

Berliner Hochschule für Technik
Fachbereich VI - Informatik und Medien

Laborprotokoll

Bildverarbeitung im Original- und Frequenzbereich

Name, Vorname:	Weis, Constantin
Matrikelnummer:	928684
Name, Vorname:	Mu, Zetian
Matrikelnummer:	930673
Fachrichtung:	Informatik
Studiengang:	Technische Informatik - Embedded Systems
Modul:	M03 - Entwurf eingebetteter Systeme
Dozent/-in:	Prof. Dr.-Ing. Peter Gregorius
Abgabedatum:	20.09.2022

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	2
2.1	Filtern im Originalbereich	2
2.2	Paddingarten	3
2.3	Filtern im Frequenzbereich	5
3	Filterarten	9
3.1	Tiefpass	9
3.2	Hochpass	15
4	Bild-Segmentierung	21
5	Ausblick	23

1 Einleitung

Durch die steigende Tendenz zu autonomen Systemen wird Bildverarbeitung immer essentiell wichtiger. Deswegen wurde entschieden, innerhalb des Labors die Bildverarbeitung genauer unter die Lupe zu nehmen. Im Rahmen dieses Berichts werden zunächst die Grundlagen für die Filterung im Original- und Frequenzbereich beleuchtet. Anschließend werden einige Filterarten des Tief- und Hochpasses analysiert und mithilfe von Python implementiert, um die Anwendung auf Testbilder zu untersuchen. Das Ziel des gesamten Projektes ist Kantendetektion später auf einem Xilinx-Zybo System-On-Chip zu implementieren, um Objekte zu erkennen. Dafür können die bereits implementierten Algorithmen in Python als Hilfe verwendet werden.

2 Grundlagen

2.1 Filtern im Originalbereich

Die Prozesse der digitalen Bildverarbeitung im Originalbereich lassen sich mit der Gleichung 1 zu beschreiben.

$$g[x, y] = T(f[x, y]) \quad (1)$$

Dabei ist die $f[x, y]$ das originale Bild und $g[x, y]$ das Ergebnisbild. Die Funktion $T()$ beschreibt der angewendete Operator. Wenn der Operator auf einem einzelnen Pixel wirkt, ist dieser ein Punktoperator. Mit einem Punktoperator lässt sich beispielsweise die Helligkeit des Bildes erhöhen. Um mehr komplexe Wirkung zu erreichen, ist es sinnvoll, ein Filter zu verwenden. Ein Filter ist auch ein Operator, welcher auf einem Bereich des Bildes wirkt. Er basiert auf die mathematische Grundlage: Faltung.

Die Faltung berechnet die Wirkung eines Filters auf einem Eingangssignal und ergibt einem daraus resultierten Signal wieder. In der Bildverarbeitung ist der Prozess sehr ähnlich. Das Eingangssignal ist eine Matrix des originalen Bildes $f[x, y]$ und der Operator $h[s, t]$ wirkt als ein Filter. Somit ist das resultierte Bild $g[x, y] = (h * f)$. Dabei wird die 2D-Faltung beschrieben in der Gleichung 2 angewendet.

$$g[x, y] = (h * f)[x, y] = \sum_{s=-a}^a \sum_{t=-b}^b h[s, t] f[x - s, y - t] \quad (2)$$

Ähnlich wie die 1D-Faltung muss der Operator bei der 2D-Faltung zuerst gespiegelt werden. Danach wird der Operator durch die Eingangsmatrix in der Zeilen- und Spalten-Richtung iteriert. Bei jeder Iteration wird eine Submatrix in der Größe des Operators an der Iterationsposition des Operators aus der Eingangsmatrix abgebildet. Danach wird die Summe der Partialprodukte zwischen der Submatrix und dem Operator gebildet. Diese Summe wird als Resultat für das Bildpixel der aktuellen Iteration in der Ausgangsmatrix gespeichert. Sobald die Iteration beendet ist, entsteht somit die vollständige Ausgangsmatrix. Die Anfangsiteration einer 2D-Faltung ist in Abbildung 1 dargestellt.

Bei der Anwendung eines Operators an einer Eingangsmatrix ist der Informationsverlust zu beachten. Mit der Faltung ist die Größe der Ausgangsmatrix kleiner als die Größe der Eingangsmatrix. Die Größe der Ausgangsmatrix ($U \times T$) lässt sich mit der Gleichung 3 und 4 berechnen. Dabei beträgt die Größe der Eingangsmatrix ($M \times N$) und die Größe des Operators ist ($S \times T$).

$$U = M - S + 1 \quad (3)$$

$$V = N - T + 1 \quad (4)$$

Um dem Informationsverlust zu vermeiden, ist das Padding vor der 2D-Faltung durchzuführen. Das Padding fügt jeweils Zeilen und Spalten vor und nach der Eingangsmatrix hinzu. Es vergrößert damit die Eingangsmatrix, um die Größe der Ausgangsmatrix und die Größe der Eingangsmatrix gleich zu halten. Die Anzahl der erweiterte Zeilen m und Spalten n ist abhängig von der Größe der angewendete Operator ($S \times T$) und die lassen

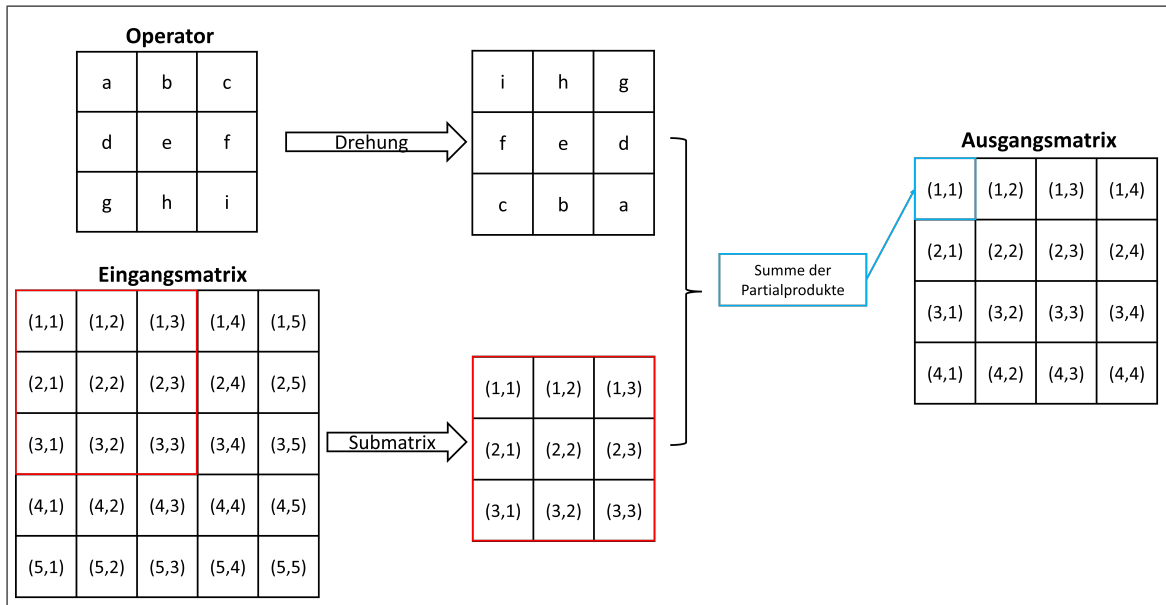


Abbildung 1: Anfangsiteration der 2D-Faltung

sich mit der Gleichung 5 und 6 berechnet. Die Erweiterung erfolgt allerdings an vier Rändern des originalen Bildes. Somit werden $m/2$ Zeilen oberhalb und unterhalb der Eingangsmatrix hinzugefügt. Außerdem wird die Eingangsmatrix um jeweils $n/2$ Spalten in der Richtung links und rechts erweitert. Die mögliche Paddingverfahren werden im Kapitel 2.2 detaillierter vorgestellt.

$$m = S - 1 \quad (5)$$

$$n = T - 1 \quad (6)$$

Somit entsteht das Vorgehensweise beim Filtern eines Bildes im Originalbereich. Diese wird in der Abbildung 2 dargestellt.

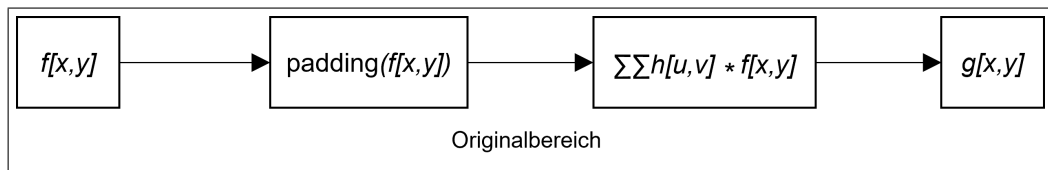


Abbildung 2: Vorgehensweise beim Filtern eines Bildes im Originalbereich

2.2 Paddingarten

Im Bildbereich lässt sich das Padding durch verschiedene Möglichkeiten anzuwenden. In diesem Kapitel werden insgesamt drei Paddingarten dargestellt: Zero-Padding, Mirror-Padding und Replicate-Padding.

Bei Zero-Padding werden die erweiterte Zeilen und Spalten mit Nullen gefüllt. Dabei entstehen einem schwarzen Rand um originalen Bild herum. Bei der Mirror-Padding handelt

es sich um die Verspiegelung der erweiterten Zeilen und Spalten entlang die Rändern des originalen Bildes. Dieser Paddingart ist am häufigsten eingesetzt, wenn die Details am Rand eines Bildes beibehaltet werden soll. Beim Replicate-Padding werden die erweiterte Pixeln mit der Randpixeln des originalen Bildes ausgefüllt. [1] Die Effekte der drei Paddingarten werden in der Abbildung 3 dargestellt.

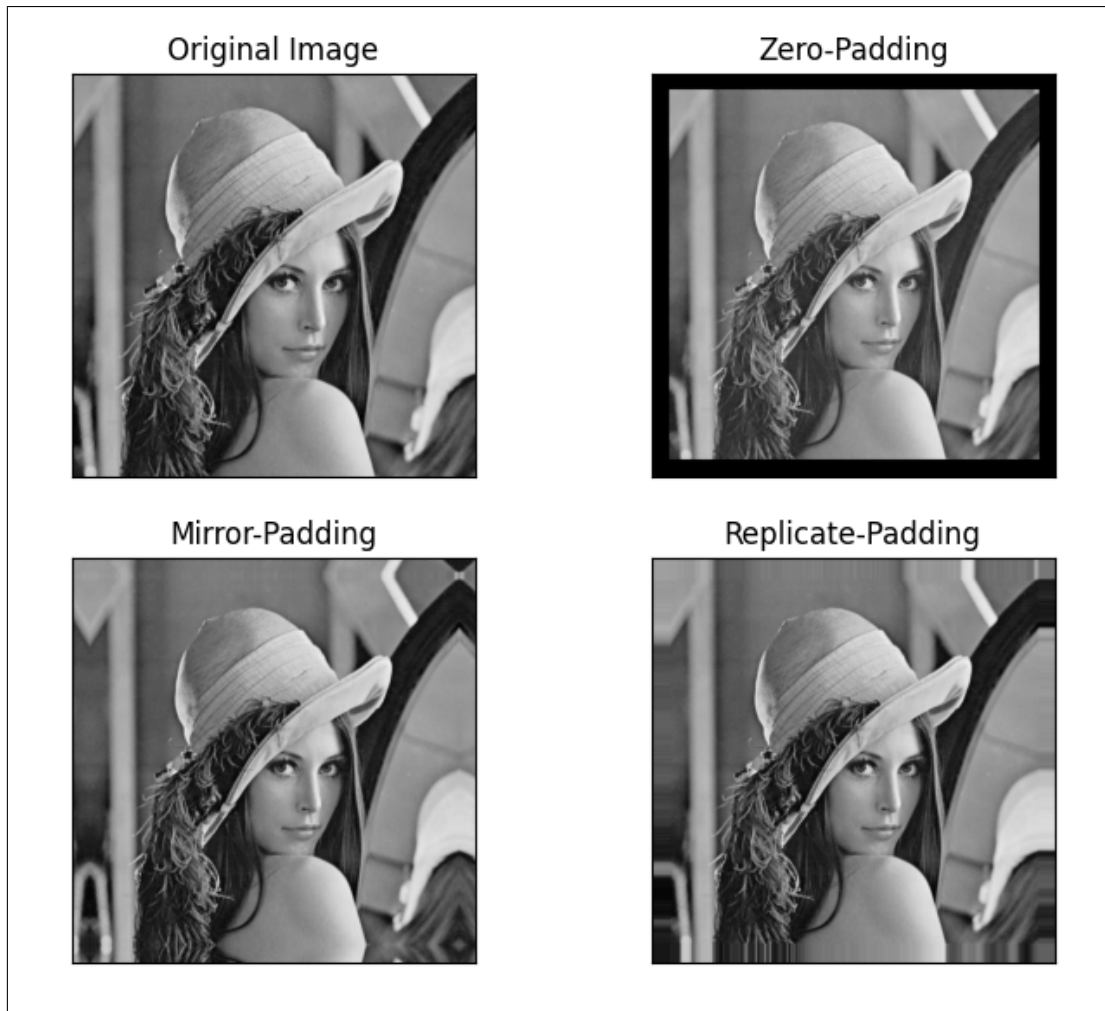


Abbildung 3: Das Bild "Lena" mit verschiedenen Paddingarten

2.3 Filtern im Frequenzbereich

In Abbildung 4 ist die prinzipielle Vorgehensweise beim Filtern eines Bildes im Frequenzbereich dargestellt. Zunächst liegt das digitale Bild $f[x, y]$ im Originalbereich vor und wird um eine bestimmte Anzahl von Bildpunkten erweitert. Diese Prozedur wird als Padding bezeichnet, auf die später noch genauer eingegangen wird. Die Transformation des Bildes in den Frequenzbereich unterliegt der diskreten 2-D Fouriertransformation (DFT) und wird mit folgender Gleichung ermittelt [1]:

$$F[u, v] = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f[x, y] e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (7)$$

Umgekehrt kann auch aus der Frequenzdarstellung das digitale Bild gewonnen werden. Dafür kommt die inverse diskrete 2-D Fouriertransformation (iDFT) zum Einsatz [1]:

$$f[x, y] = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F[u, v] e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (8)$$

Die Bildmaße $M \times N$, wobei M für die Zeilen- und N für die Spaltenanzahl steht, gibt an, wie oft die Gleichungen für eine Transformation berechnet werden müssen. Der Iterationsbereich lautet folgendermaßen: $u, x = 0, 1 \dots M-1$ und $v, y = 0, 1 \dots N-1$. Für jeden Bildpunkt müssen beide Summenformeln berechnet werden. Für Bilder mit sehr vielen Bildpunkten steigt der Rechenaufwand demnach exorbitant an. Deshalb wird die DFT mit der Fast Fourier Transformation (FFT) bestimmt, die eine Berechnung in viel kürzerer Zeit ermöglicht. In Abbildung 5 ist der Geschwindigkeitsvorteil $C(p)$ bei der Berechnung mit der FFT gegenüber der 1-D DFT dargestellt. Der Parameter p ist ausschlaggebend für die Anzahl der zu transformierenden Bildpunkte und wird in der Gleichung $M = 2^p$ eingesetzt. Wie in der Abbildung zu erkennen ist, steigt der Geschwindigkeitsvorteil mit größer werdendem p exponentiell an. Soll exemplarisch ein 1-D Vektor mit 1024 Punkten in den Frequenzbereich transformiert werden, ist die FFT 102-mal schneller als die DFT. Die Anwendung der FFT auf eine 2-D Bildmatrix erfolgt zunächst durch das Transformieren aller Zeilen und anschließend aller Spalten.

Nachdem das Bild in den Frequenzbereich konvertiert wurde, können bestimmte Frequenzen gefiltert werden. Typische Filter sind Lowpass-, Highpass-, Bandpass- oder Bandsperrfilter. Die Filterung erfolgt über die elementweise Multiplikation mit einer Filterfunktion $H[u, v]$. Anschließend kann das modifizierte Bild mithilfe der iDFT in den Originalbereich umgewandelt werden. Der letzte Schritt ist das Entfernen der beim Padding hinzugefügten Bildpunkte, um die Ursprungsgröße des Bildes wieder zu erhalten. Die Schritte für das Filtern im Frequenzbereich sind in der Gleichung 9 mathematisch zusammengefasst (ohne Padding).

$$g[x, y] = \text{real} \left[\mathcal{F}^{-1} \left\{ \mathcal{F} \{ f[x, y] \} \cdot H[u, v] \right\} \right] \quad (9)$$

Die Gleichung entspricht der zyklische Faltung im Originalbereich, die durch das Falten zweier periodisch fortgesetzter Funktionen charakterisiert wird. Bei der Transformation in den Frequenzbereich kommt ebenfalls eine Periodizität zustande, die durch die DFT

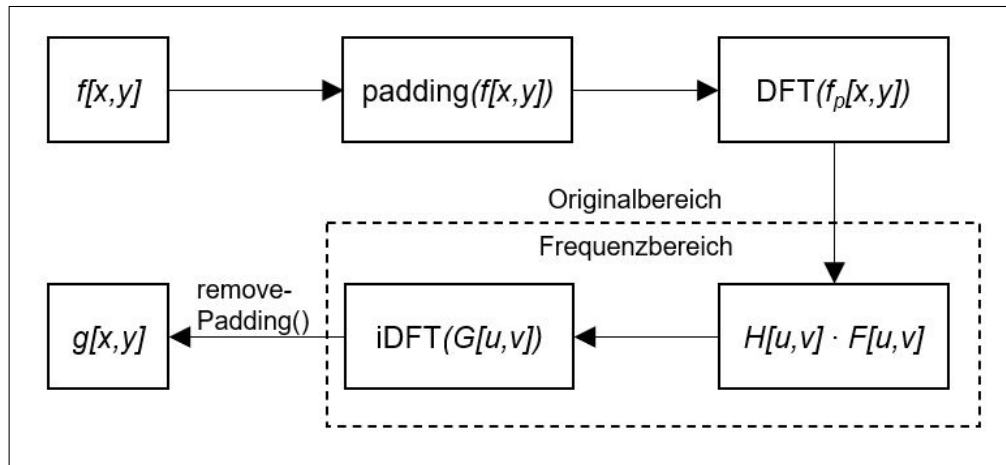


Abbildung 4: Vorgehensweise beim Filtern eines Bildes im Frequenzbereich

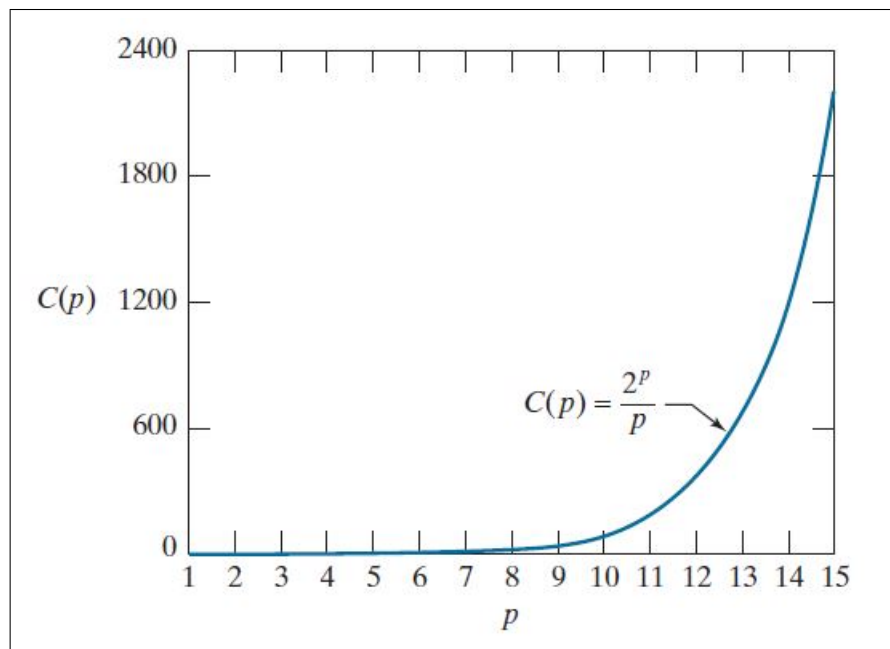


Abbildung 5: Geschwindigkeitsvorteil $C(p)$ der FFT gegenüber der 1-D DFT [1]

verursacht wird. Eine Annahme für die DFT ist nämlich, dass es sich um eine periodische Funktion handelt. Um die Diskrepanz zwischen der linearen und zyklischen Faltung zu vermeiden, wird das Zero-Padding durchgeführt. Durch das Erweitern des Bildes um Nullen, wird die direkte Periodizität des Bildes unterbrochen und das Bildende wird nicht mehr durch den Bildanfang beeinflusst, wie es bei der zyklischen Faltung der Fall ist. Der ohne Padding auftretende Fehler wird als Wraparound-Fehler bezeichnet. Laut [1] soll die Größe des Bildes $M \times N$ und die Filterfunktion $C \times D$ entsprechend den Gleichungen erweitert werden:

$$P \geq M + C \quad (10)$$

$$Q \geq N + D \quad (11)$$

$P \times Q$ geben die neuen Maße des Bildes und der Filterfunktion an. Alle elementaren Schritte sind exemplarisch in Abbildung 6 zu sehen (ohne Padding). Ob im Frequenzbereich gefiltert werden sollte, hängt von verschiedenen Aspekten ab. Ein wesentlicher Punkt ist, dass bestimmte Artefakte, wie zum Beispiel Interferenzmuster auf dem Bild, im Frequenzbereich einfacher analysiert und entfernt werden können. Weiterhin spielt die genutzte Größe der Filterfunktion eine Rolle. Bei einer kleinen Filtergröße ist die Faltung im Originalbereich zu präferieren. Übersteigt die Filtergröße jedoch einen bestimmten Wert, ist die Multiplikation im Frequenzbereich aufgrund des steigenden Geschwindigkeitsvorteils bei der Berechnung zu bevorzugen [1].

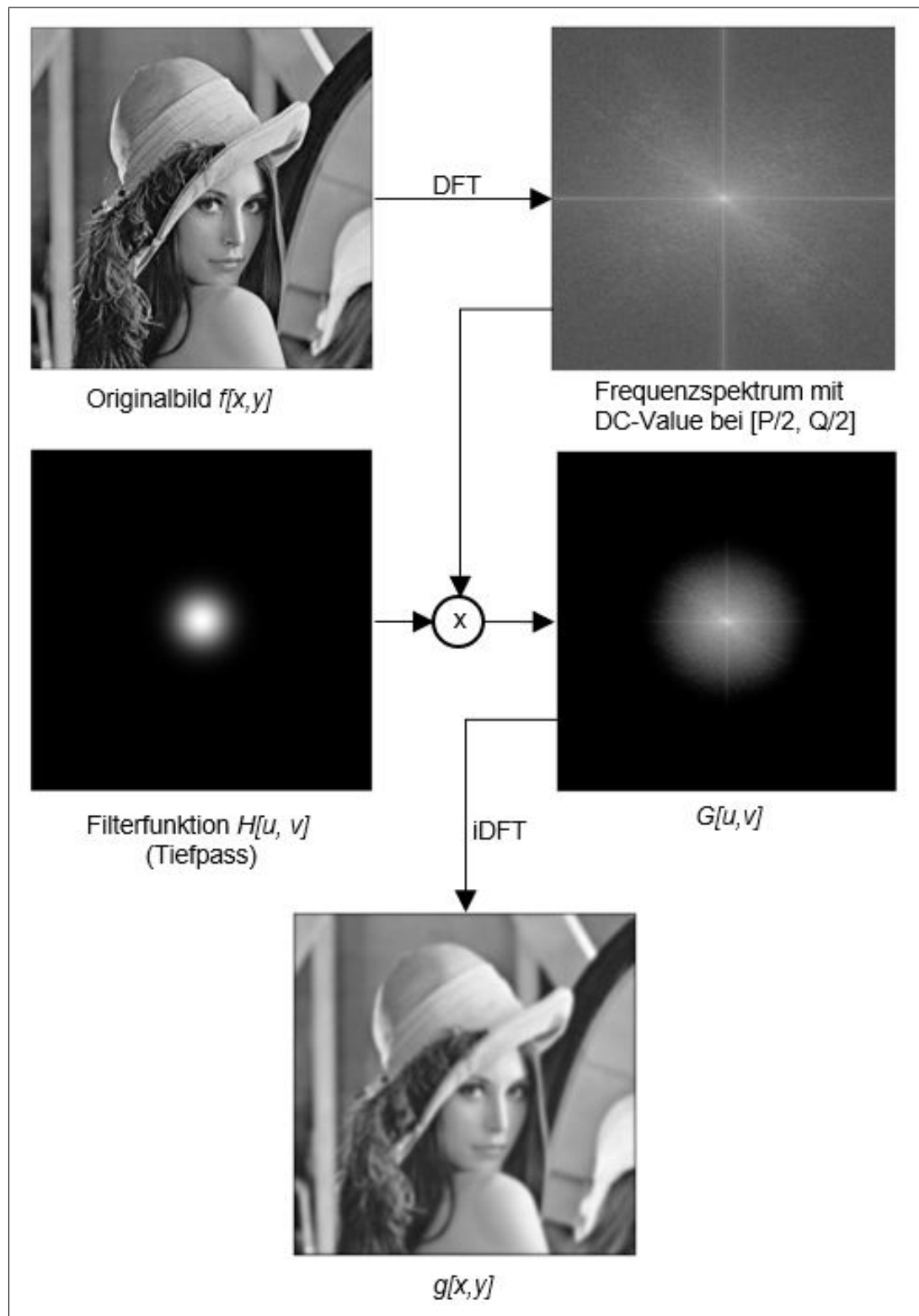


Abbildung 6: Filterung eines Bildes im Frequenzbereich

3 Filterarten

Filter erlauben es, bestimmte Frequenzen in einem Bild zu dämpfen. Die Frequenzen in einem Bild entstehen durch Intensitäts- respektive Farbwertänderungen zwischen einzelnen Bildpunkten. Abrupte Überänge, die zum Beispiel durch Rauschen oder Kanten verursacht werden, tragen hohe Frequenzen zum Spektrum bei, da sich bei Kanten die Farbwerte stark ändern. Die größtmögliche Frequenz entsteht beim Übergang von Schwarz zu Weiß. Schwach ändernde Farbwerte, wie zum Beispiel: eine fast einfarbige Wand, sorgen für niedrige Frequenzen. Nachfolgend werden einige Filter vorgestellt und die Auswirkung auf Testbilder untersucht. Die Implementierung der Filter erfolgt mit Python. Für alle nachfolgenden Betrachtungen im Frequenzbereich wird der DC-Wert des Frequenzspektrums in die Bildmitte verschoben.

3.1 Tiefpass

Tiefpassfilter werden für die Filterung von hohe Frequenzen angewendet. Dies führt dazu, dass abrupte Übergänge sowie Rauschen reduziert werden. Die Folge der Filterung ist eine Trübung des Bildes. Dieser Effekt wird nachfolgend anhand eines idealen Filters (Frequenzbereich) und eines Gaussfilters (Original-, Frequenzbereich) gezeigt.

Frequenzbereich

Der ideale Tiefpassfilter $H_{ILP}[u, v]$ wird durch die totale Dämpfung von Frequenzen oberhalb der Grenzfrequenz gekennzeichnet. Auf den Frequenzbereich bezogen, lässt der Filter in einem bestimmten Abstand um den DC-Wert alle Frequenzen passieren. Er kann wie folgt definiert werden [1]:

$$H_{ILP}[u, v] = \begin{cases} 1 & \text{if } D[u, v] \leq D_0 \\ 0 & \text{if } D[u, v] > D_0 \end{cases} \quad (12)$$

$$D[u, v] = \sqrt{\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2} \quad (13)$$

Die Gleichung 13 stellt die allgemeine Kreisgleichung dar, mit der der Abstand von einem Punkt zum Kreismittelpunkt berechnet wird. Der Mittelpunkt stellt dabei der DC-Wert des Frequenzspektrums dar. Über die Konstante D_0 kann die Grenzfrequenz respektive der Frequenzbereich um den DC-Wert, der nicht gedämpft werden soll, festgelegt werden. Die Charakteristik des idealen Tiefpassfilters im Frequenzbereich stellt die Abbildung 7 dar. Die Abbildung zeigt, dass der ideale Filter die Frequenzen oberhalb der Grenzfrequenz komplett abschneidet.

Hingegen bildet die Filterfunktion des Gausstiefpasses $H_G[u, v]$ eine Normalverteilung ab und wird mit der Gleichung 14 bestimmt [1].

$$H_G[u, v] = e^{\frac{-D^2(u,v)}{2D_0^2}} \quad (14)$$

Die Berechnung des Parameters D erfolgt mit der Gleichung 13 und über D_0 wird wieder die Grenzfrequenz des Filters festgelegt. Aufgrund der Normalverteilung Charakteristik

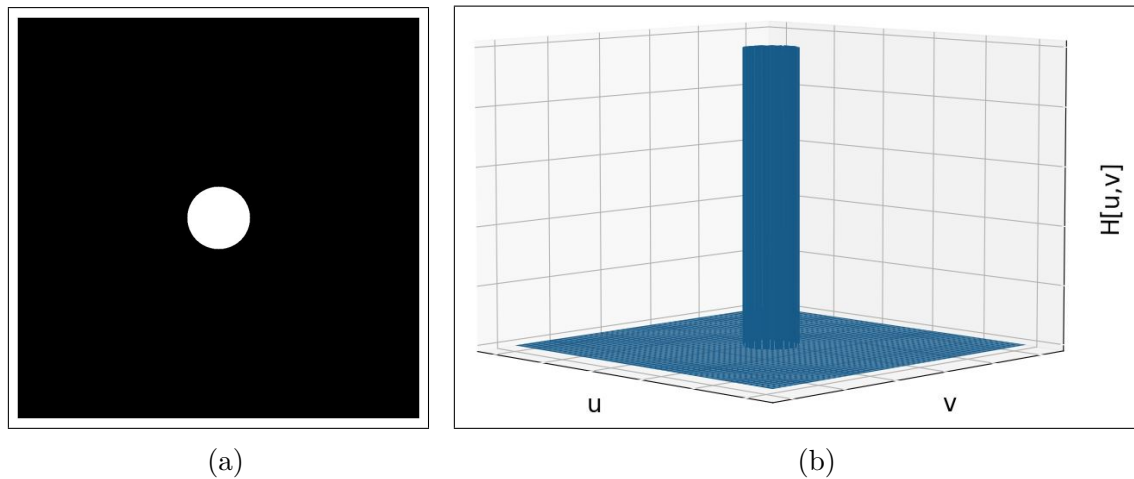


Abbildung 7: Idealer Tiefpassfilter in 2- (links) und 3-D Darstellung (rechts).

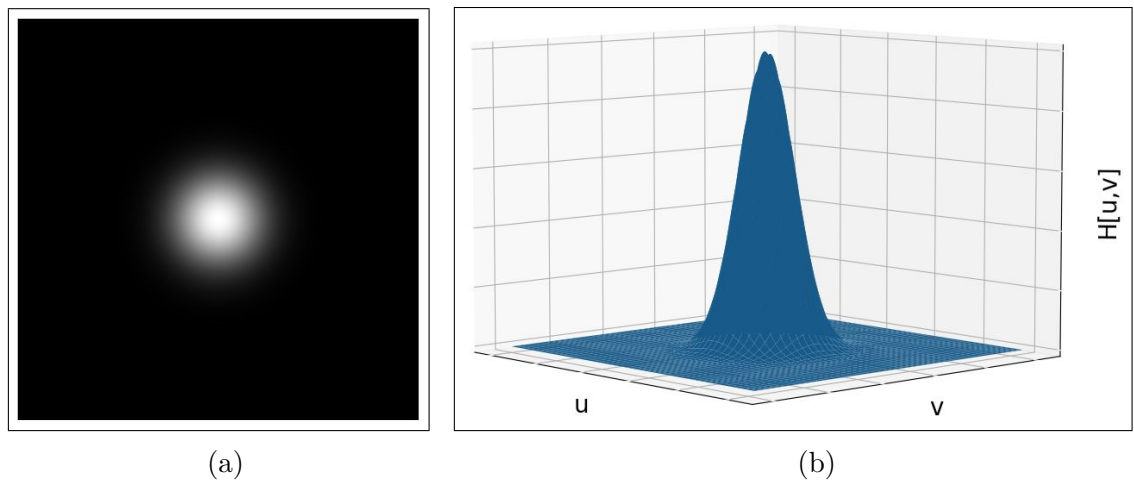


Abbildung 8: Gausstiefpass in 2- (links) und 3-D Darstellung (rechts).

besitzt der Gausstiefpass im Gegensatz zum idealen Tiefpass einen stetigen Amplitudengang. Infolgedessen werden die Frequenzen um die Grenzfrequenz nicht total abgedämpft, was die Abbildung 8 zeigt. Der Performancevergleich zwischen beiden Filtern erfolgt mit dem typischen Testbild „Lena“ und ist in Abbildung 9 zu sehen. Im Vergleich erzielt der ideale Filter einen stärkeren Trübungs-Effekt, jedoch sehen die gefilterten Bilder des Gausstiefpasses wesentlich glatter und ansprechender aus. Die Verunreinigung bei den ideal gefilterten Bildern kommt durch das sogenannte Ringing zustande, das beim genauen Betrachten der Bilder zu erkennen ist. Werden beide Filter in den Originalbereich mithilfe der iDFT transformiert, entsteht bei dem Kernel des idealen Filters ein vom Mittelpunkt sich ausbreitendes Interferenzmuster. Hingegen existiert das Interferenzmuster beim Kernel des Gausstiefpasses nicht. Beide Kernel sind in Abbildung 10 dargestellt. In dem Intensitätsprofil der horizontalen Mittellinie des idealen Filters lässt sich erkennen, dass neben der Hauptkeule viele kleinere unerwünschte Nebenkeulen existieren. Die Hauptkeule ist für die Filterung verantwortlich und die Nebenkeulen verursachen das störende Ringing.



Abbildung 9: Vergleich des idealen Tiefpass und Gausstiefpass

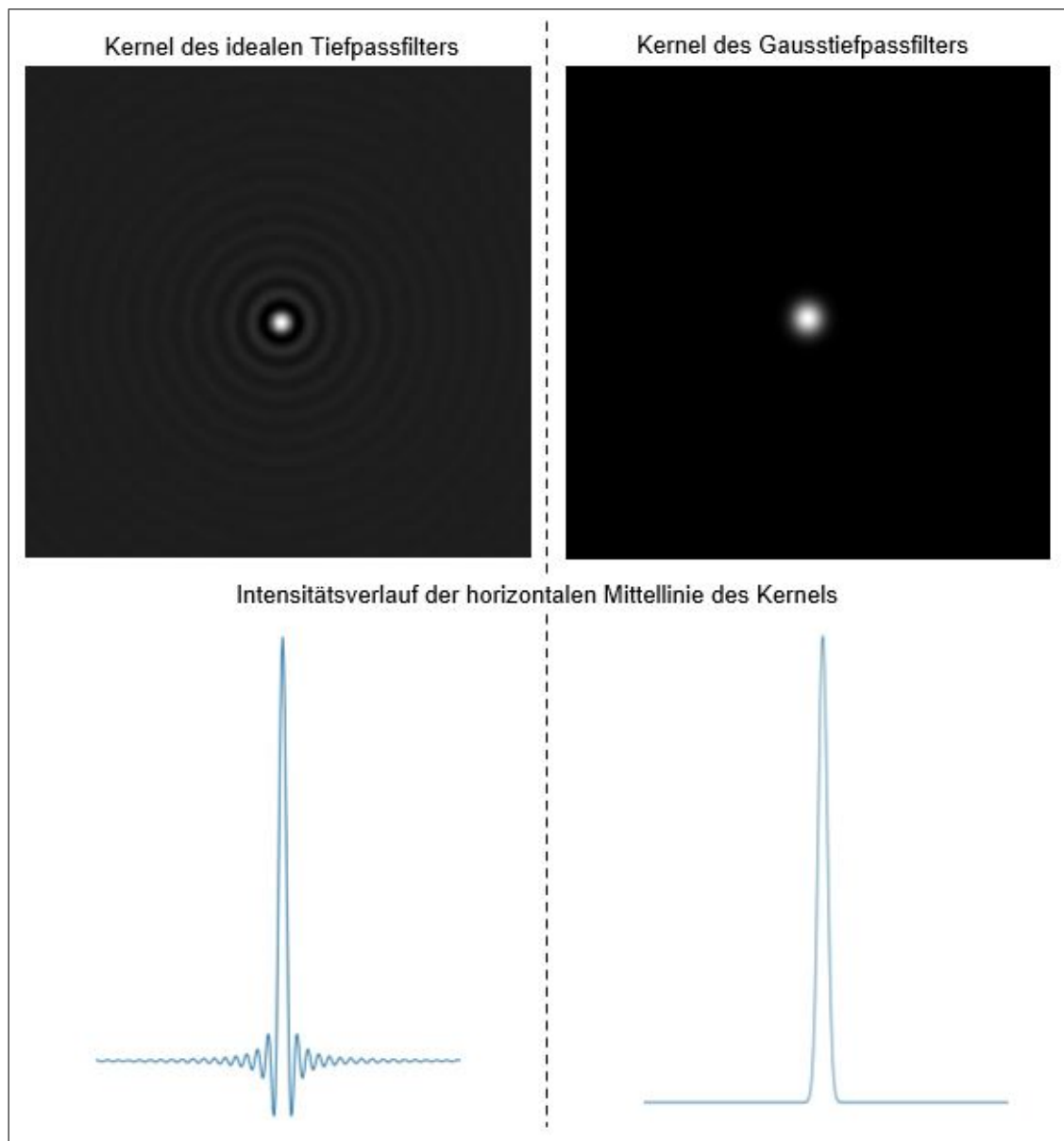


Abbildung 10: Kernel des idealen Tiefpasses und des Gausstiefpasses

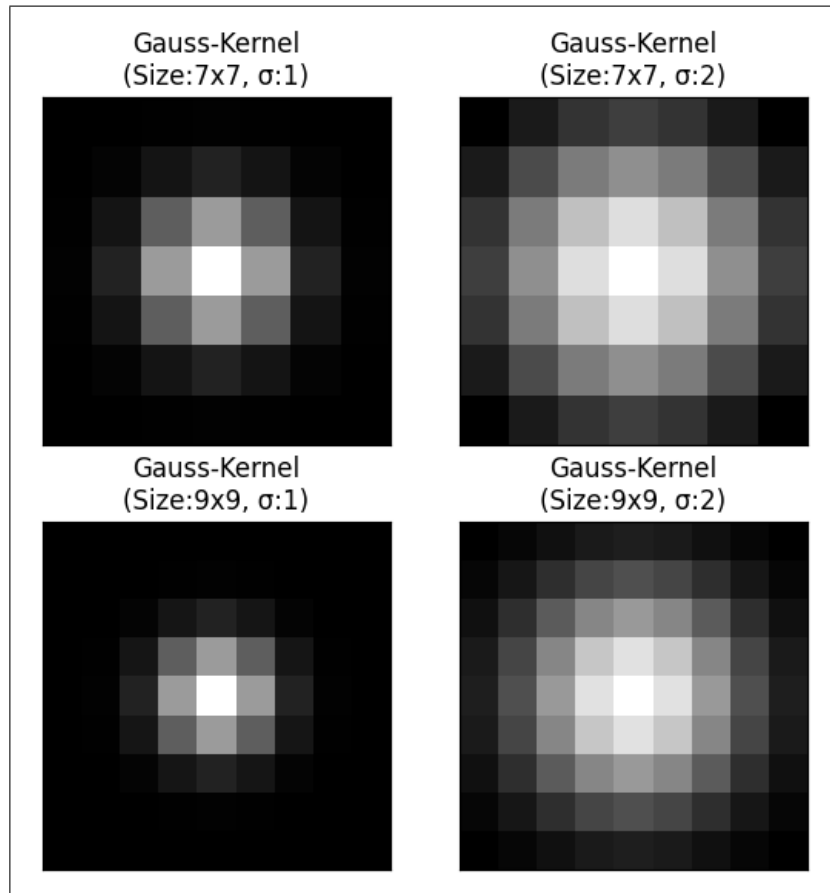


Abbildung 11: Vergleich der unterschiedlichen Gausskernel mit verschiedenen Größen und Standardabweichung

Originalbereich

Im Originalbereich lässt sich der Kernel eines Gaussfilters mit der Gleichung 15 erstellen. Dabei sind die Pixelwerten abhängig von deren Positionen x und y innerhalb des Kernels. Außerdem muss die Standardabweichung σ vorgegeben werden. Die Unterschiede zwischen Gauss-Kernel mit verschiedenen Kernelgrößen und Standardabweichungen sind in der Abbildung 11 dargestellt.

$$h_{Gauss}[x, y] = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (15)$$

Der Gauss-Filter hat ein Smoothing-Effekt, in dem er das Rauschen des Bildes unterdrücken. Allerdings werden die Kanten bei der Rauschenunterdrückung auch geglättet. Somit lässt sich die anschließende Bildsegmentierung schwieriger durchzuführen. Der Effekt von der Glättung ist in der Abbildung 12 zu erkennen. In der Abbildung ist zu erkennen, dass das Rauschen im Bild geglättet wird. Allerdings sind die Abgrenzungen zwischen Leiterbahnen und die Platine nicht mehr eindeutig wie zuvor zu erkennen. Um die Kantenkonturen zu behalten, wird häufig der Median-Filter eingesetzt. Obwohl der Median-Filter einen ähnlichen Effekt wie der Gauss-Filter erreichen kann, ist der Median-Filter keinem typischen Tiefpassfilter. Der Filtervorgang kann nicht mit Faltung beschrie-

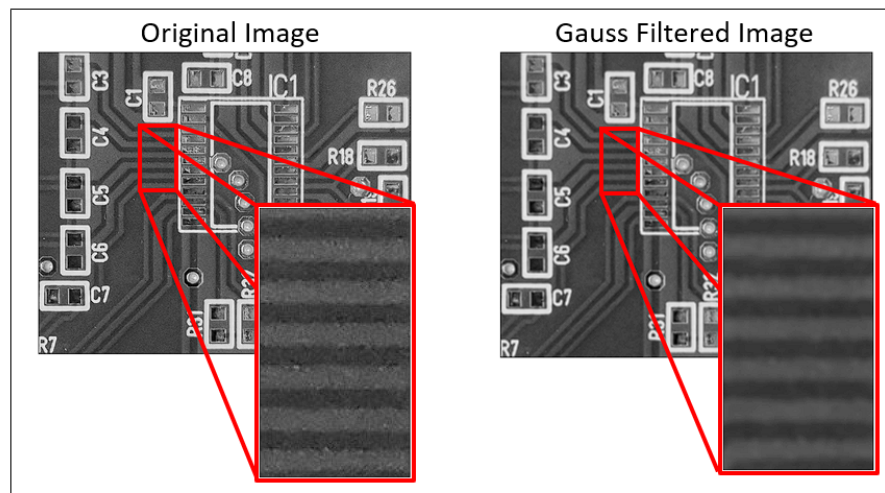


Abbildung 12: Vergleich der Bilder vor und nach der Filterung mit einem (7×7) Gauss-Filter ($\sigma = 3$)

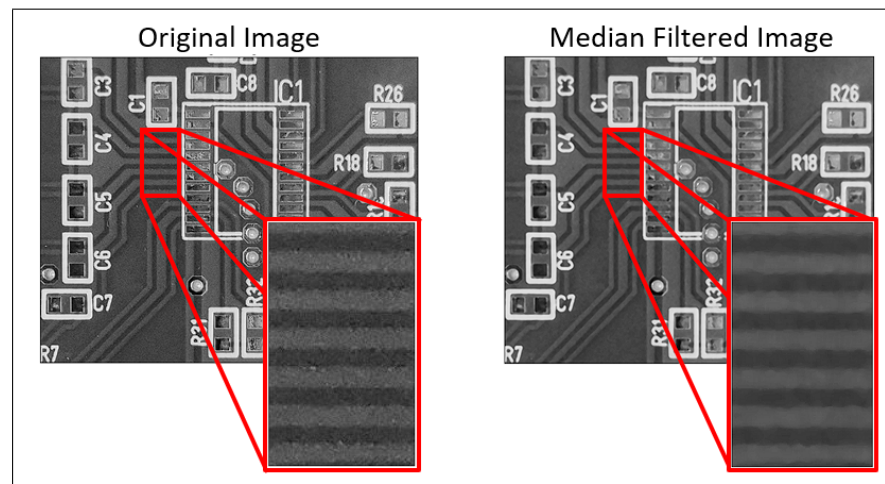


Abbildung 13: Vergleich der Bilder vor und nach der Filterung mit einem (7×7) Median-Filter

ben werden. [1] Beim Median-Filter werden die Grauwerte in einer definierten Umgebung des aktuellen Pixels aufgesammelt und der Größe nach sortiert. Der mittlere Grauwert dieser sortierten Liste wird ausgewählt und ersetzt den Grauwert des aktuellen Pixels. Somit lässt manche typische Formate des Rauschens leicht zu beseitigen, zum Beispiel: das Salt-and-Pepper-Rauschen. In der Abbildung 13 ist der Effekt vom Median-Filter dargestellt. Dabei ist zu erkennen, dass das Rauschen stark unterdrückt wird und die Leiterbahnen eindeutig zu erkennen sind.

3.2 Hochpass

Hochpassfilter kommen zum Einsatz, um scharfe Übergänge in einem Bild zu verstärken. Scharfe Übergänge, wie zum Beispiel Kanten, sorgen für hohe Frequenzen im Frequenzspektrum. Der Hochpassfilter lässt diese hohen Frequenzen passieren und dämpft niedrige ab. Deswegen werden Hochpassfilter zur Kantendetektion eingesetzt. Im folgendem werden im Frequenzbereich der ideale Hochpass sowie der Gausshochpass betrachtet und im Originalbereich zwei typische Filter für die Kantendetektion: der Sobel- und Canny-Filter.

Frequenzbereich

Der ideale Hochpass bildet die Umkehrung des idealen Tiefpasses. Das heißt, dass der Filter unterhalb der Grenzfrequenz alle Frequenzen total dämpft. Seine Filterfunktion $H_{IHP}[u, v]$ wird mit der Gleichung 13 und nachfolgender Gleichung aufgestellt:

$$H_{IHP}[u, v] = \begin{cases} 0 & \text{if } D[u, v] \leq D_0 \\ 1 & \text{if } D[u, v] > D_0 \end{cases} \quad (16)$$

Der Gausshochpass $H_{GHP}[u, v]$ ist die Umkehrung des Gausstiefpasses:

$$H_{GHP}[u, v] = 1 - e^{\frac{-D^2(u,v)}{2D_0^2}} \quad (17)$$

Der Parameter D_0 gibt wieder die Grenzfrequenz an. Die Abbildungen 14 und 15 zeigen die Charakteristiken des idealen Hochpasses und des Gausshochpasses. Der ideale Hochpass besitzt einen nicht stetigen Übergang und schneidet die Frequenzen unterhalb der Grenzfrequenz total ab. Hingegen weist der Gausshochpass aufgrund der invertierten Normalverteilung einen stetigen Übergang auf, wodurch die Frequenzen annähernd linear abgedämpft werden.

Beim Filtern mit einem Hochpass im Frequenzbereich entstehen negative Bildwerte. Für die Behandlung der negativen Bildwerte gibt es verschiedene Methoden. Die einfachste Methode ist es, alle negativen Bildwerte zu Null zu setzen. Jedoch geht dabei Bildkontrast verloren. Eine weitere Möglichkeit ist alle Bildwerte um den minimalsten Bildwert nach oben zu verschieben.

Die Anwendung der beiden Filter auf das Testbild „Lena“ zeigt die Abbildung 16. Alle gefilterten Bilder weisen eine gewisse Energielosigkeit auf. Das liegt daran, dass der Hochpassfilter den DC-Wert zu Null setzt. Der DC-Wert ist für die Durchschnittsintensität in einem Bild verantwortlich. Weiterhin ist zu erkennen, dass die Konturen beim Gausshochpassschärfer sind und auch bei einer höheren Grenzfrequenz noch besser nachvollzogen werden können. Die Ursache dafür ist wieder das Ringing, das sehr gut in dem gefilterten Bild bei der Grenzfrequenz $D_0 = 75$ erkennbar ist. Werden wieder beide Filter in den Originalbereich transformiert, zeigt sich wieder ein Interferenzmuster beim Kernel des idealen Hochpasses. Dies kann in Abbildung 17 nachvollzogen werden.

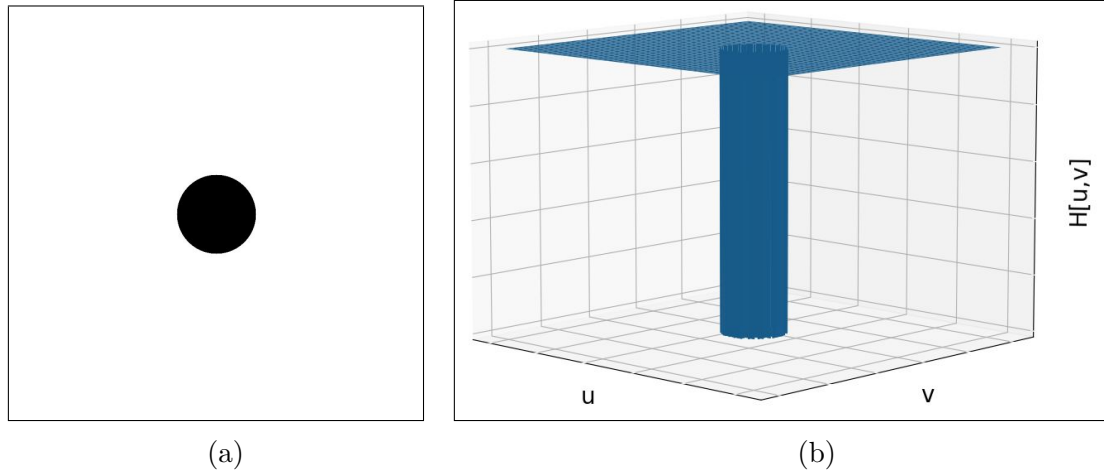


Abbildung 14: Idealer Hochpass in 2- (links) und 3-D Darstellung (rechts).

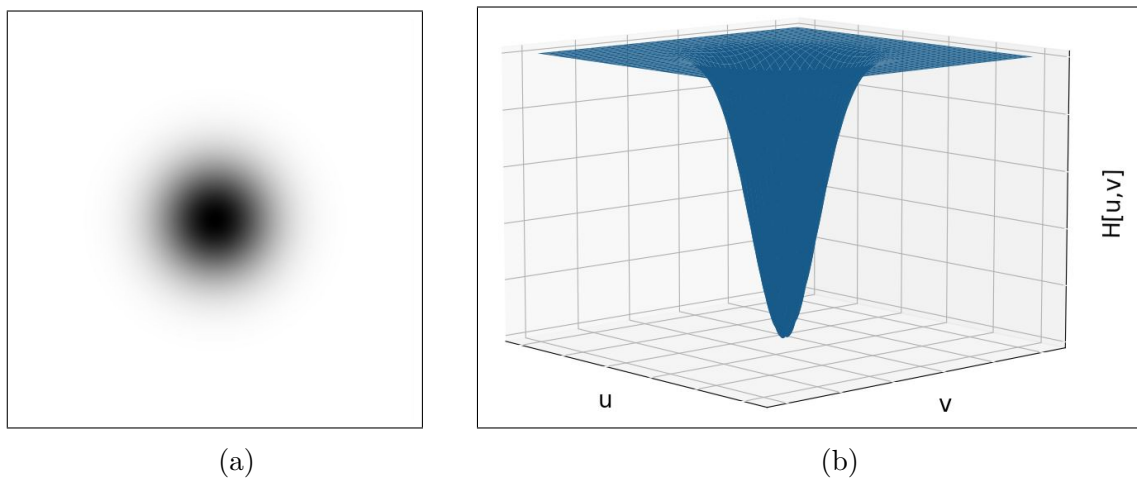


Abbildung 15: Gausshochpass in 2- (links) und 3-D Darstellung (rechts).

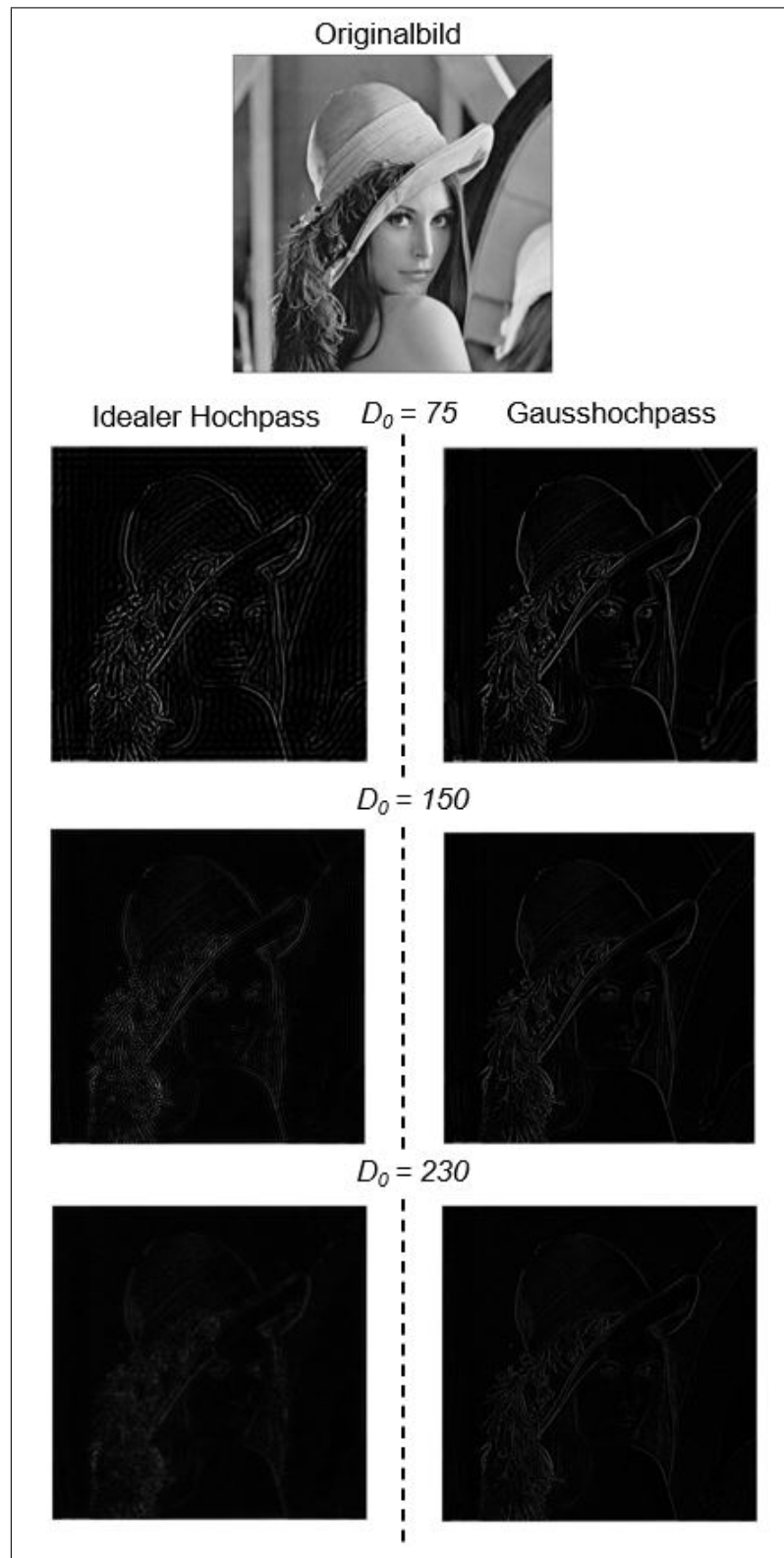


Abbildung 16: Vergleich des idealen Hochpasses und des Gausshochpasses

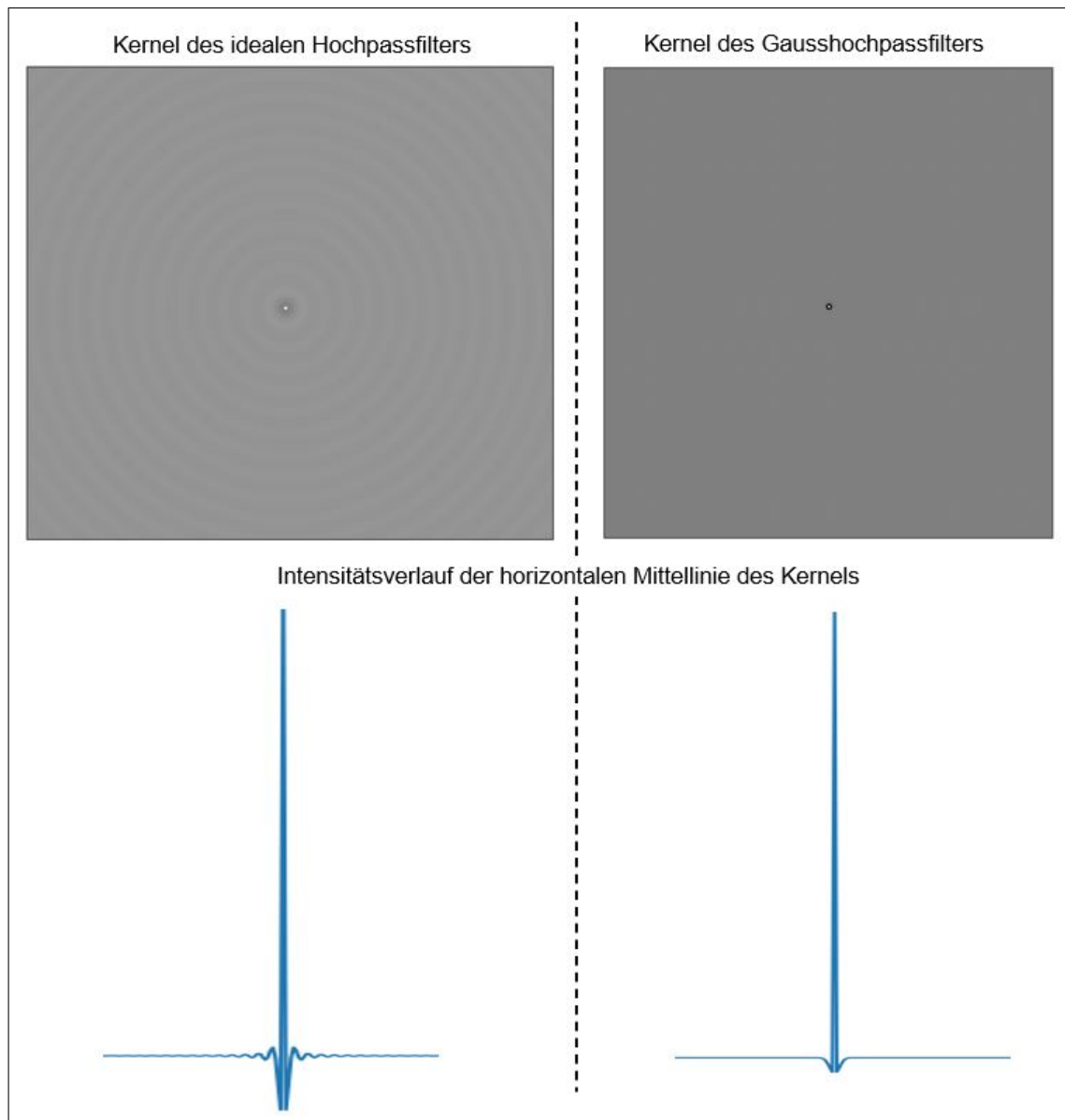


Abbildung 17: Kernel des idealen Hochpasses und des Gausshochpasses

Originalbereich

Ein typischer Hochpassfilter im Originalbereich ist der Sobel-Filter. Um die Kantemerkmale in der Zeilen- und Spaltenrichtung zu extrahieren, werden häufig zwei verschiedenen Sobel-Kernel verwendet. Der Aufbau der beiden Sobel-Kernel werden in der Abbildung 18 dargestellt.

<table><tr><td>-1</td><td>-2</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>2</td><td>1</td></tr></table>	-1	-2	-1	0	0	0	1	2	1	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-2	0	2	-1	0	1
-1	-2	-1																	
0	0	0																	
1	2	1																	
-1	0	1																	
-2	0	2																	
-1	0	1																	
Sobel-Filter (Vertikal)	Sobel-Filter (Horizontal)																		

Abbildung 18: Darstellung eines 3x3 Sobel-Filters

Durch die Anwendung der beiden Sobel-Kernels in der Spalten- und Zeilen Richtungen werden zwei Gradienten (G_x und G_y) gebildet. Der Kantengradient G_R des gesamten Bildes, entspricht die Grauwertmatrix des Bildes, wird mit der Gleichung 18 berechnet.

$$G = \sqrt{G_x^2 + G_y^2} \quad (18)$$

Um die Kanten genauer zu detektieren, wird häufig noch der Canny-Filter eingesetzt. Der Canny-Filter ist ein verbesserter Algorithmus basiert auf Sobel-Filter. Der Canny-Filter verwendet zuerst das Gauss-Filter, um das Rauschen des originalen Bildes zu unterdrücken. Danach wird der Sobel-Filter auf das geglättetes Bild angewendet. Mit dem horizontalen Gradienten G_x und dem vertikalen Gradient G_y lässt sich der Kantengradient des gesamten Bildes mit der Gleichung 18 berechnet. Außerdem ergibt sich die noch eine Richtungsmatrix der detektierten Kanten G_R , berechnet mit der Gleichung 19. Um das Rechnenskomplexität zu vereinfachen, werden die Winkel des einzelnen Pixels auf einer der vier Richtungen auf den X- und Y-Achsen gerundet. Somit entsteht eine Winkelmatrix der Kanten, in der die Winkel der Kanten auf eine der vier Richtungen zeigt: oben, unten, links oder rechts.

$$G_R = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (19)$$

Sobald die gerundete Winkelmatrix entsteht, wird das Non-Maxima-Suppression zwischen den Kantengradienten und die Richtungsmatrix angewendet. Dabei wird an jedem Pixeln überprüft, ob sein Grauwert im Kantengradienten zu seinem nächsten horizontalen oder vertikalen Nachbarn als lokalen Maximum darstellt. Die Überprüfungsrichtung ist abhängig vom Winkel des Pixels. Wenn der Winkel nach oben und unten zeigt, findet diese Überprüfung in der vertikalen Richtung statt. Sonst wird das lokale Maximum in der horizontalen Richtung gesucht. Wenn es dabei einen lokalen Maximum tatsächlich vorliegt, wird das Pixel als ein Teil der Kanten erkannt und seinem Grauwert im Kantengradienten wird beibehalten. Sonst wird das Pixel nicht als Kante betrachtet und den

entsprechenden Grauwert wird mit 0 ersetzt.

Danach wird die sogenannte Hysteresis Thresholding beim resultierten Kantengradienten angewendet. Dabei wird jeweils ein oberer und unterer Schwellenwert vorgegeben. Das Pixel wird als starke Kante erkannt, wenn seinem Grauwert im Kantengradienten größer als der obere Schwellenwert ist. Wenn der Grauwert des Pixels kleiner als der untere Schwellenwert ist, wird das Pixel nicht mehr als Kanten betrachtet. Sonst wird das Pixel als schwache Kanten erkannt und eine Kontinuitätsprüfung wird an diesen Pixeln durchgeführt. Dabei wird kontrolliert, ob das Pixel mit einer starken Kante verbindet. Wenn das der Fall ist, wird das Pixel als ein Teil der Kanten betrachtet. Sonst verfällt das Pixel und sein Grauwert wird mit 0 ersetzt.

Der Vorteil des Canny-Filters gegenüber den Sobel-Filter ist, dass die detektierten Kanten nur eine Breite von einem Pixel hat. Somit lassen sich die Kanten im Bild eindeutiger darstellen und dadurch ist eine weitere Verarbeitung der Kanten einfacher durchzuführen. Allerdings ist die Berechnung des Canny-Filters deutlich aufwendiger als die Anwendung eines Sobel-Filters. Dies führt zu einer längeren Berechnungszeit.

Der Unterschied zwischen den Sobel- und Canny-Filtern wird in der Abbildung 19 dargestellt. Im originalen Bild ist ein Teil einer Platine zu sehen. Dabei sind die Markierungen und die Texte weiß gekennzeichnet. Durch das Thresholding werden die Texte und Konturen aus dem Hintergrund extrahiert. Anschließend werden die Sobel- und Canny-Filtern jeweils an das behandelte Bild angewendet.

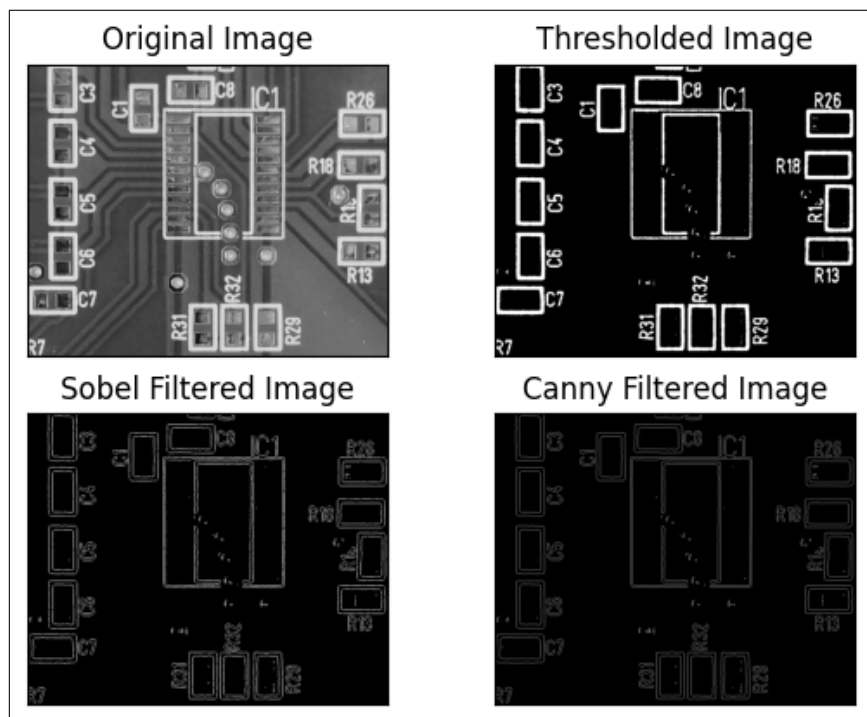


Abbildung 19: Vergleichsdarstellung zwischen dem Sobel- und Canny-Filter

4 Bild-Segmentierung

Das Ziel der Bild-Segmentierung ist, die erwünschte Merkmale eines Bildes zu extrahieren. Die zu erkennenden Objekte haben häufig einen unterschiedlichen Grauwert als der Hintergrund. Somit lässt sich das Thresholding-Verfahren häufig anzuwenden. Thresholding ist im Prinzip eine Operation mit den Schwellenwerten. Wenn der Grauwert eines Pixels den Schwellenwert erreicht hat, wird eine bestimmte Aktion an diesem Pixel durchgeführt. In diesem Bericht werden die binäre Thresholding, Inside Thresholding und Outside Thresholding realisiert.

Beim binären Thresholding werden die Pixel mit Einsen ersetzt, wenn der originale Grauwert des Pixels über den maximalen Schwellenwert liegt. Anderen Pixeln werden mit Nullen gefüllt. Beim Inside Thresholding werden die Pixeln beibehalten, wenn die Grauwerte des Pixels zwischen den beiden Schwellenwerten liegen. Alle anderen Pixeln werden mit Nullen ersetzt. Im Vergleich dazu hat das Outside Thresholding eine invertierte Logik. Die Pixel, die einem Grauwert außerhalb der Schwellenwerte besitzen, werden beibehalten. Sonst werden die Pixel mit Nullen ersetzt.

Die entsprechende Logik der Thresholdingverfahren sind in den Gleichungen 20, 21 und 22 dargestellt. Dabei ist K_{min} und K_{max} die untere und obere Schwellenwert. Die Eingangspixel und Ausgangspixel der Bildern werden mit $f[x, y]$ und $g[x, y]$ dargestellt.

$$g[x, y] = \begin{cases} 0 & f[x, y] > K_{max} \\ 1 & f[x, y] \leq K_{max} \end{cases} \quad (20)$$

$$g[x, y] = \begin{cases} f[x, y] & K_{min} \leq f[x, y] \leq K_{max} \\ 0 & sonst \end{cases} \quad (21)$$

$$g[x, y] = \begin{cases} f[x, y] & (f[x, y] \geq K_{max}) \text{ or } (f[x, y] \leq K_{min}) \\ 0 & sonst \end{cases} \quad (22)$$

In der Abbildung 20 wird der Effekt der Inside- und Outside Thresholding dargestellt. Die eingestellte Schwellenwerte liegen bei $K_{min} = 50$ und $K_{max} = 70$. Somit lässt sich die Merkmale der Leiterbahnen aus dem Bild extrahieren. In der Abbildung ist zu erkennen, dass die Leiterbahnen sich zum Großteil extrahieren lassen. Eine Kontrastverbesserung kann die Extraktion weiter erleichtern.

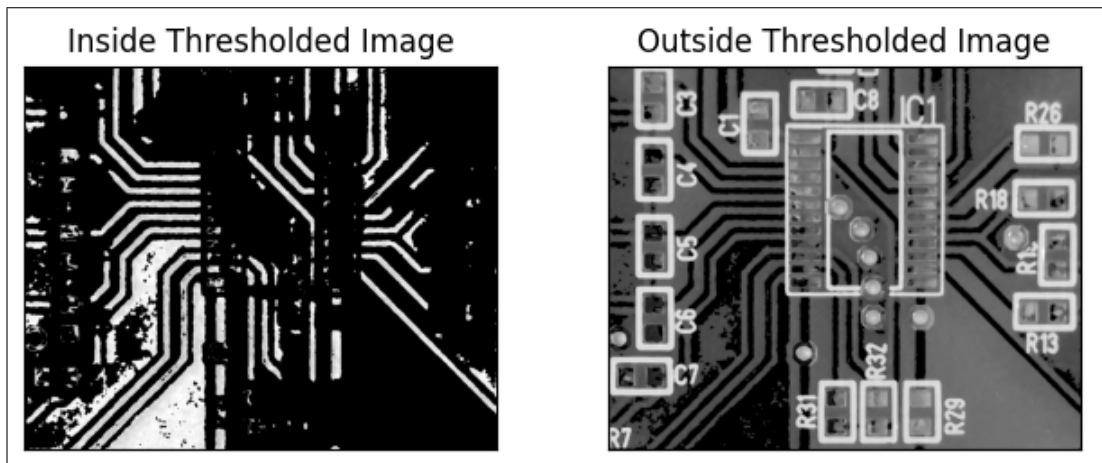


Abbildung 20: Segmentierung eines Bildes mit Thresholding Verfahren

5 Ausblick

In diesem Bericht wurden zunächst die mathematische Grundlagen für die Filterung im Original- und Frequenzbereich geschaffen. Nach den Grundlagen wurden einige Filterarten von Hoch- und Tiefpassfiltern mit Python implementiert und die Auswirkung auf Testbilder diskutiert. Dazu gehören unter anderem der ideale Hoch- und Tiefpass, der Gaussfilter sowie der Sobel- und Canny Filter. Weiterhin wurde die Thresholding-Methode für die Bildsegmentierung und Objektextraktion realisiert. Mögliche Erweiterungen für die Bildsegmentierung stellen das Opening und Closing dar, mit denen Kanten noch eindeutiger extrahiert werden können. Im nächsten Laborbericht wird der Hardware-Entwurf für die Umsetzung der Kantendetektion auf einem Xilinx Zybo Board durchgeführt und anschließend implementiert.

Literatur

- [1] E. Richard Gonzalez, C. Rafael. Woods. *Digital Image Processing*. ISBN: 1-292-22304-9. Pearson, 4th edition, April 2018.