

Bachelorthesis  
Alexander Motzek  
Erkennung unscharfer Bilder mittels  
Haar-Wavelet-Transformation

*Fakultät Technik und Informatik  
Department Informations- und  
Elektrotechnik*

*Faculty of Engineering and Computer Science  
Department of Information and  
Electrical Engineering*

Alexander Motzek  
Erkennung unscharfer Bilder mittels  
Haar-Wavelet-Transformation

Bachelorthesiseingereicht im Rahmen der Bachelorprüfung  
im Studiengang Informations- und Elektrotechnik  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuer Prüfer: Prof. Dr. rer. nat. Annabella Rauscher-Scheibe  
Zweitgutachter: Prof. Dr. habil. Henning Dierks

Abgegeben am 19. Dezember 2011

## **Alexander Motzek**

### **Thema der Bachelorthesis**

Erkennung unscharfer Bilder mittels Haar-Wavelet-Transformation

### **Stichworte**

Haar-Wavelet-Transformation, Multiskalenanalyse, Bewegungsunschärfe, Fehlfokussierung, Gauß'sche Unschärfe, Scharfzeichnung

### **Kurzzusammenfassung**

Diese Thesis befasst sich mit dem Problem, unscharfe Bilder automatisch erkennen zu können. Hierfür wird ein Algorithmus entwickelt, welcher sowohl Bewegungsunschärfe als auch eine Fehlfokussierung in einem Bild erkennen kann und in der Lage ist, den Schärfebereich eines Bildes hervorzuheben bzw. zu visualisieren. Der hier entwickelte Algorithmus basiert auf einer Multiskalenanalyse mittels der Haar-Wavelet-Transformation, welche im Detail vorgestellt wird. Außerdem deckt diese Bachelorthesis die gesamten Grundlagen der benötigten Bildverarbeitung ab; Gauß'sche Unschärfe, Bewegungsunschärfe, Scharfzeichnung und Unschärfmaskierung. Als Testobjekte dienen unbearbeitete Bilder einer digitalen Spiegelreflexkamera mit unterschiedlichen Objektiven aus verschiedenen Aufnahmeszenarien.

## **Alexander Motzek**

### **Title of the paper**

Image Blur Detection using Haar-Wavelet-Transformation

### **Keywords**

Haar-Wavelet-Transformation, multiresolution analysis, motion-blur, out-of-focus-blur, Gaussian blur, sharpening

### **Abstract**

Target of this paper is to detect blurred images. In order to detect motion- and out-of-focus-blur in images we design an algorithm which is also able to visualize sharp areas of an image. The designed algorithm is based on a multiresolution analysis using the Haar-wavelet transformation. The thesis covers the required theoretical background about the used transformation and imageprocessing; Gaussian blur, motion blur, sharpening and unsharpmasking. We use raw, unedited images from a digital single-lens reflex camera with various lenses from different locations.

## Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>3</b>
<b>2</b>	<b>Schärfe von Bildern</b>	<b>4</b>
2.1	Out-of-Focus-Blur . . . . .	4
2.2	Motion-Blur . . . . .	5
<b>3</b>	<b>Haar Wavelet</b>	<b>6</b>
3.1	Mathematische Betrachtung der Haar-Wavelet Transformation . . . . .	7
3.2	Praktische Anwendung der Haar-Wavelet-Transformation . . . . .	9
3.3	Darstellungsarten der Haar-Wavelet-Transformation . . . . .	10
3.4	Eigenschaften der Haar-Wavelet-Transformation . . . . .	11
<b>4</b>	<b>Kanten</b>	<b>12</b>
<b>5</b>	<b>Weichzeichnung</b>	<b>13</b>
5.1	Gauß'sche Unschärfe . . . . .	13
5.2	Weichgezeichneter Dirac-Impuls . . . . .	15
5.2.1	HWT des idealen Dirac-Impulses . . . . .	16
5.2.2	HWT des weichgezeichneten Dirac-Impulses . . . . .	17
5.2.3	Evaluation Dirac-Impuls . . . . .	18
5.3	Weichgezeichneter Sprung . . . . .	20
5.3.1	HWT der idealen Sprungkante . . . . .	22
5.3.2	HWT der weichgezeichneten Sprungkante . . . . .	23
5.3.3	Evaluation Sprungkante . . . . .	24
5.4	Über die Vergleichbarkeit verschiedener Dekompositionsebenen . . . . .	25
<b>6</b>	<b>Scharfzeichnung</b>	<b>26</b>
6.1	Schärfungsfilter . . . . .	27
6.1.1	Schärfung durch Ableitung . . . . .	27
6.1.2	Unschärfmaskierung . . . . .	28
6.1.3	Laplace-Scharfzeichnung und Unschärfmaskierung . . . . .	29
6.1.4	Laplacian of Gaussian . . . . .	30
6.2	Untersuchung des Verhalten von Scharfzeichnung auf die Haar-Wavelet-Transformation . . . . .	31
<b>7</b>	<b>Bewegungsunschärfe</b>	<b>33</b>
<b>8</b>	<b>Algorithmus zur Kantenklassifizierung in Signalen</b>	<b>34</b>
<b>9</b>	<b>Zweidimensionale Haar-Wavelet-Transformation</b>	<b>36</b>
<b>10</b>	<b>Algorithmus zur Erkennung scharfer Kanten eines Bildes</b>	<b>37</b>

<i>INHALTSVERZEICHNIS</i>	2
<b>11 Anwendung des Algorithmus</b>	<b>38</b>
<b>12 Algorithmus zur Klassifizierung von Bildern</b>	<b>41</b>
<b>13 Evaluation des Algorithmus</b>	<b>43</b>
13.1 Simulation . . . . .	43
13.2 Benchmark . . . . .	44
13.3 Sportaufnahmen Satz 1 . . . . .	45
13.4 Sportaufnahmen Satz 2 . . . . .	45
13.5 Landschaftsaufnahmen . . . . .	45
13.6 Automobil-Ausstellung . . . . .	46
13.7 Gesamt Evaluation . . . . .	46
<b>14 Fazit &amp; Ausblick</b>	<b>47</b>
<b>Abbildungsverzeichnis</b>	<b>48</b>
<b>Tabellenverzeichnis</b>	<b>49</b>
<b>Literatur</b>	<b>50</b>

## 1 Vorwort

Mit der immer weiteren Verbreitung von Digitalkameras und dem immer günstiger werdendem Speicherplatz wächst zugleich die Anzahl an aufgenommenen Bildern. Mit Speicherplatz von mehreren tausend Fotos auf einer Speicherkarte und einer damit einhergehenden Tendenz Bilder nicht mehr sofort zu prüfen, sondern alle Bilder zu behalten und sie erst auf dem heimischen Computer zu prüfen, steigt auch die Anzahl an Ausschuss-Bildern.

Um diese Bilderflut eindämmen zu können, verfolgt diese Bachelorthesis das Ziel, den produzierten Ausschuss zu sortieren und zu begrenzen.

Jeder Fotograf dürfte die Situation kennen, dass er nach einem Tag harter Arbeit oder privatem Vergnügen Hunderte bis Tausende Fotos nach Hause bringt. Wir müssen dann zunächst die Spreu vom Weizen trennen. Das heißt wir prüfen jedes einzelne Bild, ob es nicht verwackelt oder unscharf ist. Zusätzlich müssen wir gleiche Motive untereinander prüfen, welches von ihnen das Beste ist. Da der Teufel bekanntlich im Detail steckt, müssen die Bilder zudem in voller Auflösung begutachtet werden (die Skalierung eines Bildes kaschiert viele Bildfehler, welche im späteren Druck auffallen). Neben einer großen Motivation bedarf dies vor allem viel Zeit. Außerdem liegt hier ein erheblicher Frust-Faktor vor, wenn nun gerade *das* Bild unscharf ist, obwohl es doch, auf dem Display der Kamera, perfekt aussah.

Um diese Aufgabe zu automatisieren, kam mir die Idee nach einem geeigneten Programm hierfür zu suchen, welches einem wenigstens die unscharfen oder verwackelten Bilder heraussortiert. Jedoch blieb - zu meinem Erstaunen - die Suche erfolglos. Nur eine Veröffentlichung von Tong, Li, H. & C. Zhang (im weiteren als Tong et al. abgekürzt) beim Microsoft Research Asia [TLZZ] fand sich hierzu. Diese wurde zwar schon 2004 veröffentlicht, es gab jedoch nur spärliche weitere Informationen zu diesem Thema. Es fanden sich jedoch viele Foreneinträge, in denen genau nach solch einem Programm gefragt wurde. Tong fand heraus, dass sich die Haar-Wavelet-Transformation dazu eignet, Unschärfe und deren Stärke in Bildern zu erkennen.

Auf der Suche nach einem Thema für diese Bachelorarbeit kam mir diese Veröffentlichung wieder in den Sinn. Auch heute finden sich nur wenige weitere Veröffentlichungen zu diesem Thema.

Da jedoch Tong et al. in ihrem Paper die Thematik nur empirisch betrachteten und als Ausgangsdaten bereits fertig bearbeitete Bilder, welche anschließend mittels Filtern verwackelt oder unscharf gemacht wurden, verwendeten, soll uns diese Veröffentlichung nur als Inspiration dienen.

Wir werden in dieser Bachelorthesis die von Tong et al. empirisch festgestellten Beobachtungen theoretisch herleiten, betrachten und weiterhin einen eigenen Algorithmus entwickeln und evaluieren. Ab und zu werden wir hierbei Querverweise zu Tong herstellen und seine Arbeit begutachten.

Wir werden uns auch dem Problem annehmen, dass die Schärfe von Bildern nicht immer leicht zu beurteilen ist und der erste Eindruck stark täuschen kann. Wir werden hier verschiedene Algorithmen entwickeln, welche es uns zum Einen ermöglichen werden Schärfe zu visualisieren, zum Anderen aber auch direkt sortieren können.

Welche Bildfehler alle von diesem Algorithmus zu betrachten sind, werden wir im folgenden Kapitel näher untersuchen. Nachdem wir uns anschließend mit den Grundlagen der Haar-Wavelet-Transformation, der digitalen Bildverarbeitung der Simulation von Bildfehlern vertraut gemacht haben, werden wir anschließend aus theoretischen Überlegungen einen Algorithmus entwerfen. Diesen werden wir für zweidimensionale Signale (Bilder) erweitern, verfeinern und zum Schluss evaluieren und seine Tauglichkeit feststellen.

Wir werden hierfür unbearbeitetes Rohmaterial direkt aus einer digitalen Spiegelreflexkamera aus unterschiedlichen Szenarien verwenden.

## 2 Schärfe von Bildern

Wir beginnen damit, uns Gedanken über die Schärfe von Bildern zu machen. Dies ist jedoch ein komplizierter Themenbereich, da hier auch viel Psychologie mit hinein spielt.

Als Beispiel hierfür sei eine klassische Taktik in der Fotografie und Bildverarbeitung beschrieben, um Portraitaufnahmen zu verbessern. Das Gehirn bezieht die Information über die Schärfe eines Bildes, auf dem ein Gesicht abgebildet ist, primär aus den Augen, Haaren und Lippen. Es ist üblich - besonders bei Aufnahmen für Modemagazine - die Haut eines Modells komplett weichzeichnen, sodass diese perfekt rein erscheint. Nur die Partien um den Mund, die Augen, Augenbrauen und Haare werden scharfgezeichnet. Dieses nun eigentlich unscharfe Bild - es wurde ja weichgezeichnet - empfindet das Gehirn als scharf.

Wir möchten im Folgenden jedoch nicht auf die Psychologie des Schärfeeindrucks von Bildern eingehen, sondern uns mit den Ursachen von Unschärfen in Bildern vertraut machen.

Einstellbare Parameter eines Bildes, die die Schärfe eines Bildes beeinflussen, sind die Brennweite  $f$ , die Blendenöffnung  $F$ , die Belichtungszeit  $t$  und die Fokusdistanz  $d$ . Außerdem wird die Schärfe von weiteren Faktoren wie der Güte des Objektivs oder der Qualität des Sensors bestimmt.

### 2.1 Out-of-Focus-Blur

Die am häufigsten anzutreffende Unschärfe in einem Bild ist der „Out-of-Focus-Blur“. Dieser Begriff bezeichnet verschiedene Unschärfen zugleich. Im Deutschen können wir diesen Begriff unterteilen in Fehl-Fokussierung und Tiefenunschärfe. Was dies bedeutet und wann es auftritt, wollen wir nun betrachten.

Die Fehl-Fokussierung (Abbildung 1a) gehört zu den häufigsten Fehlern in der Fotografie, sie tritt immer dann auf, wenn die Fokusdistanz  $d$  nicht korrekt gesetzt wurde. Dies kann zu komplett unscharfen Bildern führen oder zu Bildern, in denen das eigentliche Motiv unscharf ist, und etwas anderes - eigentlich ungewünschtes - scharf. Dies ist z. B. dann der Fall, wenn die Kamera auf den Hintergrund scharfgestellt hat, obwohl die Person im Vordergrund eigentlich fotografiert werden sollte. Die Fehl-Fokussierungs-Unschärfe ist in jedem Falle unerwünscht.

Die andere Art des Out-of-Focus-Blur ist die Tiefenunschärfe (Abbildung 1b). Diese hat die gleichen Eigenschaften wie eine Fehlfokussierung, sieht also gleich aus, ist jedoch erwünscht. Die Tiefenunschärfe, auch Bokeh genannt, dient dem Freistellen von Bildelementen. Tiefenunschärfe wird aus dem Zusammenspiel von Brennweite und Blendenöffnung erzeugt. Allgemein kann man sagen: Je höher die Brennweite  $f$  und je größer die Blendenöffnung  $F$ , desto geringer die Tiefenschärfe, bzw. desto höher die Tiefenunschärfe.



(a) Fehl-Fokussierung



(b) Tiefenunschärfe

Abbildung 1: Beispiele zu Out-Of-Focus-Blur.

## 2.2 Motion-Blur

Ein weiterer Grund für unscharfe Bilder ist die Bewegungsunschärfe („Motion-Blur“). Diese tritt immer dann auf, wenn die Kamera während der Belichtungszeit bewegt wurde, oder sich das Objekt während der Aufnahme bewegte. Gründe für Bewegungsunschärfe können eine zu hohe Belichtungszeit in Verbindung mit dem Fehlen einer Stabilisierung (z. B. Stativ) der Kamera sein, aber auch einen gestalterischen Hintergrund haben.

Häufig tritt Bewegungsunschärfe jedoch ungewollt auf (Abbildung 2a). Wird bei unzureichendem Licht zu lange belichtet, wirkt sich das natürliche Zittern der Hand als Bewegungsunschärfe im Bild aus. Dieser Effekt wird mit zunehmender Brennweite verstärkt, da hier kleine Bewegungen eine große Bewegung des Bildausschnitts verursachen. Gegenmaßnahmen hierfür können ein Stativ oder ein Bildstabilisator im Objektiv sein.

Im Bereich der professionellen Fotografie wird Bewegungsunschärfe jedoch auch gezielt erzeugt (Abbildung 2b). So kann ein sich bewegendes Objekt während der Belichtung mit der Kamera verfolgt werden, wodurch der Hintergrund verschwimmt (Bewegungsunschärfe), wohingegen die Bewegung des Objektes jedoch ausgeglichen wird und so scharf abgebildet wird. Das Ziel ist es, die Bewegung des Objektes festzuhalten, um die Dynamik des Bildes zu steigern.



(a) Verwackelung



(b) Stil-Mittel

Abbildung 2: Beispiele zu Motion-Blur.

Unser Ziel ist es, die *ungewollten* Unschärfen erkennen und aussortieren zu können.

Wie Tong et al. festgestellt hat, eignet sich die Haar-Wavelet-Transformation dazu. Im Folgenden werden wir uns daher mit ihren Grundlagen und Eigenschaften beschäftigen.

### 3 Haar Wavelet

Um eine Vorstellung davon zu erhalten, wie die Haar-Wavelet Transformation funktioniert und welche Eigenschaften sie besitzt, wollen wir folgendes einfaches Signal betrachten. Dieses Signal können wir auch als ein einzeliges Bild mit vier Pixeln ansehen.

$$[ 1 \quad 3 \quad 6 \quad 10 ] \quad (1)$$

Wir können dieses in der *Haar-Basis* darstellen, indem wir die Mittelwerte von jeweils zwei Zahlenpaaren darstellen als

$$\left[ \frac{1+3}{2} \quad \frac{6+10}{2} \right] = [ 2 \quad 8 ] \quad (2)$$

Man sieht leicht, dass auch andere Signale diese Haar-Basis besitzen können. Ein Beispiel hierfür wäre z. B.  $[0 \ 4 \ 0 \ 16]$ . Da es sich bei der Haar-Wavelet Transformation jedoch um eine verlustfreie Transformation handelt, benötigen wir noch weitere Informationen. Um das Signal verlustfrei wiederherstellen zu können, definieren wir einen Detail-Koeffizienten, welcher nicht den Mittelwert der beiden Zahlenpaare darstellt, sondern deren gemittelte Differenz. Also den Teil, welcher bei der Darstellung in der Haar Basis verloren gegangen ist.

Die 2 entstand aus der 1 und 3, welche jeweils 1 von ihrem Mittelwert entfernt sind. Die 8 entstand aus  $[6 \ 10]$ , welche jeweils 2 von ihrem Mittelwert entfernt sind. Unser Detail-Koeffizient ist also

$$\left[ \frac{1-3}{2} \quad \frac{6-10}{2} \right] = [ -1 \quad -2 ] \quad (3)$$

Dieses Verfahren des Mitteln und Differenzieren wird als *averaging and differencing* bezeichnet. Die Kombination aus den gemittelten Koeffizienten und den Detail-Koeffizienten ist als die Wavelet-Transformation definiert. Diese wird auch als Wavelet-Dekomposition bezeichnet. Manchmal wird auch erst die vollständige Dekomposition als Haar-Wavelet-Transformation bezeichnet. Wir bezeichnen jedoch jeden einzelnen averaging and differencing-Schritt als Haar-Wavelet-Transformation.

$$[ 2 \quad 8 \quad -1 \quad -2 ] \quad (4)$$

Wir können dieses Verfahren nun auch rekursiv auf den aus der ersten Dekomposition erhaltenen gemittelten Vektor anwenden. Dies führt uns dann zur zweiten Dekomposition:

$$\left[ \frac{2+8}{2} \quad \frac{2-8}{2} \right] = [ 5 \quad 3 ] \quad (5)$$

Da wir auf den hier erhaltenen gemittelten Koeffizienten 5 keine weitere Dekomposition anwenden können, bezeichnen wir dies als vollständige Dekomposition. Ersetzen wir den gemittelten Koeffizienten der ersten Dekomposition durch seine Haar-Wavelet Transformation, erhalten wir so die vollständige Haar-Wavelet Dekomposition des Ursprungssignals zu

$$[ 5 \quad 3 \quad -1 \quad -2 ] \quad (6)$$

Dieser Prozess der rekursiven Anwendung des averaging and differencing wird auch als eine Filter-Bank bezeichnet.

Wir möchten uns nun ansehen was hier mathematisch passierte.

### 3.1 Mathematische Betrachtung der Haar-Wavelet Transformation

Die Haar-Wavelet-Transformation, wird im folgenden als HWT abgekürzt.

Um die HWT mathematisch zu betrachten, begeben wir uns in den Bereich der linearen Algebra. Wir werden der mathematischen Betrachtung in [SDS95a] folgen, diese stellt die HWT sehr intuitiv dar, kratzt jedoch nur an der Oberfläche. Wir werden versuchen, die Mathematik hier anschaulich zu halten, ohne dabei zu vage zu werden. Weitergehende und deutlich vertiefendere Mathematik zu Wavelets, Wavelet-Transformationen und Multiskalenanalysen findet sich z. B. in [SDS95b] oder [GWo8] und würde den Rahmen dieses Kapitels sprengen. Betrachten wir zunächst Signale (oder Bilder) als „stückweise konstante Funktionen auf dem halboffenen Intervall  $[0, 1)$ “ [SDS95a]<sup>1</sup>. Das heißt auf diesem Intervall sind unsere Abtastwerte oder Pixel verteilt. Diese sind jedoch diskrete Werte und sind klar voneinander abgegrenzt, wodurch sie das Intervall jeweils in konstante Abschnitte unterteilen. Diese einzelnen Abschnitte haben dann einen konstanten Wert.

Ein Zwei-Pixel Bild hätte so z. B. konstante Werte in den Intervallen  $[0, \frac{1}{2})$  und  $[\frac{1}{2}, 1)$ . Ein Vier-Pixel Bild hätte dann die konstanten Intervalle  $[0, \frac{1}{4})$ ,  $[\frac{1}{4}, \frac{2}{4})$ ,  $[\frac{2}{4}, \frac{3}{4})$  und  $[\frac{3}{4}, 1)$ . Um die Mathematik nicht unnötig zu verkomplizieren, betrachten wir im folgenden auch nur Signale mit  $2^n$  Abtastwerten.

Wir definieren nun einen Vektorraum  $V^j$ , welcher uns alle  $2^j$ -langen Signale liefert. Alle Zwei-Pixel Bilder können damit als Linearkombination aus dem Vektorraum  $V^1$  dargestellt werden. Den Längenfaktor  $j$  bezeichnen wir im folgenden als Dimension („Skala“ oder „Auflösung“ wäre zwar passender, aber weniger intuitiv). Das Ziel der HWT wird es sein Signale, aus einer Dimension  $j$  in eine (oder mehrere) tiefere Dimensionen zu zerlegen. Diese Zerlegung wird als Multiskalenanalyse (MRA, *multiresolution analysis*) bezeichnet.

Um dies durchführen zu können ist es wichtig zu verstehen, dass  $V^{j+1}$  auch alles aus  $V^j$  enthält.  $V^j$  ist eine Untermenge des Vektorraums  $V^{j+1}$ .

$$V^0 \subset V^1 \subset V^2 \subset \dots V^{j+1} \quad (7)$$

Nachdem wir uns nun darum gekümmert haben, alle Signale in einem Vektorraum darstellen zu können, müssen wir uns um eine Basis für diesen kümmern. Basen für diese Vektorräume werden als *Skalierungsfunktionen*  $\phi$  bezeichnet. Die Basis ist eine Funktion, mit deren Hilfe wir alle Signale als Linearkombination dieser Basis-Funktion darstellen können. Bei der Fourier-Transformation ist die Basis z. B. eine Sinus-Schwingung mit verschiedenen Frequenzen und Amplituden. Wir verwenden hier jedoch keine Sinus-Schwingung, sondern eine einfache „Box“-Funktion. Diese stellt ganz einfach diskrete Werte dar mit einer Streckung  $j$  und Verschiebung  $i$ . Der zweite Abtastwert/Pixel eines Bildes mit dem Wert 4 wäre dann in der „Box“-Basis eine um eins verschobene und um vier gestreckte Box-Funktion. Diese Box-Funktion ist definiert als

$$\phi_i^j = \sqrt{2^j} \phi(2^j x - i) \quad (8)$$

mit

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{sonst} \end{cases} \quad (9)$$

Als nächstes müssen wir uns um ein inneres Produkt im Vektorraum  $V^j$  kümmern und definieren daher für zwei Elemente  $f, g \in V^j$

$$\langle f | g \rangle := \int_0^1 f(x)g(x)dx \quad (10)$$

---

<sup>1</sup> Übersetzt vom Autor

Wir haben die Basis-Funktion  $\phi_i^j$  hierbei so definiert, dass sie *normalisiert* ist. Eine Basis-Funktion  $u$  ist dann normalisiert, wenn  $\langle u|u \rangle = 1$  gilt. Also, dass ihr inneres Produkt mit sich selbst gleich 1 ist. Wir erreichen dies durch den konstanten Faktor  $\sqrt{2^j}$ , welchen wir im einleitenden Beispiel jedoch vernachlässigt haben. Im Falle dieser normalisierten Basis müssen wir anstatt einer Division durch 2 eine Division durch  $\sqrt{2}$  durchführen.

Wir können nun einen weiteren Vektorraum  $W^j$  als Komplementärraum von  $V^j$  in  $V^{j+1}$  definieren. Einfach gesagt enthält  $W^j$  all das von  $V^{j+1}$ , was nicht schon in  $V^j$  enthalten ist.

Lineare Kombinationen aus  $W^j$  werden nun als *Wavelets* bezeichnet. Diese bilden zugleich auch die Basis von  $W^j$ . Sie werden als  $\psi_i^j(x)$  notiert und haben zwei wichtige Eigenschaften:

1. Die Basisfunktionen  $\phi_i^j$  aus  $V^j$  und  $\psi_i^j$  aus  $W^j$  bilden zusammen eine Basis für  $V^{j+1}$ .
2. Jede Basisfunktion  $\psi_i^j$  aus  $W^j$  ist orthogonal zu jeder Basisfunktion  $\phi_i^j$  aus  $V^j$ .

Die erste Eigenschaft können wir uns anschaulich dadurch erklären, dass  $\psi_i^j$  ja das enthält was in  $V^j$  fehlt, um  $V^{j+1}$  darzustellen. Hier finden wir auch schon einen ersten Verweis zu unserem Beispiel. Dieses „Fehlen“ ist das gleiche „Fehlen“, welches uns zum Detailkoeffizienten gebracht hat.

Bisher haben wir jedoch noch nichts über die Funktion der Wavelets (also der Basis von  $W^j$ ) ausgesagt. Basis-Funktionen  $\psi_i^j$  der Form der Box-Funktion werden als *Haar-Wavelets* bezeichnet. Alle zuvor beschriebenen Schritte beziehen sich also auch auf alle Wavelet-Funktionen. Das Haar-Wavelet war das erste bekannte Wavelet, welches 1909 von Alfréd Haar beschrieben wurde. Weitere bekannte Wavelets sind z. B. die Daubechies-Wavelets von Ingrid Daubechies. Da wir ja die HWT untersuchen wollen, wählen wir daher  $\psi_i^j$  als

$$\psi_i^j(x) = \sqrt{2^j} \psi(2^j x - i) \quad , \quad (11)$$

mit

$$\phi(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \\ 0 & \text{sonst} \quad . \end{cases} \quad (12)$$

Die Basis-Funktion  $\psi_i^j(x)$  ist wiederum so gewählt, dass sie normalisiert ist.

Damit haben wir die mathematischen Grundlagen der HWT erschlagen. Wir ziehen nun das vorherige Beispiel mathematisch neu auf.

Unser Vier-Pixel Bild  $I$  stammt aus dem Vektorraum  $V^2$ . Wir können es als eine Linearkombination der Basis-Funktion  $\phi$  darstellen als

$$I(x) = c_0^2 \phi_0^2(x) + c_1^2 \phi_1^2(x) + c_2^2 \phi_2^2(x) + c_3^2 \phi_3^2(x) \quad . \quad (13)$$

Wir möchten dieses Signal nun aber nicht in Dimension 2 darstellen, sondern in Dimension 1. Auf jeden Fall werden wir daher zwei Linearkombination aus  $\phi_i^1$  erhalten als  $c_0^1 \phi_0^1(x) + c_1^1 \phi_1^1(x)$ . Hierbei geht uns jedoch Information verloren. Die erste wichtige Eigenschaft der Wavelets  $\psi$  war jedoch, dass wir mit ihrer Hilfe auch Vektorräume einer höheren Dimension darstellen können. Wir benötigen also noch eine Linearkombination aus  $W^1$ , welche unseren Detailkoeffizienten darstellt. Wir können daher  $I(x)$  in der Dimension 1 ausdrücken als

$$I(x) = c_0^1 \phi_0^1(x) + c_1^1 \phi_1^1(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x) \quad . \quad (14)$$

Die verbleibende Linearkombination aus  $V^1$  können wir wiederum ausdrücken als eine Linearkombination aus  $V^0$  und  $W^0$ :

$$I(x) = c_0^0 \phi_0^0(x) + d_0^0 \psi_0^0(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x) \quad , \quad (15)$$

womit wir die vollständige Dekomposition erreicht hätten. Die Koeffizienten  $c_i^j$  und  $d_i^j$  sind hierbei nichts weiteres als die gemittelten und Detail-Koeffizienten aus unserem Einleitungsbeispiel, skaliert mit dem Faktor  $\sqrt{2}$ .

**Anmerkung 1** (Orthonormalität). *Die Haar-Basis ist eine orthonormale Basis. Das heißt sie ist sowohl normalisiert (Faktor  $\sqrt{2^j}$ ) als auch orthogonal. Eine Basis gilt als orthogonal, wenn alle ihrer Basis-Funktionen zueinander orthogonal sind. D. h. es gilt für zwei zueinander orthogonale Funktionen  $f, g$   $\langle f|g \rangle = 0$ .*

### 3.2 Praktische Anwendung der Haar-Wavelet-Transformation

Wir möchten uns anhand eines praktischen Beispiels weiter mit den Eigenschaften der HWT vertraut machen.

Betrachten wir hierfür eine Sinus-Funktion im Bereich von 0 bis  $2\pi$  mit 800 Abtastwerten, welcher z. B. ein empfangenes Signal darstelle. Allerdings ist bei der Übertragung ein Fehler aufgetreten und ein Abtastwert wurde falsch zu  $-1$  übertragen.

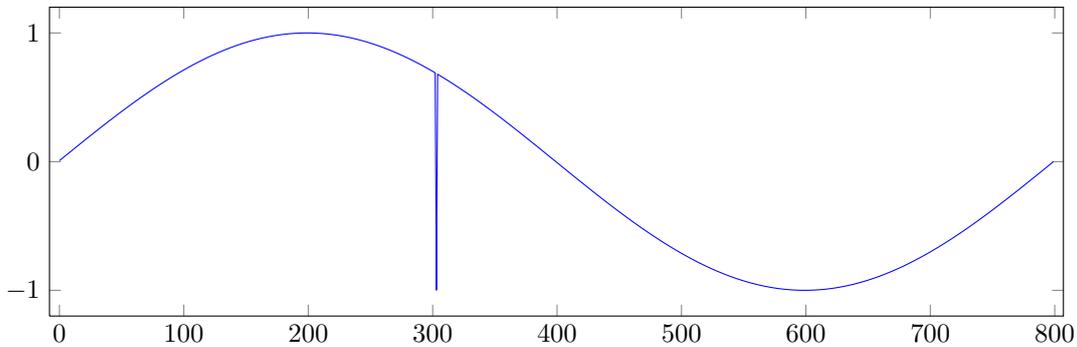


Abbildung 3: Empfangenes Sinus-Signal mit Störung.

Man kann sich den falsch übertragenen Wert auch als „Kratzer“ in einem Signal oder einzeiligem Bild vorstellen.

Wir werden im folgenden die HWT bis zur vierten Dekomposition auf dieses Signal anwenden. Zunächst wollen wir uns aber um verschiedene Darstellungsweisen der HWT kümmern.

### 3.3 Darstellungsarten der Haar-Wavelet-Transformation

Eine Art der Visualisierung der HWT ist, sie wie in Kapitel 3 ineinander verschachtelt darzustellen. Wir führen also die erste Dekomposition durch und ersetzen den gemittelten Teil wiederum durch dessen Wavelet Transformation. Dadurch bleiben die Detail-Koeffizienten  $D$  immer erhalten, die Mittelung  $A$  ist jedoch immer nur für die aktuelle Dekompositionsstufe enthalten.

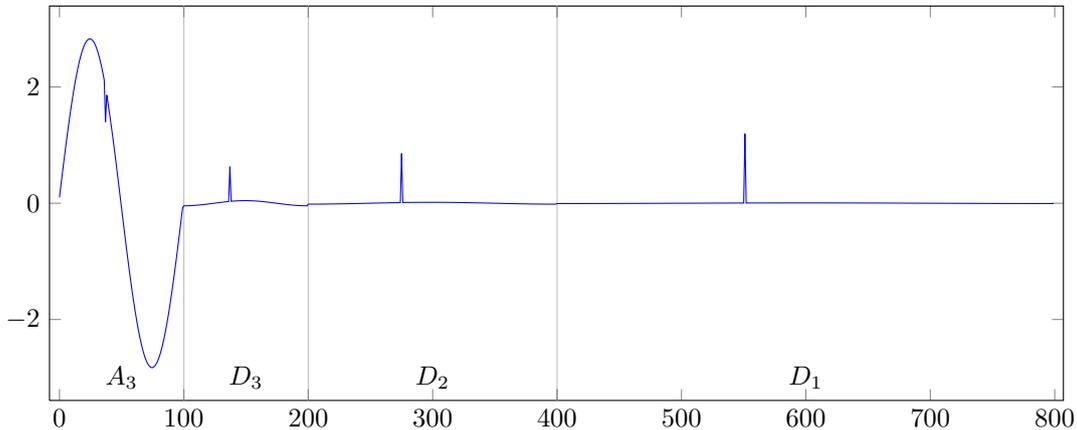


Abbildung 4: Ineinander verschachtelte Darstellungsweise der HWT

Diese Darstellungsart ist weit verbreitet in der Literatur, ist jedoch nicht sehr übersichtlich und verfehlt unser Ziel, die einzelnen Dekompositionsstufen zu betrachten.

Eine bessere Darstellungsweise ist es, den gemittelten Teil  $A$  und den Detail-Koeffizienten  $D$  separat untereinander darzustellen. Dies sieht für unser Beispiel wie folgt aus

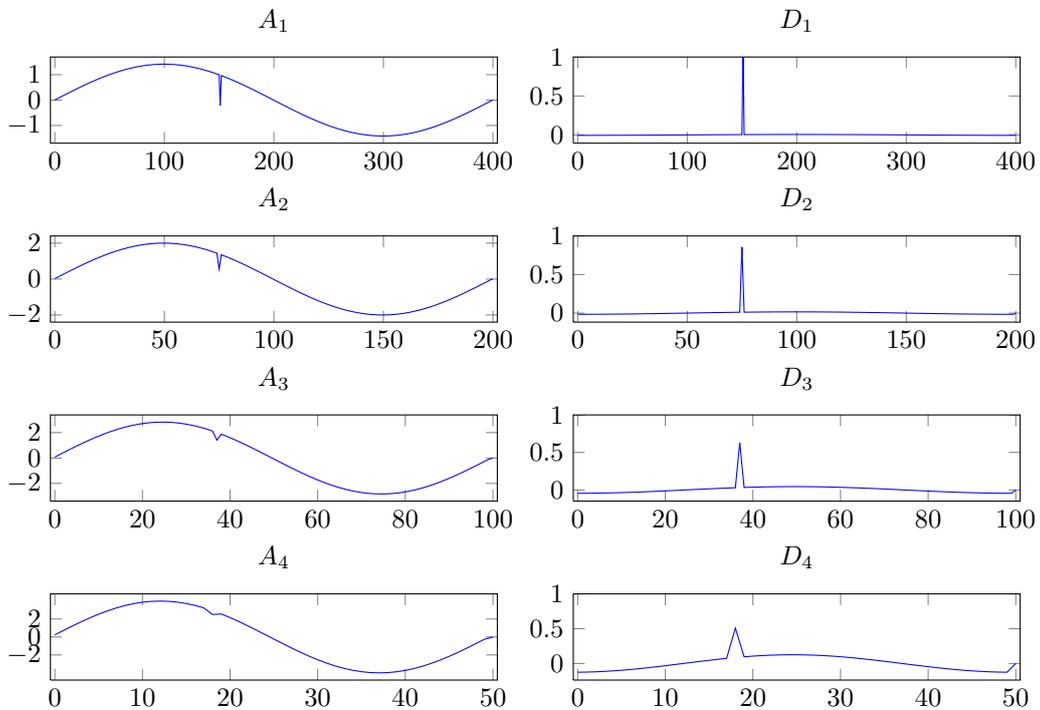


Abbildung 5: Separate Koeffizienten  $A$  und  $D$  der HWT.

Im Folgenden werden wir diese Darstellungsart gebrauchen.

### 3.4 Eigenschaften der Haar-Wavelet-Transformation

Aus der Sicht der digitalen Signalverarbeitung, aus welcher die HWT auch abstammt [SDS95a], handelt es sich bei ihr um eine Tief- und Hochpassfilterung mit nachgeschaltetem Downsampler.

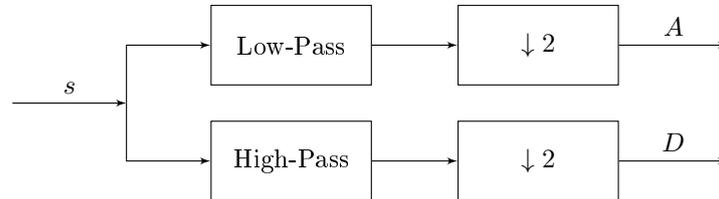


Abbildung 6: Haar-Wavelet-Transformation aus Sicht der digitalen Signalverarbeitung.

Mit dieser Erkenntnis können wir auch die einzelnen Dekompositionen unseres Beispielsignals erklären.

Wir betrachten zunächst den gemittelten Teil unseres Signals. Wir stellen fest, dass der Kratzer entfernt wird, er wird herausgemittelt, bzw. durch den Tiefpass herausgefiltert. Diese Rückgewinnung des Originalsignals geht jedoch auf Kosten der Auflösung. In jedem Dekompositionsschritt verringert sich unsere Auflösung um den Faktor zwei.

Für uns von deutlich mehr Interesse ist jedoch der Detail-Koeffizient, welcher die Hochpass-Filterung darstellt. Ein Hochpass-Filter filtert niedrige Frequenzen heraus und liefert uns hohe Frequenzen, also schnelle, abrupte Änderungen im Signal. Wir sehen, wie der Detail-Koeffizient auf den von uns eingefügten Kratzer reagiert. Diese Eigenschaft der Kantenerkennung wollen wir im Folgenden ausnutzen. Ein Beweis, dass sich die HWT zur Kantenerkennung eignet, kann in [MH92] gefunden werden.

Ein weiteres Anwendungsszenario für die HWT ist die Kompression. Die Grundidee hierbei ist, dass viele Werte im Detailkoeffizienten sehr klein sind. Wir erinnern uns, dass der Detailkoeffizient aus der „Abweichung von Mittelwert zum Ursprungswert“ entstand. Ist dieser nun sehr klein, entsteht nur ein sehr geringer Fehler, wenn wir diesen zu Null setzen würden, wodurch das Signal - verlustbehaftet - komprimiert wird. Das JPEG2000 Grafikformat basiert auf einem ähnlichen Ansatz und verwendet auch eine diskrete Wavelet-Transformation zur Kompression.

## 4 Kanten

Wir werden uns im folgenden mit den Grundlagen der Kantenerkennung und der Auswirkung verschiedener Bildfehler auf Kanten vertraut machen.

Kanten in einem Bild sind das, was ein Bild scharf erscheinen lässt. Die Präsenz und die Steilheit der Kanten in einem Bild steuert das subjektive Schärfeempfinden. [BB06]

Kanten in einem Signal, oder auch in einem Bild, sind in erster Linie rapide Veränderungen des Signals. Auf ein Bild angewendet bedeutet dies eine rapide Veränderung der Helligkeit. Wir unterscheiden hier zwei Verhalten bei der Änderung.

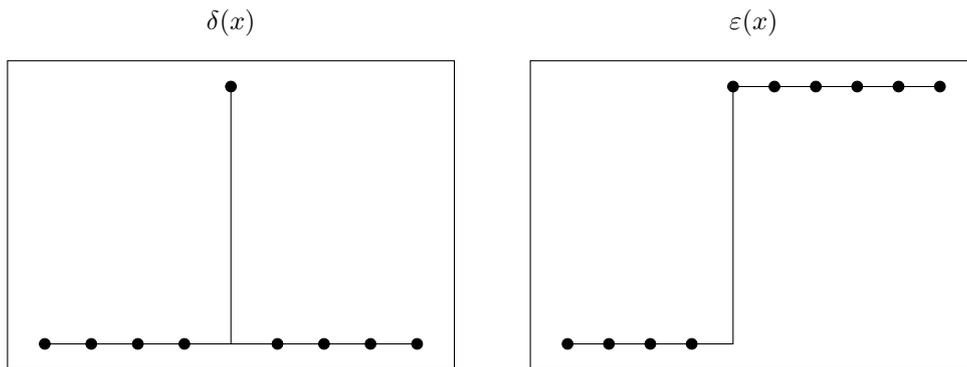


Abbildung 7: Grundarten von Kanten.

Dies ist zum einen die Dirac-Kante  $\delta(n)$ , auch „Diracs Delta Function“ oder einfach als „Impuls“ bezeichnet, mit den Eigenschaften

$$\delta(n - k) = \begin{cases} 1 & \text{für } n = k \\ 0 & \text{sonst} \end{cases} \quad (16)$$

Hier fällt die Helligkeit direkt nach der Änderung wieder auf ihre vorherige Helligkeit zurück. Dies könnte z. B. eine dünne Hochspannungsleitung in einem Bild sein.

Zum anderen die Sprungkante  $\varepsilon(n)$ , bekannt aus der Sprungfunktion, welche auch als „Heav’side Stepfunction“ bezeichnet wird. Sie stellt das Integral des Dirac-Impulses dar, bzw. umgekehrt, ist der Dirac-Impuls die Ableitung der Sprungfunktion. Für sie gilt

$$\varepsilon(n - k) = \begin{cases} 1 & \text{für } n \geq k \\ 0 & \text{sonst} \end{cases} \quad (17)$$

Hier bleibt die Helligkeit nach Änderung erhalten. Dies könnte z. B. eine weiße Mauer in einem Bild sein.

Beide dieser Kanten stellen perfekt „scharfe“ Kanten dar. Es gibt keine schnellere Änderung in der Helligkeit als mit diesen beiden möglich ist. Ab und zu wird eine „Dirac-Kante“ auch als Linie bezeichnet und nur die „Sprungkante“ als wirkliche Kante. Beide spielen jedoch für das subjektive Schärfeempfinden eines Bildes eine große Rolle, weswegen wir uns auf diese beiden konzentrieren werden.

Wir möchten im folgenden die Auswirkung der in 2.1 und 2.2 beschriebenen Unschärfen auf diese Kanten und deren Haar-Wavelet Dekompositionen untersuchen.

## 5 Weichzeichnung

### 5.1 Gauß'sche Unschärfe

Wir widmen uns im Folgenden der in 2.1 beschriebenen Unschärfe. Es handelt sich bei Out-Of-Focus-Blur um eine (partielle) Weichzeichnung eines Bildes. Wir werden die Auswirkung hiervon im Eindimensionalen untersuchen und sie auf zweidimensionale Bilder verallgemeinern.

Wir benötigen also ein gutes Filtermodell, um die Weichzeichnung, welche von Objektiven erzeugt wird, zu modellieren.

Ein sehr einfacher Weichzeichnungsfiler ist z. B. ein Mittelungsfiler mit der Übertragungsfunktion  $H(n) = [1 \ 1 \ 1]$ . Dieser liefert uns aber nur ein unzureichendes Modell des Out-Of-Focus-Blurs.

Wir verwenden im folgenden einen Gauß'schen Weichzeichner um das Bokeh eines Bildes zu simulieren. Wir gehen hierbei jedoch von Linsenobjektiven aus, da Spiegelobjektive ein komplett anderes Bokeh liefern.

Ein Gauß'scher Weichzeichner führt eine gewichtete Mittelung anhand der Gauß'schen Glockenkurve für jeden Punkt eines Bildes/Signals aus. Für die Filterfunktion  $H$  gilt somit in einer Dimension

$$H_{Gauss}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad , \quad (18)$$

welche die Standardnormalverteilung mit Erwartungswert 0 und Standardabweichung  $\sigma$  darstellt.  $\sigma$  ist hierbei der Grad der Weichzeichnung,  $x$  der Abstand vom Ursprung.

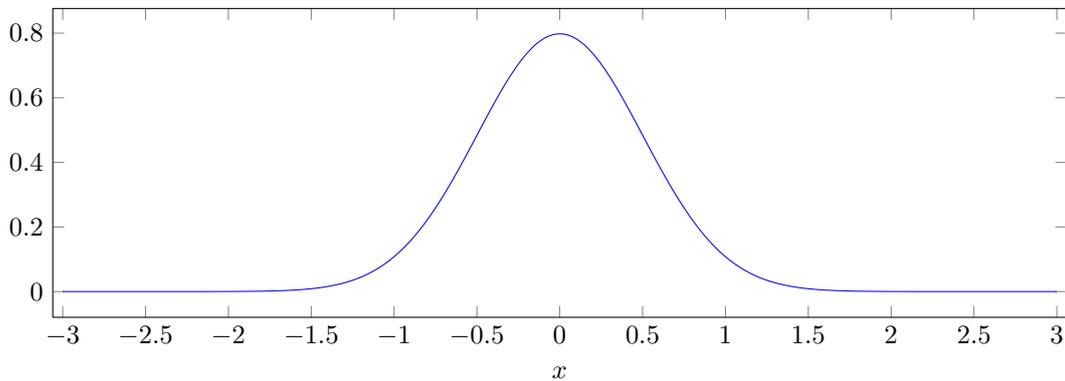


Abbildung 8: Gauß-Kern für  $\sigma = 0.5$ .

Für zwei Dimensionen ergibt sich die Filterfunktion einfach durch eine Weichzeichnung in vertikaler und anschließend horizontaler Richtung - oder umgekehrt. Wodurch wir die zweidimensionale Filterfunktion des Gauß'schen Weichzeichners berechnen können zu

$$\begin{aligned} H_{Gauss}(x, y) &= H_{Gauss}(x) \cdot H_{Gauss}(y) \\ &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} \\ &= \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad , \quad (19) \end{aligned}$$

mit dem horizontalen Abstand  $x$  zum Ursprung und dem vertikalen Abstand  $y$ .

Für  $\sigma = 0.5$  sei hier der zweidimensionale Gauß-Kern in Abbildung 9 gegeben.

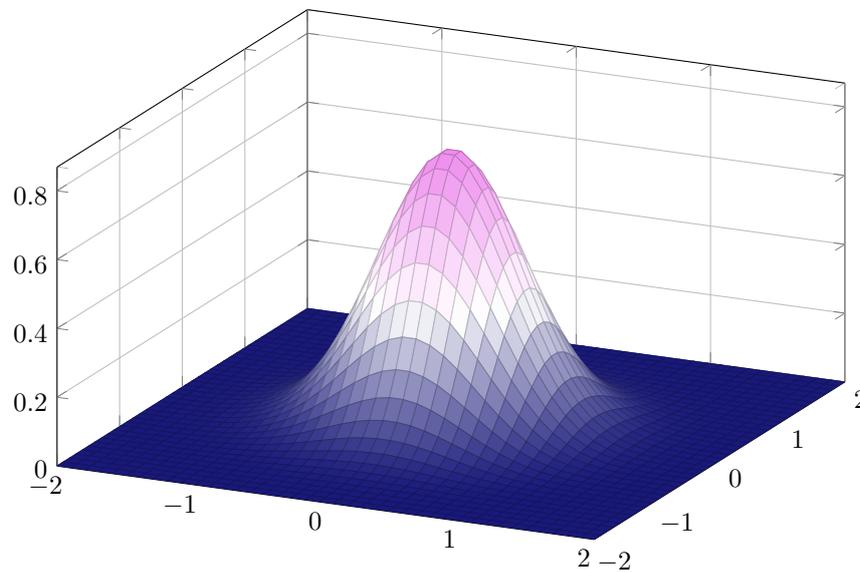


Abbildung 9: Zweidimensionaler Gauß-Kern  $\sigma = 0.5$ .

Um mit diesen Filterübertragungsfunktionen nun ein Bild oder Signal weichzuzeichnen, ist es üblich aus diesen eine „kleine“ Filtermaske zu erstellen und diese mit dem Signal oder Bild zu falten. Für Bilder sind Masken der Größe  $3 \times 3$  üblich.

Im folgenden untersuchen wir nun die Auswirkungen dieses Filters auf die zwei Kantentypen aus 4 und deren Dekompositionen.

**Anmerkung 2.** Diese Filterung ist deutlich effizienter, wenn man die Filterung mittels zweier eindimensionaler (anstatt einer zweidimensionalen) Filterungen durchführt. Konkret würde man hier statt der Faltung mit  $H_{Gauss}(x, y)$  zunächst mit  $H_{Gauss}(x)$  und anschließend mit  $H_{Gauss}(y)$ , aut vice versa, filtern. Der Rechenaufwand reduziert sich hierdurch enorm.

Dies bezieht sich auch auf die später behandelte Scharfzeichnung, da hier auch eine Weichzeichnung durchgeführt werden wird. Da die Scharfzeichnung jedoch letztendlich nur einen Bruchteil der Rechenzeit in unserem Algorithmus beansprucht, wurde hierauf verzichtet.

## 5.2 Weichgezeichneter Dirac-Impuls

Um eine bessere Vorstellung der theoretischen Dirac-Kante zu erhalten, erstellen wir ein einfaches Bild mit einer einzigen Dirac-Kante und zeichnen dieses weich.



Abbildung 10: Zweidimensionales Bild einer Dirac-Kante.

Dies ist eine Vergrößerung eines  $5 \times 1$  Pixel großen Bildes, in dem der mittlere Strich unsere Dirac-Kante darstellt. Für einen Pixel ändert sich hier in horizontaler Richtung die Helligkeit des Bildes von 1 auf 0. Auf diesen Dirac-Impuls wurde mittels GIMP ein Gauß'scher Weichzeichner angewendet mit einem Radius von 2 Pixeln. Dieses Bild soll nur dem besseren Verständnis unserer Kanten und dem Gauß'schen Weichzeichner dienen. Wie es entsteht und wie wir es mathematisch betrachten können, werden wir im Folgenden untersuchen.

Um einen Dirac-Impuls weichzuzeichnen, könnten wir eine kleine Filtermaske (z. B.  $5 \times 1$ ) erstellen und einen Dirac-Impuls hiermit falten. Da der Dirac-Impuls jedoch das neutrale Element der Faltung ist, erhalten wir dadurch die Filtermaske selbst.

Dies wäre jedoch recht ungenau und würde die „analoge“ Weichzeichnung, wie sie im Objektiv entsteht, nur unzureichend simulieren. Wir falten daher den Dirac-Impuls mit dem Gauß-Kern und erhalten dadurch die kontinuierliche Repräsentation eines weichgezeichneten Dirac-Impulses. Anders gesagt: Wir wählen eine unendlich große Filtermaske und falten mit dieser. Erst dann diskretisieren wir das Ergebnis. Wie wir die Diskretisierung durchführen, folgt später.

Wir haben mit dem Gauß-Kern also eine passende Funktion zur Simulation eines weichgezeichneten Dirac-Impulses gefunden. Als Beispiel finden sich in Abbildung 11 mehrere Dirac-Kanten in verschiedenen Weichzeichnungsstufen.

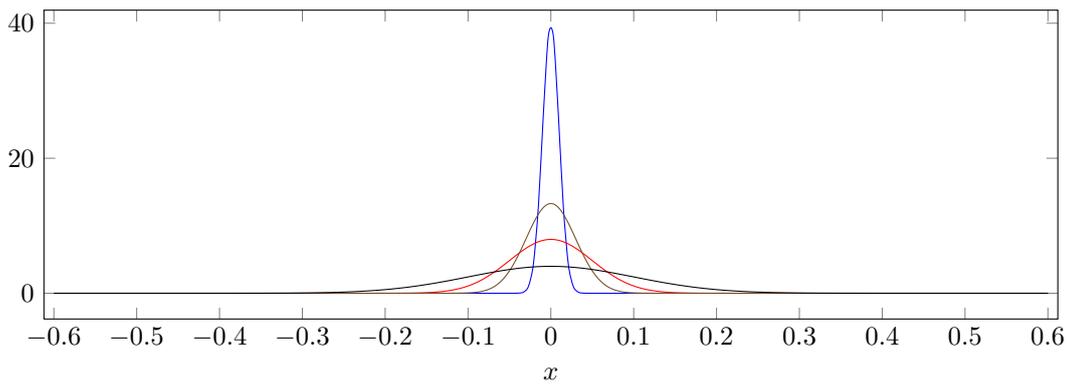


Abbildung 11: Weichgezeichnete Dirac-Impulse für  $\sigma = 0.01$ ,  $\sigma = 0.03$ ,  $\sigma = 0.05$ ,  $\sigma = 0.1$ .

### 5.2.1 HWT des idealen Dirac-Impulses

Betrachten wir nun einmal die HWT eines idealen, diskreten Dirac-Impulses bis hin zur vierten Dekomposition. Wir legen hierbei Augenmerk auf den Detail-Koeffizienten.

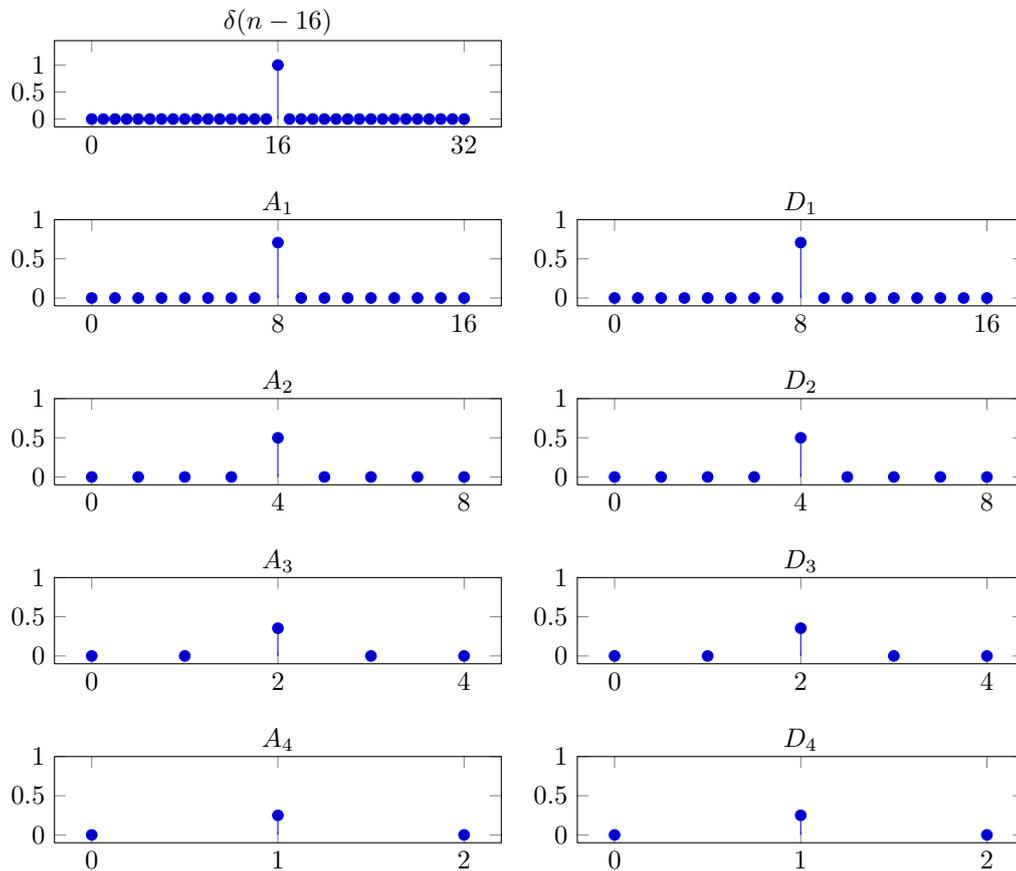


Abbildung 12: Idealer Dirac-Impuls in der HWT.

Zur besseren Vergleichbarkeit wurde die Y-Achsenkalierungen der Koeffizienten jeweils konstant gehalten.

Wir sehen, dass je tiefer wir die Dekomposition ausführen, umso kleiner wird der Dirac-Impuls und umso kleiner wird auch der Ausschlag im Detail-Koeffizienten.

### 5.2.2 HWT des weichgezeichneten Dirac-Impulses

Als nächstes betrachten wir zum Vergleich die HWT eines mit  $\sigma = 0.05$  weichgezeichneten Dirac-Impulses. Da wir hier die diskrete HWT behandeln, müssen wir die kontinuierliche Funktion diskretisieren. Hierbei legen wir Wert darauf, möglichst genau zu arbeiten. Theoretisch erstreckt sich der Gauß-Kern von  $-\infty$  bis  $\infty$ , da wir jedoch keinen unendlich großen Bereich berechnen können, begrenzen wir unser Signal hier auf  $-40\sigma$  bis  $40\sigma$ , wir können davon ausgehen, dass der Gauß-Kern an diesen Grenzen nahe Null ist. In der Praxis wird oft der Bereich  $-3\sigma$  bis  $3\sigma$  gewählt, welcher 99.7% des gesamten Kernes abdeckt (theoretisch Berechnung über das Integral). Da wir in diesem Bereich jedoch nur in diskreten Schrittweiten vorangehen, bzw. abtasten können, müssen wir auch die wichtige Eigenschaft der Standardnormalverteilung beachten, dass ihr Integral von  $-\infty$  bis  $\infty$  gleich 1 ist. Hierfür normieren wir unser abgetastetes Signal so, dass das diskrete Integral auch 1 ergibt. In unserem Beispiel beträgt die Schrittweite 0.01.

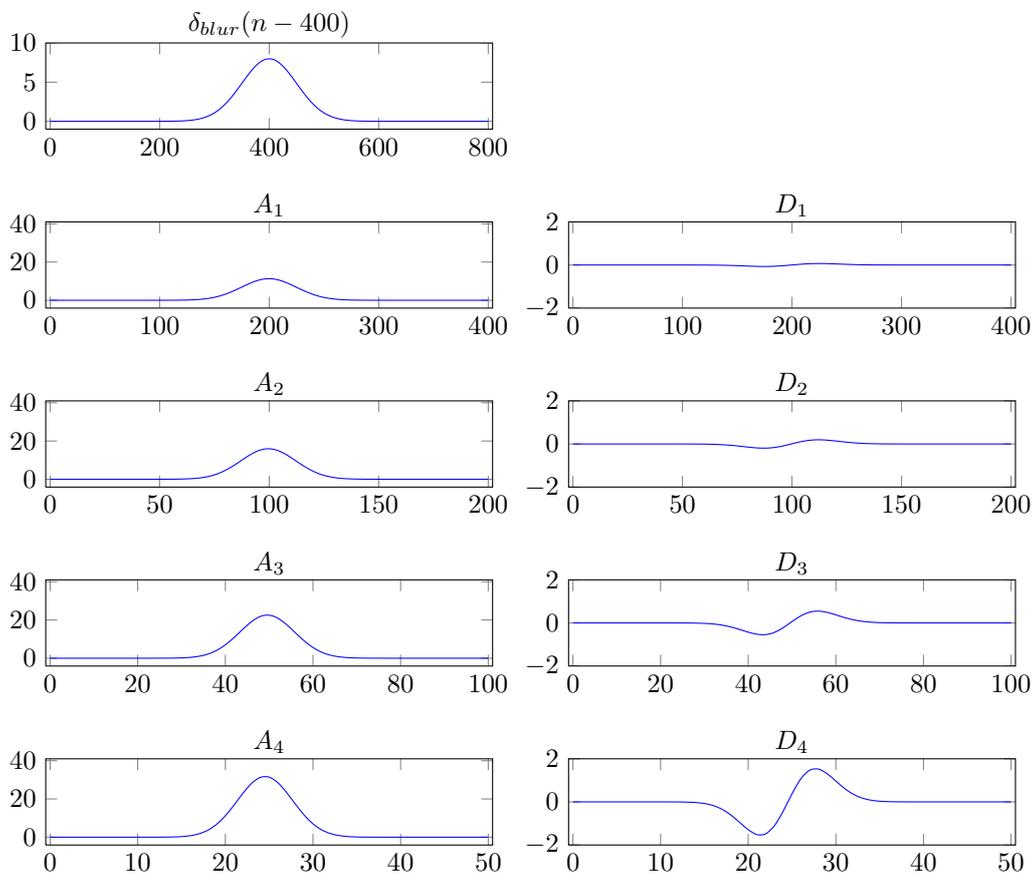


Abbildung 13: Weichgezeichneter Dirac  $\sigma = 0.05$  in der HWT.

Wieder wurden die Y-Achsenkalierungen konstant gehalten. Wie wir sehen, erhalten wir hier zwei Ausschläge im Detail-Koeffizienten. Dies sind die Ausschläge der An- und Abstände der Kante.

Uns fällt auf, dass wir diesmal eine umgekehrte Intensität in den Dekompositionen erhalten. Die Intensität steigt mit jeder Dekomposition. Wir können uns dieses Verhalten dadurch erklären, dass durch die Mittelung (siehe dem Average-Koeffizienten links) der weichgezeichnete Dirac-Impuls - zu Kosten der Auflösung - seine Schärfe zurückerhält.

Wir hegen den Verdacht, dass wir anhand der Intensitäten der Dekompositionen Aussagen über die Kanten-Art treffen können.

### 5.2.3 Evaluation Dirac-Impuls

Um unsere Vermutung zu bestätigen, führen wir eine Evaluation mit verschiedenen  $\sigma$  durch. Wir werden mit einem sehr kleinen  $\sigma$  nahe 0 beginnen, sodass wir möglichst mit einem idealen Dirac-Impuls starten und  $\sigma$  fortlaufend erhöhen. Für jedes  $\sigma$  werden wir die Dekomposition bis zur vierten Stufe durchführen und jeweils das Betrags-Maximum der  $i$ . Dekomposition notieren. Wir bezeichnen in Analogie zu Tong et al. das Maximum der  $i$ . Dekomposition als  $E_{max_i}$ .

Die Evaluation von  $\sigma = 0.00001$  bis  $\sigma = 0.2$ , über einen Bereich von  $-3\sigma_{max}$  bis  $3\sigma_{max}$  mit einer Auflösung von 0.01 führt uns zu folgendem Ergebnis:

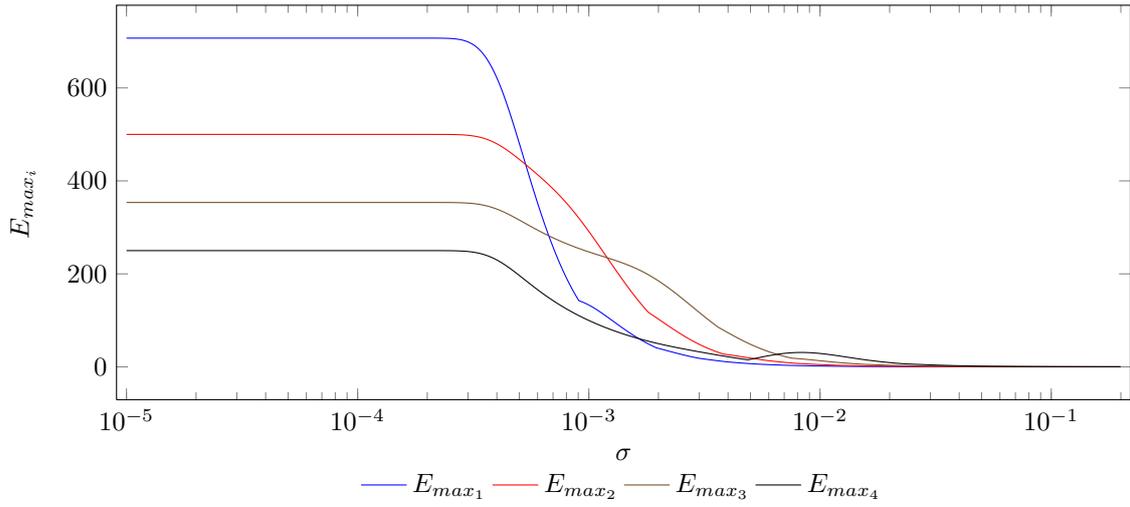


Abbildung 14: Evaluation des weichgezeichneten Dirac-Impulses.

Wir können aus unserer Evaluation mehrere Erkenntnisse gewinnen. Zum einen sehen wir, wie sich die HWT zur Kantenerkennung mittels des Detail-Koeffizienten eignet, da erst für hohe  $\sigma$ -Werte  $E_{max_i}$  gegen 0 geht. Wir erkennen auch, dass wir, für eine gegebene Grenze zur Kantenerkennung, mit tieferen Dekompositionen länger Kanten erkennen können; wir sind also auch in der Lage ehemals scharfe Kanten zu erkennen.

Eine weitere, deutlich wichtigere Erkenntnis können wir jedoch aus den einzelnen Kurvenverläufen ziehen. Wir sehen, dass unsere Vermutung bestätigt wurde, dass sich die Intensitätsverteilung mit steigendem  $\sigma$  umdreht. Gilt für kleine  $\sigma$  noch, dass  $E_{max_1}$  am größten ist, beginnt es ab einem bestimmten  $\sigma$  zu fallen. Es sei hier angemerkt, dass der Schnittpunkt der Graphen je nach Auflösung variiert, wir also kein genaues  $\sigma$  angeben können. Jedoch liegt stets derselbe Kurvenverlauf vor. Wir sehen, dass zur Unterscheidung einer scharfen Kante von einer unscharfen (also weichgezeichneten) Kante der Vergleich zwischen  $E_{max_1}$  und  $E_{max_2}$  reicht. Wir könnten aber auch  $E_{max_3}$  und  $E_{max_4}$  miteinbeziehen.

Zusammengefasst kommen wir aus dieser Evaluation zu folgenden Erkenntnissen

**Erkenntnis 1.** *Liegt ein  $E_{max_i}$  über einer festgelegten Grenze, kann von einer Kante gesprochen werden.*

**Erkenntnis 2.** *Für kleine  $\sigma$  gilt  $E_{max_1} > E_{max_2} > E_{max_3} > E_{max_4}$ .*

**Erkenntnis 3.** *Eine scharfe Kante kann mittels  $E_{max_1} > E_{max_2}$  erkannt werden.*

**Erkenntnis 4.** *Fällt  $E_{max_1}$  unter  $E_{max_2}$ , so handelt es sich um eine weichgezeichnete Kante.*

### Normierung der Dirac-Impulse

Die von uns evaluierten Erkenntnisse könnten insofern kritisiert werden, dass argumentiert werden könnte, wir hätten Äpfel mit Birnen verglichen. Da bei der Weichzeichnung die Intensität nachlässt, zeigen die Kurven nur diesen Verlauf, weil das Signal insgesamt schwächer würde.

Dass dies nicht stimmt, kann ganz einfach dadurch gezeigt werden, indem wir den Vorfaktor  $\frac{1}{2\pi\sigma^2}$  des Gauß-Kernes vernachlässigen und so stets weichgezeichnete Dirac-Impulse der Höhe 1 erzeugen. Die Evaluation dessen zeigt die selben Ergebnisse. Unsere Erkenntnisse sind hier sogar noch deutlicher zu erkennen.

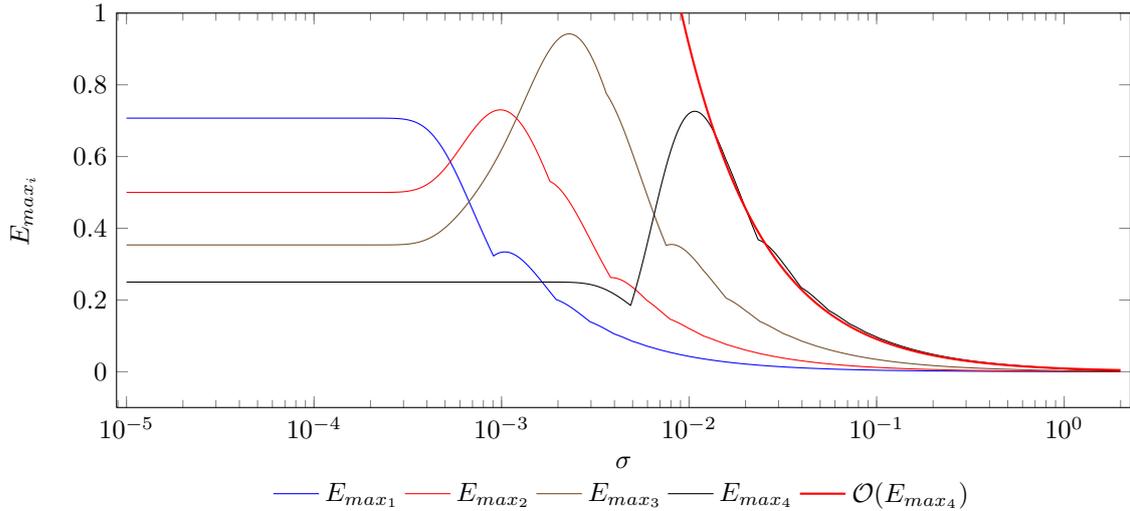


Abbildung 15: Evaluation des auf 1 normierten, weichgezeichneten Dirac-Impulses.

Da an dieser Evaluation die Kurvenverläufe sehr gut erkennbar sind, können wir hier auch eine kurze Kurvendiskussion durchführen. Wie wir sehen, entsteht anfangs ein Plateau, da hier die Weichzeichnung sehr schwach ist und die Diskretisierung dazu führt, dass ein idealer Dirac-Impuls erhalten bleibt. Erst für größere  $\sigma$  überwiegt die Weichzeichnung die Diskretisierung und die Graphen fangen an zu fallen. Für große  $\sigma$  approximiert der Detailkoeffizient - skaliert - die erste Ableitung der Funktion, da der gebildete Differenzenquotient dann sehr genau dem Differentialquotienten zwischen zwei Abtastwerten entspricht. Wir können daher das asymptotische Verhalten der  $E_{max_i}$ -Funktionen analytisch mittels einer Extremwertbetrachtung berechnen. So erhalten wir durch die Nullstellen der zweiten Ableitung des auf 1 normierten Gauß-Kernes bei  $\pm\sigma$  ein Verhalten von  $\sqrt{e^{-1}}\sigma^{-1}$ . Für  $E_{max_4}$  ist dieses asymptotische Verhalten in der Evaluation eingetragen; es wurde mit 0.015 skaliert und entspricht sehr genau dem Verhalten der Kurven. Selbige Betrachtung kann auch für die anderen Evaluationen durchgeführt werden, da dort die Kurven jedoch sehr schnell gegen Null gehen, ist das Verhalten nicht so gut wie hier zu erkennen.

**Anmerkung 3** (Knicke in den Graphen). *Die Knicke in den Evaluations-Graphen sind auf den ersten Blick sehr merkwürdig. Sie sind nicht auf Numerik-Fehler zurückzuführen, sondern zeigen eine Eigenschaft der Weichzeichnung eines Dirac-Impulses. Wie wir im zweidimensionalen Bild der Dirac-Kante sehen, entstehen durch die Weichzeichnung neben der eigentlichen Kante weitere Pixel mit schwächeren Grauwerten. Die Grauwerte nehmen mit der Gauß-Kurve nach Links und Rechts ab. Das heißt, dass die direkt benachbarten Pixel einen fast identischen Grauwert haben, die darauf folgenden jedoch rapide abfallen. In den Fällen, in denen die Knicke auftreten, wechselt dann der Ort der erkannten und analysierten Kante. Die Übergänge vom  $n+1$ . zum  $n+2$ . Pixel stellen dann eine stärkere Kante dar als der Übergang der Ursprungskante von  $n$  nach  $n+1$ . Man könnte auch sagen, dass die weichgezeichnete Kante wandert und bei einem Knick eine neue Evaluation einer neuen Kante auftritt.*

### 5.3 Weichgezeichneter Sprung

Wir stellen die selben Überlegungen wie für den Dirac-Impuls für die Sprungfunktion an. Zunächst möchten wir uns wieder eine Vorstellung einer Sprungkante verschaffen.



Abbildung 16: Zweidimensionales Bild einer Sprungkante.

Wie bei der Dirac-Kante haben wir hier eine abrupte Änderung der Helligkeit. Im Gegensatz zur Dirac-Kante fällt diese jedoch nicht sofort wieder ab, sondern behält ihre Helligkeit bei. Auch die Auswirkung des angewandten Gauß'schen Weichzeichnungsfilters sind unterschiedlich. Während die weichgezeichnete Dirac-Kante auch seine Helligkeit verlor, so bleibt die End-Intensität unserer Sprungkante erhalten, nur die Steigung wird „weicher“. Während wir bei der scharfen Sprungkante eine Änderung der Helligkeit (Ableitung) von theoretisch unendlich haben (die Ableitung einer Sprungfunktion ist ein Dirac-Impuls), haben wir bei der weichgezeichneten Sprungkante eine endliche Steigung.

Wir möchten nun zunächst wieder ein kontinuierliches Modell zur Simulation einer Weichzeichnung einer Sprungkante entwickeln. Hierzu falten wir die Sprungfunktion mit dem Gauß-Kern

$$\varepsilon_{blur}(x) = \varepsilon(x) * H_{Gauss}(x). \quad (20)$$

Da wir wissen, dass die Sprungfunktion das Integral über den Dirac-Impuls ist, also

$$\varepsilon(x) = \int_{-\infty}^x \delta(t) dt \quad , \quad (21)$$

können wir weiterhin schreiben

$$\varepsilon_{blur}(x) = \int_{-\infty}^x \delta(t) dt * H_{Gauss}(x) \quad , \quad (22)$$

was sich vereinfacht zu

$$\varepsilon_{blur}(x) = \int_{-\infty}^x H_{Gauss}(t) dt \quad . \quad (23)$$

Mit Gleichung 18 ergibt sich

$$\varepsilon_{blur}(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^x e^{-\frac{t^2}{2\sigma^2}} dt \quad . \quad (24)$$

Welche uns zur Verteilungsfunktion der Normalverteilung, bzw. zur Fehlerfunktion  $\text{erf}(x)$  führt.

Da die Fehlerfunktion  $\text{erf}(x)$  definiert ist als

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad , \quad (25)$$

müssen wir eine Variablen-Substitution von  $t' = \frac{t}{\sqrt{2}\sigma}$  durchführen. Dadurch gilt  $\frac{dt'}{dt} = \frac{1}{\sqrt{2}\sigma}$  und für die neue obere Grenze somit  $\frac{x}{\sqrt{2}\sigma}$ .

Wir erhalten somit:

$$\varepsilon_{blur}(x) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^x e^{-\left(\frac{t}{\sqrt{2}\sigma}\right)^2} dt \cdot \frac{1}{\sqrt{2}\sigma} \quad (26)$$

$$= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\frac{x}{\sqrt{2}\sigma}} e^{-t'^2} dt' \cdot \sqrt{2}\sigma \cdot \frac{1}{\sqrt{2}\sigma} \quad (27)$$

$$= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\frac{x}{\sqrt{2}\sigma}} e^{-t'^2} dt' \quad (28)$$

$$= \frac{1}{\sqrt{\pi}} \left( \int_0^{\frac{x}{\sqrt{2}\sigma}} e^{-t'^2} dt' + \int_{-\infty}^0 e^{-t'^2} dt' \right) \quad (29)$$

Mit Gleichung 25 können wir dies ausdrücken können als

$$\varepsilon_{blur}(x) = \frac{1}{2} \operatorname{erf} \left( \frac{x}{\sqrt{2}\sigma} \right) - \frac{1}{2} \operatorname{erf}(-\infty) \quad (30)$$

$$= \frac{1}{2} \left( \operatorname{erf} \left( \frac{x}{\sqrt{2}\sigma} \right) + 1 \right) \quad (31)$$

Diese Gleichung wird auch in manchen Veröffentlichungen des *International Journal of Computer Vision* zur Simulation einer Sprungkante verwendet.

Abbildung 17 veranschaulicht unsere Berechnung für verschiedene  $\sigma$ . Es spiegelt auch unsere Beobachtungen aus dem Zweidimensionalen wieder.

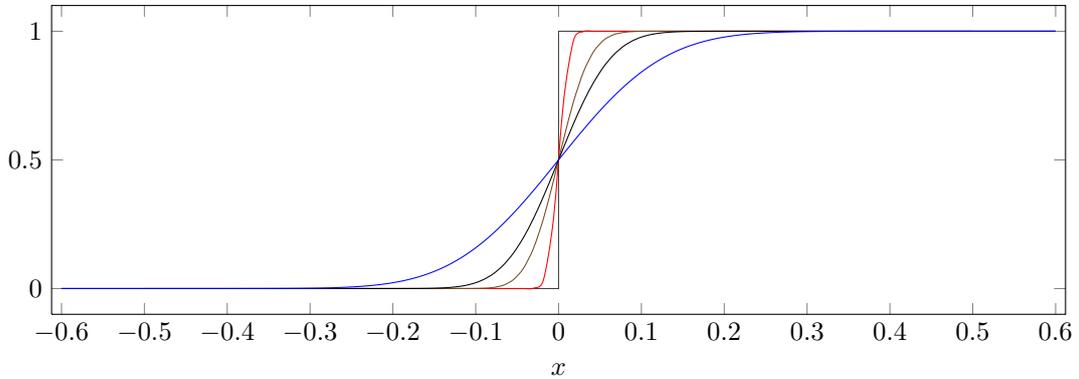


Abbildung 17: Weichgezeichnete Sprung-Kanten für  $\sigma = 0.01$ ,  $\sigma = 0.03$ ,  $\sigma = 0.05$ ,  $\sigma = 0.1$ .

## 5.3.1 HWT der idealen Sprungkante

Wir betrachten zunächst wieder eine ideale Sprungkante in der Haar Dekomposition:

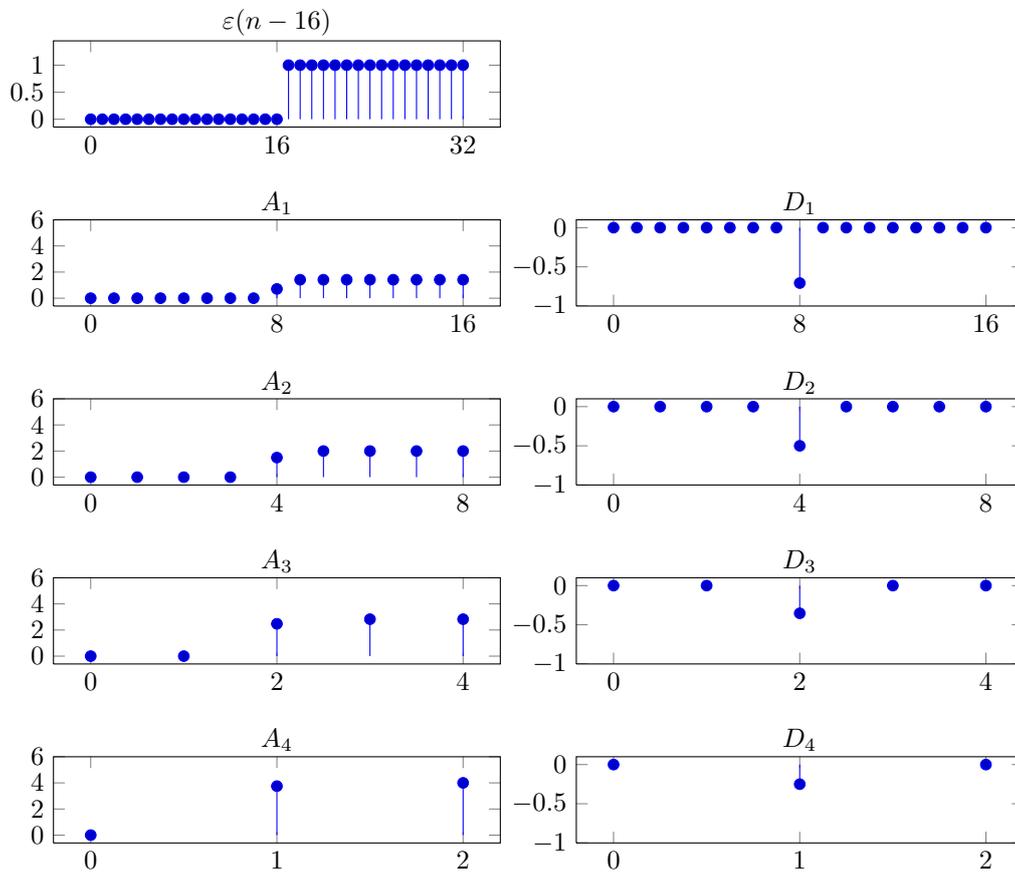


Abbildung 18: Ideale Sprung-Kante in der HWT.

Wie auch beim Dirac-Impuls erkennen wir hier die Tendenz der absteigenden Intensität bei tieferer Dekomposition.

Dass die Ausschläge diesmal negativ sind, soll uns nicht weiter stören, da wir im späteren Verlauf nur noch das Betragsmaximum, bzw. das quadratische Mittel betrachten werden.

Ein Unterschied ist jedoch, dass der Average-Koeffizient links ein ganz anderes Verhalten aufweist. Bei dem Dirac-Impuls nahm die Intensität immer weiter ab. Es blieb jedoch ein idealer Dirac-Impuls erhalten. Beim Sprung stellen wir jedoch fest, dass zum Einen die Intensität zunimmt und zum Anderen die Steigung endlich wird, also kein idealer Sprung mehr erhalten bleibt.

### 5.3.2 HWT der weichgezeichneten Sprungkante

Die selbe Betrachtung führen wir auch für den mit  $\sigma = 0.05$  weichgezeichneten Sprung durch, welche sich wie folgt darstellt:

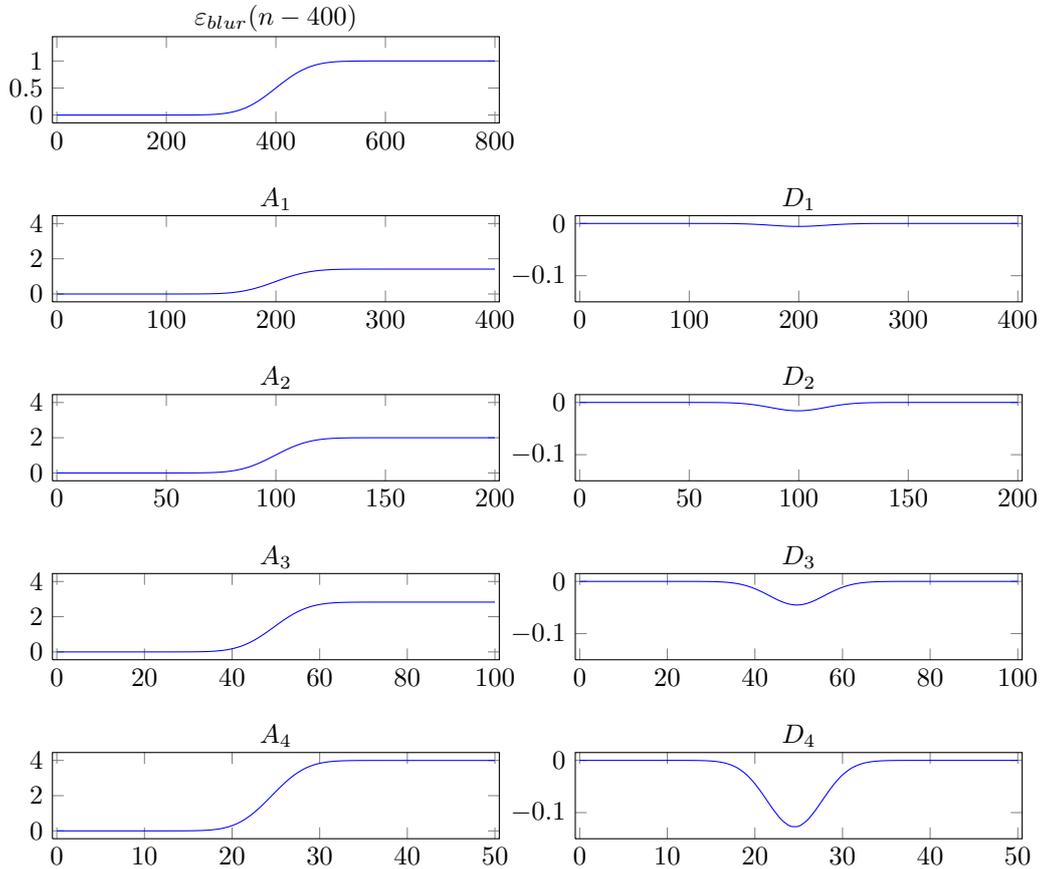


Abbildung 19: Weichgezeichneter Sprung  $\sigma = 0.05$  in der HWT.

Auch hier stellen wir dieselbe Tendenz im Detail-Koeffizienten wie beim weichgezeichneten Dirac-Impuls fest.

Wir erhalten diesmal jedoch jeweils nur einen Ausschlag im Detail-Koeffizienten. Während es bei der Dirac-Kante einen An- und Abstieg gab, gibt es bei der Sprungkante entweder nur einen An- oder nur einen Abstieg. Man kann auch sagen, dass eine Dirac-Kante eigentlich aus zwei Kanten besteht, da ihr Anstieg und Abstieg jeweils eine eigene Kante ist.

Sprung- und Dirac-Kanten zeigen außerdem ein unterschiedliches Verhalten im Average-Koeffizienten. Bei der idealen Dirac-Kante verhielten sich beide Koeffizienten umgekehrt zur weichgezeichneten Kante. Bei der Sprungkante jedoch verhält sich der Average-Koeffizient identisch; die Intensität nimmt zu. Dies heißt, dass Dirac-Kanten (Linien) in niedrigeren Auflösungen verschwinden und Sprungkanten verstärkt werden.

Um das Verhalten des Detail-Koeffizienten wieder zu überprüfen, führen wir im Folgenden die gleiche Evaluation wie beim Dirac-Impuls durch. Wir verzichten jedoch auf  $E_{max_4}$ , da uns dieser keine weitere Information liefern konnte.

### 5.3.3 Evaluation Sprungkante

Wir evaluieren das Verhalten des Detail-Koeffizienten wiederum und wählen dieselben Simulationsparameter wie bei der Dirac-Evaluation zuvor.

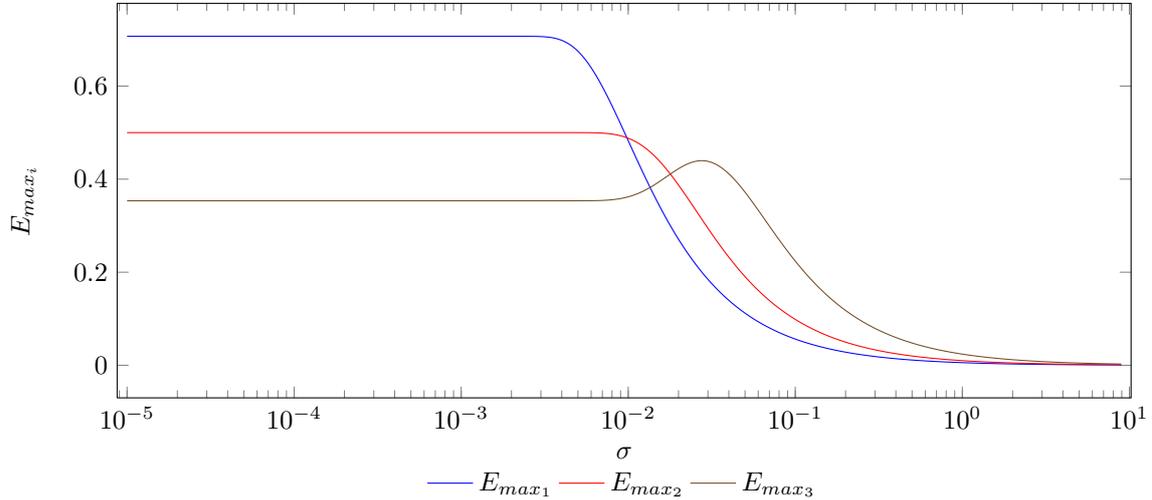


Abbildung 20: Evaluation des weichgezeichneten Sprung.

Wir stellen fest, dass unsere Hoffnung wieder erfüllt wurde. Anhand der Kurvenverläufe können wir wieder eine weichgezeichnete Sprungkante von einer scharfen Sprungkante unterscheiden. Es zeigen sich jedoch auch Unterschiede zur Evaluation der Dirac-Kante.

Während bei der Dirac-Evaluation die  $E_{max_i}$  sehr schnell gegen 0 gingen - die Dirac-Impulse verloren durch ihre Weichzeichnung ja ihre Intensität - gehen die  $E_{max_i}$  langsamer gegen 0. Wir können für eine gegebene Grenze hier 10 mal länger (logarithmische Skalierung) Kanten erkennen. Aus den Kurvenverläufen können wir wieder die selben Schlüsse ziehen wie zuvor, welche uns zu folgenden Erkenntnissen führen:

**Erkenntnis 5.** *Weichzeichnungen wirken sich auf Sprungkanten schwächer aus als auf Dirac-Kanten.*

**Erkenntnis 6.** *Um eine scharfe Sprungkante zu erkennen, gelten die selben Beziehungen wie bei Dirac-Kanten.*

**Anmerkung 4.** *Um den kontinuierlich weichgezeichneten Sprung diskret darzustellen, ist es wichtig eines zu beachten. Mit dem Faktor  $\sigma$  möchten wir die Steilheit im Nulldurchgang beeinflussen. Wir erwarten, dass dieser (bei einer endlichen Auflösung und Quantisierung) bei einem  $\sigma$  gegen 0 einen idealen Sprung von [... 000111 ...] liefert. Dieser sollte abschwächen, wenn wir  $\sigma$  erhöhen. Das Problem hierbei ist jedoch, dass die Fehlerfunktion bei  $x = 0$  den Wert 0.5 besitzt. Wird daher ein Intervall gewählt, in welchem  $x = 0$  als Punkt mitberechnet wird, so ist der steilste Sprung, den wir erhalten können [... 000.5111 ...], welcher nicht sehr steil ist. Wir wählen daher ein Intervall, welches symmetrisch um 0 liegt, jedoch  $x = 0$  nicht mitberechnet. Wir erhalten daher für kleine  $\sigma$  das von uns gewünschte Verhalten von [... 000.10.911 ...] bis [... 000111 ...].*

## 5.4 Über die Vergleichbarkeit verschiedener Dekompositionsebenen

Wir haben bisher immer nur einzelne Kanten untersucht. In den einzelnen Dekompositionen haben wir dann jeweils nur das Maximum betrachtet. In der Realität haben wir es natürlich nicht nur mit einer einzigen Kante zu tun, weshalb der Vergleich der Maxima keinen Sinn ergeben würde.

Wir müssen daher jeweils die einzelnen Dekompositionsstufen miteinander punktweise vergleichen. Nun verringert sich die Auflösung mit jeder Dekompositionsstufe jedoch um den Faktor 2. Jeweils zwei Punkte werden somit auf einen Punkt der nächsten Dekompositionsstufe abgebildet. Blicken wir daher von unten nach oben: Haben wir einen Punkt in der dritten Dekompositionsstufe an der Position  $i$  gefunden, und möchten diese mit dem zugehörigen Punkt in der zweiten Dekompositionsstufe vergleichen, so haben wir zwei Möglichkeiten - Punkt  $i \cdot 2$  und  $i \cdot 2 + 1$ .

Um diese Vergleichbarkeit zu gewährleisten, müssen wir daher entweder die tiefste Dekomposition auf die Auflösung der höchsten Dekomposition bringen, z. B. durch Pixelwiederholung oder Interpolation. Oder wir verringern die Auflösung aller Dekompositionsstufen auf die kleinste, mit der wir vergleichen wollen.

Wir wählen für unser Verfahren letztere Überlegung. Dies bedeutet für uns, dass sich für  $n$  Dekompositionsstufen die Auflösung um den Faktor  $2^n$  verringert. Wenn wir also bis zur dritten Dekompositionsstufe vergleichen werden, wird sich die Auflösung unseres Bildes um Faktor 8 verkleinern.

Um die Verkleinerung durchzuführen, werden wir in der  $i$ . Dekomposition jeweils  $2^{n-i}$  Werte zusammenführen, indem wir deren Maximum bestimmen. Dies heißt bei einer maximalen Dekompositionsstufe von 3, dass wir jeweils vier Werte aus der ersten Ebene und zwei Werte aus der zweiten zusammenfassen.

Tong et al. fasst in seinem Algorithmus auch in der tiefsten Dekompositionsebene jeweils zwei Elemente zusammen. Außer einem Geschwindigkeitsgewinn sehe ich hierfür jedoch keine Rechtfertigung.

Durch den Auflösungsverlust mit jeder Dekompositionsebene geht eine Gefahr von Nachbarzeichenbeeinflussung mit einher. Stellen wir uns zwei dicht beeinanderliegende Kanten vor. Eine hiervon sei eine scharfe Dirac-Kante, die andere eine weichgezeichnete. Wir erinnern uns, dass die Dirac-Kante eine absteigende Intensität, die weichgezeichnete eine ansteigende Intensität in den einzelnen Dekompositionstufen hat. Bei jeder weiteren Dekomposition verringert sich die Auflösung, wodurch wir Gefahr laufen, beide Ausschläge in den Detailkoeffizienten zusammenzufassen. Angenommen die beiden Kanten fallen in der dritten Dekompositionsstufe zusammen. Dann können wir bis zur zweiten Stufe noch beide Kanten unterscheiden, in der dritten fallen aber beide zusammen. Da eine weichgezeichnete Kante einen deutlich höheren Ausschlag in der dritten Dekomposition als eine scharfe Kante aufweist, werden wir so die scharfe Kante nicht mehr erkennen können. Man könnte dies in Anlehnung an die Nachbarzeichenbeeinflussung aus der Übertragungstechnik (Intersymbolinterferenz) auch als Nachbar*kanten*beeinflussung bezeichnen.

## 6 Scharfzeichnung

Nachdem wir uns nun mit der Theorie der Kantenerkennung und Klassifizierung auseinandergesetzt haben und die Auswirkungen von einer Art der Unschärfen betrachtet haben, werden wir uns nun zunächst mit den Eigenarten digitaler Kameras beschäftigen.

Maßgeblich für die Schärfe und Qualität eines Bildes ist neben dem verwendeten Sensor vor allem auch das Objektiv. Es kommt hier stark auf die Güte des verwendeten Glases und die korrekte Berechnung der einzelnen Linsen an.

Generell kann man digitale Kameras in zwei Märkte unterteilen, den Markt für den Endanwender und den Markt für den ambitionierten Fotografen. Es gibt jedoch auch noch Nischenmärkte, auf die wir hier jedoch nicht eingehen möchten.

Kompakte Digitalkameras für den Verbraucher verwenden meist kleine Bildsensoren mit Flächen zwischen ca.  $15 \text{ mm}^2$  und  $60 \text{ mm}^2$ . Dementsprechend können auch die Objektive kleiner und günstiger gebaut werden. Durch die sehr hohe Pixeldichte auf den Sensoren wird jedoch das Rauschen verstärkt. Um dem Verbraucher trotzdem ein angenehmes Bild zu liefern, wird das aufgenommene Bild in der Regel stark nachbearbeitet. Dies umfasst eine starke Rauschreduktion, Kontrastverstärkung, Farbverstärkung und vor allem - für uns wichtig - Nachschärfung. Die entstehenden Bilder auf der Speicherkarte sind bereits soweit manipuliert, dass diese direkt in den Druck gegeben werden könnten.

Professionelle Digitalkameras - meist Spiegelreflexkameras mit Wechselobjektiven - verfolgen hier eine andere Philosophie. Während bei digitalen Spiegelreflexkameras (DSLR) nicht nur der Sensor hochwertiger und größer (meist ab  $320 \text{ mm}^2$  bis zu  $860 \text{ mm}^2$ ) ist, sind vor allem die Objektive höherwertiger, aber auch aufwendiger zu berechnen. Da eine DSLR dem Fotografen den größtmöglichen Spielraum für die Bearbeitung der aufgenommenen Bilder überlassen möchte, neigen DSLRs dazu, die entstehenden Bilder kaum bis gar nicht nachzubearbeiten. Diese Bilder benötigen in jedem Falle eine Nachbearbeitung.

Da Kompaktkameras ein festeingebautes Objektiv besitzen, können dessen Fehler direkt in der Kamera herausgerechnet werden. Bei DSLRs sind jedoch theoretisch unbegrenzt viele Objektive möglich, wodurch hier keine Bildfehler herausgerechnet werden können (bzw. erst nachträglich mit verschiedener Software). In dem Bild, welches wir erhalten, treten daher noch etliche Fehler auf. Dies können unter anderem chromatische Abberationen, Verzerrungen, Farbfehler und - wichtig - Unschärfe sein. Nicht alle Objektive sind, um einen geringeren Preis zu ermöglichen, bis zur Perfektion berechnet und verwenden das beste Glas, wodurch nicht alle Objektive die schärfsten Bilder liefern. Da auch die Kamera nicht schärft, ist es für uns sehr wichtig, die Bilder nachzuschärfen, bevor wir sie analysieren. Die von den Objektiven erzeugte Unschärfe ist nur gering und kann mittels Scharfzeichnung ausgeglichen werden, würde jedoch bei unserer Analyse schon als Unschärfe erkannt werden.

Um die Auswirkungen einer Scharfzeichnung auf unsere Kantenerkennung und -Klassifizierung betrachten zu können und um die Grundlagen von Scharfzeichnungsfiltern zu erlernen, widmen wir diesem Kapitel der Scharfzeichnung.

Es gibt verschiedene Verfahren und Filter, um Bilder scharfzuzeichnen. Im Folgenden werden wir die gängigsten Verfahren betrachten.

## 6.1 Schärfungsfilter

### 6.1.1 Schärfung durch Ableitung

Da der subjektive Schärfeeindruck von der Präsenz und Steilheit von Kanten abhängt, zielt eine Scharfzeichnung darauf ab, die Steilheit von weichgezeichneten Kanten wiederherzustellen.

Grundlegend basieren Scharfzeichnungsfilter darauf, dem Ursprungsbild ein berechnetes Kantenbild hinzuzufügen. Es kann aber auch negativ hinzugefügt werden, also abgezogen werden. Um zu diesem Kantenbild zu gelangen, gibt es verschiedene Varianten.

Wir betrachten hier die einfachste Art der Scharfzeichnung, bei der die Änderung der Steigung einer Kante skaliert vom Ursprungssignal abgezogen wird. Um dies zu erreichen, wird die zweite Ableitung des Signals oder Bildes generiert und vom Ursprungssignal abgezogen.

Wir betrachten dies theoretisch am Beispiel eines weichgezeichnetem Sprungs, dessen zweite Ableitung der Ableitung des Gauß-Kernes entspricht (Gl. 32). Das Verfahren ist grafisch in Abbildung 21 dargestellt.

$$\varepsilon_{blur}(x)'' = -\frac{1}{\sqrt{2 * \pi \sigma^3}} x \exp\left(-\frac{x^2}{2\sigma^2}\right) . \quad (32)$$

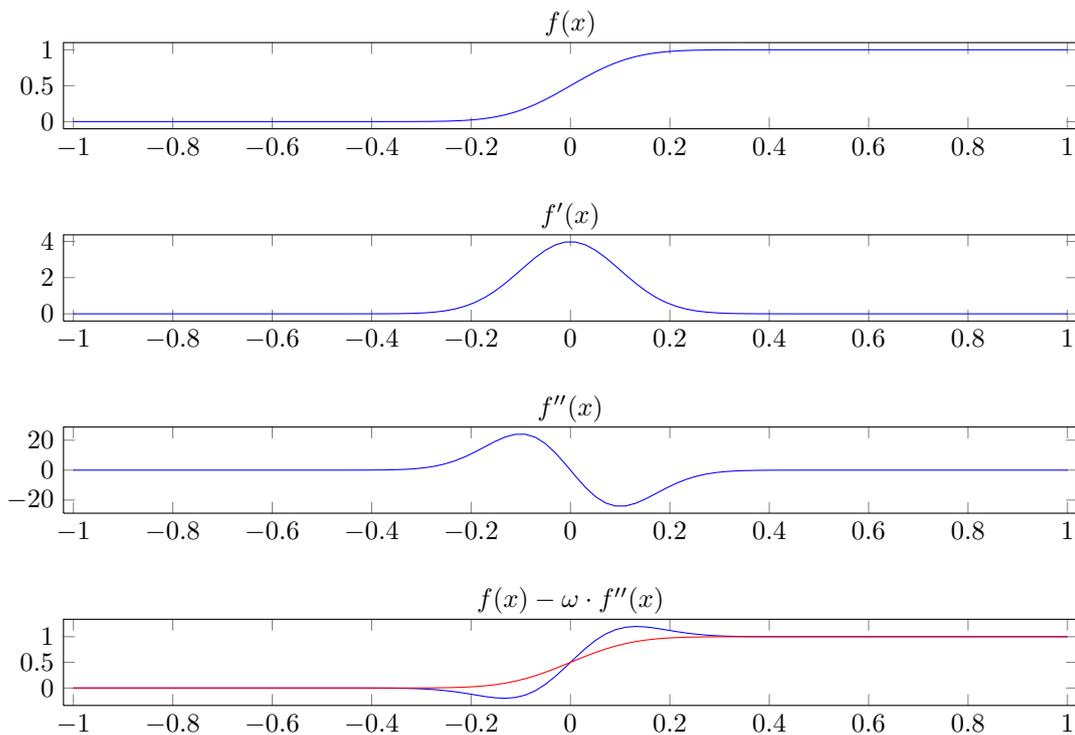


Abbildung 21: Scharfzeichnung durch Ableitung.

Hier wurde die zweite Ableitung um den Faktor  $\omega = 75$  reduziert und vom Ursprungssignal abgezogen. Wie wir sehen, zeigt unser geschärftes Signal eine höhere Steigung - was gut ist - aber auch Überschwinger - was nicht gut ist. Der Faktor  $\omega$ , mit dem die zweite Ableitung skaliert wird, stimuliert hierbei die Stärke der Scharfzeichnung. Je stärker wir scharfzeichnen, desto steiler wird die Kante, aber umso höhere Überschwinger treten auch auf.

Mathematisch können wir diese Scharfzeichnung des Signals  $u$  beschreiben als

$$u_{sharp}(x) = u(x) - \omega \cdot u(x)'' . \quad (33)$$

Da wir es jedoch mit diskreten Signalen zu tun haben, können wir hier eine keine zweite Ableitung mathematisch berechnen. Jedoch können wir die Ableitung und auch die zweite Ableitung mittels eines Filters realisieren. Ein Filter, welcher die zweite Ableitung annähert, ist ein Laplace-Filter.

Hierbei wird der Differentialquotient durch einen Differenzenquotienten angenähert (es wird ein „Steigungsdreieck“ angelegt). Häufig zu finden ist hierbei der Laplace-Operator in  $x$ - und  $y$ -Richtung zu

$$H_{L_x} = [1 \quad -2 \quad 1] \quad H_{L_y} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} . \quad (34)$$

Um den Laplace-Filter in zwei Dimensionen zu berechnen, reicht dann die einfache Addition beider Filtermatrizen, wodurch wir den zweidimensionalen Laplaceoperator erhalten zu

$$H_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} . \quad (35)$$

Es gibt jedoch noch weitere gebräuchliche, lineare Approximationen des Laplace Operators im zweidimensionalen [BB06, GWo8]. Diese reagieren auch auf diagonale Kanten.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (36)$$

Das geschärfte Signal/Bild  $I$  ergibt sich dann mittels der diskreten Faltung zu

$$I_{sharp} = I - \omega(I * H_L) . \quad (37)$$

Ein Vorteil dieser Scharfzeichnung ist, dass wir für ein gegebenes  $\omega$  einen direkten Scharfzeichnungsfilter berechnen können. Häufig wird hierbei  $\omega = 1$  gesetzt, wodurch sich ein oft erwähnter Scharfzeichnungsfilter

$$\begin{aligned} I_{sharp} &= I * \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - I * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\ &= I * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \\ &= I * H_{sharp} \end{aligned} \quad (38)$$

ergibt. Diese Art der Scharfzeichnung führt jedoch zu einer recht starken Schärfung des Bildes, wodurch schnell Schärfungsartefakte im Bild entstehen können und vor allem Rauschen verstärkt wird. Sie ist nicht sehr flexibel, da wir die Scharfzeichnung nur durch den Faktor  $\omega$  verstellen können.

### 6.1.2 Unschärfmaskierung

Eine andere Methode der Scharfzeichnung ist die Unschärfmaskierung. Diese stammt ursprünglich aus der analogen Bildbearbeitung und findet sich in vielen Programmen wie Adobe Photoshop oder GIMP wieder.

Bei der Unschärfmaskierung wird eine Kantenmaske erstellt, indem ein weichgezeichnetes Bild vom Ursprungsbild abgezogen wird. Diese entstehende Maske wird dann zum Ursprungsbild mit

einem Gewichtungsfaktor  $\alpha$  hinzuaddiert. Wir erhalten somit zwei Drehschrauben um die Scharfzeichnung zu beeinflussen. Dies wären der Stärke-Faktor  $\alpha$  und der Weichzeichnungs-Faktor  $\sigma$ . In Bildbearbeitungsprogrammen wird  $\sigma$  meist als „Radius“ in Pixeln bezeichnet.

In den meisten Fällen wird für die Weichzeichnung auch ein Gauß'scher Weichzeichner  $H$  verwendet, wie wir ihn in 5.1 kennengelernt haben. Wir können damit die Unschärfmaskierung formulieren als

$$\begin{aligned} I_{sharp} &= I + \alpha (I - (I * H)) \\ &= I + \alpha I - \alpha(I * H) \\ &= (1 + \alpha)I - \alpha(I * H) \end{aligned} \quad (39)$$

Die Unschärfmaskierung kann deutlich feiner eingestellt werden als der einfache Laplace-Filter. Sie verstärkt jedoch auch das Rauschen. Es ist wichtig, hier ein gutes Verhältnis zwischen  $\alpha$  und  $\sigma$  zu finden.

### 6.1.3 Laplace-Scharfzeichnung und Unschärfmaskierung

In vielen Büchern wird die Scharfzeichnung mittels eines Laplace-Filters und die Unschärfmaskierung getrennt behandelt. Jedoch kann gezeigt werden, dass die Laplace-Scharfzeichnung ein Spezialfall der Unschärfmaskierung ist. In bestimmten Fällen kann also auch die „bessere“ Unschärfmaskierung zu den gleichen „schlechten“ Ergebnissen der Laplace-Scharfzeichnung führen.

In Bildverarbeitungsprogrammen wie GIMP oder Adobe Photoshop sind meist mindestens zwei Scharfzeichnungsfunktionen eingebaut. Zum einen eine einfache Scharfzeichnung mittels eines Laplace-Filters und zum anderen eine Unschärfmaskierung.

Ein Beweis findet sich in [BB06, S. 136] und sei hier kurz vorgestellt. Er betrachtet hierbei den Laplace-Filter 35.

Der Laplace-Filter lässt sich zerlegen in zwei symmetrische Matrizen

$$H_L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} - 5 \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 5 (\tilde{H}_L - \delta) \quad , \quad (40)$$

wobei  $\delta$  den Dirac-Impuls als Matrix darstellt.

Mit Hilfe der Gleichung zur Scharfzeichnung mittels eines Laplace-Operators

$$I_{sharp} = I - \omega (I * H_L) \quad (41)$$

und dem eben zerlegtem Laplace-Filter folgt

$$\begin{aligned} I_{sharp} &= I - \omega (I * 5 (\tilde{H}_L - \delta)) \\ &= I - 5 \cdot \omega (I * \tilde{H}_L - I * \delta) \quad . \end{aligned} \quad (42)$$

Durch die Faltungseigenschaft des Dirac-Impulses folgt daraus

$$\begin{aligned} I_{sharp} &= I - 5 \cdot \omega (I * \tilde{H}_L - I) \\ &= I + 5\omega I - 5\omega (I * \tilde{H}_L) \\ &= (1 + 5\omega)I - 5\omega (I * \tilde{H}_L) \quad . \end{aligned} \quad (43)$$

Welche der Gleichung 39 der Unschärfmaskierung mit  $\alpha = 5\omega$  und einem Weichzeichnungsfiler  $H = \tilde{H}_L$  entspricht.

Dies lässt sich auch einfach für die anderen vorgestellten Laplace-Operatoren zeigen.

### 6.1.4 Laplacian of Gaussian

Ein selten verwendeter, aber sehr interessanter Filter ist der Laplacian of Gaussian.

Er funktioniert ähnlich wie die Laplace-Filterung, jedoch wird nicht die zweite Ableitung des Bildes subtrahiert, sondern die zweite Ableitung eines weichgezeichnetes Bildes. Dadurch wird das Rauschen unterdrückt und der Filter ist weniger anfällig für Rauschen.

Ein Vorteil hiervon ist, dass wir den Laplace-Operator direkt auf den Weichzeichnungsfiler anwenden und ihn so konkret berechnen können. Wir müssen uns nicht auf lineare Approximationen des Laplace-Operators verlassen.

Wir werden diesen Filter nun entwickeln.

Wie in 5.1 beschrieben, gilt für einen Gauß'schen Weichzeichnungsfiler

$$H_{Gauss}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad , \quad (44)$$

auf den der Laplace-Operator angewendet

$$\begin{aligned} H_{LoG}(x) &= \Delta \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \\ &= \frac{d^2}{dx^2} \left( \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \right) \\ &= \frac{d}{dx} \left( -\frac{x}{\sigma^2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \right) \\ &= \frac{d}{dx} \left( -\frac{1}{\sqrt{2\pi}\sigma^3} x e^{-\frac{x^2}{2\sigma^2}} \right) \\ &= -\frac{1}{\sqrt{2\pi}\sigma^3} \left( e^{-\frac{x^2}{2\sigma^2}} - \frac{x^2}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \right) \\ &= -\frac{1}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}} \left( 1 - \frac{x^2}{\sigma^2} \right) \end{aligned} \quad (45)$$

ergibt, welches nun unsere Filterfunktion darstellt. Ein Problem mit diesem Filter ist jedoch die Tatsache dar, dass er ein Kantenbild erstellt, welches weit aus dem Wertebereich des Ursprungsbildes herausragen kann. Es muss daher geschickt skaliert werden.

Auch die zweidimensionale Form des Filters kann berechnet werden, hierfür wenden wir den Laplace-Operator auf den zweidimensionalen Gauß-Kern aus 5.1 an. Wir müssen also die partielle Ableitung nach  $x$  und  $y$  des Gauß-Kernes bilden, wodurch wir den Laplacian of Gaussian (LoG) erhalten als

$$H_{LoG}(x, y) = -\frac{1}{\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \left( 1 - \frac{x^2+y^2}{2\sigma^2} \right) \quad . \quad (46)$$

Um diese Filterfunktion darzustellen, wurde die Summe der Filtermaske auf 1 gebracht, um die Intensität des Ausgangsbildes beizubehalten. Ein Laplacian of Gaussian für  $\sigma = 0.45$  ist in Abbildung 22 gegeben.

Wie man in 22 erkennt, wird er auf Grund seiner Form auch „Sombbrero-Filter“ oder „Mexican-Hat-Filter“ genannt.

Es muss hier ein geschickter Wert für  $\sigma$  gefunden werden und anschließend ein guter Skalierungsfaktor. Es stellte sich heraus, dass LoG mit einem  $\sigma = 0.45$  abgezogen vom ursprünglichem Bild (Subtraktion von einem Dirac-Impuls) und einer Normierung der Summe auf 1 einen guten Scharfzeichnungsfiler ergibt.

$$H_{LoG} = \begin{bmatrix} -0.0544 & -0.2399 & -0.0544 \\ -0.2399 & 2.1774 & -0.2399 \\ -0.0544 & -0.2399 & -0.0544 \end{bmatrix} \quad (47)$$

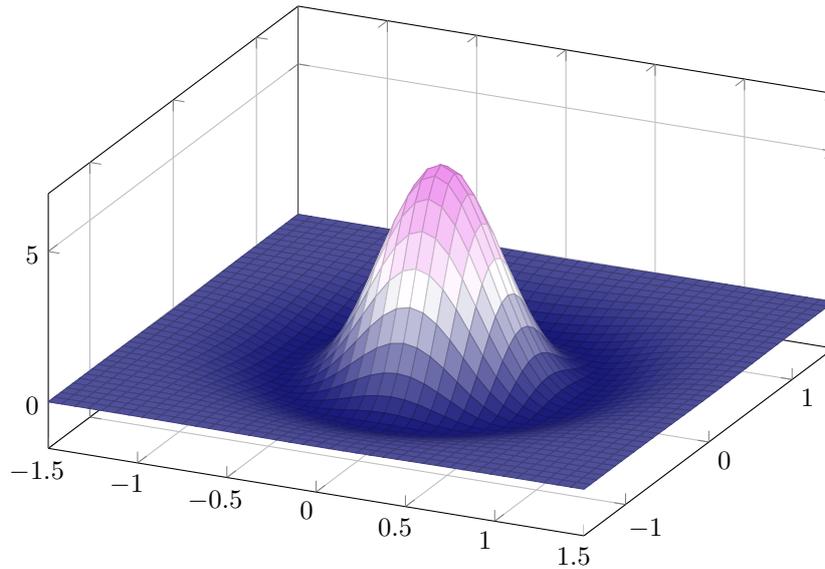


Abbildung 22: 2D Kern des Laplacian of Gaussian.

## 6.2 Untersuchung des Verhalten von Scharfzeichnung auf die Haar-Wavelet-Transformation

Nachdem wir uns nun die Grundlagen des Scharfzeichnens angeeignet haben, möchten wir nun die Auswirkung einer Scharfzeichnung, wie wir sie für Bilder aus einer DSLR benötigen, auf weichgezeichnete Kanten untersuchen.

Wir möchten hierfür unsere Evaluation von oben wiederholen, nachdem wir jedoch die Kanten weichgezeichnet haben, eine Scharfzeichnung anwenden. Wir haben im vorherigen Teil erörtert, dass theoretisch der Sombrero-Filter die besten Ergebnisse zeigen sollte. Dieser ist jedoch durch die benötigte Skalierung schwer zu implementieren, weshalb wir den häufig verwendeten Unschärfmaskierungsfiler verwenden.

Einen schönen Scharfzeichnungsfiler, welcher zu nicht allzu vielen Artefakten führt, liefert uns die Matlab-Funktion `special` mit den Parametern „unsharp“ und einem  $\alpha$ -Wert von 1.

Während für zweidimensionale Signale, also z. B. Bilder, solch eine  $3 \times 3$  Faltungsmatrix genügt, möchten wir für die Evaluation über unser eindimensionales Signal eine genauere Auflösung als einen  $3 \times 1$  Faltungsvektor erhalten. Die gängige Literatur geht jedoch leider nicht auf eindimensionale Scharfzeichnungsfiler ein und beschränkt sich bei zweidimensionalen Faltungsmatrizen auf  $3 \times 3$  Matrizen.

Um einen eindimensionalen Scharfzeichnungsfiler auf Basis der Unschärfmaskierung zu erstellen, bedienen wir uns der Gleichung 18 des Gauß-Kerns, subtrahieren dieses von einem Dirac-Impuls und addieren dies auf unser Ursprungsbild. Dies entspricht Gleichung 39 mit  $\alpha = 1$ .

Korrekt normiert sollte uns daher

$$H_{unsharp}(x) = \delta(x) + (\delta(x) - H_{Gauss}(x)) \quad (48)$$

einen guten Unschärfmaskierungsfiler beliebiger Auflösung liefern. Für  $\sigma = 2$  und für einen  $11 \times 1$  Filter erhalten wir damit

$$H_{unsharp}(x) = \begin{bmatrix} -0.0088 & -0.0271 & -0.0651 & -0.1216 & -0.1770 \\ 1.7994 & -0.1770 & -0.1216 & -0.0651 & -0.0271 & -0.0088 \end{bmatrix}, \quad (49)$$

welcher in der Tat eine gute Scharfzeichnung durchführt.

Wir führen also wie erwähnt die Evaluation wie oben beschrieben wieder durch. Es gelten die gleichen Simulationseinstellungen wie zuvor. Abbildung 23 zeigt die Simulationsergebnisse für die Dirac-Kanten und Abbildung 24 für Sprungkanten.

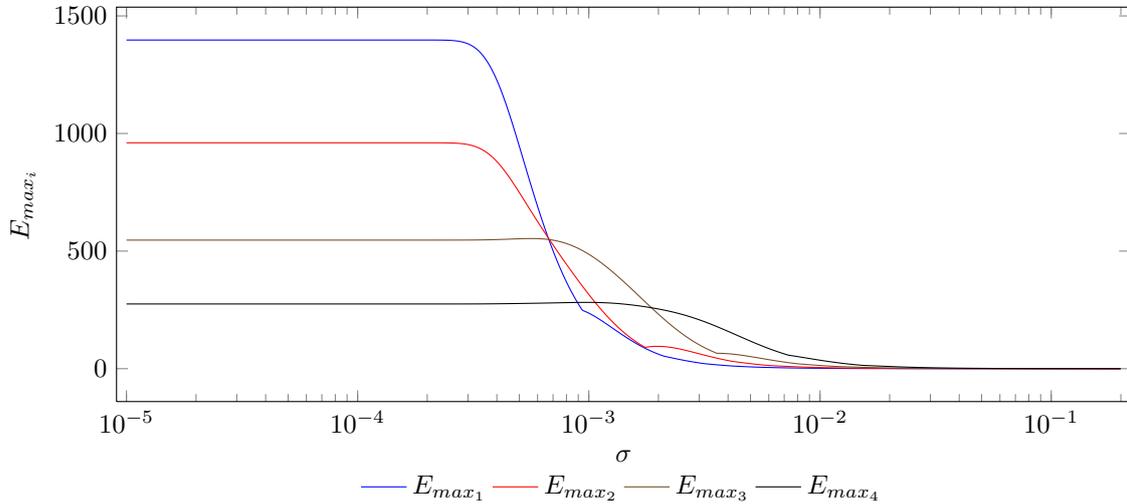


Abbildung 23: Evaluation des weichgezeichneten Dirac-Impulses mit Scharfzeichnung.

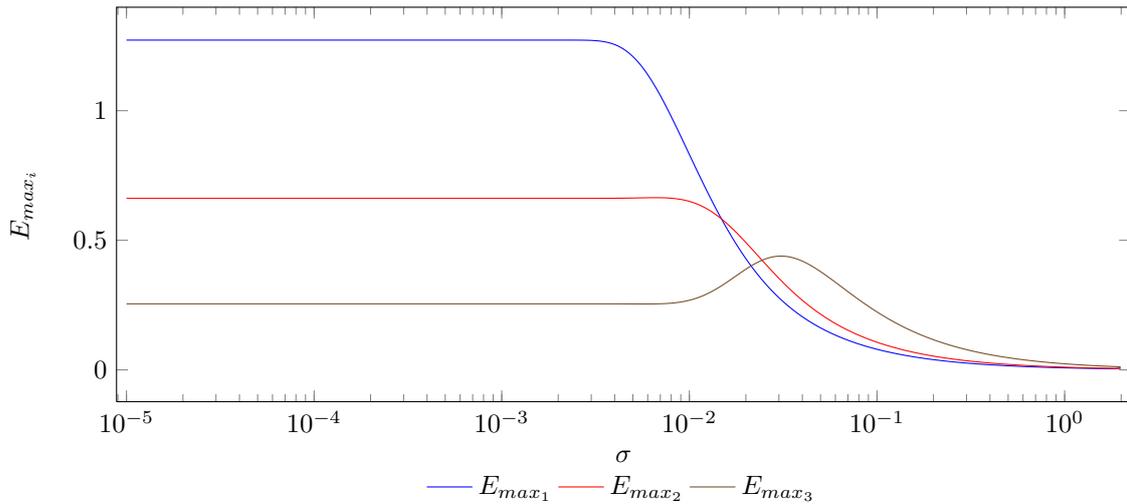


Abbildung 24: Evaluation des weichgezeichneten Sprung mit Scharfzeichnung.

Wir stellen fest, dass die Kurvenverläufe tendenziell gleich bleiben, jedoch schmiegt sich  $E_{max_2}$  stark an  $E_{max_1}$  an. Bei der Sprung-Evaluation wirkt sich die Scharfzeichnung auch dahingehend aus, dass die Kurven erst deutlich später anfangen zu fallen. Durch die Scharfzeichnung können wir also auch länger Sprungkanten als scharf klassifizieren.

Dadurch, dass sich  $E_{max_2}$  an  $E_{max_1}$  anschmiegt, müssen wir davon ausgehen, dass die Erkenntnis 3 zwar noch hinreichend ist, aber wir eine weitere notwendige Bedingung mit einführen müssen.

**Erkenntnis 7.** *Scharfzeichnung wirkt sich auf Sprungkanten stärker als auf Dirac-Kanten aus.*

**Erkenntnis 8.** *Zur Erkennung scharfer Kanten benötigen wir die notwendige Bedingung  $E_{max_2} > E_{max_3}$ .*

## 7 Bewegungunschärfe

Nachdem wir nun die Weichzeichnung und die leichte Korrektur der Weichzeichnung bearbeitet haben, betrachten wir nun die weitere Form der Unschärfe - die Bewegungsunschärfe.

Zunächst wollen wir uns wieder um eine Simulationsbasis kümmern. Im Eindimensionalen können wir nur eine Bewegungsunschärfe in eine Richtung simulieren. In zwei Dimensionen kommt hier ein Winkel hinzu, in der sich die Bewegungsunschärfe auswirken kann. In realen Bildern können sich jedoch auch mehrere Bewegungsunschärfen überlagern. Sie treten also zugleich aus unterschiedlichen Winkeln auf.

Um eine erste Vorstellung von Bewegungsunschärfe auf Sprung- und Dirac-Kanten zu erhalten und uns auch das Problem vor Augen zu führen, stellen wir uns eine weiße, verputzte Betonmauer vor blauem Himmel vor. Der Bildausschnitt sei so gewählt, dass die Mauer die Bildbreite ausfüllt. Wir nehmen hiervon ein Bild auf. Der Fokus sei so gesetzt, dass alles scharf ist.

Bei einem unverwackelten Bild tritt hier dann eine Sprungkante von Mauer zu Himmel auf, welche wir gut erkennen können. Da die Mauer verputzt ist, treten hier auch keine weiteren Kanten auf. Wir bewegen nun die Kamera während der Aufnahme ausschließlich in  $x$ -Richtung. Das führt eindeutig zu Bewegungsunschärfe. Aber wir können diese nicht erkennen, die Sprungkante von (verwackelter) Betonmauer zu (verwackeltem) Himmel ist immer noch vorhanden.

Ein etwas abstrakteres komplizierteres Beispiel ist unser Dirac-Impuls (Linie) aus 5.2, auf welchen Bewegungsunschärfe angewandt wird.

Betrachten wir aber zunächst den einfachen Fall einer Bewegungsunschärfe auf eine Sprungkante.



Abbildung 25: Sprungkante mit Bewegungsunschärfe

Die Auswirkung der Bewegungsunschärfe auf eine Sprungkante stellt kein Problem dar, solange die Unschärfe in die selbe Richtung erfolgt, in der die Kante verläuft. Wir erhalten dann ein ähnliches Bild wie bei der Gauß'schen Unschärfe. Jedoch ist hier der Anstieg linear und folgt nicht dem Gauß-Kern ( $e^{-x^2}$ ).

Wie schon erwähnt, ist eine verwischte Dirac-Kante jedoch ein komplizierterer Fall und stellt ein Problem dar.



Abbildung 26: Dirac-Impuls mit Bewegungsunschärfe

Wie wir sehen, wurde hier der Dirac-Impuls in nur exakt einer Richtung verwischt. Dadurch nimmt zwar die Intensität der Kante ab, aber er stellt nun theoretisch zwei scharfe Sprungkanten (Anfang & Ende) dar.

Wir haben also immer dann ein Problem, wenn sich Bewegungsunschärfe in nur genau einer Richtung auswirkt und es sich um Dirac-Kanten handelt, oder die Bewegungsunschärfe orthogonal zur Sprungkante verläuft.

Der Vorteil ist jedoch, dass die häufigste Ursache für Bewegungsunschärfe Verwackelung ist; z. B. durch ein Zittern der Hand. Diese Verwackelung ist dann eine Bewegungsunschärfe in mehrere Richtungen und führt nicht mehr zur Verwechslung mit einer Sprungkante. Die Steigung unserer Kanten folgt dann zwar nicht mehr dem Gauß-Kern (es sei denn die Hand zittert genauso), sie wird jedoch endlich, also an- oder absteigend. Wir gehen davon aus, dass unsere Ergebnisse aus der Gauß'schen Unschärfe auch auf andere Formen des Anstiegs erweiterbar sind.

**Erkenntnis 9.** *Bewegungsunschärfe in nur einer Richtung kann zu falsch positiv erkannten Kanten führen.*

**Erkenntnis 10.** *Bewegungsunschärfe einer Sprungkante zeigt ähnliche Eigenschaften wie die Weichzeichnung dieser.*

## 8 Algorithmus zur Kantenklassifizierung in Signalen

Aus unseren Erkenntnissen können wir nun einen Algorithmus für die Klassifizierung von Kanten entwickeln. Wir wenden zunächst den von uns entwickelten Scharfzeichnungsfiler an, um eine leichte Unschärfe zu korrigieren. Anschließend berechnen wir die HWT bis zur dritten Ebene.

Hiervon interessieren uns nur die Detail-Koeffizienten. In Anlehnung an Tong et. al bezeichnen wir die einzelnen Detail-Koeffizienten als  $E_{map_i}$ . Wie in 5.4 beschrieben, müssen wir die einzelnen Detail-Koeffizienten punktweise miteinander vergleichen können, weshalb wir diese zusammenfassen müssen. Wir berechnen daher für jede Ebene  $i$  das Maximum von jeweils  $2^{n-i}$  Elementen. Da wir drei Dekompositionen durchführen, fassen wir also jeweils vier Elemente in der ersten Ebene und zwei in der zweiten zusammen. Das Ergebnis dieser Reduzierung bezeichnen wir als  $E_{max_i}$ .

Wie wir in Erkenntnis 2 und 4 festgestellt haben, reagieren die einzelnen Dekompositionsstufen unterschiedlich auf scharfe und weichgezeichnete Kanten. Die erste Dekomposition reagiert stark auf scharfe Kanten, die  $n$ . Dekomposition jedoch sehr stark auf weichgezeichnete Kanten. Da wir sowohl scharfe als auch weiche Kanten erkennen möchten, betrachten wir jeden Punkt  $x$  als einen Kandidaten für eine Kante, wenn er in einer der Dekompositionsstufen einen Ausschlag größer als eine festgelegte Grenze  $t$  hat.

Mit Hilfe der Erkenntnisse 3 und 8 können wir anhand der unterschiedlichen Intensitäten Aussagen über die Art der Kante treffen. In Erkenntnis 6 haben wir festgestellt, dass die gleichen Beobachtungen sowohl für Dirac- als auch Sprungkanten gelten. Hat die Kante ihren höchsten Ausschlag in  $E_{max_1}$  und ist abnehmend nach  $E_{max_2}$  und  $E_{max_3}$ , so handelt es sich um eine Sprung- oder Dirac-Kante, also eine als scharf zu betrachtende Kante. Liegt jedoch  $E_{max_2}$  über  $E_{max_1}$ , so handelt es sich mit großer Wahrscheinlichkeit um einen weichgezeichneten Dirac-Impuls oder Sprung, also eine unscharfe Kante.

Außerdem haben wir in Erkenntnis 10 festgestellt, dass die Aussagen über weichgezeichnete Kanten auch für verwischte Kanten gelten sollten (Bewegungsunschärfe).

In Kurzform:

1. Scharfzeichnungsfiler anwenden.
2. Haar-Wavelet Dekomposition bis zur dritten Ebene durchführen.  $\Rightarrow E_{map_i}$ .
3. Detail-Koeffizienten fenstern.  $\Rightarrow E_{max_i}$ .
4. Jeder Punkt  $x$ , für den  $E_{map_i}(x) > t$  gilt, ist eine Kante.
5. Jede Kante  $x$  ist scharf, wenn  $E_{max_1} > E_{max_2} > E_{max_3}$ .
6. Jede Kante  $x$  ist weichgezeichnet, wenn  $E_{max_1} < E_{max_2}$ .

### Anmerkung zu Problemen mit Sprungkanten

Es gibt einen Wermutstropfen bei der HWT in Bezug auf die Erkennung von sprungförmigen Kanten.

Es handelt sich bei der HWT nicht um eine translations-invariante Transformation. Das heißt, geben wir ein verschobenes Signal ein, so kommt nur in manchen Fällen die selbe Transformation dabei heraus.

Wir können uns dies und das Problem mit Sprungkanten an einem kurzen Beispiel klarmachen. Gegeben sei ein diskreter Sprung zu  $A = [0\ 0\ 0\ 1\ 1\ 1]$ . Hierfür berechnen wir die HWT (nicht normalisiert):

$$A = [0\ 0.5\ 1] \quad (50)$$

$$D = [0\ 0.5\ 0] \quad (51)$$

Wir sehen, dass es hier zu keinem Problem kam. Wir verschieben nun den Sprung um einen Abtastwert  $A = [0\ 0\ 0\ 1\ 1]$  und berechnen wieder die HWT:

$$A = [0\ 0\ 1] \quad (52)$$

$$D = [0\ 0\ 0] \quad (53)$$

wobei wir feststellen, dass der Detail-Koeffizient diesmal nicht auf die Kante reagiert hat. Dies liegt daran, dass wir immer zwei Werte zusammenfassen. Liegt nun die Kante so, dass wir den Wechsel in keinem Fenster haben, so fällt er nicht auf.

Dass wir das Problem bei Dirac-Kanten nicht haben, kann man sich insofern vorstellen, als dass wir dort zwei Chancen haben, den Dirac-Impuls zu erkennen und ihn somit immer sehen werden.

Wir werden also nicht alle Sprungkanten erkennen können. Das Gute ist jedoch, dass es sich selten um perfekte Sprungkanten handelt - welche wir erkennen können - und in echten Bildern Sprungkante auch meist ein Ende haben, welches wir mit Glück erkennen können. Auch haben wir bei zweidimensionalen Signalen noch weitere Informationen zur Verfügung. Hier haben wir zusätzlich noch Informationen in vertikaler und diagonaler Richtung, welche uns hier zu Hilfe kommen werden.

Wird nur eine Dekomposition durchgeführt, so ist die HWT translations-invariant, für jeweils um zwei gegeneinander verschobene Signale. Werden mehrere Dekompositionen durchgeführt, so pflanzt sich die Verschiebung fort, wodurch die HWT bis zur  $i$ . Dekomposition nur für um  $2^i$  verschobene Signale translations-invariant ist. Für zweidimensionale Signale quadriert sich dieser Fehler zudem. Um theoretisch alle HWT in unserem Algorithmus durchzuführen, müssten wir (wir führen drei Dekompositionen aus)  $(2^3)^2$  Untersuchungen durchführen.

Ich habe anhand verschiedener Bilder die unterschiedliche Erkennungsrate durch Verschiebungen untersucht. Hierbei stellte sich heraus, dass der Algorithmus in der Tat jeweils unterschiedliche Punkte erkennt, die Gesamttendenz - ob ein Bild scharf oder unscharf ist - stimmte jedoch immer.

## 9 Zweidimensionale Haar-Wavelet-Transformation

Wir werden nun die Erkenntnisse aus Teil eins in das Zweidimensionale übertragen. Hierfür betrachten wir zunächst, wie die HWT für zweidimensionale Signale funktioniert und wie sie reagiert. Anschließend werden wir den Algorithmus in das Zweidimensionale übertragen und uns der praktischen Anwendung auf Bilder widmen.

Zur Erzeugung der zweidimensionalen HWT werden, wie bei vielen anderen Transformation, zunächst die Zeilen und anschließend die Spalten - oder vice versa - transformiert. Hierbei entstehen insgesamt drei Detailkoeffizienten bzw. hochpassgefilterte Signale und ein gemittelt bzw. tiefpassgefiltertes Signal.

Wie diese entstehen und welche Bedeutung sie haben, wollen wir im Folgenden betrachten.

Transformiert man zunächst die Zeilen, so entsteht zunächst auf der linken Seite das gemittelte bzw. tiefpassgefilterte Bild, jedoch verzerrt. Auf der rechten Seite erhalten wir den Detailkoeffizienten der Zeilen bzw. die Hochpass-Informationen der Zeilen, also Informationen über die horizontalen Kanten.

Anschließend werden die Spalten dieses neu entstanden Bildes transformiert. Wir erhalten dann in der linken, oberen Ecke das gemittelte, gemittelte Bild, bzw. doppelt tiefpassgefilterte Bild. Dieses ist nun wieder im korrekten Seitenverhältnis und stellt unser gemittelt Bild dar, auf welches die nächste Dekompositionsstufe angewandt werden kann. Im rechten oberen Bereich entsteht der gemittelte Detailkoeffizient der Zeilen, also die Information über die horizontalen Kanten in der richtigen Auflösung. In Anlehnung an die angewandten Filter der digitalen Signalverarbeitung wird dieser Bereich als  $HL_i$  bezeichnet - Highpass Lowpass der  $i$ . Dekomposition.

Im linken, unteren Bereich erhalten wir den Detailkoeffizienten des gemittelten Bildes, also die Kanteninformation der Spalten. Dies ist die Hochpassfilterung der Tiefpassfilterung, also  $LH_i$ .

Im rechten, unteren Bereich erhalten wir das doppelt hochpassgefilterte Bild. Also den Detailkoeffizienten des Detailkoeffizienten der Zeilen  $HH_i$ . Dies sind die Informationen über diagonale Kanten.

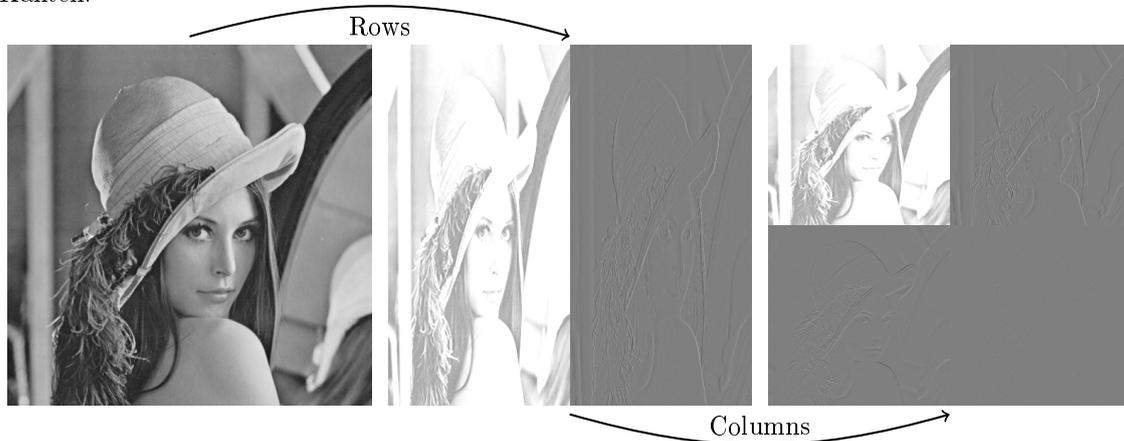


Abbildung 27: Zweidimensionale Haar-Wavelet-Transformation.

**Anmerkung 5.** Um auch negative Werte im Detailkoeffizienten darstellen zu können, wurde dieser Darstellung ein Offset hinzugefügt. Der Wertebereich wurde jeweils auf das theoretische Maximum skaliert.

Entsprechend ihrer Bedeutung werden wir die Bereiche benennen als  $A_i$ verage,  $V_i$ ertikal,  $H_i$ orizontal und  $D_i$ agonal.

Das gemittelte Bild  $A_i$  kann nun als neues Eingangssignal betrachtet werden, um die nächst-tiefere Dekomposition zu erhalten.

## 10 Algorithmus zur Erkennung scharfer Kanten eines Bildes

Wir werden nun den Algorithmus aus 8 in das Zweidimensionale übertragen.

Hierfür müssen wir jedoch ein paar Vorüberlegungen treffen.

Zunächst haben wir nun 3 Detailkoeffizienten und nicht mehr nur einen. Da uns jedoch Kanten in alle Richtungen interessieren, müssen wir alle Detailkoeffizienten miteinbeziehen. Wir bilden hierfür das quadratische Mittel der drei Koeffizienten  $\sqrt{H^2 + V^2 + D^2}$ , wodurch wir - im Gegensatz zur einfachen Mittelwertbildung - starke Ausschläge (also Kanten) auch stärker mit einfließen lassen. Wir verzichten hierbei auf die Normierung, da wir die Werte untereinander vergleichen.

Außerdem müssen wir die Fensterung der Koeffizienten anpassen. Haben wir vorher zwei Werte zusammengefasst, müssen wir nun 2x2 Werte zusammenfassen, dementsprechend auch bei vier Werten.

Eine wichtige Überlegung ist es, wie wir mit Farbbildern umgehen. Wir haben bisher nur monochromatische Bilder betrachtet. Die Bilder, die wir untersuchen möchten, liegen jedoch im RGB-Format vor. D. h. für jede Farbe Rot, Grün, Blau haben wir ein eigenes Bild vorliegen, welches für sich monochromatisch ist. Eine erste Idee ist es, das Farbbild einfach in ein monochromatisches umzuwandeln. Dies geschieht durch Mittelung der drei Kanäle mit verschiedenen Gewichtungen. Je nachdem welcher Kanal mehr gewichtet wird, entstehen hier andere Effekte. Es ist z. B. üblich den Rotkanal stärker zu gewichten, wodurch sich ein dunklerer Himmel ergibt und Hautpartien markanter hervortreten. Dies hat jedoch einen entscheidenden Nachteil: es verschwinden Kanten. Man betrachte hierfür ein einfaches Bild, welches in vier Quadranten unterteilt ist; Rot, Grün, Blau und eine weitere Farbe. Es ist klar zu erkennen, dass zwischen Rot, Grün und Blau dann eine Kante vorliegen würde. Bei der Umwandlung würden diese jedoch verschwinden, da ein reines Rot/Grün/Blau bei einfacher Mittelung jeweils zu  $\frac{255}{3}$  gemittelt würde. Für andere Mittelungen gibt es entsprechende Extrembeispiele.

Wir entscheiden uns daher - auf Kosten der Performance - jeden Farbkanal getrennt für sich zu analysieren. Die Ergebnisse werden dann anschließend zusammengefasst. Es ist hierbei jedoch darauf zu achten, dass Kanten nicht doppelt gezählt werden, wenn sie an der gleichen Stelle in verschiedenen Kanälen vorkommen.

In Kurzfassung heißt dies für den Algorithmus, um scharfe Kanten in einem zweidimensionalen Farbbild zu erkennen:

1. Scharfzeichnungsfilter anwenden.
2. Haar-Wavelet Dekomposition bis zur dritten Ebene durchführen.
3. Vertikale, horizontale und diagonale Informationen zusammenfassen.  
 $\Rightarrow E_{map_i} = \sqrt{V_i^2 + H_i^2 + D_i^2}$ .
4. Detail-Koeffizienten fenstern.  $\Rightarrow E_{max_i}$ .
5. Jeder Punkt  $x$ , für den  $E_{map_i}(x, y) > t$  gilt, ist eine Kante.
6. Jede Kante  $x$  ist scharf, wenn  $E_{max_1} > E_{max_2} > E_{max_3}$ .
7. Jede Kante  $x$  ist weichgezeichnet, wenn  $E_{max_1} < E_{max_2}$ .

## 11 Anwendung des Algorithmus

Nachdem wir uns nun lange mit der Theorie des Algorithmus und seiner Entwicklung beschäftigt haben, wollen wir ihn nun testen und ausprobieren.

Zunächst wollen wir die Kantenerkennung mittels der HWT testen. Wir wählen hierfür ein Bild, bei dem die Kanten sehr leicht zu erkennen sind, da es sich um einen Schriftzug handelt.



Abbildung 28: Demonstrationsbild zur Kantenerkennung.

Da der Schriftzug aber nicht komplett im Fokusbereich liegt, sondern nach links und rechts unschärfer wird, können wir gleichzeitig testen, ob der Algorithmus auch unscharfe Kanten erkennt. Wir haben zu Beginn festgestellt, dass die HWT noch sehr lange auf Kanten reagiert, auch wenn sie unscharf sind.

Wie wir in Abbildung 28b sehen, wurden die Kanten unseres Schriftzuges sehr gut erkannt. Auch die unscharfen Buchstaben wurden gut erkannt.

Da das Bild nur einen sehr kleinen Schärfebereich ringsum das „orm“ von „Performante“ hat, wählen wir einen kleineren Ausschnitt hieraus.

Wir testen nun ob der Algorithmus auch nur das scharfe „orm“ erkennt. Erkennt der Algorithmus eine scharfe Kante, so wird die Rotkomponente dieses Pixels auf 100% angehoben. Wir können so gut erkennen, wo der Algorithmus scharfe Kanten gefunden hat, bzw. der Algorithmus visualisiert uns scharfe Bereiche im Bild.



Abbildung 29: Visualisierte Dirac- und Sprungkanten im Bild.

Wir sehen sehr gut, dass nur Pixel in der Schärfebene erkannt wurden. Dass dies kein Einzelfall ist, betrachten wir an einem weiteren Beispiel.



Abbildung 30: Demonstrationsbild zur Schärfekerennung.

Die Abbildung 30a zeigt einen „AMG“-Schriftzug auf einem Motorblock. Da das Bild mit einer kleinen Blendenöffnung aufgenommen wurde, verläuft die Schärfebene schräg durch das M. Die feine Struktur des Aluminiums sorgt zudem dafür, dass viele Kanten erkannt werden können. Abbildung 30b zeigt den vom Algorithmus visualisierten Schärfebereich.

Wie zuvor funktioniert der Algorithmus auch hier erstaunlich gut, was auch daran liegt, dass hier besonders viele Kanten erkannt werden konnten. Abbildung 31 zeigt noch weitere Beispiele, wie gut unser Algorithmus den Schärfebereich visualisieren kann.



Abbildung 31: Beispiele zur Visualisierung des Schärfebereichs.

Wir halten fest, dass der Algorithmus bestens scharfe Pixel erkennt und dadurch sehr gut den Schärfebereich eines Bildes hervorheben kann. Wir wollen jedoch nicht nur visualisieren, sondern erkennen bzw. sortieren.

Zuvor müssen wir jedoch noch ein Problem lösen. Denn nicht jedes Bild, in dem der Algorithmus eine Schärfenebene erkennt, muss automatisch scharf sein.

Abbildung 32a zeigt solch ein Problem. Das Hauptmotiv dieses Bild ist klar der Wagen in der Mitte. Er ist jedoch sehr unscharf und jeder Mensch würde dieses Bild als unscharf klassifizieren. Was man jedoch nur schwer sieht, ist, dass der Asphalt im Vordergrund im Fokusbereich liegt. Unser Algorithmus visualisiert in Abbildung 32b diesen Bereich jedoch hervorragend.



Abbildung 32: Erkennung ungewollter, scharfer Kanten.

Jedoch ist dies kein Fehler des Algorithmus, denn er arbeitet korrekt und erkennt den scharfen Bereich des Bildes. Wir haben aber die Ambition auch solche Bilder aussortieren zu können. Wir müssen also dem Algorithmus beibringen zu wissen, *wo* das Bild scharf sein sollte.

## Partitionierung des Bildes

Um das gerade aufgedeckte Problem zu lösen, benötigen wir weitere Informationen. Wir benötigen die Information, wo das Bild scharf sein sollte.

Glücklicherweise speichern heutige Kameras viele Informationen in den sogenannten EXIF oder IPTC Headern ein. Canon und Nikon speichern hier unter anderem auch ab, welche Fokuspunkte bei der Aufnahme aktiv waren.

Um diese Information auswerten zu können, müssen wir jedoch einen kurzen Exkurs in den Umgang mit Autofokusfeldern von Spiegelreflexkameras unternehmen.

Die Anordnung und Anzahl von Autofokuspunkten unterscheidet sich von Modell zu Modell. Üblich sind in der Einstiegs- und Mittelklasse jedoch zwischen Sieben und Neun Autofokuspunkten, welche im Kreuz oder im Kreis um den mittleren Autofokuspunkt angeordnet sind. Da jedoch nicht immer dort ein Autofokuspunkt vorliegt, wo der Fotograf gerne den Fokus im aktuellen Bildausschnitt hätte, ist es üblich, die Kamera leicht zum nächsten Autofokuspunkt zu verschwenken, zu fokussieren und zurückzuschwenken. Hierdurch verschiebt sich zwar leicht der Fokusbereich, was aber durch die Tiefenschärfe der gewählten Blende im Alltag ausgeglichen wird.

Um dies miteinzukalkulieren wäre eine Möglichkeit, Ansammlungen von scharfen Kanten im Bild zu erkennen und diese mit den aktiven Autofokuspunkten in Verbindung zu stellen. Liegen diese in einem plausiblen Abstand zueinander, so wäre das Bild scharf. Dies wäre jedoch mit Kanonen auf Spatzen geschossen.

Wir verwenden daher eine einfachere Variante und wählen einen Bildausschnitt ringsum den aktivierten Autofokuspunkt, um diesen einzeln zu analysieren.

## 12 Algorithmus zur Klassifizierung von scharfen und unscharfen Bildern

Wir verfeinern den von uns entworfenen Algorithmus nun dadurch, dass nur gezielte Bereiche analysiert werden. Dies führt auch dazu, dass der Algorithmus schneller arbeitet.

Um den Bildausschnitt zu wählen, werden die aktiven Autofokuspunkte ausgelesen. Hierbei kann es vorkommen, dass auch mehrere Autofokuspunkte aktiviert sind. Wir betrachten dann jeden Autofokuspunkt für sich alleine.

Um die Toleranz des Verschwenkens der Kamera miteinzubeziehen, wählen wir einen Bereich von 33% um den Autofokuspunkt. Dies reicht aus, um auch starkes Verschwenken miteinzubeziehen und analysiert einen nicht zu kleinen Teil des Bildes.

Sind alle Autofokuspunkte aktiviert, so heißt das, dass die Kamera in diesem Bereich alles als scharf erkannt hat. Wir wählen dann einen Ausschnitt ringsum den Mittelpunkt von der Hälfte des Bildes, welcher alle Autofokuspunkte miteinbezieht. Wird mit allen Autofokuspunkten aktiv fotografiert, so handelt es sich meist um frontale Aufnahmen eines Objektes in der Mitte des Bildes.

Da, wie erwähnt, die Autofokuspunkte bei vielen Modellen und Herstellern unterschiedlich sind, müssen wir nun noch herausfinden, wo sich welcher Autofokuspunkt im Bild befindet. Zumindest Canon speichert dies glücklicherweise auch in den EXIF-Daten ab.

Leider werden diese Informationen nicht als normales EXIF-Tag abgespeichert, sondern verstecken sich in den herstellereigenen „Maker-Notes“ im EXIF-Header. Phil Harvey betrieb hier viel reverse engineering und programmierte hieraus das „ExifTool“ (<http://www.sno.phy.queensu.ca/~phil/exiftool/>). Dieses kommandozeilenbasierte, plattformunabhängige Tool kann die von uns benötigten Informationen elegant auslesen.

Es fehlt nun nur noch der finale Schritt, ein Kriterium zu finden, an dem wir festmachen können, ob ein Bild scharf oder unscharf ist. Wir wissen, dass der Algorithmus sehr gut auf scharfe Kanten reagiert. Zudem wissen wir, dass er auch unscharfe Kanten erkennen kann. Zuletzt haben wir nun nur noch einen Bildausschnitt, welcher komplett scharf sein sollte. Wir wählen daher - in Analogie zu Tong et al. - den Koeffizienten aus Anzahl von scharfen Kanten  $N_{da}$  zu Anzahl aller Kanten  $N_{edge}$  als Kriterium. Dieser sollte möglichst hoch sein. Da auch unscharfe Kanten erkannt werden, erhalten wir dadurch ein gutes Verhältnis von unscharfen zu scharfen Bereichen im Bild. Wir nennen diesen Koeffizienten *Sharpness*, als Faktor für die „Scharfheit“ des Bildes.

Tong et al. führte noch einen weiteren Koeffizienten ein. Dieser beschreibt das Ausmaß der Weichzeichnung. Wenn eine Kante als unscharf erkannt wird ( $E_{max_1} < E_{max_2}$ ), so unterscheiden wir diese in zwei Kategorien. Wird eine Kante als unscharf erkannt und gilt  $E_{max_1} < t$ , dann heißt dies, dass der Punkt schon soviel an Schärfe verloren hat, dass er im ersten Detailkoeffizienten nicht mehr als Kante erkannt wird. Durch die Mittelungen erhält dieser Punkt allmählich seine Schärfe zurück und wird in darauffolgenden Dekompositionsstufen erkannt. Wir zählen diese unscharfen Kanten als  $N_{brg}$ , wohingegen wir unscharfe Kanten, welche in der ersten Dekomposition noch als Kante erkannt werden als  $N_{rg}$  zählen. Das Verhältnis von beiden ergibt den Faktor *BlurExtent* und steigt mit wachsender Weichzeichnung.

In Kurzform heißt dies:

1. Lese die aktiven Autofokuspunkte aus.
2. Wähle einen geeigneten Bildausschnitt um jeden Punkt.
3. Wende Scharfzeichnungsfilter an.
4. Führe Haar-Wavelet Dekomposition bis zur dritten Ebene durch.  $\Rightarrow E_{map_i}$ .
5. Detail-Koeffizienten fenstern.  $\Rightarrow E_{max_i}$ .
6. Jeder Punkt  $x$ , für den  $E_{map_i}(x, y) > t$  gilt, ist eine Kante.  $N_{edge} ++$ .
7. Jede Kante  $x$  ist scharf, wenn  $E_{max_1} > E_{max_2} > E_{max_3}$ .  $N_{da} ++$ .
8. Jede Kante  $x$  ist weichgezeichnet, wenn  $E_{max_1} < E_{max_2}$ .  $N_{rg} ++$  und  $N_{brg} ++$ .

Ausführlich alles zusammengefasst:

1. Lese Mittels „exifTool“ die aktiven Autofokuspunkte und die Koordinaten der Autofokuspunkte aus.
2. Sind alle Autofokuspunkte gewählt, so analysiere einen Bereich von 50% des Bildes ringsum den Mittelpunkt.
3. Sind mehrere Autofokuspunkte gewählt, so führe das Folgende für jeden Bereich durch, wobei ein Bereich von 33% um den jeweiligen Punkt gewählt wird.
4. Zeichne das Bild mittels einer Filtermaske scharf.
5. Führe die Haar-Wavelet-Dekomposition bis zur dritten Ebene durch. Bilde für jede Ebene das quadratische Mittel der Vertikalen, Horizontalen und Diagonalen Koeffizienten. Dies sei  $E_{map_i}(x, y)$ .
6. Bringe die Ebenen durch Fensterung auf die gleiche Auflösung. Fasse in der zweiten Ebene jeweils 2x2 Pixel zusammen, in der ersten Ebene jeweils 4x4 Pixel. Dies sei  $E_{max_i}(x, y)$ .
7. Jeder Punkt  $(x, y)$ , für den  $E_{max_i}(x, y) > t$  gilt, ist eine Kante. Zähle diese in  $N_{edge}$ .
8. Jeder Punkt  $(x, y)$  ist eine scharfe Kante, wenn  $E_{max_1}(x, y) > E_{max_2}(x, y) > E_{max_3}(x, y)$ . Zähle diese in  $N_{da}$ .
9. Jeder Punkt  $(x, y)$  ist eine weichgezeichnete Kante, wenn  $E_{max_1}(x, y) < E_{max_2}(x, y)$ . Gilt hierbei  $E_{max_1}(x, y) > t$ , so zähle als  $N_{rg}$ , sonst als  $N_{brg}$ .
10. Ist  $Sharpness = \frac{N_{da}}{N_{edge}} > t_{sharp}$  und  $BlurExtent = \frac{N_{brg}}{N_{rg}} > t_{blur}$ , so handelt es sich um ein scharfes Bild.

**Anmerkung 6.** Die Nomenklatur der Faktoren wurde zur Vergleichbarkeit von Tong et al. übernommen. Die Bedeutung stimmt jedoch nur begrenzt überein.  $N_{da}$  bedeutet „Dirac or A-Step“ - also Dirac- oder Sprungkante. Tong et al. geht jedoch von einer linearen Steigung bei unscharfen Kanten aus (im Gegensatz zu unserer - genaueren - Gauß-Kurve). Hierbei entsteht aus einem Dirac-Impuls ein Dach (**R**oof) und der Sprung erhält eine endliche Steigung (ansteigend - gradual) und wird zum „G-Step“. Dies wird als  $N_{rg}$  gezählt.

## 13 Evaluation des Algorithmus

### 13.1 Simulation

Um den finalen Algorithmus zu testen und Größenordnungen für die verschiedenen Werte zu finden, wurde ein Satz von 34 Bildern, welche zweifelsohne scharf sind, gewählt. Auf diese wurde dann ein Weichzeichnungsfiler angewandt und anschließend der Algorithmus durchgeführt.

Für die Kantenerkennung wurde in Anlehnung an Tong et al. ein Wert von 35 gewählt.

Für den Weichzeichnungsfiler wurde ein „Circular averaging filter (pillbox)“ von Matlab gewählt, dessen Radius  $r$  von 0.5 bis 6 exponentiell gesteigert wurde. Hierbei ist anzumerken, dass dieser für einen Radius  $< 0.5$  den Filter 1 liefert, also keine Auswirkungen hat.

Für jeden Radius wurde der Mittelwert und die Standardabweichung von *Sharpness* und *BlurExtent* berechnet. Folgende Grafik zeigt den  $2\sigma$ -Bereich von *Sharpness* zu jedem berechneten Weichzeichnungsradius an. Wie erwartet, fällt der Sharpness-Koeffizient mit steigender

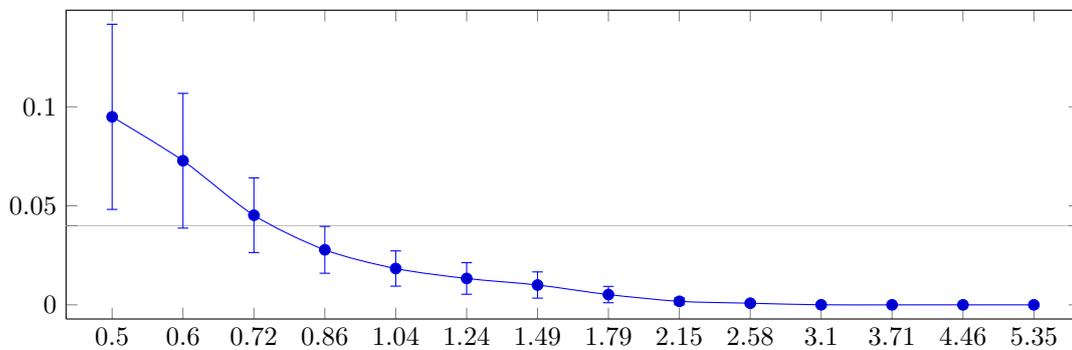


Abbildung 33: Verhalten von *Sharpness*.

Weichzeichnung ab. Für ein nicht weichgezeichnetes Bild erhalten wir als untere  $2\sigma$ -Schwelle einen Sharpness-Koeffizienten von 0.05. Ab einem Radius von 0.86 fällt auch die obere  $2\sigma$ -Schwelle unter 0.05. Lassen wir eine gewisse Unschärfe zu, sollte uns die Entscheidungsschwelle  $t_{sharp}$  in der Nähe von 0.05 gute Werte liefern. Zur einfachen Übersicht ist die später im Algorithmus eingesetzte Grenze von  $t_{sharp} = 0.04$  eingezeichnet.

Selbiges können wir auch für *BlurExtent* auftragen. Hier sehen wir, wie der Faktor sehr schön mit steigender Weichzeichnung  $\infty$  wächst. Um dieses zu visualisieren, wurde eine doppelt-logarithmische Skalierung gewählt, und die Division durch 0 durch eine Division durch 0.01 ersetzt.

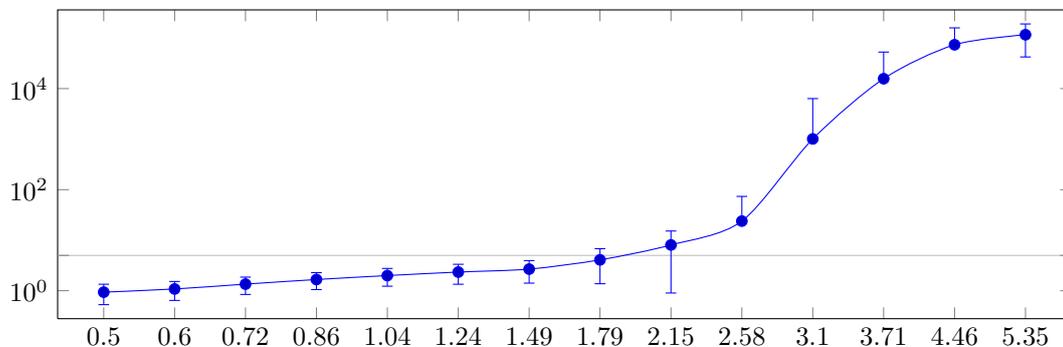


Abbildung 34: Verhalten von *BlurExtent*.

Wiederum ist die später verwendete Grenze von  $t_{blur} = 5$  in Abbildung 34 eingetragen. Sehr schön zu erkennen ist die - in logarithmischer Skalierung - lineare Steigung von *BlurExtent* im unteren Bereich. Hierdurch könnte z. B. der Radius des verwendeten Weichzeichnungsfilter aus einem Bild bestimmt werden.

### 13.2 Benchmark

Um den Algorithmus genau zu untersuchen und herauszufinden, wie effizient und korrekt er arbeitet, wurde ein Benchmark-Satz zusammengestellt. Dieser besteht aus den unterschiedlichsten Bildern. Es wurde zwar verstärkt Wert auf Sportaufnahmen gelegt, er beinhaltet aber auch Detail-Aufnahmen und Landschaftsaufnahmen. Der Benchmark wurde so zusammengestellt, dass die unterschiedlichsten Arten von Unschärfe abgedeckt sind. Er beinhaltet sowohl beabsichtigte, so wie unbeabsichtigte Out-Of-Focus-Blur, als auch Bewegungsunschärfe.

Für diesen Benchmark wurde zuerst eine Datenbank erstellt, welche alle Bilder klassifiziert. Anschließend wurde das Ergebnis des Algorithmus gegengeprüft. Es ist anzumerken, dass der Mensch hierbei auch Fehler begeht. Bei der ersten Durchsicht habe ich etliche Bilder als scharf markiert, welche eigentlich unscharf waren. Dies liegt zum einen daran, dass ich sie in verkleinerter Darstellung analysiert habe, zum anderen daran, dass das subjektive Schärfeempfinden des Menschen von vielen Faktoren abhängt. Diese Fehler wurden im Nachhinein geändert - es wurde dem Algorithmus recht gegeben.

Diese Evaluation ergab, dass 87% aller scharfen Bilder und 97% aller unscharfen Bilder erkannt wurden, wenn man einen  $t_{sharp}$  von 0.04 verwandt. Zusätzlich wurde ein  $t_{blur}$  von 5 verwandt. Das  $t_{blur}$ -Kriterium wurde eingeführt, da bei einem sehr unscharfen Bild nur sehr wenige Kanten erkannt wurden, diese jedoch fälschlicherweise als scharf deklariert wurden, wodurch ein extrem hoher Sharpness-Koeffizient entstand. Diese Fehlerkennungen waren hierbei auf Rauschen zurückzuführen und waren sehr verteilt im Bild. Hier half der BlurExtent-Koeffizient, welcher bei diesem Bild sehr hoch lag.

	Scharf	Unscharf	Total
Korrekt	138	155	293
Falsch	5	20	25

Tabelle 1: Evaluation Benchmark.

Das Ergebnis ist zwar zufriedenstellend, aber es sollte hier erwähnt sein, dass bei vielen Entscheidungen mir die Entscheidung sehr schwer fiel. Lagen die Meinung von mir und dem Algorithmus auseinander, konnte man oft Argumente finden, warum die Entscheidung des Algorithmus richtig ist. Hier wurde nach dem ersten Eindruck entschieden.

Viele fälschlicherweise als scharf erkannte Bilder waren auch von Bewegungsunschärfe betroffen. Viele Bilder waren nur leicht verwackelt, aber dennoch erkennbar. Jedoch ist z. B. Asphalt, welcher nur in einer Richtung verwackelt ist, für den Algorithmus immer noch eine homogene Fläche, welche viele Kanten aufweist. Dies ist genau das Problem, welches in Kapitel 7 beschrieben wurde. Ein Pixel, welcher nur in einer Richtung verwischt wird, wird zu einem Balken - also einer Sprungkante.

Auch geringe Tiefenschärfen stellen für den Algorithmus ein Problem dar. Da wir einen recht großzügigen Bildausschnitt wählen, fällt viel Unschärfe in diesen Bereich, wenn das Bild allgemein nur einen sehr kleinen Tiefenschärfebereich hat. Der scharfe Bereich wird zwar zuverlässig erkannt, jedoch überwiegen in diesem Bildteil dann die unscharfen Elemente.

### 13.3 Sportaufnahmen Satz 1

Um die Ergebnisse des Benchmarks weiter zu analysieren, wurde der nun eingestellte Algorithmus auf einen gesamten Satz einer Sportwagen-Veranstaltung angewandt. Anschließend wurden die Entscheidungen des Algorithmus beurteilt. Hierbei traten des Öfteren starke Entscheidungsschwierigkeiten auf. Bei Problembildern, die auch von einem Zweitgutachter nicht klar als scharf oder unscharf deklariert werden konnten, wurde dem Algorithmus recht gegeben.

	Scharf	Unscharf	Total
Korrekt	353	332	675
Falsch	51	46	97

Tabelle 2: Evaluation Sportaufnahmen 1.

Hier wurden 88% aller scharfen Bilder und 87% aller unscharfen Bilder erkannt.

Es ist anzumerken, dass viele der Falscherkennungen wieder auf Bewegungsunschärfe oder sehr geringe Tiefenschärfe zurückzuführen sind.

### 13.4 Sportaufnahmen Satz 2

Es wurden 89% aller scharfen und 91% aller unscharfen Bilder richtig zugeordnet. Erstaunlich ist hier, dass bei dieser Veranstaltung sehr viele unscharfe Bilder vorkommen. Interessanterweise wurde hier ein schlechteres Objektiv als beim ersten Satz verwendet. Das Objektiv aus dem ersten Satz besitzt einen Ultraschallfokusmotor und einen eingebauten Bildstabilisator. Das hier verwendete Objektiv benutzt einen langsameren, mechanischen Motor zum Fokussieren und hat keinen Bildstabilisator.

	Scharf	Unscharf	Total
Korrekt	110	282	392
Falsch	27	13	40

Tabelle 3: Evaluation Sportaufnahmen 2.

### 13.5 Landschaftsaufnahmen

Um den Algorithmus nicht nur mit einer Art von Sportfotos zu testen, wurde der Algorithmus auf einen Satz Landschaftsaufnahmen angewandt. Es handelt sich hier um Landschaftsaufnahmen vom Cap Menorca auf Mallorca.

Hier kommen deutlich weniger unscharfe Bilder vor, da für jedes Bild natürlich mehr Zeit und Ruhe zur Verfügung stand. Sehr interessant ist hier aber, dass ein Ultra-Weitwinkel Objektiv eingesetzt wurde. Dieses bildet nahezu 180° Blickfeld ab. Es zählt zu den teuersten und besten Objektiven in diesem Bereich, jedoch bildet es bei weitem nicht so scharf ab, wie ein gutes Tele-Objektiv. Da die Gesamtwirkung bei diesen Bildern jedoch im Vordergrund steht, fällt diese Objektivunschärfe meist nicht so stark auf. Der Algorithmus reagiert jedoch trotzdem auf diese Unschärfe.

	Scharf	Unscharf	Total
Korrekt	173	18	191
Falsch	3	9	12

Tabelle 4: Evaluation Landschaftsaufnahmen.

Dies entspricht 95% aller scharfen und 86% aller unscharfen Bildern.

### 13.6 Automobil-Ausstellung

Außerdem wurde ein Satz von Bildern analysiert, welche auf einer Automobil-Ausstellung aufgenommen wurden. Hier wurde viel mit Tiefenunschärfe experimentiert. Dadurch entstanden viele Bilder mit sehr geringer Tiefenschärfe, bzw. viel Tiefenunschärfe - welche ja erwünscht ist. Wir erinnern uns, dass der Algorithmus mit dieser häufig Probleme hatte. Es wurde hier ein recht gutes Objektiv verwendet, welches eine gute Qualität auch bei großen Blendenöffnungen liefert.

Da die Lichtverhältnisse nicht sehr gut waren, entstanden zum Teil auch verwackelte Bilder. Diese wurden zuverlässig erkannt.

	Scharf	Unscharf	Total
Korrekt	163	100	263
Falsch	17	22	39

Tabelle 5: Evaluation Detailaufnahmen.

Die hohe Zahl der unscharfen Bilder liegt daran, dass ich des Öfteren Detailaufnahmen von Scheinwerfern - also durch Glas hindurch - machte. Diese Bilder wurden als Unscharf erkannt. Ich kann jedoch selbst nicht sagen, ob diese Bilder nun scharf oder unscharf sind. Um die Statistik nicht zu zerreißen, wurde hier dem Algorithmus jeweils recht gegeben.

### 13.7 Gesamt Evaluation

Fassen wir alle Evaluation zusammen, so erkennt unser Algorithmus 90% aller scharfen und 89% aller unscharfen Bilder. Insgesamt wurden 1.817 Bilder analysiert, wovon 921 scharf und 887 unscharf sind.

Es ist erstaunlich, dass fast 50% der Bilder als unscharf gelten. Natürlich wurden die Evaluationen auch so ausgelegt, dass möglichst viele unscharfe Bilder vorhanden sind, man erkennt hierbei jedoch trotzdem sehr gut, wie praktisch ein Programm wäre, welches einem die Sortierung abnehmen würde.

	Scharf	Unscharf	Total
Korrekt	828	797	1625
Falsch	90	93	192

Tabelle 6: Gesamt Evaluation.

Alle Evaluationen zusammen benötigten 4.918 Sekunden ( $\approx 82$  Minuten). Dies ergibt im Schnitt eine Verarbeitungszeit von 2.4s pro Bild auf einem vier-kernigen Intel CPU mit 2.66GHz (allerdings wurde der Algorithmus nicht auf multiple Kerne optimiert). Die jeweilige Verarbeitungszeit hängt jedoch stark davon ab, welche Autofokuspunkte aktiviert sind. Die Analyse eines gesamten Bildes dauert natürlich länger, als die eines kleinen Ausschnitts. Dies zeigt auch Tabelle 7.

Satz	Bilder <sup>1</sup>	Zeit	Verarbeitungszeit
Benchmark	318	777s	2.44 s/I
Sportaufnahmen 1	775	1417s	1.83 s/I
Sportaufnahmen 2	432	1280s	2.96 s/I
Landschaftsaufnahmen	216	762s	3.53 s/I
Automobil-Ausstellung	309	682s	2.21 s/I

Tabelle 7: Geschwindigkeit des Algorithmus.

Beim ersten Satz der Sportaufnahmen wurde gezielt mit nur einem Fokuspunkt gearbeitet. Dementsprechend ist auch die Verarbeitungszeit Sekunden/Image entsprechend klein. Bei den Landschaftsaufnahmen waren hingegen sehr oft alle Fokuspunkte aktiv. Die Verarbeitungszeit ist dementsprechend höher.

<sup>1</sup>Die Anzahl der Bilder weicht teilweise ab, da Personenaufnahmen aus den Benchmarks und Sätzen entfernt wurden.

## 14 Fazit & Ausblick

Wir haben in dieser Arbeit gezeigt, dass mittels der HWT scharfe Kanten sehr zuverlässig erkannt werden können.

Hierfür haben wir einen Algorithmus entwickelt, welcher zum Einen scharfe Bereiche in einem Bild visualisieren bzw. erkennen kann, zum Anderen aber auch in der Lage ist, scharfe von unscharfen Bildern zu trennen.

Wie wir im letzten Kapitel untersucht haben, geschieht dies recht zuverlässig mit Trefferquoten im Bereich von 90%. Aufbauend hierauf wäre eine Implementation in bestehende Foto-Management-Systeme sehr interessant, um schnell unscharfe Bilder herauszusortieren. Solch ein System wäre vor allem für Berufsfotographen von Veranstaltungen interessant. Wenn z. B. ein Fußballspiel oder Sportwagenrennen fotografiert wird, so geht es meist darum, schnell die besten Bilder an einen Verlag zu übermitteln. Hat der Fotograf nun die Möglichkeit, direkt alle unscharfen Bilder auszuschließen, so wäre dies eine große Zeitersparnis für ihn. Außerdem könnte man den Bildern die Information, wie scharf das jeweilige Bild ist, hinzufügen (Faktor *Sharpness*). Die Bilder könnten dann anhand dieses Faktors sortiert werden und es wäre leichter, die besten Bilder herauszusuchen. Des Weiteren erleichtert es auch die Entscheidung, welche Aufnahme von einem Motiv man auswählen sollte - anhand des Faktors könnte einfach die schärfste Aufnahme ausgewählt werden. Da der Algorithmus auch auf feinste Unschärfen reagiert, welche bei einem flüchtigen Blick nur schwer zu erkennen sind, wird auch die Gefahr vermindert, schlechte Bilder letztendlich weiterzuverarbeiten.

Der gesamte Prozess könnte auch direkt in der Kamera stattfinden. Wird ein Foto aufgenommen, wird es sofort analysiert. Sollte der Algorithmus ein Bild als unscharf erkennen, so wird eine Warnung ausgegeben. Vor allem für Kompaktkameras wäre dies interessant - sind diese ja eh schon voll gespickt mit vielen Features und Automatismen -, da hier nicht immer viele Aufnahmen von einem Motiv erstellt werden, sondern meistens nur gezielt fotografiert wird. Warnt die Kamera nun davor, dass dieses Bild unscharf sein könnte, so wird die Gefahr von Enttäuschungen, z. B. dass das wichtigste Bild der Hochzeit unscharf ist, vermindert.

Wird der Algorithmus direkt in einer Kamera implementiert, zeigt sich jedoch noch eine weitere viel interessantere mögliche Verwendung. Viele Kameras zeigen mittlerweile über- und/oder unterbelichtete Bereiche direkt auf dem Bildschirm an. Auch die Einblendung eines Histogramms ist direkt möglich. Man kann also schon viele Informationen darüber gewinnen, wie gut das Bild geworden ist. Jedoch ist die Beurteilung der Schärfe auf den kleinen Bildschirmen der Kameras sehr schwierig. Hier könnten, analog zur Visualisierung von überbelichteten Bereichen, auch die scharfen Bereiche visualisiert werden. Dies könnte z. B. so implementiert werden, wie es in Kapitel 10 erfolgte. Man würde so sofort sehen, ob dieses Bild auch dort scharf ist, wo man es scharf haben wollte. Es passiert häufig, dass auf dem kleinen Bildschirm ein Bild scharf aussieht, welches sich später als komplett unbrauchbar herausstellte. Hier wäre es interessant, wie effizient der Algorithmus implementiert werden könnte, zumal moderne Kameras schon direkt eingebaute Bildprozessoren besitzen.

Ein weiterer Forschungspunkt wäre, ob der Algorithmus nicht auch dafür benutzt werden könnte, die scharfen bzw. unscharfen Kanten und Bereiche nicht nur zu erkennen, sondern auch zu verbessern. Durch unseren Algorithmus erhalten wir sowohl eine Information darüber, wie scharf das Bild letztendlich ist (Faktor *Sharpness*), als auch wie stark sich die Unschärfe auf das Bild ausgewirkt hat (*BlurExtent*). Wie wir in Abbildung 34 gesehen haben, besteht ein exponentieller Zusammenhang zwischen verwendetem Radius  $r$  des Weichzeichnungsfilters und *BlurExtent*. Mit Hilfe einer „Anti-Faltung“ könnte so die, durch den Weichzeichnungsfiler hinzugefügte, Unschärfe wieder entfernt werden.

Außerdem müsste sich durch letztere Überlegung auch ein sehr guter Scharfzeichnungsalgorithmus entwickeln lassen, bei dem nur die scharfen Bereiche scharfgezeichnet und die unscharfen Bereiche außer Acht gelassen werden. Hierdurch würde das Rauschen in den unscharfen Bereichen nicht extra verstärkt werden.

## Abbildungsverzeichnis

1	Beispiele zu Out-Of-Focus-Blur. . . . .	4
2	Beispiele zu Motion-Blur. . . . .	5
3	Empfangenes Sinus-Signal mit Störung. . . . .	9
4	Ineinander verschachtelte Darstellungsweise der HWT . . . . .	10
5	Separate Koeffizienten A und D der HWT. . . . .	10
6	Haar-Wavelet-Transformation aus Sicht der digitalen Signalverarbeitung. . . . .	11
7	Grundarten von Kanten. . . . .	12
8	Gauß-Kern für $\sigma = 0.5$ . . . . .	13
9	Zweidimensionaler Gauß-Kern $\sigma = 0.5$ . . . . .	14
10	Zweidimensionales Bild einer Dirac-Kante. . . . .	15
11	Weichgezeichnete Dirac-Impulse für $\sigma = 0.01, \sigma = 0.03, \sigma = 0.05, \sigma = 0.1$ . . . . .	15
12	Idealer Dirac-Impuls in der HWT. . . . .	16
13	Weichgezeichneter Dirac $\sigma = 0.05$ in der HWT. . . . .	17
14	Evaluation des weichgezeichneten Dirac-Impulses. . . . .	18
15	Evaluation des auf 1 normierten, weichgezeichneten Dirac-Impulses. . . . .	19
16	Zweidimensionales Bild einer Sprungkante. . . . .	20
17	Weichgezeichnete Sprung-Kanten für $\sigma = 0.01, \sigma = 0.03, \sigma = 0.05, \sigma = 0.1$ . . . . .	21
18	Ideale Sprung-Kante in der HWT. . . . .	22
19	Weichgezeichneter Sprung $\sigma = 0.05$ in der HWT. . . . .	23
20	Evaluation des weichgezeichneten Sprung. . . . .	24
21	Scharfzeichnung durch Ableitung. . . . .	27
22	2D Kern des Laplacian of Gaussian. . . . .	31
23	Evaluation des weichgezeichneten Dirac-Impulses mit Scharfzeichnung. . . . .	32
24	Evaluation des weichgezeichneten Sprung mit Scharfzeichnung. . . . .	32
25	Sprungkante mit Bewegungsunschärfe . . . . .	33
26	Dirac-Impuls mit Bewegungsunschärfe . . . . .	33
27	Zweidimensionale Haar-Wavelet-Transformation. . . . .	36
28	Demonstrationsbild zur Kantenerkennung. . . . .	38
29	Visualisierte Dirac- und Sprungkanten im Bild. . . . .	38
30	Demonstrationsbild zur Schärfekerennung. . . . .	39
31	Beispiele zur Visualisierung des Schärfebereichs. . . . .	39
32	Erkennung ungewollter, scharfer Kanten. . . . .	40
33	Verhalten von <i>Sharpness</i> . . . . .	43
34	Verhalten von <i>BlurExtent</i> . . . . .	43

## Tabellenverzeichnis

1	Evaluation Benchmark. . . . .	44
2	Evaluation Sportaufnahmen 1. . . . .	45
3	Evaluation Sportaufnahmen 2. . . . .	45
4	Evaluation Landschaftsaufnahmen. . . . .	45
5	Evaluation Detailaufnahmen. . . . .	46
6	Gesamt Evaluation. . . . .	46
7	Geschwindigkeit des Algorithmus. . . . .	46

## Literatur

- [BB06] Burger, Wilhelm und Mark James Burge: *Digitale Bildverarbeitung*. Springer, 2. Auflage, 2006.
- [GW08] Gonzalez, Rafael C. and Richard E. Woods: *Digital Image Processing*. Pearson Education, 3rd edition, 2008.
- [MH92] Mallat, S. and W.L. Hwang: *Singularity detection and processing with wavelets*. Information Theory, IEEE Transactions on, 38(2):617–643, 1992.
- [SDS95a] Stollnitz, Eric J., Tony D. Deroose, and David H. Salesin: *Wavelets for computer graphics: A primer, part 1*. IEEE Computer Graphics and Application, 15(3):76–84, May 1995.
- [SDS95b] Stollnitz, Eric J., Tony D. Deroose, and David H. Salesin: *Wavelets for computer graphics: A primer, part 2*. IEEE Computer Graphics and Application, 15(4):75–85, July 1995.
- [TLZZ] Tong, Hanghang, Mingjing Li, Hongjiang Zhang, and Changshui Zhang: *Blur detection for digital images using wavelet transform*. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 1, pages 17–20. IEEE.

## Anlagen

Die Anlagen zu dieser Bachelorthesis sind bei Prof. Dr. rer. nat. Annabella Rauscher-Scheibe in Raum 12.80 in digitaler Form einsehbar.

Der Datenträger enthält

- verwendete Matlab-Files.
- verwendete Bilder in komprimierter Form
  - Simulation
  - Benchmark
  - Benchmark mit visualisierten Schärfebereichen (`Benchmark_Vis`)
  - Sportaufnahmen Satz 1 (`09_07_05_Sportwagenmeeting_2009`)
  - Sportaufnahmen Satz 2 (`10_07_04_Sportwagenmeeting`)
  - Landschaftsaufnahmen (`10_09_21_Mallorca_Cap_Menorca_Formentor`)
  - Automobil-Ausstellung (`11_05_06_AUTOmobil`)

# Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 12. Dezember 2011

Ort, Datum

Unterschrift