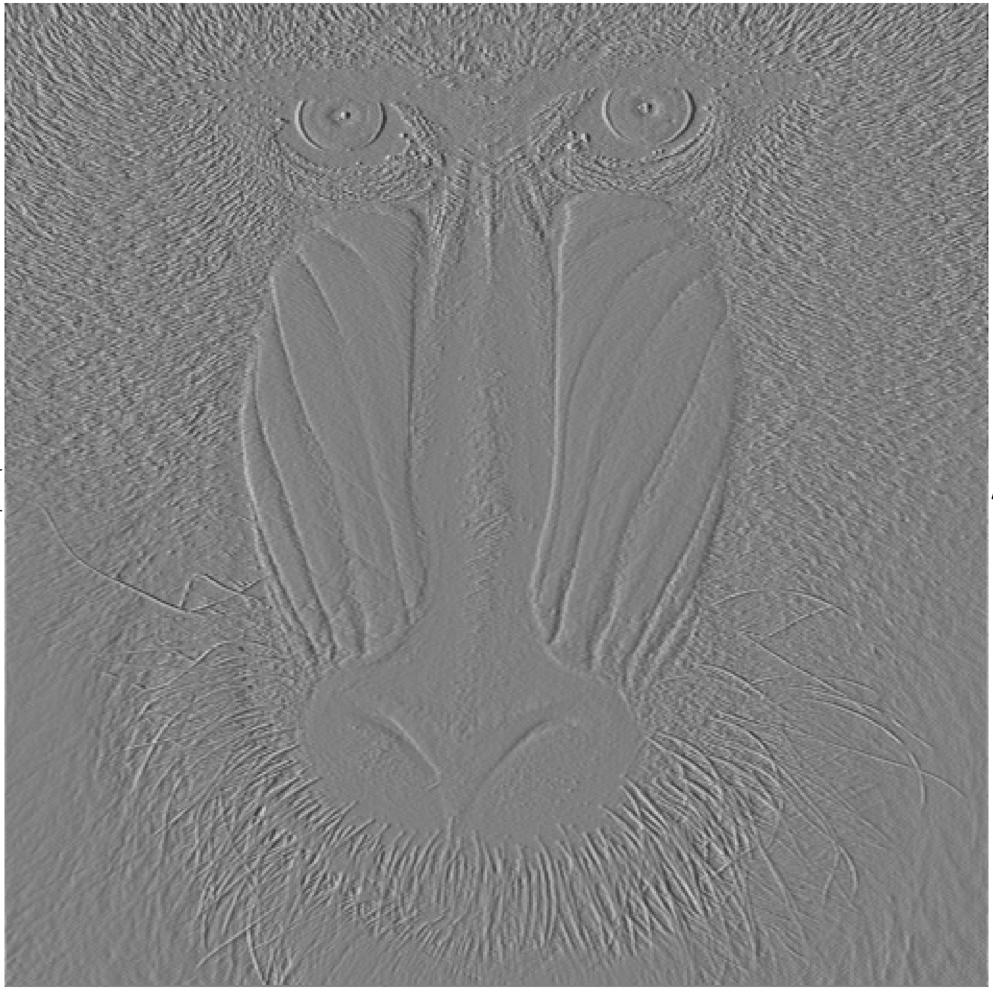
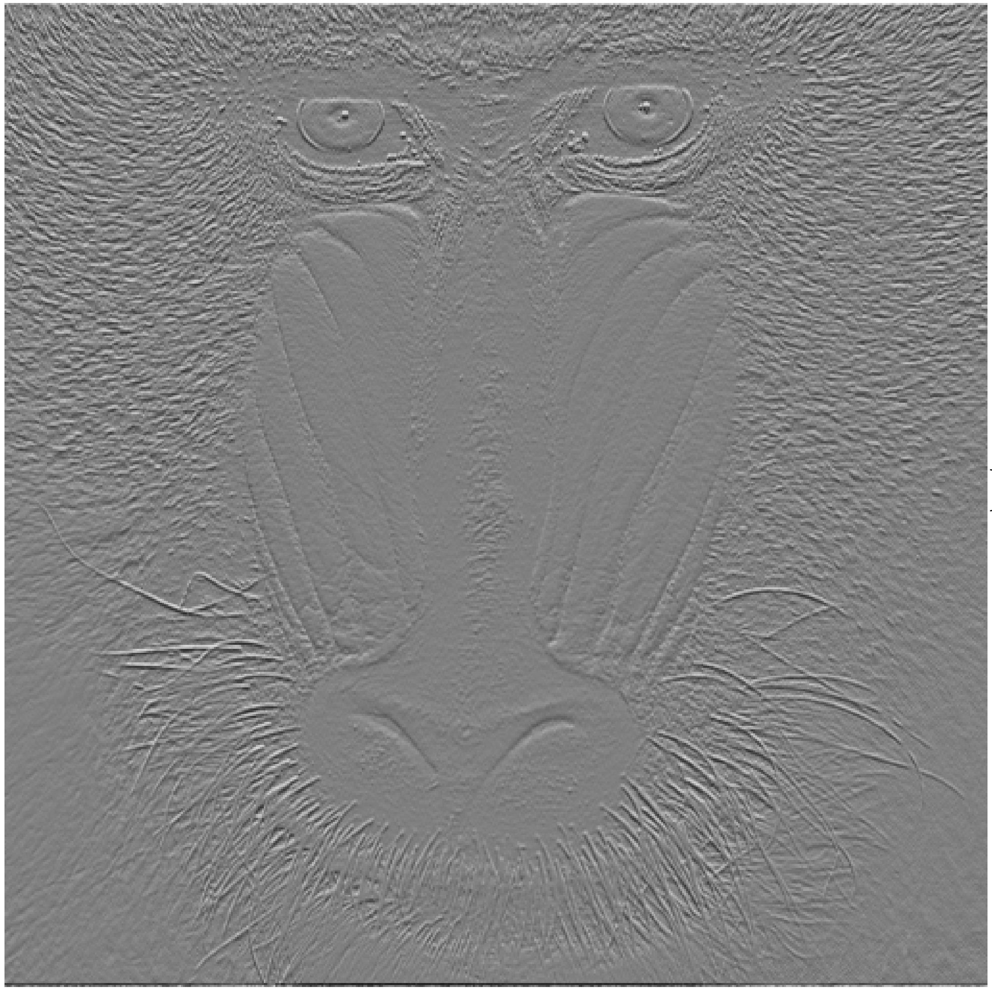
```
{Show[Image[0.5 + ListConvolve[#1, ImageData[wellenbild], {(Dimensions[#1] + 1) / 2}]]],
   Show[Image[0.5 + ListConvolve[#2, ImageData[wellenbild], {(Dimensions[#2] + 1) / 2}]]]} &[
 Transpose[{identität[3]}].{ableitungsymm}, Transpose[{ableitungsymm}].{identität[3]}]
```

{, }

```
{Show[Image[0.5 + ListConvolve[#1, ImageData@ImageCrop[First@ColorSeparate@Lym3CD21dreikanalausgleichF2, {pagewidth / 2, pagewidth / 2}],
        {(Dimensions[#1] + 1) / 2}]], ImageSize → pagewidth / 2],
   Show[Image[0.5 + ListConvolve[#2, ImageData@ImageCrop[First@ColorSeparate@Lym3CD21dreikanalausgleichF2, {pagewidth / 2, pagewidth / 2}],
        {(Dimensions[#2] + 1) / 2}]], ImageSize → pagewidth / 2]} &[
 Transpose[{identität[3]}].{ableitungsymm}, Transpose[{ableitungsymm}].{identität[3]}]
```

{, }

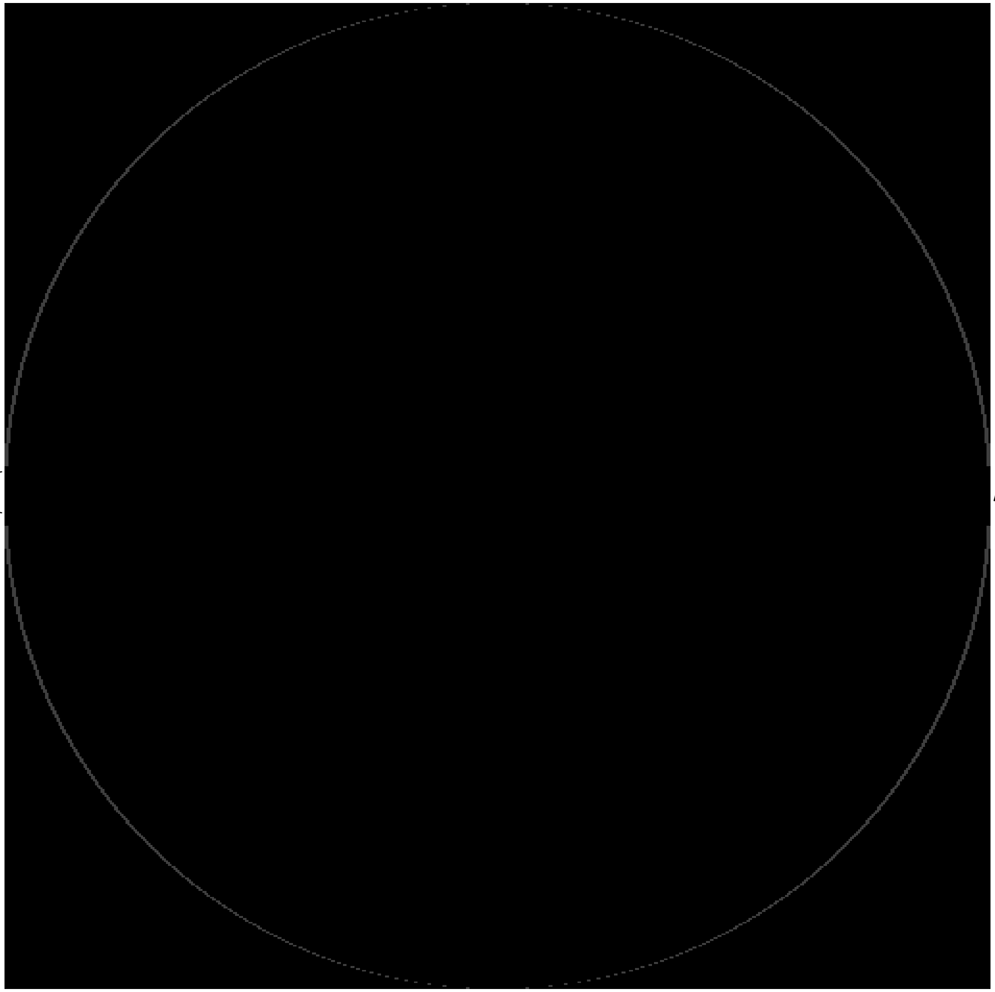
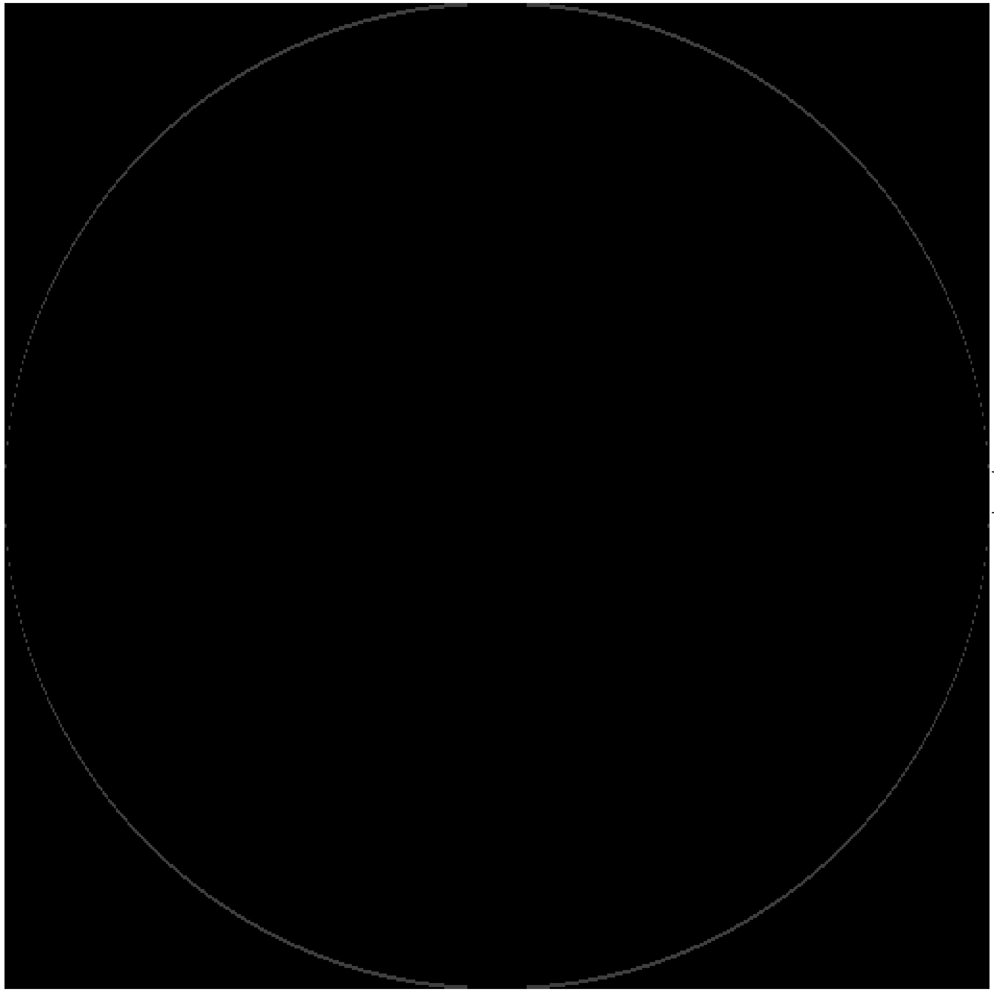
```
{Show[Image[0.5 + ListConvolve[#1,
      ImageData[ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Mandrill"}]]], {(Dimensions[#1] + 1) / 2}]]],
   Show[Image[0.5 + ListConvolve[#2, ImageData[ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Mandrill"}]]],
      {(Dimensions[#2] + 1) / 2}]]]} &[
 Transpose[{identität[3]}].{ableitungsymm}, Transpose[{ableitungsymm}].{identität[3]}]
```

{, }

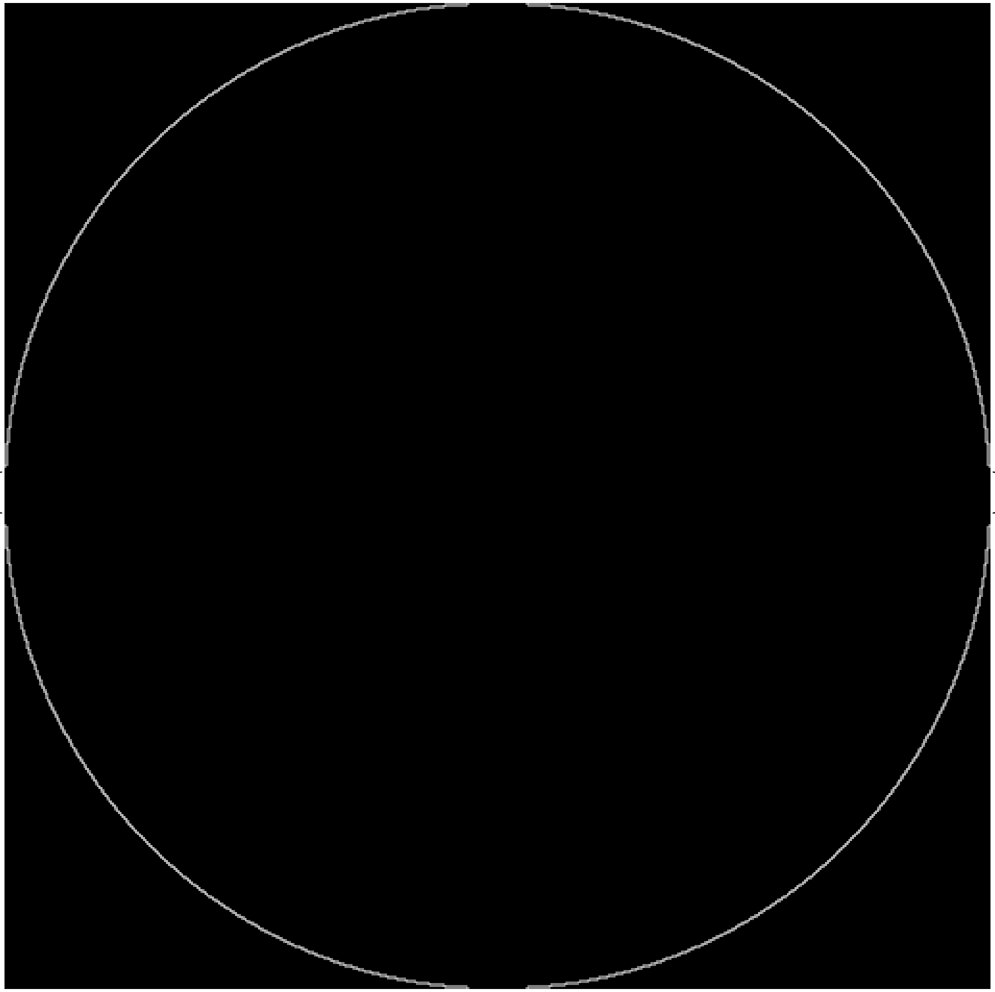
Um der erheblichen Richtungsabhängigkeit entgegenzuwirken, kann man tricksen:

- pixelweises Summieren der Quadrate der jeweiligen Richtungsableitungen und anschließendes Radizieren

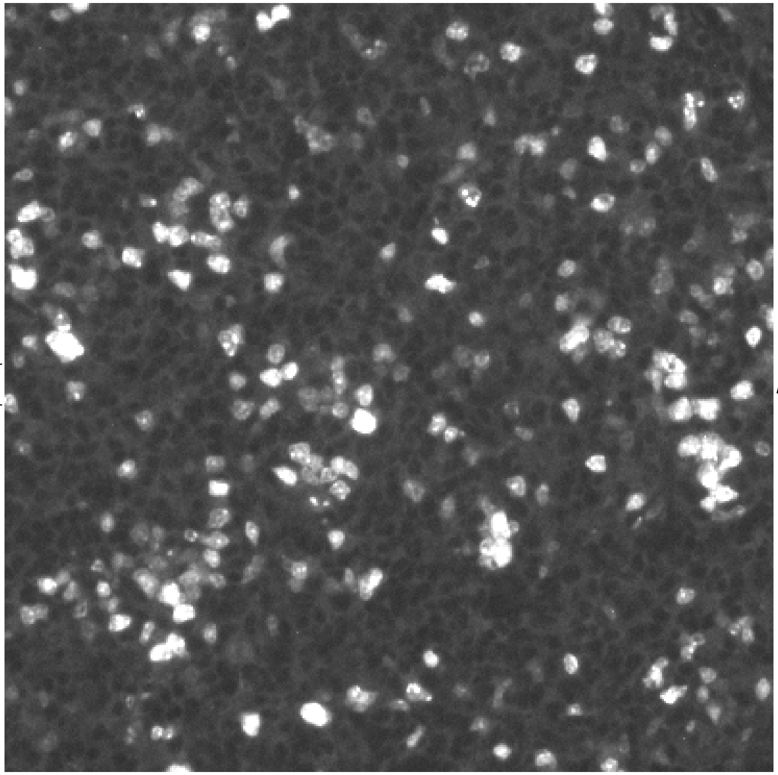
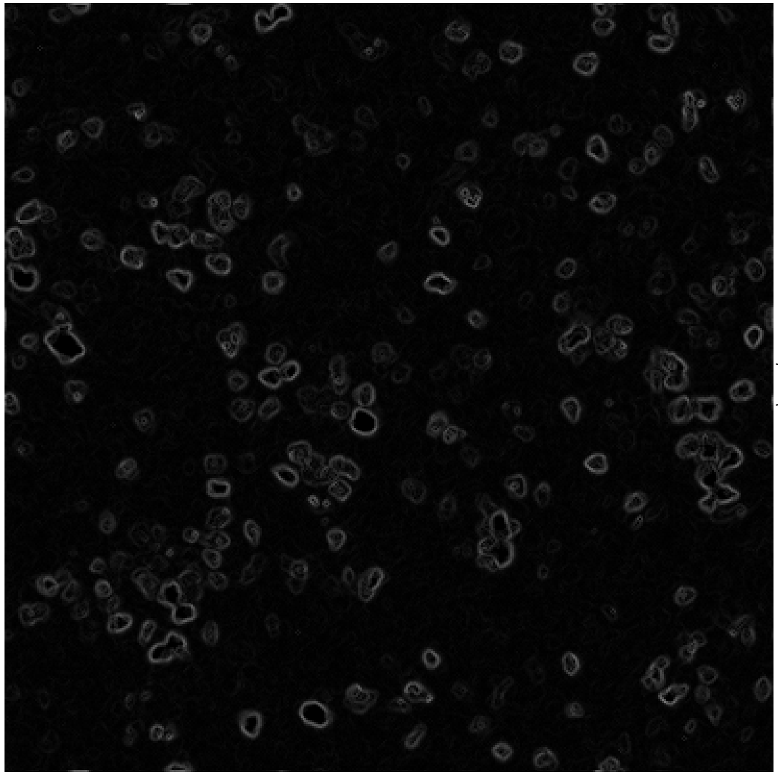
```
{Show[Image[ListConvolve[Transpose[{identität[3]}].{ableitungsymm}, #, {{2, 2}}]^2]],
   Show[Image[ListConvolve[Transpose[{ableitungsymm}].{identität[3]}, #, {{2, 2}}]^2]]} &[maske]
```

{  ,  }

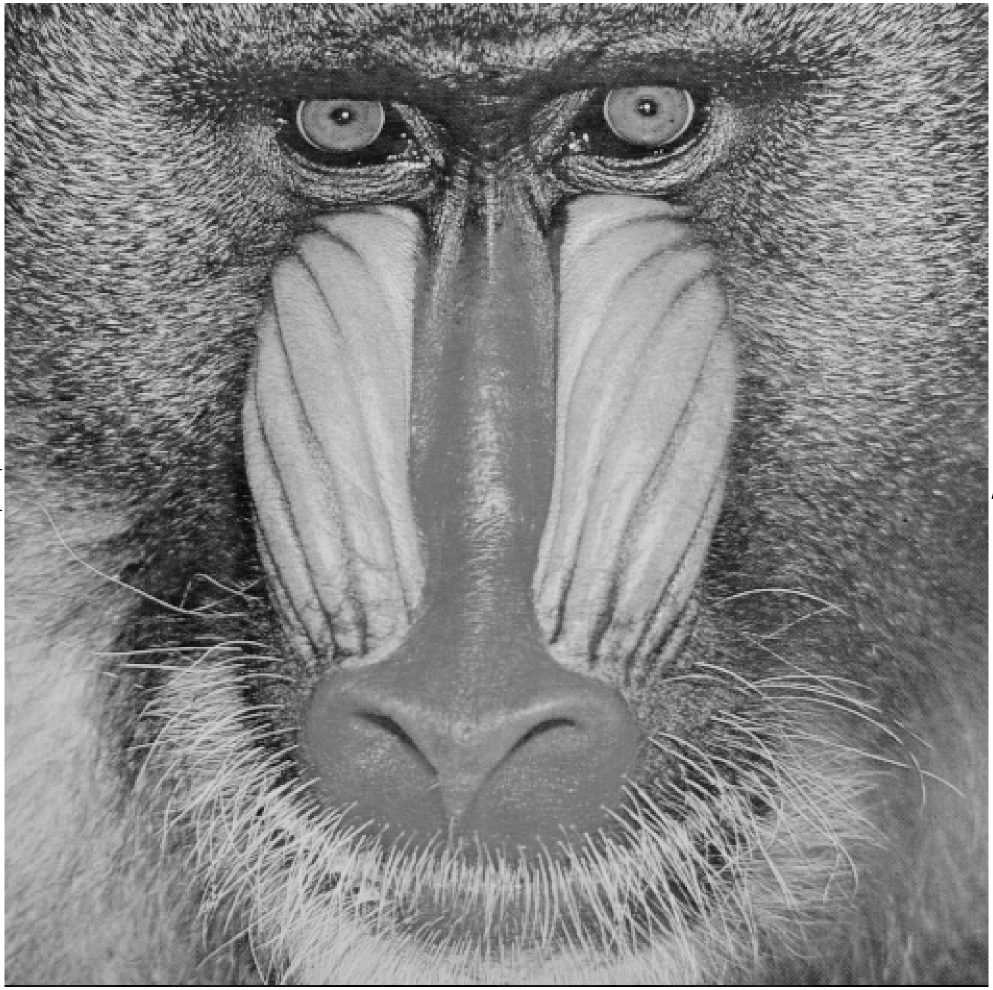
```
{Show[Image[Sqrt[ListConvolve[Transpose[{identität[3]}].{ableitungsymm}, #, {{2, 2}}]^2 +
      ListConvolve[Transpose[{ableitungsymm}].{identität[3]}, #, {{2, 2}}]^2]]]} &[maske]
```

{}

```
{Show[Image[#], ImageSize → pagewidth / 2], Show[Image[Sqrt[ListConvolve[Transpose[{identität[3]}].{ableitungsymm}, #, {{2, 2}}]^2 +
      ListConvolve[Transpose[{ableitungsymm}].{identität[3]}, #, {{2, 2}}]^2]], ImageSize → pagewidth / 2]} &[
 ImageData@ImageCrop[First@ColorSeparate@Lym3CD21dreikanalausgleichF2, {pagewidth / 2, pagewidth / 2}]]
```

{, }

```
{Show[Image[#]], Show[Image[Sqrt[ListConvolve[Transpose[{identität[3]}].{ableitungsymm}, #, {{2, 2}}]^2 +
      ListConvolve[Transpose[{ableitungsymm}].{identität[3]}, #, {{2, 2}}]^2]]]} &[
 ImageData@ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Mandrill"}]]]
```

{, }

```
{Show[Image[#]], Show[Image[Sqrt[ListConvolve[Transpose[{identität[3]}].{ableitungsymm}, #, {{2, 2}}]^2 +
       ListConvolve[Transpose[{ableitungsymm}].{identität[3]}, #, {{2, 2}}]^2]]]} &[
 ImageData@ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Lena"}]]]
```

{, }

Gibt es verfeinerte Gradientenoperatoren?

Idee: mehr als zwei Richtungen, Querglättung

- Sobelfilter

```
Transpose[{binom[2]}].{ableitungsymm} // MatrixForm
```

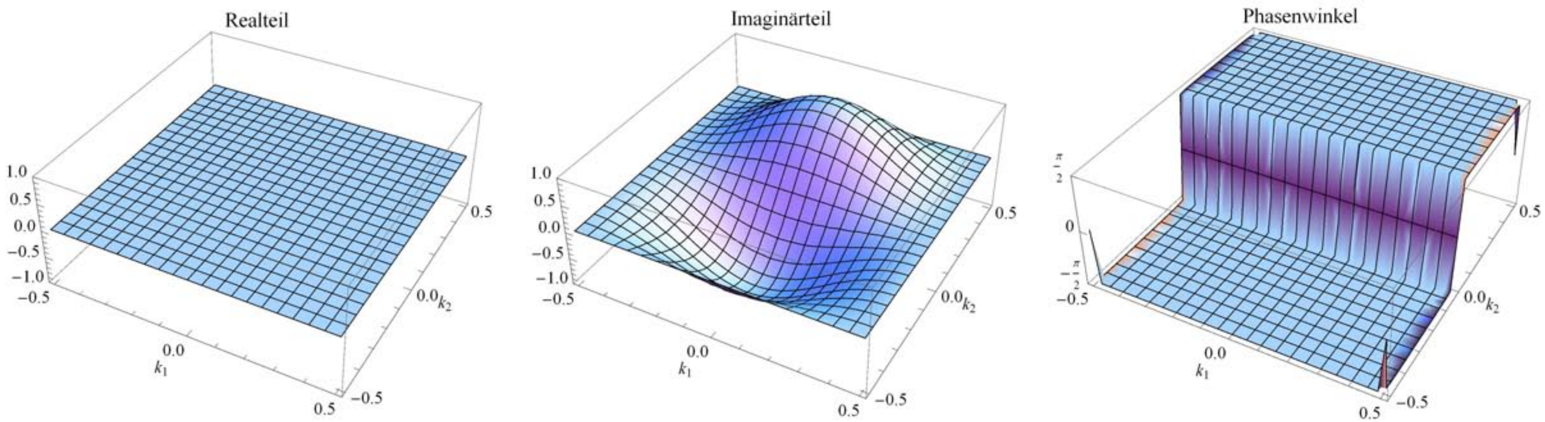
$$\begin{pmatrix} \frac{1}{8} & 0 & -\frac{1}{8} \\ \frac{1}{4} & 0 & -\frac{1}{4} \\ \frac{1}{8} & 0 & -\frac{1}{8} \end{pmatrix}$$

```
sobel1 = {{1/8, 0, -1/8}, {1/4, 0, -1/4}, {1/8, 0, -1/8}};
```

$$\text{sobel1} = \begin{pmatrix} \frac{1}{8} & 0 & -\frac{1}{8} \\ \frac{1}{4} & 0 & -\frac{1}{4} \\ \frac{1}{8} & 0 & -\frac{1}{8} \end{pmatrix};$$

```
transferfunktion2d[sobel1]
Show[plottransferfunktion2d[transferfunktion2d[sobel1]], ImageSize → pagewidth]
```

$i \, \mathrm{Cos}[\pi k_1]^2 \, \mathrm{Sin}[2 \pi k_2]$



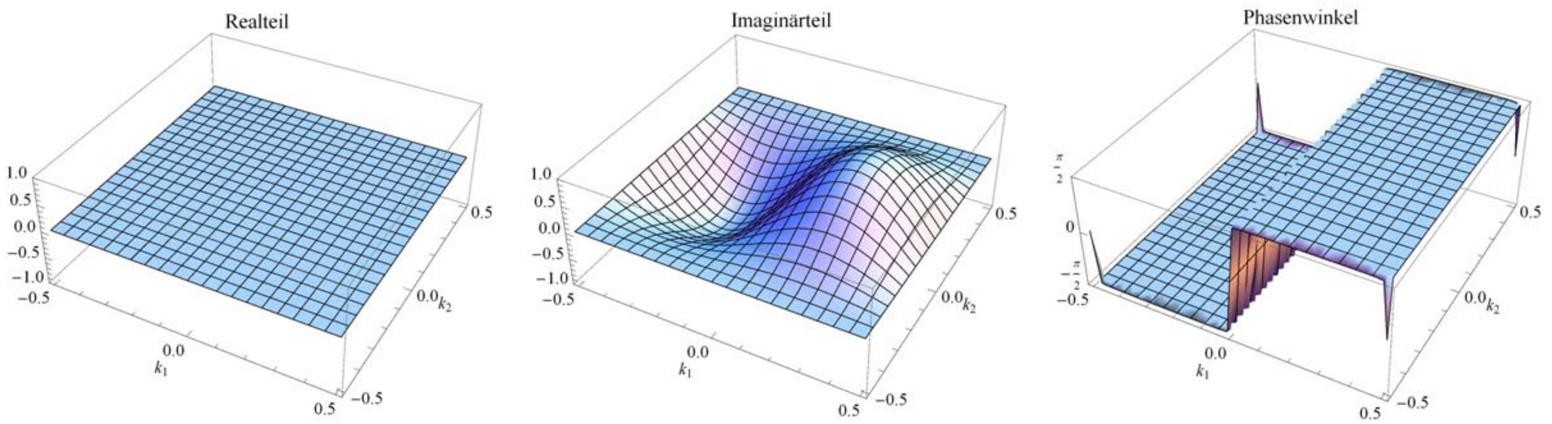
```
Transpose[{ableitungsymm}].{binom[2]} // MatrixForm
```

$$\begin{pmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ 0 & 0 & 0 \\ -\frac{1}{8} & -\frac{1}{4} & -\frac{1}{8} \end{pmatrix}$$

sobel2 = $\begin{pmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ 0 & 0 & 0 \\ -\frac{1}{8} & -\frac{1}{4} & -\frac{1}{8} \end{pmatrix}$;

```
transferfunktion2d[sobel2]
Show[plottransferfunktion2d[transferfunktion2d[sobel2]], ImageSize → pagewidth]
```

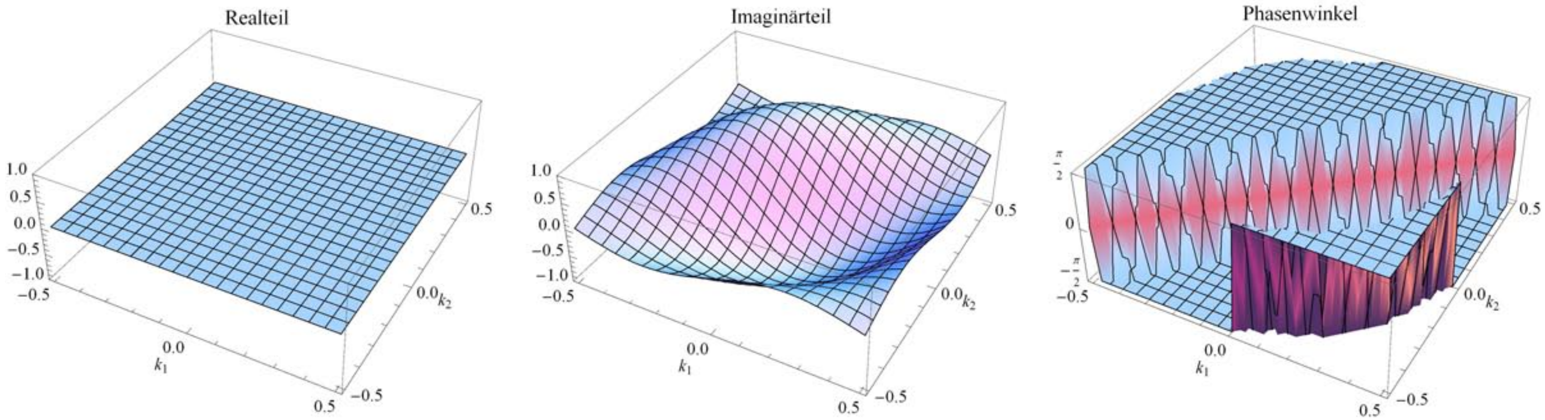
$i\, \mathrm{Cos}[\pi\, k_2]^2\, \mathrm{Sin}[2\, \pi\, k_1]$



sobel3 = $\begin{pmatrix} 0 & -\frac{1}{8} & -\frac{1}{4} \\ \frac{1}{8} & 0 & -\frac{1}{8} \\ \frac{1}{4} & \frac{1}{8} & 0 \end{pmatrix}$;

```
transferfunktion2d[sobel3]
Show[plottransferfunktion2d[transferfunktion2d[sobel3]], ImageSize → pagewidth]
```

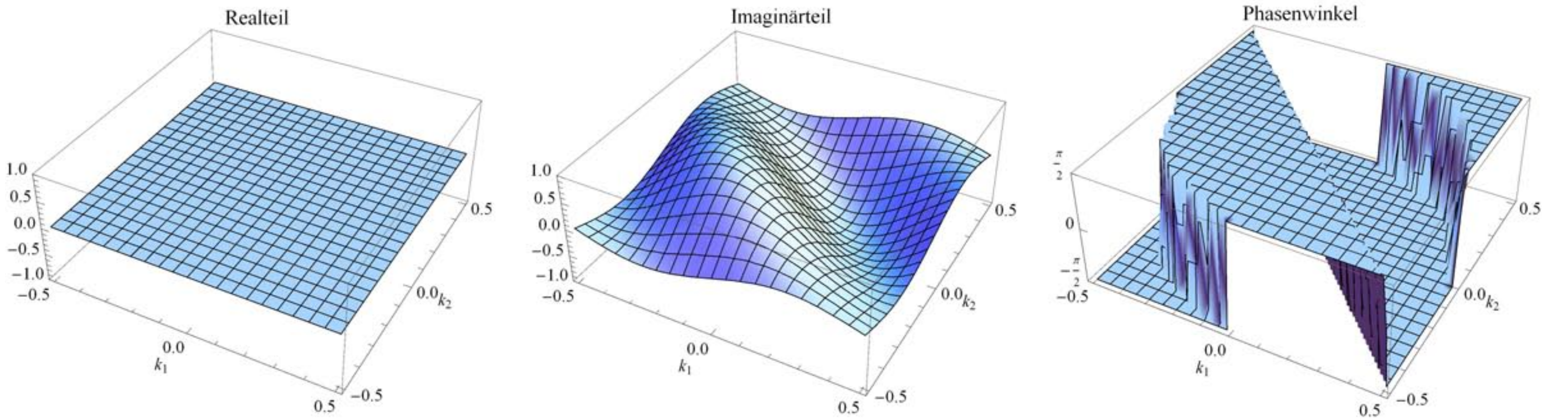
$$-\frac{1}{4}\, i\, (\text{Sin}[2\,\pi\, k_1] + 2\,\text{Sin}[2\,\pi\,(k_1 - k_2)] - \text{Sin}[2\,\pi\, k_2])$$



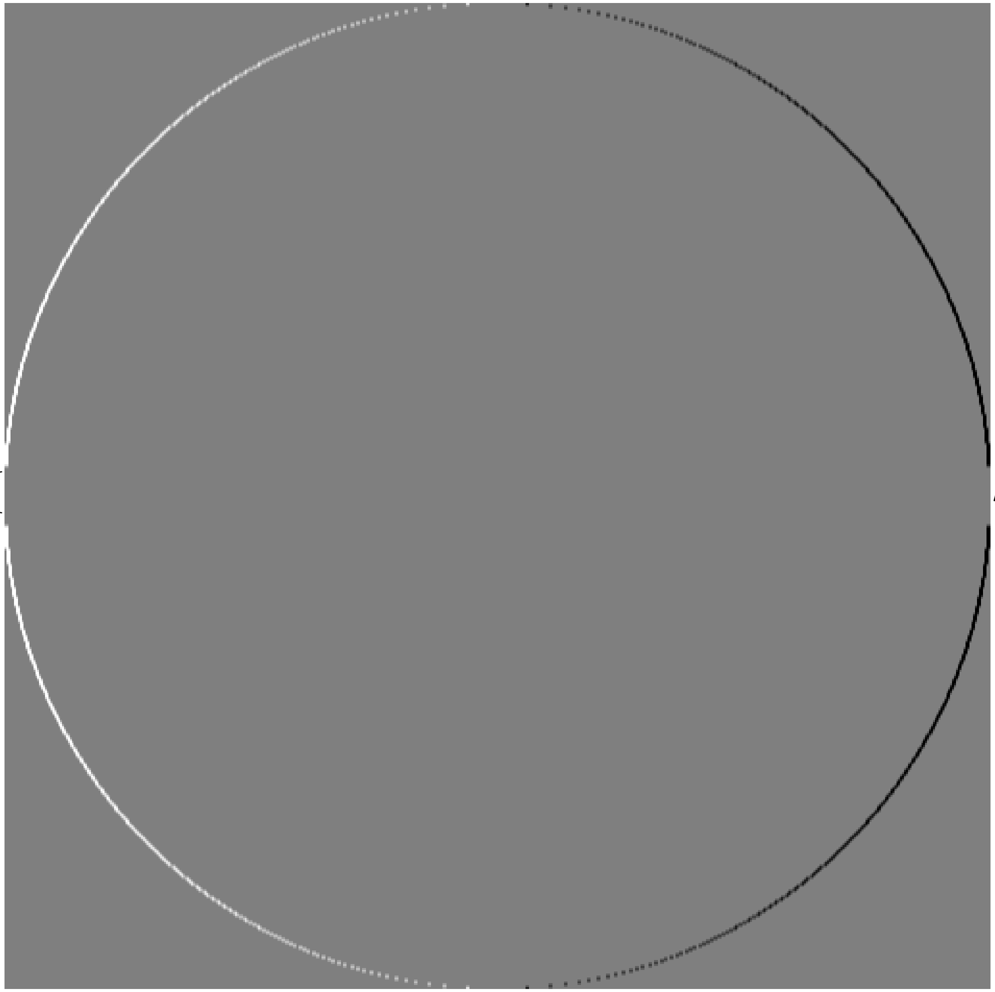
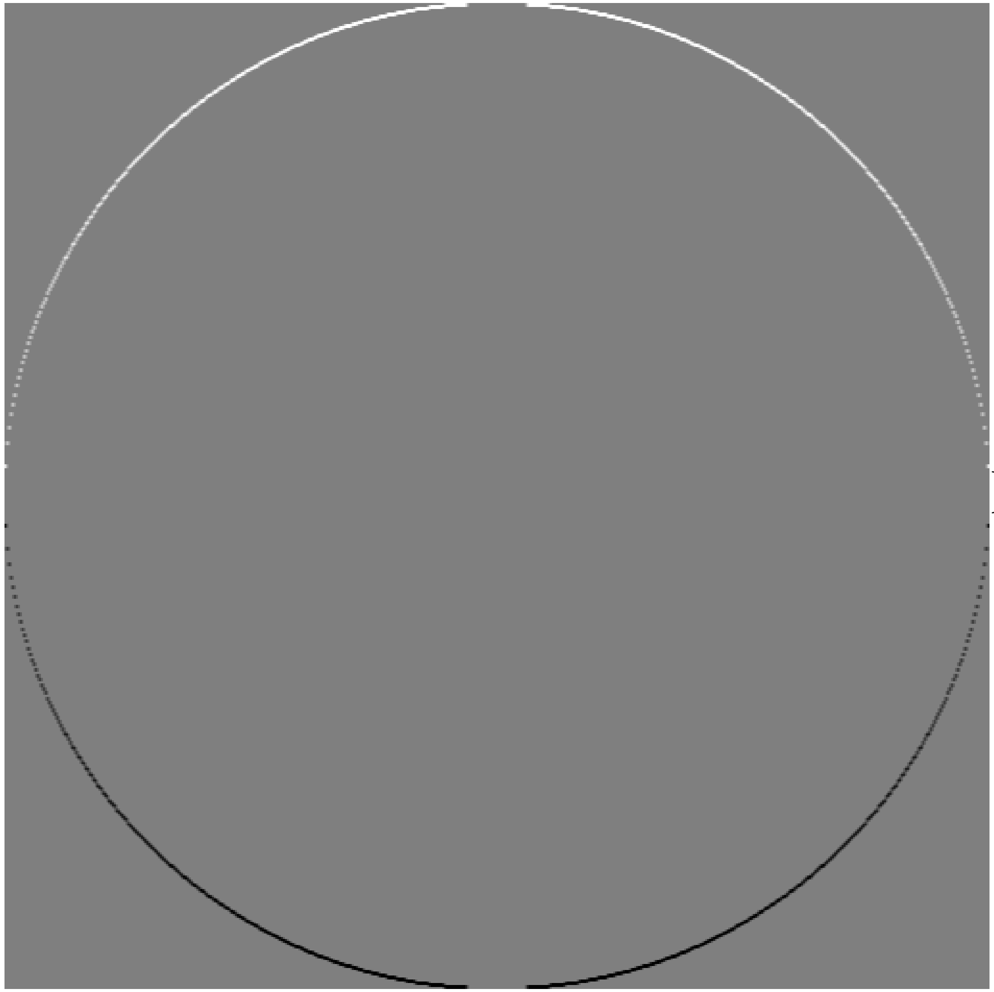
$$\text{sobel4} = \begin{pmatrix} -\frac{1}{4} & -\frac{1}{8} & 0 \\ -\frac{1}{8} & 0 & \frac{1}{8} \\ 0 & \frac{1}{8} & \frac{1}{4} \end{pmatrix};$$

```
transferfunktion2d[sobel4]
Show[plottransferfunktion2d[transferfunktion2d[sobel4]], ImageSize → pagewidth]
```

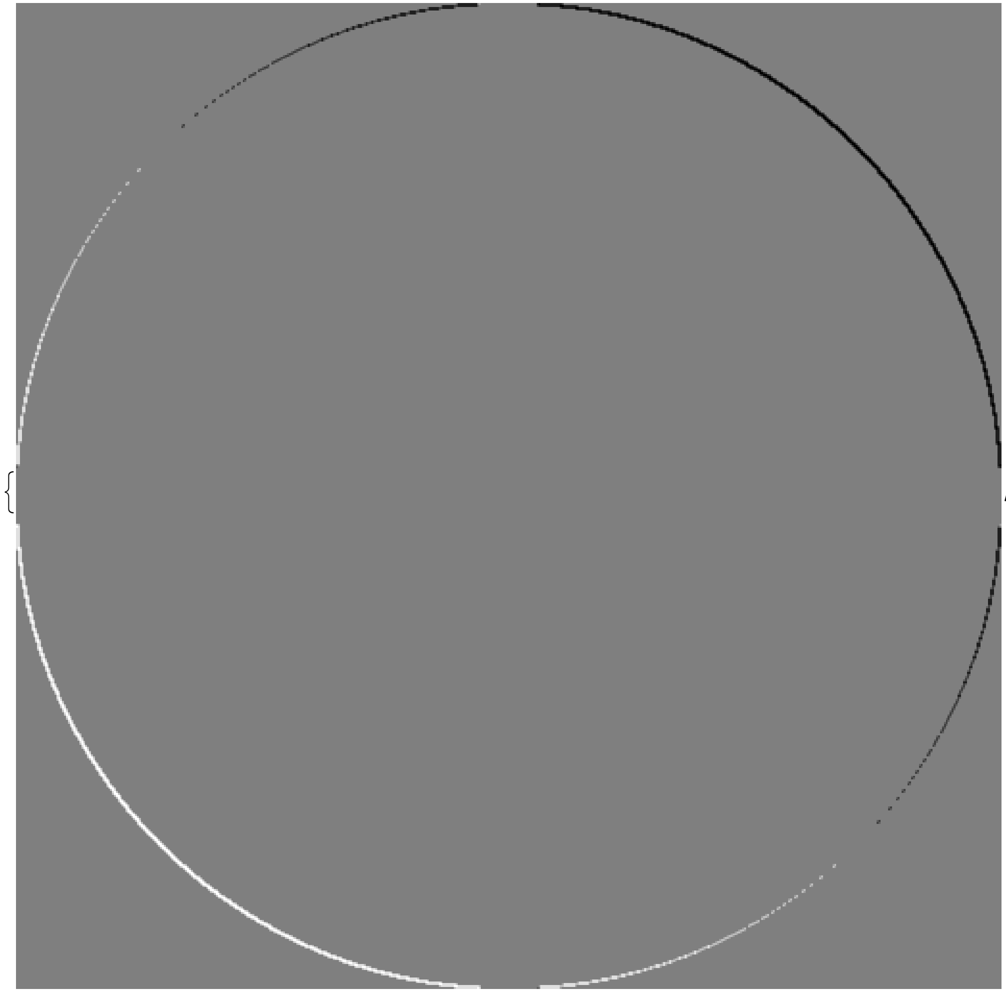
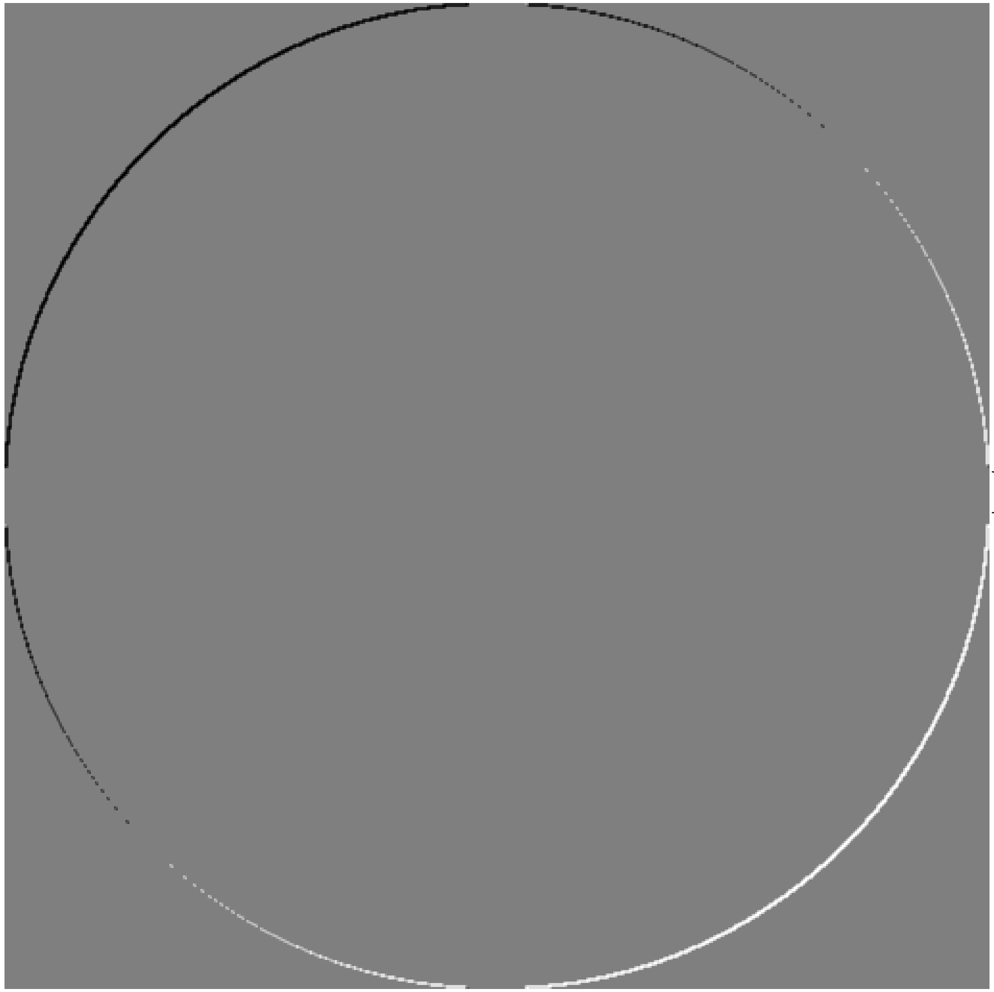
$$-\frac{1}{4}\, i\, (\mathrm{Sin}[2\pi k_1] + \mathrm{Sin}[2\pi k_2] + 2\,\mathrm{Sin}[2\pi (k_1 + k_2)])$$



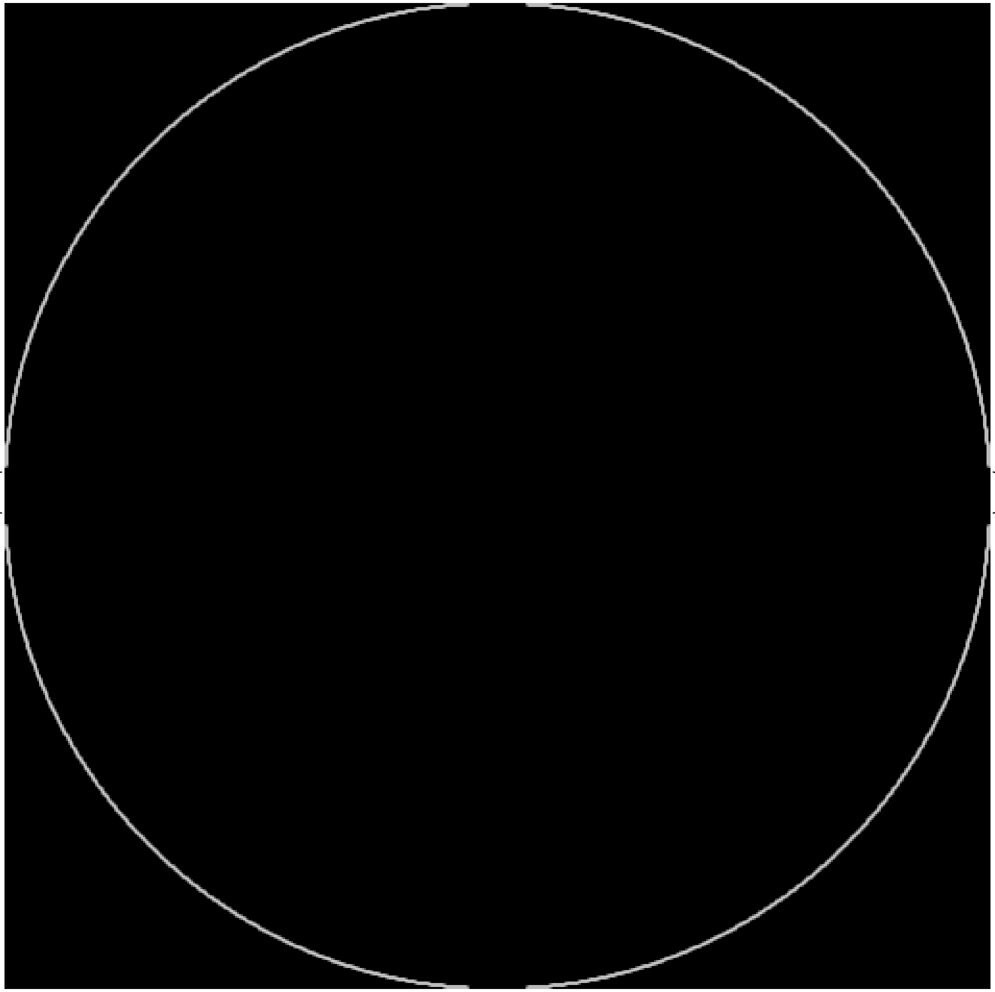
```
{Show[Image[0.5 + ListConvolve[sobel1, #, {{2, 2}}]]], Show[Image[0.5 + ListConvolve[sobel2, #, {{2, 2}}]]]} &[maske]
```

{, }

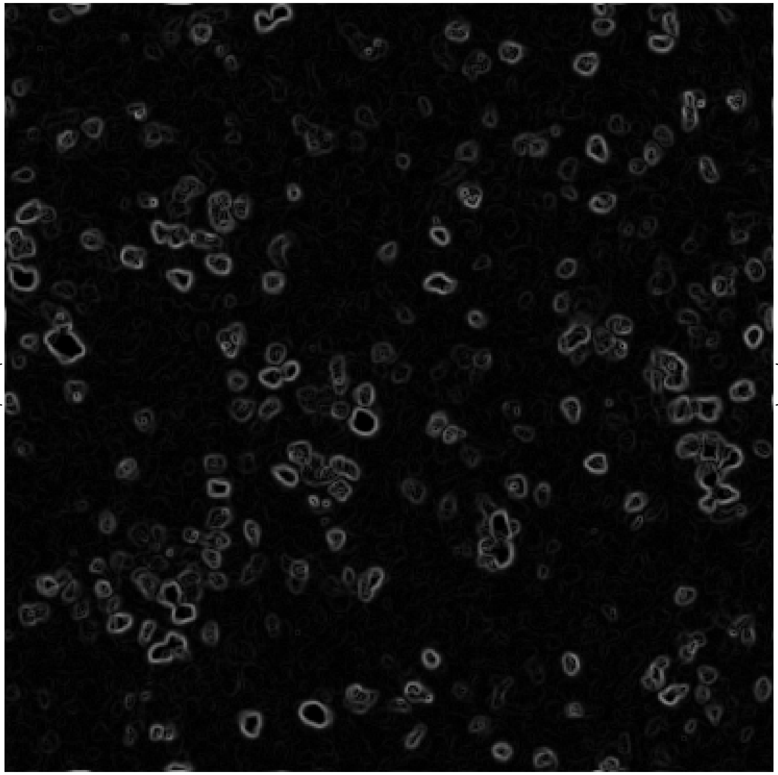
```
{Show[Image[0.5 + ListConvolve[sobel3, #, {{2, 2}}]]], Show[Image[0.5 + ListConvolve[sobel4, #, {{2, 2}}]]]} &[maske]
```

{, }

```
{Show[Image[Sqrt[ListConvolve[sobel1, #, {{2, 2}}]^2 + ListConvolve[sobel2, #, {{2, 2}}]^2 +
      ListConvolve[sobel3, #, {{2, 2}}]^2 + ListConvolve[sobel4, #, {{2, 2}}]^2]]]} &[maske]
```

{}

```
{Show[Image[Sqrt[ListConvolve[sobel1, #, {{2, 2}}]^2 + ListConvolve[sobel2, #, {{2, 2}}]^2 +
        ListConvolve[sobel3, #, {{2, 2}}]^2 + ListConvolve[sobel4, #, {{2, 2}}]^2]], ImageSize → pagewidth / 2]} &[
 ImageData@ImageCrop[First@ColorSeparate@Lym3CD21dreikanalausgleichF2, {pagewidth / 2, pagewidth / 2}]]
```

{}

```
{Show[Image[Sqrt[ListConvolve[sobel1, #, {{2, 2}}]^2 +
      ListConvolve[sobel2, #, {{2, 2}}]^2 + ListConvolve[sobel3, #, {{2, 2}}]^2 + ListConvolve[sobel4, #, {{2, 2}}]^2]]]} &[
 ImageData@ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Mandrill"}]]]
```

{}

```
{Show[Image[Sqrt[ListConvolve[sobel1, #, {{2, 2}}]^2 +
      ListConvolve[sobel2, #, {{2, 2}}]^2 + ListConvolve[sobel3, #, {{2, 2}}]^2 + ListConvolve[sobel4, #, {{2, 2}}]^2]]]} &[
 ImageData@ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Lena"}]]]
```

{}

Approximation von Ableitungsoperatoren 2. Ordnung durch lineare Filter

Idee: aus zweimaliger Anwendung von Ableitungsoperatoren, Unabhängigkeit von konstanten Änderungen

- Laplacefilter
- Addition der beiden Richtungs-Laplacefilter

```
ableitungsymm
```

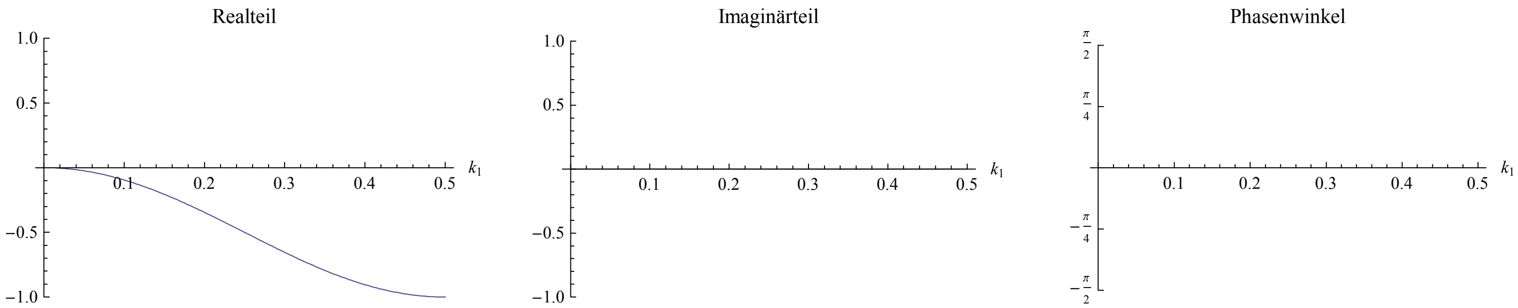
$\left\{\frac{1}{2}, 0, -\frac{1}{2}\right\}$

```
ListConvolve[ableitungsymm, ableitungsymm, 2]
```

$\left\{\frac{1}{4}, -\frac{1}{2}, \frac{1}{4}\right\}$

```
transferfunktion[ListConvolve[ableitungsymm, ableitungsymm, 2]]
Show[plottransferfunktion[transferfunktion[ListConvolve[ableitungsymm, ableitungsymm, 2]]], ImageSize → pagewidth]
```

$-\mathrm{Sin}[\pi k_1]^2$




```
laplacekern = Transpose[{identität[3]}].{ListConvolve[ableitungsymm, ableitungsymm, 2]} +
   Transpose[{ListConvolve[ableitungsymm, ableitungsymm, 2]}].{identität[3]};
(*als Addition für diesen einfachsten Fall!*)
```

```
laplacekern // MatrixForm
```

$$\begin{pmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & -1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix}$$

```
Transpose[{identität[3]}].{ListConvolve[ableitungminus, ableitungplus, 2]} // MatrixForm
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

```
Transpose[{ListConvolve[ableitungminus, ableitungplus, 2]}].{identität[3]} // MatrixForm
```

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

```
(Transpose[{ListConvolve[ableitungminus, ableitungplus, 2]}].{identität[3]} +
    Transpose[{identität[3]}].{ListConvolve[ableitungminus, ableitungplus, 2]}) / 4 // MatrixForm
```

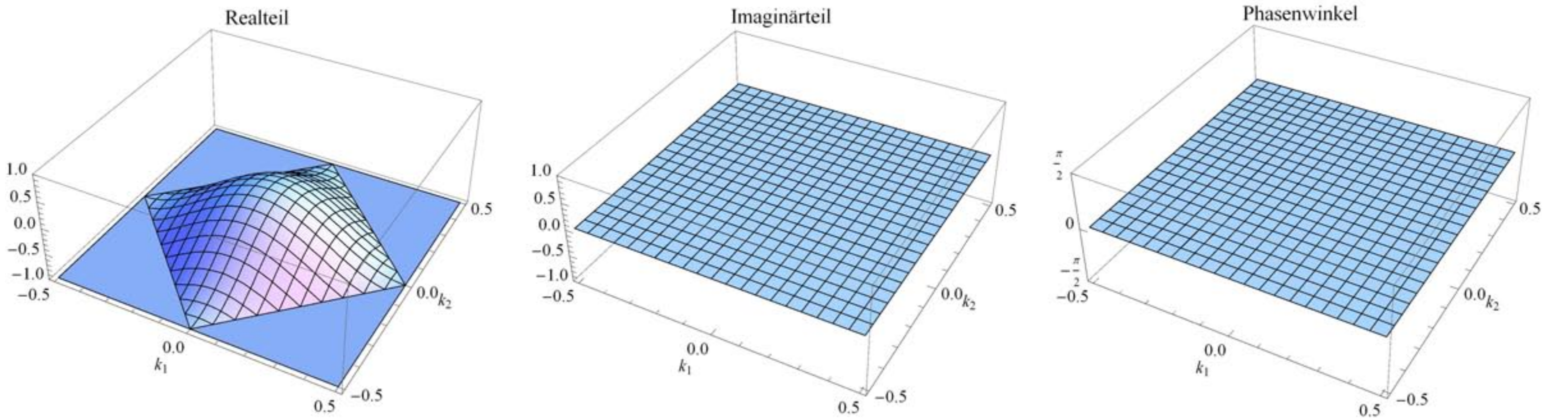
$$\begin{pmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & -1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix}$$

```
MatrixRank[laplacekern]
```

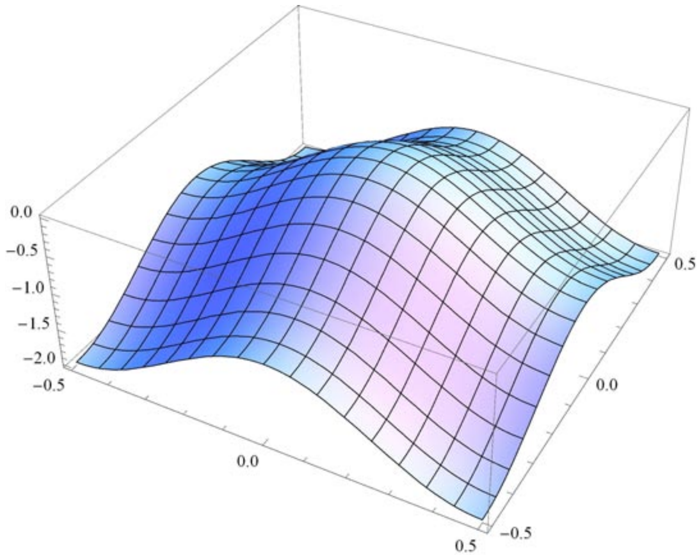
2

```
transferfunktion2d[laplacekern]
Show[plottransferfunktion2d[transferfunktion2d[laplacekern]], ImageSize → pagewidth]
```

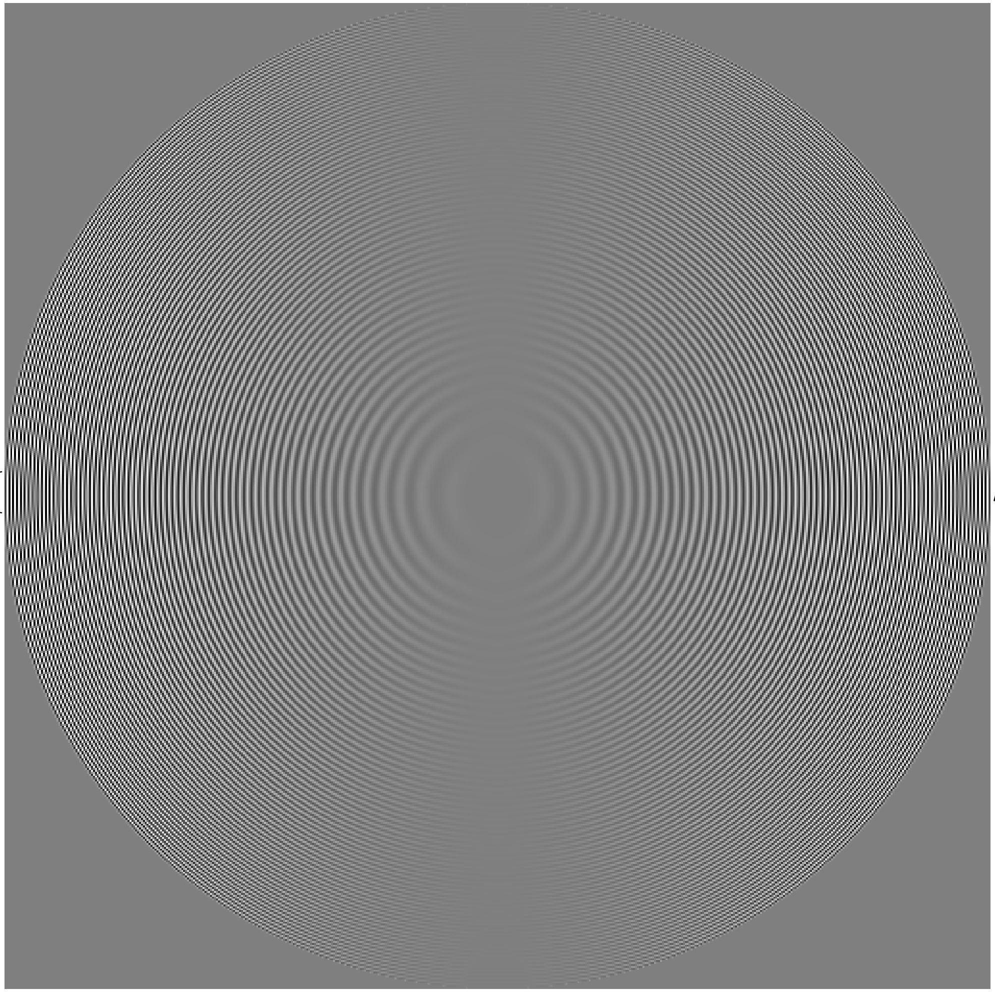
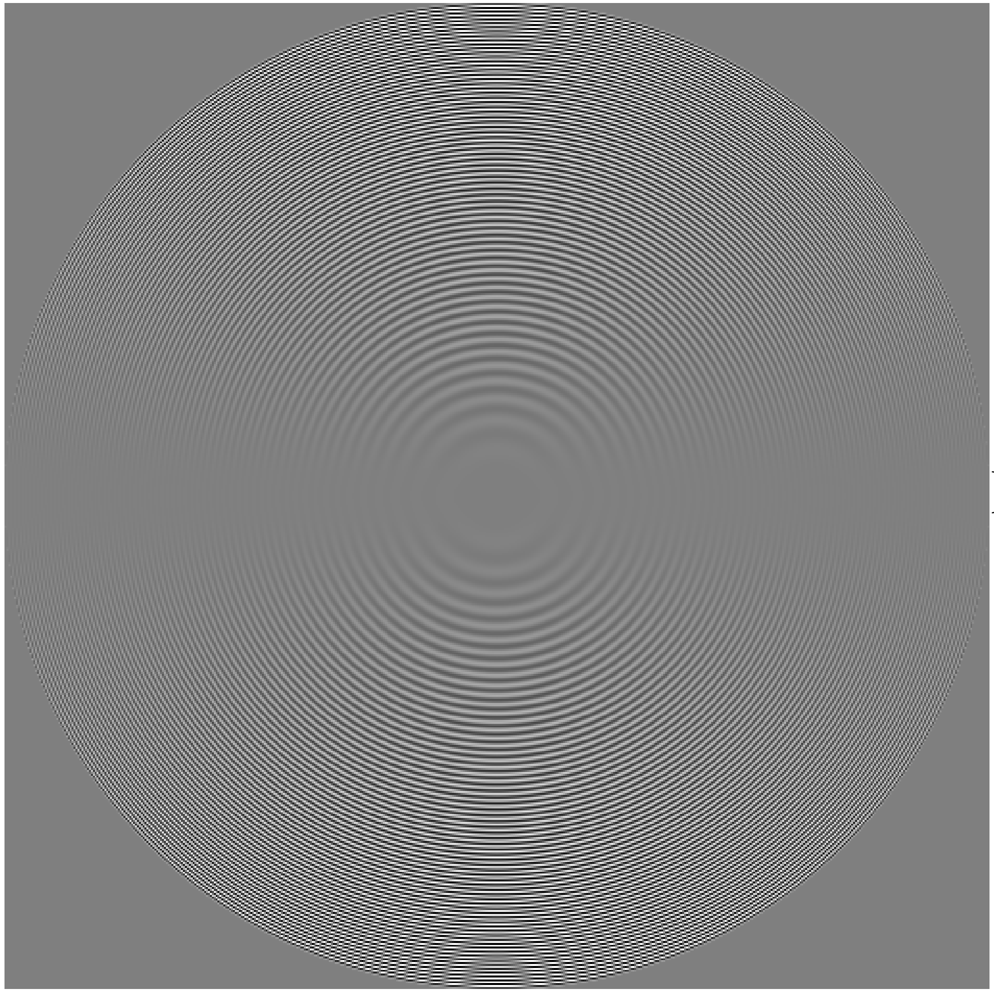
$$\frac{1}{2}\left(-2+\text{Cos}[2\,\pi\,k_1]+\text{Cos}[2\,\pi\,k_2]\right)$$



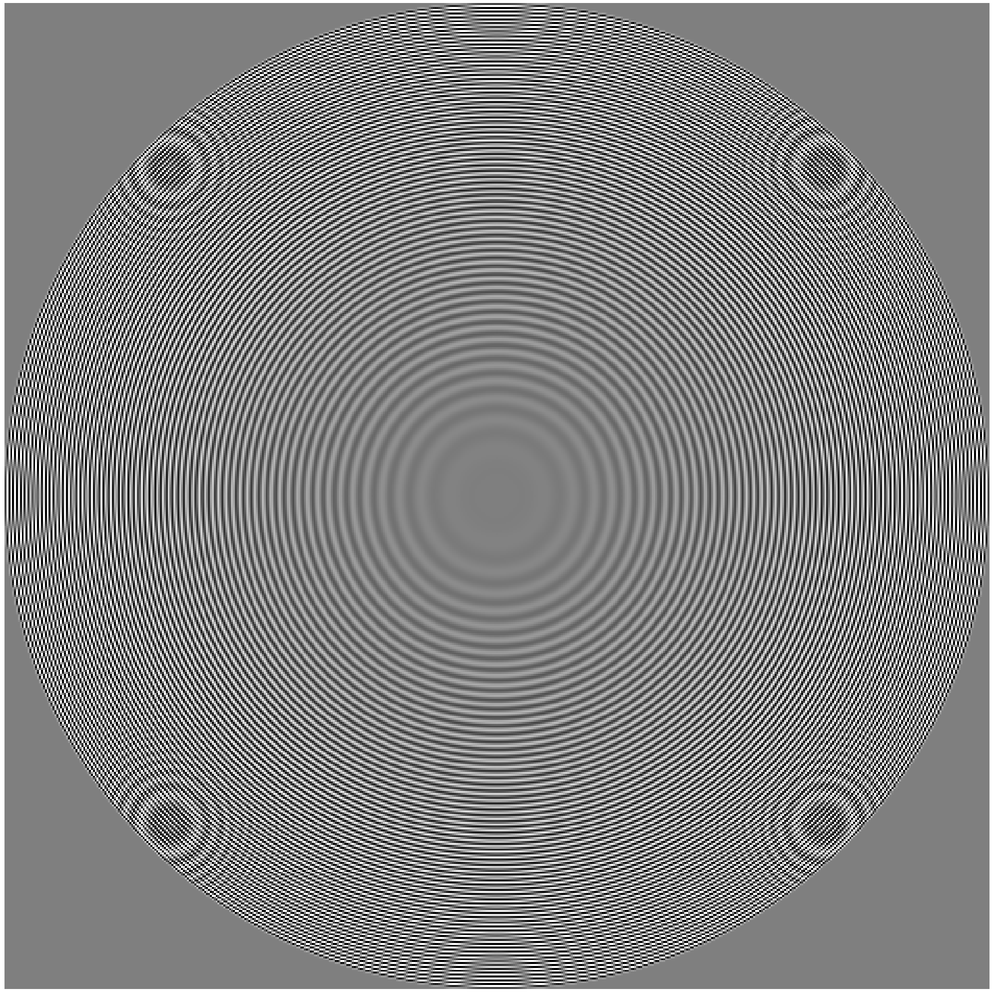

```
Plot3D[Evaluate[transferfunktion2d[laplacekern]], {k1, -1 / 2, 1 / 2}, {k2, -1 / 2, 1 / 2}]
```



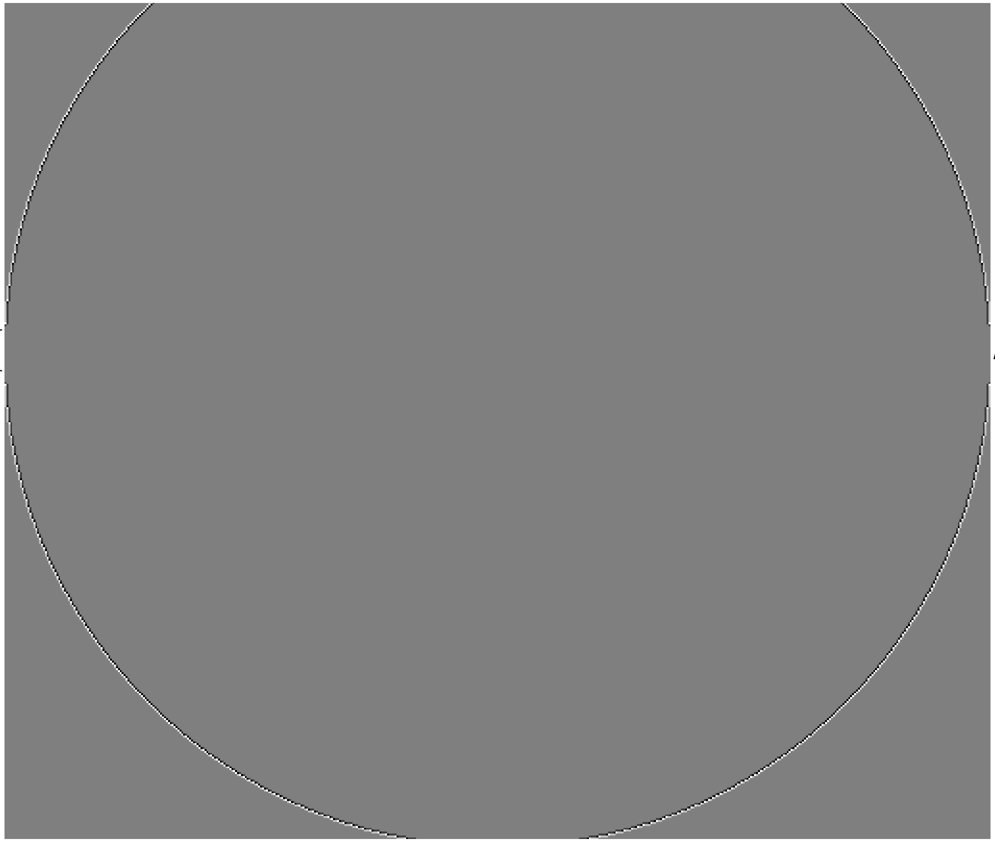
```
{Show[Image[0.5 + ListConvolve[#1, ImageData[wellenbild], {(Dimensions[#1] + 1) / 2}]]],
   Show[Image[0.5 + ListConvolve[#2, ImageData[wellenbild], {(Dimensions[#2] + 1) / 2}]]]} &[
 Transpose[{identität[3]}].{ListConvolve[ableitungsymm, ableitungsymm, 2]},
 Transpose[{ListConvolve[ableitungsymm, ableitungsymm, 2]}].{identität[3]}]
```

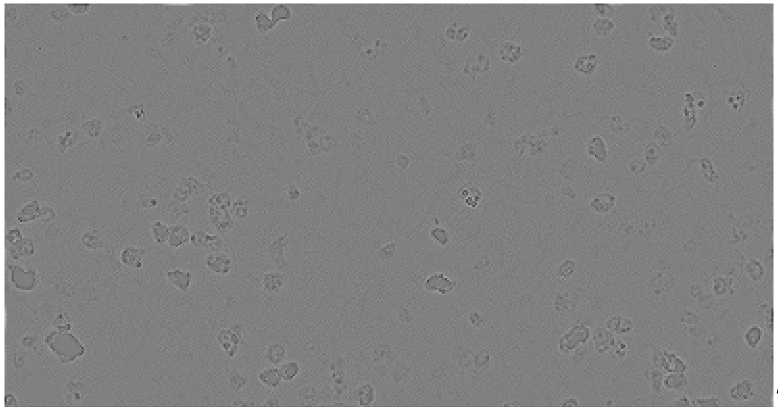
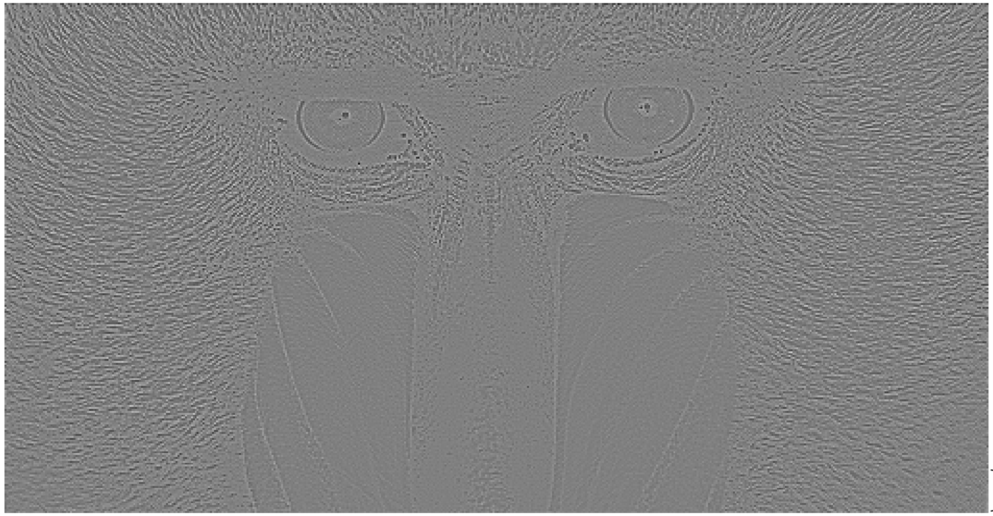
{, }

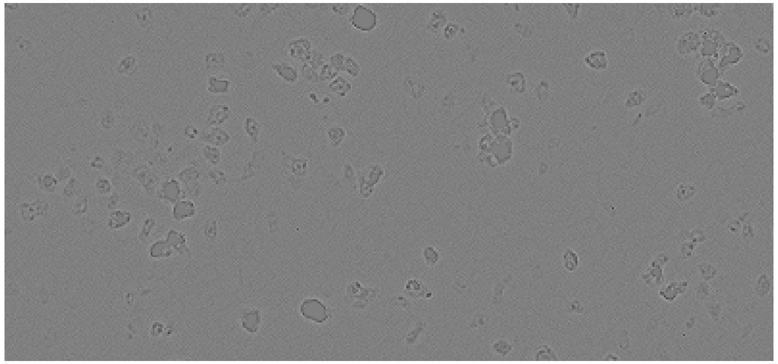
```
Show[Image[0.5 + ListConvolve[#1, #2, {(Dimensions[#1] + 1) / 2}]]] &[laplacekern, ImageData[wellenbild]]
```

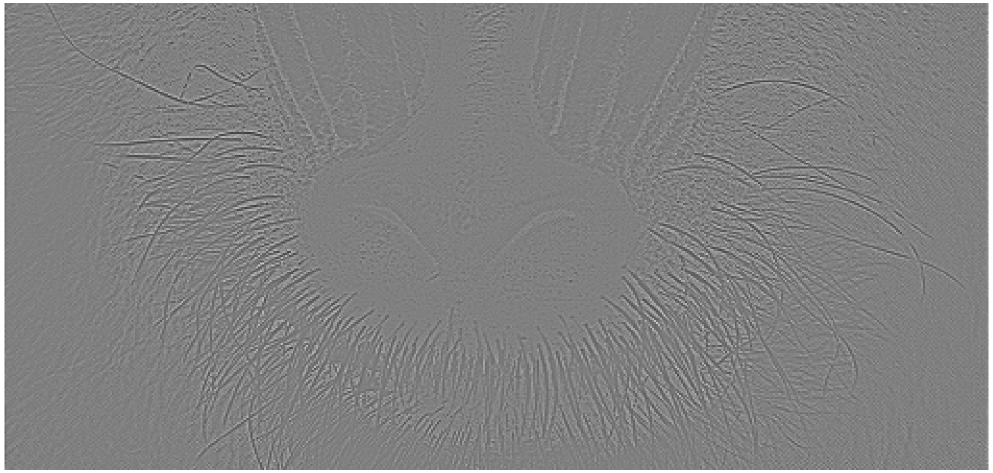


```
{Show[Image[0.5 + ListConvolve[#1, maske, {(Dimensions[#1] + 1) / 2}]]],
  Show[Image[0.5 + ListConvolve[#1, ImageData@ImageCrop[First@ColorSeparate@Lym3CD21dreikanalausgleichF2, {pagewidth / 2, pagewidth / 2}],
      {(Dimensions[#1] + 1) / 2}]], ImageSize → pagewidth / 2],
  Show[Image[0.5 + ListConvolve[#1, ImageData[ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Mandrill"}]]],
      {(Dimensions[#1] + 1) / 2}]]]} &[laplacekern]
```

{,

, }





Zusatz: Wie kann man die Glättungsfilter “versteilern”?

Glättungsfilter 1D mit steilerem Abfall:

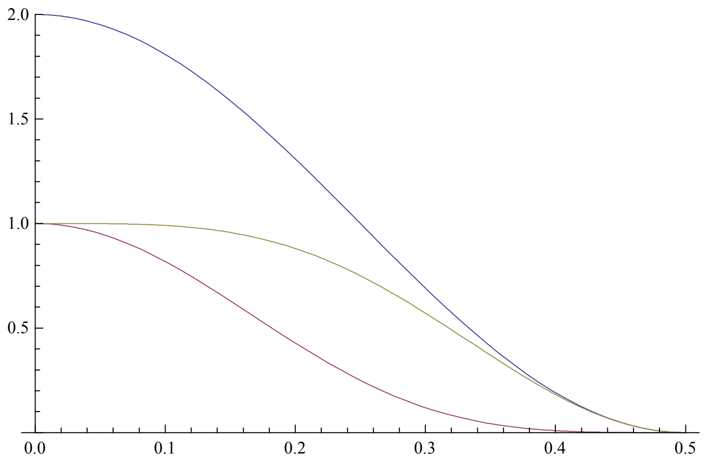
- durch geeignete Kombination des Identitätsoperators $\boldsymbol{I} = (0, 1, 0)$ mit der einfachen Binomialmaske zweiter Ordnung $^{2}\boldsymbol{B} = (1/4, 1/2, 1/4)$
- steilerer Abfall zu höheren Wellenzahlen hin
- Ansatz für einen entsprechenden Filter zweiter Ordnung: $^{(2,1)}\boldsymbol{B} = \boldsymbol{I} - \left(\boldsymbol{I} -^{2}\boldsymbol{B}\right)^2 = \boldsymbol{I} - \boldsymbol{I}^2 + 2\cdot{}^{2}\boldsymbol{B} -^{2}\boldsymbol{B}^2 = 2\cdot{}^{2}\boldsymbol{B} -^{4}\boldsymbol{B}$

- Verallgemeinerter Ansatz: $^{(n,l)}\boldsymbol{B} = \left[\boldsymbol{I} - \left(\boldsymbol{I} -^{2}\boldsymbol{B}\right)^n\right]^l$

- Verallgemeinerter Ansatz : $^{(n,l)}\hat{\boldsymbol{B}} = \left\{1 - \frac{1}{2^n}[1 - \cos(2\pi k)]^n\right\}^l$

- Koeffizienten des Filters zweiter Ordnung: $^{(2,1)}\boldsymbol{B} = 2\cdot{}^{2}\boldsymbol{B} -^{4}\boldsymbol{B}$

```
Plot[Evaluate@{2 transferfunktion[binom[2]], transferfunktion[binom[4]],
   transferfunktion[2 ListConvolve[binom[2], identität[5], 2] - binom[4]]}, {k1, 0, kmax}, PlotRange → {Full, {-0, 2}}]
```



Die Umformung für das obige Beispiel

```
(1 - ((1 - b^2)^2))^1 // Simplify // Expand
```

$2\,b^2 - b^4$

Minuend (“doppelter” Binomialfilter 2. Ordnung)

```
2 ListConvolve[binom[2], identität[5], 2]
```

$\left\{0, \frac{1}{2}, 1, \frac{1}{2}, 0\right\}$

Subtrahend (Binomialfilter 4. Ordnung)

```
binom[4]
```

$\left\{\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right\}$

Ergebnis gemäß $^{(2,1)}\boldsymbol{B} = 2\cdot{}^{2}\boldsymbol{B} - {}^{4}\boldsymbol{B}$

```
2 ListConvolve[binom[2], identität[5], 2] - binom[4]
```

$\left\{-\frac{1}{16}, \frac{1}{4}, \frac{5}{8}, \frac{1}{4}, -\frac{1}{16}\right\}$

Dieselben Koeffizienten aus allgemeinem Ansatz: $^{(2,1)}\boldsymbol{B} = \boldsymbol{I} - \left(\boldsymbol{I} -^{2} \boldsymbol{B}\right)^2$

```
identität[5] - ListConvolve[identität[5] - ListConvolve[binom[2], identität[5], 2], identität[5] - ListConvolve[binom[2], identität[5], 2], 3]
```

$\left\{-\frac{1}{16}, \frac{1}{4}, \frac{5}{8}, \frac{1}{4}, -\frac{1}{16}\right\}$

- Für $n = 1$ folgt $^{(1,1)}\boldsymbol{B} = \boldsymbol{I} - \left(\boldsymbol{I} -^{2} \boldsymbol{B}\right)^1 =^{2} \boldsymbol{B}$

Transferfunktion aus Binomialfilter gebildet

```
transferfunktion[binom[2]] // TrigExpand
```

$\text{Cos}[\pi\, k_1]^2$

Transferfunktion gemäß Bildungsvorschrift (s.o.)

```
(1 - 1 / 2^1 (1 - Cos[2 π k1]) ^1) ^1 // Simplify
```

$\text{Cos}[\pi\, k_1]^2$

Binomialfilter 2. Ordnung (hier mit 5 Stützstellen)

```
identität[5] - ListConvolve[identität[5], identität[5] - ListConvolve[binom[2], identität[5], 2], 3]
```

$\left\{0, \frac{1}{4}, \frac{1}{2}, \frac{1}{4}, 0\right\}$

- die Bildungsvorschrift der Transferfunktion läßt sich für $n = 2$ verifizieren:

Binomialfilter 4. Ordnung aus Faltung des Binomialfilter 2. Ordnung mit sich selbst

```
ListConvolve[ListConvolve[binom[2], identität[5], 2], ListConvolve[binom[2], identität[5], 2], 3]
```

$\left\{\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right\}$

```
binom[4]
```

$\left\{\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right\}$

Transferfunktion ${}^{(2,1)}\hat{\boldsymbol{B}}$ des Glättungsfilters ${}^{(2,1)}\boldsymbol{B}$ gebildet aus Filterkoeffizienten gemäß umgeformter Bildungsvorschrift ${}^{(2,1)}\boldsymbol{B} = 2 \cdot {}^{2}\boldsymbol{B} - {}^{4}\boldsymbol{B}$

```
transferfunktion[2 ListConvolve[binom[2], identität[5], 2] - binom[4]]
```

$-\frac{1}{2}\,\mathrm{Cos}[\pi\, k_1]^2\,(-3 + \mathrm{Cos}[2\,\pi\, k_1])$

Transferfunktion ${}^{(2,1)}\hat{\boldsymbol{B}}$ des Glättungsfilters ${}^{(2,1)}\boldsymbol{B}$ gebildet der gewichteten Differenz der Transferfunktionen ${}^{(2,1)}\hat{\boldsymbol{B}} = 2 \cdot {}^{2}\hat{\boldsymbol{B}} - {}^{4}\hat{\boldsymbol{B}}$

```
2 transferfunktion[binom[2]] - transferfunktion[binom[4]]
```

$2\,\mathrm{Cos}[\pi\, k_1]^2 - \mathrm{Cos}[\pi\, k_1]^4$

```
2 transferfunktion[binom[2]] - transferfunktion[binom[4]] // TrigFactor
```

$-\frac{1}{2}\,\mathrm{Cos}[\pi\, k_1]^2\,(-3 + \mathrm{Cos}[2\,\pi\, k_1])$

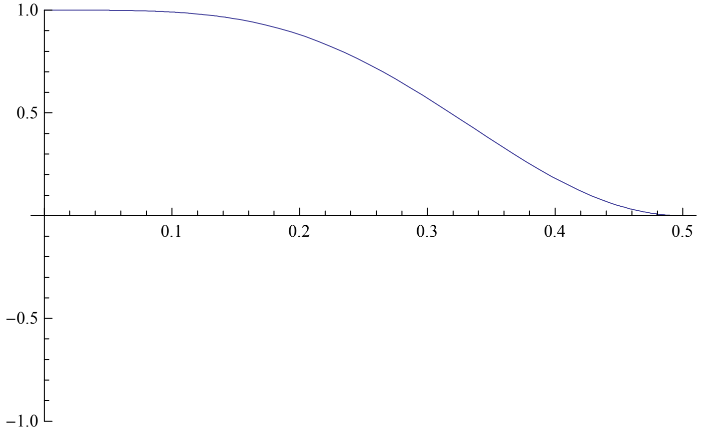
Transferfunktion ${}^{(2,1)}\hat{\boldsymbol{B}}$ des Glättungsfilters ${}^{(2,1)}\boldsymbol{B}$ direkt gebildet aus ${}^{(n,l)}\hat{\boldsymbol{B}} = \left\{1 - \frac{1}{2^n}[1 - \cos(2\pi k)]^n\right\}^l$

```
(1 - 1 / 2^2 (1 - Cos[2 π k1]) ^2) ^1 // TrigFactor
```

$-\frac{1}{2}\,\mathrm{Cos}[\pi\, k_1]^2\,(-3 + \mathrm{Cos}[2\,\pi\, k_1])$

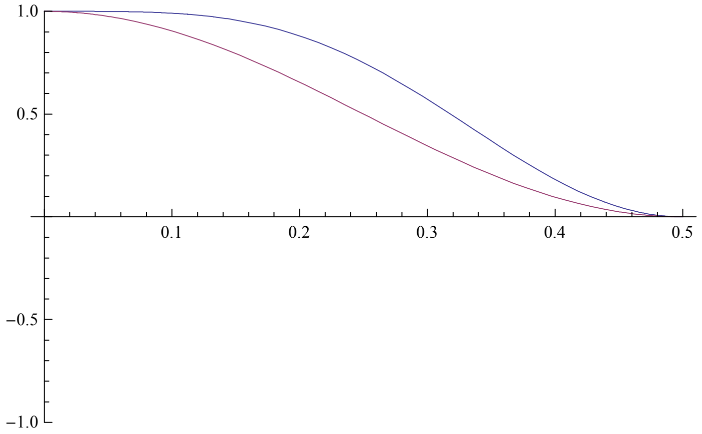
Plot der Transferfunktion ${}^{(2,1)}\hat{\boldsymbol{B}}$ für ${}^{(2,1)}\boldsymbol{B} = \boldsymbol{I} - \left(\boldsymbol{I} -{}^{2}\boldsymbol{B}\right)^2 = 2\cdot{}^{2}\boldsymbol{B} -{}^{4}\boldsymbol{B}$

```
Plot[Evaluate@transferfunktion[2 ListConvolve[binom[2], identität[5], 2] - binom[4]], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```



zum Vergleich ${}^{2}\hat{\boldsymbol{B}}$ des Binomialfilters ${}^{2}\boldsymbol{B}$

```
Plot[Evaluate@{transferfunktion[2 ListConvolve[binom[2], identität[5], 2] - binom[4]], transferfunktion[binom[2]]},
 {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```



- Für $n = 1$ und $l > 1$ ergeben sich einfache Binomialfilter $2\,l$. Ordnung:

Für $n = 1$ und $l = 2$ folgt $^{(1,2)}\boldsymbol{B} = \left[\boldsymbol{I} - \left(\boldsymbol{I} -^{2} \boldsymbol{B}\right)^{1}\right]^{2} =^{2} \boldsymbol{B}^{2} =^{4} \boldsymbol{B}$

```
(1 - ((1 - b^2)^1))^2 // Simplify // Expand
```

b^4

```
ListConvolve[identität[5] - ListConvolve[identität[5], identität[5] - ListConvolve[binom[2], identität[5], 2], 3],
 identität[5] - ListConvolve[identität[5], identität[5] - ListConvolve[binom[2], identität[5], 2], 3], 3]
```

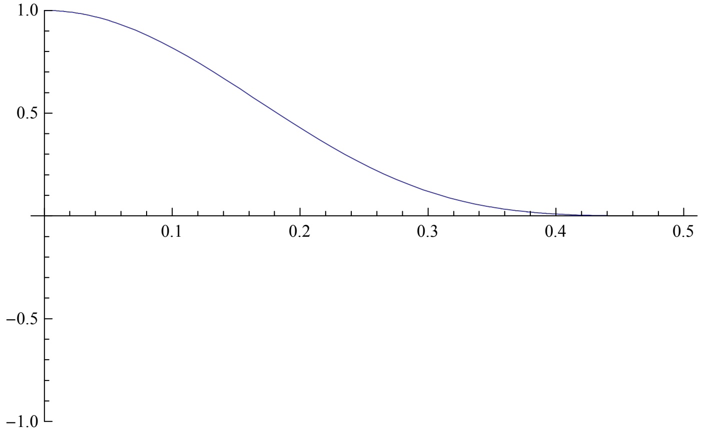
$\left\{\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right\}$

```
binom[4]
```

$\left\{\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right\}$

Transferfunktion von $^{(1,2)}\hat{\boldsymbol{B}} =^{4} \hat{\boldsymbol{B}}$

```
Plot[Evaluate@transferfunktion[binom[4]], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```



- Für $n \geq 1$ und $l \geq 1$ ergeben sich viele Varianten, z.B. mit $n = 2$ und $l = 2$ folgt $^{(2,2)}\boldsymbol{B} = \left[\boldsymbol{I} - \left(\boldsymbol{I} - {}^{2}\boldsymbol{B}\right)^2\right]^2 = \left[2\,{}^{2}\boldsymbol{B} - {}^{4}\boldsymbol{B}\right]^2$

Differenz aus Identitätsoperator und Binomialfilter 2. Ordnung (hier mit 9 Stützstellen)

```
identität[9] - ListConvolve[binom[2], identität[9], 2]
```

$\left\{0, 0, 0, -\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}, 0, 0, 0\right\}$

Differenz aus Identitätsoperator und o.g. quadrierter Differenz aus Identitätsoperator und Binomialfilter 2. Ordnung

```
identität[9] - ListConvolve[identität[9] - ListConvolve[binom[2], identität[9], 2], identität[9] - ListConvolve[binom[2], identität[9], 2], 5]
```

$\left\{0, 0, -\frac{1}{16}, \frac{1}{4}, \frac{5}{8}, \frac{1}{4}, -\frac{1}{16}, 0, 0\right\}$

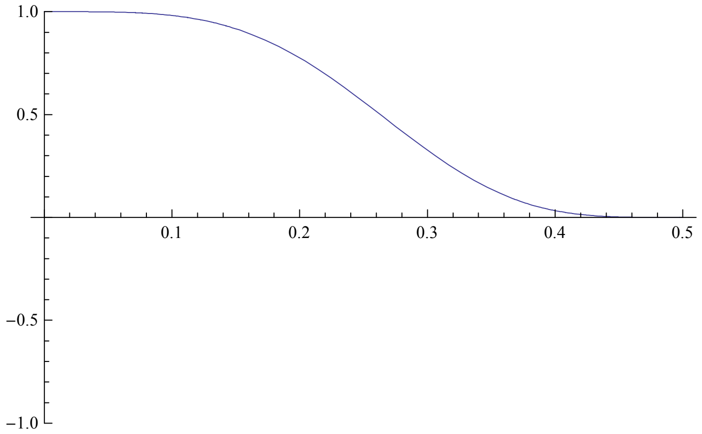
Glättungsfilter $^{(2,2)}\boldsymbol{B} = \left[\boldsymbol{I} - \left(\boldsymbol{I} - {}^{2}\boldsymbol{B}\right)^2\right]^2$ aus Quadrat (Faltung mit sich selbst) vorgenannter Differenz:

```
ListConvolve[identität[9] - ListConvolve[identität[9] - ListConvolve[binom[2], identität[9], 2],
   identität[9] - ListConvolve[binom[2], identität[9], 2], 5], identität[9] -
  ListConvolve[identität[9] - ListConvolve[binom[2], identität[9], 2], identität[9] - ListConvolve[binom[2], identität[9], 2], 5], 5]
```

$\left\{\frac{1}{256}, -\frac{1}{32}, -\frac{1}{64}, \frac{9}{32}, \frac{67}{128}, \frac{9}{32}, -\frac{1}{64}, -\frac{1}{32}, \frac{1}{256}\right\}$

Transferfunktion des Glättungsfilters $^{(2,2)}\hat{B}$

```
Plot[Evaluate@transferfunktion[ListConvolve[identität[9] -
    ListConvolve[identität[9] - ListConvolve[binom[2], identität[9], 2], identität[9] - ListConvolve[binom[2], identität[9], 2], 5],
   identität[9] - ListConvolve[identität[9] - ListConvolve[binom[2], identität[9], 2],
     identität[9] - ListConvolve[binom[2], identität[9], 2], 5], 5]], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```



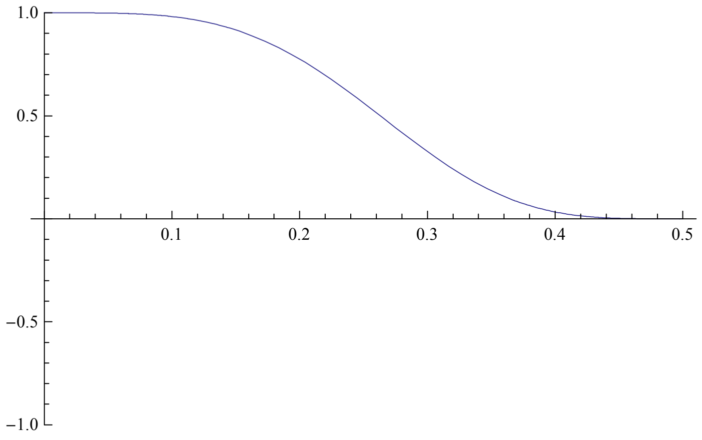

```
transferfunktion[ListConvolve[identität[9] - ListConvolve[
   identität[9] - ListConvolve[binom[2], identität[9], 2], identität[9] - ListConvolve[binom[2], identität[9], 2], 5], identität[9] -
  ListConvolve[identität[9] - ListConvolve[binom[2], identität[9], 2], identität[9] - ListConvolve[binom[2], identität[9], 2], 5], 5]]
```

$$\frac{1}{4} \operatorname{Cos}[\pi k_1]^4 (-3 + \operatorname{Cos}[2 \pi k_1])^2$$

dieselbe Transferfunktion des Glättungsfilters $^{(2,2)}\hat{B}$ gemäß der Bildungsvorschrift

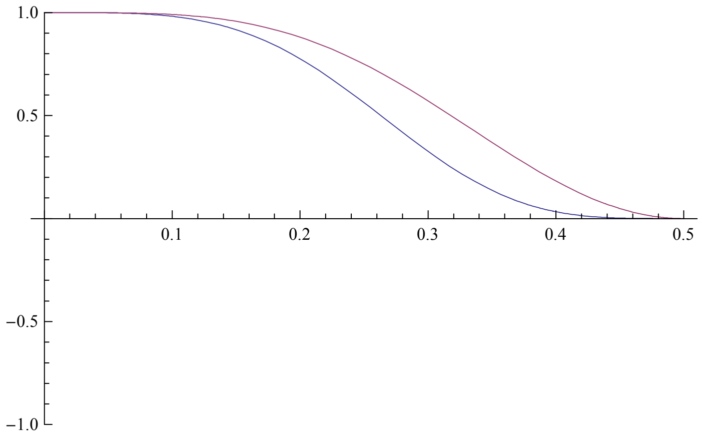
```
n = 2;
l = 2;
(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l // TrigFactor
Plot[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```

$$\frac{1}{4}\,\mathrm{Cos}[\pi\, k_1]^4\,(-3+\mathrm{Cos}[2\,\pi\, k_1])^2$$



zum Vergleich die Transferfunktion des o.g. Glättungsfilters $^{(2,1)}\hat{B}$

```
Plot[Evaluate@{(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, transferfunktion[2 ListConvolve[binom[2], identität[5], 2] - binom[4]]},
 {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```

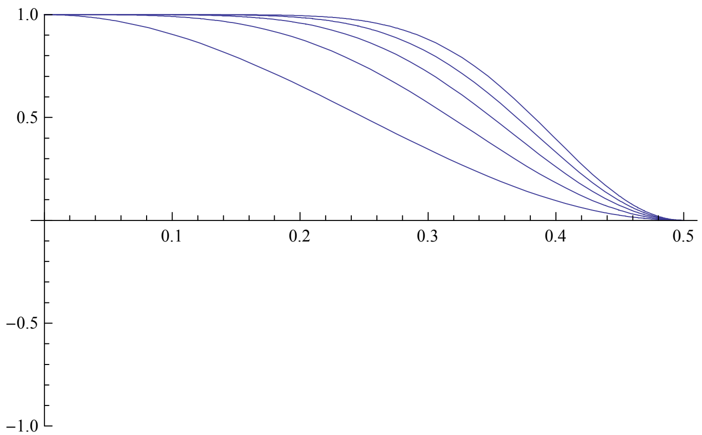


Transferfunktionen $^{(1,1)}\hat{B} = {}^{2}\hat{B}$, $^{(2,1)}\hat{B}$, $^{(3,1)}\hat{B}$, $^{(4,1)}\hat{B}$, $^{(5,1)}\hat{B}$

```
Table[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, {n, 1, 5}, {l, 1, 1}]
```

$$\left\{\left\{1+\frac{1}{2}\left(-1+\mathrm{Cos}[2\pi k_1]\right)\right\},\ \left\{1-\frac{1}{4}\left(1-\mathrm{Cos}[2\pi k_1]\right)^2\right\},\ \left\{1-\frac{1}{8}\left(1-\mathrm{Cos}[2\pi k_1]\right)^3\right\},\ \left\{1-\frac{1}{16}\left(1-\mathrm{Cos}[2\pi k_1]\right)^4\right\},\ \left\{1-\frac{1}{32}\left(1-\mathrm{Cos}[2\pi k_1]\right)^5\right\}\right\}$$

```
Plot[Table[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, {n, 1, 5}, {l, 1, 1}], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```

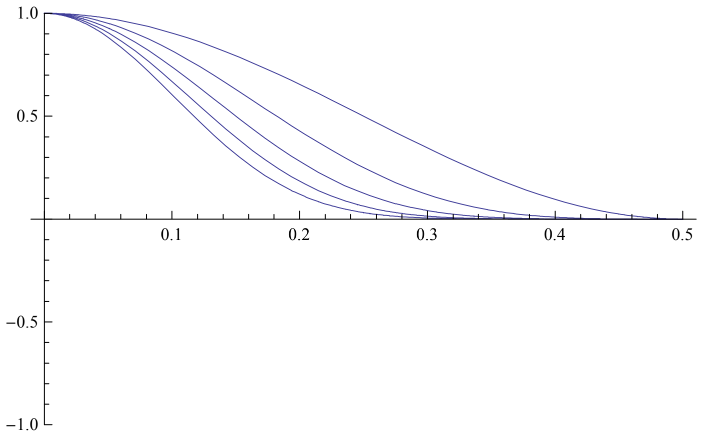


Transferfunktionen $^{(1,1)}\hat{B} = ^{2}\hat{B}$, $^{(1,2)}\hat{B} = ^{4}\hat{B}$, $^{(1,3)}\hat{B} = ^{6}\hat{B}$, $^{(1,4)}\hat{B} = ^{8}\hat{B}$, $^{(1,5)}\hat{B} = ^{10}\hat{B}$

```
Table[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, {n, 1, 1}, {l, 1, 5}]
```

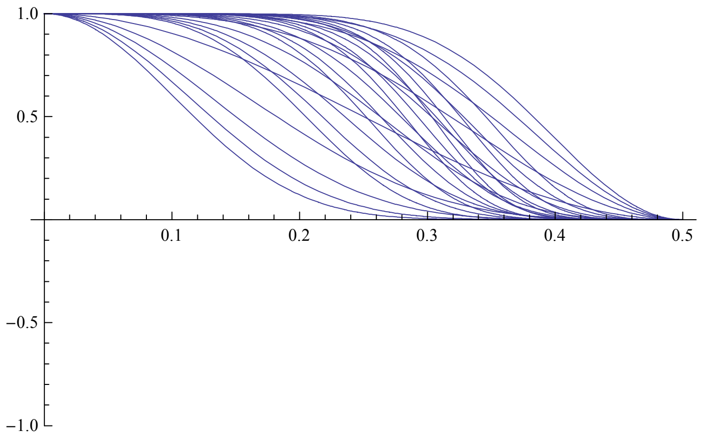
$$\left\{\left\{1+\frac{1}{2}(-1+\text{Cos}[2\pi k_1]),\ \left(1+\frac{1}{2}(-1+\text{Cos}[2\pi k_1])\right)^2,\ \left(1+\frac{1}{2}(-1+\text{Cos}[2\pi k_1])\right)^3,\ \left(1+\frac{1}{2}(-1+\text{Cos}[2\pi k_1])\right)^4,\ \left(1+\frac{1}{2}(-1+\text{Cos}[2\pi k_1])\right)^5\right\}\right\}$$

```
Plot[Table[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, {n, 1, 1}, {l, 1, 5}], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```



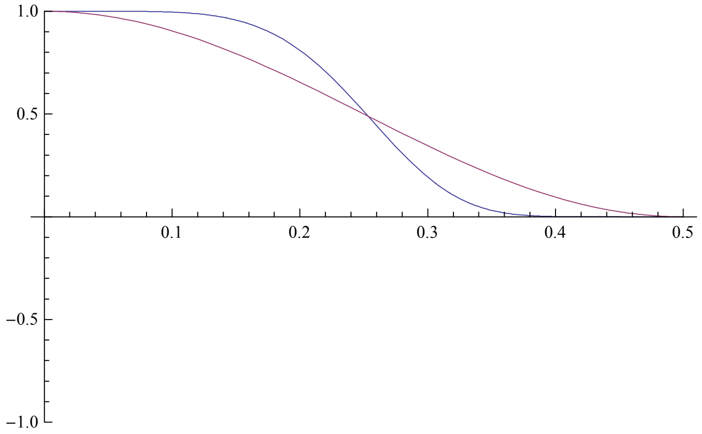
Transferfunktionen $^{(1,1)}\hat{B} = ^{2}\hat{B}$ bis $^{(5,5)}\hat{B}$

```
Plot[Table[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, {n, 1, 5}, {l, 1, 5}], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```



Transferfunktionen ${}^{(1,1)}\hat{B} = {}^{2}\hat{B}$ und ${}^{(3,5)}\hat{B}$

```
Plot[{(1 - 1 / 2^3 (1 - Cos[2 π k1]) ^3) ^5, (1 - 1 / 2^1 (1 - Cos[2 π k1]) ^1) ^1}, {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```



extreme Transferfunktionen ${}^{(50,1)}\hat{B}$, ${}^{(100,1)}\hat{B}$, ${}^{(150,1)}\hat{B}$, ${}^{(200,1)}\hat{B}$, ${}^{(250,1)}\hat{B}$

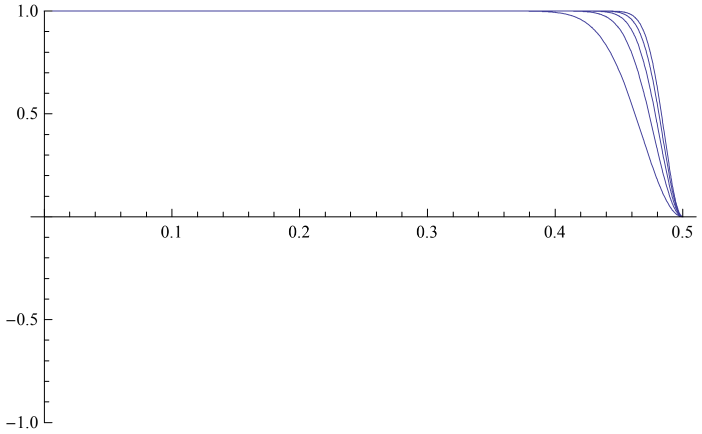
```
Table[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^1, {n, 50, 250, 50}, {1, 1, 1}]
```

$$\left\{\left\{1-\frac{(1-\mathrm{Cos}[2\,\pi\,k_1])^{50}}{1\,125\,899\,906\,842\,624}\right\},\left\{1-\frac{(1-\mathrm{Cos}[2\,\pi\,k_1])^{100}}{1\,267\,650\,600\,228\,229\,401\,496\,703\,205\,376}\right\},\right.$$

$$\left\{1-\frac{(1-\mathrm{Cos}[2\,\pi\,k_1])^{150}}{1\,427\,247\,692\,705\,959\,881\,058\,285\,969\,449\,495\,136\,382\,746\,624}\right\},\left\{1-\frac{(1-\mathrm{Cos}[2\,\pi\,k_1])^{200}}{1\,606\,938\,044\,258\,990\,275\,541\,962\,092\,341\,162\,602\,522\,202\,993\,782\,792\,835\,301\,376}\right\},$$

$$\left.\left\{1-\frac{(1-\mathrm{Cos}[2\,\pi\,k_1])^{250}}{1\,809\,251\,394\,333\,065\,553\,493\,296\,640\,760\,748\,560\,207\,343\,510\,400\,633\,813\,116\,524\,750\,123\,642\,650\,624}\right\}\right\}$$

```
Plot[Table[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, {n, 50, 250, 50}, {l, 1, 1}], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```

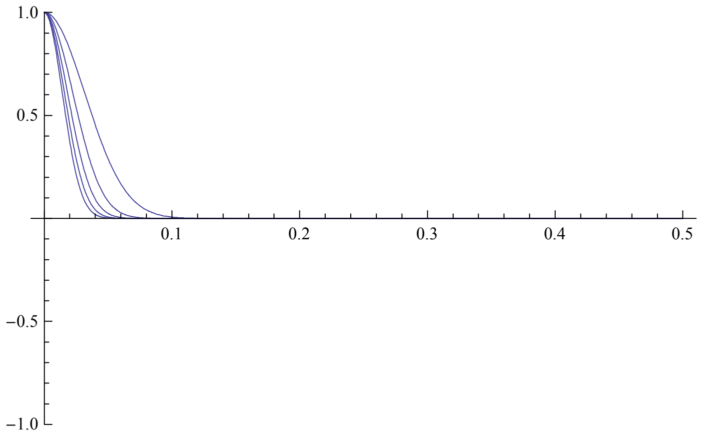


extreme Transferfunktionen $^{(1,50)}\hat{B} = {}^{100}\hat{B}$, $^{(1,100)}\hat{B} = {}^{200}\hat{B}$, $^{(1,150)}\hat{B} = {}^{300}\hat{B}$, $^{(1,200)}\hat{B} = {}^{400}\hat{B}$, $^{(1,250)}\hat{B} = {}^{500}\hat{B}$

```
Table[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, {n, 1, 1}, {l, 50, 250, 50}]
```

$$\left\{\left\{\left(1+\frac{1}{2}\left(-1+\mathrm{Cos}[2\,\pi\,k_1]\right)\right)^{50},\ \left(1+\frac{1}{2}\left(-1+\mathrm{Cos}[2\,\pi\,k_1]\right)\right)^{100},\ \left(1+\frac{1}{2}\left(-1+\mathrm{Cos}[2\,\pi\,k_1]\right)\right)^{150},\ \left(1+\frac{1}{2}\left(-1+\mathrm{Cos}[2\,\pi\,k_1]\right)\right)^{200},\ \left(1+\frac{1}{2}\left(-1+\mathrm{Cos}[2\,\pi\,k_1]\right)\right)^{250}\right\}\right\}$$

```
Plot[Table[(1 - 1 / 2^n (1 - Cos[2 π k1]) ^n) ^l, {n, 1, 1}, {l, 50, 250, 50}], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}]
```



14. Nichtlineare Filter

Medianfilter

Nutzen: Beseitigung singulärer Bildfehler oder störender kleiner Details

Pixelumgebung durch Strukturelement definiert

“Rangordnungsfilter”: immer das in der Sortierreihenfolge mittlere Element aus Pixelumgebung wird dem Pixel neu zugewiesen

```
Clear[boxmaske];
boxmaske[r_] := Table[1, {2 r + 1}, {2 r + 1}];

MatrixForm[boxmaske[10]]
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
Clear[kreismaske];
kreismaske[r_] := Ceiling[Rescale[Sign[Table[(x^2 + y^2), {x, -r, r}, {y, -r, r}] - r * (r + 1 / 2)], {1, -1}]];
```

```
MatrixForm[kreismaske[10]]
```

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

Medianfilter im Test: Grauwertbild mit boxförmiger Maske

```
SeedRandom[2405];
in = Developer`ToPackedArray[ArrayPad[RandomInteger[{0, 255}, {5, 7}], {1, 1}]];
out = ArrayPad[MedianFilter[ArrayPad[in, 1], 1], -1];
{MatrixForm[in], MatrixForm[BoxMatrix[1]], MatrixForm[out]}
```

0	0	0	0	0	0	0	0	0
0	201	57	106	243	47	8	5	0
0	72	136	155	251	195	218	244	0
0	110	196	248	99	132	237	98	0
0	20	191	118	6	68	16	243	0
0	238	169	98	169	16	112	112	0
0	0	0	0	0	0	0	0	0

1	1	1
1	1	1
1	1	1

0	0	0	0	0	0	0	0	0
0	0	72	106	106	47	8	0	0
0	72	136	155	155	195	132	8	0
0	72	136	155	132	132	195	98	0
0	110	169	169	99	99	112	98	0
0	0	98	98	16	16	16	0	0
0	0	0	0	0	0	0	0	0

```
Clear[MyMedianFilter];
MyMedianFilter[im_, el_] := Module[{flatelpos},
   flatelpos = Flatten[Position[Flatten[el], 1]];
   Developer`PartitionMap[(Median[Flatten[#, 1][[flatelpos]]]) &, im, Dimensions[el], {1, 1}, Ceiling[Dimensions[el] / 2]]
  ];
```

Medianfilter im Test: Grauwertbild kreisförmiger Maske

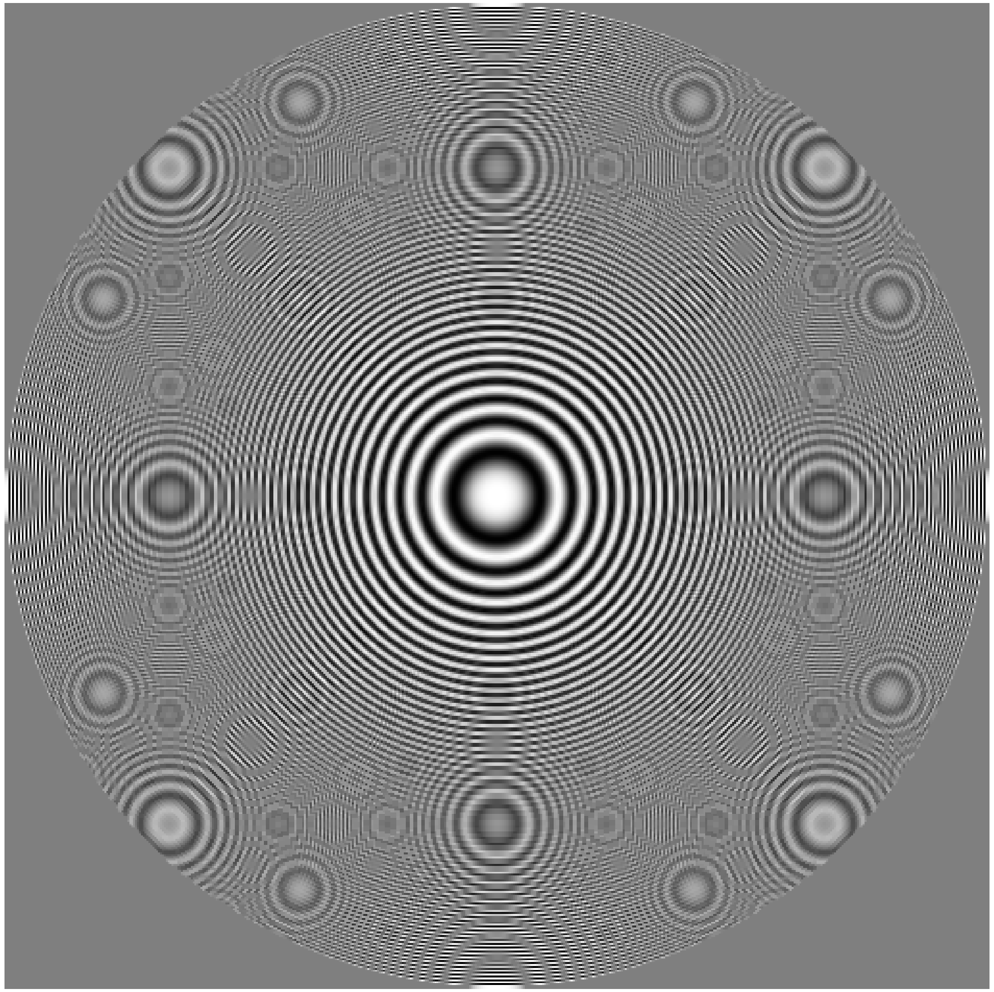
```
SeedRandom[2405];
in = Developer`ToPackedArray[ArrayPad[RandomInteger[{0, 255}, {5, 7}], {1, 1}]];
out = MyMedianFilter[in, kreismaske[1]];
{MatrixForm[in], MatrixForm[kreismaske[1]], MatrixForm[out]}
```

$$\left\{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 201 & 57 & 106 & 243 & 47 & 8 & 5 & 0 \\ 0 & 72 & 136 & 155 & 251 & 195 & 218 & 244 & 0 \\ 0 & 110 & 196 & 248 & 99 & 132 & 237 & 98 & 0 \\ 0 & 20 & 191 & 118 & 6 & 68 & 16 & 243 & 0 \\ 0 & 238 & 169 & 98 & 169 & 16 & 112 & 112 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 57 & 106 & 106 & 106 & 47 & 8 & 5 & 0 \\ 0 & 110 & 136 & 155 & 195 & 195 & 218 & 98 & 0 \\ 0 & 72 & 191 & 155 & 132 & 132 & 132 & 237 & 0 \\ 0 & 110 & 169 & 118 & 99 & 16 & 112 & 98 & 0 \\ 0 & 20 & 169 & 118 & 16 & 68 & 16 & 112 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right\}$$

Medianfilter: Boxmaske

```
Manipulate[Show[Image[Partition[Map[Median[Flatten[#]] &, Flatten[Partition[#, {größe, größe}, {1, 1},
        {{(größe + 1) / 2, (größe + 1) / 2}, {(größe + 1) / 2, (größe + 1) / 2}}], 1]], Dimensions[#][[2]]], ImageSize → Dimensions[#]]],
  {{größe, 3, "Medianfenster"}, 1, 9, 2, Appearance → "Labeled"}, ContinuousAction → False, SaveDefinitions → True] &[ImageData[wellenbild]]
```

Medianfenster 3



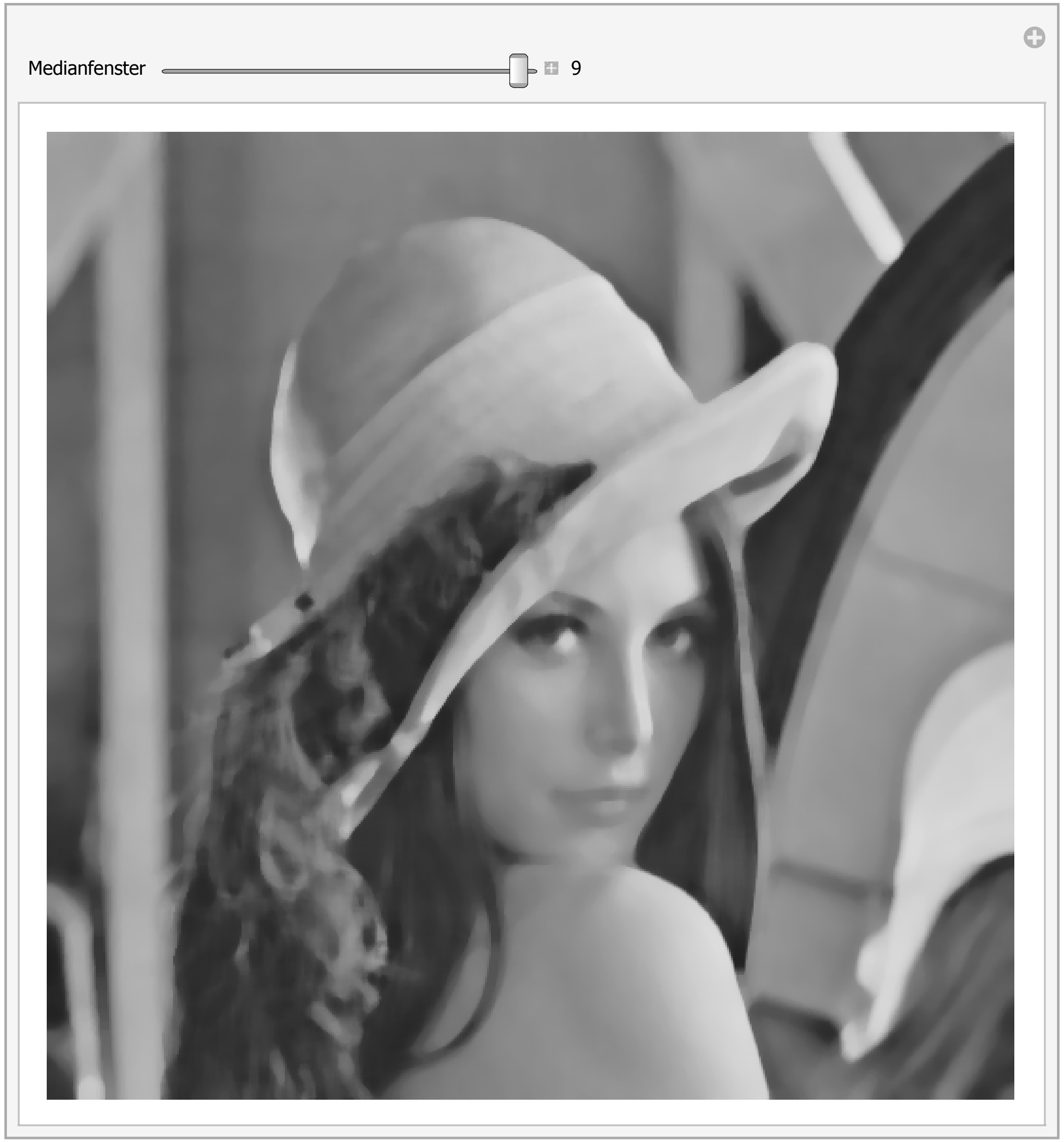
Medianfilter mit quadratischem Strukturelement, eigene Lösung 1

```
Manipulate[Show[Image[Partition[Map[Median[Flatten[#]] &, Flatten[Partition[#, {größe, größe}, {1, 1},
          {{(größe + 1) / 2, (größe + 1) / 2}, {(größe + 1) / 2, (größe + 1) / 2}}], 1]], Dimensions[#][[2]]], ImageSize → Dimensions[#]]],
    {{größe, 3, "Medianfenster"}, 1, 9, 2, Appearance → "Labeled"}, ContinuousAction → False, SaveDefinitions → True] &[
  Map[{0.299, 0.587, 0.114}.# &, ImageData[ExampleData[{"TestImage", "Lena"}]], {2}]]
```

Medianfenster 3

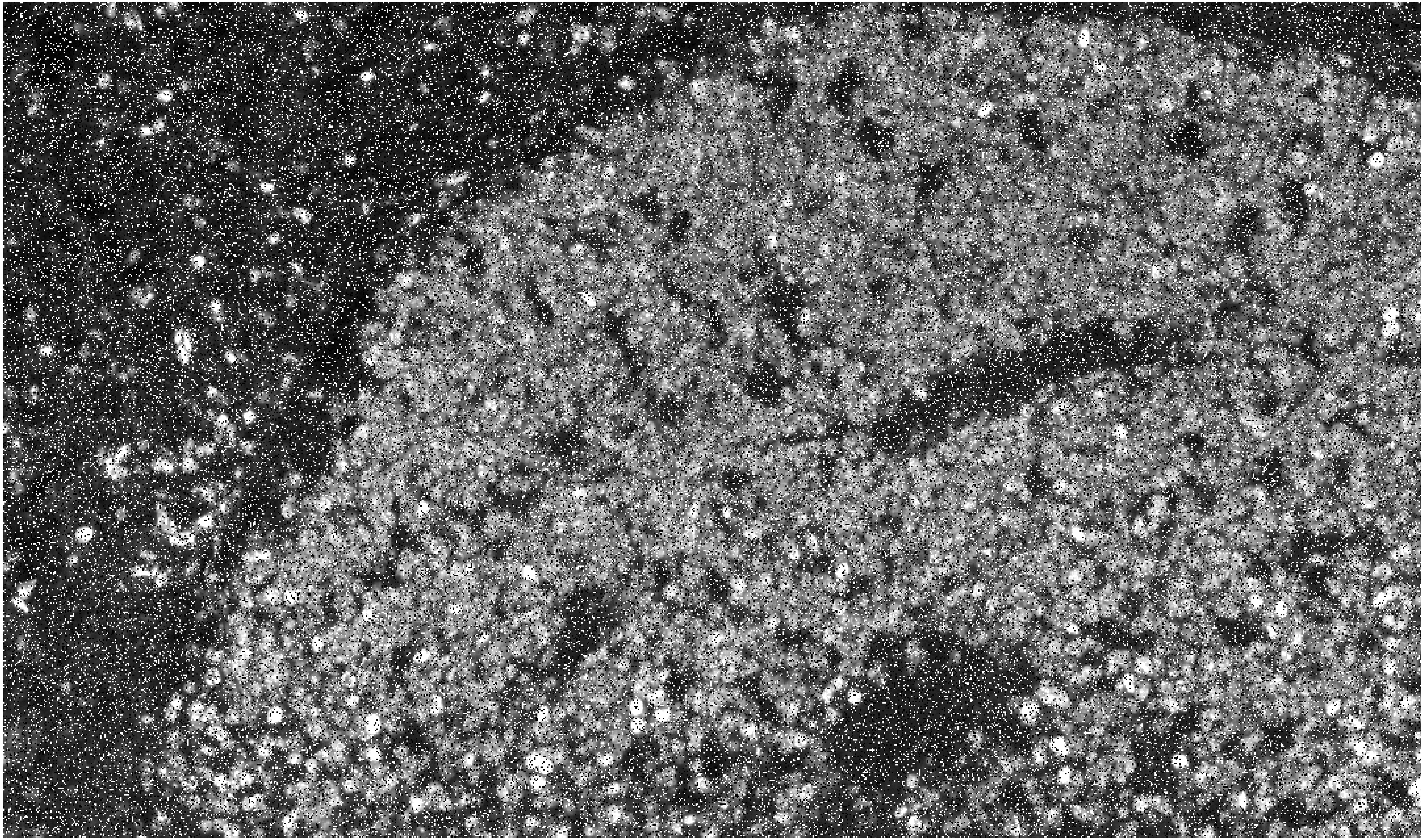
Medianfilter mit quadratischem Strukturelement, eingebaute Lösung

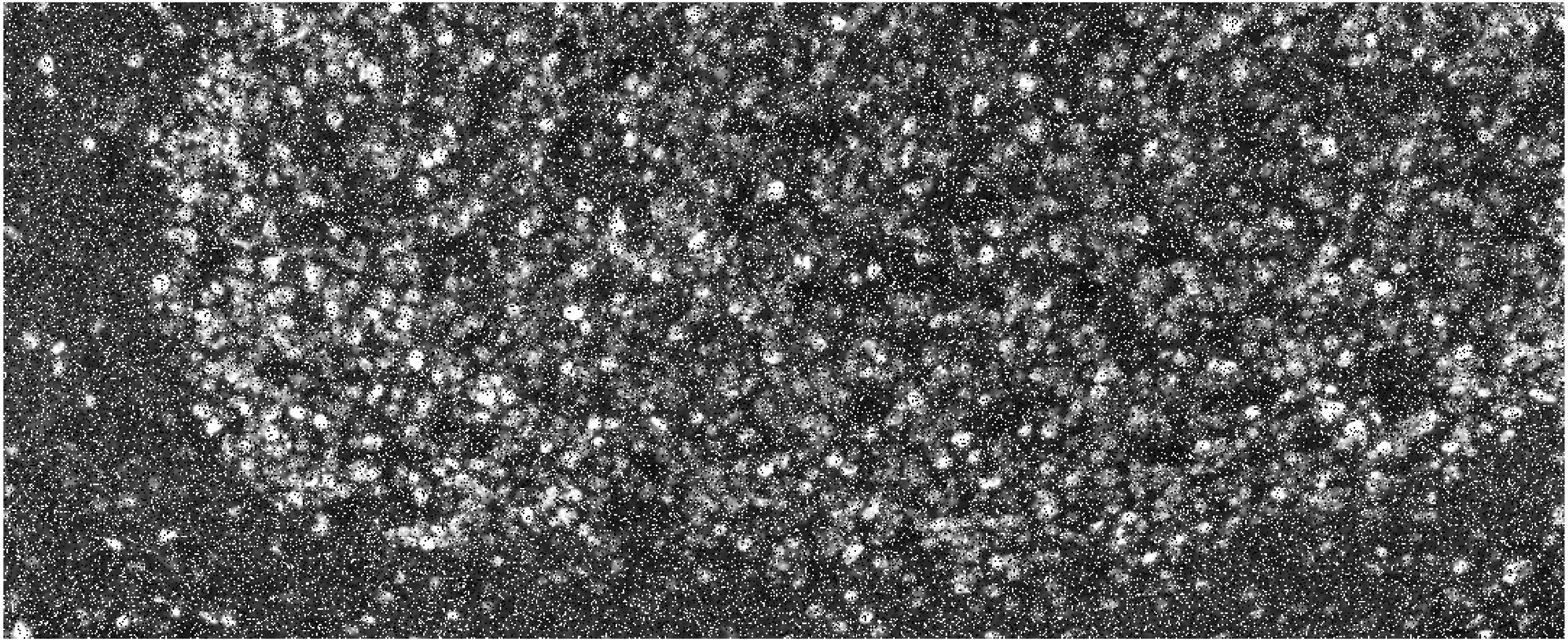
```
Manipulate[Show[MedianFilter[#, (größe - 1) / 2]], {{größe, 3, "Medianfenster"}, 1, 9, 2, Appearance → "Labeled"},
  ContinuousAction → False, SaveDefinitions → True] &[ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Lena"}]]]
```



```
SeedRandom[2405];
defektesbild =
  Image[Map[If[RandomReal[] < 0.2, If[RandomReal[] ≤ 0.5, 0, 1], #] &, ImageData[First@ColorSeparate[Ton3CD21dreikanalausgleichF2]], {2}]];
```

```
Show[defektesbild, ImageSize → ImageDimensions@defektesbild]
```





Mit linearem Glättungsfilter kriegt man diese Art von Rauschen nicht weg:

```
Manipulate[Show[ImageConvolve[defektesbild, Transpose[{binom[ordnung]}].{binom[ordnung]}], ImageSize → ImageDimensions@defektesbild],
 {{ordnung, 2, "Binom-Ordnung"}, 0, 30, 2, Appearance → "Labeled"}, ContinuousAction → False, SaveDefinitions → True]
```

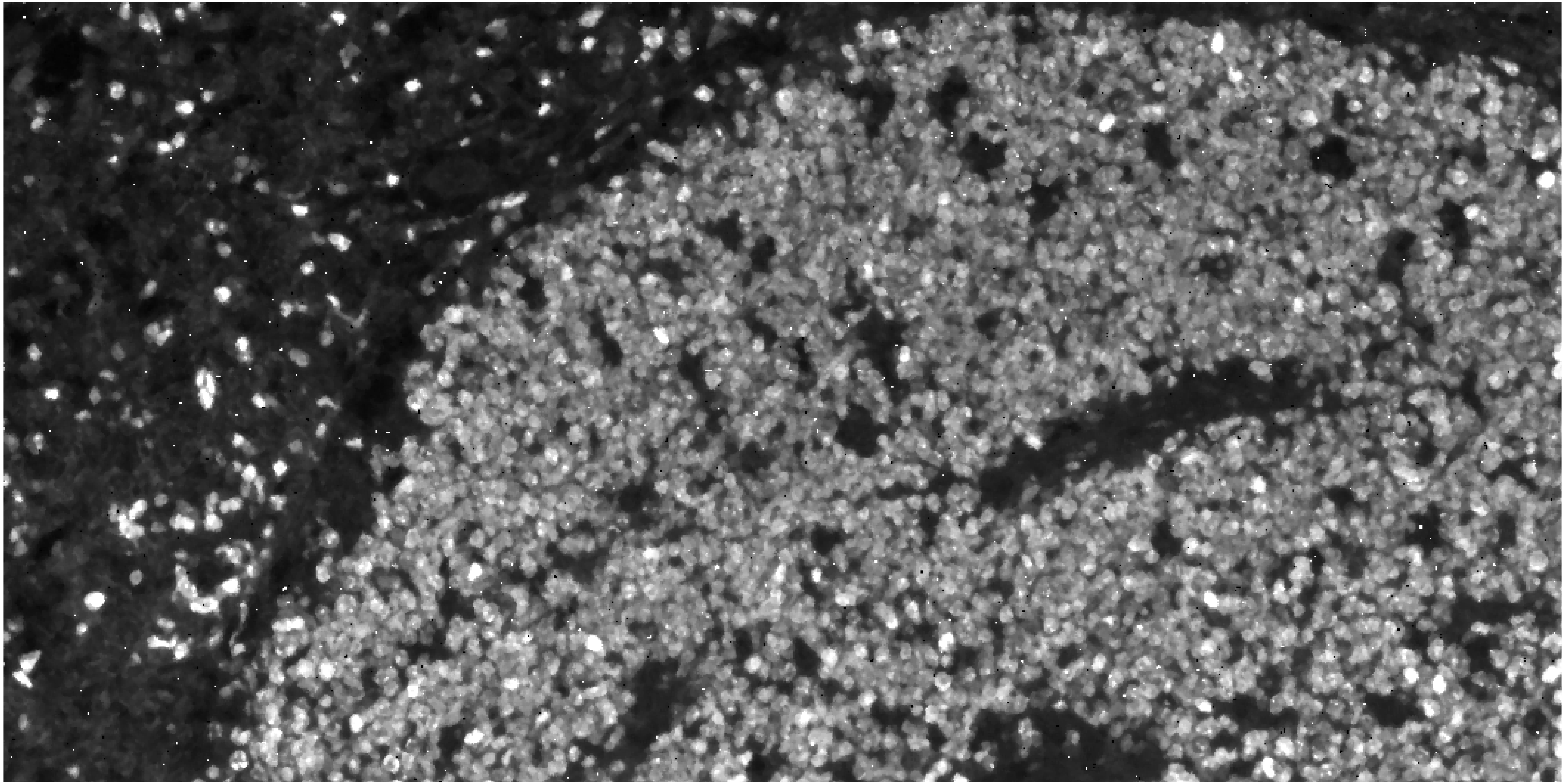
$Aborted

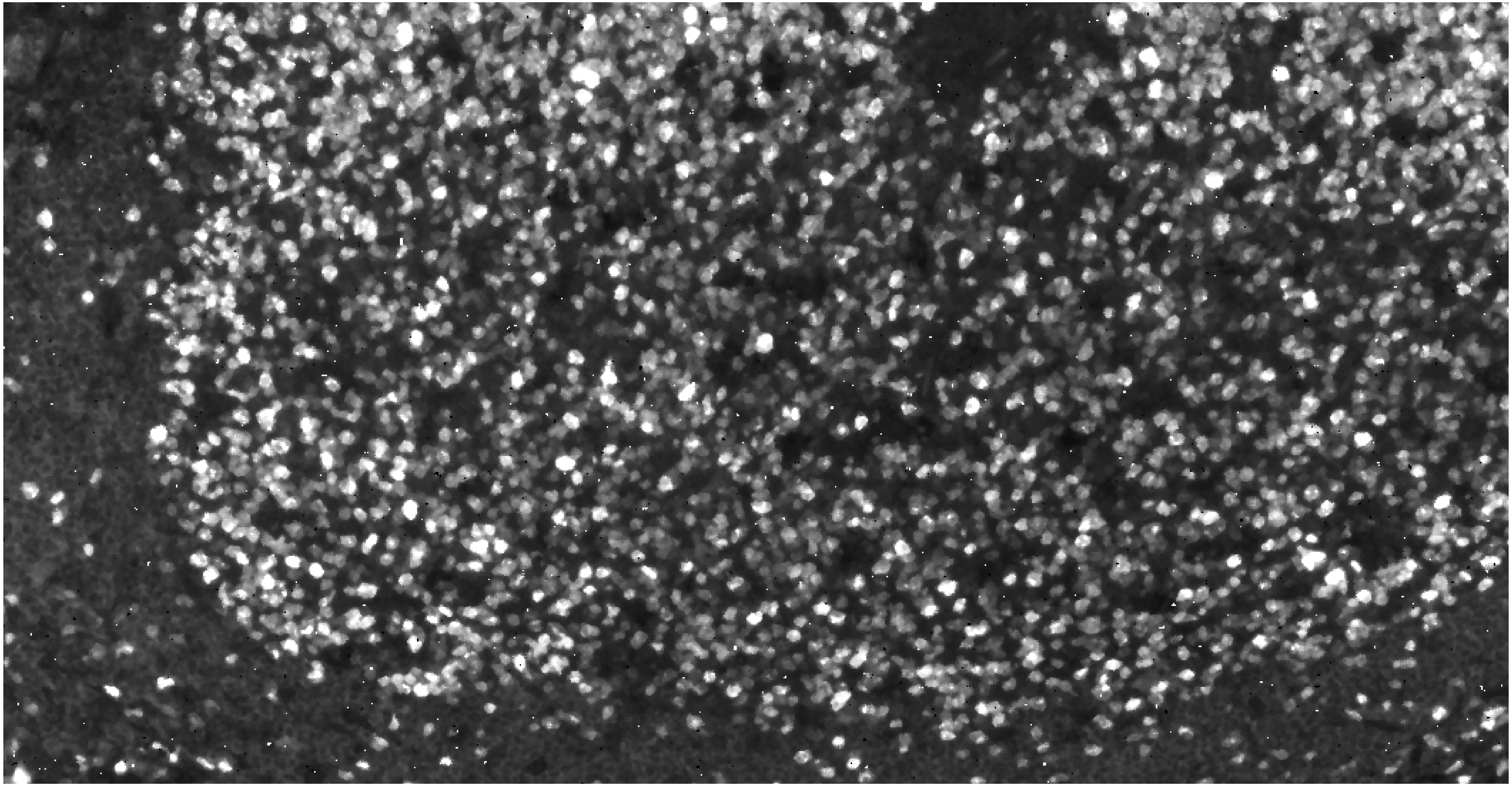
Binom–Ordnung 2

Medianfilter mit quadratischem Strukturelement, eigene Lösung 1 wie oben

```
Manipulate[Show[Image[Partition[Map[Median[Flatten[#]] &,
      Flatten[Partition[#, {größe, größe}, {1, 1}, {{(größe + 1) / 2, (größe + 1) / 2}, {(größe + 1) / 2, (größe + 1) / 2}}], 1]],
     Dimensions[#][[2]]], ImageSize → Reverse@Dimensions[#]]], {{größe, 3, "Medianfenster"}, 1, 9, 2,
   Appearance → "Labeled"}, ContinuousAction → False, SaveDefinitions → True] &[ImageData[defektesbild]]
```

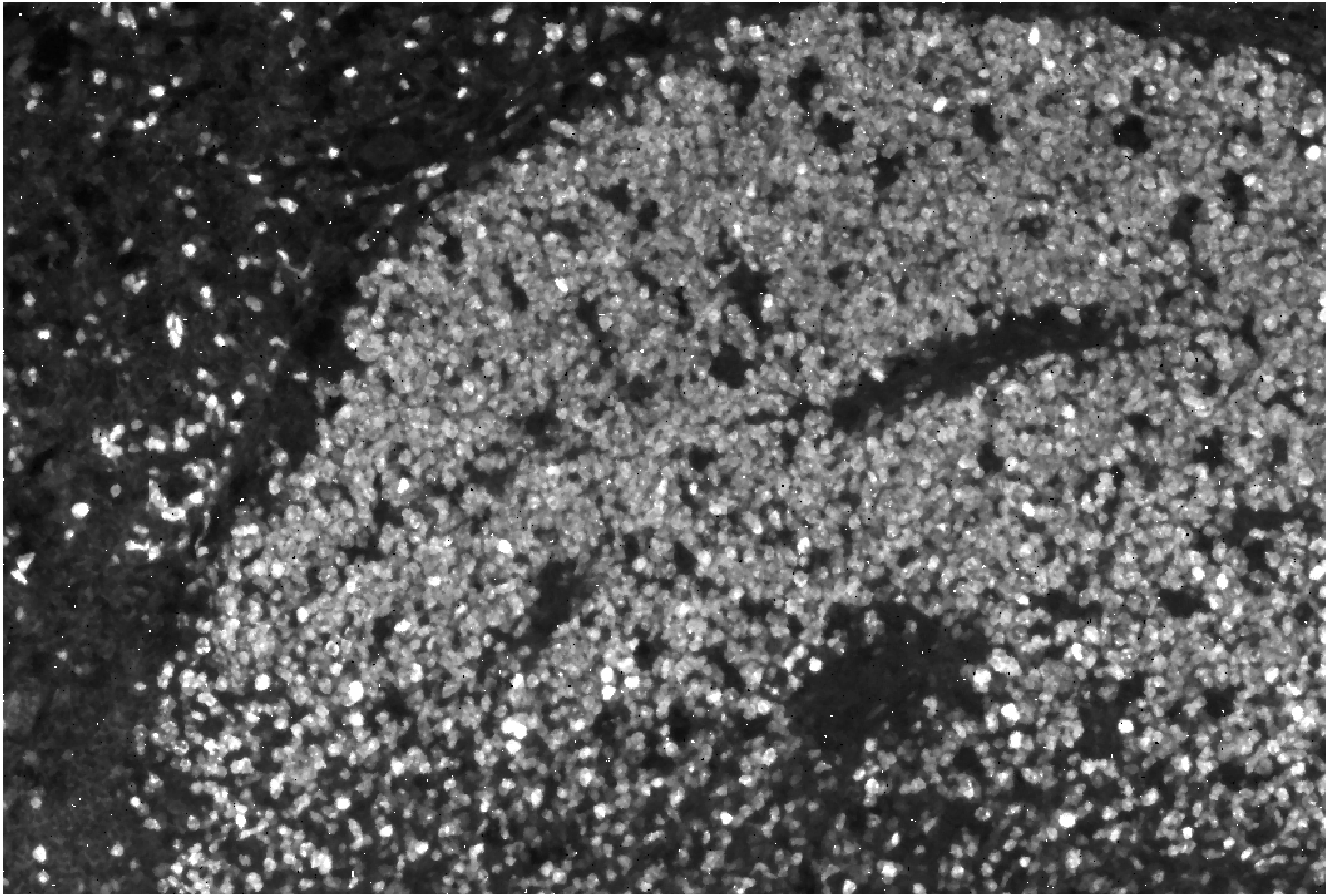
Medianfenster 3

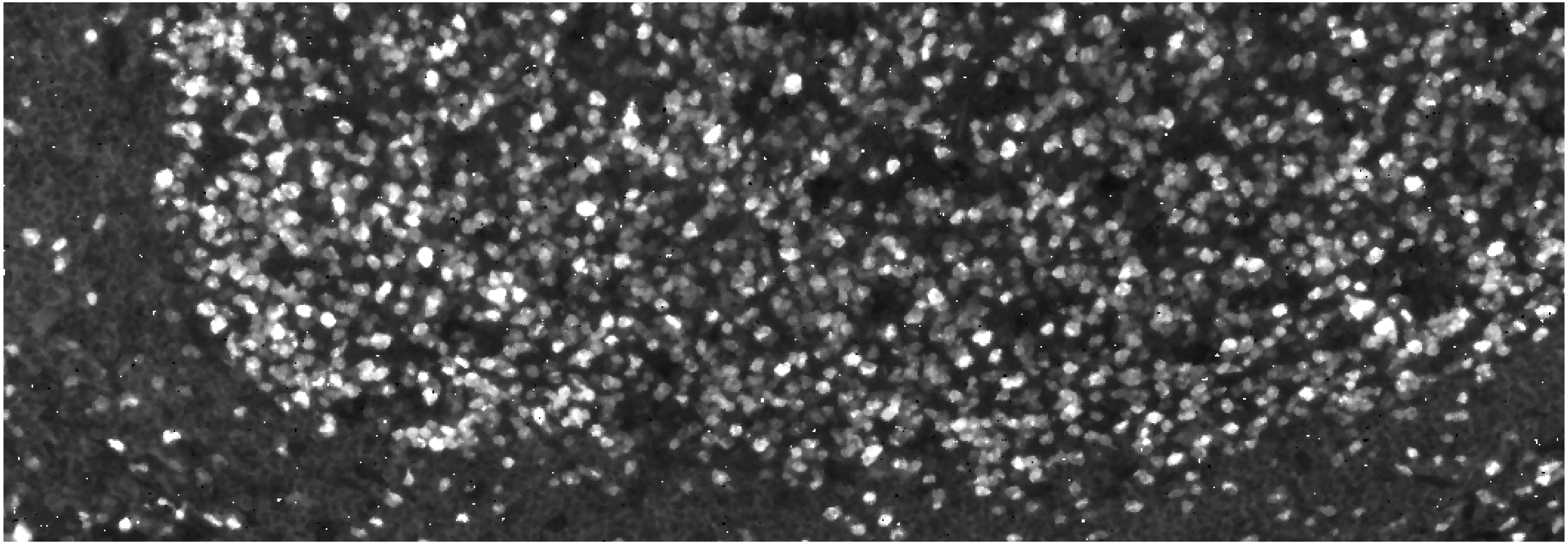




Medianfilter mit quadratischem Strukturelement, eingebaute Lösung wie oben

```
Manipulate[Show[MedianFilter[#, (größe - 1) / 2]],
  {{größe, 3, "Medianfenster"}, 1, 9, 2, Appearance → "Labeled"}, ContinuousAction → False, SaveDefinitions → True] &[defektesbild]
```





Medianfilter mit quadratischem (links) und kreisförmigem (rechts) Strukturelement , eigene Lösung 2

```
Manipulate[
  ControlActive[größe, Grid[{{Show[Image[MyMedianFilter[#1, boxmaske[(größe - 1) / 2]], "Byte"], ImageSize → Reverse@Dimensions[#1]],
     Show[Image[MyMedianFilter[#1, kreismaske[(größe - 1) / 2]], "Byte"], ImageSize → Reverse@Dimensions[#1]]}}]],
  {{größe, 3, "Medianfenster"}, 1, 9, 2, Appearance → "Labeled"}, ContinuousAction → False,
  SaveDefinitions → True] &[ImageData[wellenbild, "Byte"]]
```

Medianfenster 3

$Aborted

Maximumfilter

“Rangordnungsfilter”: immer das in der aufsteigenden Sortierreihenfolge letzte Element aus Pixelumgebung wird dem Pixel neu zugewiesen

```
Clear[GrayDilate];
(*GrayDilate[im_,el_]:=ListConvolve[el,im,{Ceiling[Dimensions[el]/2]},im,Times,Max];*)
GrayDilate = Compile[{{im, _Integer, 2}, {el, _Integer, 2}}, ListConvolve[el, im, {Ceiling[Dimensions[el] / 2]}, im, Times, Max],
   CompilationTarget → "WVM", RuntimeAttributes → {Listable}, Parallelization → True];

SeedRandom[2405];
in = Developer`ToPackedArray[ArrayPad[RandomInteger[{0, 255}, {5, 7}], {1, 1}]];
out = GrayDilate[in, kreismaske[1]];
{MatrixForm[in], MatrixForm[kreismaske[1]], MatrixForm[out]}
```

0	0	0	0	0	0	0	0	0
0	201	57	106	243	47	8	5	0
0	72	136	155	251	195	218	244	0
0	110	196	248	99	132	237	98	0
0	20	191	118	6	68	16	243	0
0	238	169	98	169	16	112	112	0
0	0	0	0	0	0	0	0	0

0	1	0
1	1	1
0	1	0

0	201	57	106	243	47	8	5	0
201	201	201	243	251	243	218	244	5
72	201	196	251	251	251	244	244	244
110	196	248	248	251	237	237	244	98
20	238	196	248	169	132	243	243	243
238	238	238	169	169	169	112	243	112
0	238	169	98	169	16	112	112	0

Maximumfilter mit kreisförmigem Strukturelement

```
Manipulate[ControlActive[größe, Show[Image[GrayDilate[#1, kreismaske[(größe - 1) / 2]], "Byte"], ImageSize → Reverse@Dimensions[#1]]],
   {{größe, 3, "Maximumfenster"}, 1, 9, 2, Appearance → "Labeled"}, ContinuousAction → False, SaveDefinitions → True] &[
 ImageData[First@ColorSeparate[Ton3CD21dreikanalausgleichF2], "Byte"]]
```

Minimumfilter

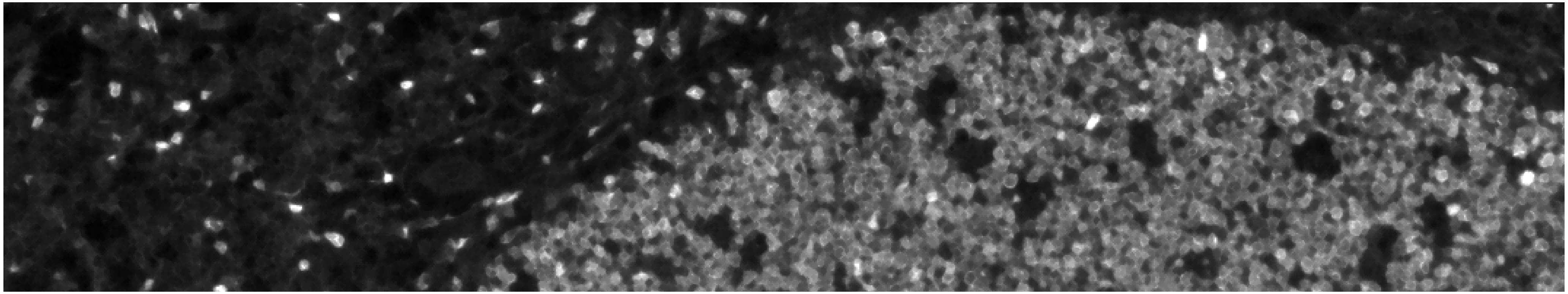
“Rangordnungsfilter”: immer das in der aufsteigenden Sortierreihenfolge erste Element aus Pixelumgebung wird dem Pixel neu zugewiesen

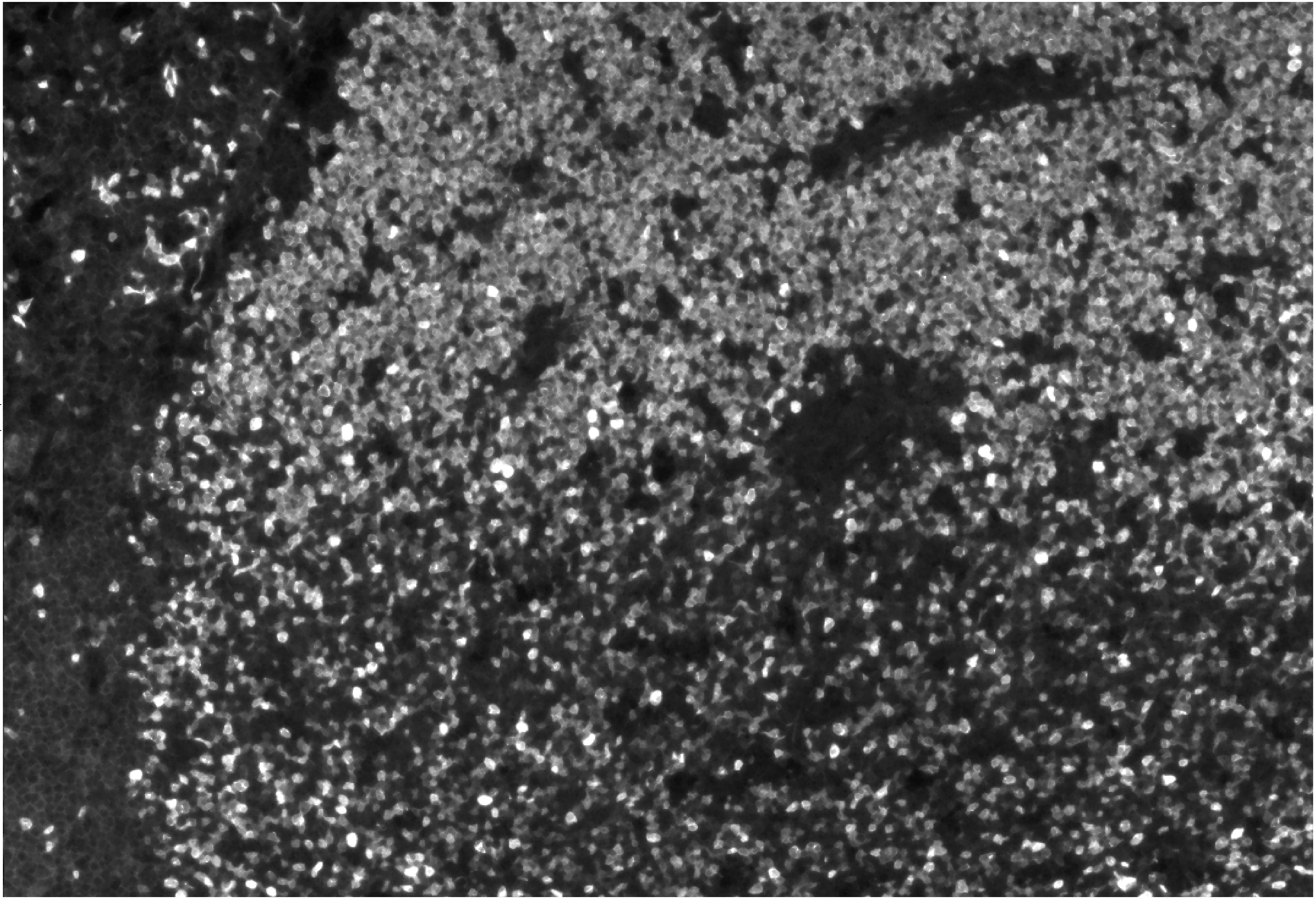
```
Clear[GrayErode];
(*GrayErode[im_,el_]:=
  Module[{implus1=im+1},ListConvolve[el,implus1,{Ceiling[Dimensions[el]/2]},implus1,Times,Min[Cases[{##},Except[0]]]&]-1];*)
(*GrayErode[im_,el_]:=Module[{negim=255-im},255-ListConvolve[el,negim,{Ceiling[Dimensions[el]/2]},negim,Times,Max]];*)
GrayErode = Compile[{{im, _Integer, 2}, {el, _Integer, 2}},
   Module[{negim = 255 - im}, 255 - ListConvolve[el, negim, {Ceiling[Dimensions[el] / 2]}, negim, Times, Max]],
   CompilationTarget → "WVM", RuntimeAttributes → {Listable}, Parallelization → True];
```

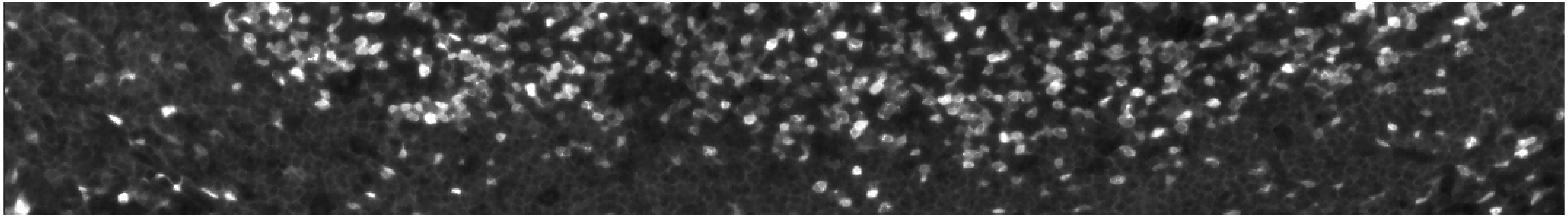
```
SeedRandom[2405];
in = Developer`ToPackedArray[ArrayPad[RandomInteger[{0, 255}, {5, 7}], {1, 1}]];
out = GrayErode[in, kreismaske[1]];
{MatrixForm[in], MatrixForm[kreismaske[1]], MatrixForm[out]}
```

$$\left\{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 201 & 57 & 106 & 243 & 47 & 8 & 5 & 0 \\ 0 & 72 & 136 & 155 & 251 & 195 & 218 & 244 & 0 \\ 0 & 110 & 196 & 248 & 99 & 132 & 237 & 98 & 0 \\ 0 & 20 & 191 & 118 & 6 & 68 & 16 & 243 & 0 \\ 0 & 238 & 169 & 98 & 169 & 16 & 112 & 112 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 57 & 106 & 99 & 47 & 8 & 0 & 0 \\ 0 & 0 & 110 & 99 & 6 & 68 & 16 & 0 & 0 \\ 0 & 0 & 20 & 6 & 6 & 6 & 16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right\}$$

```
Map[Show[Image[GrayErode[ImageData[First@ColorSeparate[Ton3CD21dreikanalausgleichF2], "Byte"], #], "Byte"],
   ImageSize → ImageDimensions@Ton3CD21dreikanalausgleichF2] &, {kreismaske[(3 - 1) / 2], kreismaske[(5 - 1) / 2]}]
```

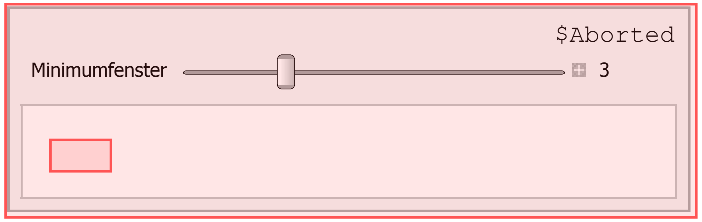






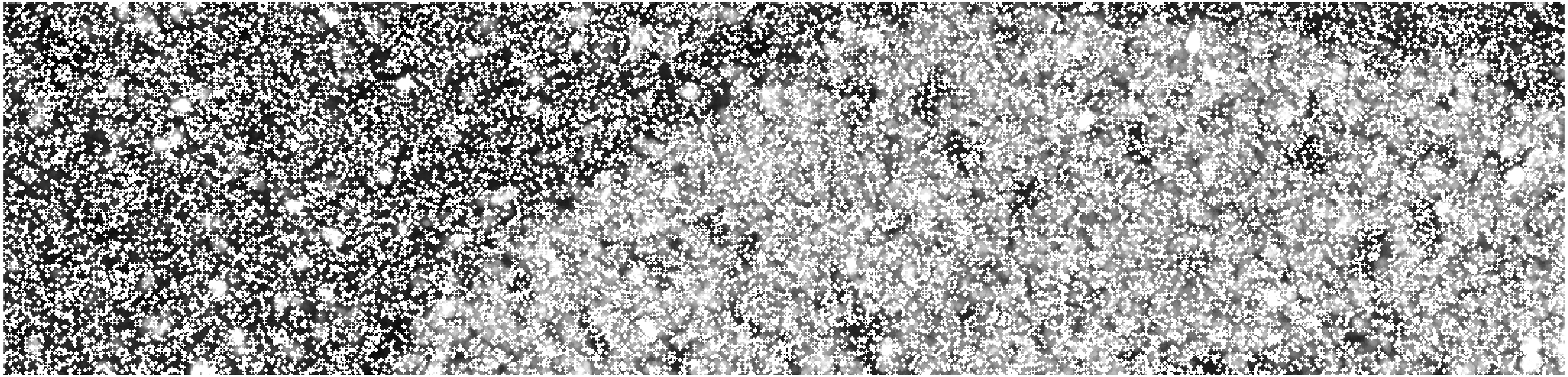
Minimumfilter mit kreisförmigem Strukturelement

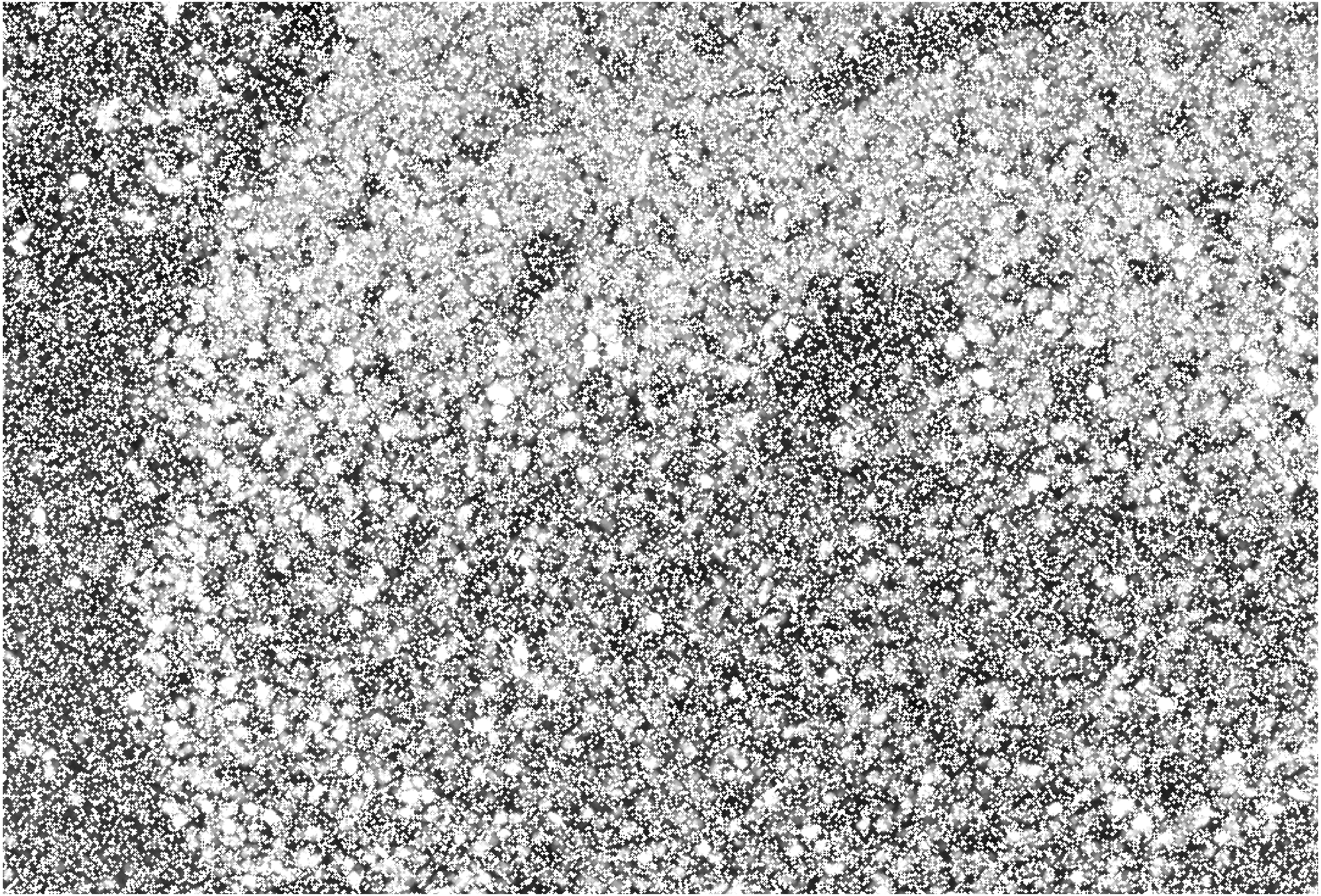
```
Manipulate[ControlActive[größe, Show[Image[GrayErode[#1, kreismaske[(größe - 1) / 2]], "Byte"], ImageSize → Reverse@Dimensions[#1]]],
  {{größe, 3, "Minimumfenster"}, 1, 9, 2, Appearance → "Labeled"}, ContinuousAction → False, SaveDefinitions → True] &[
 ImageData[First@ColorSeparate[Ton3CD21dreikanalausgleichF2], "Byte"]]
```

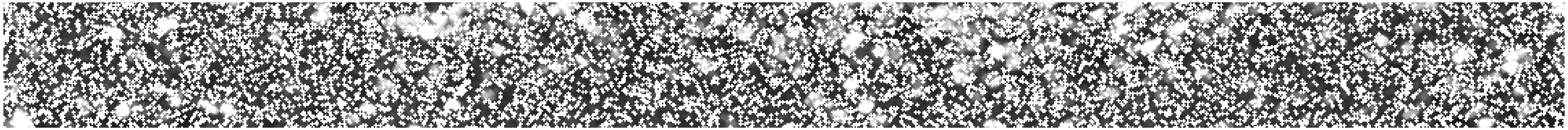


Cave: So wirken Maximumfilter oder Minimumfilter auf extreme Bildstörungen:

```
Show[Image[GrayDilate[#1, kreismaske[(#2 - 1) / 2]], "Byte"], ImageSize → Reverse@Dimensions[#1]] &[ImageData[defektesbild, "Byte"], 3]
```







```
Show[Image[GrayErode[#1, kreismaske[(#2 - 1) / 2]], "Byte"], ImageSize → Reverse@Dimensions[#1]] &[ImageData[defektesbild, "Byte"], 3]
```