

3. Vorlesung Mikroskopische Bildverarbeitung

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

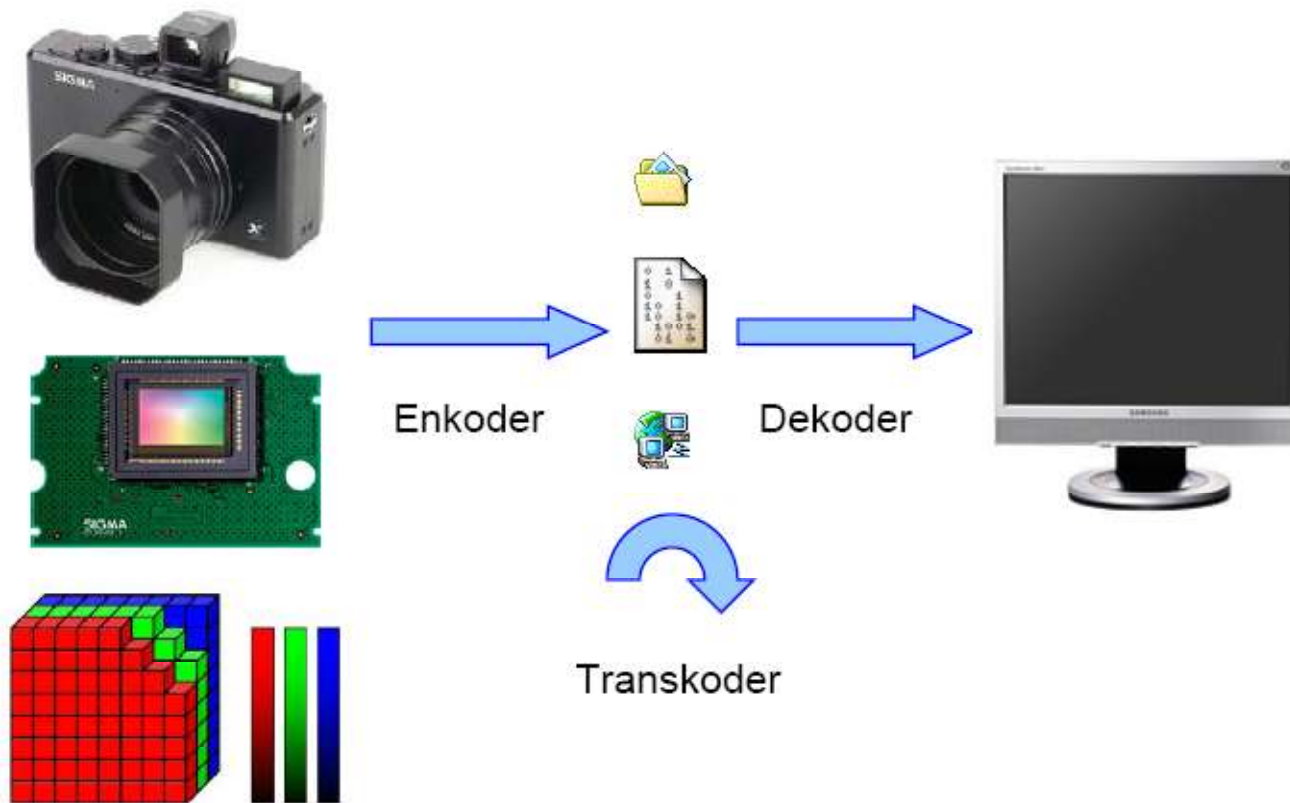
Initialisierung

20. November 2015

Wiederholung aus der 2. Vorlesung

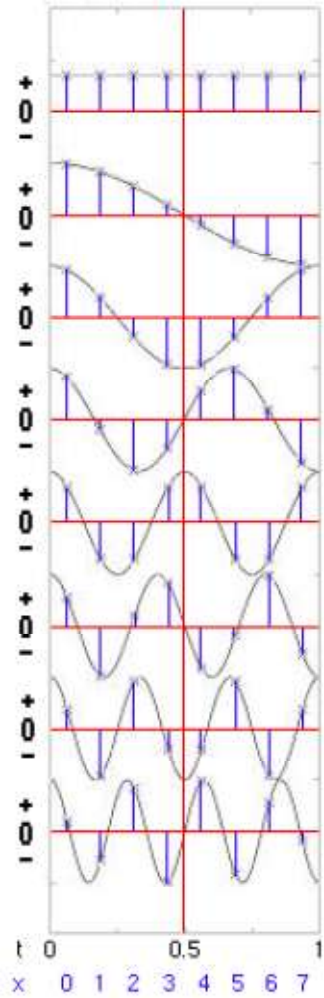
Aus 9. Digitale Bildrepräsentation (Guido Vollbeding)

Die drei grundlegenden Schritte:



Die vier Fundamenteigenschaften der digitalen Bildrepräsentation mittels DCT:

1. Skalierbarkeit



Index (u)

0



1



2



3



4



5



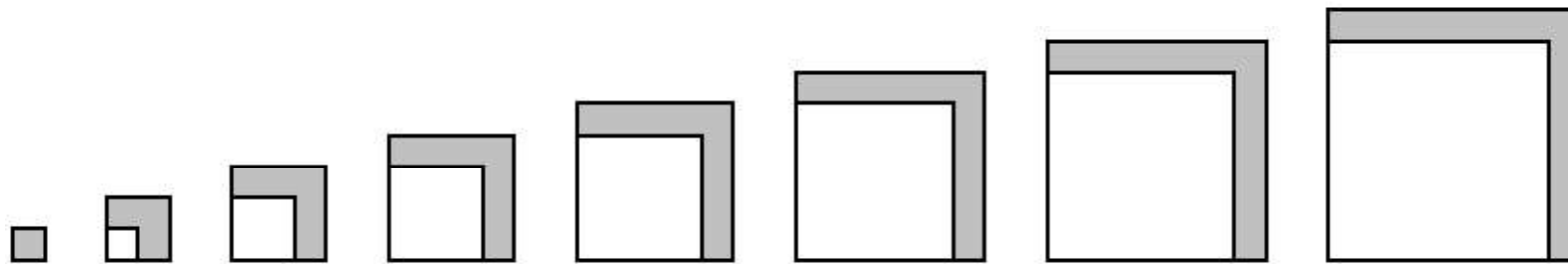
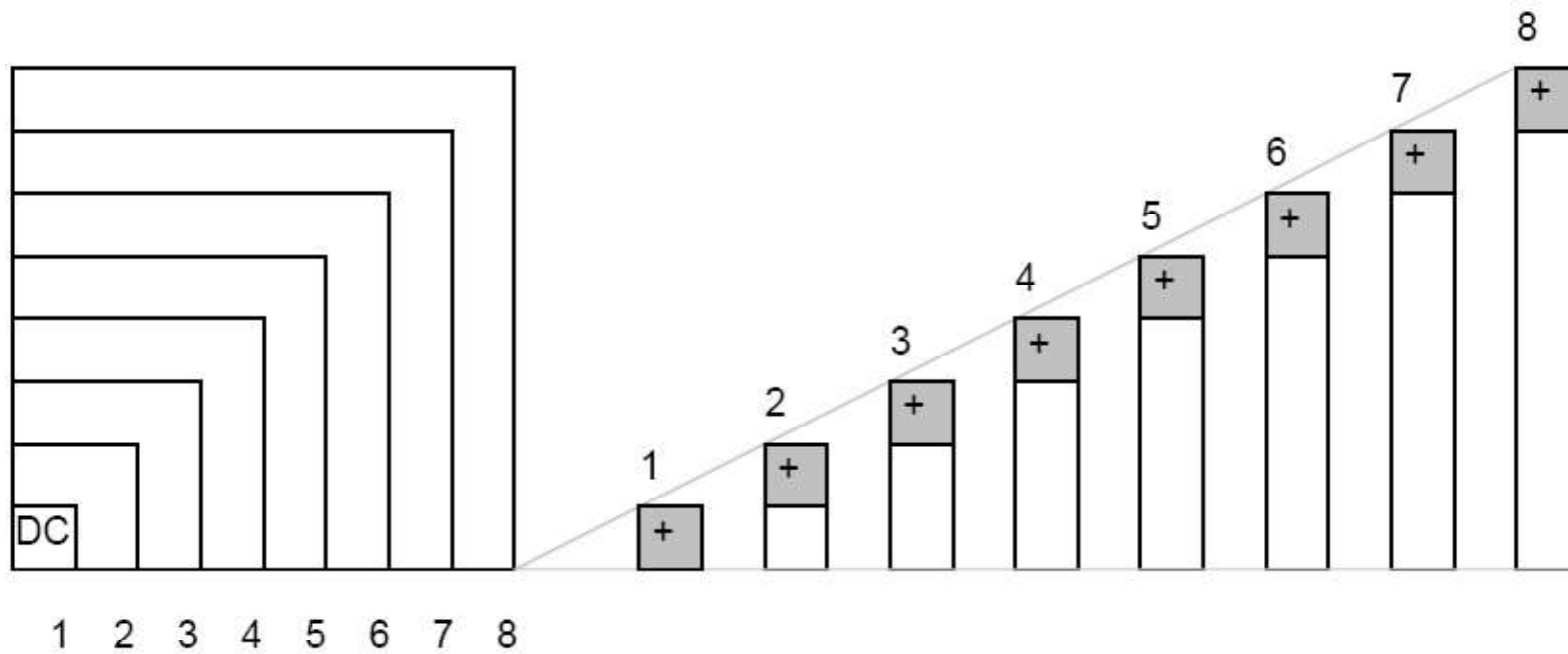
6



7

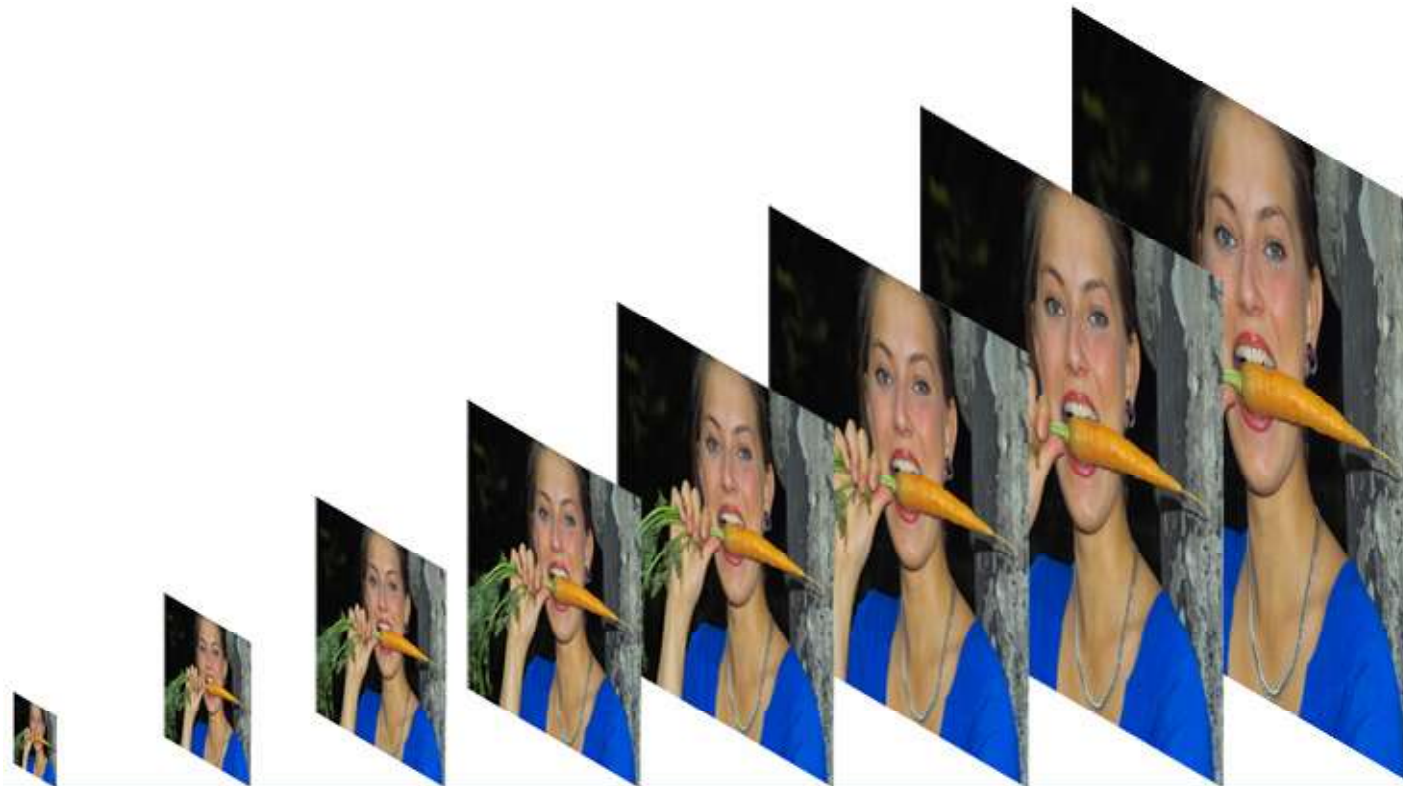


Schwingungsmuster (Basisfunktionen) der 8 – Punkt – DCT



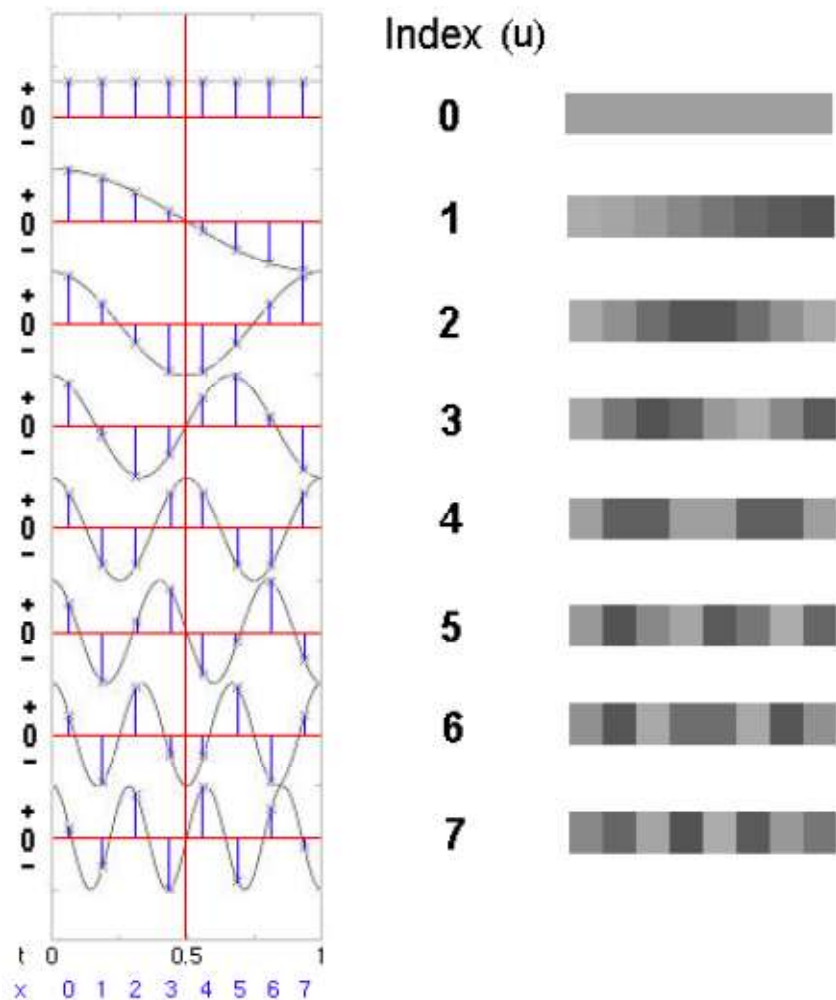
DCT-Block Auflösungsbander und additive Auflösungsfolge

Interpretation der DCT als "Bildpyramide"



Komprimierbarkeit mittels DCT-Verfahren, indem DCT-Koeffizienten höherer Auflösungsstufen stärker reduziert (Quantisierung der Koeffizienten) oder ganz weggelassen werden.

2. Stetiges Skalierungsgesetz



Zusammenhang zwischen diskreter und kontinuierlicher Form

→ Potential für freie Skalierbarkeit

Diskrete Cosinustransformation (DCT) : reellwertige, diskrete, lineare, orthogonale Transformation

$$1\text{ D - Hintransformation : } \hat{f}_u = \sum_{m=0}^{M-1} f_m \cos\left[\frac{\pi\left(m + \frac{1}{2}\right)u}{M}\right] \quad \text{mit } u = 0 \dots M-1$$

$$1\text{ D - Rücktransformation : } f_m = \frac{1}{M} \left(\hat{f}_0 + 2 \sum_{u=1}^{M-1} \hat{f}_u \cos \left[\frac{\pi \left(m + \frac{1}{2} \right) u}{M} \right] \right) \quad \text{mit } m = 0 \dots M - 1$$

$$2\text{ D - Hintransformation : } \hat{f}_{u,v} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n} \cos \left[\frac{\pi \left(m + \frac{1}{2} \right) u}{M} \right] \cos \left[\frac{\pi \left(n + \frac{1}{2} \right) v}{N} \right] \quad \text{mit } u = 0 \dots M - 1 \text{ und } v = 0 \dots N - 1$$

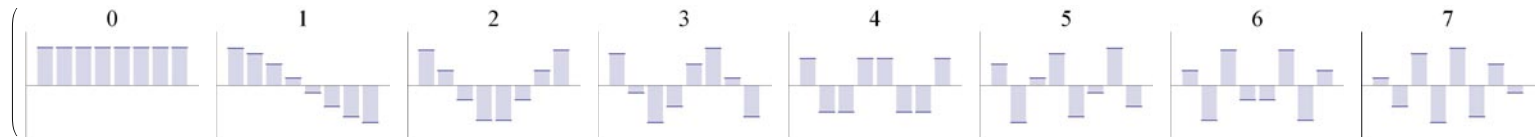
$$2\text{ D - Rücktransformation : } f_{m,n} =$$

$$\frac{1}{MN} \left(\hat{f}_{0,0} + 2 \sum_{u=1}^{M-1} \sum_{v=1}^{N-1} \hat{f}_{u,v} \cos \left[\frac{\pi \left(m + \frac{1}{2} \right) u}{M} \right] \cos \left[\frac{\pi \left(n + \frac{1}{2} \right) v}{N} \right] \right) \quad \text{mit } m = 0 \dots M - 1 \text{ und } n = 0 \dots N - 1$$

Basisfunktionen der eindimensionalen 8-Punkt-DCT:

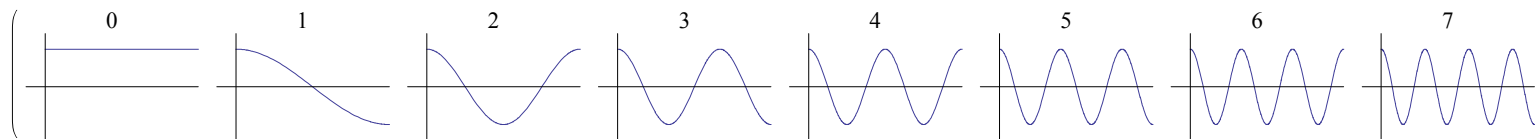
`mm = 8;`

`MatrixForm[{Table[DiscretePlot[Cos[$\pi (2m+1)u / (2mm)$] / Sqrt[2], {m, 0, mm-1}], PlotMarkers → None,
Ticks → None, PlotRange → {{-1, mm}, {-1, 1}}, ExtentSize → .8, PlotLabel → u], {u, Range[0, mm-1]}]}`



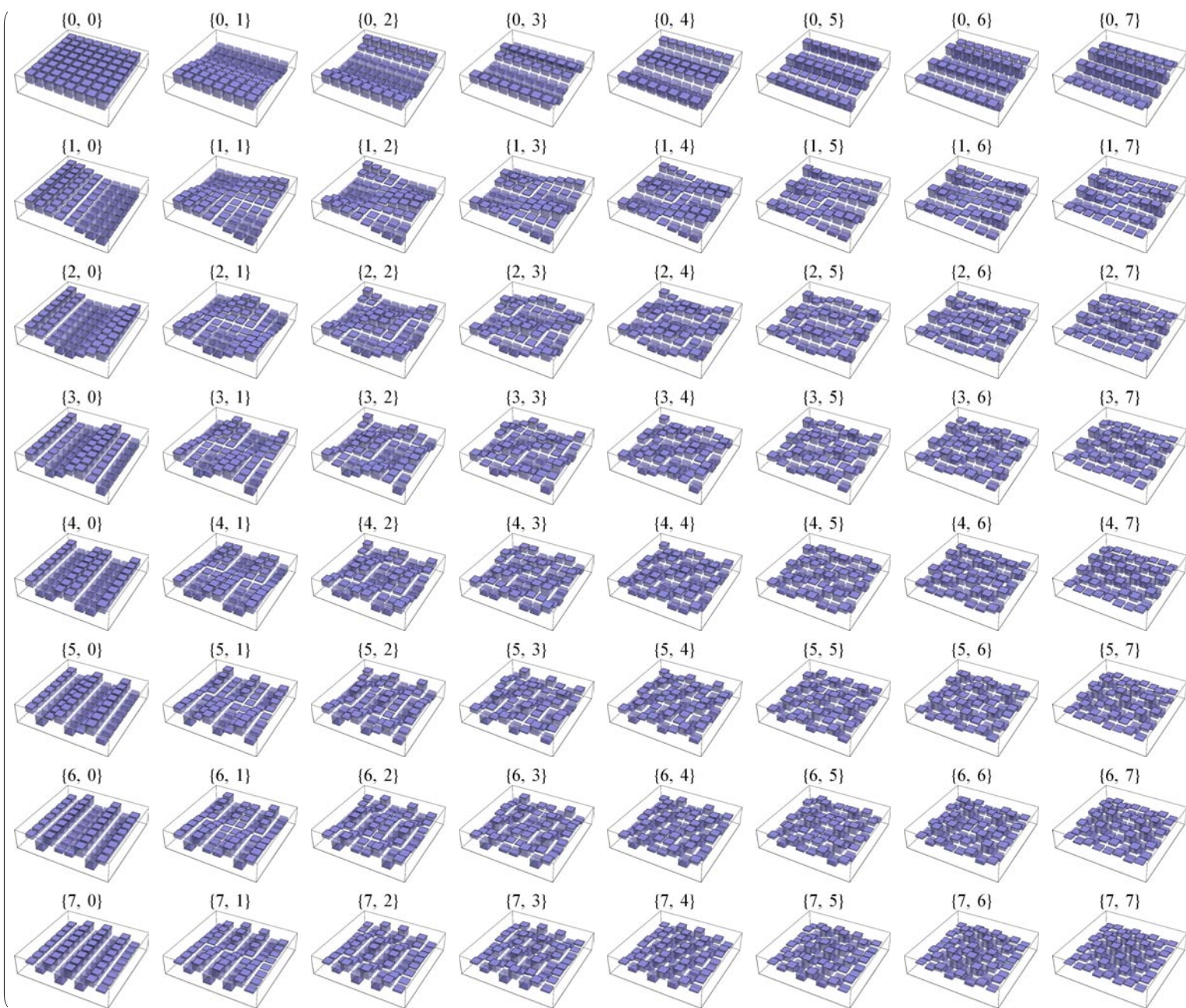
`mm = 8;`

`MatrixForm[{Table[Plot[Cos[ $\pi u m / mm$ ] / Sqrt[2], {m, 0, mm}], Ticks → None, PlotRange → {{-1, mm}, {-1, 1}}, PlotLabel → u], {u, Range[0, mm-1]}]}`



Basisfunktionen der zweidimensionalen 8-Punkt-DCT:

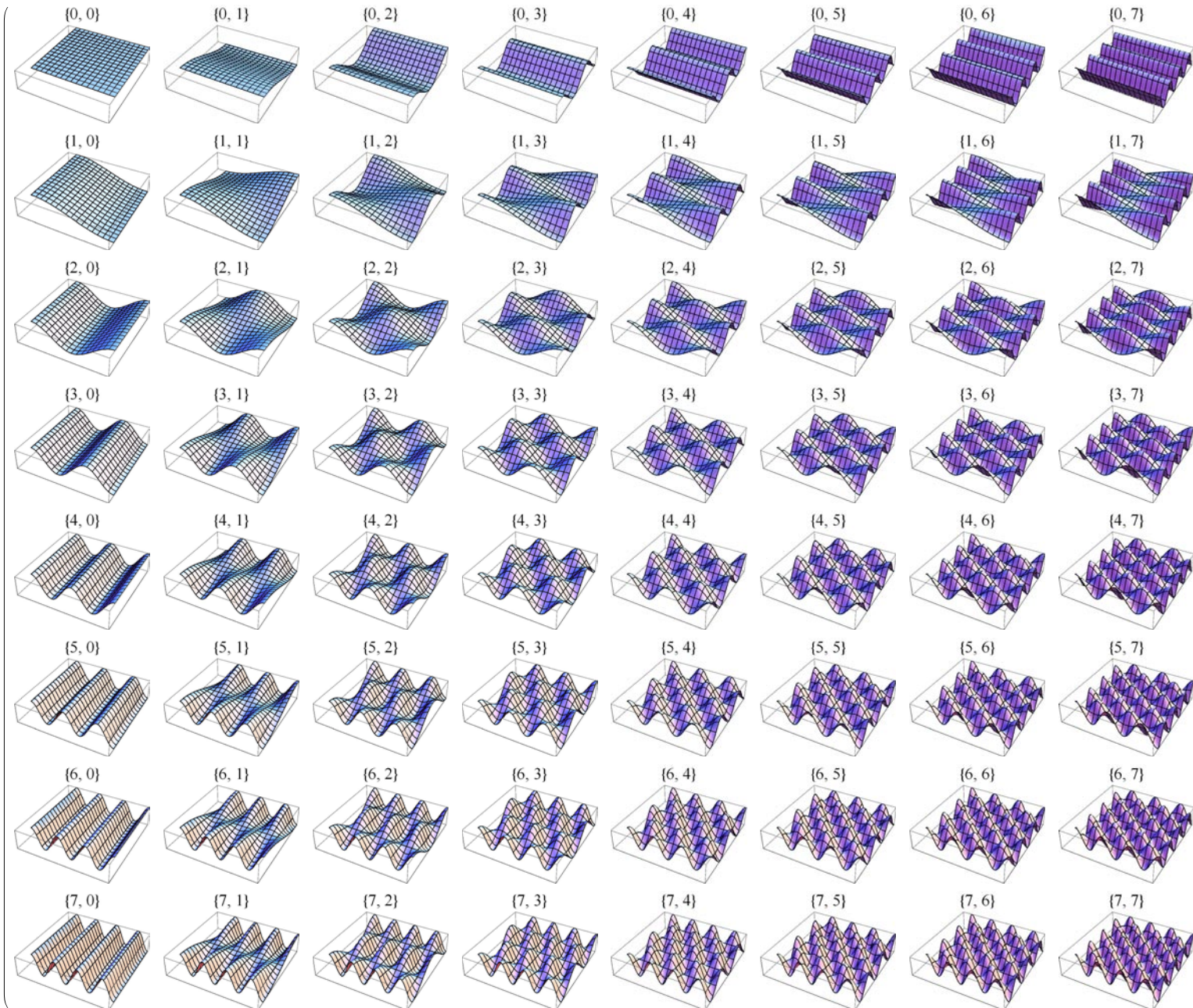
```
mm = 8;  
nn = 8;  
MatrixForm[Table[DiscretePlot3D[Cos[ $\pi$  (m + 1 / 2) u / mm] * Cos[ $\pi$  (n + 1 / 2) v / nn], {m, 0, mm - 1},  
  {n, 0, nn - 1}, PlotMarkers → None, Ticks → None, PlotRange → {{-1, mm}, {-1, nn}, {-1, 1}}, ExtentSize → .8,  
  PlotLabel → {u, v}, BoxRatios → {mm + 1, nn + 1, 2}, RotationAction → "Clip"], {u, Range[0, mm - 1]}, {v, Range[0, nn - 1]}]]
```



```
mm = 8;
```

```
nn = 8;
```

```
MatrixForm[Table[Plot3D[Cos[ $\pi$  u m / mm] * Cos[ $\pi$  v n / nn], {m, 0, mm}, {n, 0, nn}, Ticks → None, PlotRange → {{-1, mm}, {-1, nn}, {-1, 1}},  
PlotLabel → {u, v}, BoxRatios → {mm + 1, nn + 1, 2}, RotationAction → "Clip"], {u, Range[0, mm - 1]}, {v, Range[0, nn - 1]}]]
```



3. “Gesetz” der Acht

Experimentell hat sich erwiesen, daß 8 die passende Länge ist, hinsichtlich der Verarbeitbarkeit und der Detailliertheit der Bilder, und hinsichtlich des Größenverhältnisses von Vollbild- und Vorschaudarstellung

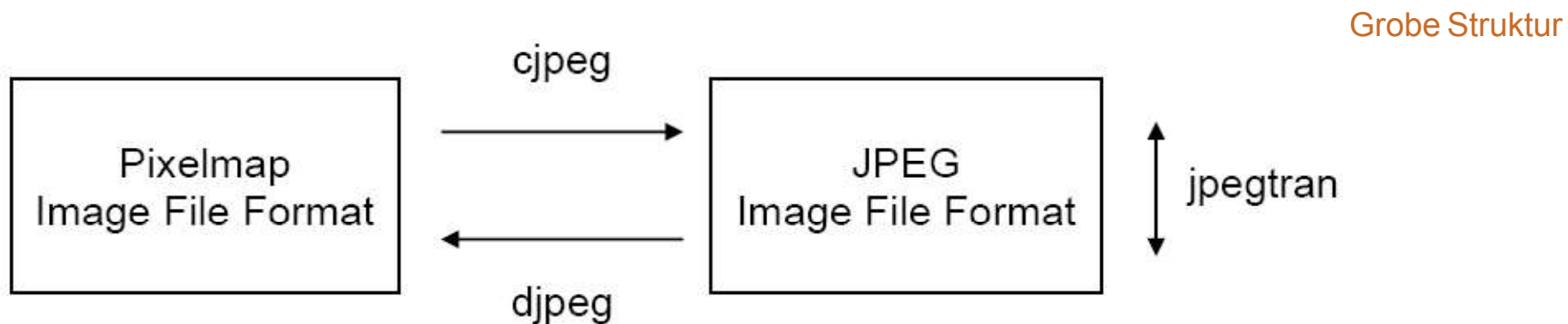
4. Analogie zur Signal-/Energieübertragung

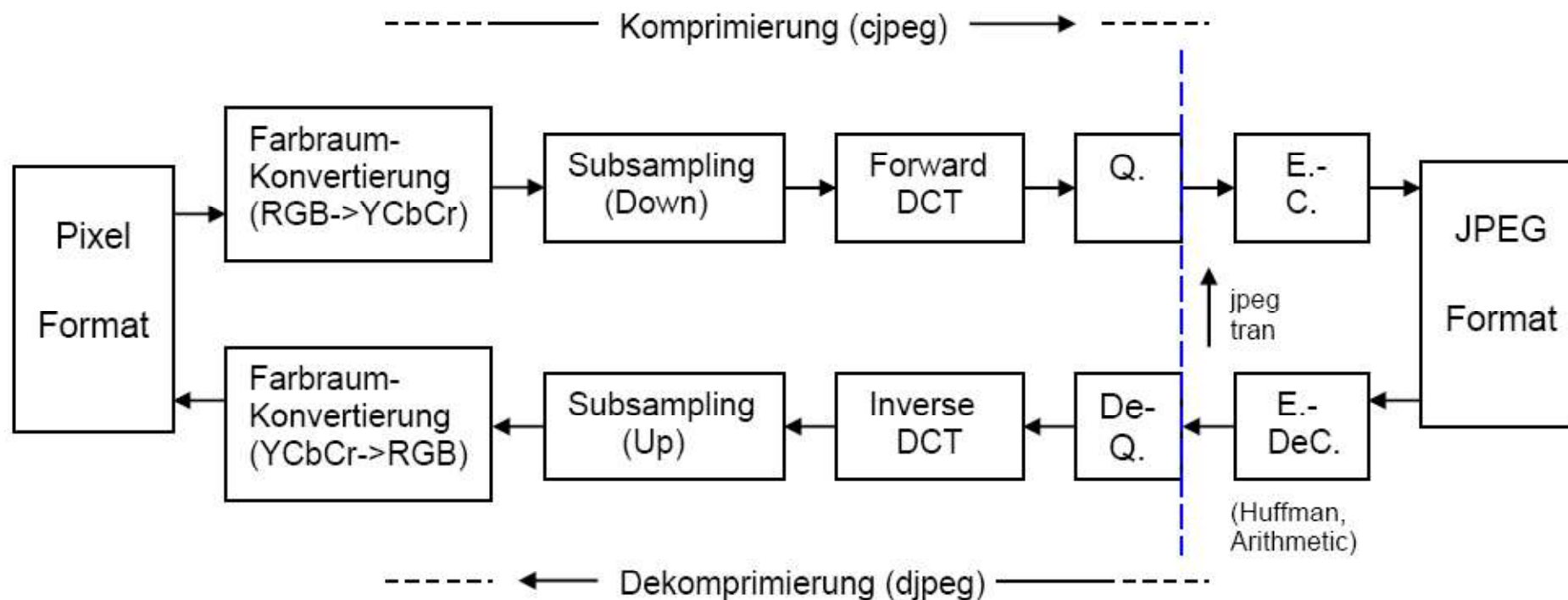
0. Koeffizient steht für DC, höhere Koeffizienten stehen für AC

Die Prozeßgliederung des JPEG-Verfahrens:

Independent JPEG Group

JPEG Software Tools





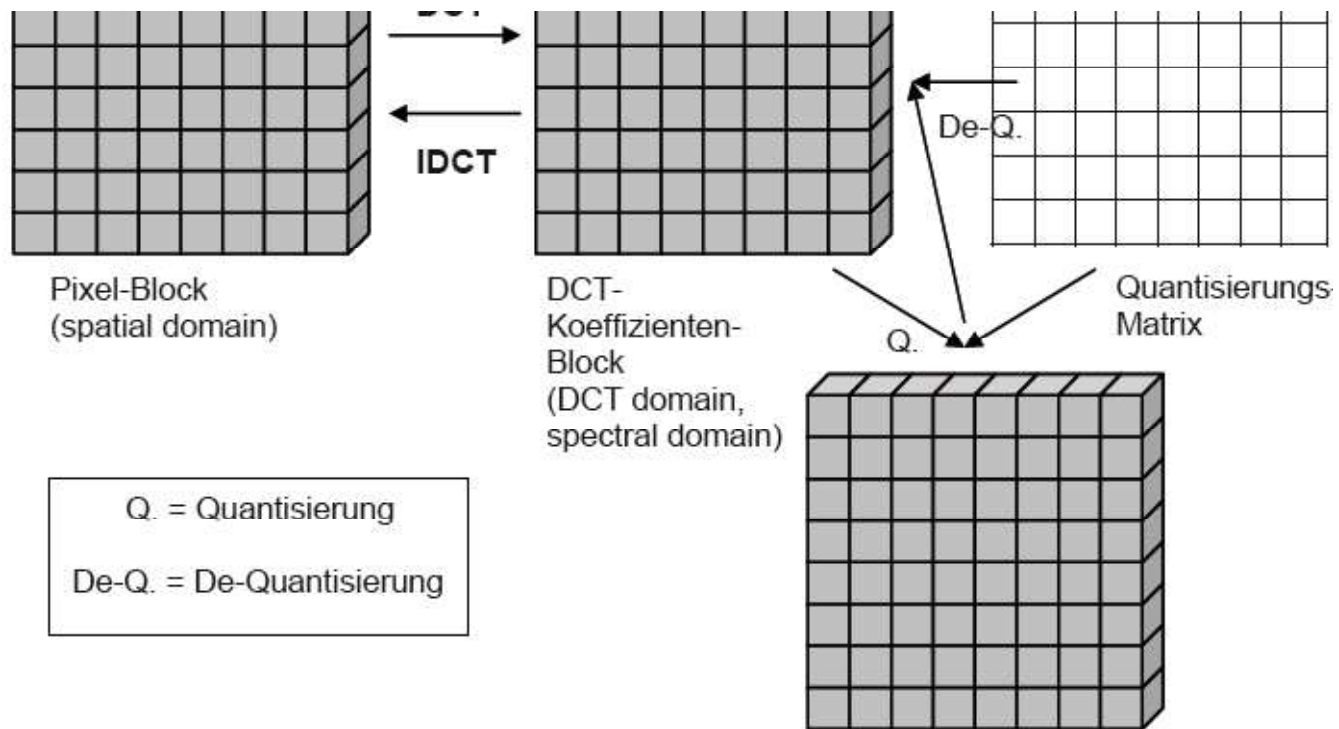
Im Detail

- Teilprozess der Entropie-Kodierung bzw. -Dekodierung am einen Ende der Prozesskette operiert vollkommen verlustfrei
- alle anderen Teile unterliegen einem mehr oder weniger großen Verlustpotential
- Hauptangriffspunkt zur „Verluststeuerung“ bei JPEG liegt im Teilprozess der Quantisierung
- durch Vorgabe wird dort der Verlustgrad bestimmt
- weiterer Angriffspunkt zur JPEG Verluststeuerung (in prakt. Anwendungen leider oft unterschätzt / übersehen): Farbrunterabtastung
- „Herz“ des JPEG Prozesses: DCT oder FDCT [Forward DCT]) bzw. ihr Gegenstück IDCT
- Pixelwerte werden in „Schwingungswerte“ umgewandelt und umgekehrt
- Umwandlung geschieht „theoretisch“ verlustfrei, aufgrund begrenzter Rechengenauigkeit nicht hundertprozentig garantiert
- ebenso kann der Teilprozeß Farbraumkonvertierung rechengenauigkeitsabhängige Fehler hinzufügen



DCT





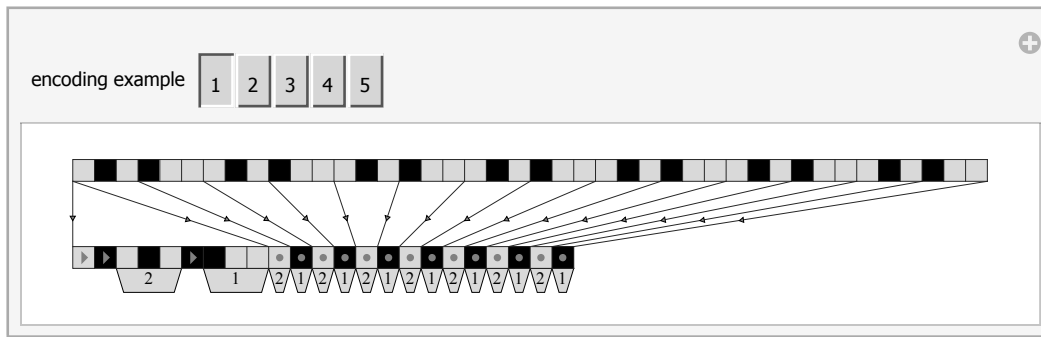
exemplarische Quantisierungstabellen

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Luminanz (table K.1)

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Chrominanz (table K.2)



Veranschaulichung der Huffman-Kodierung als einer Variante der Entropie-Kodierung - siehe <http://mathworld.wolfram.com/HuffmanCoding.html>

```
HuffmanStep[l0_List] := Module[{l = l0, s2 = Take[Select[Sort[l0], Positive], 2]}, l[[Take[Flatten[Position[l, #] & /@ s2], 2]] = 0;
  l[[Last[Position[l, 0]]] = Plus @@ s2;
  {l, s2}]

HuffmanList[l_List] := Module[{}, Plus @@@ Last /@ NestWhileList[HuffmanStep[First[#]] &, HuffmanStep[l], Length[Union[First[#]]] > 2 &]]

HuffmanTable[l_List] := NestWhileList[First[HuffmanStep[#]] &, l, Length[Union[#]] > 2 &]
```

Beispiel anhand einer fiktiven Häufigkeit von 13 verschiedenen Symbolen mit der ebenso fiktiven Häufigkeit entsprechend der ersten 13 Primzahlen:

Prime@Range@13

```
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41}
```

MatrixForm@HuffmanTable@Prime@Range@13

```
( 2 3 5 7 11 13 17 19 23 29 31 37 41
  0 5 5 7 11 13 17 19 23 29 31 37 41
  0 0 10 7 11 13 17 19 23 29 31 37 41
  0 0 0 17 11 13 17 19 23 29 31 37 41
  0 0 0 17 0 24 17 19 23 29 31 37 41
  0 0 0 0 0 24 34 19 23 29 31 37 41
  0 0 0 0 0 24 34 0 42 29 31 37 41
  0 0 0 0 0 0 34 0 42 53 31 37 41
  0 0 0 0 0 0 0 0 42 53 65 37 41
  0 0 0 0 0 0 0 0 42 53 65 0 78
  0 0 0 0 0 0 0 0 0 0 65 95 78
  0 0 0 0 0 0 0 0 0 0 0 95 143
  0 0 0 0 0 0 0 0 0 0 0 0 238)
```

Diese Matrix kodiert einen Baum.

Wollte man das zugrundeliegende Symbol kodieren, daß mit 41 Auftritten am häufigsten erscheint, braucht man die Folge 000, also in der Spalte gerade herunter, es kommen drei Erhöhungen durch Summanden von links. Wollte man hingegen das 31mal auftretende Symbol kodieren, braucht man die Folge 010, also eine Erhöhung in der Spalte (Summation mit einem links stehenden Summanden), dann eine Summation mit der rechts stehenden Zwischensumme, und dann wieder einer Erhöhung in der Spalte.

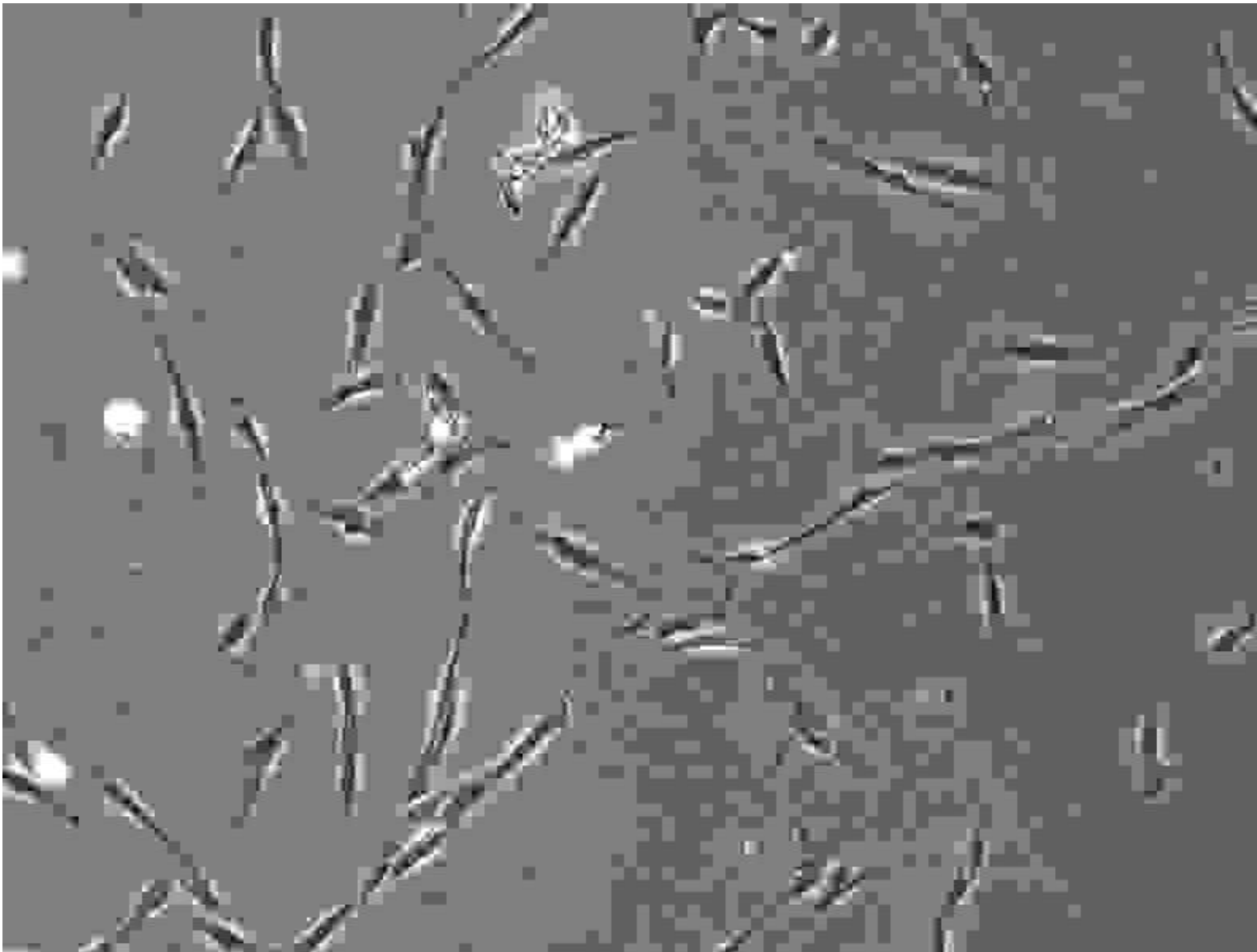
Nutzungsmöglichkeiten fortgeschrittener JPEG-Funktionen:

- verlustfreie Re-Komprimierung bestehender JPEGs durch Varianten der Entropiekodierung
- verlustfreie Operationen am Bild (Skalierung, Rotation, Ausschneiden)
- vergrößerter Farbumfang (Wide Gamut)
- erweiterte Bittiefen

Können wir die Bildgüte vergleichend messen? Und wie? Mit PSNR (Spitzen-Signal-Rausch-Verhältnis)! Und via MSE (mittlerer quadratischer Abweichung). Wir betrachten das Bildpaar I und K:

$$\text{PSNR} = 10 * \text{Log}_{10} \left[\frac{I_{\max}^2}{\text{MSE}} \right] \text{ dB mit } \text{MSE} = \frac{1}{m \cdot n} \sum_{i=1}^m \sum_{j=1}^n (I(i, j) - K(i, j))^2$$

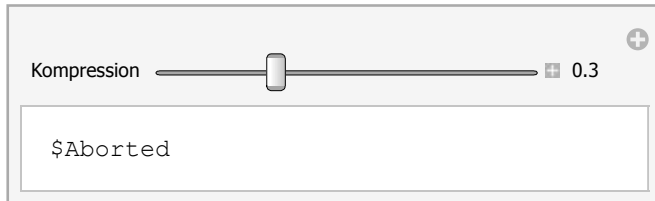
```
Function[ColorConvert[ImportString[ExportString[#, "JPEG", "CompressionLevel" -> 1.0], "JPEG"], ImageColorSpace[#]] [
  ImageResize[phasenkontrast, Scaled[1 / 2]]]
```



```
Function[{bild}, -10 * Log10[
  ImageDistance[bild, ColorConvert[ImportString[ExportString[bild, "JPEG", "CompressionLevel" → 1.0], "JPEG"], ImageColorSpace[bild]],
  DistanceFunction → "MeanSquaredEuclideanDistance"] / ImageChannels[bild]] [ImageResize[phasenkontrast, Scaled[1 / 2]]]
```

24.4383

```
Function[{bild},
  Manipulate[Block[{komprimiertesbild = ColorConvert[ImportString[ExportString[bild, "JPEG", "CompressionLevel" → compression], "JPEG"],
    ImageColorSpace[bild]]}, Grid[{{Show[komprimiertesbild, ImageSize → ImageDimensions[bild]],
    -10 * Log10[ImageDistance[bild, komprimiertesbild, DistanceFunction → "MeanSquaredEuclideanDistance"] / ImageChannels[bild]]}}],
  {{compression, .5, "Kompression"}, 0., 1., .05, Appearance → "Labeled"}, ContinuousAction → False,
  SaveDefinitions → True][ImageResize[phasenkontrast, Scaled[1 / 2]]]
```



10. Histogramme, Histogrammausgleich, Histogrammanpassung

Bildstatistik: diskrete Häufigkeitsverteilungen von (hier: skalaren) Bildintensitäten

Histogramme

- geben Aufschluß über die Einhaltung des Wertebereichs
- bieten Anhaltspunkte für die Korrektur oder Normierung der Intensitäten
- erlauben die Gewinnung von Kriterien für Schwellwertverfahren

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 ($G_{in} \rightarrow G_{out}$) erstellt:

$$G_{out}(i) = \frac{G_{max} \sum_{j=0}^{G_{in}(i)} h(j)}{M N}$$

M, N : Bildhöhe/-breite

$G_{\max}$: maximale Anzahl von Intensitätsstufen (Vorgabe)

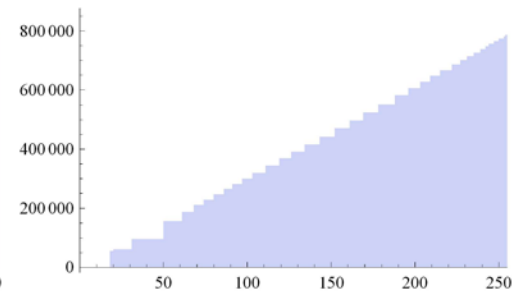
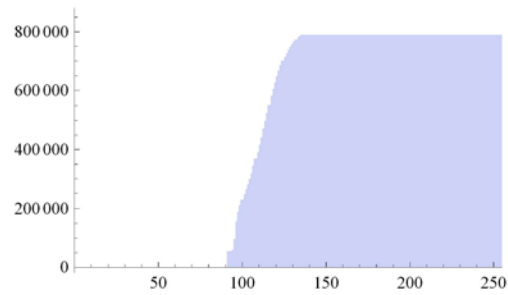
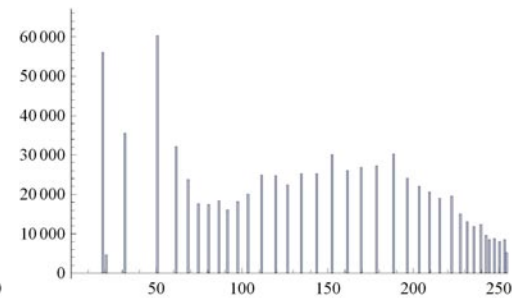
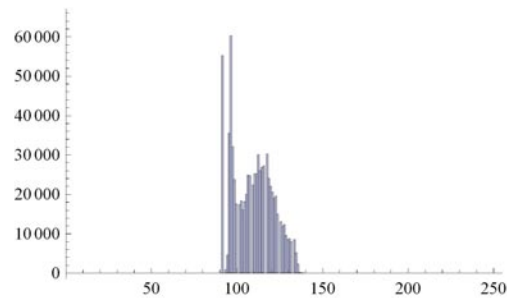
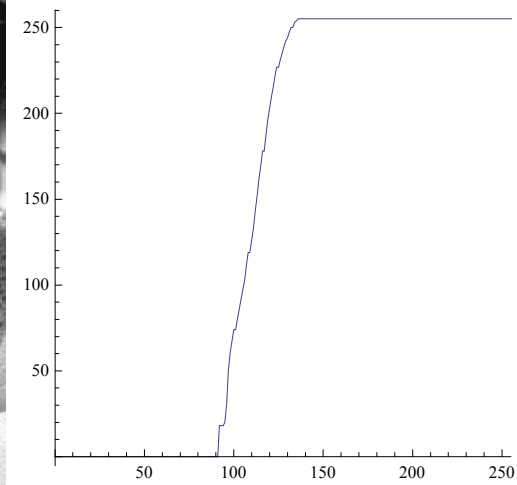
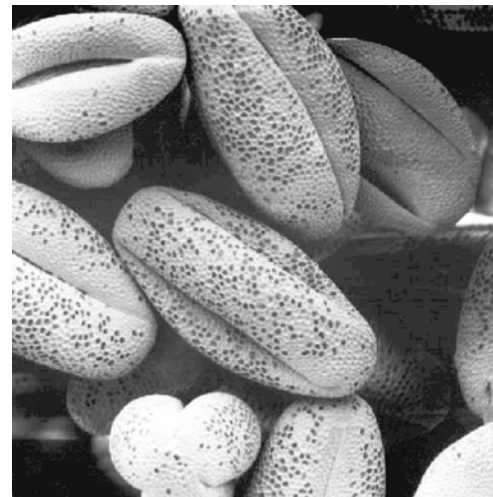
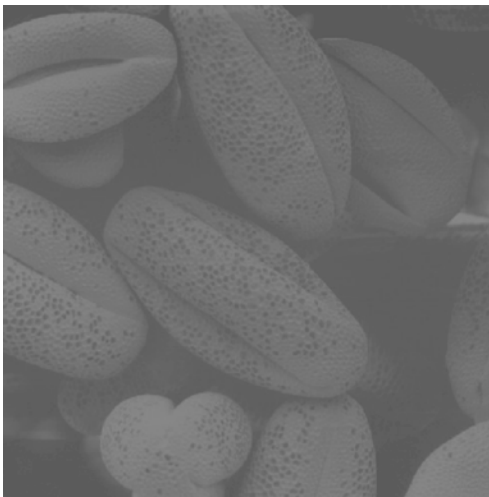
$h(i)$: absolute Häufigkeit der Intensitätsstufe i

```
Clear[histogrammausgleich]
histogrammausgleich[skalarbildein_Image] := Module[{stufenanzahlen, kumulation, skalarbilddaus},
  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]}}]];
  *)
  skalarbilddaus = Image[Map[kumulation[[# + 1]] &, ImageData[skalarbildein, "Byte"], {2}], "Byte"];
  Print[Grid[{{Show[skalarbildein, ImageSize -> 256], Show[skalarbilddaus, ImageSize -> 256],
    Show[ListLinePlot[kumulation, AspectRatio -> 1, PlotRange -> {{0, 255}, All}], ImageSize -> 256]}}]];

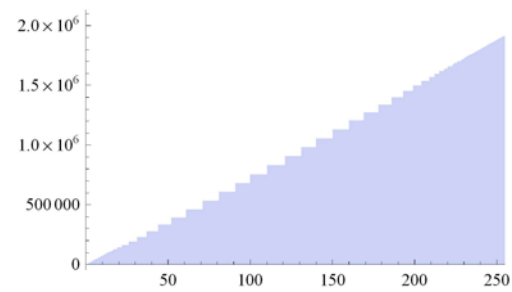
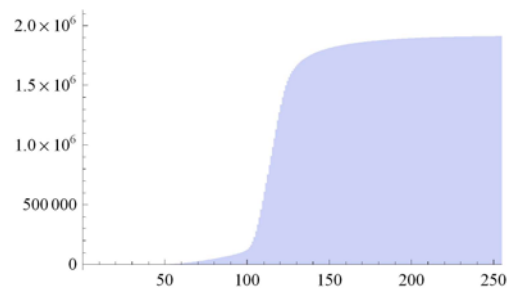
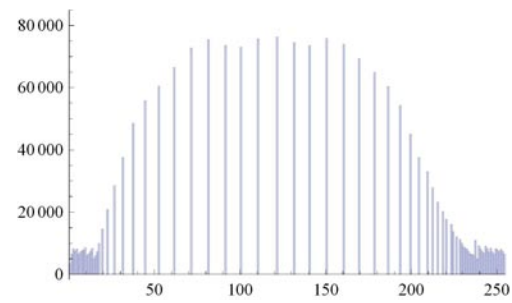
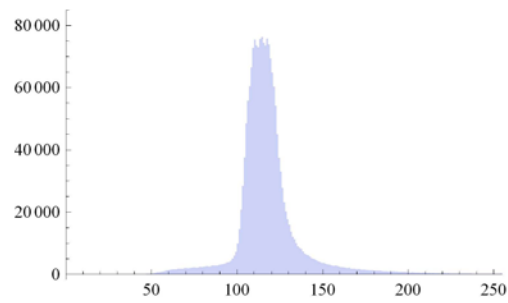
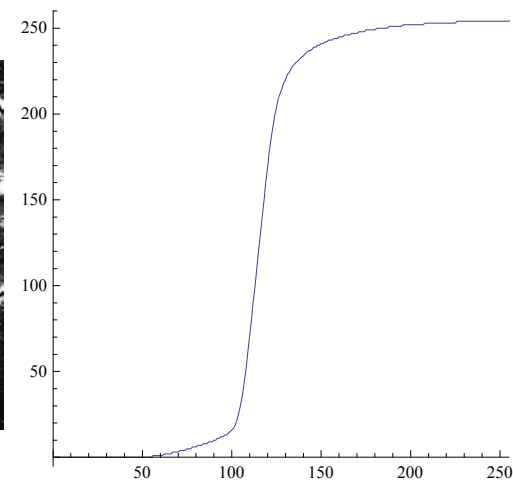
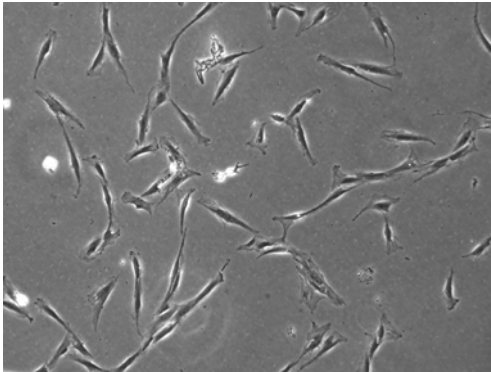
  Print[Grid[{{Show[Histogram[Flatten[ImageData[skalarbildein, "Byte"]], {0, 255, 1}, "Count", PlotRange -> {{0, 255}, Automatic},
    AxesOrigin -> {0, 0}], ImageSize -> 256], Show[Histogram[Flatten[ImageData[skalarbilddaus, "Byte"]],
    {0, 255, 1}, "Count", PlotRange -> {{0, 255}, Automatic}, AxesOrigin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

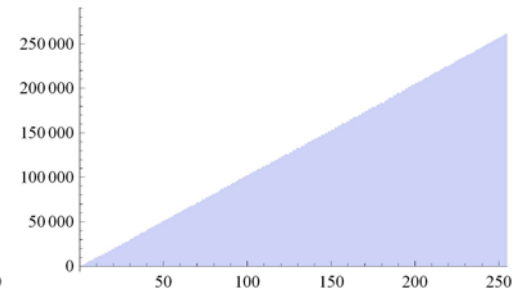
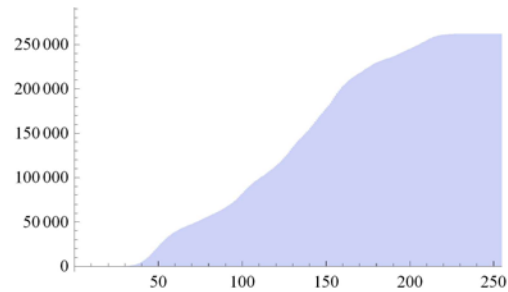
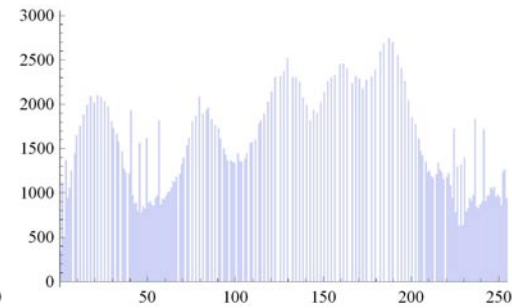
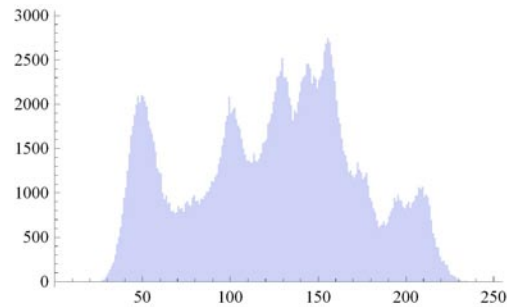
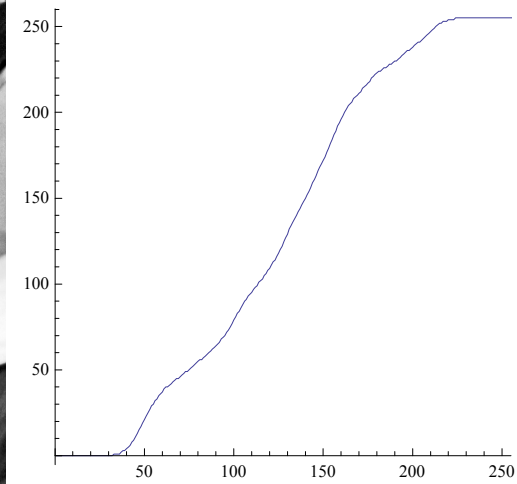
histogrammausgleich[pollen];
```



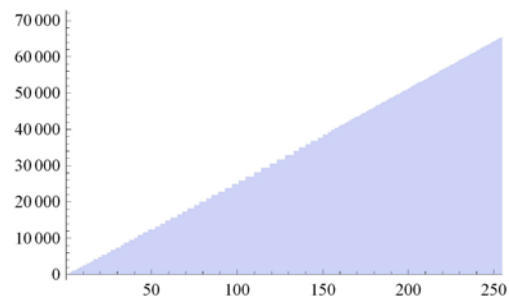
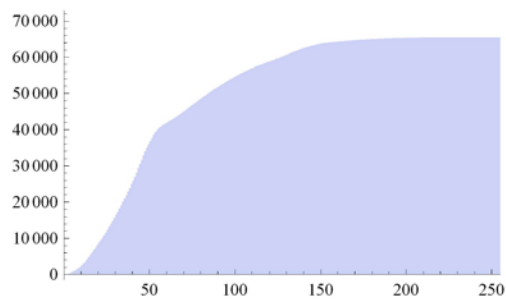
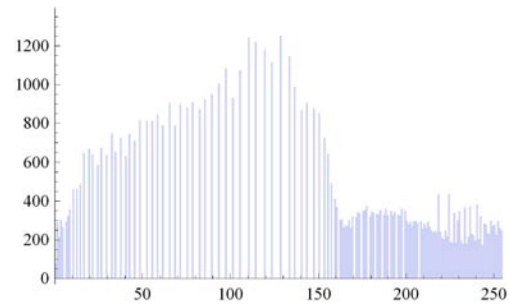
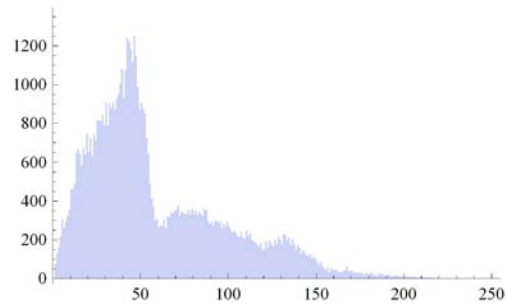
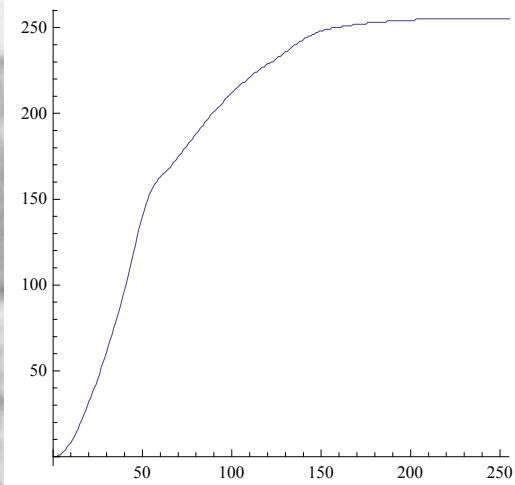
```
histogramausgleich[phasenkontrast];
```



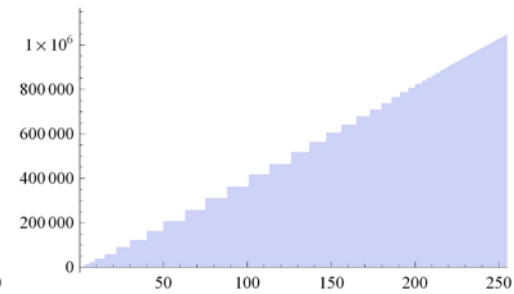
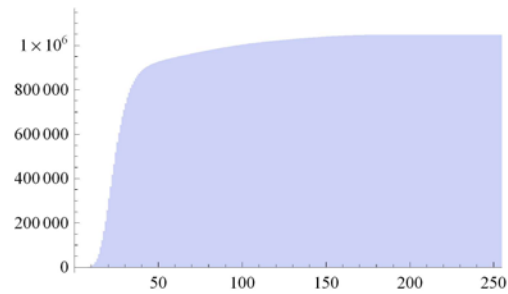
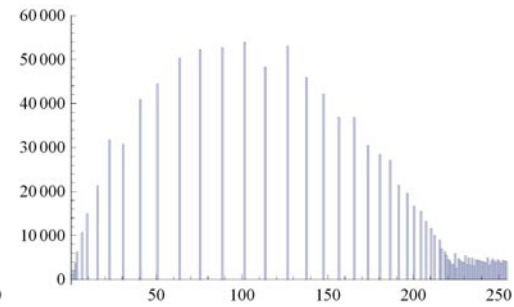
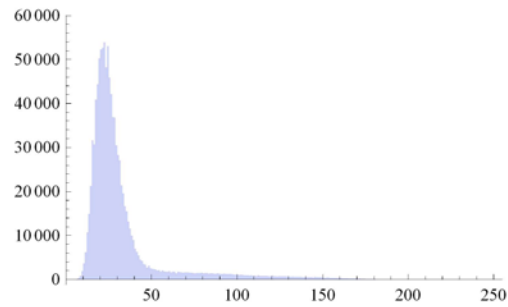
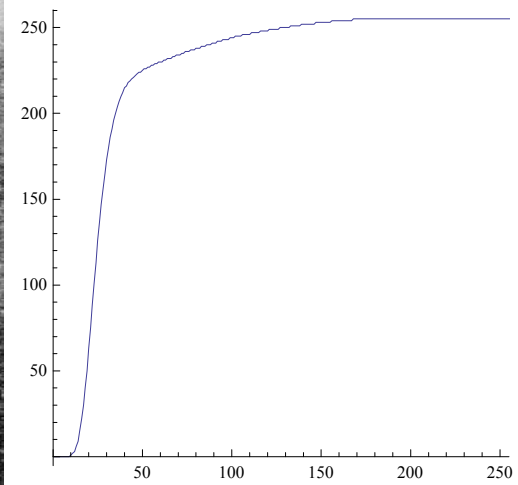
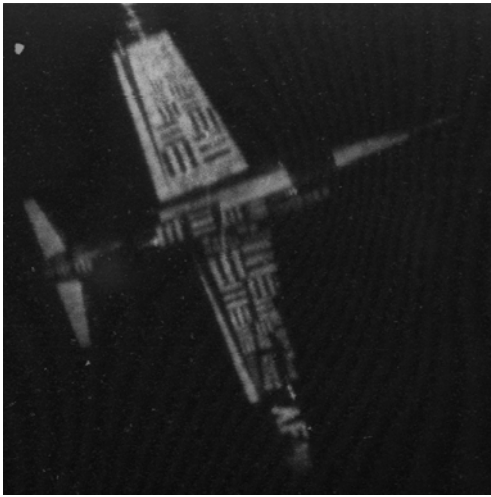
```
histogramausgleich[ColorConvert[ExampleData[{"TestImage", "Lena"}], "Graylevel"]];
```



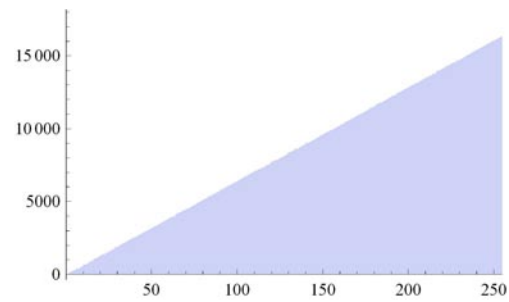
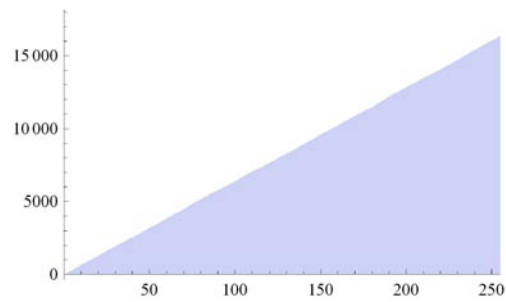
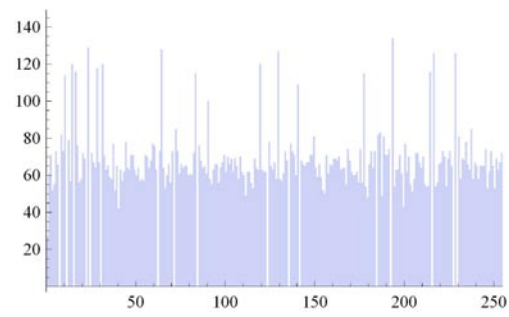
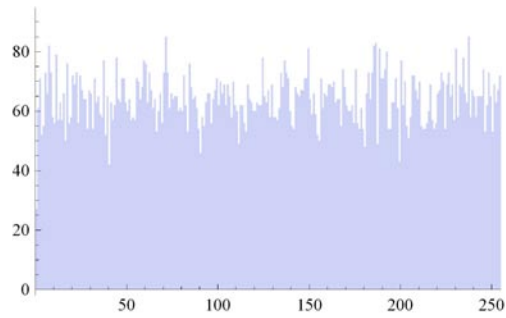
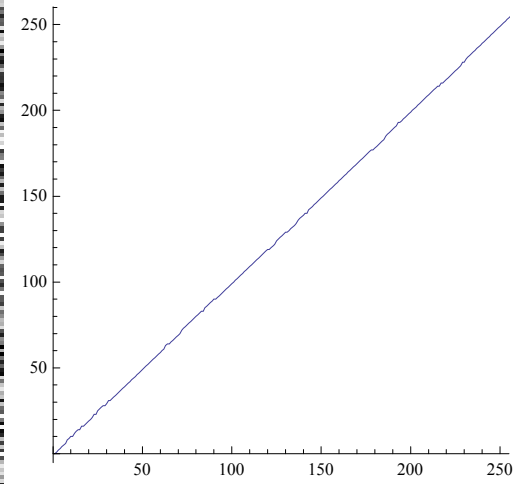
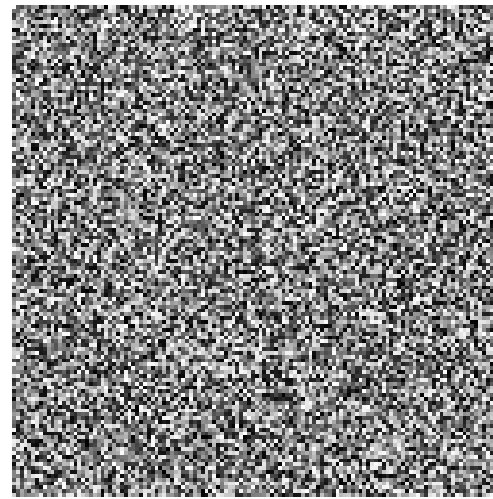
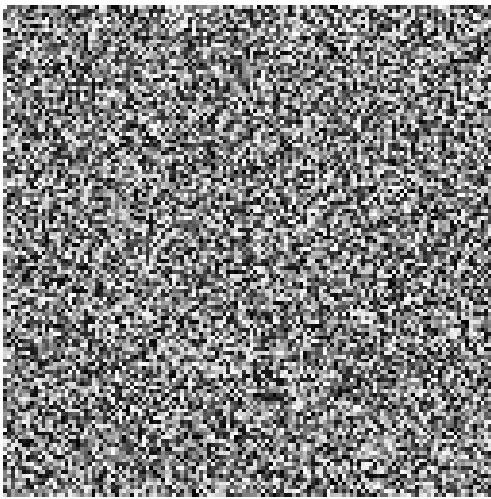
```
histogramausgleich[ColorConvert[ExampleData[{"TestImage", "Girl"}], "Graylevel"]];
```



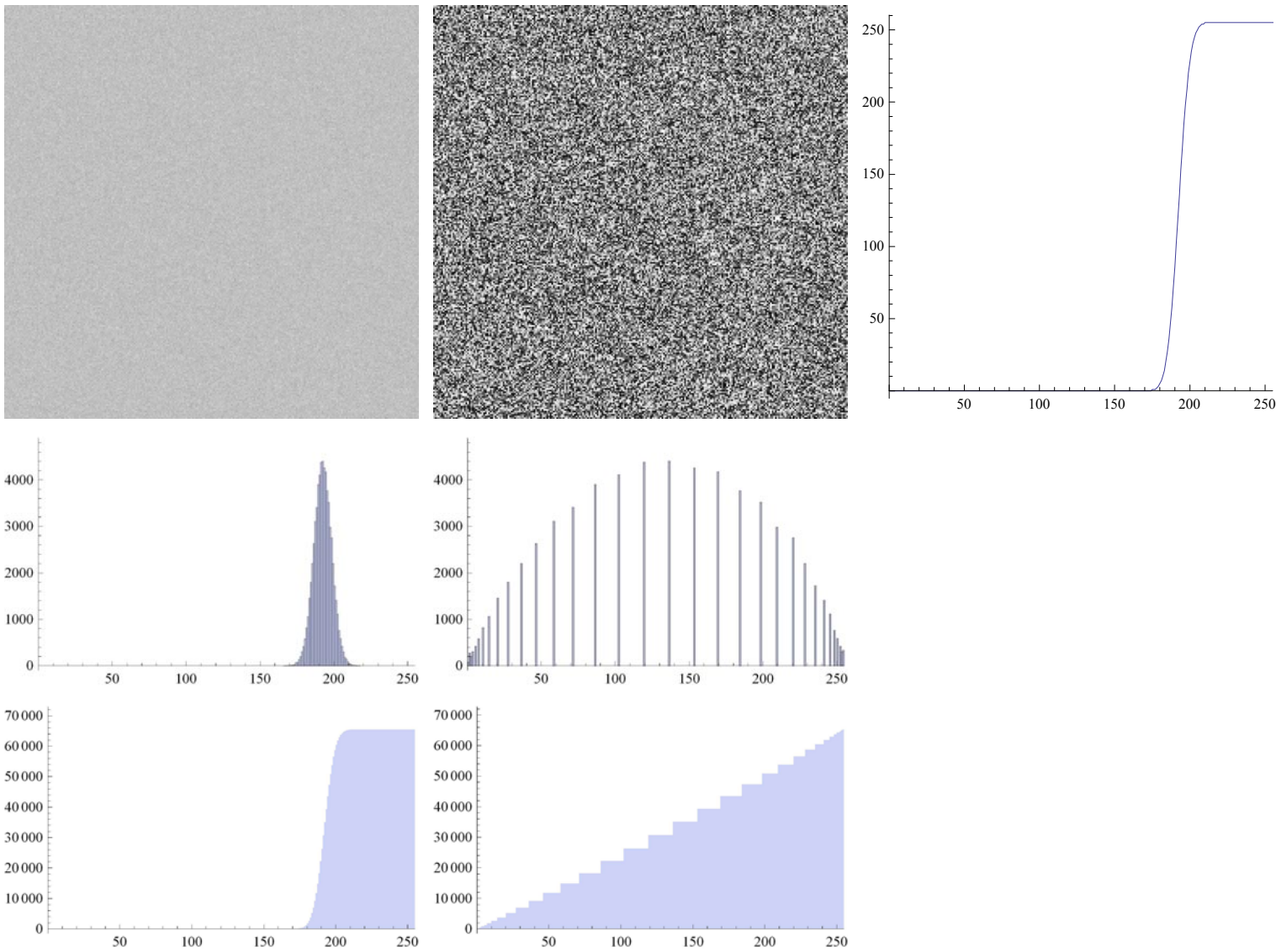
```
histogramausgleich[ExampleData[{"TestImage", "U2"}]];
```



```
histogramausgleich[randombild];
```



```
histogrammausgleich[rauschbild];
```



Histogrammanpassung

- Ziel ist die möglichst genaue Anpassung der Verteilungsfunktion $p_{\text{in}}(i)$ eines Bildes G_{in} an die Verteilungsfunktion $p_{\text{ref}}(i)$ eines Referenzbildes G_{ref} für alle i des Wertebereichs $[0 \dots G_{\text{max}}]$

$$p_{\text{in}}(i) = \frac{\sum_{j=0}^{G_{\text{in}}(i)} h_{\text{in}}(j)}{h_{\text{in}}(G_{\text{max}})}$$

$$p_{\text{ref}}(i) = \frac{\sum_{j=0}^{G_{\text{in}}(i)} h_{\text{ref}}(j)}{h_{\text{ref}}(G_{\text{max}})}$$

G_{max} : maximale Anzahl von Intensitätsstufen (Vorgabe, typ. 255)

$h(i)$: absolute Häufigkeit der Intensitätsstufe i

- 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:

$$G_{\text{out}}(i) = p_{\text{ref}}^{-1}(p_{\text{in}}(i))$$

realisiert als

$$G_{\text{out}}(i) = \text{Min} \{ j \mid (0 \leq j \leq G_{\text{max}}) \wedge (p_{\text{in}}(i) \leq p_{\text{ref}}(j)) \}$$

```
Clear[histogrammanpassung]
histogrammanpassung[skalarbildein_Image, skalarbildref_Image] :=
Module[{stufenanzahlenein, stufenanzahlenref, kumulationein, kumulationref, skalarbilddaus, 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}];

skalarbilddaus = Image[Map[lut[[#]] &, ImageData[skalarbildein, "Byte"] + 1, {2}], "Byte"];
(*Print[Grid[{{Show[skalarbildein,ImageSize→256],Show[skalarbilddaus,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[skalarbilddaus, 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[skalarbilddaus,"Byte"]],{0,255,1},PlotRange→{{0,255},Automatic},AxesOrigin→{0,0}],ImageSize→256];
*)

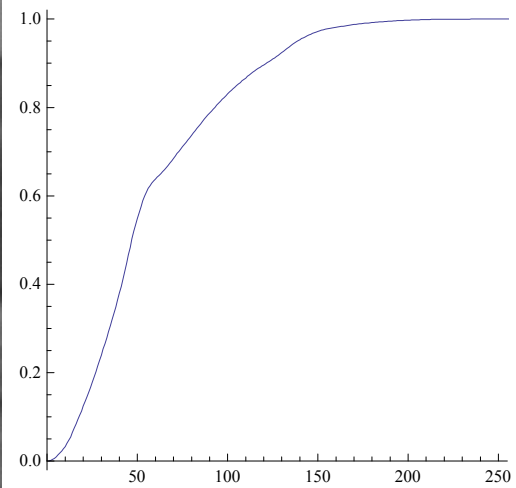
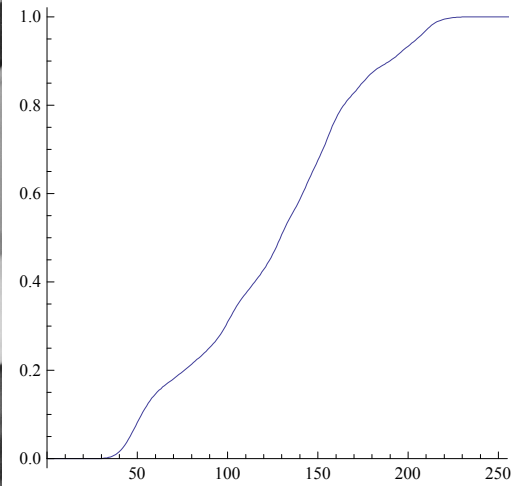
Print[Grid[{{Show[skalarbildein, ImageSize → 256], Show[skalarbildref, ImageSize → 256], Show[skalarbilddaus, ImageSize → 256]}]}];

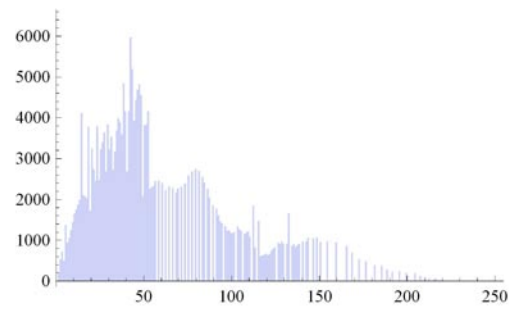
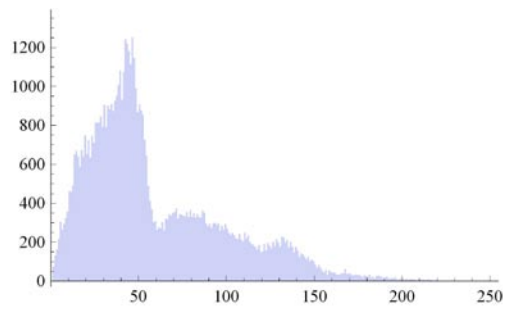
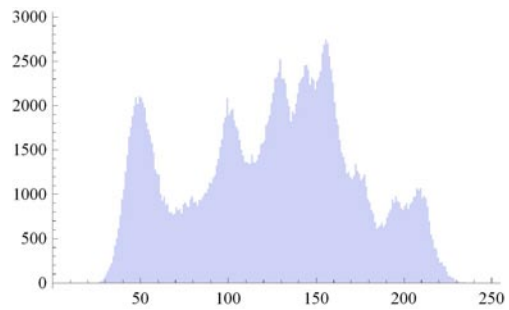
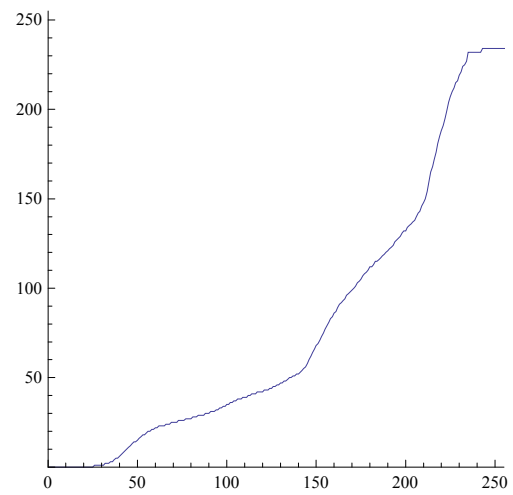
Print[Grid[{{Show[Histogram[Flatten[ImageData[skalarbildein, "Byte"]], {0, 255, 1}, "Count", PlotRange → {{0, 255}, Automatic},
  AxesOrigin → {0, 0}], ImageSize → 256], Show[Histogram[Flatten[ImageData[skalarbildref, "Byte"]], {0, 255, 1}, "Count",
  PlotRange → {{0, 255}, Automatic}, AxesOrigin → {0,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

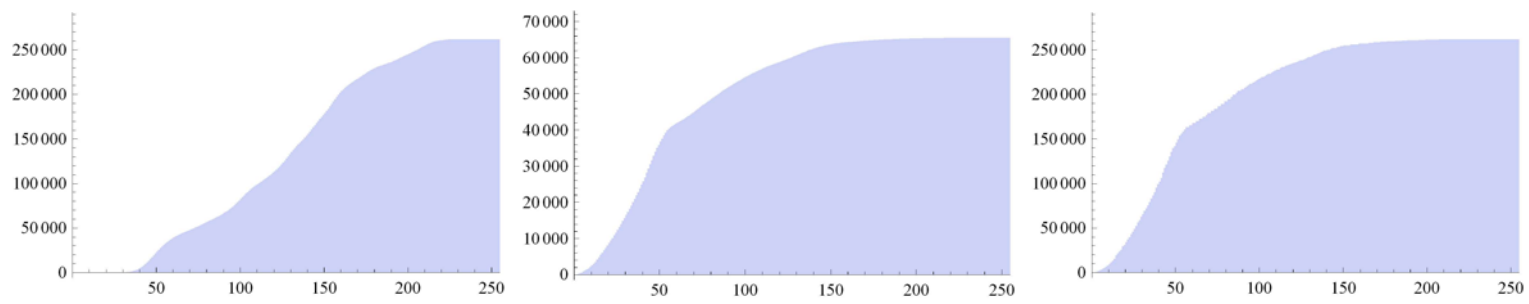
Print[Grid[{{Show[Histogram[Flatten[ImageData[skalarbildein, "Byte"]], {0, 255, 1}, "CumulativeCount", PlotRange → {{0, 255}, Automatic},
  AxesOrigin → {0, 0}], ImageSize → 256], Show[Histogram[Flatten[ImageData[skalarbildref, "Byte"]], {0, 255, 1}, "CumulativeCount",
  PlotRange → {{0, 255}, Automatic},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

```

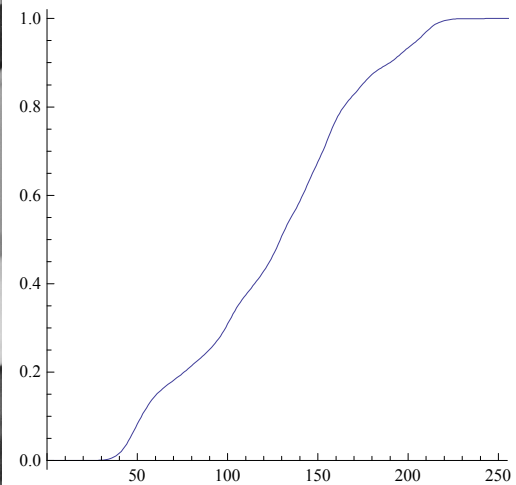
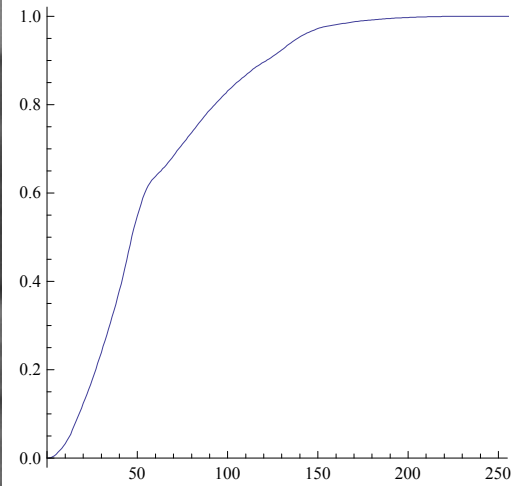
```
histogrammanpassung[ColorConvert[ExampleData[{"TestImage", "Lena"}], "Graylevel"],  
  ColorConvert[ExampleData[{"TestImage", "Girl"}], "Graylevel"]];
```

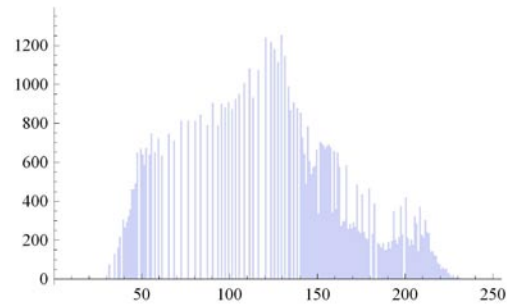
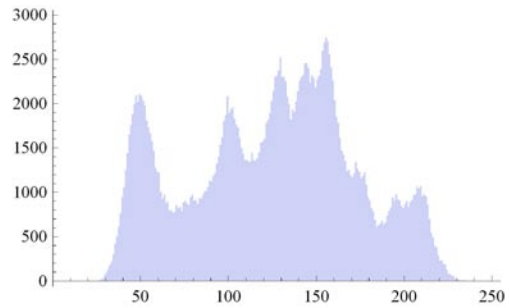
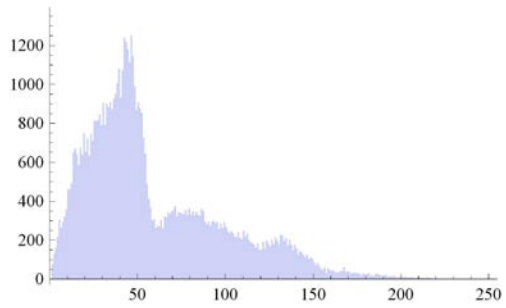
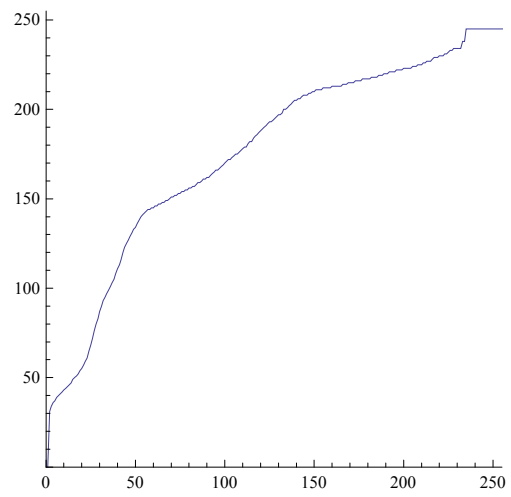


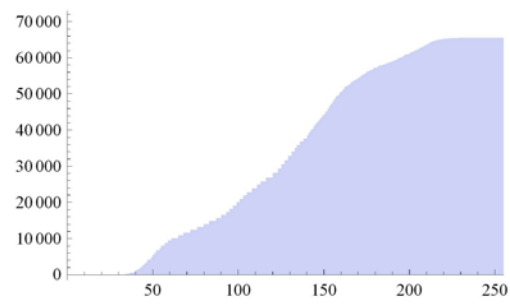
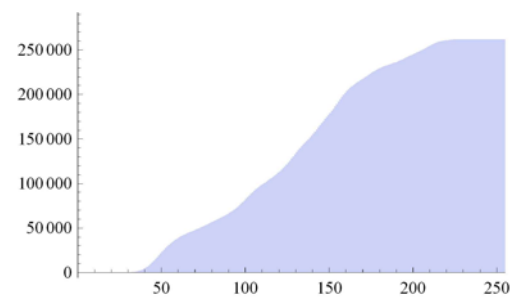
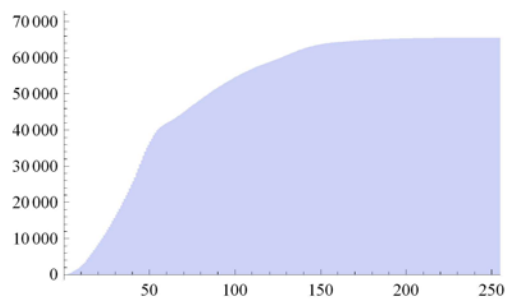




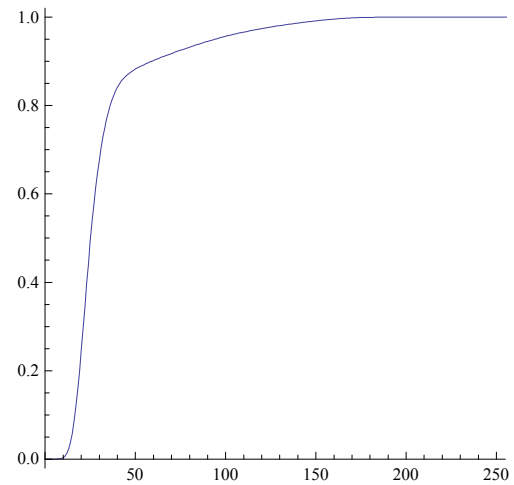
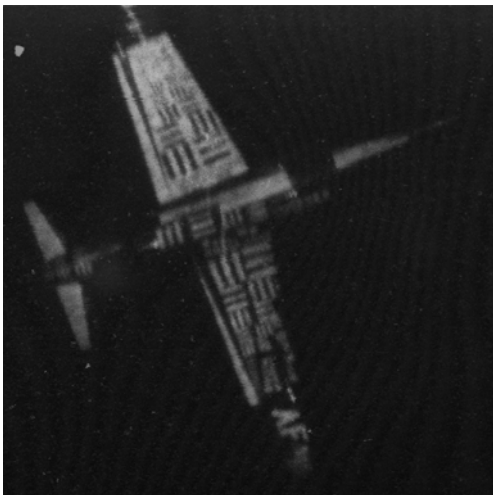
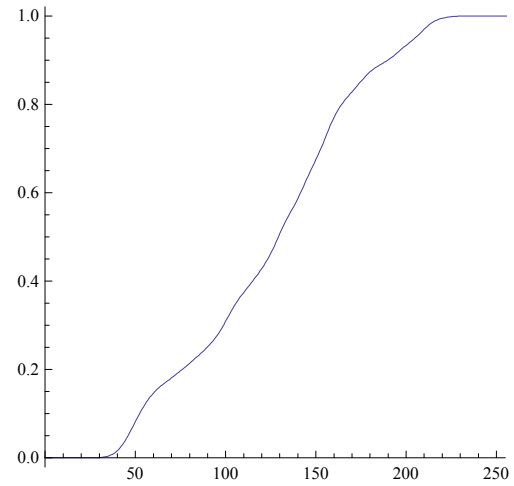
```
histogrammanpassung[ColorConvert[ExampleData[{"TestImage", "Girl"}], "Graylevel"],
  ColorConvert[ExampleData[{"TestImage", "Lena"}], "Graylevel"]];
```

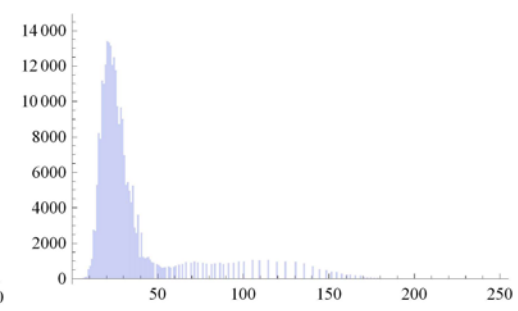
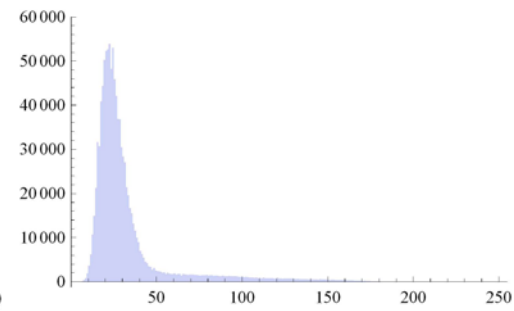
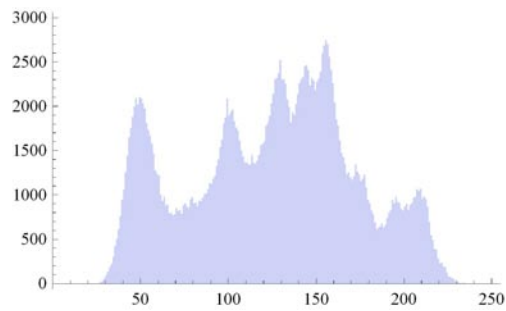
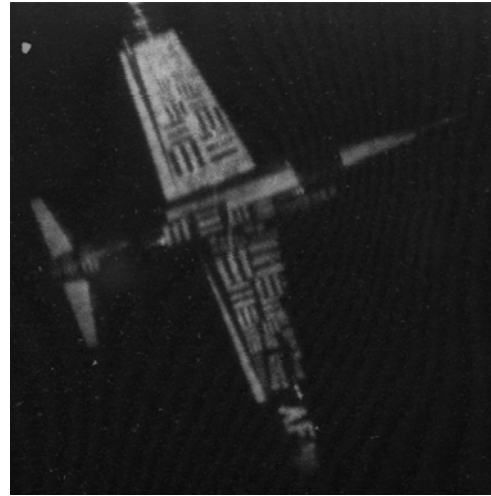
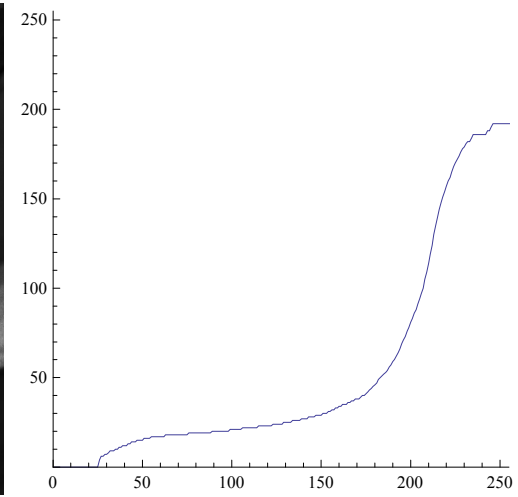


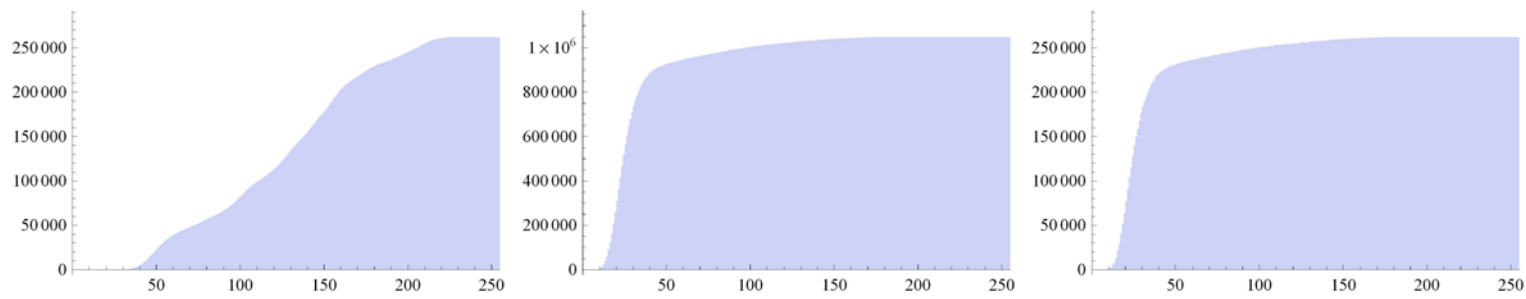




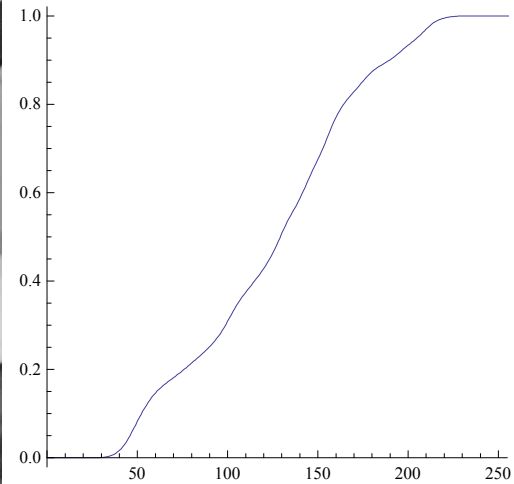
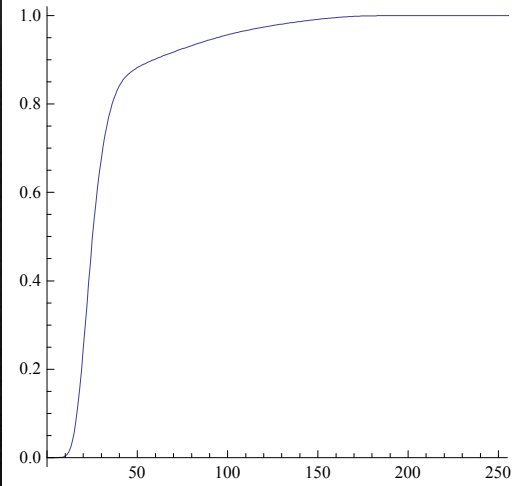
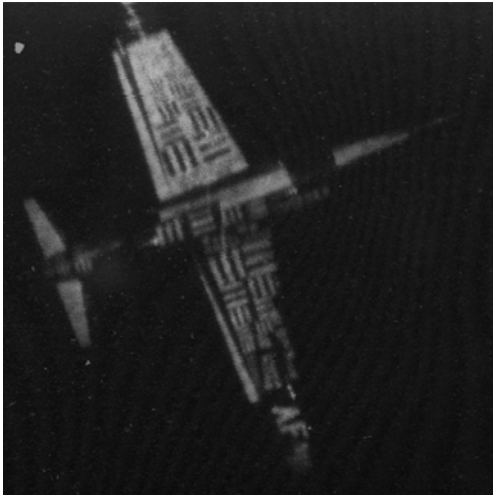
```
histogrammanpassung[ColorConvert[ExampleData[{"TestImage", "Lena"}], "Graylevel"], ExampleData[{"TestImage", "U2"}]];
```

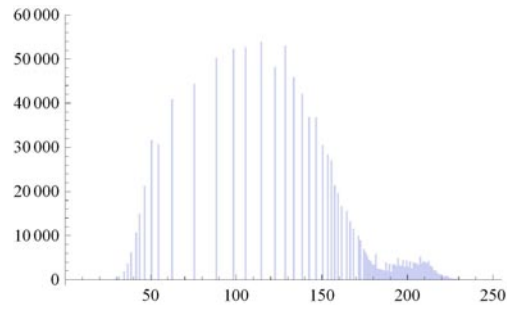
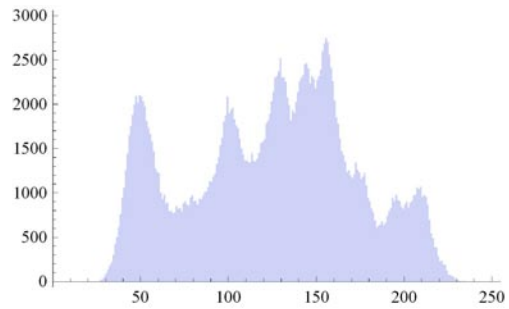
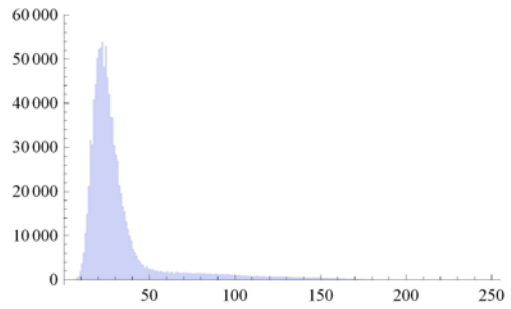
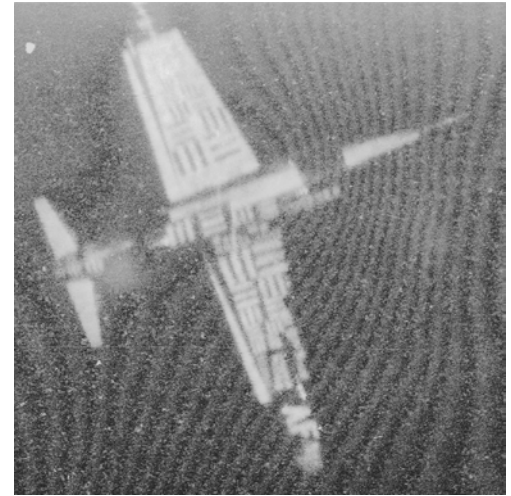
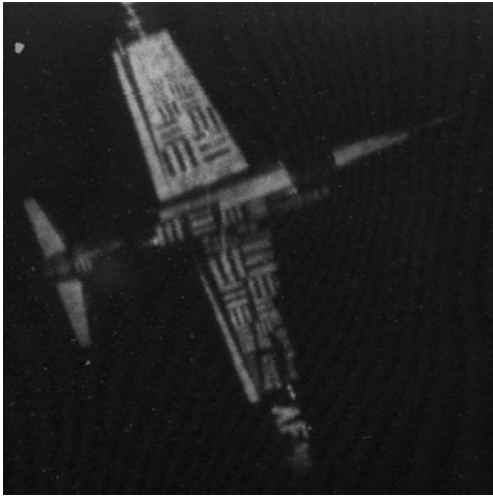
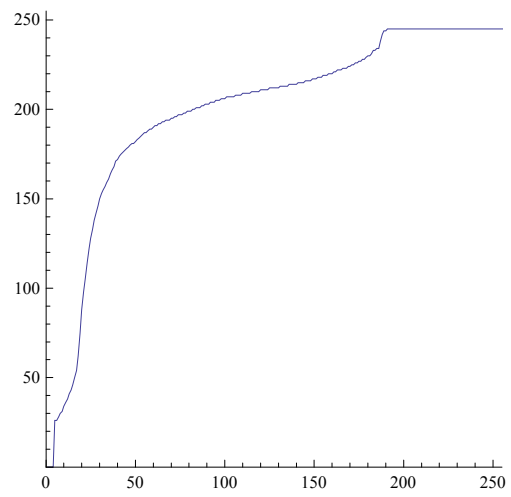
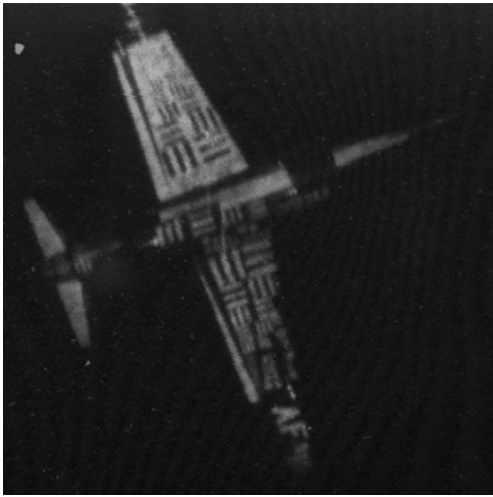


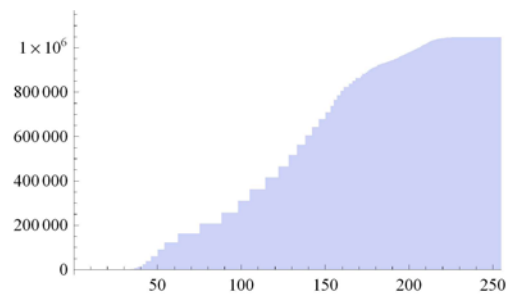
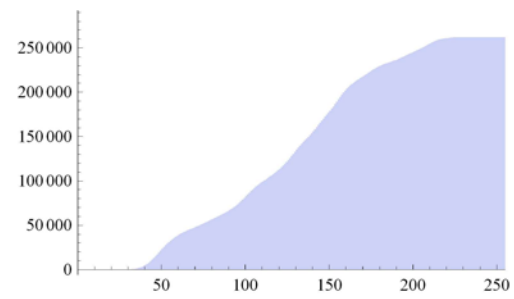
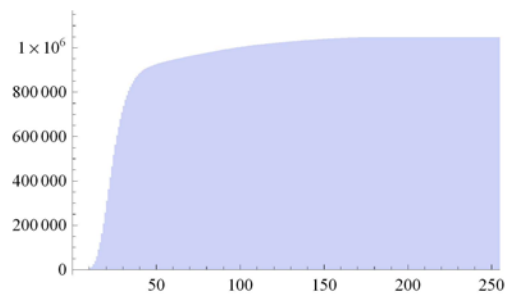




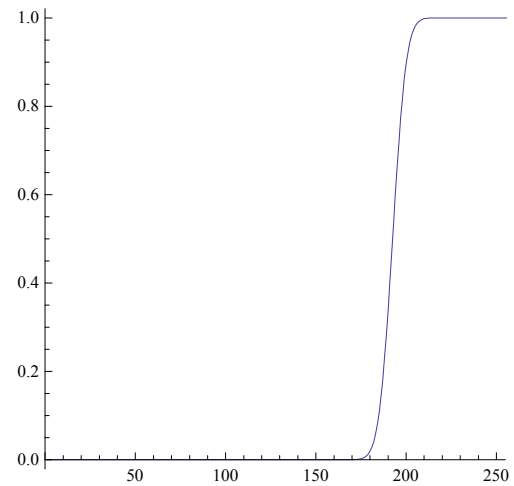
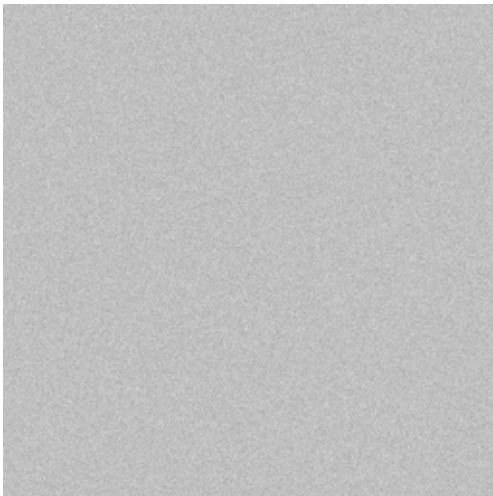
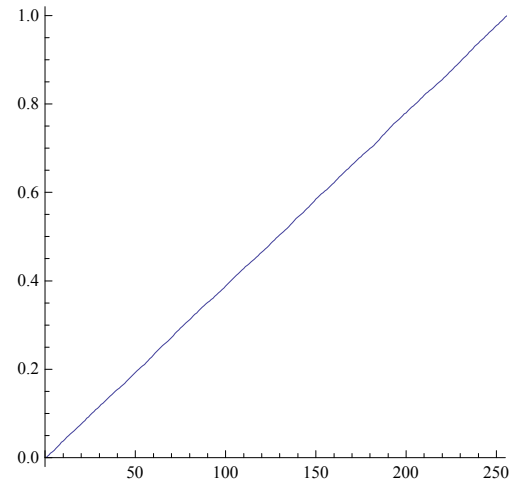
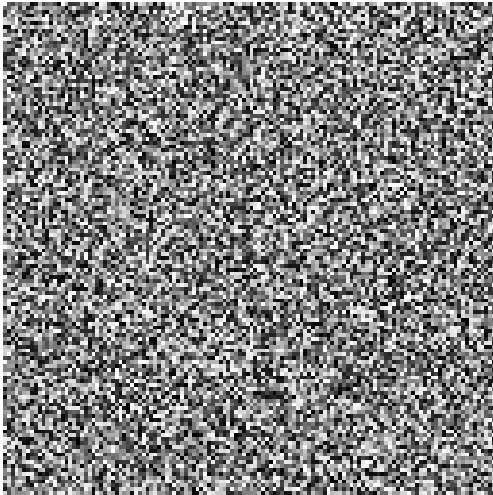
```
histogrammanpassung[ExampleData[{"TestImage", "U2"}], ColorConvert[ExampleData[{"TestImage", "Lena"}], "Graylevel"]];
```

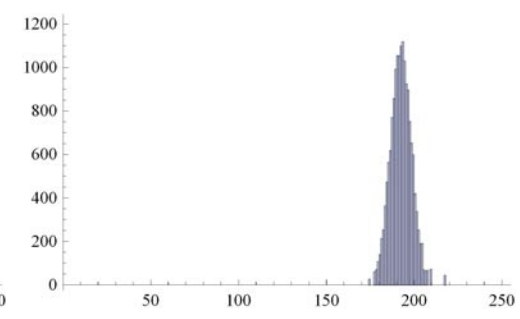
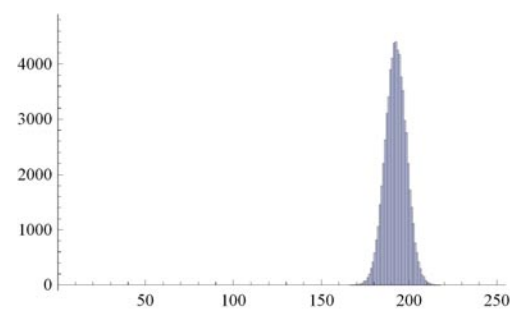
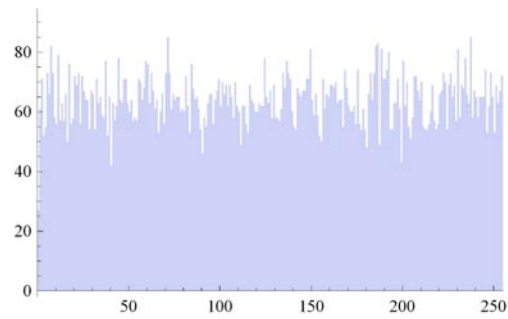
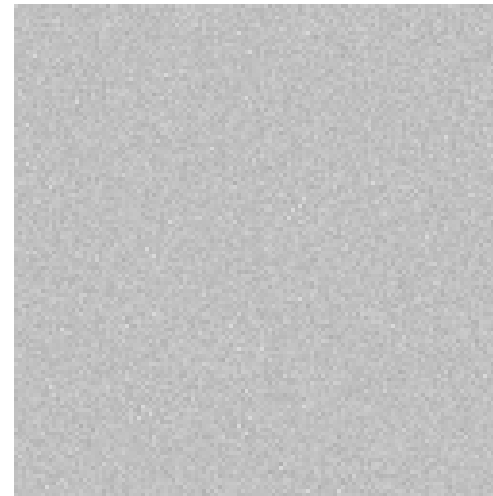
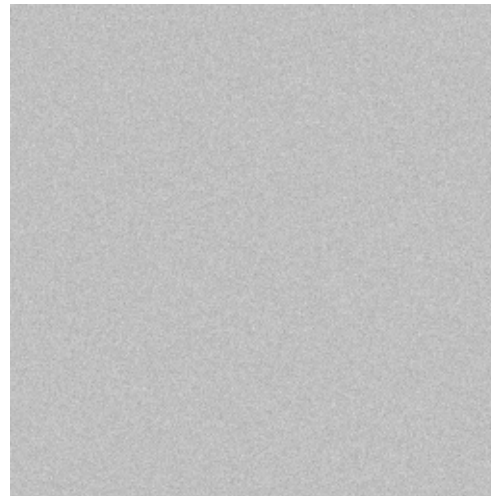
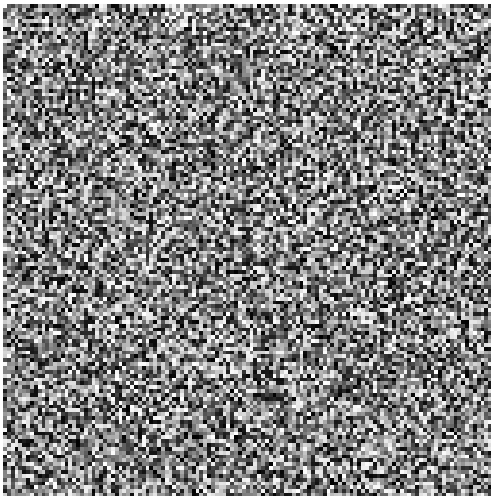
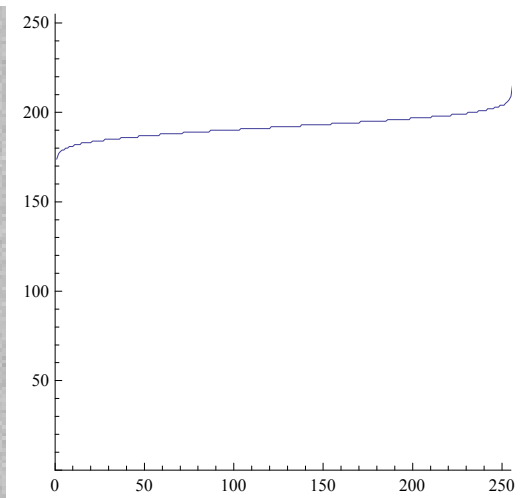
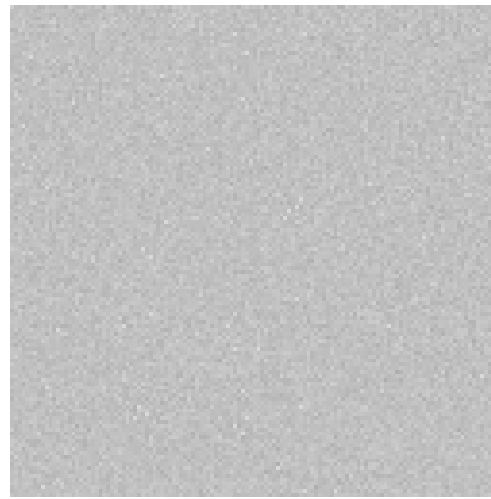
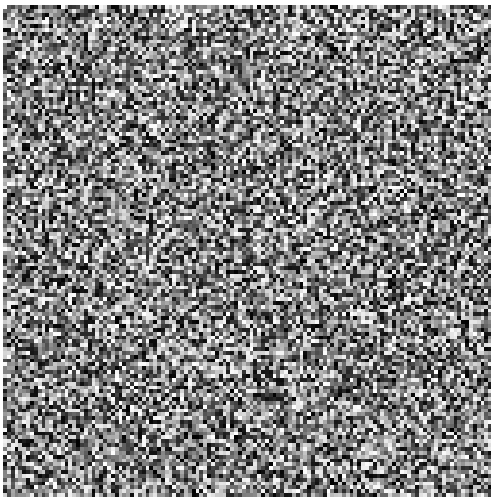


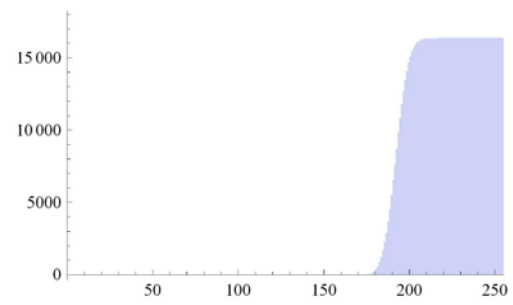
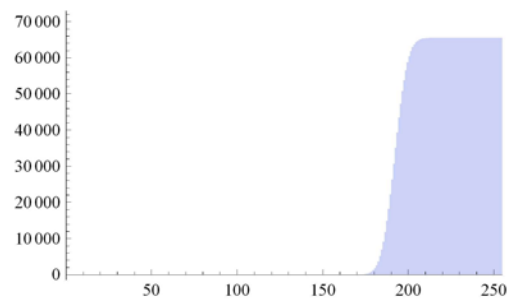
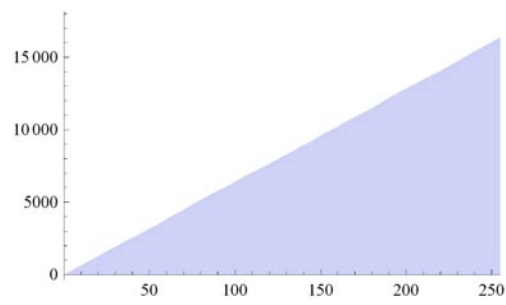




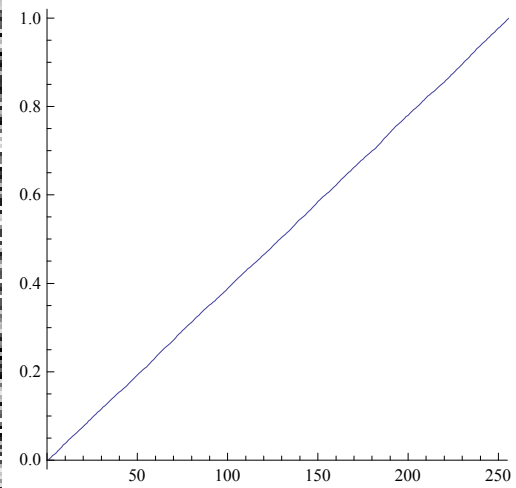
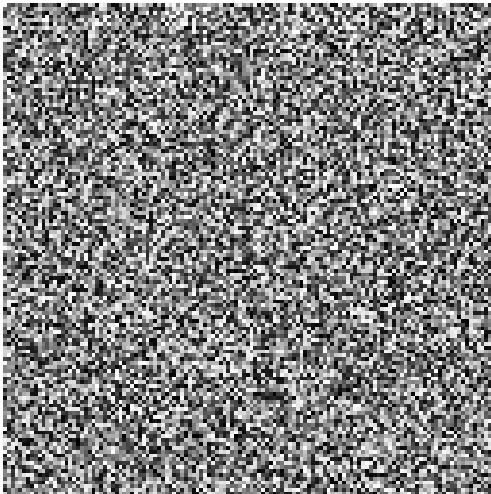
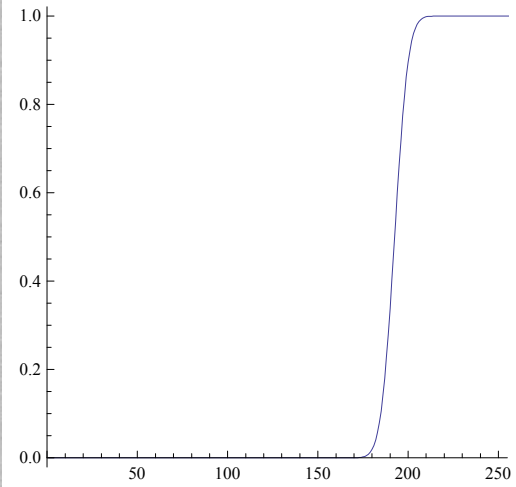
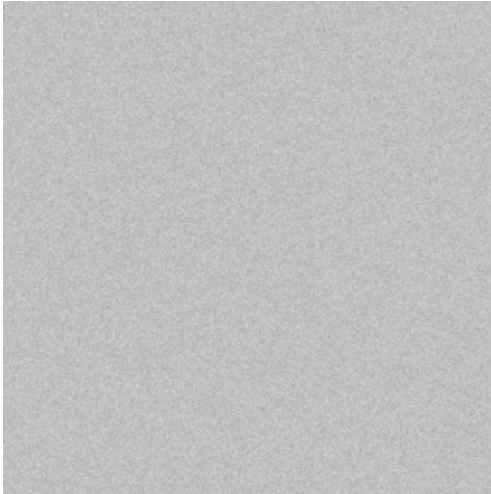
```
histogrammanpassung[randombild, rauschbild];
```

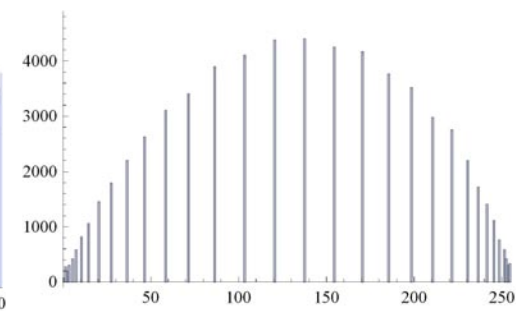
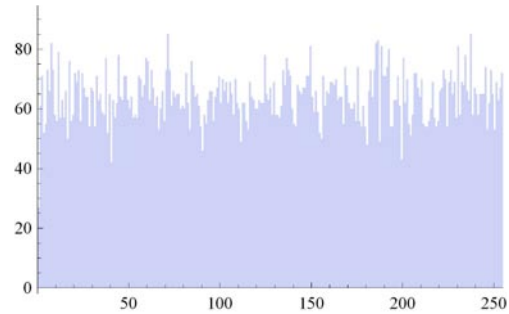
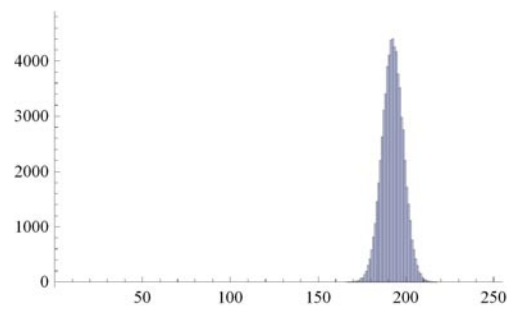
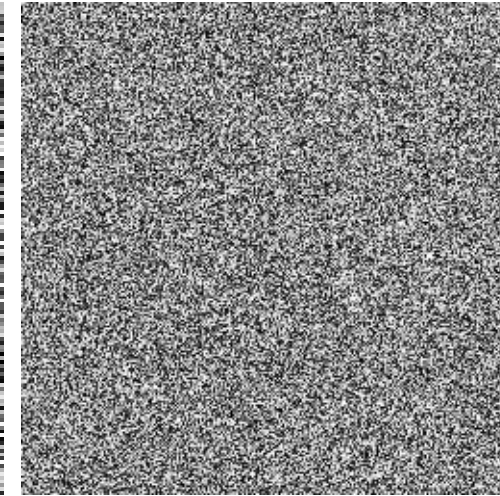
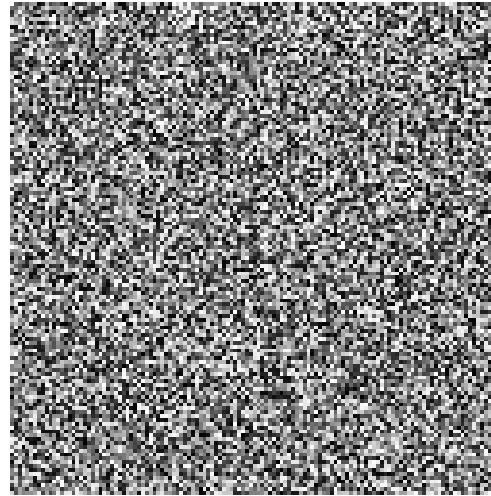
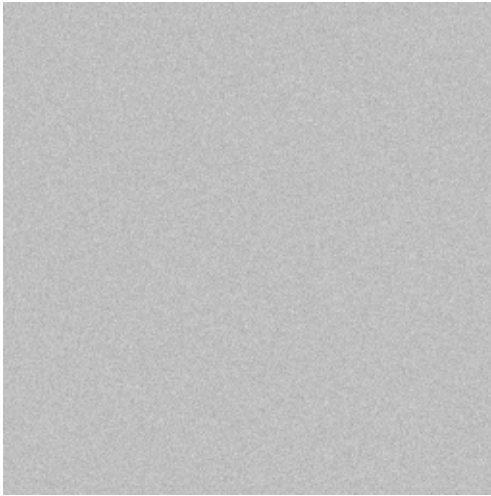
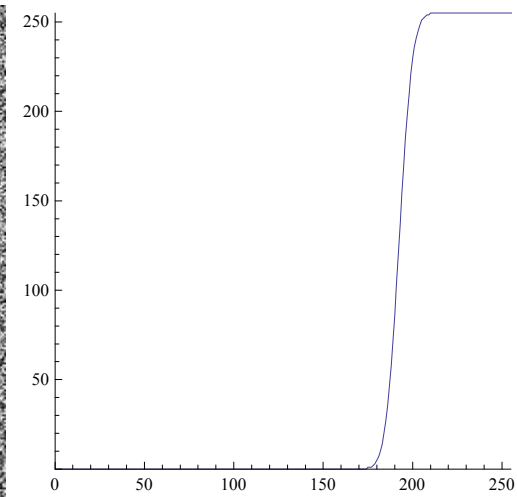
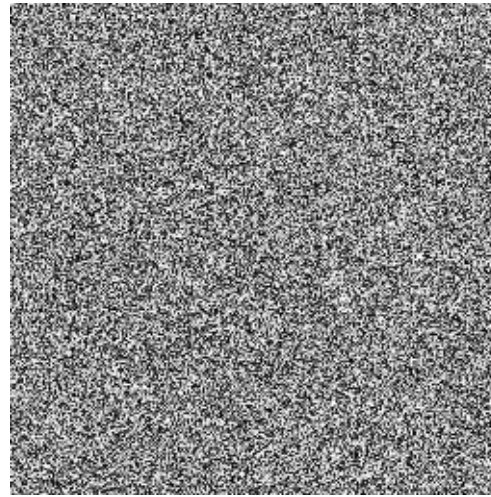
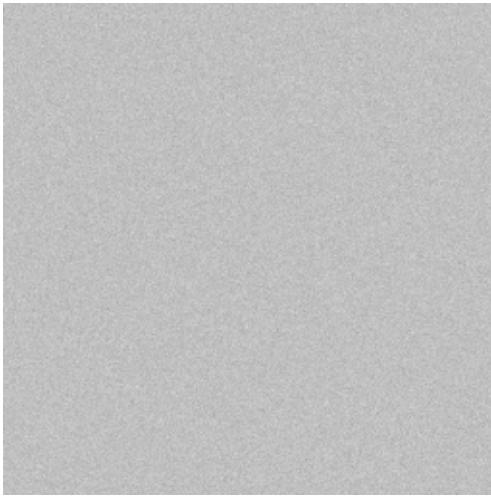


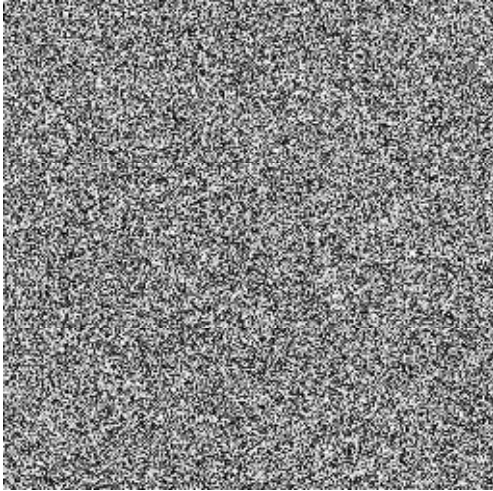
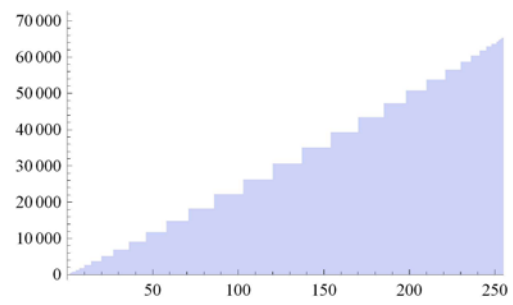
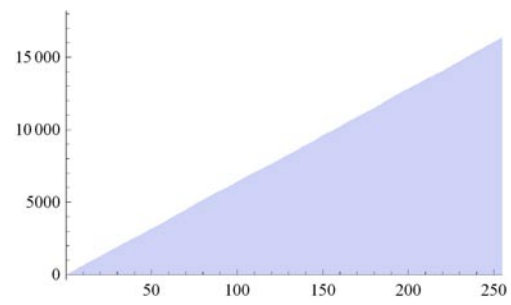
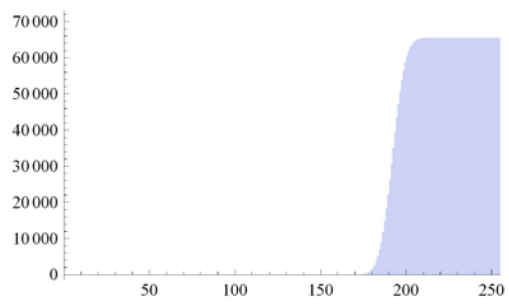




`histogrammanpassung[rauschbild, randombild]`







11. Graylevel-Co-Occurence-Matrix

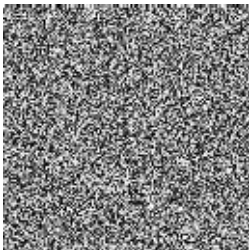
Als Hilfsmittel für die Analyse von textuellen (musterbezogenen) Bildeigenschaften, auch als Grauwertübergangsmatrix benannt

- anhand einer definierten binären Relation (“Nachbarschafts-Template”) werden geordnete Paare von Intensitätswerten erfaßt, z.B. zum jew. rechten Nachbarn
- zusammenhängend besetzte Hauptdiagonalen weisen auf homogene Bildbereiche hin
- Einträge weitab der Hauptdiagonalen deuten auf starke lokale Kontraste hin

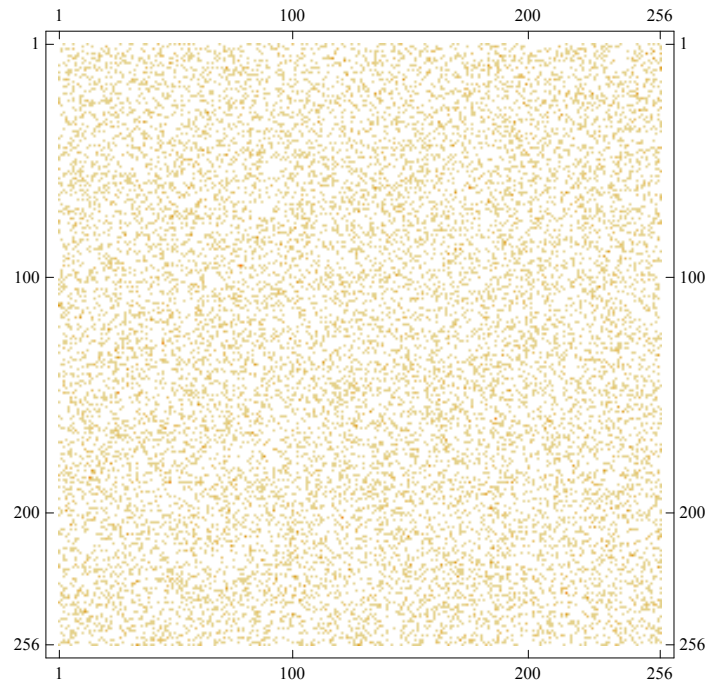
```
{{0, 0, 0}, {0, 0, 1}, {0, 0, 0}} // MatrixForm
```

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

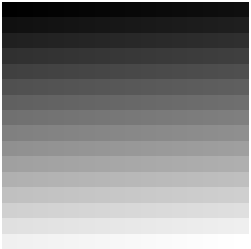
randombild



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



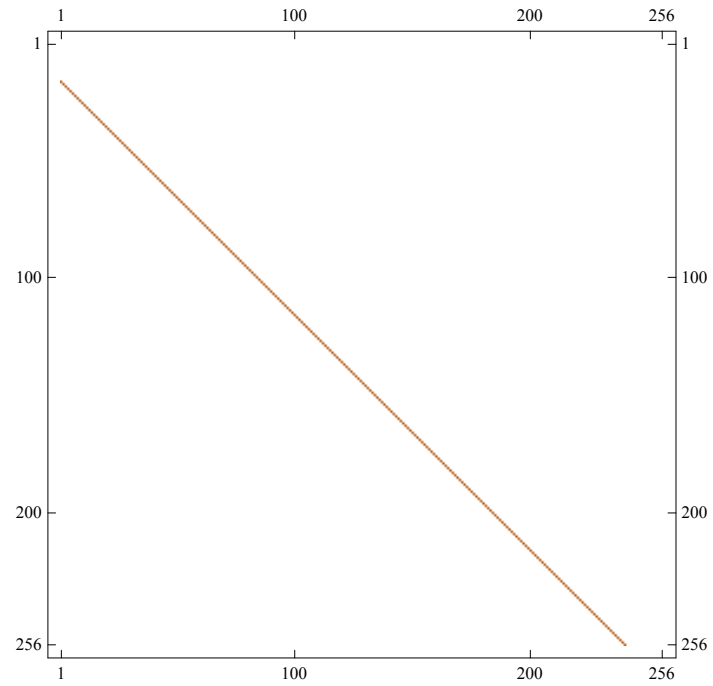
```
Show[Image[Partition[Range[0, 255], 16], "Byte"], ImageSize -> 128]
```



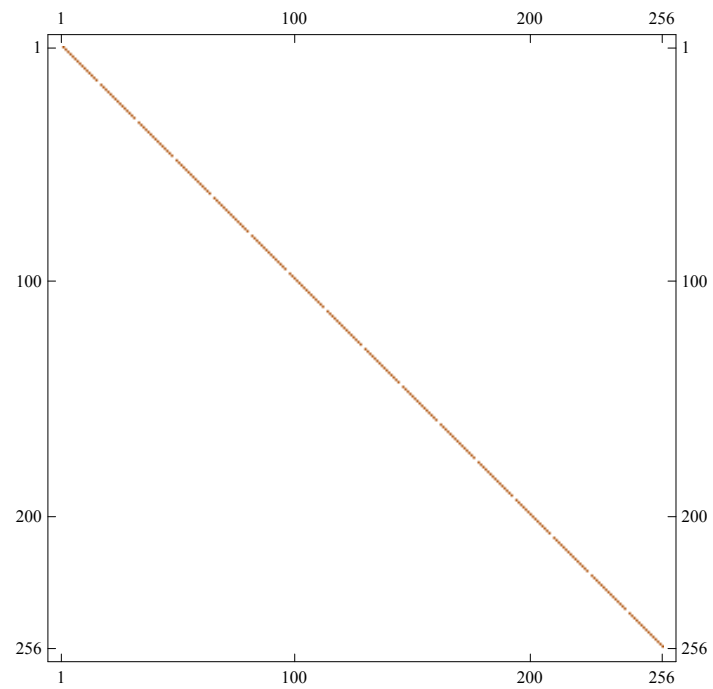
```
{{0, 1, 0}, {0, 0, 0}, {0, 0, 0}} // MatrixForm
```

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

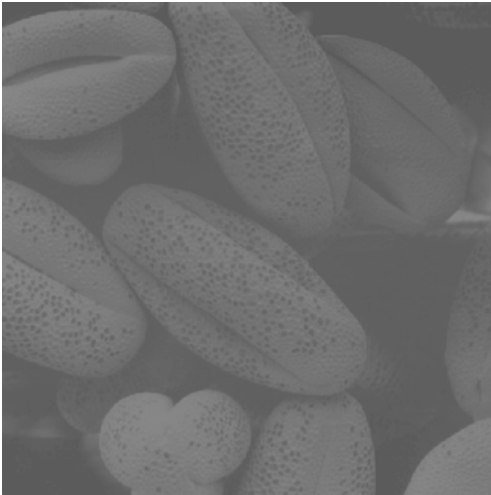
```
MatrixPlot[ImageCooccurrence[Image[Partition[Range[0, 255], 16], "Byte"], 256, {{0, 1, 0}, {0, 0, 0}, {0, 0, 0}}]]
```



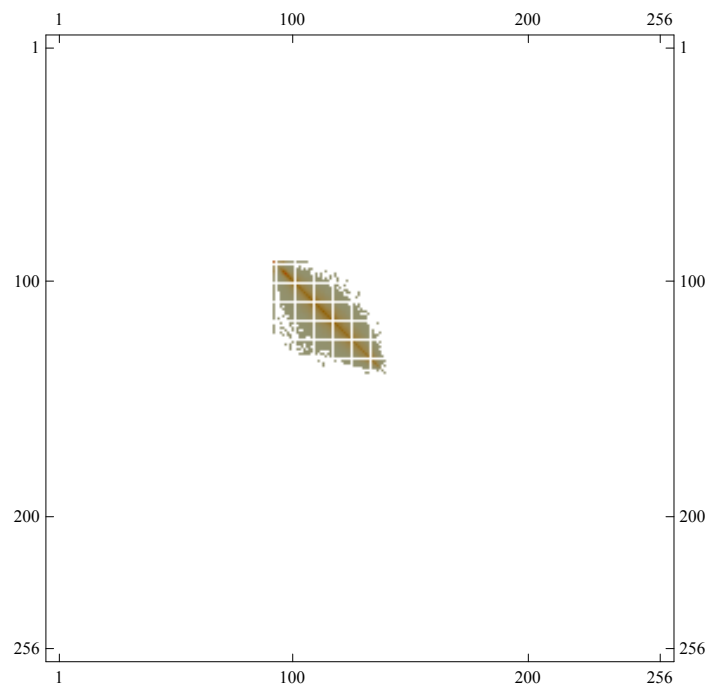
```
MatrixPlot[ImageCooccurrence[Image[Partition[Range[0, 255], 16], "Byte"], 256, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}]]
```



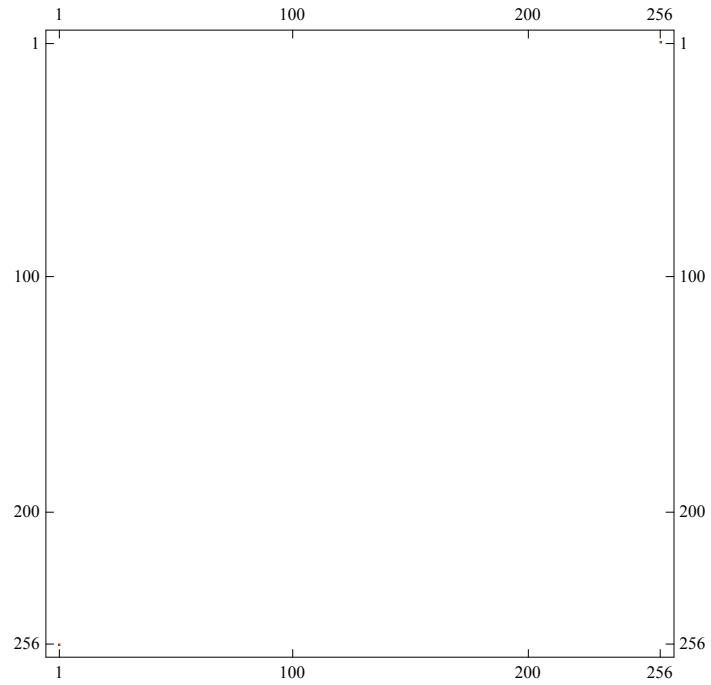
```
Show[pollen, ImageSize -> 256]
```



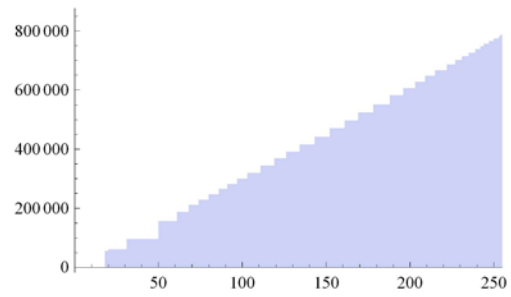
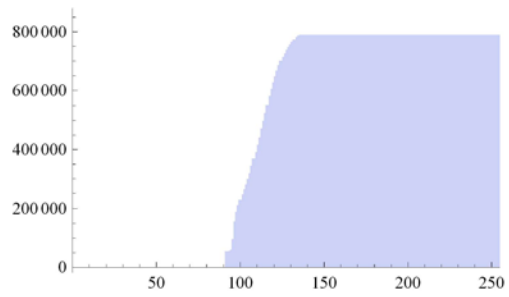
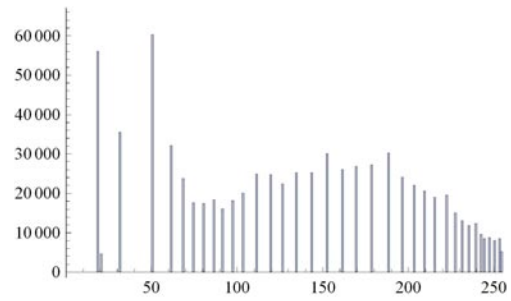
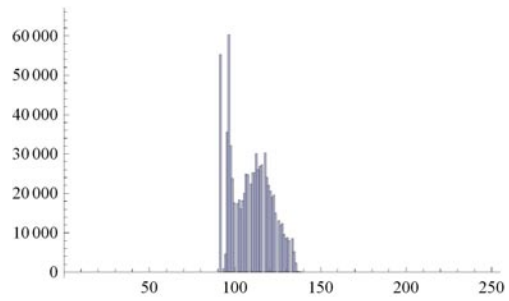
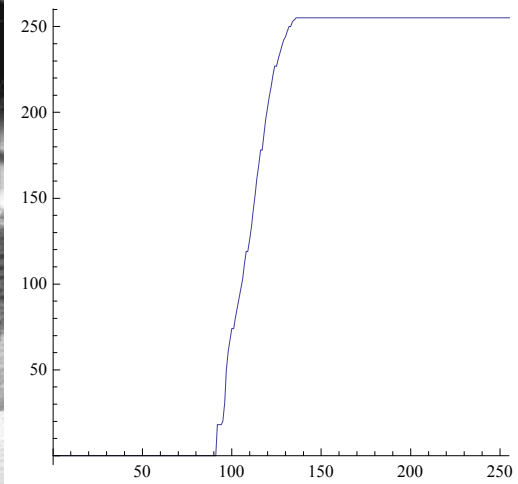
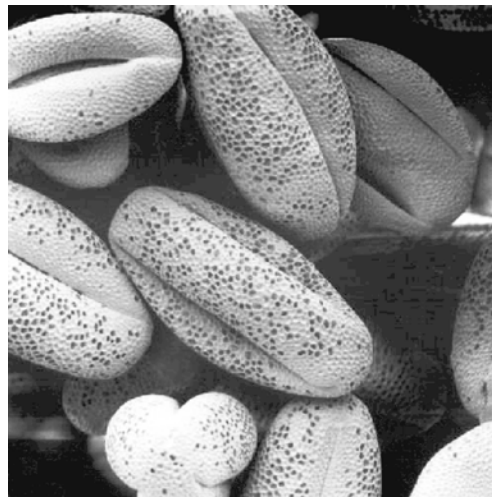
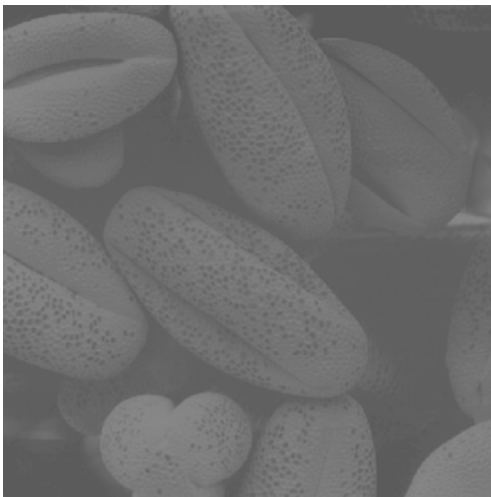
```
MatrixPlot[ImageCooccurrence[pollen, 256]]
```

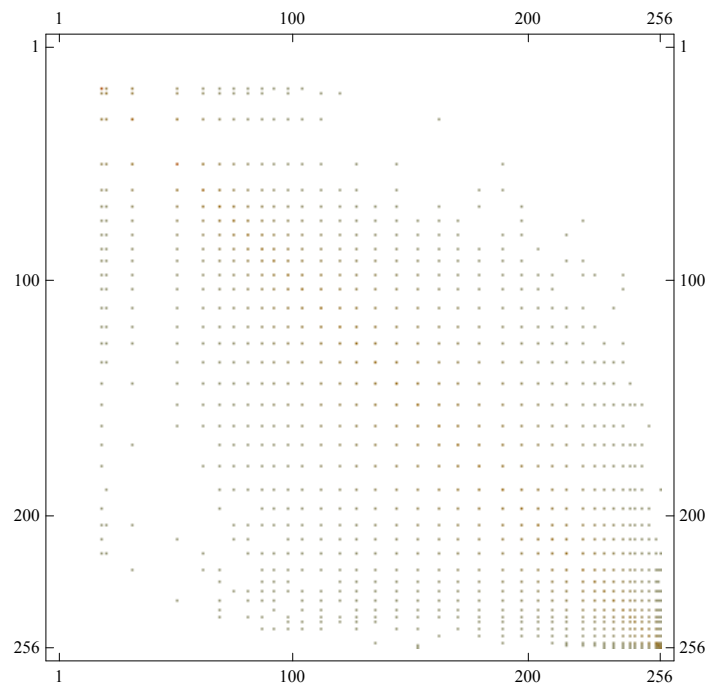


```
MatrixPlot[ImageCooccurrence[, 256]]
```

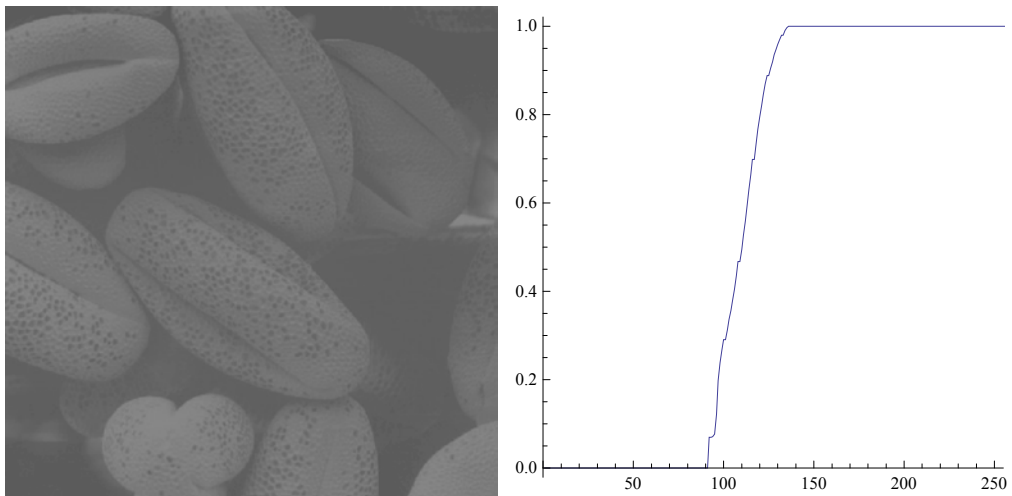


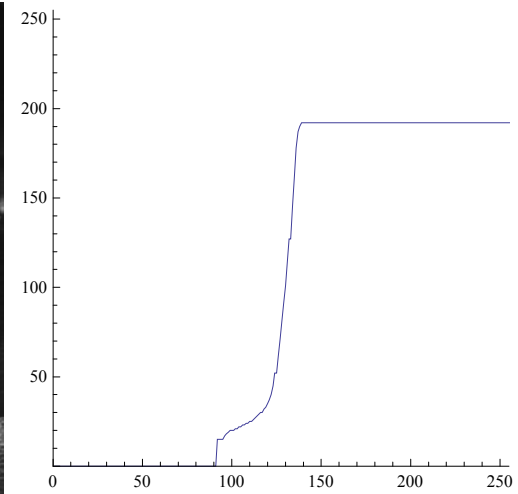
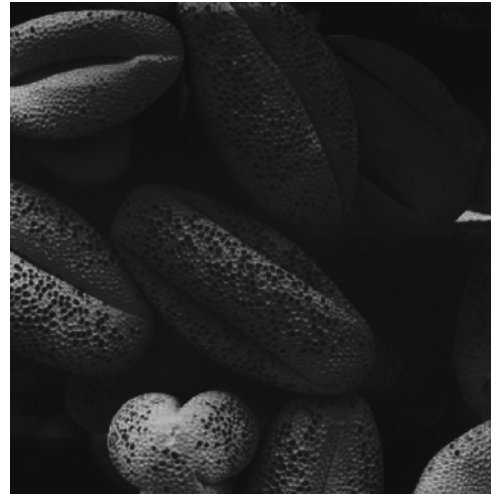
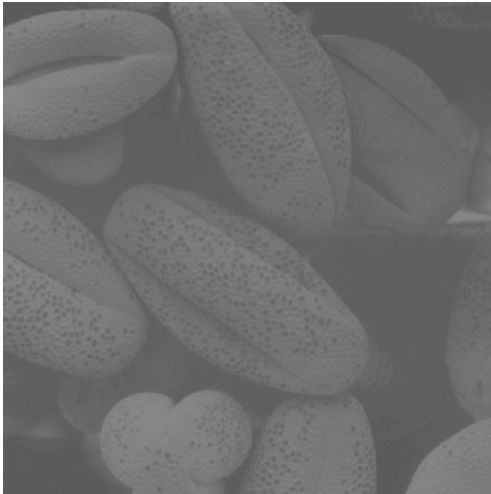
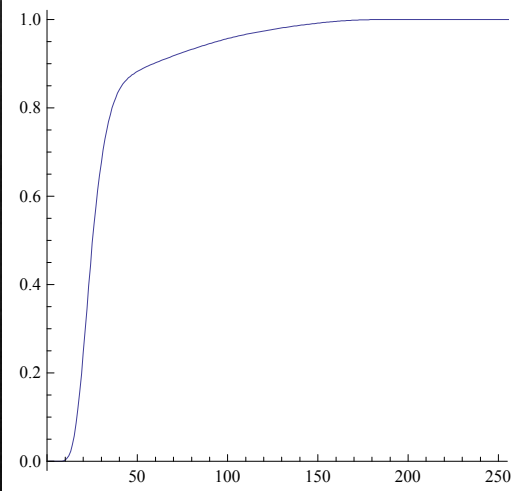
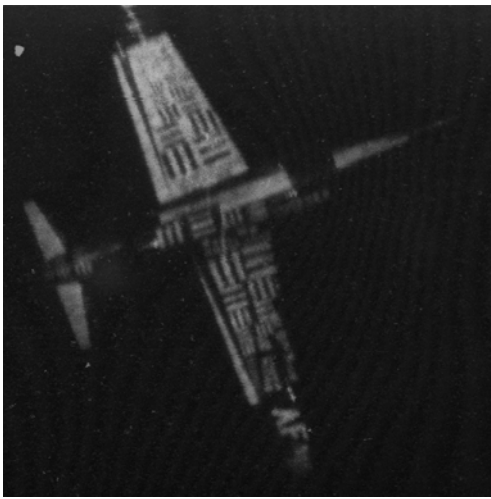
```
MatrixPlot[ImageCooccurrence[histogrammausgleich[pollen], 256]]
```

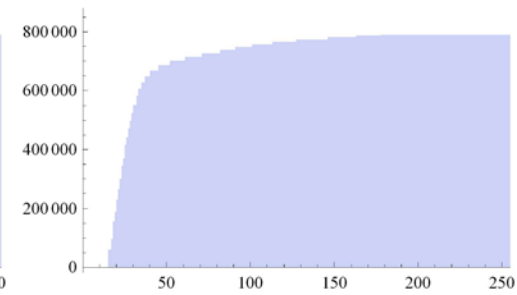
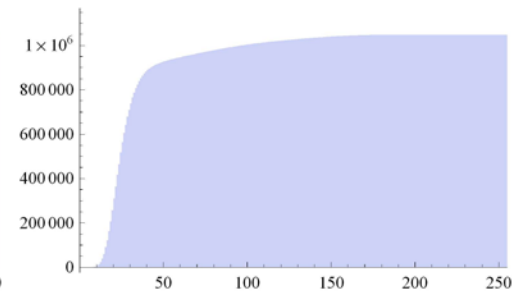
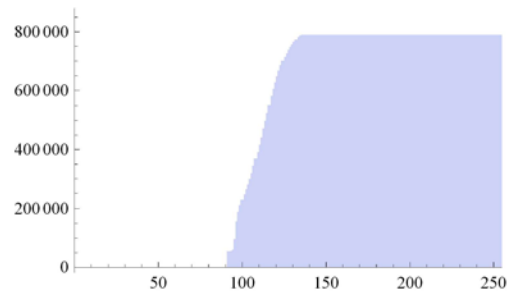
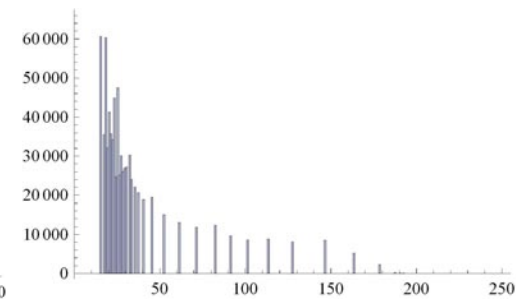
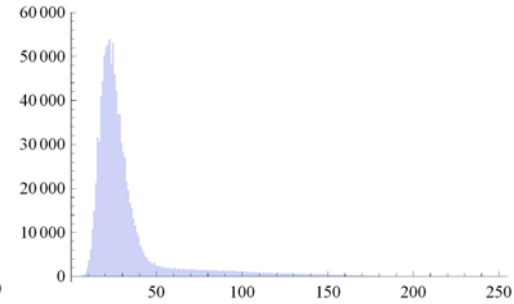
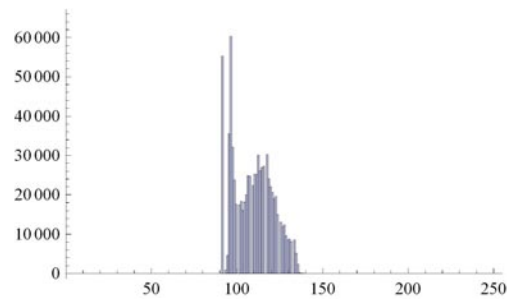
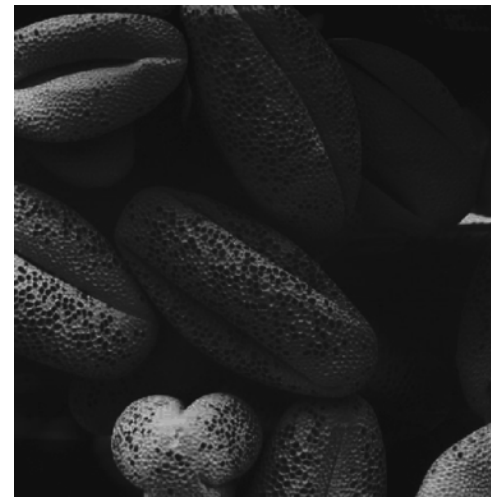
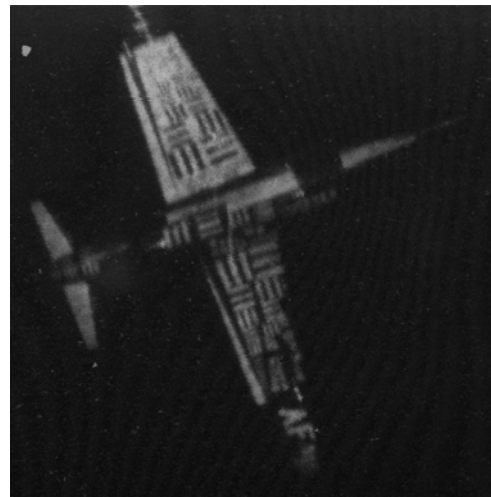
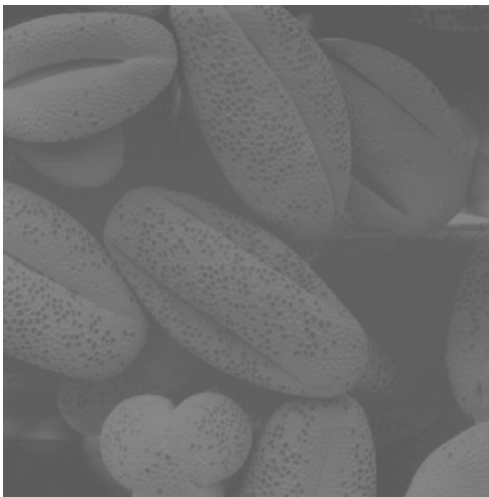


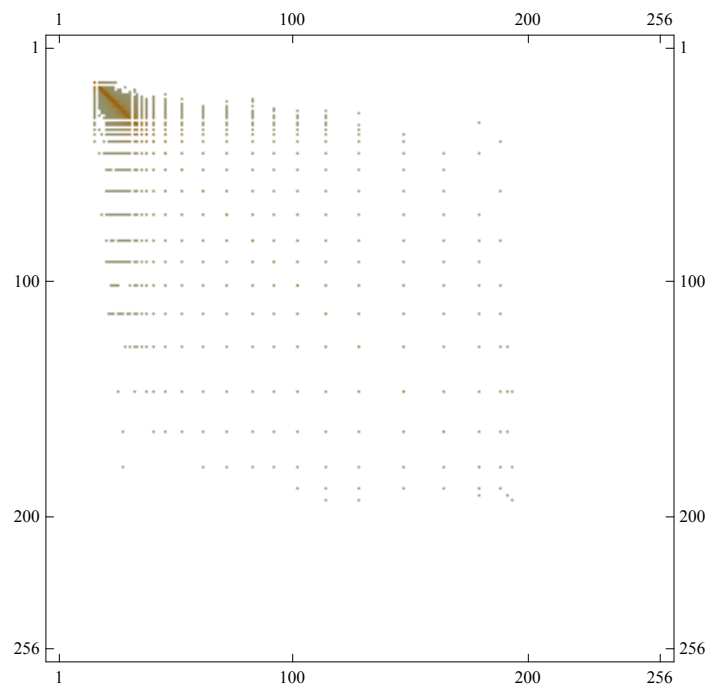


```
MatrixPlot[ImageCooccurrence[histogrammanpassung[pollen, ExampleData[{"TestImage", "U2"}]], 256]]
```

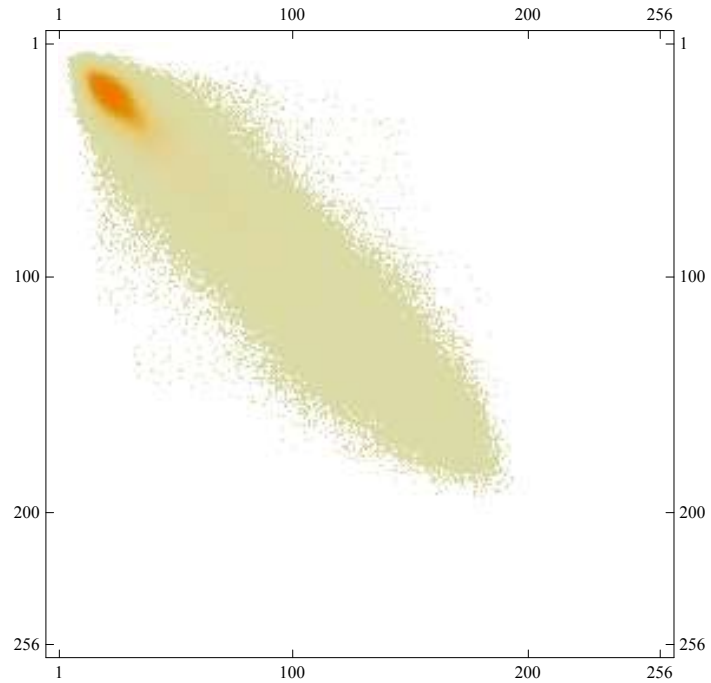




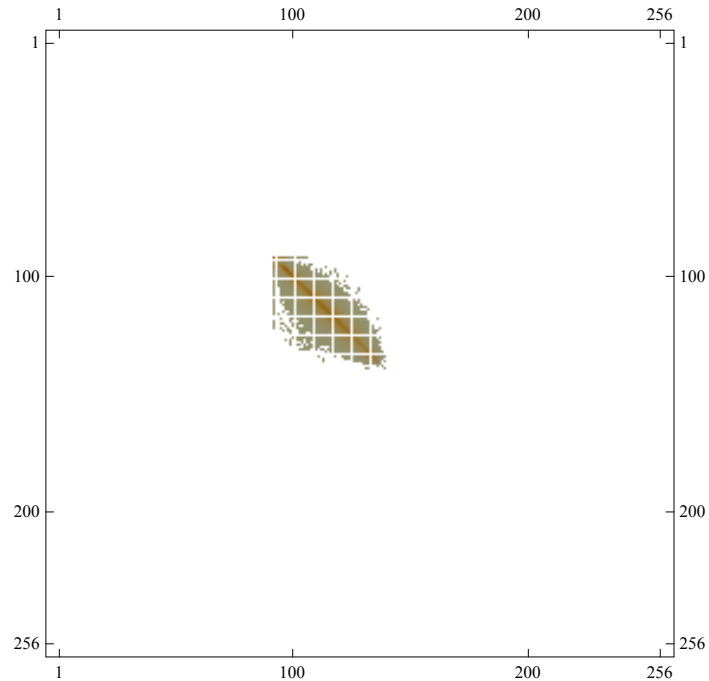




```
MatrixPlot[ImageCooccurrence[ExampleData[{"TestImage", "U2"}], 256]]
```



```
MatrixPlot[ImageCooccurrence[pollen, 256]]
```

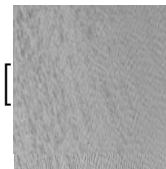


Analyse der Grauwertübergangsmatrix mittels Haralick'scher Merkmale (insgesamt 14)

Haralick-Merkmal 1: Energie anhand der Grauwertübergangsmatrix

```
Clear[haralickenergie];
haralickenergie[image_, ker_] := Total[Flatten[ImageCooccurrence[image, 256, ker]]^2];
```

```
{haralickenergie[#, {{0, 0, 1}}], haralickenergie[#, {{0}, {0}, {1}}]} &[
```



```
{0.00069469, 0.00155443}]
```

Haralick-Merkmal 2: Kontrast anhand der Grauwertübergangsmatrix

```
Clear[haralickcontrast];
haralickcontrast[image_] := Total[Table[(i - j)^2, {i, 256}, {j, 256}] * ImageCooccurrence[image, 256] / 255^2, 2];
```

$\left\{ \text{haralickcontrast} \left[\begin{array}{c} \text{Checkerboard Pattern} \end{array} \right], \text{haralickcontrast} \left[\begin{array}{c} \text{Solid Gray} \end{array} \right] \right\}$
 $\{1., 0.\}$

Haralick-Merkmal 3: Korrelation anhand der Grauwertübergangsmatrix

```

Clear[haralickcorrelation];
haralickcorrelation[image_, ker_] :=
Module[{miux, mt, m, miuy, stdx, stdy, cm = ImageCooccurrence[image, 256, ker], in = Range[256]}, {m, mt} = Total /@ {cm, Transpose[cm]};
{miuy, miux} = {m.in, mt.in};
{stdy, stdx} = {(in - miuy)^2 . m, (in - miux)^2 . mt};
(Total[KroneckerProduct[in, in] * cm, 2] - miux * miuy) / Sqrt[stdx * stdy]]

```

$\left(\{ \text{haralickcorrelation}[\#, \{\{0, 0, 1\}\}], \text{haralickcorrelation}[\#, \{\{0\}, \{0\}, \{1\}\}] \} \right) \& \left[\begin{array}{c} \text{Noise Image} \end{array} \right]$
 $\{0.795395, 0.852629\}$

$\left(\{ \text{haralickcorrelation}[\#, \{\{0, 0, 1\}\}], \text{haralickcorrelation}[\#, \{\{0\}, \{0\}, \{1\}\}] \} \right) \& \left[\begin{array}{c} \text{Vertical Stripes Image} \end{array} \right]$
 $\{0.648159, 0.938001\}$

lenay

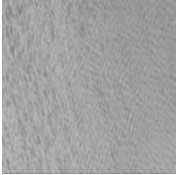


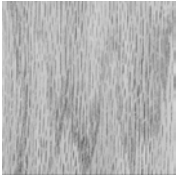
$\left(\{ \text{haralickcorrelation}[\#, \{\{0, 0, 1\}\}], \text{haralickcorrelation}[\#, \{\{0\}, \{0\}, \{1\}\}] \} \right) \&[\text{lenay}]$
 $\{0.898423, 0.949308\}$

```
({haralickcorrelation[#, {{0, 0, 1}}], haralickcorrelation[#, {{0}, {0}, {1}}]}) &[randombild]
{-0.0124227, -0.0084769}
```

Haralick-Merkmal 4: Varianz anhand der Grauwertübergangsmatrix

```
Clear[haralickvarianz];
haralickvarianz[image_, ker_] :=
Module[{miux, miuy, m, mt, cm = ImageCooccurrence[image, 256, ker], in = Range[256], c = ConstantArray[1, 256]},
{m, mt} = Total /@ {cm, Transpose[cm]};
{miuy, miux} = {m.in, mt.in};
(*Total[(in-miux*miuy)^2.cm, 2] *)
Total[(in * in - 2 * in * Sqrt[miux * miuy] + miux * miuy) * cm, 2]
]
```

```
({haralickvarianz[#, {{0, 0, 1}}], haralickvarianz[#, {{0}, {0}, {1}}]}) &[]
{258.99, 208.309}
```

```
({haralickvarianz[#, {{0, 0, 1}}], haralickvarianz[#, {{0}, {0}, {1}}]}) &[]
{760.087, 699.303}
```

```
({haralickvarianz[#, {{0, 0, 1}}], haralickvarianz[#, {{0}, {0}, {1}}]}) &[lenay]
{3845.68, 3845.52}
```

```
({haralickvarianz[#, {{0, 0, 1}}], haralickvarianz[#, {{0}, {0}, {1}}]}) &[randombild]
{5485.03, 5493.33}
```

Haralick-Merkmal 9: Entropie anhand der Grauwertübergangsmatrix

```

Clear[haralickentropie];
haralickentropie[image_, ker_] := Module[{m = ImageCooccurrence[image, 256, ker], h, p, d, ent},
  h = Tally[Flatten[m]];
  d = h[[All, 1]];
  p = h[[All, 2]];

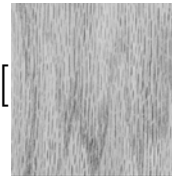
  (*Verhindern, dass Log[x] mit Argument 0 aufgerufen wird*)
  i = Position[d, 0];
  If[i != {},
    p[[i[[1]]]] = 0;
    d[[i[[1]]]] = 1;
  ];
  ent = -Plus @@ (p * d * Log[2, d]);
  N[ent]
]

```

```

({haralickentropie[#, {{0, 0, 1}}], haralickentropie[#, {{0}, {0}, {1}}]}) &[

```



```

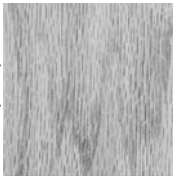
{12.1658, 10.5594}

```

```

haralickentropie[

```



```

, {{1, 1, 1}, {1, 0, 1}, {1, 1, 1}}]

```

```

12.033

```

```

({haralickentropie[#, {{0, 0, 1}}], haralickentropie[#, {{0}, {0}, {1}}]}) &[phasenkontrast]

```

```

{9.17759, 9.20004}

```

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

Haralick Feature 1 - Angular Second Momentum

```
Clear[haralick1];
haralick1[image_, steps_, ker_] := Total[Flatten[ImageCooccurrence[image, steps, ker]]^2];
```

Haralick Feature 2 - Contrast

```
Clear[haralick2];
haralick2[image_, steps_, ker_] :=
  Total[Table[(i - j)^2, {i, steps}, {j, steps}] * ImageCooccurrence[image, steps, ker] / (steps - 1)^2, 2];
```

Haralick Feature 3 - Correlation

```
Clear[haralick3];
haralick3[image_, steps_, ker_] :=
  Module[{miux, mt, m, miuy, stdx, stdy, cm = ImageCooccurrence[image, steps, ker], in = Range[steps]}, {m, mt} = Total /@ {cm, Transpose[cm]};
  {miuy, miux} = {m.in, mt.in};
  {stdy, stdx} = {(in - miuy)^2 . m, (in - miux)^2 . mt};
  (Total[KroneckerProduct[in, in] * cm, 2] - miux * miuy) / Sqrt[stdx * stdy]]
```

Haralick Feature 4 - Variance

```
Clear[haralick4];
haralick4[image_, steps_, ker_] :=
  Module[{miux, miuy, m, mt, cm = ImageCooccurrence[image, steps, ker], in = Range[steps], c = ConstantArray[1, steps]},
  {m, mt} = Total /@ {cm, Transpose[cm]};
  {miuy, miux} = {m.in, mt.in};
  (*Total[(in - miux * miuy)^2 . cm, 2] *)
  Total[(in * in - 2 * in * Sqrt[miux * miuy] + miux * miuy) * cm, 2]
  ]
```

Haralick Feature 9 - Entropie

```

Clear[haralick9];
haralick9[image_, steps_, ker_] := Module[{m = ImageCooccurrence[image, steps, ker], h, p, d, ent},
  h = Tally[Flatten[m]];
  d = h[[All, 1]];
  p = h[[All, 2]];

  (*Verhindern, dass Log[x] mit Argument 0 aufgerufen wird*)
  i = Position[d, 0];
  If[i != {},
    p[[i[[1]]]] = 0;
    d[[i[[1]]]] = 1;
  ];
  (*ent = -Total[p*d*Log[2,d]];*)
  ent = -Plus @@ (p*d*Log[2, d]);
  N[ent]
  (*N[Entropy[Flatten[m]]/Log[2]]*)
  (*-Sum[Limit[p*Log[2,p],p->m[[i,j]]],{i,1,256},{j,1,256}]*)
]

```

Manipulate - Umgebung

```

Manipulate[
  Switch[feat,
    1, {fimg1, fimg2} = {haralick1[img1, graylevelsteps, nbh], haralick1[img2, graylevelsteps, nbh]};
    label = "angular second momentum = ";
    ,
    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:  89

Bild 1:

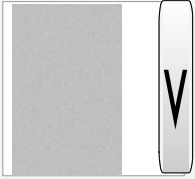
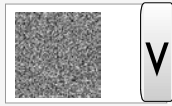


Bild 2:



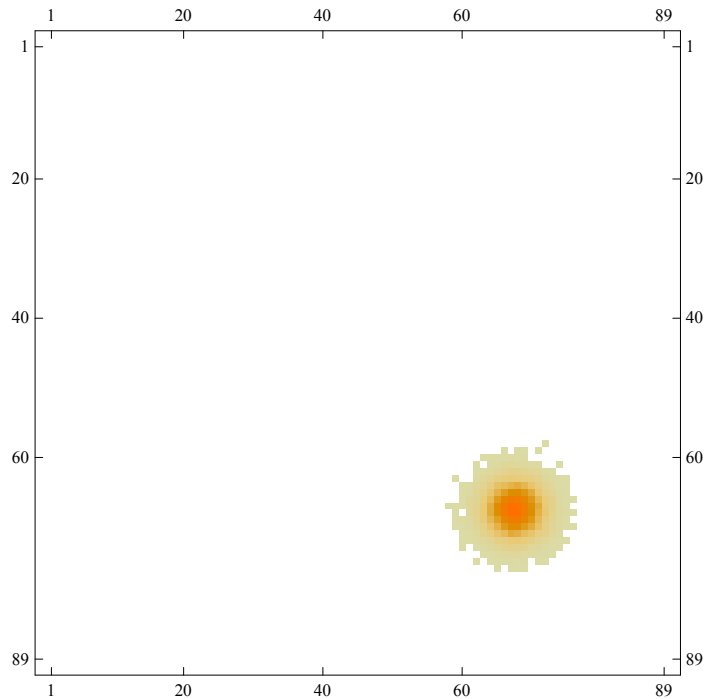
3x3-Nachbarschaft:

rechts

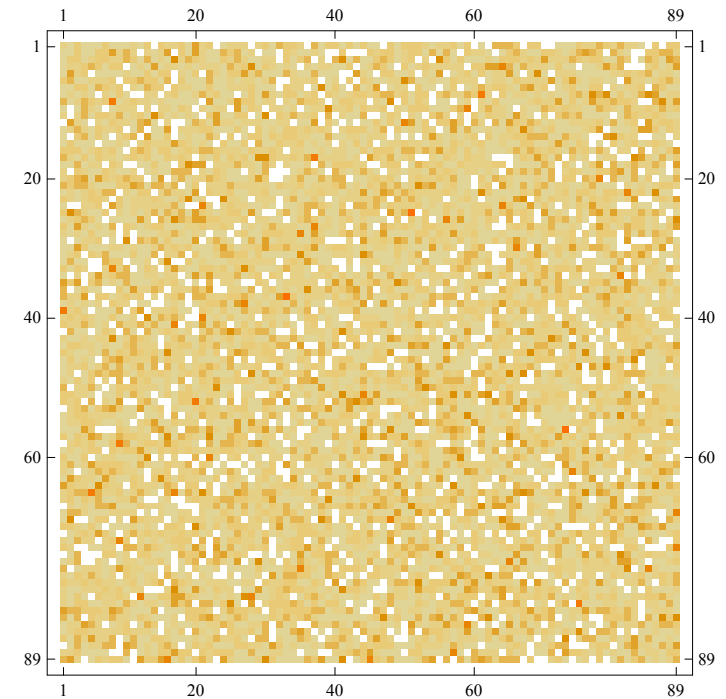


Haralick-Feature:

entropy



```
entropy = haralick9[-Image-,  
    89, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}]
```



```
entropy = haralick9[-Image-,  
    89, {{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}]
```

12. Lineare diskrete verschiebungsinvariante Filter

Aufgaben:

- Verminderung von Bildrauschen
- Detektion einfacher Strukturen (z.B. Kanten)

Implementation:

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

Eigenschaften:

- Linearität (Superpositionsprinzip): $O(aG + bG') = aOG + bOG'$
- Additivität (Distrib.): $O_1 G + O_2 G = (O_1 + O_2)G$
- Kommutativität: $O_1 O_2 G = O_2 O_1 G$
- Assoziativität: $O_1(O_2 G) = (O_1 O_2) G = OG$
- Verschiebungsinvarianz: wenn Signal verschoben, dann ist Ergebnis des Operators ebenso verschoben

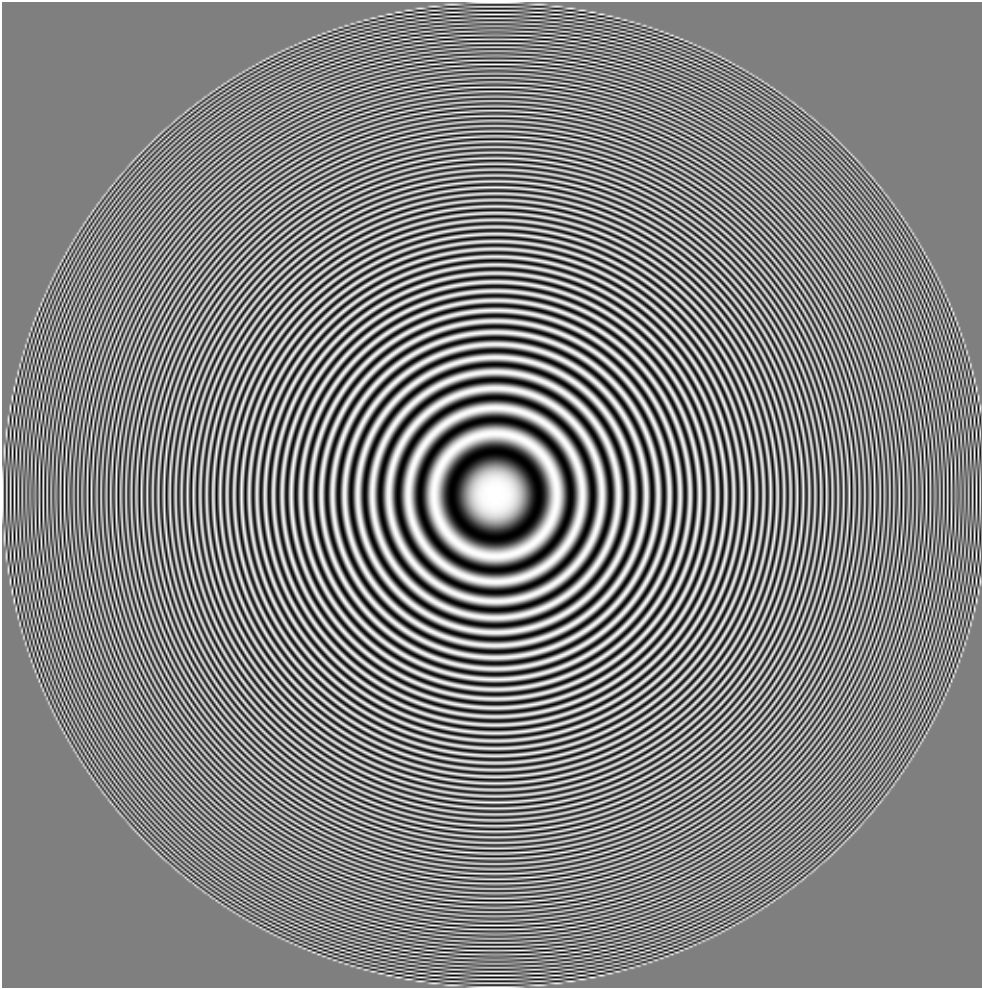
```
ListConvolve[{a, b, c}, {1, 2, 1, 3, 1, 3}, 2]
{2 a + b + 3 c, a + 2 b + c, 3 a + b + 2 c, a + 3 b + c, 3 a + b + 3 c, a + 3 b + c}
```

```
ListCorrelate[{a, b, c}, {1, 2, 1, 3, 1, 3}, 2]
{3 a + b + 2 c, a + 2 b + c, 2 a + b + 3 c, a + 3 b + c, 3 a + b + 3 c, a + 3 b + c}
```

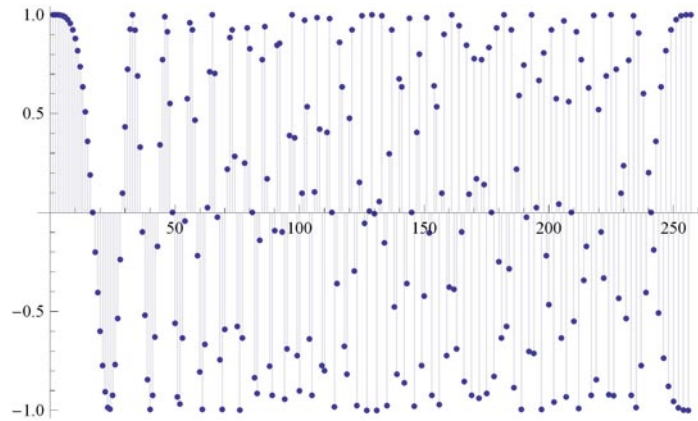
```
ListCorrelate[Reverse[{a, b, c}], {1, 2, 1, 3, 1, 3}, 2]
{2 a + b + 3 c, a + 2 b + c, 3 a + b + 2 c, a + 3 b + c, 3 a + b + 3 c, a + 3 b + c}
```

```
maxwellenzahl = 0.0625;
maxwellenzahl = 0.125;
maxwellenzahl = 0.25;
maxwellenzahl = 0.5;
kmax = maxwellenzahl;
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}]];
```

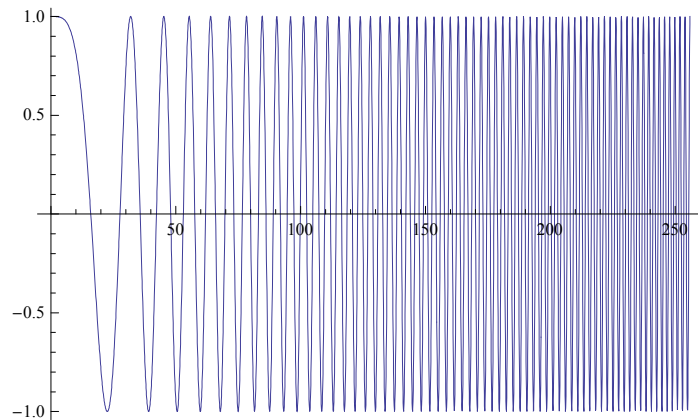
```
Show[wellenbild, ImageSize -> 513]
```



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



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



Diskrete Fouriertransformation (DFT)

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

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

– unitäre Transformation, d.h. umkehrbar eindeutig und linear

- Zerlegung in reelle und imaginäre Basisfunktionen
- lineare Operation : $af_m + bg_m = a\hat{f}_u + b\hat{g}_u$
- Symmetrieeigenschaften $f_m = -f_{-m}$ (ungerade) und $f_m = f_{-m}$ (gerade) bleiben unter DFT erhalten
- Periodizität : $f_m = -f_{M-m}$ (unger.) und $f_m = f_{M-m}$ (ger.)
- bei reellem Signal $\rightarrow$ hermitesche Transformierte : $\hat{f}_u^* = \hat{f}_{M-u}$
- Verschiebungssatz : $f_{m-l} \rightarrow \hat{f}_u e^{\left(\frac{-2\pi i l u}{M}\right)}$ d.h. konst. Versch. führt zu Phasenverschiebung
- Faltungssatz : $(f * g)_m = \sum_{j=0}^{M-1} f_j g_{m-j} = \sum_{j=0}^{M-1} f_{m-j} g_j \rightarrow M \hat{f}_u \hat{g}_u$

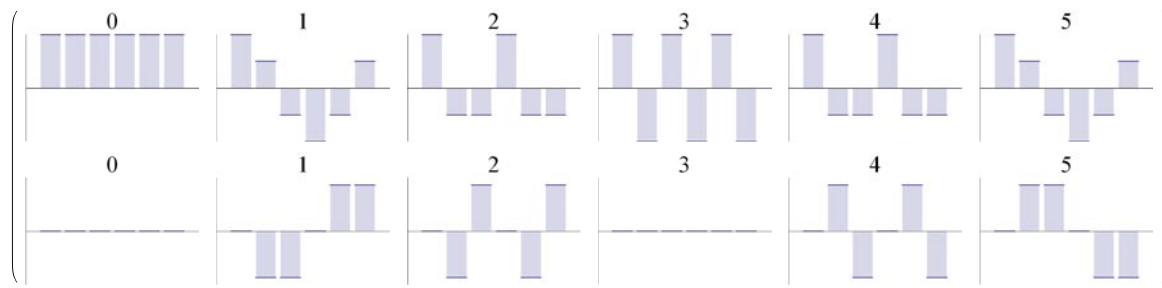
Rücktransformation : $f_m = \frac{1}{M} \sum_{u=0}^{M-1} \hat{f}_u e^{\left(\frac{2\pi i m u}{M}\right)}$ mit $m = 0 \dots M - 1$

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

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

Basisfunktionen der 1D-DFT für M=6

```
mm = 6;
MatrixForm[
{Table[DiscretePlot[Re[Exp[-2 i π m u / mm]], {m, 0, mm - 1}, PlotMarkers → None, Ticks → None, PlotRange → {{-1, mm}, {-1, 1}}, ExtentSize → .8,
PlotLabel → u], {u, Range[0, mm - 1]}], Table[DiscretePlot[Im[Exp[-2 i π m u / mm]], {m, 0, mm - 1}, PlotMarkers → None,
Ticks → None, PlotRange → {{-1, mm}, {-1, 1}}, ExtentSize → .8, PlotLabel → u], {u, Range[0, mm - 1]}]}
```



Beispiel : Boxfilter

$$g = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right)^T$$

$$g_0 = \frac{1}{3}$$

$$g_1 = \frac{1}{3}$$

$$g_{M-1} = \frac{1}{3}$$

Anwendung von $\hat{f}_u = \sum_{m=0}^{M-1} f_m e^{\left(\frac{-2\pi i m u}{M}\right)}$ mit $u = 0 \dots M-1$

$$\hat{g}_u = \frac{1}{3} + \frac{1}{3} e^{\left(\frac{-2\pi i u}{M}\right)} + \frac{1}{3} e^{\left(\frac{-2\pi i (M-1) u}{M}\right)}$$

$$\hat{g}_u = \frac{1}{3} + \frac{1}{3} e^{\left(\frac{-2\pi i u}{M}\right)} + \frac{1}{3} e^{\left(-2 i \pi u + \frac{2 i \pi u}{M}\right)}$$

$$\hat{g}_u = \frac{1}{3} + \frac{1}{3} e^{\left(\frac{-2\pi i u}{M}\right)} + \frac{1}{3} e^{\left(-2 i \pi u\right)} e^{\left(\frac{2 i \pi u}{M}\right)}$$

Simplify $\left[e^{\left(-2 i \pi u\right)}, u \in \text{Integers} \right]$

1

Solve $\left[e^{\left(-2 i \pi u\right)} = 1, u \right]$

$\{\{u \rightarrow \text{ConditionalExpression}[-C[1], C[1] \in \text{Integers}]\}\}$

$$\hat{g}_u = \frac{1}{3} + \frac{1}{3} e^{\left(\frac{-2\pi i u}{M}\right)} + \frac{1}{3} e^{\left(\frac{2i\pi u}{M}\right)}$$

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

$$\cos\left[\frac{2\pi u}{M}\right] - i \sin\left[\frac{2\pi u}{M}\right]$$

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

$$\cos\left[\frac{2\pi u}{M}\right] + i \sin\left[\frac{2\pi u}{M}\right]$$

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

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

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

$$e^{-i x} + e^{i x} \quad // \text{ExpToTrig}$$

$$2 \cos[x]$$

$$\frac{1}{3} \left(1 + e^{\left(\frac{-2\pi i u}{M}\right)} + e^{\left(\frac{2i\pi u}{M}\right)} \right) \quad // \text{ExpToTrig} \quad // \text{FullSimplify}$$

$$\frac{1}{3} \left(1 + 2 \cos\left[\frac{2\pi u}{M}\right] \right)$$

$$\hat{g}_u = \frac{1}{3} \left(2 \cos\left(\frac{2\pi u}{M}\right) + 1 \right)$$

Das Verhältnis u / M wird als Wellenzahl $k = 0 \dots 0.5$ bezeichnet

$$\hat{g}_k = \frac{1}{3} (2 \cos(2\pi k) + 1)$$

Beispiel : Boxfilter

$$\hat{g}_u = 1/3 + 1/3 e^{\left(\frac{-2\pi i u}{M}\right)} + 1/3 e^{\left(\frac{2\pi i u}{M}\right)} \quad // \text{ FullSimplify}$$

$$\frac{1}{3} \left(1 + 2 \cos\left[\frac{2\pi u}{M}\right] \right)$$

$$\hat{g}_k = 1/3 + 1/3 e^{(-2\pi i k)} + 1/3 e^{(2\pi i k)} \quad // \text{ FullSimplify}$$

$$\frac{1}{3} (1 + 2 \cos[2\pi k])$$

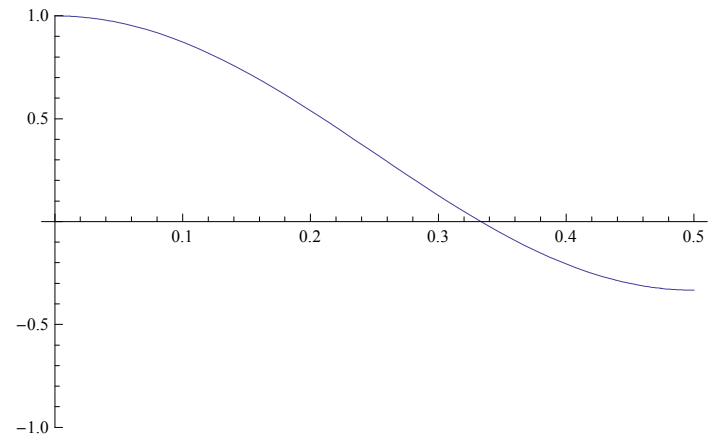
Transferfunktion $\hat{g}_k$ des Boxfilters $g = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^T$

$$\text{Solve}[1/3 * (1 + 2 * \cos[2\pi k]) == 0 \&\& 0 \leq k \leq 1/2, k]$$

$$\left\{ \left\{ k \rightarrow \frac{1}{3} \right\} \right\}$$

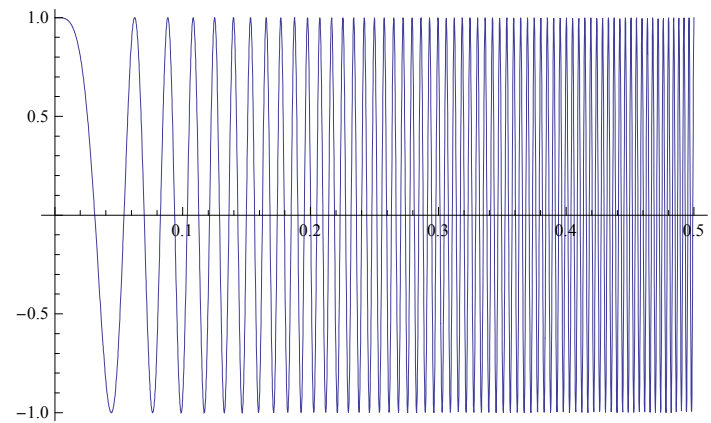
```
Plot[1 / 3 * (1 + 2 * Cos[2 π k]), {k, 0, maxwellenzahl}, PlotRange → {Full, {-1, 1}}]
```

```
Solve[1 / 3 * (1 + 2 * Cos[2 π k]) == 0 && 0 ≤ k ≤ 1 / 2, k]
```

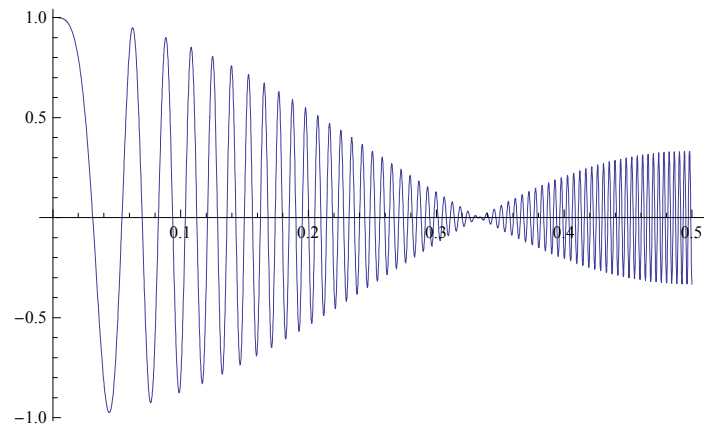


$$\left\{ \left\{ k \rightarrow \frac{1}{3} \right\} \right\}$$

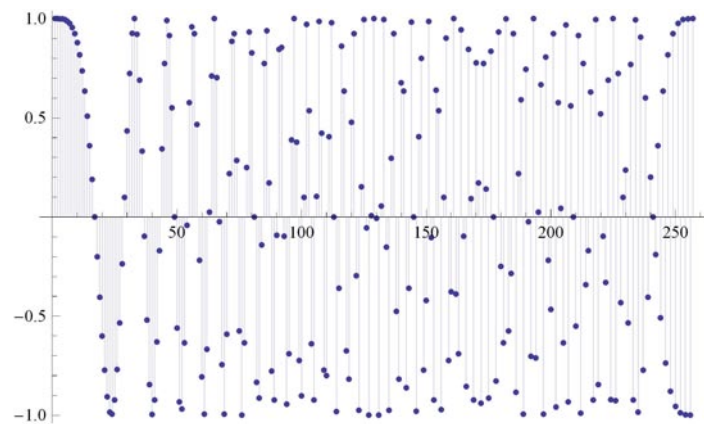
```
Plot[Cos[x * π * x * 0.5 / maxwellenzahl * 512], {x, 0, maxwellenzahl}]
```



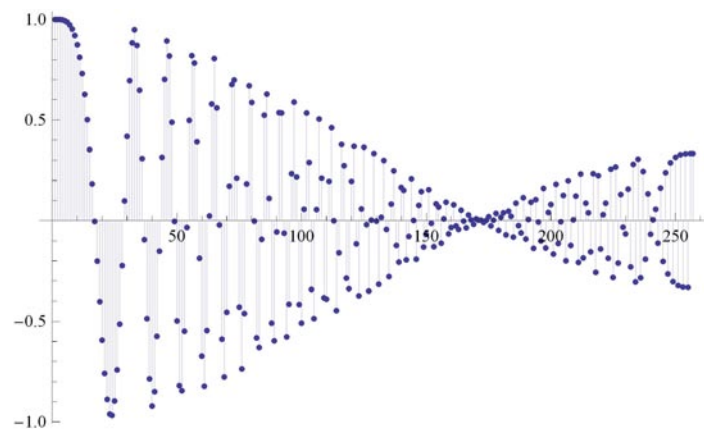
```
Plot[Cos[k *  $\pi$  * k * 0.5 / maxwellenzahl * 512] * (1 / 3 * (1 + 2 * Cos[2  $\pi$  k])), {k, 0, maxwellenzahl}]
```



```
ListPlot[Table[Cos[x *  $\pi$  * x * maxwellenzahl / 256], {x, 0, 256}], Filling -> Axis]
```



```
ListPlot[ListConvolve[{1 / 3, 1 / 3, 1 / 3}, Table[Cos[x *  $\pi$  * x * maxwellenzahl / 256], {x, 0, 256}], 2], Filling -> Axis]
```



$$h = \left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5} \right)^T$$

Anwendung von $\hat{f}_u = \sum_{m=0}^{M-1} f_m e^{\left(\frac{-2\pi i m u}{M}\right)}$ mit $u = 0 \dots M-1$

$$\hat{h}_u = \frac{1}{5} e^{(-2\pi i 0 k)} + \frac{1}{5} e^{(-2\pi i 1 k)} + \frac{1}{5} e^{(-2\pi i 2 k)} + \frac{1}{5} e^{\left(\frac{-2\pi i (M-2) u}{M}\right)} + \frac{1}{5} e^{\left(\frac{-2\pi i (M-1) u}{M}\right)}$$

$$\hat{h}_u = \frac{1}{5} e^{(-2\pi i 0 k)} + \frac{1}{5} e^{(-2\pi i 1 k)} + \frac{1}{5} e^{(-2\pi i 2 k)} + \frac{1}{5} e^{\left(-2i\pi u + \frac{4i\pi u}{M}\right)} + \frac{1}{5} e^{\left(-2i\pi u + \frac{2i\pi u}{M}\right)}$$

$$\hat{h}_k = \frac{1}{5} e^{(-2\pi i 0 k)} + \frac{1}{5} e^{(-2\pi i 1 k)} + \frac{1}{5} e^{(-2\pi i 2 k)} + \frac{1}{5} e^{(4i\pi k)} + \frac{1}{5} e^{(2i\pi k)}$$

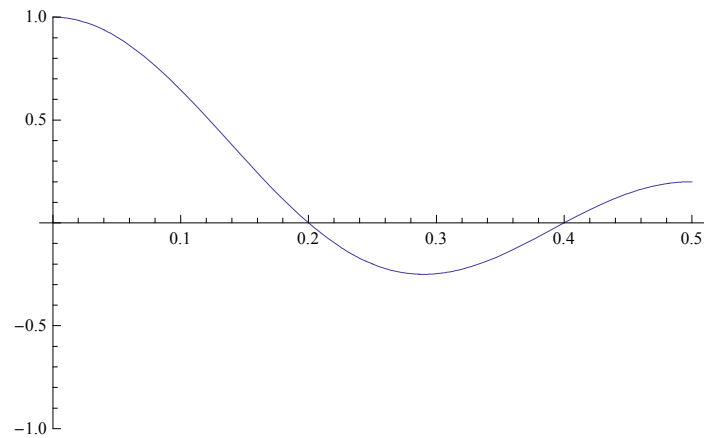
$$\hat{h}_k = \frac{1}{5} e^{(-2\pi i 0 k)} + \frac{1}{5} e^{(-2\pi i 1 k)} + \frac{1}{5} e^{(-2\pi i 2 k)} + \frac{1}{5} e^{(4i\pi k)} + \frac{1}{5} e^{(2i\pi k)} // \text{FullSimplify}$$

$$\frac{1}{5} (1 + 2 \cos[2 k \pi] + 2 \cos[4 k \pi])$$

Transferfunktion $\hat{h}_k$ des Boxfilters $h = \left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5} \right)^T$

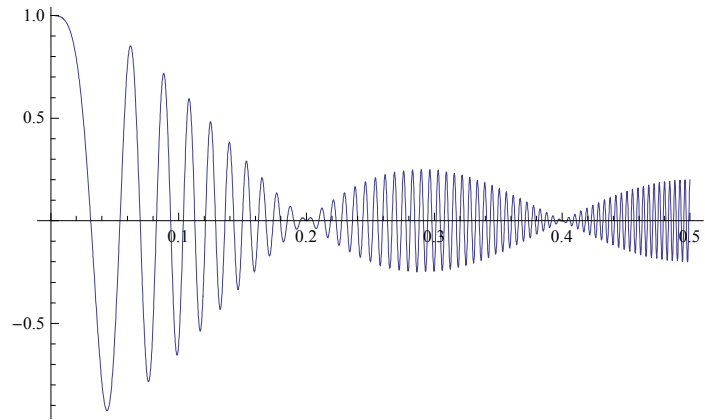
`Plot[1/5 (1 + 2 Cos[2 k π] + 2 Cos[4 k π]), {k, 0, maxwellenzahl}, PlotRange → {Full, {-1, 1}}]`

`Solve[1/5 (1 + 2 Cos[2 k π] + 2 Cos[4 k π]) == 0 && 0 ≤ k ≤ 1/2, k]`

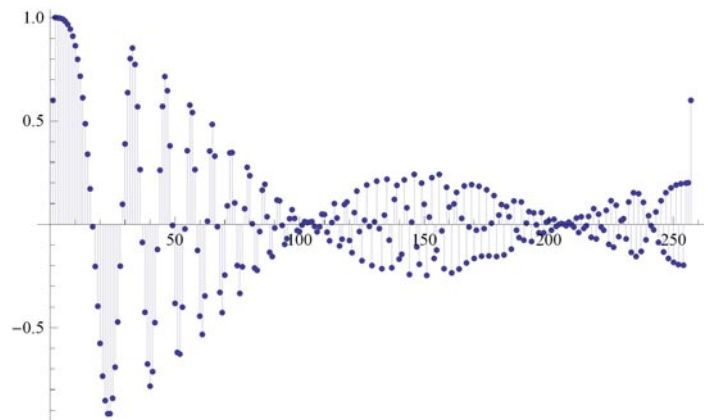


$$\left\{ \left\{ k \rightarrow \frac{1}{5} \right\}, \left\{ k \rightarrow \frac{2}{5} \right\} \right\}$$

```
Plot[Cos[k * π * k * 0.5 / maxwellenzahl * 512] * 1 / 5 (1 + 2 Cos[2 k π] + 2 Cos[4 k π]), {k, 0, maxwellenzahl}]
```



```
ListPlot[ListConvolve[{1 / 5, 1 / 5, 1 / 5, 1 / 5, 1 / 5}, Table[Cos[x * π * x * maxwellenzahl / 256], {x, 0, 256}], 3], Filling -> Axis]
```

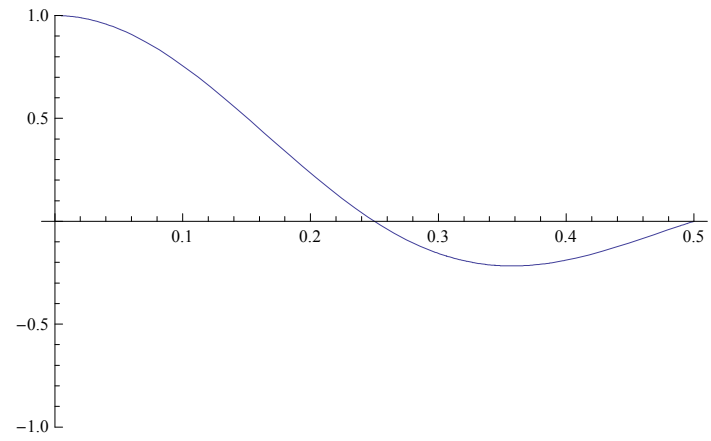


Für größere Maskengrößen des Boxfilters ist die DFT mittels $(\sin x)/x$ - Funktion approximierbar :

```
FourierTransform[UnitBox[x / (8 π)] / (4 √(2 π)), x, u]
```

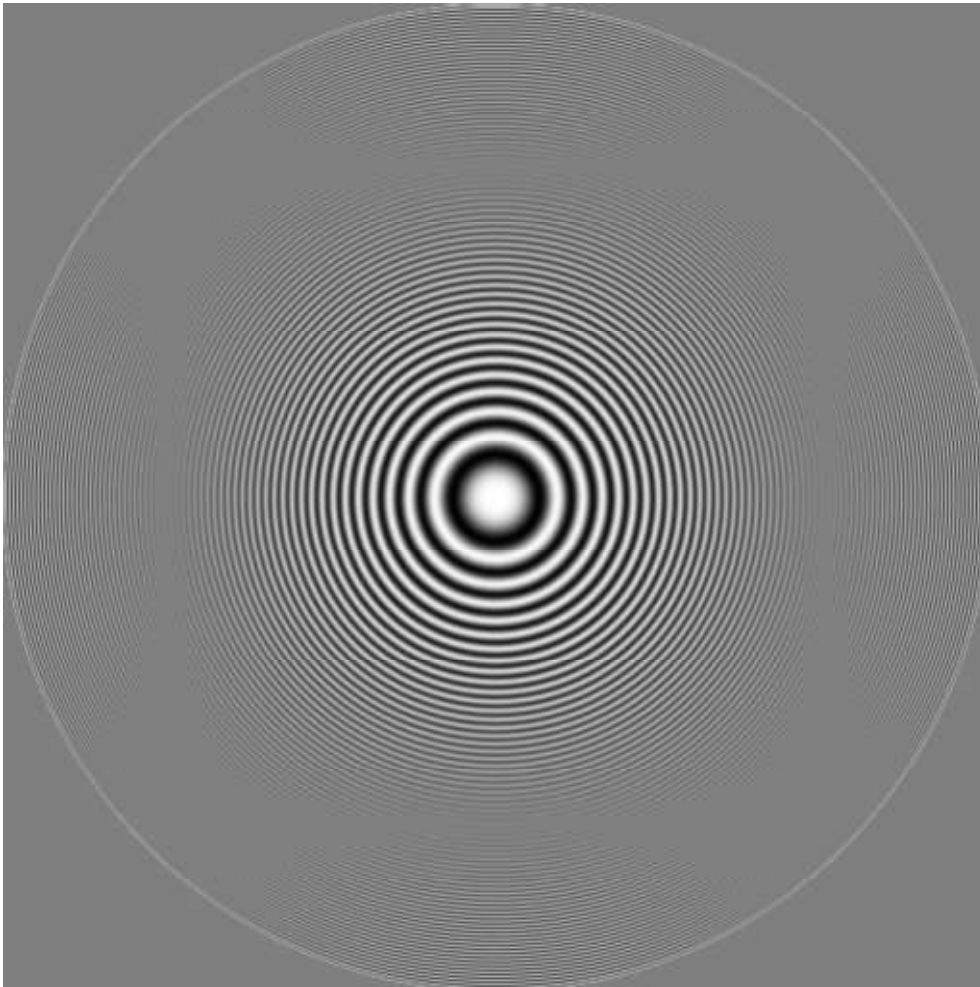
```
Sinc[4 π u]
```

```
Plot[Sinc[4  $\pi$  u], {u, 0, 1/2}, PlotRange -> {Full, {-1, 1}}]
```



Wie sieht das nun für das Ringwellenbild aus?

```
Image[ListConvolve[Table[1 / 9, {3}, {3}], ImageData[wellenbild], 2]]
```



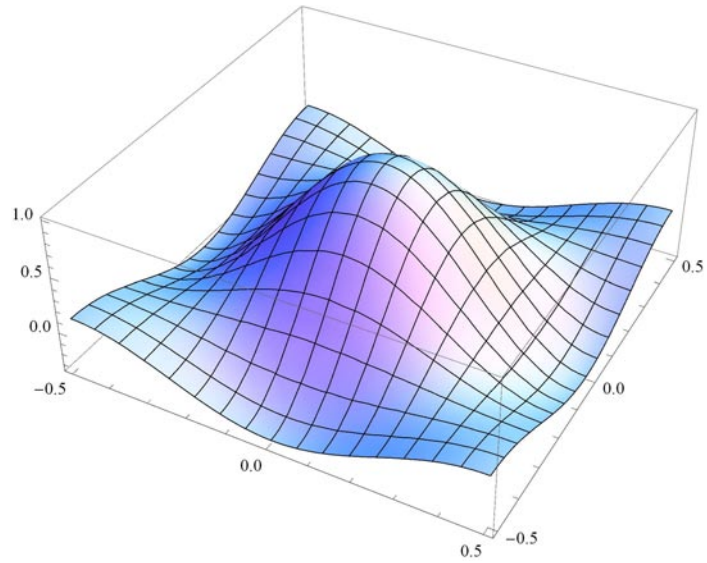
```
Transpose[{Table[1 / 3, {3}]]].{Table[1 / 3, {3}]}
```

$$\left\{ \left\{ \frac{1}{9}, \frac{1}{9}, \frac{1}{9} \right\}, \left\{ \frac{1}{9}, \frac{1}{9}, \frac{1}{9} \right\}, \left\{ \frac{1}{9}, \frac{1}{9}, \frac{1}{9} \right\} \right\}$$

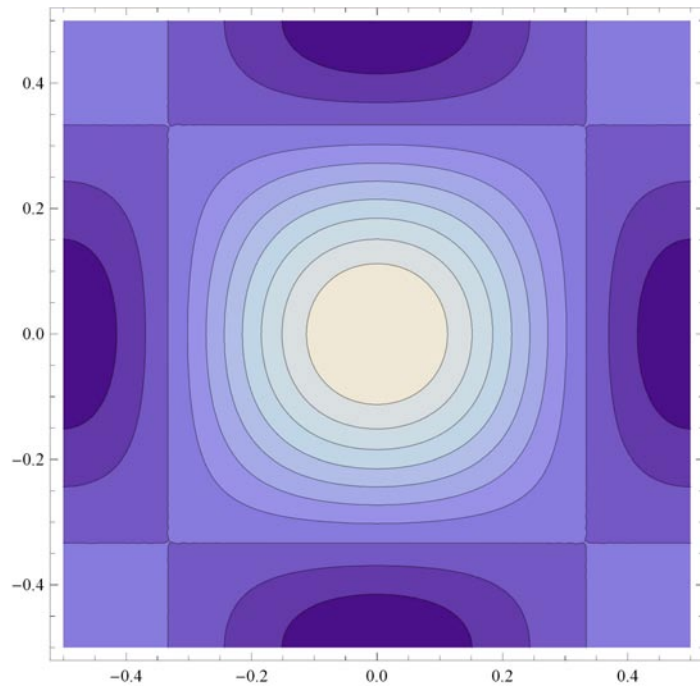
```
Table[1 / 9, {3}, {3}]
```

$$\left\{ \left\{ \frac{1}{9}, \frac{1}{9}, \frac{1}{9} \right\}, \left\{ \frac{1}{9}, \frac{1}{9}, \frac{1}{9} \right\}, \left\{ \frac{1}{9}, \frac{1}{9}, \frac{1}{9} \right\} \right\}$$

```
Plot3D[1 / 3 * (1 + 2 * Cos[2  $\pi$   $k_1$ ]) * 1 / 3 * (1 + 2 * Cos[2  $\pi$   $k_2$ ]), { $k_1$ , -maxwellenzahl, maxwellenzahl}, { $k_2$ , -maxwellenzahl, maxwellenzahl}]
```



```
ContourPlot[1 / 3 * (1 + 2 * Cos[2  $\pi$   $k_1$ ]) * 1 / 3 * (1 + 2 * Cos[2  $\pi$   $k_2$ ]),  
{ $k_1$ , -maxwellenzahl, maxwellenzahl}, { $k_2$ , -maxwellenzahl, maxwellenzahl}, Contours  $\rightarrow$  10]
```



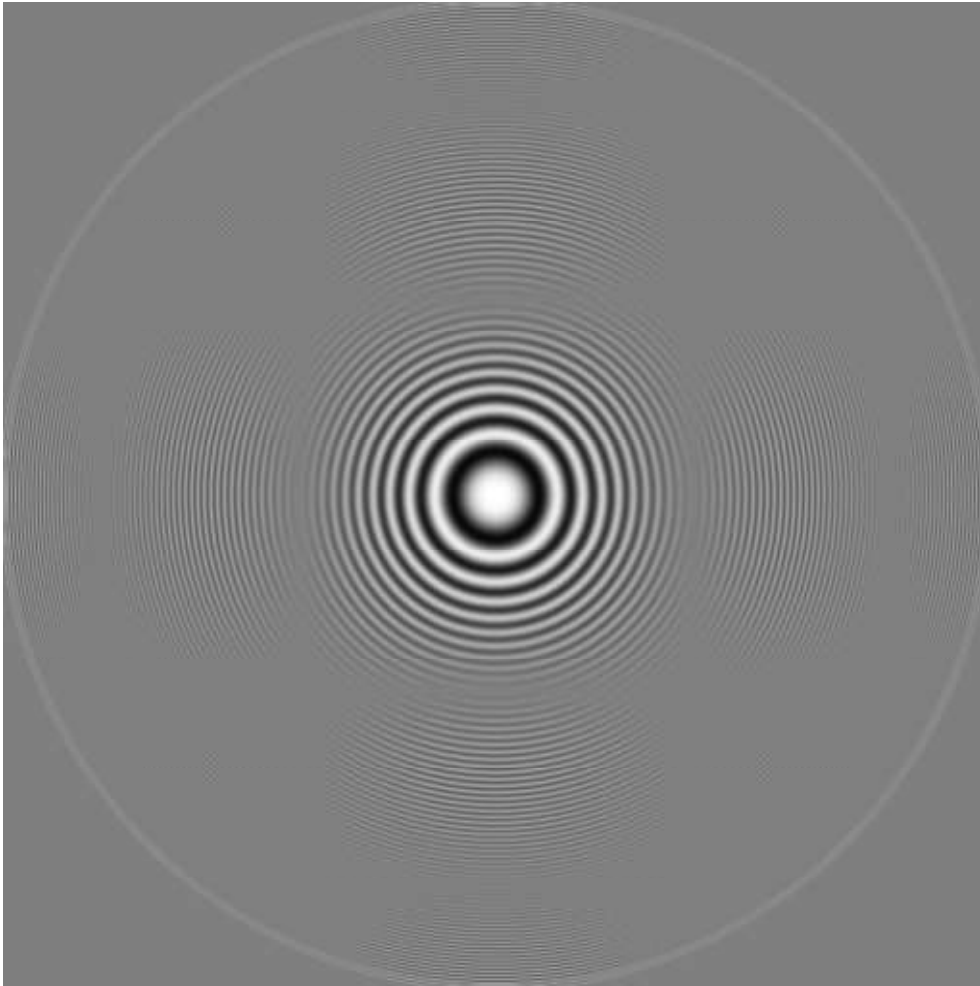
Transpose[{Table[1 / 5, {5}]]}.{Table[1 / 5, {5}]}

$\left\{ \left\{ \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25} \right\}, \left\{ \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25} \right\}, \left\{ \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25} \right\}, \left\{ \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25} \right\}, \left\{ \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25}, \frac{1}{25} \right\} \right\}$

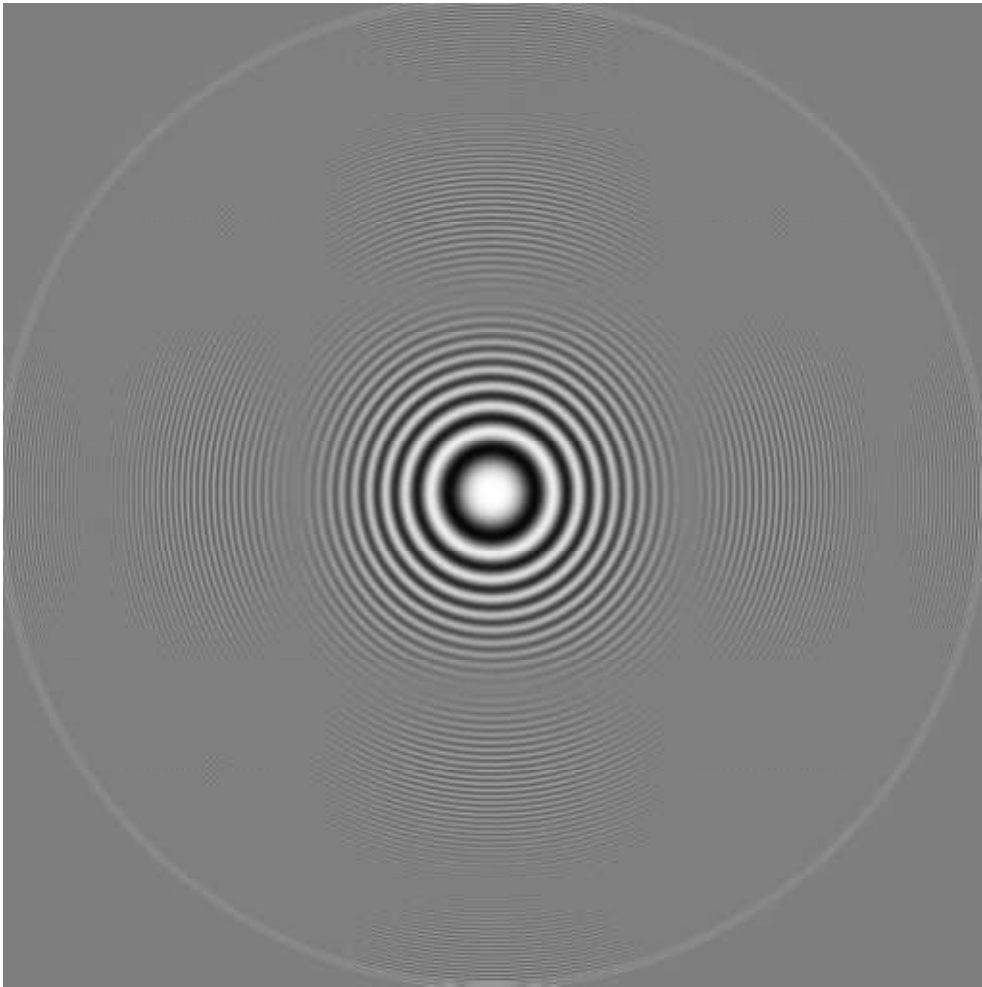
Einschub: Separierbarkeit

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 $2N$ Lesezugriffe und Multiplikationen sowie $2(N - 1)$ Additionen zustande.

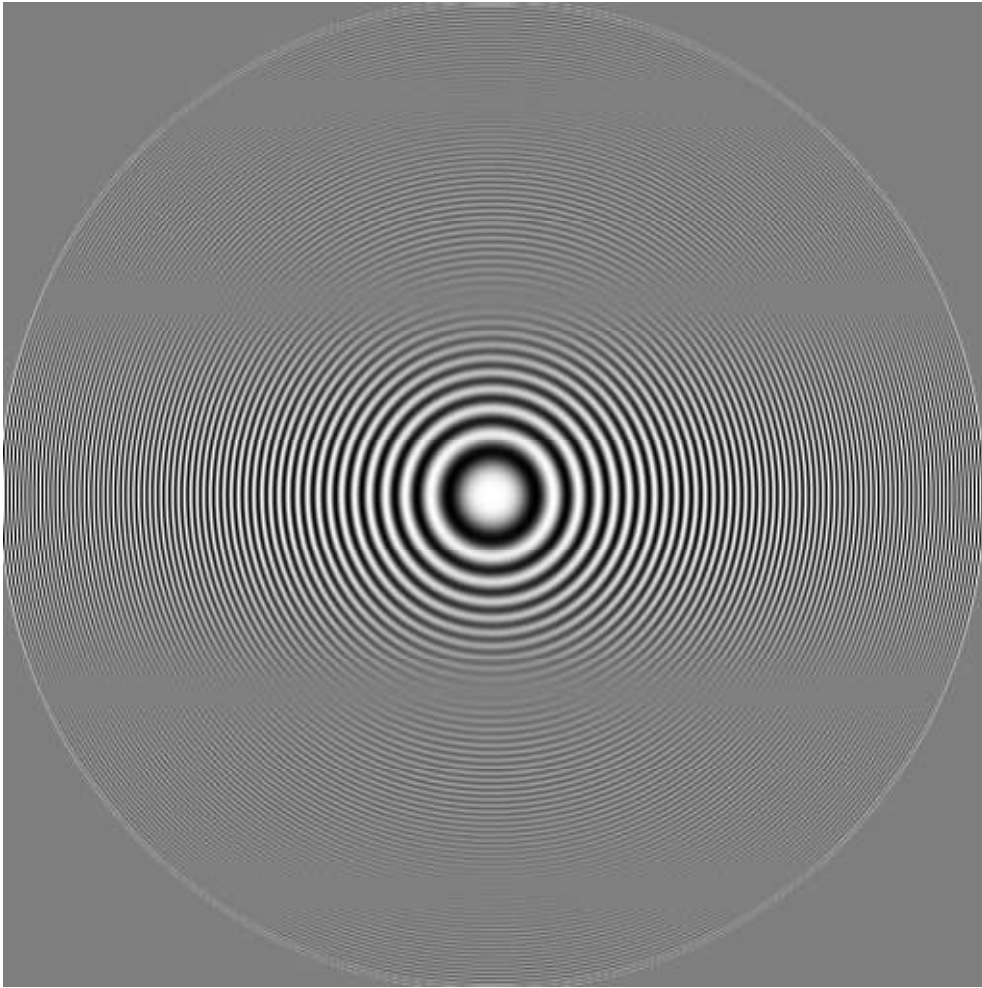
```
Image[ListConvolve[Transpose[{Table[1 / 5, {5}]}].{Table[1 / 5, {5}]}, ImageData[wellenbild], 3]]
```



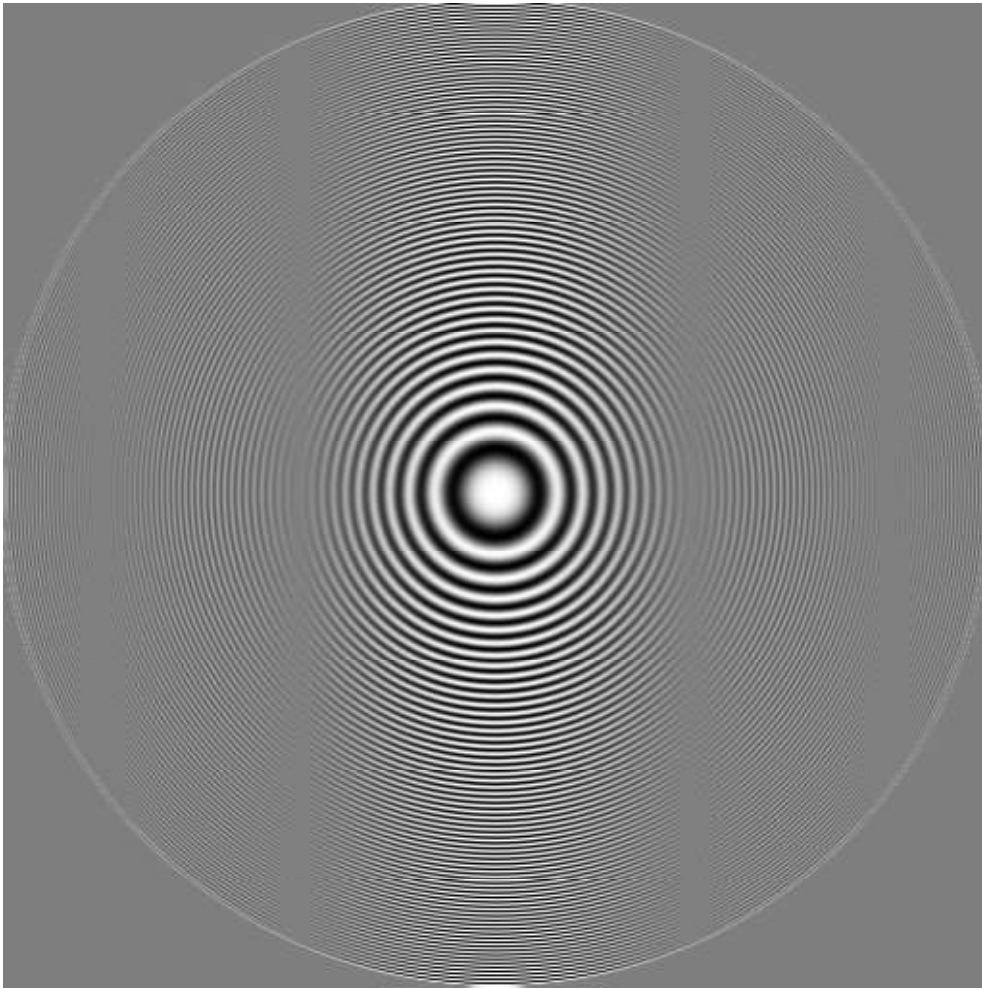
```
Image[ListConvolve[{Table[1 / 5, {5}]}], ListConvolve[Transpose[{Table[1 / 5, {5}]}], ImageData[wellenbild], 3], 3]]
```



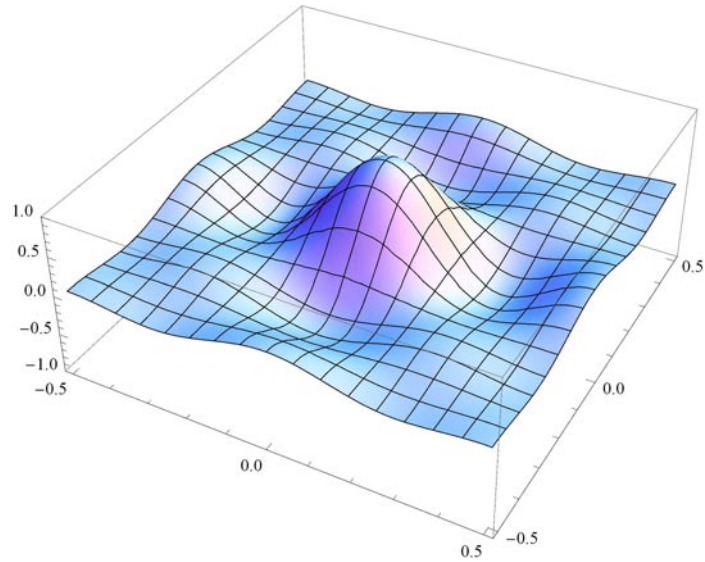
```
Image[ListConvolve[Transpose[{Table[1 / 5, {5}]}], ImageData[wellenbild], 3]]
```



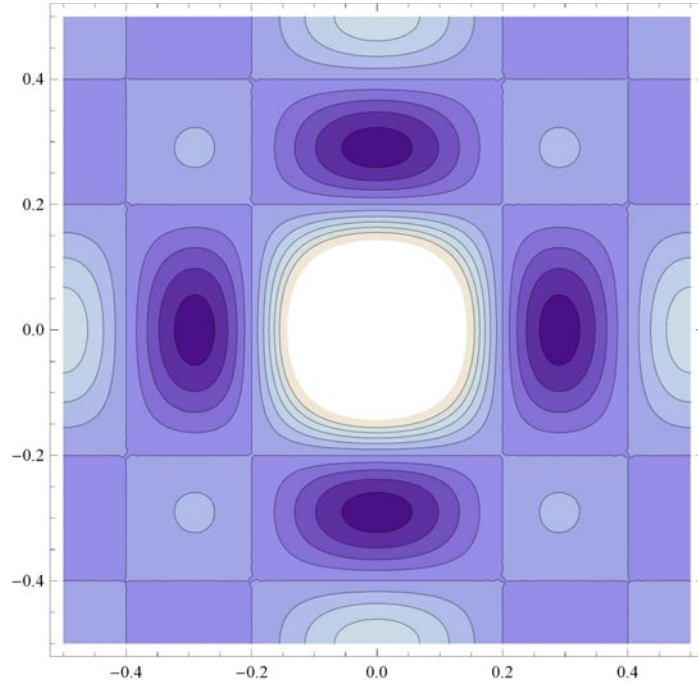
```
Image[ListConvolve[{Table[1 / 5, {5}]}], ImageData[wellenbild], 3]]
```



```
Plot3D[1 / 5 (1 + 2 Cos[2 k1 π] + 2 Cos[4 k1 π]) * 1 / 5 (1 + 2 Cos[2 k2 π] + 2 Cos[4 k2 π]),  
{k1, -maxwellenzahl, maxwellenzahl}, {k2, -maxwellenzahl, maxwellenzahl}, PlotRange → {Full, Full, {-1, 1}}]
```



```
ContourPlot[1 / 5 (1 + 2 Cos[2 k1 π] + 2 Cos[4 k1 π]) * 1 / 5 (1 + 2 Cos[2 k2 π] + 2 Cos[4 k2 π]),  
{k1, -maxwellenzahl, maxwellenzahl}, {k2, -maxwellenzahl, maxwellenzahl}, Contours → 10]
```



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

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

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

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

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

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

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

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

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

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

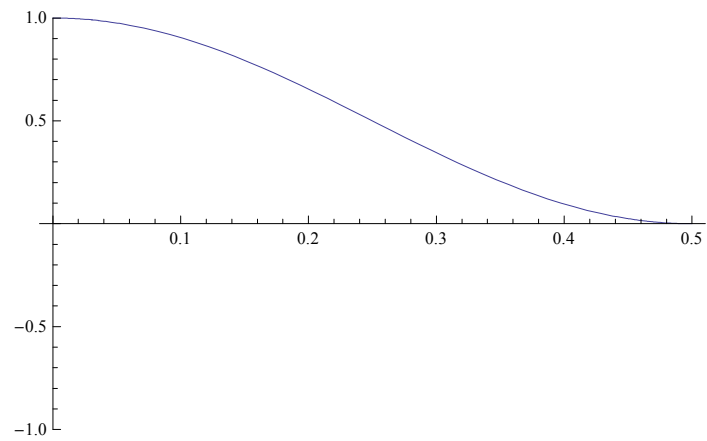
$$\hat{b}_k = 1/2 + 1/4 e^{(-2 \pi i k)} + 1/4 e^{(2 i \pi k)} \quad // \text{ExpToTrig}$$

$$\frac{1}{2} + \frac{1}{2} \cos[2 k \pi]$$

Transferfunktion $\hat{b}_k$ des Binomialfilters $\mathbf{b} = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)^T$

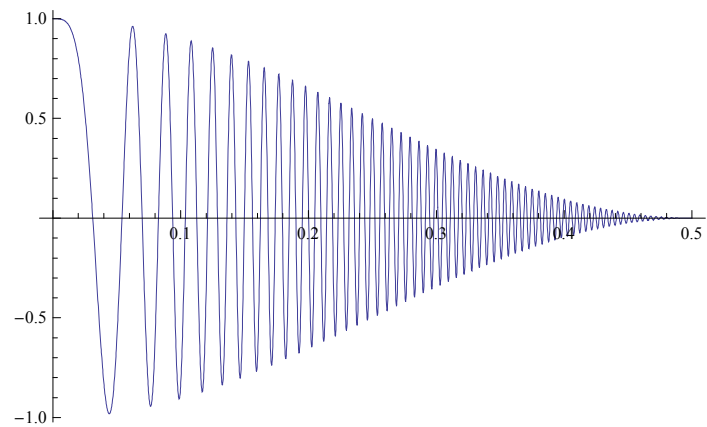
```
Plot[1/2 (1 + Cos[2 k π]), {k, 0, maxwellenzahl}, PlotRange → {Full, {-1, 1}}]
```

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



```
{ {k → 1/2}, {k → 1/2} }
```

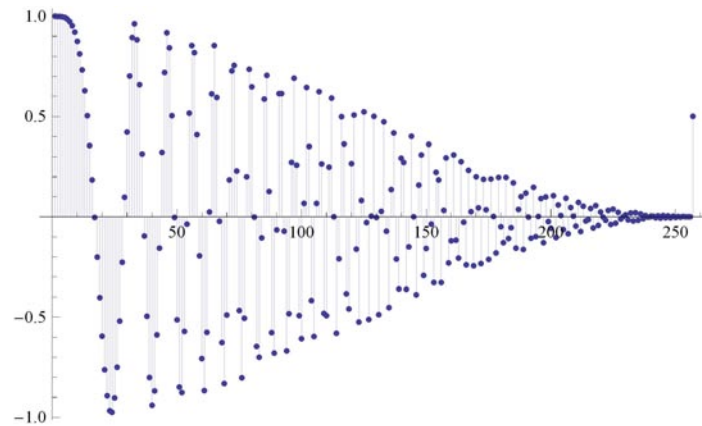
```
Plot[Cos[k * π * k * 0.5 / maxwellenzahl * 512] * 1/2 (1 + Cos[2 k π]), {k, 0, maxwellenzahl}]
```



```
Column[Table[Binomial[n, k], {n, 0, 6, 1}, {k, 0, n}], Center]
```

```
{1}
{1, 1}
{1, 2, 1}
{1, 3, 3, 1}
{1, 4, 6, 4, 1}
{1, 5, 10, 10, 5, 1}
{1, 6, 15, 20, 15, 6, 1}
```

```
ListPlot[ListConvolve[{1/4, 1/2, 1/4}, Table[Cos[x* $\pi$ *x*maxwellenzahl/256], {x, 0, 256}], 2], Filling  $\rightarrow$  Axis]
```

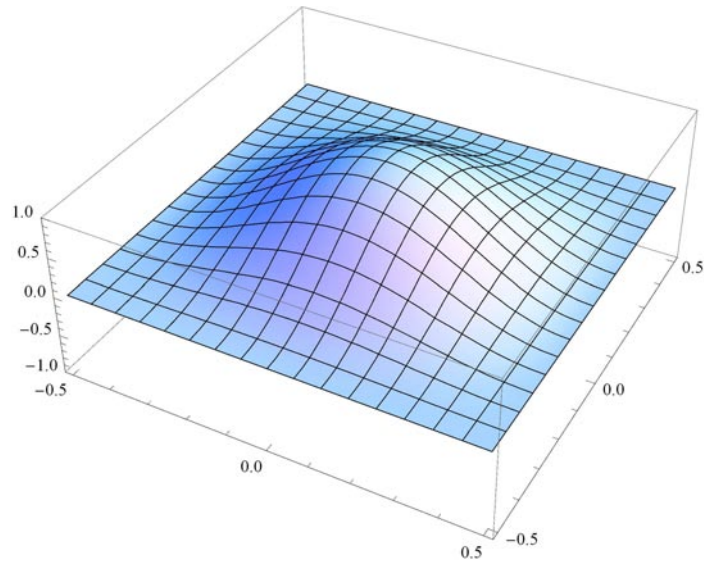


```
ordnung = 2;
binom[ordnung]
binom[2]
```

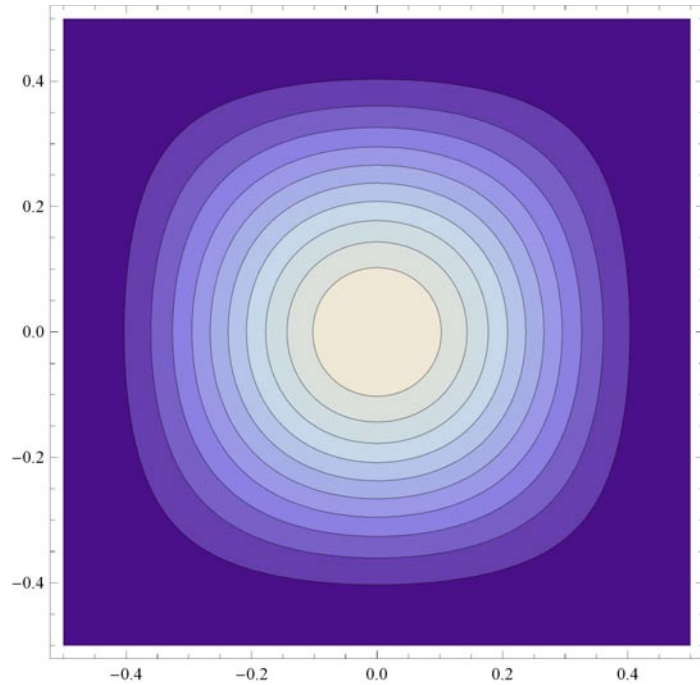
```
Transpose[{binom[ordnung]}.{binom[ordnung]} // MatrixForm
```

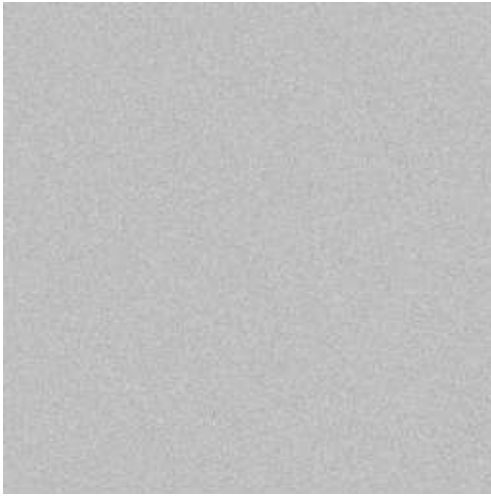
```
ImageData[wellenbild], 2]]
```

```
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}}]
```

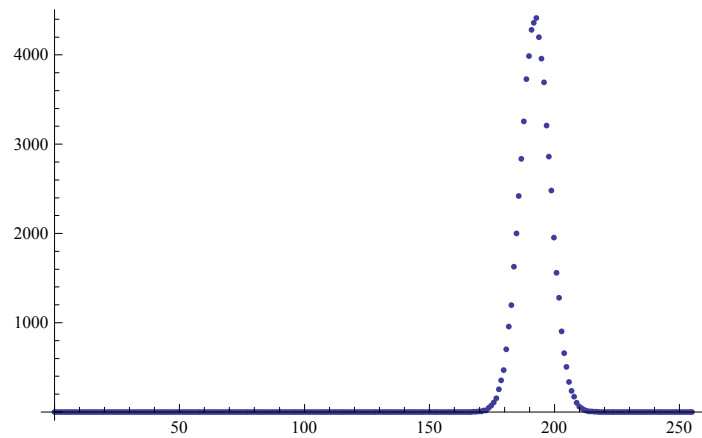


```
ContourPlot[1/2 (1 + Cos[2 k1 π]) * 1/2 (1 + Cos[2 k2 π]),
{k1, -maxwellenzahl, maxwellenzahl}, {k2, -maxwellenzahl, maxwellenzahl}, Contours → 10]
```



rauschbild

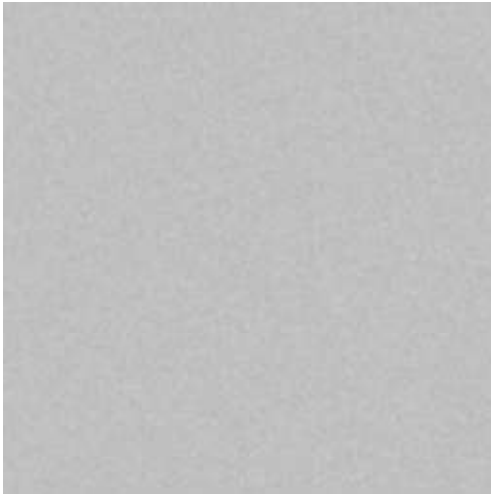
```
ListPlot[BinCounts[Flatten[ImageData[rauschbild]], {0, 1, 1 / 255}], PlotRange → Full, DataRange → {0, 255}]
```



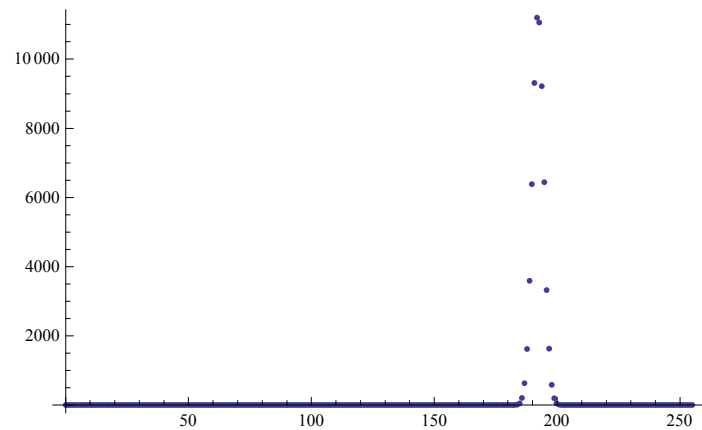
```
StandardDeviation[Flatten[ImageData[rauschbild] * 255]]
```

6.00362

```
geglättetesrauschbild = Image[ListConvolve[Transpose[{binom[ordnung]}].{binom[ordnung]}, ImageData[rauschbild], 2]]
```



```
ListPlot[BinCounts[Flatten[ImageData[geglättetesrauschbild]], {0, 1, 1 / 255}], PlotRange → Full, DataRange → {0, 255}]
```



```
StandardDeviation[Flatten[ImageData[geglättetesrauschbild] * 255]]
```

2.25813

Senkung des Rauschens um die Wurzel der Summe der Quadrate der Filterkoeffizienten

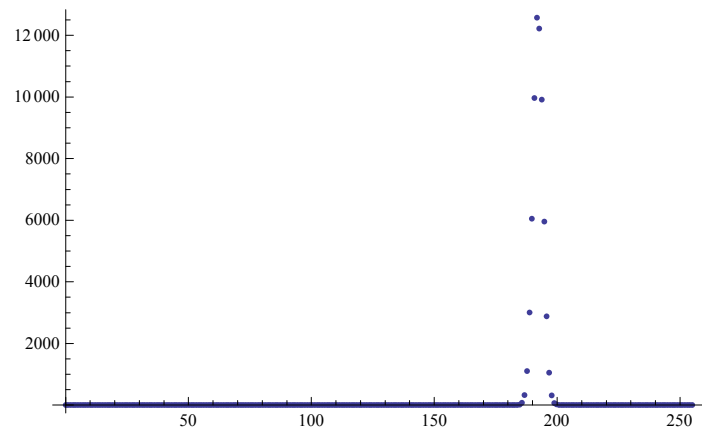
```
StandardDeviation[Flatten[ImageData[rauschbild] * 255]] * Sqrt[Plus @@ ((Flatten@ (Transpose[{binom[ordnung]}].{binom[ordnung]}))^2)]
```

2.25136

```
geglättetesrauschbild = Image[ListConvolve[Transpose[{box@3}].{box@3}, ImageData[rauschbild], 2]]
```



```
ListPlot[BinCounts[Flatten[ImageData[geglättetesrauschbild]], {0, 1, 1 / 255}], PlotRange → Full, DataRange → {0, 255}]
```



```
StandardDeviation[Flatten[ImageData[geglättetesrauschbild] * 255]]
```

```
2.00665
```

```
StandardDeviation[Flatten[ImageData[rauschbild] * 255]] * Sqrt[Plus @@ ((Flatten@ (Transpose[{box[ordnung + 1]}].{box[ordnung + 1]})) ^ 2)]
```

```
2.00121
```

```
ListConvolve[{1, 2, 1}, {0, 1, 2, 1, 0}, 2]
```

```
{1, 4, 6, 4, 1}
```

```
ListConvolve[binom[ordnung], Join[{0}, binom[ordnung], {0}], 2]
```

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

Reprise:

Diskrete Fouriertransformation (DFT):

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

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

zudem

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

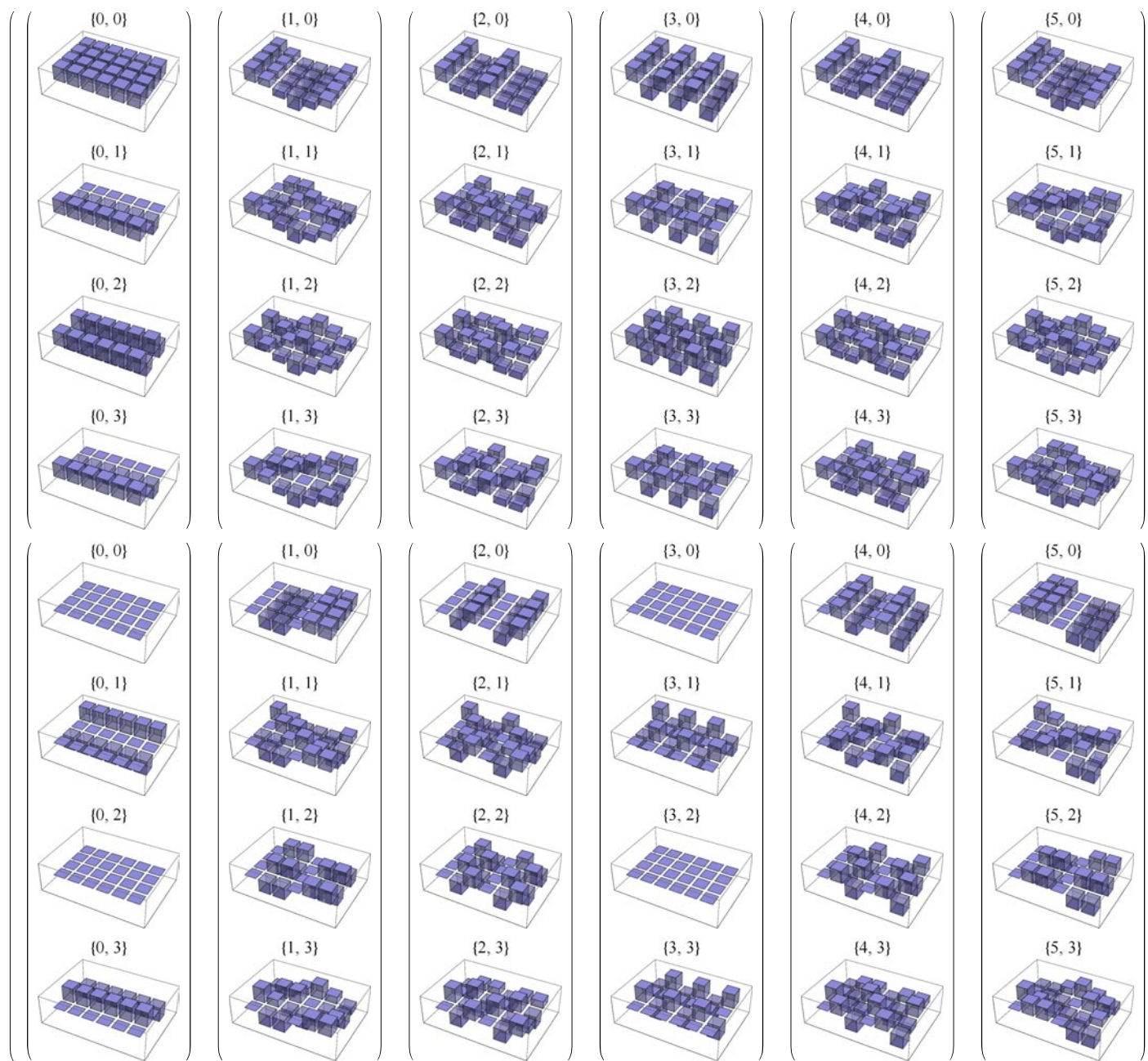
$$2 \text{ D – Rücktransformation : } f_{m,n} = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}_{u,v} e^{\left(\frac{2\pi i m u}{M}\right)} e^{\left(\frac{2\pi i n v}{N}\right)} \quad \text{mit } m = 0 \dots M - 1 \text{ und } n = 0 \dots N - 1$$

Das Verhältnis u/M bzw. v/N wird als Wellenzahl $k_1=0\dots0.5$ bzw. $k_2=0\dots0.5$ bezeichnet.

```
mm = 6;
```

```
nn = 4;
```

```
MatrixForm[{Table[DiscretePlot3D[Re[Exp[-2 i π m u / mm] * Exp[-2 i π n v / nn]], {m, 0, mm - 1}, {n, 0, nn - 1}, PlotMarkers → None, Ticks → None,
  PlotRange → {{-1, mm}, {-1, nn}, {-1, 1}}, ExtentSize → .8, PlotLabel → {u, v}, BoxRatios → {mm + 1, nn + 1, 2}, RotationAction → "Clip"],
{u, Range[0, mm - 1]}, {v, Range[0, nn - 1]}], Table[DiscretePlot3D[Im[Exp[-2 i π m u / mm] * Exp[-2 i π n v / nn]],
  {m, 0, mm - 1}, {n, 0, nn - 1}, PlotMarkers → None, Ticks → None, PlotRange → {{-1, mm}, {-1, nn}, {-1, 1}}, ExtentSize → .8,
  PlotLabel → {u, v}, BoxRatios → {mm + 1, nn + 1, 2}, RotationAction → "Clip"], {u, Range[0, mm - 1]}, {v, Range[0, nn - 1]}]]]
```



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  $\pi$  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  $\rightarrow$  Element[u, Integers]] /. u / mm  $\rightarrow$  k1);
  (*Print[TrigReduce[resultat]];*)
  Return[resultat];
];

```

Automatisierte analytische Berechnung der 2D-DFT

```

Clear[transferfunktion2d];
(*Variante 0 ... M-1 und 0 ... N-1*)
transferfunktion2d[g_List] := Module[{resultat, u, v, mm, nn, m, n},
  resultat =
  ((
    FullSimplify[
      FullSimplify[
        Sum[
          Sum[
            g[[Mod[m + (Dimensions[g][[1]] - 1) / 2, Dimensions[g][[1]]] + 1, Mod[n + (Dimensions[g][[2]] - 1) / 2, Dimensions[g][[2]]] + 1]] *
            Exp[-2  $\pi$  i * Join[Table[i, {i, 0, (Dimensions[g][[1]] - 1) / 2, 1}],
              Table[mm - i, {i, (Dimensions[g][[1]] - 1) / 2, 1, -1}]]][[m + 1]] * u / mm] *
            Exp[-2  $\pi$  i * Join[Table[j, {j, 0, (Dimensions[g][[2]] - 1) / 2, 1}], Table[nn - j, {j,
              (Dimensions[g][[2]] - 1) / 2, 1, -1}]]][[n + 1]] * v / nn],
            {m, 0, Dimensions[g][[1]] - 1, 1}],
            {n, 0, Dimensions[g][[2]] - 1, 1}],
            Assumptions  $\rightarrow$  Element[u, Integers]],
            Assumptions  $\rightarrow$  Element[v, Integers]]
            /. u / mm  $\rightarrow$  k1 )
            /. v / nn  $\rightarrow$  k2
          );
  (*Print[FullSimplify[ComplexExpand[ExpToTrig[resultat]] /. u / mm  $\rightarrow$  k1 /. v / nn  $\rightarrow$  k2]];*)
  Return[FullSimplify[ComplexExpand[ExpToTrig[resultat]] /. u / mm  $\rightarrow$  k1 /. v / nn  $\rightarrow$  k2]];
];

```

```

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
    );
  (*Print[FullSimplify[ComplexExpand[ExpToTrig[resultat]] /. u/mm→k1 /. v/nn→k2]];*)
  Return[FullSimplify[ComplexExpand[ExpToTrig[resultat]] /. u / mm → k1 /. v / nn → k2]];
];

```

Beispiele 1D-DFT:

```

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

```

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

$$\frac{1}{3} (1 + 2 \cos[2 \pi k_1])$$

```

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

```

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

$$\cos[\pi k_1]^2$$

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

transferfunktion[{1 / 4, 1 / 2, -1 / 4}] // TrigReduce

$$\frac{1}{2} i (-i + \sin[2 \pi k_1])$$

binom[4]

transferfunktion[**binom**[4]]

transferfunktion[**binom**[4]] // TrigReduce

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

$$\cos[\pi k_1]^4$$

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

ableitungsymm

transferfunktion[**ableitungsymm**] // TrigReduce

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

$$i \sin[2 \pi k_1]$$

binom[10]

transferfunktion[**binom**[10]]

transferfunktion[**binom**[10]] // TrigReduce

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

$$\cos[\pi k_1]^{10}$$

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

1D-Plots von Transferfunktionen :

```
Show[(Print[TableForm@TrigReduce[#[[1]]]]];
      Plot[Evaluate[#[[1]]], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}, PlotStyle → Thick, PlotLegend → (MatrixForm[#] & /@ #[[2]]),
      LegendShadow → None, LegendPosition → {-0.75, -0.75}, LegendSize → {1.5, .5}, LegendOrientation → Vertical)] &[
      Transpose[{transferfunktion[#], MatrixForm[{#]}] & /@ {box[1], box[3], box[5], box[7], box[9], box[11]}}], ImageSize → 800]
```

1

$$\frac{1}{3} (1 + 2 \cos[2 \pi k_1])$$

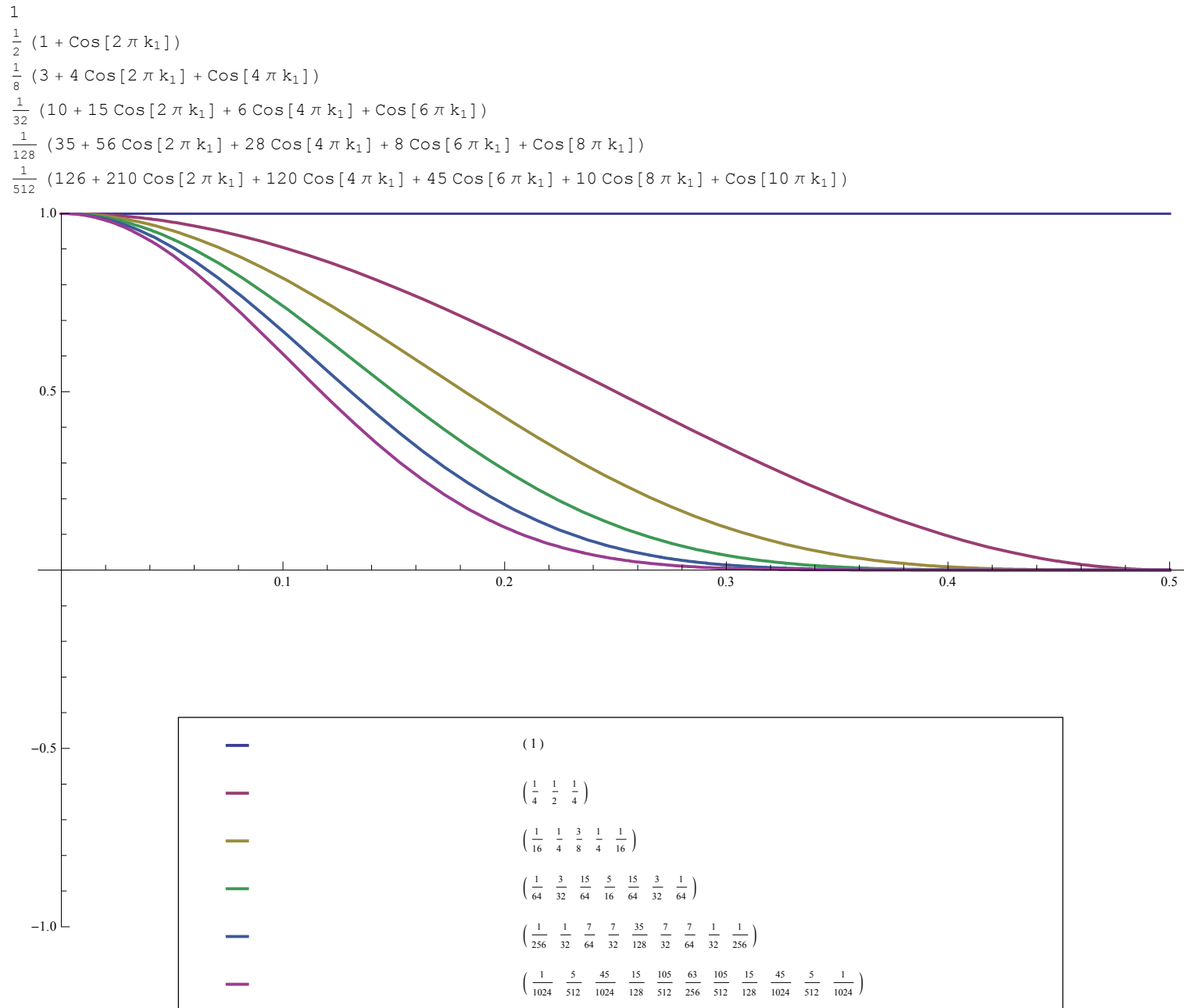
$$\frac{1}{5} (1 + 2 \cos[2 \pi k_1] + 2 \cos[4 \pi k_1])$$

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

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

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


```
Show[ (Print[TableForm@TrigReduce#[[1]]];
Plot[Evaluate#[[1]], {k1, 0, kmax}, PlotRange → {Full, {-1, 1}}, PlotStyle → Thick, PlotLegend → Evaluate[MatrixForm[#] & /@#[[2]]],
LegendShadow → None, LegendPosition → {-0.75, -0.75}, LegendSize → {1.5, .5}, LegendOrientation → Vertical]) &[
Transpose[{transferfunktion[#], MatrixForm[#]} & /@ {binom[0], binom[2], binom[4], binom[6], binom[8], binom[10]}], ImageSize → 800]
```



Beispiele 2D-DFT:

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

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

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

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

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

$$\cos[\pi k_1]^2 \cos[\pi k_2]^2$$

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

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

$$\cos[\pi k_1]^4 \cos[\pi k_2]^4$$

```
Transpose[{binom[2]}.{box[3]} // MatrixForm
transferfunktion2d[Transpose[{binom[2]}.{box[3]}]
```

$$\begin{pmatrix} \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix}$$

$$\frac{1}{3} \cos[\pi k_1]^2 (1 + 2 \cos[2 \pi k_2])$$

```
Clear[identität]
identität[größe_Integer] := Table[If[i == (größe + 1) / 2, 1, 0], {i, grösse}]
```

```
Transpose[{identität[3]}] . {binom[2]} // MatrixForm
transferfunktion2d[Transpose[{identität[3]}] . {binom[2]}]
```

$$\begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & 0 \end{pmatrix}$$

$$\cos[\pi k_2]^2$$

```
transferfunktion[box[3]]
```

$$\frac{1}{3} (1 + 2 \cos[2 \pi k_1])$$

```
(transferfunktion[box[3]] /. k1 -> k2)
```

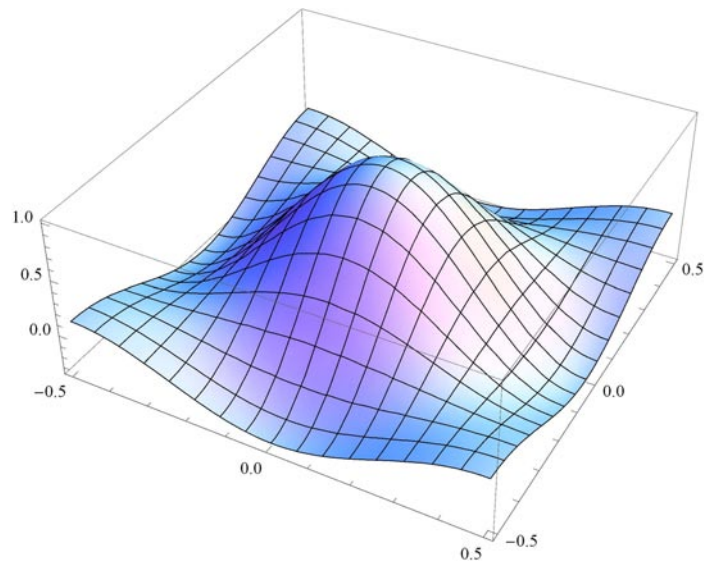
$$\frac{1}{3} (1 + 2 \cos[2 \pi k_2])$$

2D-Plots von Transferfunktionen :

```
transferfunktion[box[3]] * (transferfunktion[box[3]] /. k1 -> k2)
```

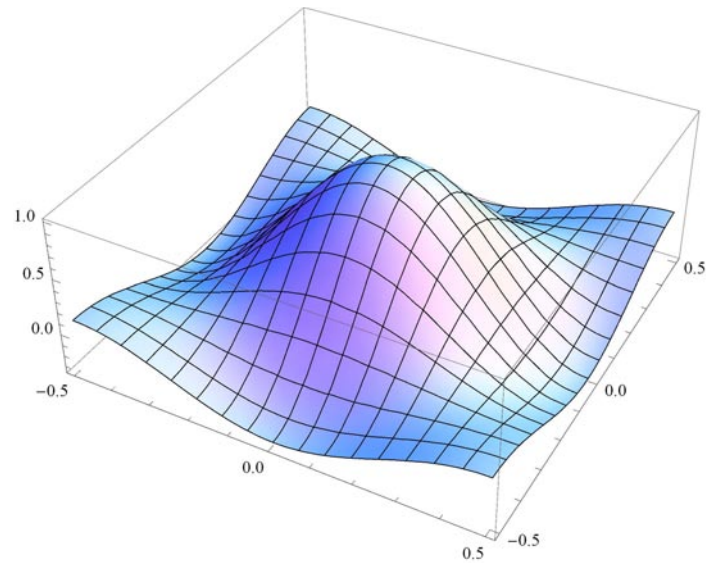
```
Plot3D[Evaluate[transferfunktion[box[3]] * (transferfunktion[box[3]] /. k1 -> k2)], {k1, -kmax, kmax}, {k2, -kmax, kmax}]
```

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



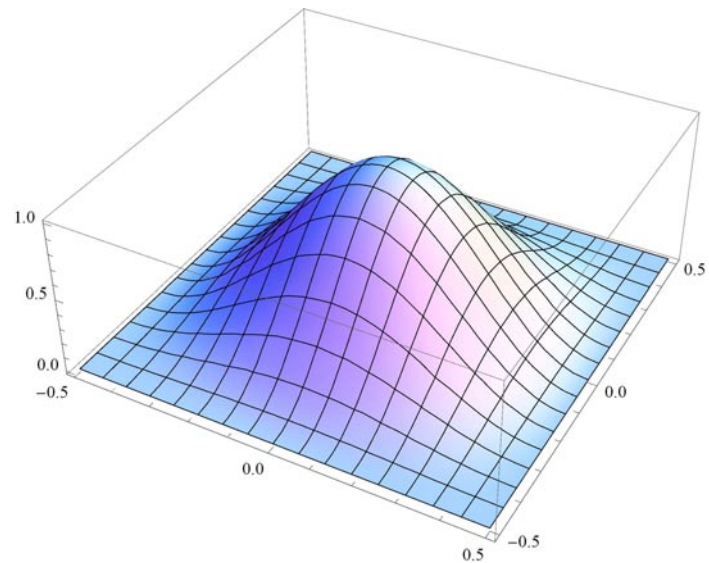
```
transferfunktion2d[Transpose[{box[3]}.{box[3]}]
Plot3D[Evaluate[transferfunktion2d[Transpose[{box[3]}.{box[3]}]], {k1, -kmax, kmax}, {k2, -kmax, kmax}]
```

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

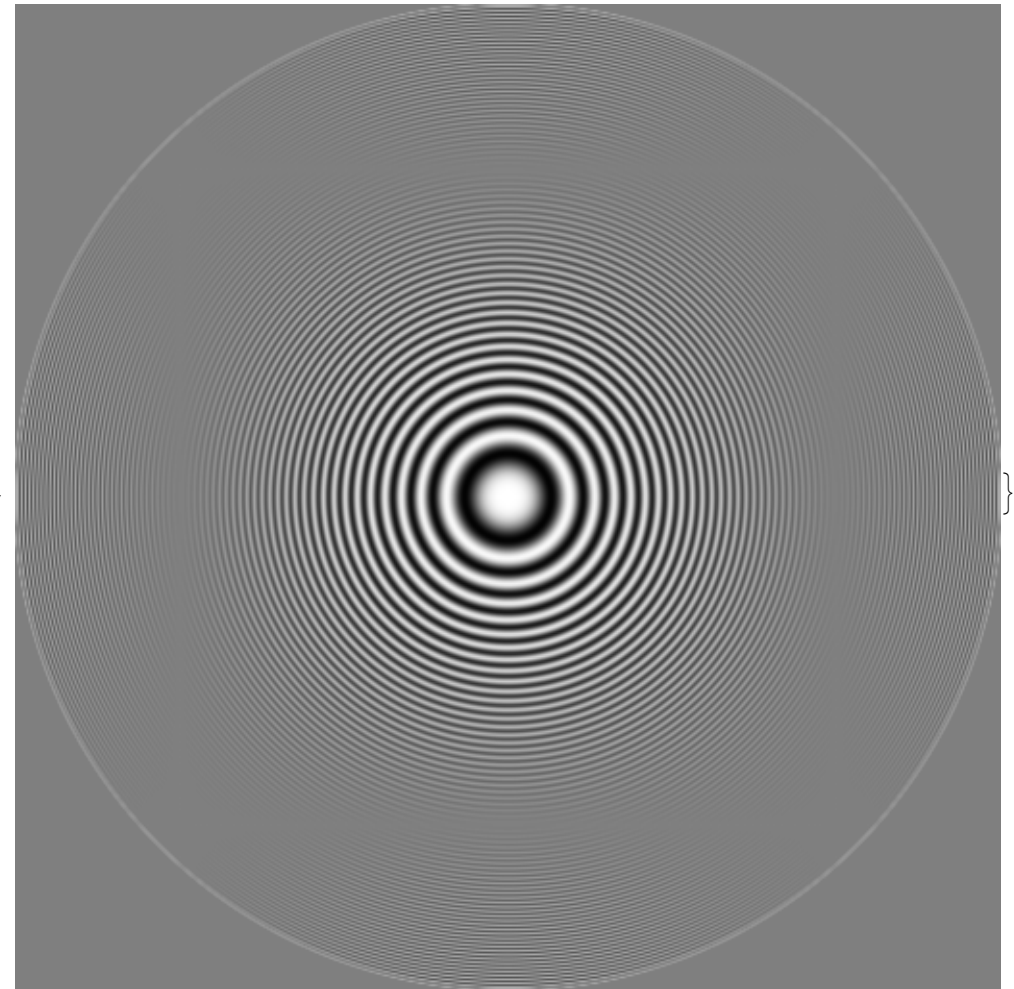
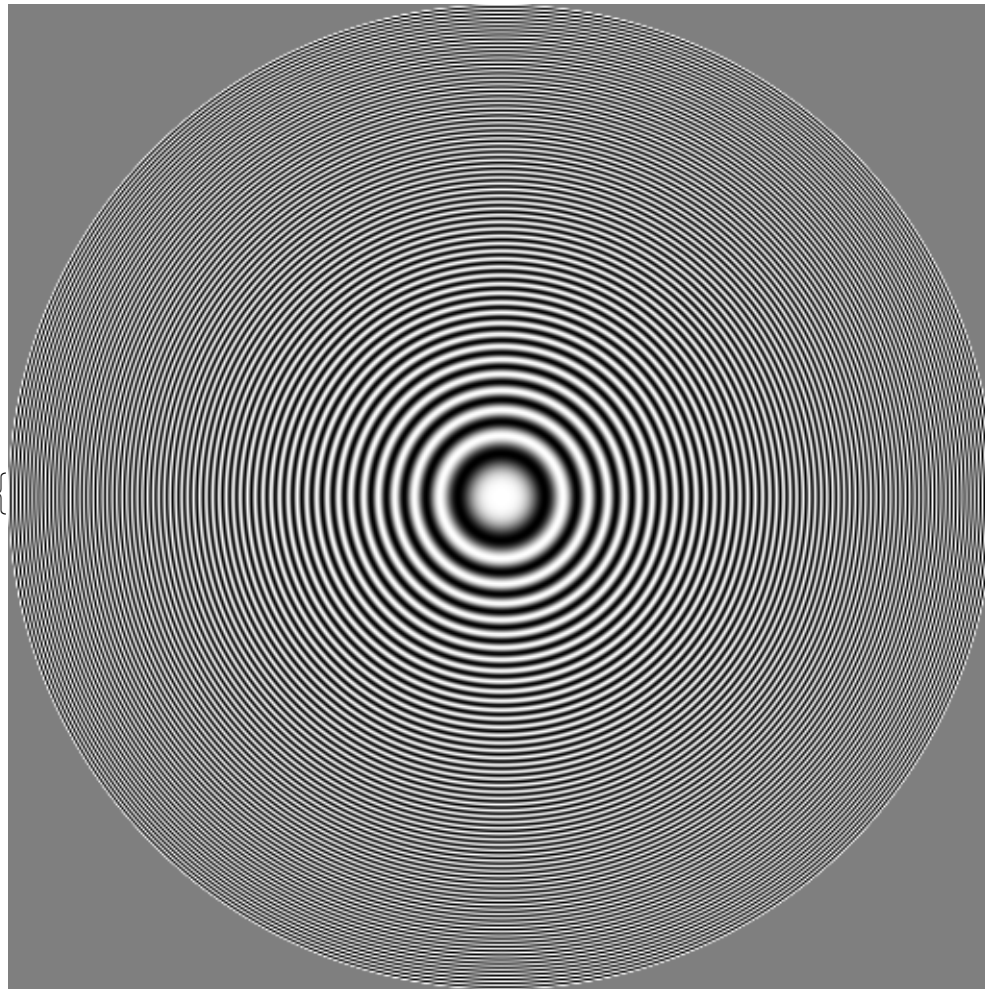


```
transferfunktion2d[Transpose[{binom[2]}.{binom[2]}]
Plot3D[Evaluate[transferfunktion2d[Transpose[{binom[2]}.{binom[2]}]], {k1, -kmax, kmax}, {k2, -kmax, kmax}]
```

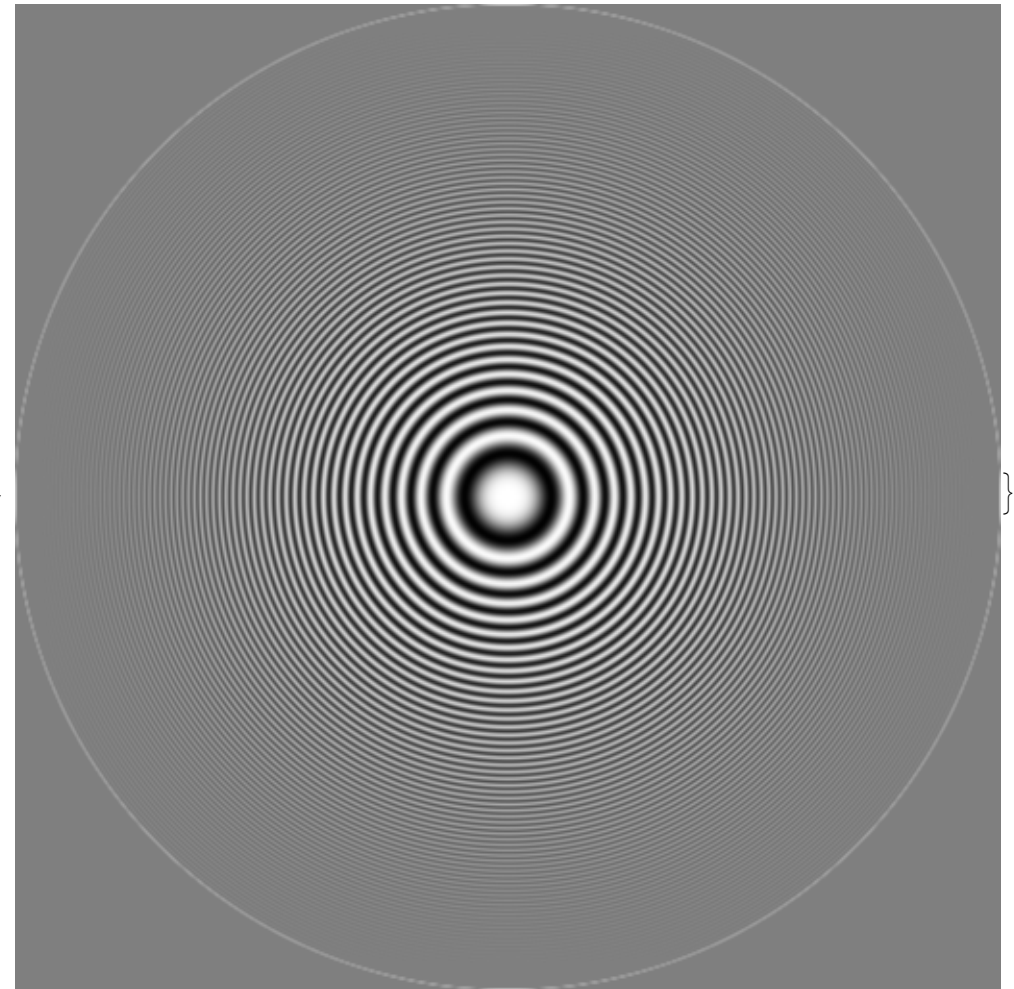
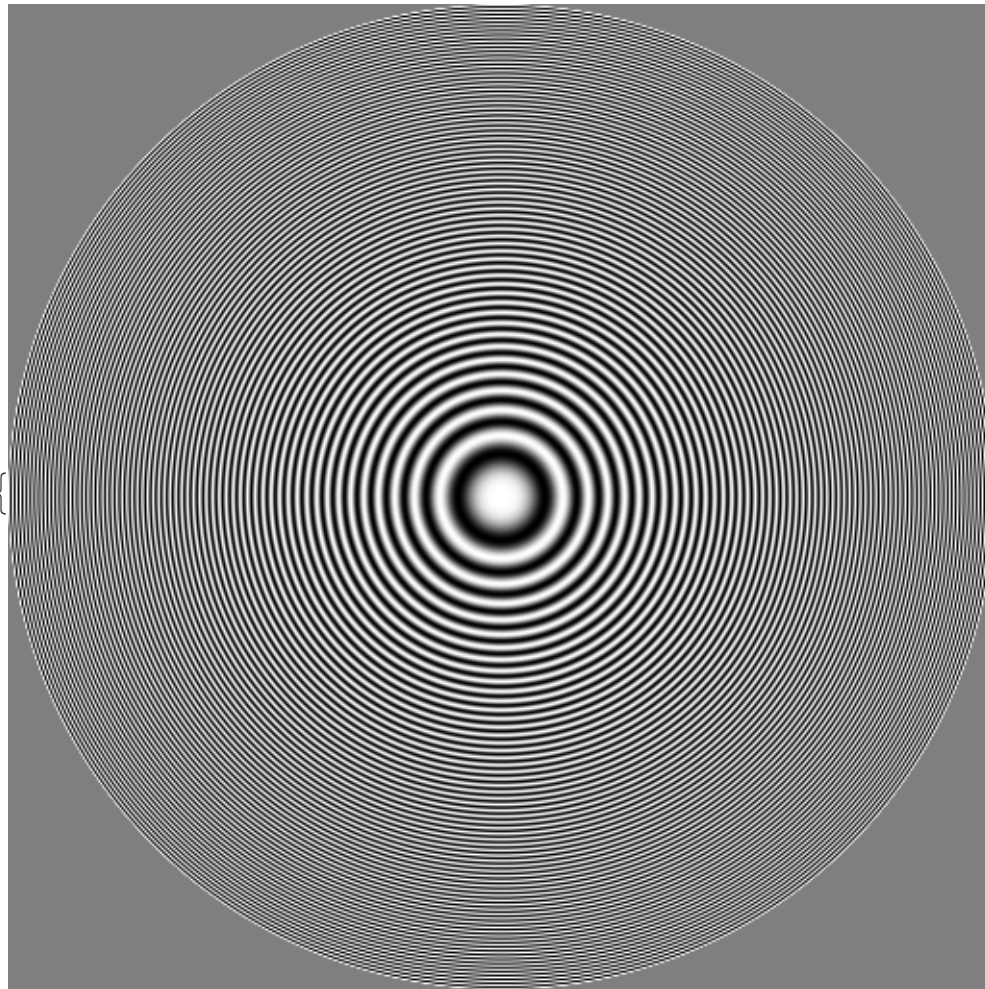
$$\cos[\pi k_1]^2 \cos[\pi k_2]^2$$



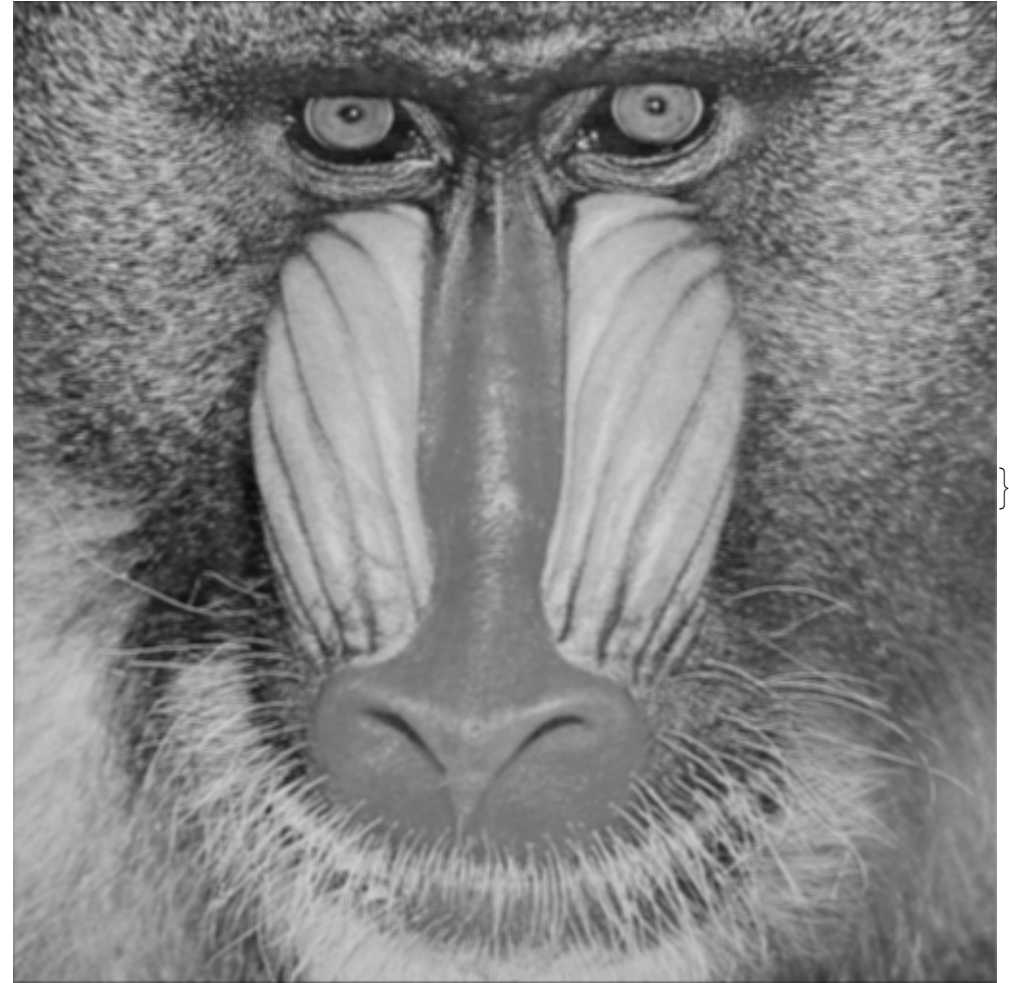
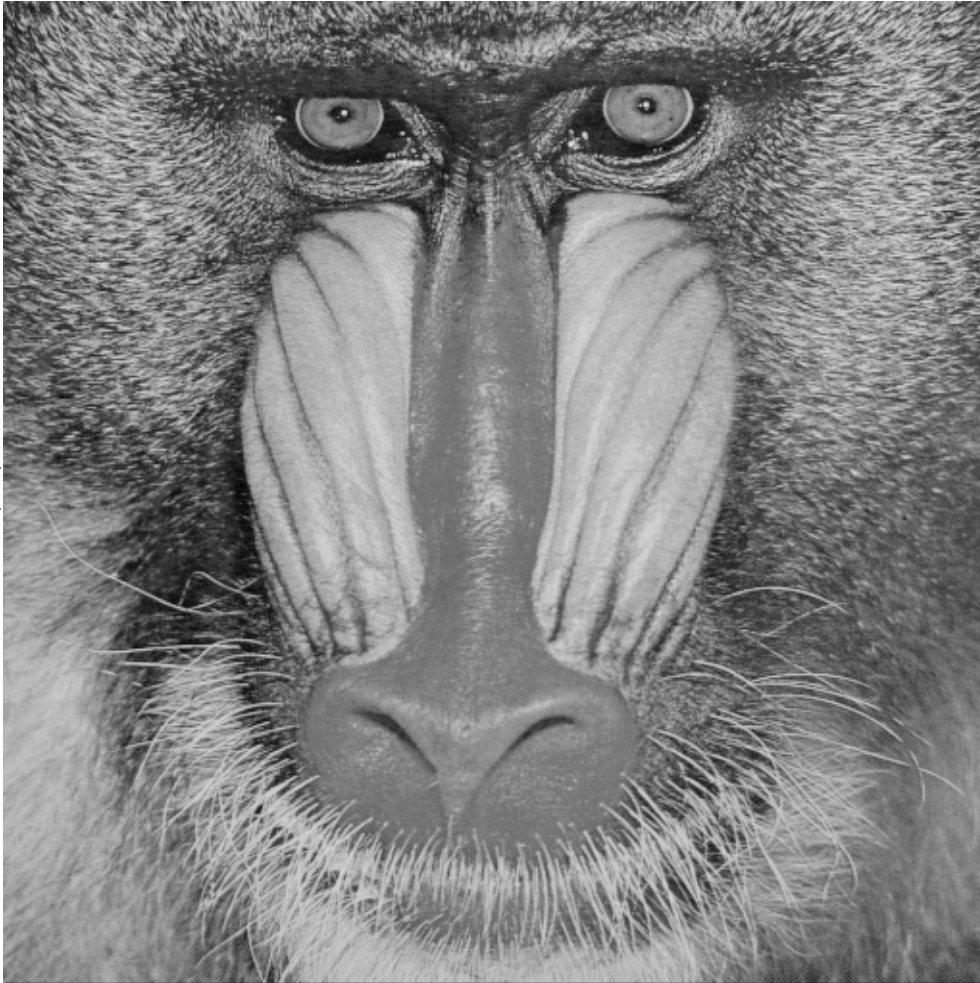
```
{Show[Image[#]], Show[Image[ListConvolve[Transpose[{box[3]}].{box[3]}, #, 2]]]} &[ImageData[wellenbild]]
```



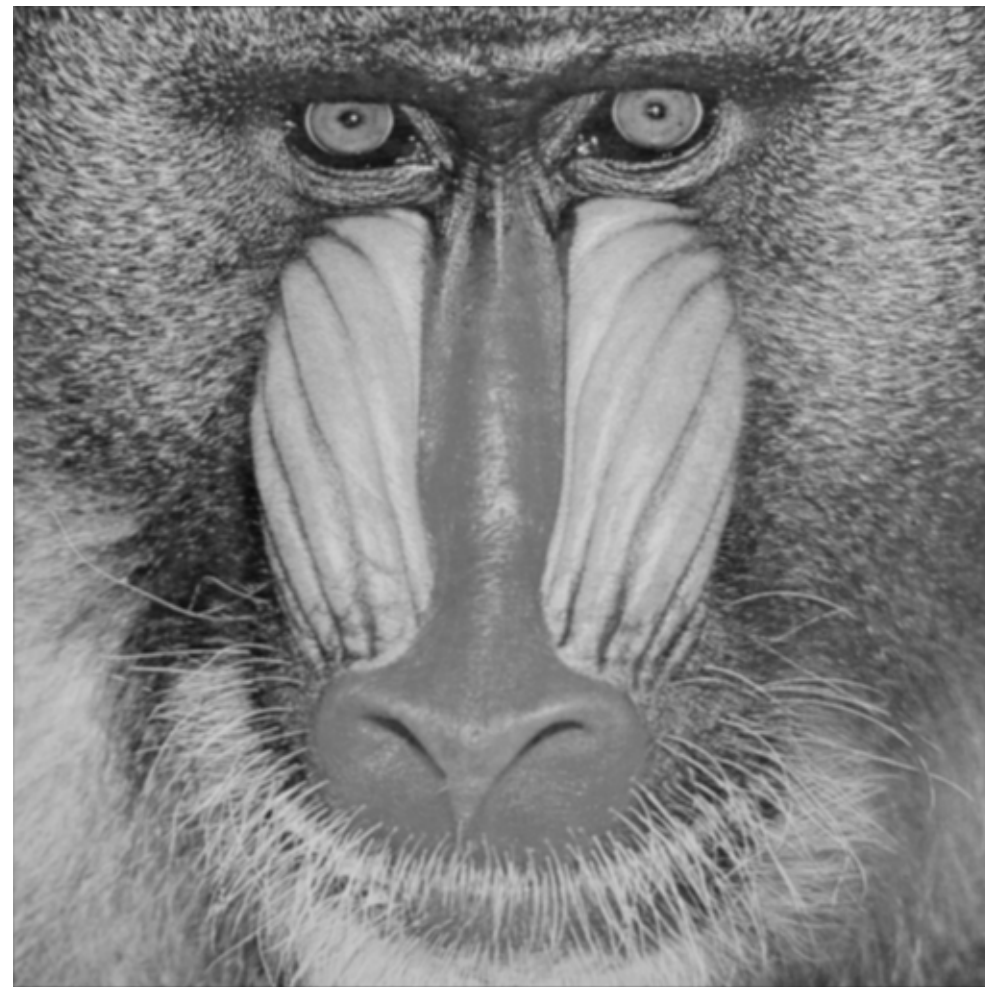
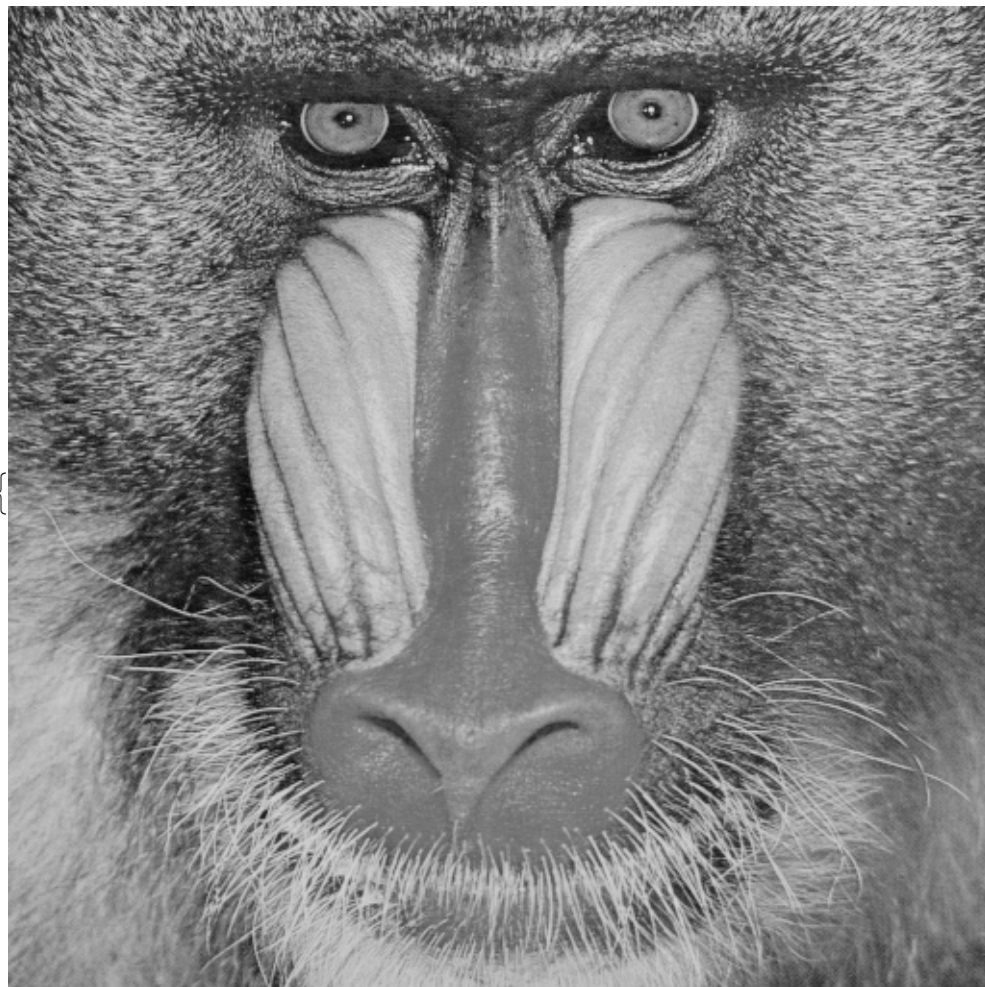
```
{Show[Image[#]], Show[Image[ListConvolve[Transpose[{binom[2]}].{binom[2]}, #, 2]]]} &[ImageData[wellenbild]]
```



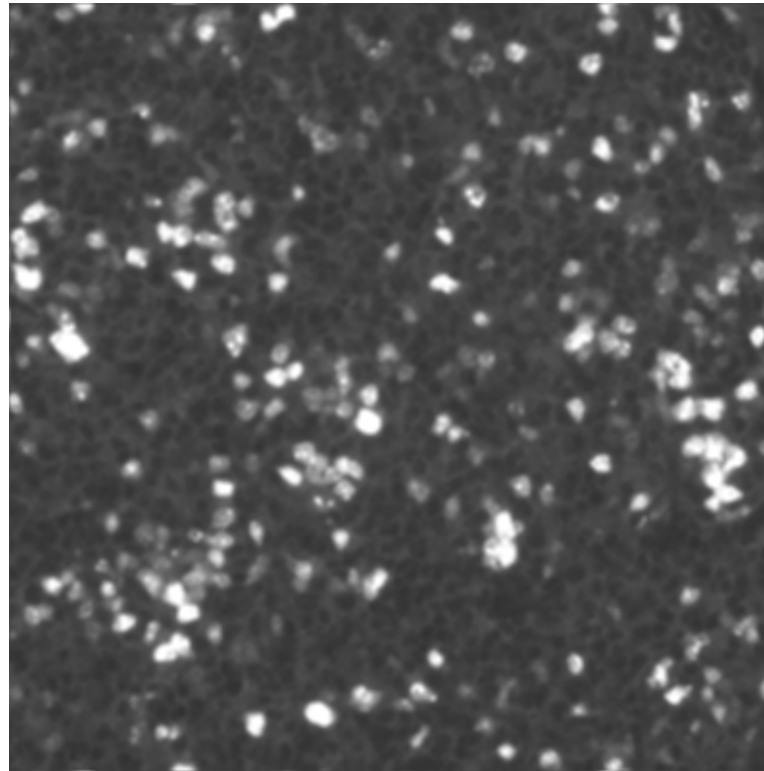
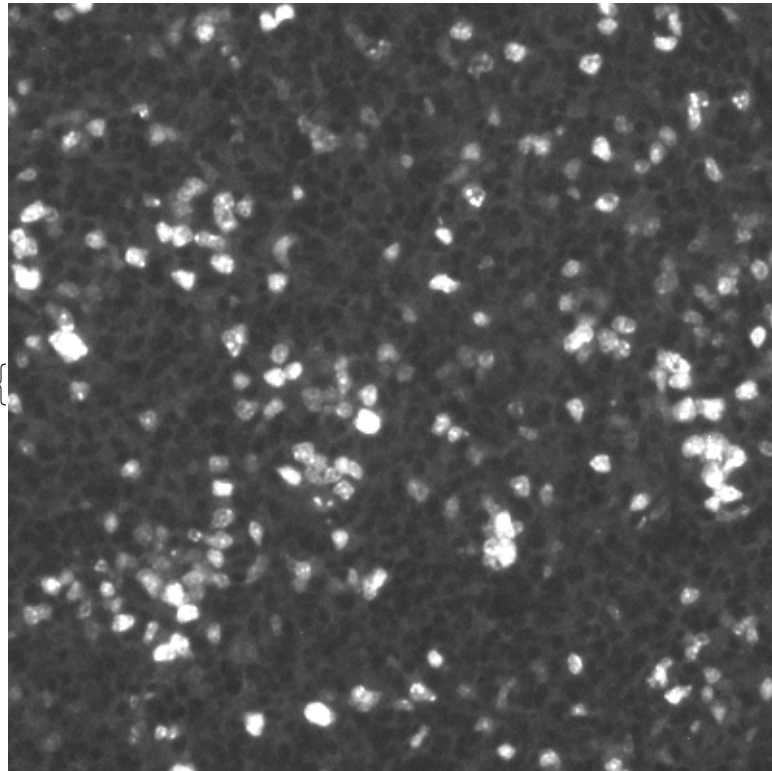
```
{Show[#], Show[Image[ListConvolve[Transpose[{box[3]}].{box[3]}, ImageData@#, {{2, 2}}]]]} &[  
ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Mandrill"}]]]
```



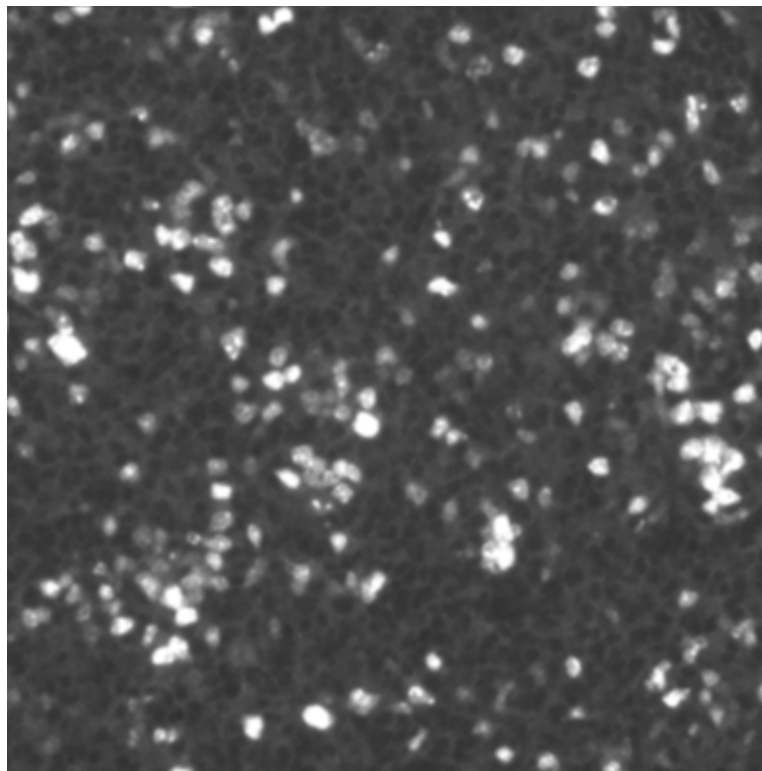
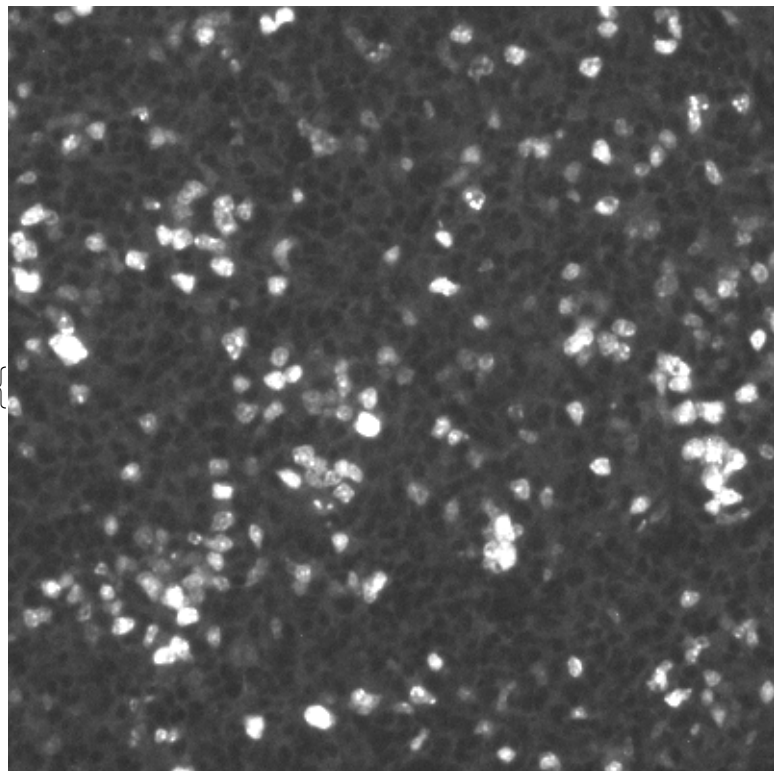
```
{Show[#], Show[Image[ListConvolve[Transpose[{binom[2]}].{binom[2]}, ImageData@#, {{2, 2}}]]]} &[  
ImageApply[{0.299, 0.587, 0.114}.# &, ExampleData[{"TestImage", "Mandrill"}]]]
```



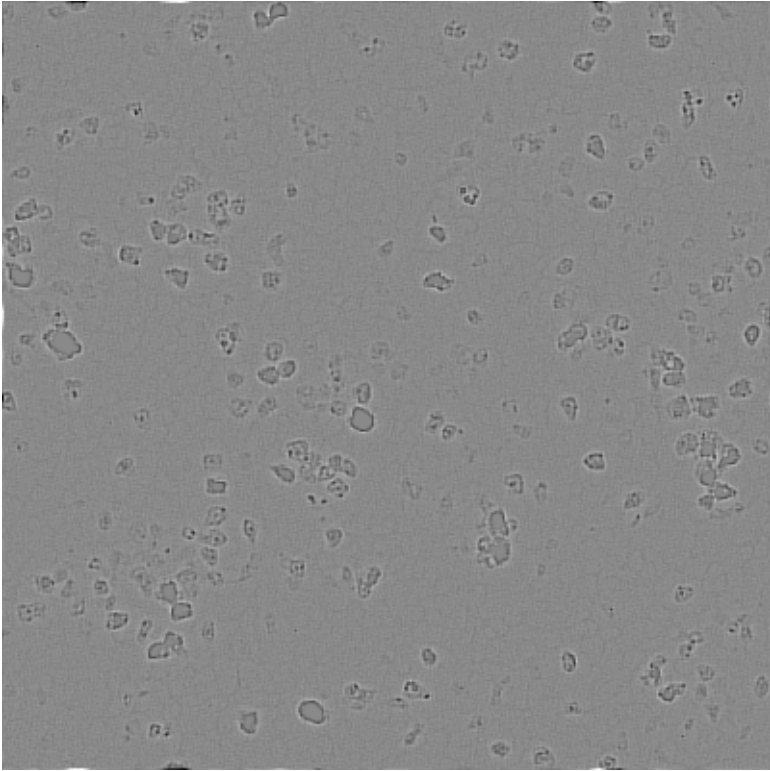
```
Show[#, ImageSize -> pagewidth / 2] & /@ ({#, Image[ListConvolve[Transpose[{box[3]}].{box[3]}, ImageData@#, {{2, 2}}]]} &[  
  ImageCrop[First@ColorSeparate@Lym3CD21dreikanalausgleichF2, {pagewidth / 2, pagewidth / 2}]])
```



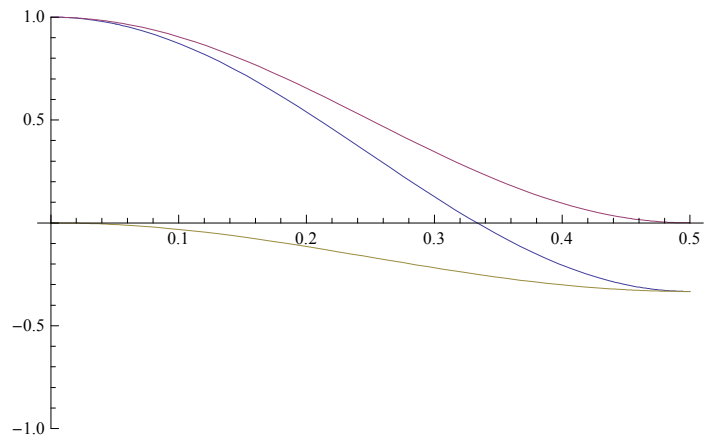
```
Show[#, ImageSize -> pagewidth / 2] & /@ ({#, Image[ListConvolve[Transpose[{binom[2]}].{binom[2]}, ImageData@#, {{2, 2}}]}] &[
  ImageCrop[First@ColorSeparate@Lym3CD21dreikanalausgleichF2, {pagewidth / 2, pagewidth / 2}]])
```



```
Show[#, ImageSize -> pagewidth / 2] &@
(ImageAdjust@ImageSubtract[Image[ListConvolve[Transpose[{box[3]}].{box[3]}, ImageData@#, {{2, 2}}]], Image[
ListConvolve[Transpose[{binom[2]}].{binom[2]}, ImageData@#, {{2, 2}}]]] &[
ImageCrop[First@ColorSeparate@Lym3CD21dreikanalausgleichF2, {pagewidth / 2, pagewidth / 2}]])
```

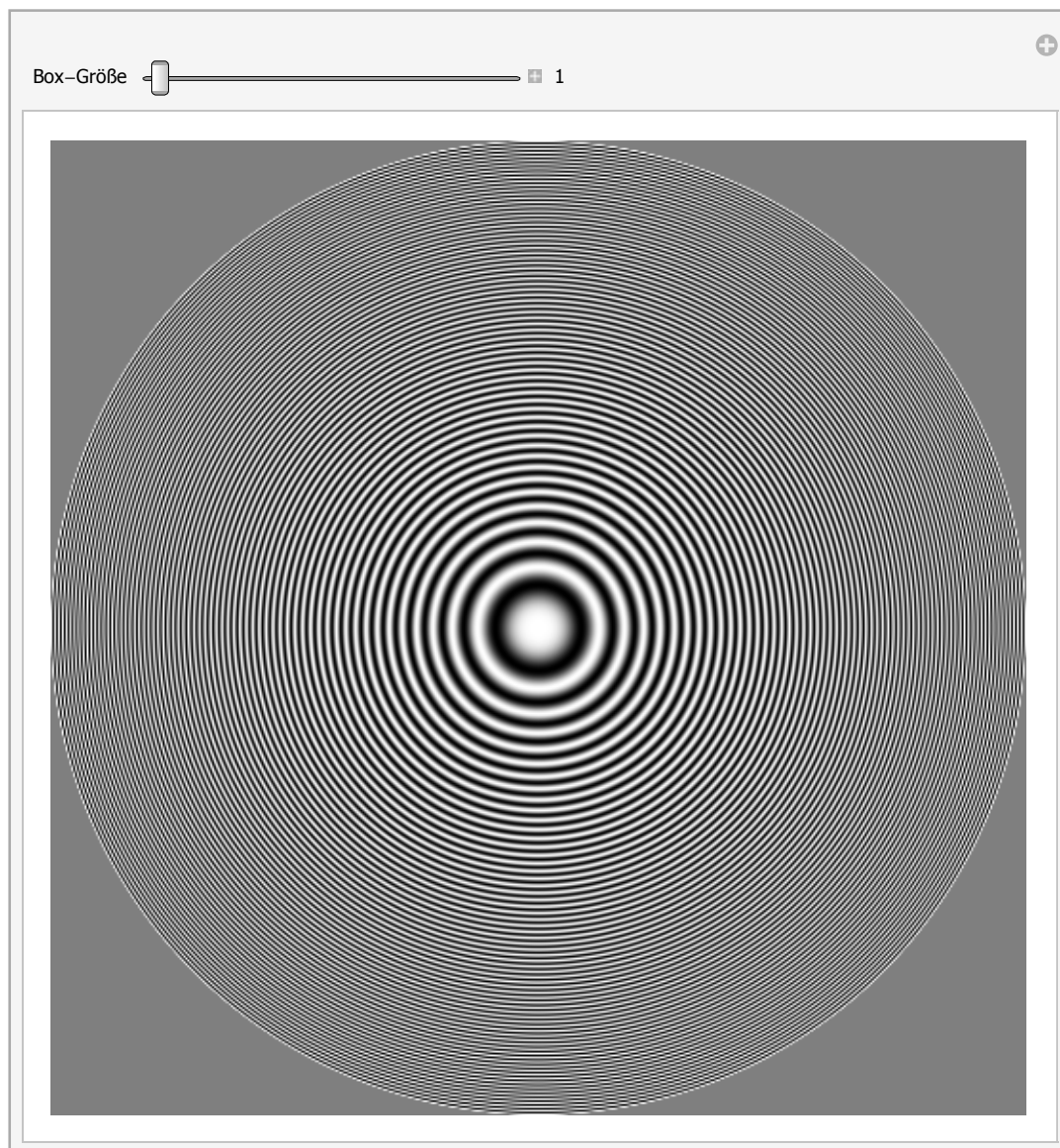


```
Plot[{1 / 3 * (1 + 2 * Cos[2 π k]), 1 / 2 (1 + Cos[2 k π]), 1 / 3 * (1 + 2 * Cos[2 π k]) - 1 / 2 (1 + Cos[2 k π])},
{k, 0, maxwellenzahl}, PlotRange -> {Full, {-1, 1}}]
```



```
r = 256;
```

```
Manipulate[Show[Image[ListConvolve[Transpose[{box[größe]}] . {box[größe]}, ImageData@wellenbild, (größe + 1) / 2]], ImageSize → 2 * r + 1],  
{größe, 3, "Box-Größe"}, 1, 21, 2, Appearance → "Labeled", SaveDefinitions → True]
```



```
Manipulate[Show[Image[ListConvolve[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]
```

