

Wavelet Transformation von Standbildern

Studienarbeit
von
Christoph Esser
aus
Mannheim

vorgelegt am
Lehrstuhl für Praktische Informatik IV
Prof. Dr. W. Effelsberg
Fakultät für Mathematik und Informatik
Universität Mannheim

Februar 2001

Betreuerin: Dipl.-Math. oec. Claudia Schremmer

Inhaltsverzeichnis

ABBILDUNGSVERZEICHNIS.....	II
1 EINLEITUNG.....	1
1.1 MOTIVATION.....	1
1.2 AUFGABENSTELLUNG.....	1
1.3 THEMENÜBERBLICK.....	1
2 DIE GESCHICHTE DER WAVELETS.....	2
2.1 DIE FOURIER-TRANSFORMATION.....	3
2.2 WAVELET GRUNDLAGEN.....	6
2.3 DIE KONTINUIERLICHE WAVELET-TRANSFORMATION.....	8
2.4 DIE SCHNELLE WAVELET-TRANSFORMATION.....	11
2.5 WAVELET-TRANSFORMATION OHNE WAVELETS: FILTER.....	13
3 WAVELET-TRANSFORMATION DEMO APPLLET.....	15
3.1 AUFBAU UND ABLAUF DES APPLETS.....	15
3.1.1 <i>Selektion der Transformations-Parameter.....</i>	<i>16</i>
3.1.2 <i>Anzeige und Steuerung der Transformation.....</i>	<i>20</i>
3.1.3 <i>Die Applet-Menüleiste.....</i>	<i>21</i>
3.1.4 <i>Das Hilfe-Fenster.....</i>	<i>21</i>
3.2 IMPLEMENTIERUNG DES APPLETS.....	22
3.2.1 <i>Wie werden die einzelnen Filter berechnet ?.....</i>	<i>22</i>
3.2.2 <i>Wie läuft die Analyse ab ?.....</i>	<i>22</i>
3.2.3 <i>Wie läuft die Synthese ab ?.....</i>	<i>23</i>
3.2.4 <i>Wie wird die Transformation dargestellt und welche Probleme gibt es dabei ?.....</i>	<i>23</i>
3.3 HEURISTISCHE ERGEBNISSE.....	26
3.3.1 <i>Beispiel 1: Horizontale und vertikale Filterung.....</i>	<i>26</i>
3.3.2 <i>Beispiel 2: Einfluß der Filterlänge beim Ringschluß-Verfahren.....</i>	<i>27</i>
3.3.3 <i>Beispiel 3: Quantisierung.....</i>	<i>27</i>
3.3.4 <i>Abschließende Beurteilung der Bildqualität bei der Synthese (inkl. Quantisierung).....</i>	<i>29</i>
4 DIE ZUKUNFT DER WAVELETS.....	30
4.1 JPEG2000.....	30
LITERATURVERZEICHNIS.....	31

Abbildungsverzeichnis

ABBILDUNG 1: HELBIGKEITSVERLAUF BEI BILDERN	2
ABBILDUNG 2: ÜBERLAGERUNG VON SINUS UND KOSINUS	3
ABBILDUNG 3: FT IM VERGLEICH ZUR WFT	4
ABBILDUNG 4: DIE „MEXICAN HAT“ WAVELETFAMILIE	6
ABBILDUNG 5: VERSCHIEDENE BASIS-WAVELETS	7
ABBILDUNG 6: EINE WAVELET-TRANSFORMATION	8
ABBILDUNG 7: ANALYSE EINES SIGNALS MIT EINEM WAVELET	9
ABBILDUNG 8: VERSCHIEBEN UND STECKEN EINES WAVELETS BEI DER DWT	10
ABBILDUNG 9: ABLAUF DER FWT (ANALYSE)	12
ABBILDUNG 10: ABLAUF DER FWT (SYNTHESE)	12
ABBILDUNG 11: ALTERNATING FLIP PATTERN	14
ABBILDUNG 12: HAAR UND DAUBECHIES 4 FILTERBÄNKE	14
ABBILDUNG 13: DAS PARAMETER-AUSWAHL FENSTER (PAF)	15
ABBILDUNG 14: DAS TRANSFORMATIONS-ANZEIGE FENSTER (TAF)	15
ABBILDUNG 15: STANDARD DEKOMPOSITION	17
ABBILDUNG 16: NON-STANDARD DEKOMPOSITION	17
ABBILDUNG 17: INFORMATIONSBEREICH DES TAF	20
ABBILDUNG 18: SETTINGS FENSTER	21
ABBILDUNG 19: TRIMMEN DER APPROXIMATION BEIM ZEROPADDING VERFAHREN	25
ABBILDUNG 20: MIRRORPADDING VERFAHREN	25
ABBILDUNG 21: BILDVIERTEL BEI DER NON-STANDARD DEKOMPOSITION	26

1 Einleitung

1.1 Motivation

Sei es bei Digitalkameras, der Datenübertragung im Internet, oder dem Aufbau von großen Bilddatenbanken – der Wunsch der Anwender nach immer effizienteren und qualitativ besseren Algorithmen zur Bilddatenkompression steht oft an erster Stelle. Die heute im Einsatz befindlichen Methoden (GIF, JPEG, TIFF, usw.) erscheinen oftmals bezüglich der Kriterien Kompressionsfaktor und Bildqualität als zu unausgewogen. Die Bilddatenkompression auf Basis der Wavelet-Theorie könnte sich dabei in Zukunft als recht guter Kompromiß erweisen.

Die vorliegende Studienarbeit ist im Rahmen des *VIROR-Projektes* (Virtuelle Hochschule Oberrhein) im Fachbereich *Praktische Informatik IV - Multimediatechnik* der Universität Mannheim entstanden. Ziel des VIROR-Projektes ist es, daß Studierende neben den traditionellen Veranstaltungen (Vorlesungen und Übungen) auch die Möglichkeit haben, den behandelten Stoff über das Internet abzurufen und mit entsprechenden interaktiven Demos zu veranschaulichen und zu vertiefen. Ferner bietet das VIROR-Projekt auch Studenten anderer Hochschulen, sowie allen Interessierten die Möglichkeit, die Vorlesungen „virtuell“ zu besuchen.

Diese Arbeit soll anhand eines Java-Applets die Anwendung der Wavelet-Theorie auf 2D-Standbilder und deren Möglichkeit zur Bilddatenkompression erläutern.

1.2 Aufgabenstellung

Das Demo-Applet sollte in der Programmiersprache Java unter Verwendung der Swing Klassen (ab Version 1.2) geschrieben werden.

Es sollen 2D-Standbilder eingelesen, jeweils unter Berücksichtigung entsprechender Parameter mittels geeigneter Wavelet-Verfahren transformiert (Analyse) und anschließend wieder rücktransformiert (Synthese) werden.

Als Parameter der Analyse sind vorgesehen (mehr dazu in Kapitel 3.1):

- Auswahl von verschiedenen Filtern,
- Auswahl der Dekompositions-Methode,
- Auswahl des Rand-Extensions Verfahrens.

Als Parameter der Synthese sind vorgesehen:

- Auswahl eines Schwellenwertes für die Quantisierung,
- Auswahl von verschiedenen Darstellungsmethoden der Synthese.

1.3 Themenüberblick

In *Kapitel 2* sollen zunächst die historische Entwicklung und die Grundlagen der Wavelet-Theorie dargestellt werden. *Kapitel 3* befaßt sich anschließend mit dem Aufbau und der Funktionsweise des Demo-Applets zur Wavelet Transformation und beschreibt einige wichtige Aspekte der Implementation. *Kapitel 4* gibt schließlich einen Ausblick auf die geplante Standardisierung der Bildkompression mit Wavelets im JPEG2000-Verfahren.

2 Die Geschichte der Wavelets

Die Wavelet-Theorie wurde um 1980 maßgebend von zwei Physikern – Jean Morlet und Alex Grossmann – in Frankreich entwickelt. Ausgangspunkt dabei war die bis dato in der Signalverarbeitung häufig angewandte *Fourier-Transformation*, auf welche im folgenden Kapitel näher eingegangen werden soll. Daran anschließend werden die Grundlagen der Wavelet-Theorie, sowie deren Anwendung in der *kontinuierlichen, diskreten und schnellen Wavelet-Transformation* behandelt.

Zuvor jedoch noch zwei kleine Begriffserklärungen:

- Unter einer **Transformation** versteht man mathematisch gesehen einen Basiswechsel, d.h. man sieht den gleichen Sachverhalt aus einem anderen Blickwinkel – also auf einer anderen Basis aufbauend. Bei der Transformation geht die Information nicht verloren, sie wird nur anders dargestellt. Transformationen sind i.A. reversibel. Ein triviales Beispiel für eine Transformation ist die Multiplikation zweier römischer Zahlen [BUR97, S. 31]. Hat man die Aufgabe die beiden römischen Zahlen LXXXVI und XLI zu multiplizieren, so übersetzt (transformiert) man diese zunächst in arabische Zahlen, multipliziert $86 \cdot 41 = 3526$ und übersetzt (retransformiert) das Ergebnis wieder in römische Zahlen. Also: $LXXXVI \cdot XLI = MMMDXXVI$.
- In den nachfolgenden Kapiteln ist als Ausgangspunkt einer Transformation immer wieder von **Funktionen**, welche **Signalverläufe** beschrieben, die Rede. Dies können beispielsweise der Signalverlauf eines EKGs, oder der zeitliche Verlauf der Abkühlung eines zuvor erhitzten Metallstabes sein. Im Bereich der Bildverarbeitung kann man als Signalverlauf die Änderung von jeweils benachbarten Pixelwerten verstehen. Diese Pixelwertänderungen können anschließend als Funktion formuliert werden. Ein Beispiel dafür zeigt Abbildung 1. Ein Bild besteht aus einem schwarzen Kreis auf weißem Hintergrund. Der Verlauf der Änderung der Pixelwerte ist als Funktion im unteren Teil dargestellt.

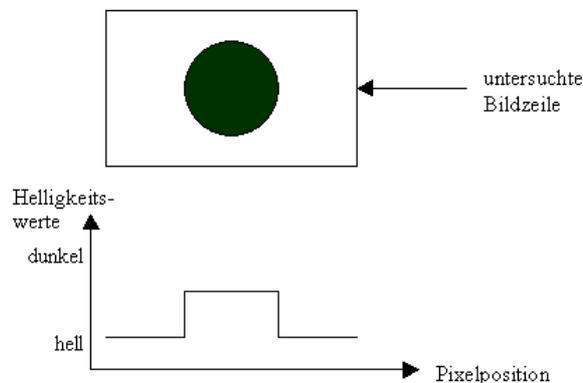


Abbildung 1: Helligkeitsverlauf bei Bildern

2.1 Die Fourier-Transformation

Die von Jean Baptiste Joseph Fourier zu Beginn des 19. Jahrhunderts entwickelte Theorie der **Fourier-Transformation (FT)** (auch **Fourier-Analyse** genannt), kann man sich am besten als eine Art mathematisches Prisma vorstellen. So wie ein Prisma das Licht in seine Farbbestandteile aufspaltet, zerlegt die Fourier-Transformation eine gegebene zeit- oder auch ortsabhängige Funktion g , in die in ihr vorkommenden Frequenzanteile. Diese Frequenzbestandteile von g werden in einer neuen, frequenzabhängigen Funktion \hat{g} (der **Fourier-Transformierten** oder **Fourier-Reihe**) beschrieben.

Inhaltlich wird in beiden Funktionen g und \hat{g} der gleiche Sachverhalt beschrieben: g stellt den Verlauf eines Signals (und dessen Änderungen) abhängig von der Zeit dar – der Ort bzw. der Zeitpunkt einer Signaländerung sind direkt erkennbar. Man spricht dabei auch von einer Darstellung des Signals im **Ortsraum**. Die Funktion \hat{g} beschreibt das Signal im **Frequenzraum**. Hierbei werden die im Signal enthaltenen Frequenzen (also die Intensitäten der Signaländerungen) dargestellt. Eine Aussage über das örtliche bzw. zeitliche Auftreten der Frequenzen wird dabei nicht gemacht.

Der von Fourier 1807 formulierte Satz besagt nun, daß jede periodische Funktion (also ein zeitabhängiges Signal) als Summe von geeigneten Kosinus- und Sinustermen als Basis-schwingungen (bzw. -frequenzen) dargestellt werden kann. Die Beschreibung der Ausgangsfunktion erfolgt hierbei durch Überlagerung der Amplituden der einzelnen, geeignet phasenverschobenen Kosinus- und Sinusschwingungen. Auch gewisse nichtperiodische Funktionen lassen sich mit Hilfe der Fourier-Transformation zerlegen. Voraussetzung dafür ist jedoch, daß diese so schnell abfallen, daß die Fläche unter dem Graphen endlich ist [BUR97, S. 25].

Ein Beispiel für die Überlagerung von Funktionen ist in Abbildung 2 dargestellt: Hierbei ergibt sich die Funktion in c) durch die Überlagerung der Funktionen in a) und b).

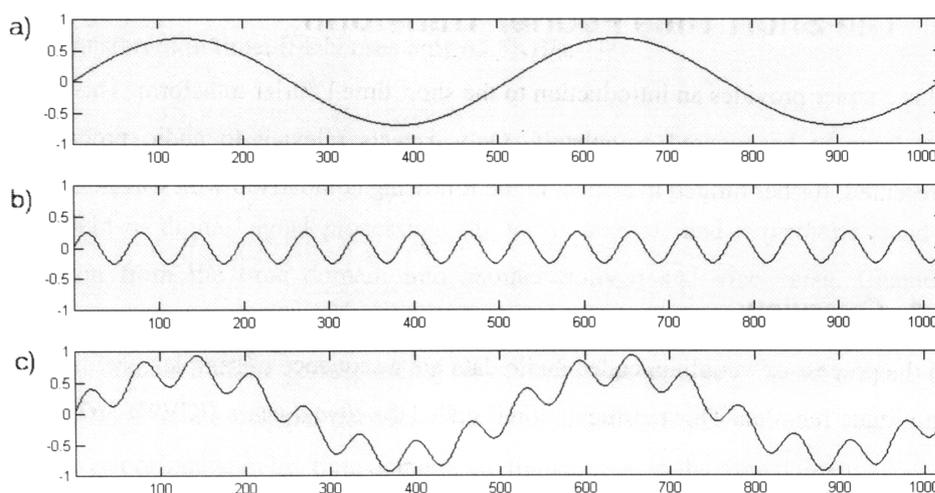


Abbildung 2: Überlagerung von Sinus und Kosinus [BOE00, S. 12]

Die Fourier-Transformation ist auch wieder umkehrbar, d.h. aus einer gegebenen Fourier-Reihe bzw. Transformierten kann auch wieder die entsprechende Ausgangsfunktion rekonstruiert werden.

Probleme der Fourier-Transformation:

Die Information über das zeitliche Auftreten der einzelnen Frequenzen einer Fourier transformierten Funktion ist nicht direkt erkennbar. Diese Zeitinformation ist versteckt in den Phasen der zugrundeliegenden Sinus- und Kosinusschwingungen und wird erst durch deren Überlagerung sichtbar. Bei vielen Untersuchungsproblemen, wie bspw. der Analyse eines Musikstückes, ist jedoch die gleichzeitige Information über die vorkommenden Frequenzen, sowie auch deren zeitliches Auftreten, erwünscht. Dieses Manko der Fourier-Transformation sollte (zumindest zum Teil) durch die nachfolgend beschriebene gefensterte Fourier-Transformation behoben werden.

Die Grundidee der *gefensterten Fourier-Transformation* (*Windowed Fourier-Transform: WFT*) liegt darin, die im Signal enthaltenen Frequenzen abschnittsweise mittels der klassischen Fourier-Analyse zu berechnen, um somit innerhalb dieser Abschnitte (Fenster) Aussagen über die auftretenden Frequenzen machen zu können. Diese sind dann abschnittsweise zeitlich lokalisiert. Das Analyse-Fenster wird entsprechend entlang des Signals verschoben, bis das gesamte Signal analysiert wurde.

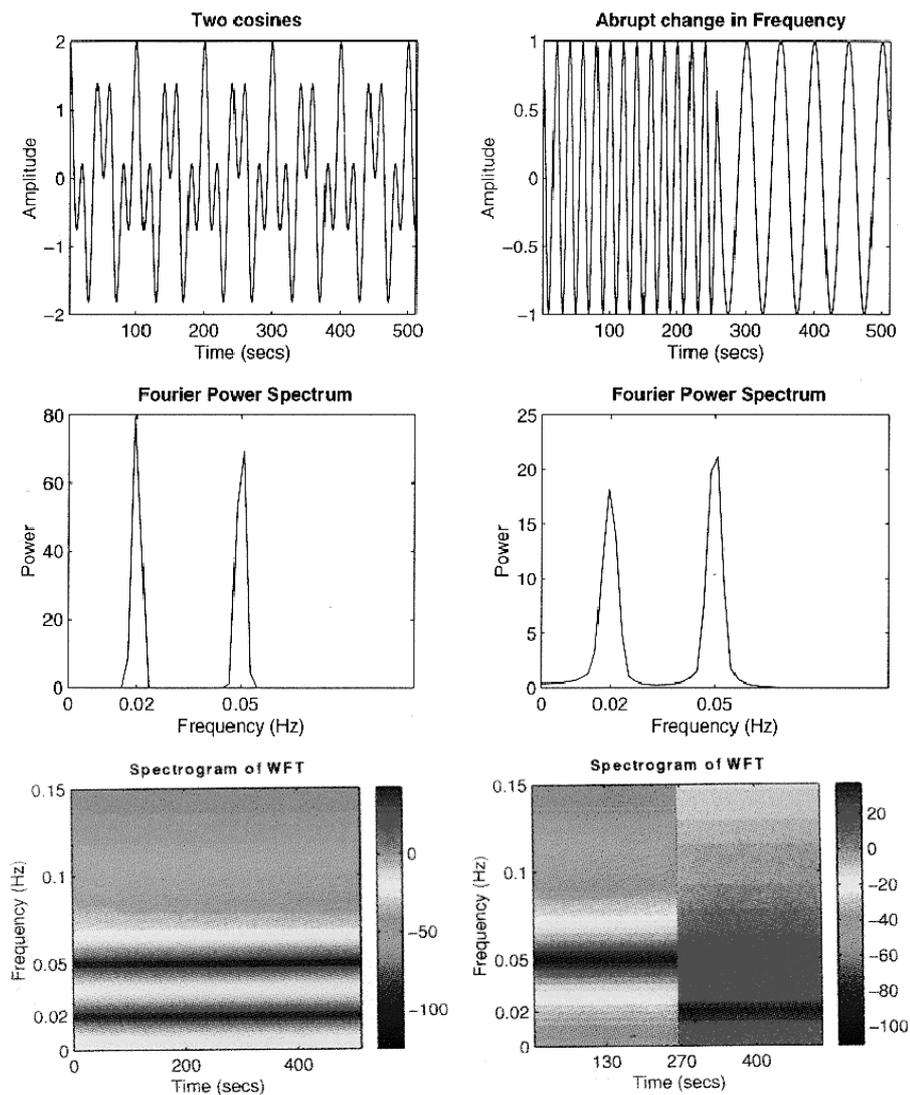


Abbildung 3: FT im Vergleich zur WFT [RUS00, Kap. 8]

Abbildung 3 zeigt zwei verschiedene Signalverläufe:

- *linke Abbildungsseite*: zwei überlagerte Kosinusschwingungen (das Signal ist periodisch über seinen gesamten Verlauf),
- *rechte Abbildungsseite*: eine abrupte Frequenzänderung im Signal (welches nur lokal beschränkt periodisch ist).

Die beiden mittleren Abbildungen stellen jeweils die Fourier-Transformierte der beiden Ausgangssignale dar. Dabei wird deutlich, daß beide Signale die gleichen Frequenzen enthalten. Mit der klassischen Fourier-Analyse kann jedoch nicht unterschieden werden, wo (bzw. wann) im Signal die enthaltenen Frequenzen auftreten. Deshalb sehen beide mittleren Spektrogramme ähnlich aus.

Die beiden unteren Abbildungen zeigen jeweils die Spektrogramme der entsprechenden gefensterten Fourier-Transformation. Beim linken WFT Spektrogramm ändert sich aufgrund der Periodizität des Ausgangssignals, gegenüber der klassischen FT nichts. Das rechte WFT Spektrogramm zeigt die Verwendung eines entsprechenden Fensters bei der Analyse des Ausgangssignals. Dort ist es nun möglich eine zeitlich lokalisierte Aussage über die vorkommenden Frequenzen zu machen.

Der aufmerksame Leser wird sich an dieser Stelle sicherlich fragen: „Wie groß soll ich den nun das Analyse-Fenster machen? Was sind die Auswirkungen?“

Anhand dieser Fragen werden auch die Beschränkungen der FWT deutlich: Ein schmales Analyse-Fenster bietet eine sehr gute *Zeitauflösung*, d.h. plötzliche Signaländerungen sowie Signalspitzen und Sprünge können gut lokalisiert werden. Nachteil dabei ist dann jedoch, daß die Analyse unempfindlich gegenüber niedrigen Signalfrequenzen wird, welche nur in einem kleinen Bereich durch das Analyse-Fenster abgedeckt sind. Ein schmales Fenster hat also eine schlechte *Frequenzauflösung*, da das Fenster zu klein ist, um die niederfrequenten Schwingungen zu erfassen. Umgekehrt bieten große Fenster eine gute Erfassung der vorhandenen Frequenzen, können diese jedoch nicht mehr entsprechend zeitlich lokalisieren. Dieses Phänomen der Unschärfe zwischen Zeit- und Frequenzauflösung kann durch die von Werner Heisenberg definierte *Unschärferelation* erklärt werden (siehe dazu auch [BUR97, S. 72]).

Bei der Signalanalyse mit einer gegebenen Fenstergröße ist in der gefensterten Fourier-Analyse die Unschärfe zwischen Zeit- und Frequenzauflösung konstant, d.h. man legt vorher fest, wie fein (im Sinne von lokalisierten Frequenzen) die Analyse sein soll.

Auch die im folgenden Abschnitt beschriebene *Wavelet-Transformation* (welche ein alternatives Verfahren zur WFT darstellt) kann die Heisenberg'sche Unschärferelation nicht außer Kraft setzen. In Worten der WFT gesprochen, ist es bei der Wavelet-Transformation nun möglich, das Analyse-Fenster (welches hier dehn- und stauchbare Funktionen sind) variabel während einer Analyse an das Signal anzupassen. Damit können sowohl tiefe Frequenzen (mit hoher Frequenzauflösung und schlechter Zeitauflösung) als auch rasche Signaländerungen (mit hoher Zeitauflösung und niedriger Frequenzauflösung) in einer Analyse erfaßt werden [LUE97, Kap. 4.1].

2.2 Wavelet Grundlagen

Wavelets („kleine Wellen“) sind mathematische Funktionen, welche, ähnlich den Sinus- und Kosinus Funktionen der Fourier-Transformation, dazu benutzt werden, eine gegebene Funktion (Signal) auf die in ihr enthaltenen Frequenzbestandteile hin zu untersuchen. Zu diesem Zweck können Wavelets gestreckt bzw. gestaucht (*skaliert*) werden oder an eine bestimmte Stelle des Signals *verschoben* werden. Ausgangspunkt einer Wavelet-Transformation ist ein **Basis-** bzw. **Mutter-Wavelet** ψ . In der Transformation selbst kommen dann die entsprechend skalierten und örtlich verschobenen Varianten des Basis-Wavelets zum Einsatz. Diese Varianten werden auch allgemein als **Waveletfamilie** bezeichnet. Abbildung 4 zeigt beispielhaft die „Mexican hat“-Waveletfamilie.

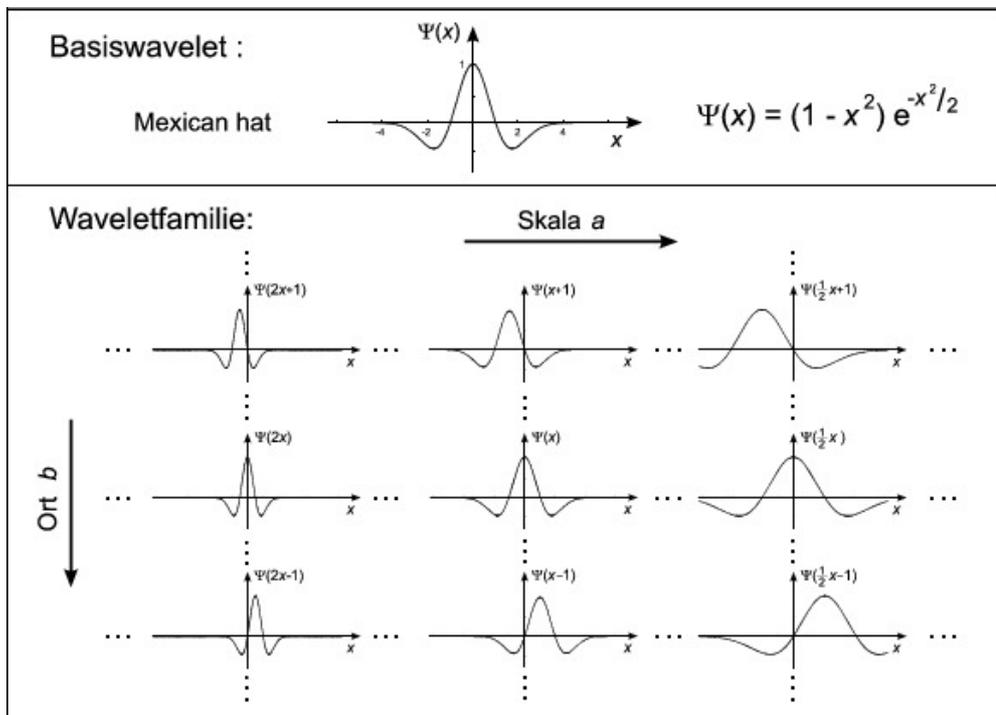


Abbildung 4: Die „Mexican hat“-Waveletfamilie [RAU99, Kap. 2.3]

Grundsätzlich muß ein Basis-Wavelet ψ folgenden Bedingungen genügen:

- Es muß die Zulässigkeitsbedingung

$$0 < c_\psi := 2\pi \int \frac{|\psi(\omega)|^2}{|\omega|} d\omega < +\infty$$

Gleichung 1: Zulässigkeitsbedingung [EFF00, Kap. 2]

erfüllen, d.h. das Wavelet ist nur auf einem begrenzten Intervall ungleich Null.

- Das Integral über der Wavelet-Funktion muß gleich Null sein, also $\int \psi(x) dx = 0$.

Gleichung 2: Integral über eine Wavelet-Funktion [EFF00, Kap. 2]

Ein Wavelet $\psi_{a,b}$ aus einer Waveletfamilie kann beschrieben werden durch:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \text{ mit } a \in \mathfrak{R}^+, b \in \mathfrak{R},$$

Gleichung 3: Analyse-Wavelet [RAU99, Kap.2.1]

wobei a der **Skalierungsfaktor** (also der Grad der Streckung bzw. Stauchung) ist und b den **Verschiebungsfaktor** des Wavelets angibt.

Abbildung 5 zeigt beispielhaft verschiedene Basis-Wavelets:

- Haar,
- Daubechies 4,
- Daubechies 20.

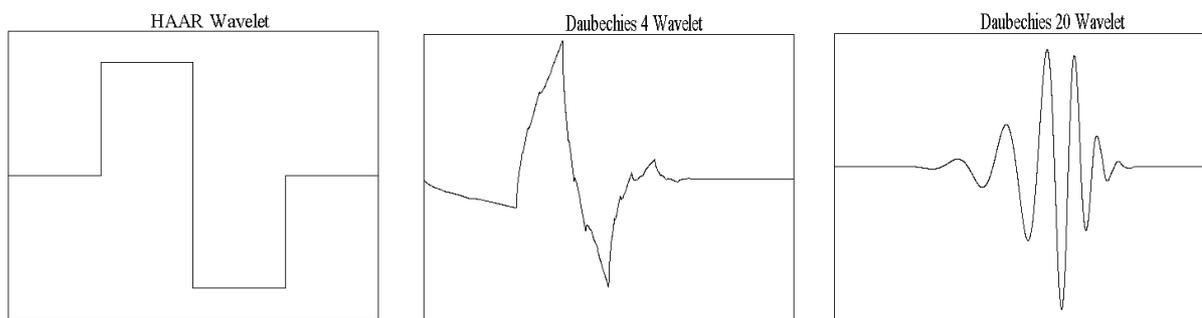


Abbildung 5: Verschiedene Basis-Wavelets [CLE99]

2.3 Die kontinuierliche Wavelet-Transformation

Bei der *kontinuierlichen Wavelet-Transformation* (*Continuous Wavelet-Transform: CWT*) beruht die Analyse des Signals auf der Erkenntnis, daß die Wavelets beim Strecken oder Stauchen ihre Frequenz ändern, wodurch sie sich den verschiedenen Signalkomponenten automatisch anpassen können [BUR97, S. 51]. Somit wird bei der Analyse das Signal zunächst mit einem stark gestreckten (also hoch skalierten) Wavelet untersucht, um die tiefen Frequenzen (Grundfrequenzen) zu bestimmen. Anschließend komprimiert man das Wavelet schrittweise - es wird also in verschiedenen *Auflösungen* dargestellt - um immer feinere Frequenzen (Details bzw. schnelle Änderungen) zu erfassen.

Die durch vollständige Abtastung des Signals mit einem entsprechend skalierten Wavelet entstandene Funktion wird als *Wavelet-Transformierte* bezeichnet. Sie beschreibt also die Übereinstimmung des Signals an allen Stellen mit dem analysierenden Wavelet (mit einem bestimmten Skalierungsfaktor).

Die Rekonstruktion des Ausgangssignals erfolgt analog zur Fourier-Transformation (bei der geeignete Sinus- und Kosinusschwingungen überlagert werden), durch die Addition der entsprechenden Wavelet-Transformierten.

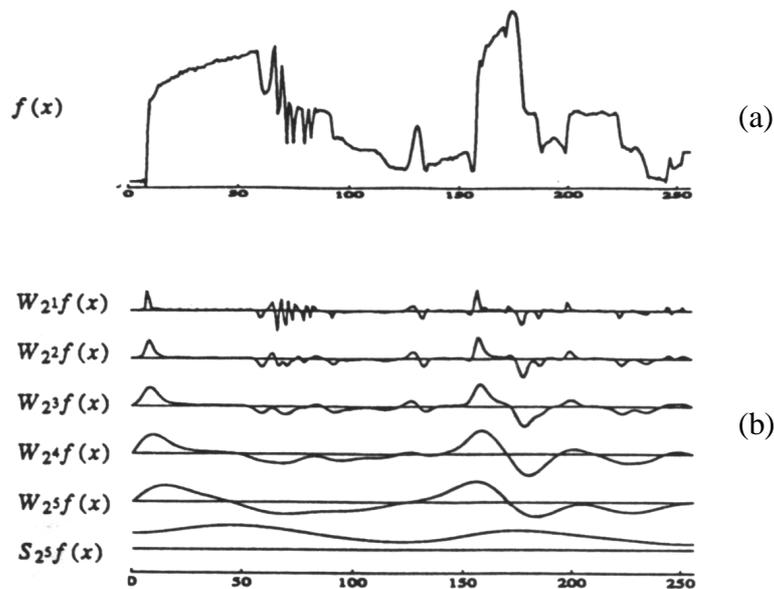


Abbildung 6: Eine Wavelet-Transformation [BUR97, S. 50]

Abbildung 6 zeigt ein Beispiel für eine Wavelet-Transformation. Die Funktion in (a) beschreibt das Ausgangssignal. Die Funktionen in (b) stellen die jeweiligen Wavelet-Transformierten in unterschiedlichen Auflösungen (durch Skalierung des Wavelets) dar. Die unterste Funktion wurde mit einem stark gestreckten Wavelet analysiert. Sie beschreibt die Grundfrequenzen, welche im Ausgangssignal vorkommen. Alle weiter oben dargestellten Wavelet-Transformierten enthalten zunehmend mehr Details. Das verwendete Wavelet wurde entsprechend immer stärker gestaucht, um eventuelle Übereinstimmungen mit dem Signal zu finden. Überlagert man nun alle Funktionen in (b), so erhält man wieder das Ausgangssignal.

Nun stellt sich die Frage, wann ein entsprechend gestrecktes Wavelet mit einem Teil des Signals übereinstimmt. Dazu multipliziert man die Wavelet-Funktion mit dem überdeckten Signalabschnitt. Diese Multiplikation ergibt eine neue Funktion, über welche man anschließend das Integral berechnet. Den Wert dieses Integrals bezeichnet man auch als **Wavelet-Koeffizient**.

Auf Basis der Skalierungs- und Verschiebungs-Parameter a bzw. b aus Kapitel 2.2, stellt sich die kontinuierliche Wavelet-Transformation als Funktion $CWT(a,b)$ in Abhängigkeit zweier Variablen (*Skale* und *Zeit*), wie folgt dar:

$$CWT(a,b) = \int_{-\infty}^{+\infty} x(t) * \psi_{a,b}(t) dt \quad x(t) : \text{Wert der Ausgangsfunktion zur Zeit } t$$

Gleichung 4: Definition der CWT [RAU99, Kap. 2.1]

Abbildung 7 zeigt wie ein Wavelet sukzessive mit einer Ausgangsfunktion verglichen wird. Die Ausgangsfunktion ist in (a) dargestellt, das analysierende Wavelet in (b). In (c) stimmt das Wavelet mit dem gerade überdeckten Abschnitt des Signals annähernd überein. Entsprechend ergibt die Multiplikation von Wavelet und dem Signal eine neue Funktion (d). Berechnet man nun deren zugehörigen Wavelet-Koeffizienten (also die Fläche unter der Funktion), so stellt man fest, daß dieser relativ groß ist. Große Wavelet-Koeffizienten sind also ein Maß für gute Übereinstimmung zwischen analysierendem Wavelet und dem betrachteten Signalabschnitt. Verschiebt man das Wavelet entlang des Signals weiter nach rechts, so ergibt sich die Situation dargestellt in (e). Multiplikation beider Funktionen ergibt die Funktion in (f). Deren Wavelet-Koeffizient ist sehr klein, beschreibt also ein Maß für geringe bzw. keine Übereinstimmung zwischen Wavelet und dem Signalabschnitt.

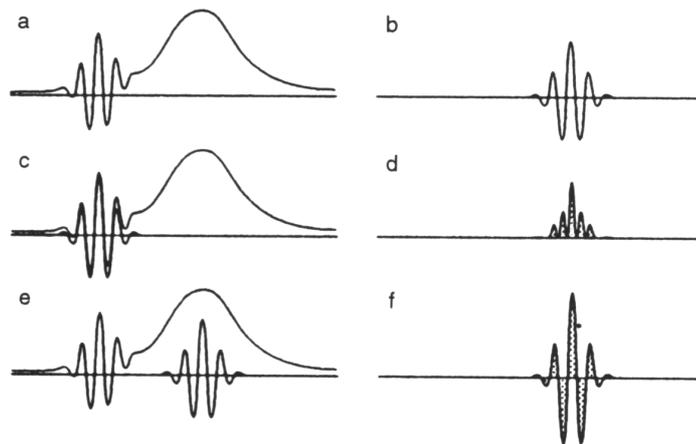


Abbildung 7: Analyse eines Signals mit einem Wavelet [BUR97, S. 53]

In weiteren Analyseschritten würden dann entsprechend unendlich viele skalierte und verschobene Wavelets aus der entsprechenden Waveletfamilie abschnittsweise mit dem Signal verglichen und die jeweilige Übereinstimmung in Form der Wavelet-Koeffizienten bestimmt.

Durch die theoretisch mögliche Bestimmung von unendlich vielen Wavelet-Koeffizienten ist die CWT natürlich wegen Überlappung einzelner Wavelets in höchstem Maße redundant und

sehr rechenintensiv. In der praktischen Anwendung versucht man daher die Skalen- und Zeitparameter der CWT einzuschränken. Dies ist beispielsweise mit der **diskreten Wavelet-Transformation** (*Discrete Wavelet-Transform: DWT*) möglich. Die Berechnung der DWT erfolgt analog zu der CWT, jedoch mit dem Unterschied, daß nun für die Skalen und Zeit Parameter a bzw. b , jeweils diskrete Werte verwendet werden. Dabei gilt es jedoch a und b so geschickt zu wählen, daß zum einen nur eine sehr geringe Redundanz bei den Wavelet-Koeffizienten auftritt (im Besten Fall gar keine) und zum anderen, daß das Ursprungssignal trotzdem noch vollständig beschrieben werden kann, also rekonstruierbar ist. Als Beispiel wäre hier die *dyadische DWT* zu nennen, bei der die Wavelets um entsprechende Potenzen von 2 gestreckt werden, also $a = 2^m$, mit $m \in \mathbb{Z}$ als Transformationsstufen.

Abbildung 8 zeigt die Anwendung der DWT auf ein Signal. Das Wavelet wird zunächst entlang des Signals verschoben (obere Bildhälfte); anschließend wird es entsprechend skaliert (untere Bildhälfte) und die Verschiebung wiederholt sich.

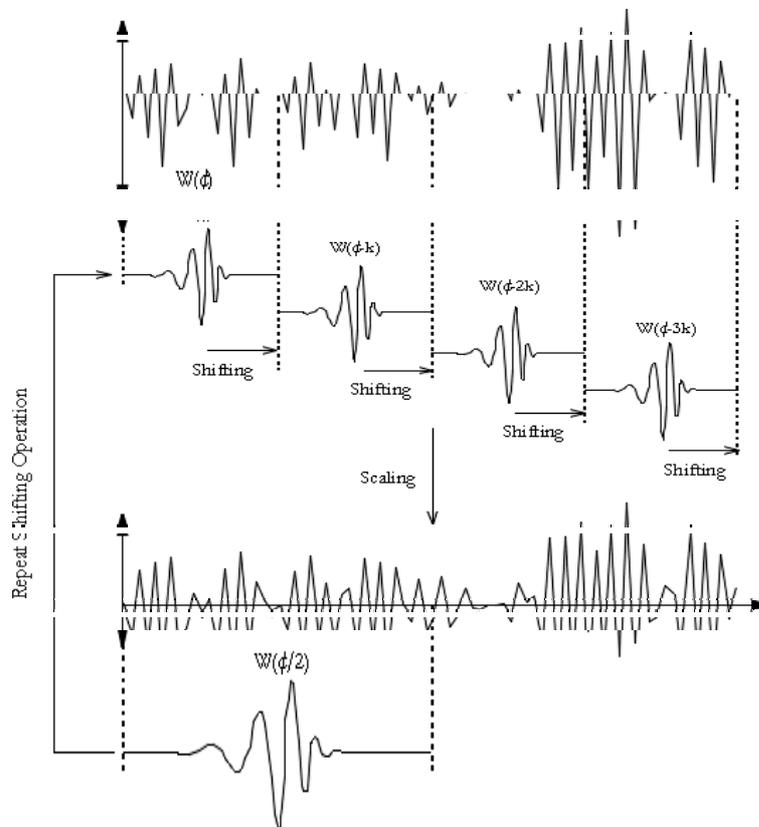


Abbildung 8: Verschieben und Strecken eines Wavelets bei der DWT [ALT96]

Auch die Diskretisierung der Wavelet Parameter in der DWT brachte in der Praxis nicht immer den gewünschten Erfolg der Transformationsvereinfachung. Der Durchbruch, der vor allem in der Bildverarbeitung eine stärkere Anwendung der Wavelet-Theorie ermöglichte, gelang 1986 mit der von Stéphane Mallat und Yves Meyer entwickelten **schnellen Wavelet-Transformation**.

2.4 Die schnelle Wavelet-Transformation

Die *schnelle Wavelet-Transformation* (*Fast Wavelet-Transform: FWT*) basiert auf einem Verfahren, welches schon seit einiger Zeit aus der Signalverarbeitung bekannt ist: Man verwendet verschiedene *Filter*, um entsprechende Informationen aus einem Signal (bspw. Pixelwerte in den Zeilen und Spalten eines Bildes) zu extrahieren. Mallat und Meyer schafften es jedoch als erste eine Verbindung zwischen den Filtern aus der Signalverarbeitung und der Wavelet-Theorie aufzuzeigen. Dieser Zusammenhang wird in der sogenannten *Mehrfachauflösung* (*Multiresolution Analysis: MRA*) beschrieben, welche nachfolgend in Bezug auf die Anwendung im Bereich der Bildverarbeitung dargestellt werden soll. Zur genauen Definition der MRA siehe auch [BUR97, S. 175ff].

Der Ansatz der MRA wird von der psychooptischen Vorstellung motiviert, daß das Gehirn Bilder gleichzeitig in mehreren Auflösungen verarbeitet [RAU99, Kap. 3.1]. Die Umsetzung dieser Erkenntnis erfolgt nun dadurch, daß man mittels einer *Skalierungsfunktion* ein Bild in verschiedenen Auflösungen approximiert. Die Realisierung der Skalierungsfunktion erfolgt durch einen *Tiefpaß-Filter*, welcher nur entsprechend tiefe Frequenzen eines Signals erfaßt. Anschaulich kann man sich die Aufgabe der Skalierungsfunktion (bzw. des zugehörigen Tiefpaß-Filters) als Mittelwertbildung von benachbarten Pixeln vorstellen. Die bei einer bestimmten Auflösung durch Anwendung der Skalierungsfunktion wegfallenden Details (also hohe Frequenzen) werden durch geeignete Wavelets aus einer Waveletfamilie beschrieben. Die technische Realisierung erfolgt hier durch die Verwendung eines entsprechenden *Hochpaß-Filters*, welcher nur hohe Frequenzen erfaßt. Die sich nach der Hochpaß-Filterung ergebenden Werte werden analog zur CWT/DWT als *Wavelet-Koeffizienten* bezeichnet.

Nach erfolgter Filterung wird das Signal (also jew. Hoch- und Tiefpaß gefilterter Anteil) einem *Subsampling*¹ Prozeß unterzogen bei dem es um den Faktor 2 dezimiert wird [BUR97, S. 190]. Übertragen auf die Skalierungsfunktion würde dies durch entsprechendes Strecken der Funktion abgebildet. Anschließend wird das gefilterte Signal in einer tieferen Auflösungsstufe bearbeitet. Als Eingangsgröße hierbei dient der dezimierte Tiefpaß-Anteil des Signals.

Der Prozeß der Hoch- und Tiefpaß-Filterung mit anschließendem Subsampling wird entsprechend mit immer weiter abnehmender Auflösung fortgeführt, bis schließlich das Signal kleiner als die Länge ist Filters ist. Der letzte Tiefpaß entspricht dann dem mittleren Farbwert des analysierten Bildes².

Auch bei der Bildanalyse mit der FWT unter Verwendung geeigneter Filter, gehen keine Informationen verloren. Das Ursprungsbild wird sukzessive in *Approximationen* und *Details* einer bestimmten Auflösung zerlegt.

¹ Beim Subsampling wird jeder 2. Wert des Ausgangssignals entfernt.

² Dies ist natürlich nur dann der Fall, wenn Länge des letztmalig Tiefpaß gefilterten Signals, 1 Pixel beträgt, was eine Filterlänge von 2 voraussetzt. Ist der verwendete Filter länger, so besteht der letzte Tiefpaß aus entsprechend vielen gemittelten Farbwerten des Bildes. Siehe hierzu auch die empirischen Ergebnisse in Kapitel 3.3.2.

Abbildung 9 zeigt den Ablauf der schnellen Wavelet-Transformation. Hierbei bezeichnet $x[n]$ das Eingangssignal und h_1 bzw. h_0 den verwendeten (Analyse-) Hochpaß- bzw. (Analyse-) Tiefpaß-Filter. Man sieht hier wie sich nach dem Subsampling des Hochpasses die Wavelet-Koeffizienten der jew. Transformationsstufe ergeben, welche anschließend abgespeichert werden.

Durch Anwendung der FWT auf ein Bild erhält man lediglich eine transformierte Darstellung. In den meisten Fällen möchte man nach der Transformation (und einer evt. Quantisierung) anschließend das Ursprungsbild wieder rekonstruieren. Dies erfolgt in der sogenannten *Synthese*, bei welcher ebenfalls geeignete Hoch- und Tiefpaß-Filter verwendet werden. Die Synthesefilter unterscheiden sich von denen der Analyse, können jedoch aus letzteren berechnet werden. Hierzu mehr in Kapitel 2.5.

Abbildung 10 zeigt die Durchführung der Synthese. Ausgangspunkt hierfür ist der letzte Tiefpaß aus der Analyse sowie die zugehörigen Wavelet-Koeffizienten, welche aus der letzten Hochpaß-Filterung entstanden sind. Die Hoch- und Tiefpaß-Anteile werden anschließend jeweils überabgetastet¹ (*Upsampling*) und mit den zugehörigen Synthese Hoch- bzw. Tiefpaß-Filtern (f_1 bzw. f_0) gefiltert. Die Addition der Ergebnisse aus der Filterung ergibt den rekonstruierten Tiefpaß-Anteil der nächst höheren² Auflösungsstufe. Dieser Prozeß wird entsprechend rekursiv fortgesetzt, bis das Ursprungsbild vollständig rekonstruiert wurde.

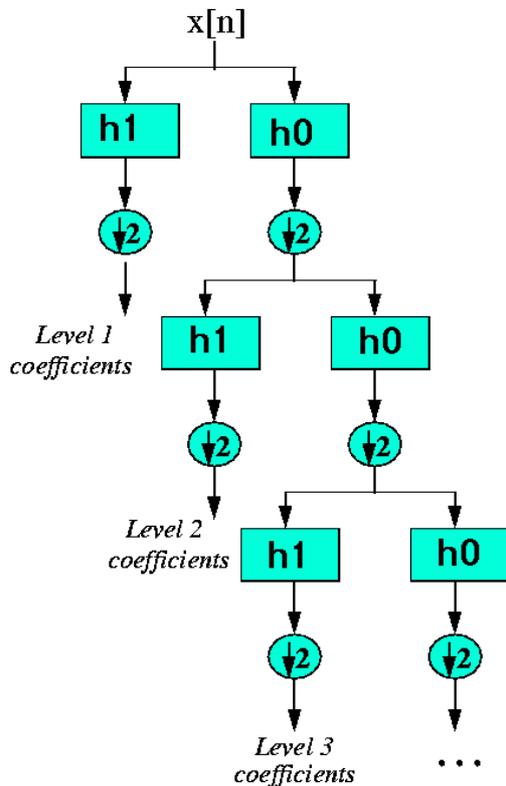


Abbildung 9: Ablauf der FWT (Analyse)
(modifiziert übernommen aus: [POL95, Kap. 4])

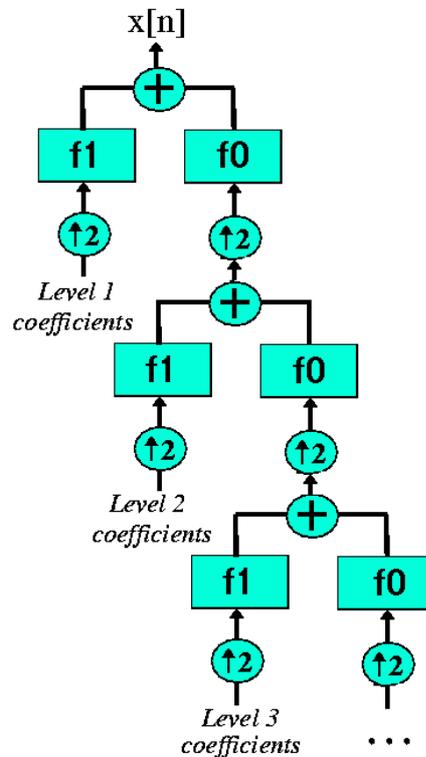


Abbildung 10: Ablauf der FWT (Synthese)
(modifiziert übernommen aus: [POL95, Kap. 4])

¹ Hierbei wird das Signal durch Einfügen von Null man nach jedem Wert künstlich verlängert.

² „höhere Auflösungsstufe“ meint hier: „mit mehr Details“

2.5 Wavelet-Transformation ohne Wavelets: Filter

Wie in Kapitel 2.4 gesehen, braucht man zur Berechnung der Wavelet-Transformation eines Bildes eigentlich gar nichts von Wavelets oder Skalierungsfunktionen zu wissen. Die komplette Transformation besteht lediglich aus der sogenannten **Faltung** (*Convolution*) des Ausgangssignals mit geeigneten¹ Filtern. Ein **Filter** h stellt dabei einen Vektor von reellen Zahlen (den sogenannten *Filter-Koeffizienten*) mit N Einträgen dar.

Die Faltung von Signal und Filter ist mathematisch definiert als:

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i)$$

x : Eingangssignal
 y : Ausgangssignal (nach der Faltung)
 h : verwendeter Filter
 N : Länge des Filters
 n : Signal Position

Gleichung 5: Faltung [BOE00, S. 26]

Man sieht also, daß bei der Faltung in Abhängigkeit von der Filterlänge, entsprechend viele Signalwerte mit den Filter-Koeffizienten „gewichtet“ bzw. multipliziert werden. Dieser gewichtete Wert ist dann der neue transformierte Signalwert, welcher je nach verwendetem Filter die Signalwerte annähernd mittelt (Approximation), oder deren Abweichung (Details) beschreibt. Durch weitere Verschiebung der Position n im Ausgangssignal, ergibt sich somit die vollständige Signaltransformation auf einer Auflösungsstufe.

Welcher Zusammenhang besteht aber nun zwischen Wavelets und Skalierungsfunktionen auf der einen und Filtern auf der anderen Seite? Wie bereits gesehen, sind bei der FWT die Wavelets und Skalierungsfunktionen nicht explizit gegeben. Bei der Faltung des Signals mit den Filtern ergeben sie sich jedoch als Grenzwerte von mehreren Iterationen [BUR97, 187]. In der Praxis geht man sogar häufig so vor, daß man lediglich geeignete Hoch- und Tiefpaß-Filter entwerfen muß, um die FWT durchzuführen. Aus den gegebenen Filtern lassen sich dann die zugehörige Wavelet- und Skalierungsfunktion berechnen. Diesen Zusammenhang beschreiben Gleichungen 6 und 7:

Die *Skalierungsfunktion* ϕ wird rekursiv berechnet durch :

$$\phi(t) = 2 \sum_{k=0}^N h_0(k) \phi(2t - k)$$

h_0 : Tiefpaß-Filter,
 N : Länge des Tiefpaß-Filters

Gleichung 6: Berechnung der Skalierungsfunktion [STN96, S. 22]

Die *Wavelet-Funktion* ω kann aus dem Hochpaß-Filter und der in Gleichung 6 berechneten Skalierungsfunktion ϕ folgendermaßen bestimmt werden:

$$\omega(t) = 2 \sum_{k=0}^N h_1(k) \phi(2t - k)$$

h_1 : Hochpaß-Filter,
 N : Länge des Hochpaß-Filters

Gleichung 7: Berechnung der Wavelet-Funktion [STN96, S. 24]

¹ Beim Entwurf von Filtern, mit welchen die Wavelet-Transformation berechnet werden kann, gibt es zahlreiche Bedingungen, denen die Filter genügen müssen. Welche Bedingungen dies sind und wie man beim Filter-Design vorgeht, kann u.a. in [STN96, S. 53ff und 107ff] nachgelesen werden.

Die Menge der in der Analyse und Synthese verwendeten Hoch- und Tiefpaß-Filter bezeichnet man auch als **Filterbank**. Da die Synthesefilter f_0 und f_1 die Analyse wieder rückgängig machen sollen, kann man diese aus den zugehörigen Analysefiltern h_0 und h_1 gemäß dem in Abbildung 11 gezeigten **Alternating flip pattern** berechnen (siehe auch [STN96, S. 110]). Ferner ist es ebenfalls möglich, den in der Analyse verwendeten Hochpaß-Filter h_1 auf Basis des gewählten Tiefpaß-Filters h_0 zu bestimmen, da die Wirkungsweise beider Filter bezüglich der Frequenzfassung komplementär sein soll. Es genügt daher die Definition eines geeigneten Tiefpaß-Filters.

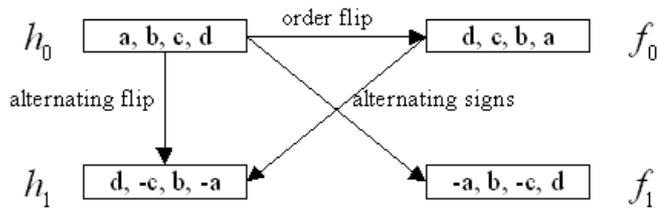


Abbildung 11: Alternating flip pattern [STN96, 110]

- Order flip:* Umkehrung der Reihenfolge der Filter-Koeffizienten
- Alternating signs:* Abwechselnde Negation der Filter-Koeffizienten
- Alternating flip:* Umkehrung der Reihenfolge und abwechselnde Negation

Zum Abschluß des Kapitels über Filter zeigt Abbildung 12 in Analogie zu Abbildung 5 die Haar und Daubechies 4 Filterbänke.

Haar: (Filterlänge = 2)

h_0 :	0.70710678118655	0.70710678118655
h_1 :	0.70710678118655	-0.70710678118655
f_0 :	0.70710678118655	0.70710678118655
f_1 :	0.70710678118655	0.70710678118655

Daubechies 4: (Filterlänge = 8)

h_0 :	0.23037781330886	0.71484657055254	0.63088076792959	-0.02798376941698
	-0.18703481171888	0.03084138183599	0.03288301166698	-0.010597401785
h_1 :	-0.010597401785	-0.03288301166698	0.03084138183599	0.18703481171888
	-0.02798376941698	-0.63088076792959	0.71484657055254	-0.23037781330886
f_0 :	-0.010597401785	0.03288301166698	0.03084138183599	-0.18703481171888
	-0.02798376941698	0.63088076792959	0.71484657055254	0.23037781330886
f_1 :	-0.23037781330886	0.71484657055254	-0.63088076792959	-0.02798376941698
	0.18703481171888	0.03084138183599	-0.03288301166698	-0.010597401785

Abbildung 12: Haar und Daubechies 4 Filterbänke

3 Wavelet-Transformation Demo Applet

Das zu dieser Studienarbeit gehörige englischsprachige Java-Applet demonstriert die Transformation eines Standbildes mittels der schnellen Wavelet Transformation (FWT). Nach der Analyse kann wahlweise eine Datenreduktion (Quantisierung) der Bildinformationen vorgenommen werden. Auf dieser Basis wird anschließend die Synthese des Bildes durchgeführt.

Dieses Kapitel gliedert sich in drei Unterkapitel. In 3.1 werden zunächst der Aufbau des Applets und die Parameter der Transformation vorgestellt. Kapitel 3.2 befaßt sich mit einigen wichtigen Fragestellungen der Implementation und Kapitel 3.3 stellt schließlich die heuristischen Ergebnisse verschiedener Transformations-Durchführungen dar.

3.1 Aufbau und Ablauf des Applets

Nach Betätigung des **Start Wavelet-Transformation Demo** Buttons erscheinen zwei Fenster: Eines zur Auswahl des Ausgangsbildes und der Analyse- bzw. Synthese-Parameter (siehe Abbildung 13) sowie ein weiteres Fenster zur Darstellung und Steuerung der Transformation (siehe Abbildung 14).

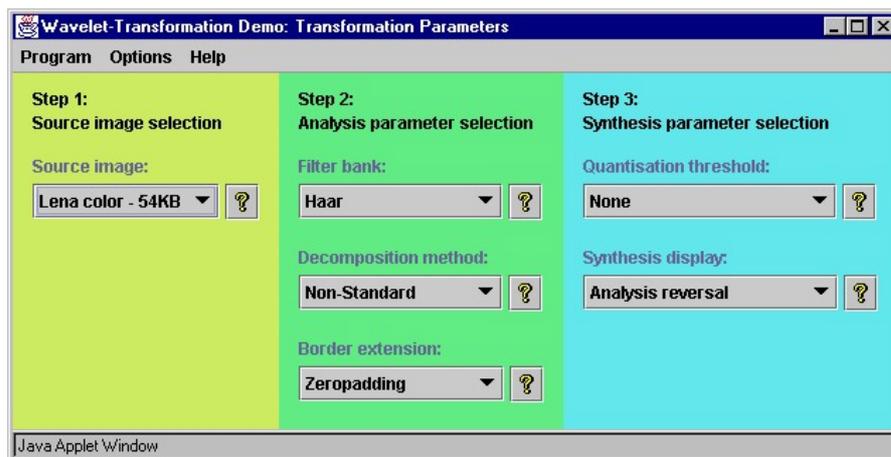


Abbildung 13: Das Parameter-Auswahl Fenster (PAF)

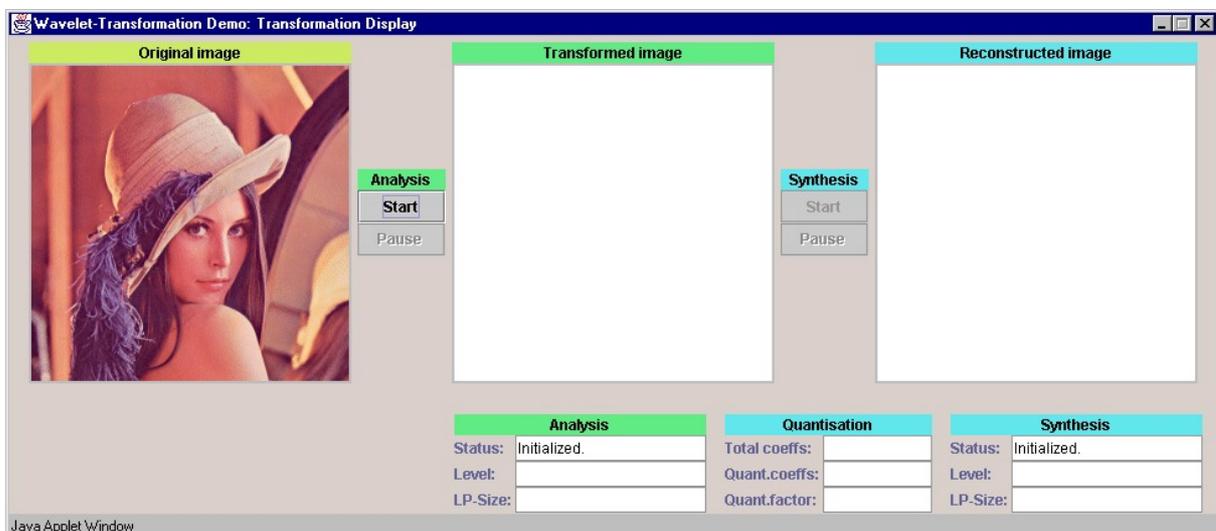


Abbildung 14: Das Transformations-Anzeige Fenster (TAF)

3.1.1 Selektion der Transformations-Parameter

Die Selektion der zur Durchführung der Analyse/Synthese benötigten Parameter, erfolgt im *Parameter-Auswahl Fenster (PAF)*. Zur Vereinfachung sind beim Start des Applets bereits alle Parameter mit einem Standardwert versehen. Neben jeder Parameter-Auswahlbox gibt es einen Hilfebutton, bei dessen Betätigung sich ein neues Fenster öffnet, welches Hilfeinformationen zu dem entsprechenden Parameter anzeigt. Mehr zur Funktion der Hilfeseiten siehe Kapitel 3.1.4.

Die Auswahl der Parameter erfolgt in drei Schritten: Im ersten Schritt beginnt man zunächst damit, ein gewünschtes **Ausgangsbild** (*Source image*) auszuwählen, welches dann im *Transformations-Anzeige Fenster (TAF)* angezeigt wird.

Im zweiten Schritt erfolgt die Auswahl der, bei der Analyse benötigten Parameter. Dies sind im einzelnen:

- Auswahl einer **Filterbank**:

Die Bandbreite der zur Auswahl stehenden Filtern reicht von sehr kurzen Filtern wie den Haar oder Daubechies 2 Filtern, bis zu sehr langen wie dem Daubechies 20 Filter (vgl. hierzu auch Kapitel 2.5).

- Auswahl einer **Dekompositions-Methode** (*Decomposition method*):

Die Wavelet-Transformation ist zunächst nur im eindimensionalen Raum definiert. Die Transformation eines zweidimensionalen Bildes erfolgt daher durch Anwendung der FWT auf die einzelnen Zeilen und Spalten. Dafür gibt es zwei Verfahren:

- Bei der Standard Dekomposition wird die eindimensionale FWT zunächst auf alle Zeilen des Bildes angewendet. Die Tief- und Hochpaß-Filterung ergibt Approximations- und Detailkoeffizienten in den Bildzeilen. Die Transformation der Zeilen wird anschließend rekursiv auf alle Tiefpaß gefilterten Bildhälften angewendet, bis der letzte Tiefpaß-Anteil kleiner ist als die Länge des verwendeten Filters. Nach den Zeilen, erfolgt nun die Transformation der Spalten, die ebenfalls rekursiv auf dem Tiefpaß-Anteil bis Erreichen der Filterlänge abläuft. Abbildung 15 veranschaulicht nochmals diesen Transformationsprozeß.
- Bei der Non-Standard Dekomposition werden die Bildzeilen und -spalten abwechselnd transformiert. Man beginnt mit der Tief- und Hochpaß-Filterung der Zeilen. Dieses Ergebnis wird anschließend wiederum Tief- und Hochpaß gefiltert, nun jedoch für alle Bildspalten. Das transformierte Bild (nach dem ersten Transformationsschritt) besteht nun aus vier Bildteilen, die jeweils doppelt gefiltert wurden:

Links oben: durch Zeilen und Spalten Transformation zwei mal Tiefpaß gefiltert

Rechts oben: durch Zeilen Transformation Hochpaß gefiltert, dann durch Spalten Transformation Tiefpaß gefiltert

Links unten: durch Zeilen Transformation Tiefpaß gefiltert, dann durch Spalten Transformation Hochpaß gefiltert

Rechts unten: durch Zeilen und Spalten Transformation zwei mal Hochpaß gefiltert

Die weiteren Transformationsschritte erfolgen analog jeweils rekursiv auf dem doppelt Tiefpaß gefilterten Anteil, bis zum Erreichen der Filterlänge. Abbildung 16 zeigt die Non-Standard Dekomposition an einem Beispiel.

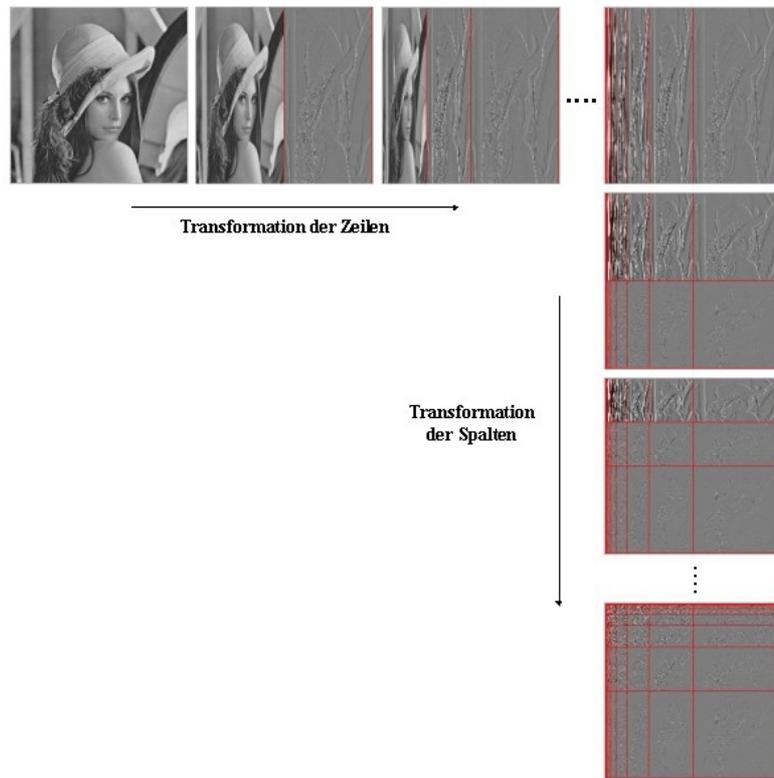


Abbildung 15: Standard Dekomposition

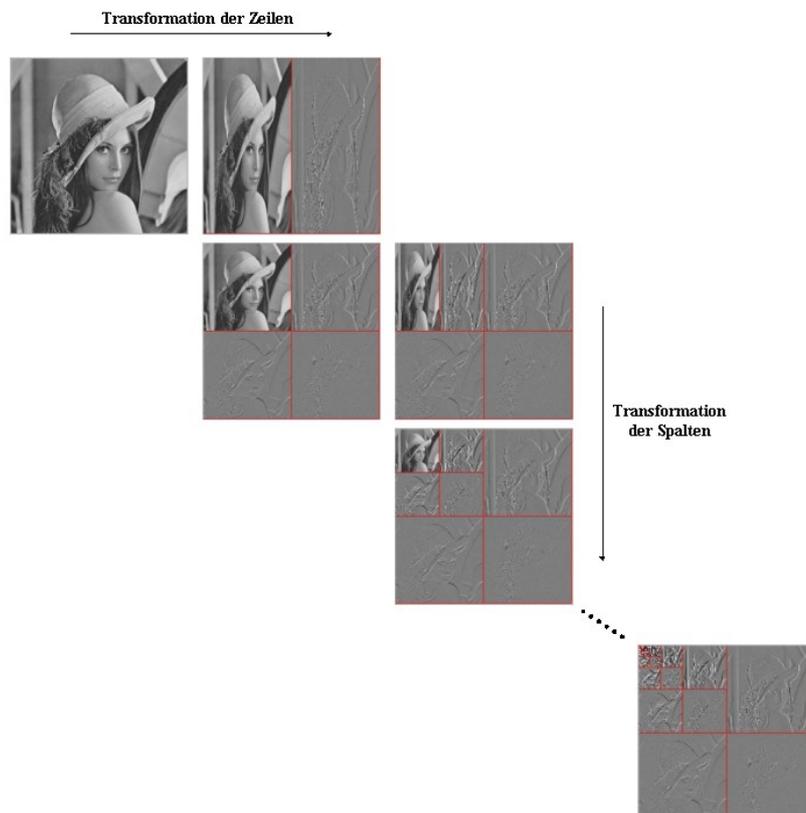


Abbildung 16: Non-Standard Dekomposition

- Auswahl der **Rand-Extensions Methode** (*Border extension*):

Die Faltung gemäß Gleichung 5 ist eigentlich nur für ein unendliches Ausgangssignals definiert. Bei der Anwendung in der Bildverarbeitung, mit einer endlichen Anzahl von Pixelwerten in einer Zeile/Spalte, kommt es zu sogenannten **Randproblemen**.

Dies bedeutet: Bei der Faltung wird jeder Pixelwert in einer Bildzeile oder -spalte aufgrund der Verschiebung des Filters mehrmals (gleich oft) gefaltet, was jedoch nicht für die Zeilen- bzw. Spalten-Randpunkte gilt. Würde nämlich die Faltung beim ersten Pixel beginnen, so würden, in Abhängigkeit von der Länge des gewählten Filters, die ersten Pixelwerte weniger häufig gefaltet werden als die nachfolgenden Pixel. Entsprechendes würde auch bei der Faltung der Endpunkte in einer Zeile/Spalte passieren. Die Konsequenz aus der Tatsache, daß nicht alle Pixelwerte gleich oft gefaltet werden, ist, daß bei der Synthese des Bildes, die Randpunkte nicht mehr vollständig rekonstruiert werden können, was sich in Artefakten an den Bildrändern äußert. Um dem entgegenzuwirken, wird in der Praxis das Bildsignal einer Zeile/Spalte an den Rändern entsprechend künstlich verlängert. Diese Verlängerung erfolgt jeweils auf allen Stufen der Analyse. Die Anzahl der Pixel, die an den Rändern einzufügen sind, ist abhängig von der Länge des gewählten Filters – siehe dazu auch Kapitel 3.2.2. Bei der Synthese können nun durch die Signalverlängerung auch die Randpunkte verlustfrei rekonstruiert werden.

Im vorliegenden Applet werden folgende Rand-Extensions Methoden unterstützt:

- Zeropadding: Hierbei erfolgt die Verlängerung des Bildsignals an den Rändern durch Hinzufügen von Nullen.
- Mirrorpadding: Hierbei werden die Signalaränder entsprechend der benötigten Länge gespiegelt.
- Circular-Convolution (Ringschluß): Bei Anwendung des Ringschlusses erfolgt am Signalende ein Umbruch zum Signalanfang. Die Pixelwerte an den Zeilen- bzw. Spaltenrändern werden somit ebenfalls gleich oft, wie alle anderen Pixel, gefaltet.

Im dritten Schritt der Transformations-Parameter Selektion erfolgt schließlich die Auswahl der zur Bildsynthese benötigten Parameter:

- **Festlegung eines Quantisierungs-Schwellenwertes** (*Quantisation threshold*):

Die Quantisierung der Bildinformationen erfolgt nach der Analyse auf Basis der Wavelet-Koeffizienten (vgl. Kapitel 2.4). Hierbei wird für jeden Koeffizienten (außer den doppelt Tiefpaß gefilterten Anteilen) geprüft, ob dieser unterhalb des in der Auswahlbox selektierten Quantisierungs-Schwellenwertes liegt. Ist dies der Fall, so wird der entsprechende Wert auf Null gesetzt. Da die Quantisierung nur auf die Hochpaß gefilterten Bildanteile angewendet wird, werden dadurch entsprechende Details aus dem Bild entfernt. Bei maximalem Quantisierungsfaktor, wenn also alle Details entfernt wurden, bleibt nur noch ein grobes und unscharfes Grundbild übrig.

- **Auswahl der Darstellungsart der Synthese** (*Synthesis display*):
Die folgenden Möglichkeiten beschreiben lediglich verschiedene Anzeigemethoden der Synthese, mit dem Ziel bestimmte Eigenschaften der Transformation besser darstellen zu können. Die Berechnung der Synthese erfolgt bei allen Alternativen gleich.
 - Umkehr der Analyse-Darstellung (*Analysis reversal*): Die Synthese wird je nach gewählter Dekompositions-Methode in Umkehrung der Analyse dargestellt, d.h. es werden alle Hoch- und Tiefpaß gefilterten Bildinformationen angezeigt.
 - Rekonstruktion des Tiefpaß ohne Interpolation (*LP reconstruction w/o interpolation*): Hierbei wird bei jedem Syntheseschritt nur der doppelt gefilterte Tiefpaß Anteil angezeigt. Absicht dieser Darstellungsart ist es, das „Wachsen“ des doppelten Tiefpasses bis zum vollständig rekonstruierten Bild zu verdeutlichen, sowie die Auswirkungen der gewählten Rand-Extensions Methode aufzuzeigen. Diese Darstellungsart ist jedoch nur zur Anzeige der Synthese bei Anwendung der Non-Standard Dekomposition sinnvoll, da diese bei jedem Syntheseschritt den doppelt gefilterten Tiefpaß-Anteil der jew. nächst höheren Stufe wiederherstellt. Bei Verwendung der Standard Dekompositions-Methode ist die Auswahl dieser Darstellungsart nicht möglich.
 - Rekonstruktion des Tiefpaß mit Interpolation (*LP reconstruction with interpolation*): Diese Option verläuft analog zur Rekonstruktion ohne Interpolation (siehe oben). Der aktuell rekonstruierte doppelte Tiefpass wird jedoch auf die gesamte Darstellungsfläche (256x256 Pixel) interpoliert. Folgende Pixel-Interpolationsverfahren werden dabei unterstützt. Zum genaueren Ablauf der Verfahren siehe [SUN99, Kap. 8.2].
 - *Nearest Neighbor* (Interpolation auf Basis des am nächsten liegenden Pixelwertes)
 - *Bilinear* (lineare Interpolation auf Basis von 4 umgebenden Pixelwerten)
 - *Bicubic* (polynomiale Interpolation auf Basis von 16 umgebende Pixelwerten)
 - *Bicubic2* (wie Bicubic, jedoch mit anderen Polynomial-Funktionen)

Die Auswahl der Parameter in den einzelnen Schritten ist jeweils aufeinander aufbauend, d.h. wählt man in Schritt 1 ein Ausgangsbild aus, so dient dieses als Grundlage der Analyse (Schritt 2), sowie der Synthese (Schritt 3). Genauso verhält es sich mit den Parametern in Schritt 2. Nach Auswahl der gewünschten Filterbank, der Dekompositions-Methode und der Rand-Extensions Methode und anschließender Durchführung der Analyse, sind diese Parameter für die nachfolgende Synthese (Schritt 3) bindend. Eine Änderung der Parameter nach erfolgter Analyse in Schritt 2 und vor Durchführung der Synthese hat keinen Einfluß. Erst beim nächsten Start der Analyse werden diese wieder berücksichtigt.

Die einzelnen Schritte zur Parameter Selektion im *PAF* sind farblich voneinander abgetrennt. Diese Trennung stellt die Verbindung zur Darstellung der Transformation im *TAF* her, welches im nächsten Kapitel beschrieben wird.

3.1.2 Anzeige und Steuerung der Transformation

Wie bereits erwähnt, erfolgt im *Transformations-Anzeige Fenster (TAF)* die Anzeige und Steuerung der Analyse bzw. Synthese eines gewählten Bildes. Das *TAF* gliedert sich logisch in drei **Bild-Darstellungsbereiche** (siehe dazu auch Abbildung 14):

- der linke Bereich zur Darstellung des Ausgangsbildes (*Original image*),
- der mittlere Bereich zur Darstellung des transformierten Bildes während der Durchführung der Analyse (*Transformed image*),
- und schließlich der rechte Bereich, in dem die Rekonstruktion (Synthese) des Ausgangsbildes stattfindet (*Reconstructed image*).

Zwischen den Bild-Darstellungsbereichen befinden sich die jeweils zugehörigen **Steuerungs-Buttons** zum Start der Analyse bzw. Synthese mittels des **Start** Buttons oder deren Pausieren mittels des **Pause** Buttons.

Im unteren Drittel des *TAF* befindet sich ein **Informationsbereich**, in dem Informationen zum Status der Transformation usw. angezeigt werden.

Analysis		Quantisation		Synthesis	
Status:	Initialized.	Total coeffs:		Status:	Initialized.
Level:		Quant.coeffs:		Level:	
LP-Size:		Quant.factor:		LP-Size:	

Abbildung 17: Informationsbereich des TAF

Im einzelnen sind dies:

- **Informationen zur Analyse:**
 - *Status*: Initialisierungszustand; aktuelle Richtung der Bildbearbeitung (Transformation der Zeilen bzw. Spalten)
 - *Level*: aktuelle Transformationsstufe
 - *LP-Size*: Größe des aktuellen Tiefpaß gefilterten Bildbereichs
- **Informationen zur gewählten Quantisierung:**
 - *Total coeffs*: Anzahl aller zur Quantisierung untersuchten Wavelet-Koeffizienten, d.h. alle nicht doppelt Tiefpaß gefilterten Werte
 - *Quant.coeffs*: Anzahl der quantisierten Koeffizienten
 - *Quant.factor*: Quantisierungsfaktor, also die Anzahl der quantisierten Koeffizienten dividiert durch die Gesamtzahl der Wavelet-Koeffizienten
- **Informationen zur Synthese:**
 - analog zur Analyse

Im *TAF* sind logisch zusammengehörige Bereiche, jeweils in Übereinstimmung mit den entsprechenden Parameter-Selektions Schritten im *PAF*, unterschiedlich farblich markiert. Dies soll dem Benutzer die gedankliche Zuordnung von Parameterauswahl im *PAF* und deren Auswirkungen in der Transformation, dargestellt im *TAF*, erleichtern.

3.1.3 Die Applet-Menüleiste

Das *Parameter-Auswahl Fenster* beinhaltet die Applet-Menüleiste, welche weitere Optionen zu globalen Programmeinstellungen sowie dem Aufruf von Hilfeseiten bietet.

Hier die wichtigsten Optionen:

- Unter dem Menüpfad *Options* → *Settings* können in einem separaten Fenster layoutspezifische Einstellungen der Transformation vorgenommen werden (siehe Abbildung 18):
 - Wahl der Trennlinienfarbe zwischen Hoch- und Tiefpaß-Anteilen bei der Analyse/Synthese (*Separator line color*)
 - Normalisierung¹ der Anzeige bei der Analyse/Synthese (*Display normalization*)
 - Wahl der Wartezeit zwischen den einzelnen Transformationsstufen bei der Anzeige der Analyse bzw. Synthese (*Analysis/Synthesis display rate*)
- Der Menüpfad *Help* → *Index* verzweigt zur Startseite des Hilfeindex zu diesem Applet
- Der Menüpfad *Help* → *API-Specification* verzweigt zur Anzeige der API Spezifikation des Applets



Abbildung 18: Settings Fenster

3.1.4 Das Hilfe-Fenster

Wie bereits in Kapitel 3.1.1 erwähnt, kann sich der Benutzer zu jedem Parameter im *PAF* durch Drücken des zugehörigen  Buttons entsprechende Hilfeseiten in einem separaten Fenster anzeigen lassen. Alle Hilfeseiten wurden im HTML-Format geschrieben, welches sowohl eine bessere Gestaltung des Textes, als auch eine Verknüpfung der einzelnen Dokumente untereinander, gegenüber einfachen unformatierten Textseiten, erlaubt.

Die Dokumente sind in einer Baumstruktur miteinander verknüpft. Die Wurzel des Baumes kann über den Menüeintrag *Help* → *Index* erreicht werden. Alle Hilfeseiten zu den Parametern im *PAF* sind entsprechende Knoten oder Blätter im Baum. Die Navigation innerhalb des Baumes erfolgt, wie in HTML üblich, über Hyperlinks.

¹ Bei der Transformation werden, mit den verwendeten Filtern, die Beträge der transformierten Werte immer größer. Zur sinnvollen Darstellung müssen diese in den Wertebereich zwischen 0 und 255 normalisiert werden. Siehe dazu auch Kapitel 3.2.4.1.

3.2 Implementierung des Applets

Die nachfolgenden Kapitel beschäftigen sich mit einigen wichtigen Fragestellungen der Implementierung der FWT in dieser Arbeit. Hierbei soll neben der Umsetzung der Analyse bzw. Synthese, insbesondere auch auf Probleme bei Anzeige der Transformation eingegangen werden. Grundsätzliche Kenntnisse der Programmiersprache Java werden an einigen Stellen vorausgesetzt.

3.2.1 Wie werden die einzelnen Filter berechnet ?

Das ist relativ einfach: Zu Beginn wird für jede Filterbank lediglich der Analyse Tiefpaß-Filter als ein konstantes Array mit `double` Werten definiert. Alle anderen Filter einer Filterbank werden durch die in Kapitel 2.5 beschriebenen Methoden „*order flip*“ und „*alternating flip*“ berechnet. So ergibt sich bei gegebenem `analysisLowPass` Filter:

```
//create the synthesis low pass by using the order flip pattern
synthesisLowPass = orderFlip(analysisLowPass);
//create the analysis high pass by using the alternating flip pattern
analysisHighPass = alternatingFlip(analysisLowPass);
//create the synthesis high pass by using the order flip pattern
synthesisHighPass = orderFlip(analysisHighPass);
```

3.2.2 Wie läuft die Analyse ab ?

Die Pixelwerte des Ausgangsbildes werden in einem 3-dimensionalen `double` Array gespeichert, wobei die 1. Dimension den Farbbandindex, die 2. den Zeilenindex und die 3. Dimension den Spaltenindex angibt. Je nach gewählter Dekompositions-Methode werden die Zeilen und Spalten des Bildes in der entsprechenden Reihenfolge durchlaufen. In der Implementation werden auch die Spalten des Bildes, durch Vertauschung von Zeilen und Spalten, als Zeilen aufgefaßt. Dies ermöglicht die Übergabe einer Referenz auf die gerade betrachtete Zeile an die Methoden zur Randextension und Faltung.

Bei der **Extension der Zeilenränder** werden bei den Verfahren *Zeropadding* und *Mirrorpadding* jeweils $N-2$ Werte am Zeilenanfang und -ende angehängt, wobei N die Länge des gewählten Filters ist. Dadurch ist gewährleistet, daß alle Pixelwerte in einer Zeile gleich oft gefaltet werden und dementsprechend die Randpunkte bei der Synthese verlustfrei rekonstruiert werden können. Beim der Randextension gemäß dem *Ringschluß-Verfahren*, genügt es am Ende der Zeile entsprechend $N-2$ Werte vom Anfang der betreffenden Zeile einzufügen oder umgekehrt. In der Implementierung wurde hier der Ringschluß am Ende der Zeile gewählt.

Die **Faltung** der nun verlängerten Zeile mit dem Filter analog zu Gleichung 5, wurde effizient so implementiert, daß der nachfolgende Schritt des Entfernens jeden 2. Wertes (*Downsampling*) nicht mehr explizit nötig ist. Dies wurde implizit dadurch umgesetzt, daß der Filter jeweils um zwei Positionen in der Zeile weiterverschoben wird, statt nur um eine Position, wie in Gleichung 5 definiert.

3.2.3 *Wie läuft die Synthese ab ?*

Bei der Synthese faltet man jeweils die aus der Analyse entstandenen Approximationen und Details mit dem zugehörigen Tief- und Hochpaß Synthesefiltern. Die Addition beider Faltungsergebnisse ergibt dann die Approximation der nächst niedrigeren Transformationsstufe. Diese und die zugehörigen Details werden ebenfalls wieder gefaltet und anschließend addiert usw., bis schließlich das ursprüngliche Bild rekonstruiert wurde.

Der Faltungs- und Additionsprozeß rekonstruiert also immer den Tiefpaß-Anteil der nächst niedrigeren Transformationsstufe. Das eigentlich nötige Upsampling der Approximationen und Details vor der Faltung, wurde hier ebenfalls wieder implizit durch Inkrementieren des Positionszeigers des Filters um den Wert 2 in die Faltung mit eingebaut. Dadurch, daß beim Upsampling hinter jedem Wert eine 0 eingefügt wird, kann man sich die Multiplikation von Null und dem entsprechenden Index im Filter jeweils sparen.

3.2.4 *Wie wird die Transformation dargestellt und welche Probleme gibt es dabei ?*

Nach jedem Transformationsschritt werden die transformierten Werte, die als 3-dimensionales `double` Array vorliegen in `byte` Werte umgewandelt und anschließend als `RenderedImage` dargestellt. Dabei ergeben sich jedoch folgende Probleme:

3.2.4.1 *Normalisierung*

Die in dieser Arbeit verwendeten Analyse- und Synthese Tiefpaß-Filter (Haar, Daubechies 2 bis 20) haben die Eigenschaft, daß bei der Faltung eines Zeilenabschnittes mit dem Filter nicht der Mittelwert der betrachteten Pixelwerte berechnet wird, sondern ein Wert oberhalb des Mittelwertes. Die Summe der Filter-Koeffizienten beträgt demnach nicht "1", sondern " $\sqrt{2}$ ".

In Formelschreibweise ausgedrückt:

Es gilt nicht: $\sum_i tp(i) = 1$, sondern: $\sum_i tp(i) = \sqrt{2}$ mit tp als Tiefpaß-Filter der Analyse oder Synthese.

Für die zugehörigen Hochpaß-Filter hp der Analyse bzw. Synthese gilt: $\sum_i hp(i) = 0$.

Dies hat zur Folge, daß nach der Analyse alle Tiefpaß gefilterten Anteile höhere absolute Helligkeitswerte haben, als vorher. Bei der anschließenden Konvertierung in `byte` Werte, würden schließlich sehr viele Werte auf 255 nach oben hin beschränkt und man könnte von dem transformierten Bild kaum noch etwas erkennen. Aus diesem Grund werden alle transformierten Werte vor der Anzeige der Transformation entsprechend **normalisiert**:

- Alle Bildanteile, die mindestens eine Tiefpaß-Filterung durchlaufen haben, werden in Abhängigkeit von der jeweiligen Transformationsstufe mit Potenzen von $\sqrt{2}$ normalisiert.
- Zusätzlich werden alle Hochpaß gefilterten Anteile um 128 Graustufen erhöht, da diese im Bereich um Null liegen und somit eine bessere Visualisierung der Details ermöglicht werden kann.

Im Menüpfad des Applets befindet sich unter *Program* → *Settings* der Eintrag „*Display normalisation*“. Wählt man für diesen die Option „*off*“, so kann man die Transformation auch ohne Normalisierung der Darstellung durchführen.

3.2.4.2 Bildvergrößerung

Die Größe der in diesem Applet verwendeten Bilder beträgt 256x256 Pixel. Die Rand-Extensions Verfahren *Zeropadding* und *Mirrorpadding* haben die Eigenschaft, daß in jedem Analyseschritt die resultierenden Teilbilder größer werden. Je länger dabei die verwendeten Filter sind, desto größer wird das transformierte Bildausschnitt. Bei Verwendung des *Ringschluß*-Verfahrens tritt dieses Problem nicht auf, da hierbei jede Zeile nur um die halbe Menge an Werten verlängert wird und somit die Transformation gleich viele Werte liefert wie das Ausgangsbild. Zum besseren Verständnis der Problematik hier ein Beispiel mit Zahlen:

Bsp.: Analyse eines 256x256 Pixel großen Bildes zum einen mit der *Haar*- und zum anderen mit der *Daubechies 20* Filterbank. Als Dekompositions-Methode wird die *Non-Standard Dekomposition* eingesetzt, sowie die Bildränder gemäß dem *Zeropadding* Verfahren verlängert. Die folgende Tabelle zeigt jeweils die Größe des doppelt Tiefpaß gefilterten Bildviertels (linkes oberes Viertel) in jeder Transformationsstufe.

Level	Größe des oberen Bildviertels	
	<i>Haar</i>	<i>Daubechies 20</i>
1	128x128	147x147
2	64x64	93x93
3	32x32	66x66
4	16x16	52x52
5	8x8	45x45
6	4x4	42x42
7	2x2	40x40
8	1x1	39x39

Beim Haar-Filter (Filterlänge $N=2$) wird jeder Rand einer Bildzeile mit $N-2 = 2-2 = 0$ Pixel verlängert – bleibt also unverändert. Durch das Verschieben des Haar-Filters um jeweils 2 Positionen, wird jeder Pixelwert auch ohne Ergänzung von Randwerten entsprechend nur einmal gefaltet. Die Bedingung, daß alle Werte in einer Zeile gleich oft gefaltet werden ist also erfüllt. Bei der Transformation halbiert sich somit auf jeder Stufe die Zeilen- und Spaltenanzahl der Bild-Approximation. Dieser Prozeß wird solange fortgeführt, bis die Länge der zu faltenden Zeile kleiner ist, als der Filter.

Im Vergleich dazu wird unter Verwendung des *Daubechies 20* Filters ($N=40$) jeder Rand mit $40-2 = 38$ Pixel, insgesamt also 76 Pixel pro Zeile, verlängert. Dies hat eine Vergrößerung der Approximation in jedem Transformationsschritt zur Folge. Große Teile der Approximationen bestehen jedoch aus den künstlich verlängerten Werten und nur ein kleiner Teil aus dem „geschrumpften“ Ausgangsbild.

Die linke Seite der Abbildung 19 zeigt hierzu (vergrößert) das Aussehen der Bild-Approximation nach der 4. Transformationsstufe bei Verwendung des *Daubechies 20* Filters, der *Non-Standard Dekomposition* und des *Zeropadding*-Verfahrens. Aufgrund der Ergänzung

der Bildränder mit Nullen, sind große Teile des Approximations-Bildes schwarz. Analog zur obigen Tabelle wäre dessen Gesamtgröße 52x52 Pixel. Das eigentliche Bild in der rechten unteren Ecke dürfte allerdings nur ca. 16x16 Pixel groß sein.

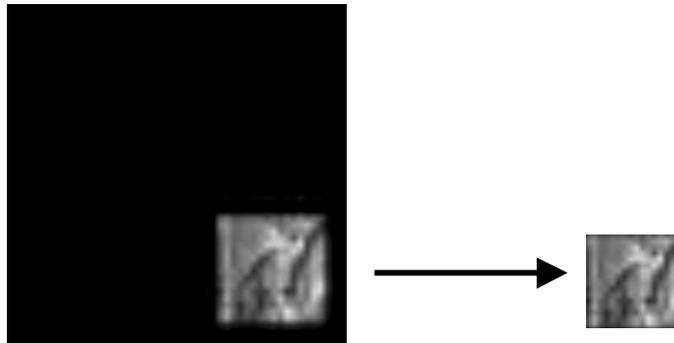


Abbildung 19: Trimmen der Approximation beim Zeropadding Verfahren

Was macht man also nun, um auch solche „wachsenden“ Transformationen auf einer Gesamtfläche von 256x256 Pixeln darzustellen? In der Implementierung wird versucht, daß „geschrumpfte“ Bild zu finden und anschließend den umgebenden schwarzen Rand „abzuschneiden“, um dieses als Ergebnis eines Transformationsschrittes darstellen zu können¹. Dieser Prozeß ist in Abbildung 19 auf der rechten Seite dargestellt.

Diese Problematik tritt nicht nur bei der Darstellung der Approximationen auf, auch bei den Details – also den Hochpaß gefilterten Anteilen – müssen zur sinnvollen Darstellung die Ränder entsprechend getrimmt werden.

Bei Verwendung des *Mirrorpadding* als Rand-Extensions Verfahren treten ebenfalls solche Probleme auf. Abbildung 20 zeigt die gleiche Situation wie Abbildung 19, nur mit dem Unterschied, daß die schwarzen Bereiche des Zeropadding hier mehrmals hintereinander gespiegelten Bildbereichen entsprechen. In der Implementierung bereitet dies noch mehr Probleme als schon beim Zeropadding Verfahren. Zur Vereinfachung wird daher einfach ein, der jeweiligen Transformationsstufe in der Größe entsprechendes Teilbild, aus der Mitte des transformierten Bildes „herausgeschnitten“. Dies gilt sowohl für die Darstellung der Approximationen als auch der Details.



Abbildung 20: Mirrorpadding Verfahren

¹ Bei sehr langen Filtern und in hohen Transformationsstufen wird es recht schwierig, daß „geschrumpfte“ Bild zu finden, da es dann sehr klein ist. Aus diesem Grund kann es bei der Transformation zu Darstellungsfehlern kommen. Dies gilt bei Anwendung des Zeropadding als auch des Mirrorpadding Verfahrens.

3.3 Heuristische Ergebnisse

In diesem Kapitel sollen einige interessante Durchführungen der Analyse bzw. Synthese und deren Ergebnisse an Beispielen dargestellt werden. Dabei wird nicht jede Transformation mit einem Screenshot dokumentiert, sondern teilweise „nur“ mit Worten erläutert. Der interessierte Leser kann (und soll) die jeweiligen Transformations-Parameter der Beispiele im Applet selbst auswählen und sich das Ergebnis ansehen. Die Erklärung dazu findet sich dann in diesem Text.

3.3.1 Beispiel 1: Horizontale und vertikale Filterung

<i>Source image:</i>	Wafer
<i>Filter bank:</i>	Haar
<i>Decomposition method:</i>	Non-Standard
<i>Border extension:</i>	Mirrorpadding

An diesem Beispiel kann man recht gut erkennen, welche Informationen aus dem Originalbild durch Anwendung der Filter entfernt werden. In Abbildung 21 sind die einzelnen Bildviertel der Transformation folgendermaßen bezeichnet:

- LL (Low/Low): Tiefpaß-Filterung für Zeilen und Spalten
- LH (Low/High): zuerst Zeilen Tiefpaß-Filterung, dann Spalten Hochpaß-Filterung
- HL (High/Low): zuerst Zeilen Hochpaß-Filterung, dann Spalten Tiefpaß-Filterung
- HH (High/High): Hochpaß-Filterung für Zeilen und Spalten

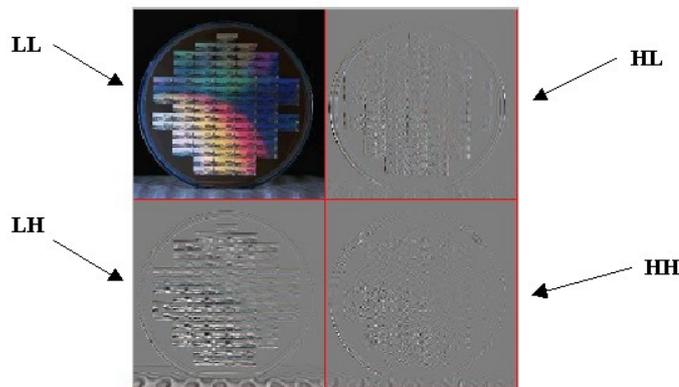


Abbildung 21: Bildviertel bei der Non-Standard Dekomposition

Das Ausgangsbild *Wafer* ist durch horizontale und vertikale Linienverläufe gekennzeichnet. Beide Mischfälle der Filterung (HL und LH) liefern jew. komplementäre Ergebnisse. Der HL-gefilterte Bildbereich zeigt v.a. die vertikalen Linienverläufe, horizontale Linien werden dagegen „weggefiltert“. Im Gegensatz dazu betont das LH-gefilterte Bildviertel sehr stark die auftretenden horizontalen Linien – hier werden entsprechend die vertikalen Verläufe entfernt.

Der HH-gefilterte Bereich umfaßt die übrigbleibenden Details sowohl in horizontaler als auch in vertikaler Richtung.

In der Synthese, kann nun durch „Verknüpfung“ der Bildviertel gemäß der Non-Standard (De-)Komposition, das Ursprungsbild wieder vollständig rekonstruiert werden.

3.3.2 Beispiel 2: Einfluß der Filterlänge beim Ringschluß-Verfahren

<i>Source image:</i>	beliebig
<i>Filter bank:</i>	Haar und Daubechies 20
<i>Decomposition method:</i>	beliebig
<i>Border extension:</i>	Circular convolution

Wenn man mit den obigen Parametern die Analyse durchführt, fällt auf, daß unter Verwendung des Haar-Filters, das Ausgangsbild insgesamt 8 Transformationsstufen durchläuft und der letzte Tiefpaß-Anteil eine Größe von 1x1 Pixeln hat. Im Gegensatz dazu werden beim Daubechies 20 Filter jedoch nur 3 Transformationsstufen mit einer Endgröße der letzten Tiefpasses von 32x32 Pixeln.

Dies ist dadurch zu erklären, daß beim Ringschluß-Verfahren bei der Analyse gleich viele transformierte Werte berechnet werden, wie zuvor Eingangswerte vorhanden waren. Die Anzahl der transformierten Werte halbiert sich also auf jeder Transformationsstufe.

Der Haar-Filter hat die Länge 2. Somit dauert es bei einer Ausgangsbildgröße von 256x256 Pixeln 8 Transformationen, bis der letzte Tiefpaß-Anteil kleiner als die Filterlänge ist. Der Daubechies 20 ist jedoch 40 Werte lang. Dementsprechend ist hier die Transformation schon nach 3 Schritten bzw. einer Länge von 32x32 Pixeln beendet (128→64→32).

3.3.3 Beispiel 3: Quantisierung

Die Quantisierung der Bildinformationen erfolgt auf Basis der Transformation des Ausgangsbildes in der Analyse. Bei der Standard als auch der Non-Standard Dekomposition wird die Quantisierung jeweils auf allen mindestens einmal Hochpaß gefilterten Bildanteilen durchgeführt. Ausschließlich Tiefpaß gefilterte Anteile werden bei der Quantisierung weggelassen, da diese die eigentliche Bild-Approximation beschreiben und dort deshalb keine Informationen entfernt werden dürfen.

Die in der Parameter-Auswahlbox „*Quantisation*“ verfügbaren Werte beschreiben Schwellenwerte für den Vergleich mit den Bildwerten. Ist ein aktuell betrachteter Bildwert kleiner als der ausgewählte Schwellenwert, so wird die entsprechende Bildinformation (Detail) entfernt – der Bildwert also auf Null gesetzt.

Die nachfolgende Tabelle zeigt eine Übersicht von unterschiedlichen Durchführungen der Synthese (inkl. Quantisierung) und deren Auswirkungen auf die subjektive Qualität des rekonstruierten Bildes.

Source image	Filter bank	Decomposition method	Border extension method	Quan. Threshold	Quan. Factor	subjektive Bildqualität
Lena (gray)	Haar	Non-Std.	Zeropadding/ Mirrorpadding/ Circular conv.	1	15 %	keine sichtbaren Qualitätseinbußen
Lena (gray)	Haar	Non-Std.	Zeropadding/ Mirrorpadding/ Circular conv.	10	76 %	leichte Blockartefakte im Bild (v.a. bei gleichfarbigen Bildflächen; bei raschen Helligkeitswechseln dagegen recht gute Qualität)

Lena (gray)	Haar	Non-Std.	Zeropadding/ Mirrorpadding/ Circular conv.	45	95 %	recht starke Blockartefakte (Helligkeitsverläufe werden zu einem Wert gemittelt und als Block dargestellt; schnelle Helligkeitswechsel werden wieder besser rekonstruiert)
Lena (gray)	Haar	Non-Std.	Zeropadding/ Mirrorpadding/ Circular conv.	MAX	100%	Keine Bildumrisse mehr erkennbar; alle Details wurden entfernt, übrig ist nur noch die letzte Approximation des Bildes (hier: Grundfarbwert des Ausgangsbildes)
Lena (gray)	Dau- bechies 20	Non-Std.	Zeropadding	10	83 %	Bessere Bildqualität bei höherem Quantisierungsfaktor im Vergleich zum Haar-Filter; keine Blockartefakte, jedoch leichtes Flimmern im Bild, sowie etwas unschärfer; horizontale und vertikale Streifenartefakte an den Bildrändern
Lena (gray)	Dau- bechies 20	Non-Std.	Mirrorpadding	10	64 %	Gleiche Bildqualität wie Zeropadding, jedoch keine Streifenartefakte an den Bildrändern (in der Bildmitte gibt es keine Unterschiede zwischen Zero- und Mirrorpadding); geringerer Quantisierungsfaktor als Zeropadding
Lena (gray)	Dau- bechies 20	Non-Std.	Circular conv.	10	77 %	Ähnlich zu Mirrorpadding; leicht unscharfes Bild; besserer Quantisierungsfaktor
Lena (gray)	Dau- bechies 20	Non-Std.	Zeropadding/ Mirrorpadding/ Circular conv.	45	zwi- schen 83 % und 96 %	sehr starkes Bild-Flimmern; Bildbereiche verwischen; relativ hoher Quantisierungsfaktor
Lena (gray)	Dau- bechies 20	Non-Std.	Zeropadding/ Mirrorpadding/ Circular conv.	MAX	100 %	Unterschiede zwischen den Rand-Extensions Verfahren treten deutlich hervor: <i>Zeropadding/ Mirrorpadding:</i> Zerlegung des Bildes in allen Transformationsstufen → Rekonstruktion von mittleren Farbwerten bzgl. des Ausgangsbildes <i>Circular conv.:</i> rekonstruiertes Bild entspricht der Approximation auf der 3. Transformationsstufe
Lena (gray)	Daube- chies 20	Non-Std.	Circular conv.	25	91 %	„unruhiges Bild“, starkes Bildflimmern
Lena (gray)	Dau- bechies 5	Non-Std.	Circular conv.	25	91 %	weniger Bildflimmern als bei Daubechies 20; insgesamt „ruhiger“; Guter Kompromiß zwischen Bildqualität und Kompressionsfaktor
Lena (gray)	Dau- bechies 5	Standard	Circular conv.	25	92 %	schlechterer subjektiver Bildeindruck als bei Verwendung der Non-Standard Dekomposition; unschärfer; kleine horizontale und vertikale Streifen im Bild

3.3.4 Abschließende Beurteilung der Bildqualität bei der Synthese (inkl. Quantisierung)

- Sehr kurze Filter (bspw. Haar) erzeugen Blockartefakte im Bild, da in den Faltungs-Prozeß nur wenige Werte einfließen und es somit zu harten Übergängen kommen kann. Sehr lange Filter dagegen (bspw. Daubechies 20) verursachen störende Flimmerartefakte, hervorgerufen dadurch, daß eine Veränderung im Bild auf viele Faltungs-Prozesse Einfluß hat. Eine Helligkeitsänderung „zieht“ sich somit durch das rekonstruierte Bild – dadurch entsteht das Flimmern.
- Die Non-Standard Dekomposition erzeugt bei der Synthese „ruhigere“ Bilder mit weniger Flimmerartefakten im Vergleich zur Standard Dekomposition. Ferner verursacht die zunächst vertikale, dann horizontale Zerlegung des Bildes bei der Standard Dekomposition kleine Streifen im rekonstruierten Bild, welche bei der Non-Standard Dekomposition nicht zu erkennen sind.
- Die Auswahl der Rand-Extensions Methode hat ebenfalls großen Einfluß auf die Bildqualität:
 - Das *Zeropadding-Verfahren* ist nicht zu empfehlen, da es sehr störende horizontale und vertikale Streifen an den Rändern erzeugt, welche durch die Extension der Ränder mit Nullen entstehen. Die Faltung „schleppt“ diese Nullen quasi mit ins Bild und verursacht so die Streifen.
 - Das *Mirrorpadding-Verfahren* kennt diese Probleme nicht. Hier gibt es durch die Spiegelung der Werte an den Rändern keine Probleme mit der Bildqualität. Nachteil jedoch ist, daß der Quantisierungsfaktor im Vergleich zum Zeropadding um einiges niedriger liegt und somit eine schlechtere Kompressionsrate erreicht werden kann. Grund hierfür ist, daß beim Mirrorpadding auch an den verlängerten Rändern nicht alle Werte unter der gewählten Quantisierungsschranke liegen müssen – beim Zeropadding sind dort alle Werte Null und können somit quantisiert werden.
 - Die Rand-Extension gemäß dem *Ringschluß-Verfahren (Circular convolution)* ist wohl für die Praxis am bedeutendsten, da hier ein guter Kompromiß zwischen Bildqualität und Quantisierungsfaktor vorliegt.

4 Die Zukunft der Wavelets

Im letzten Kapitel dieser Arbeit soll abschließend ein kurzer Ausblick auf die (nahe) Zukunft des Wavelet-Verfahrens und dessen standardisierte Anwendung in der Bildverarbeitung erfolgen.

4.1 JPEG2000

Das in den 80er Jahren entwickelte JPEG-Kompressionsverfahren ist mittlerweile in die Jahre gekommen und bietet für heutige Anwendungsgebiete (Digitale Fotografie, Internetübertragung, etc.) nicht mehr optimale Voraussetzungen.

Hauptkritikpunkte am aktuellen JPEG-Standard sind unter anderem die geringe Bildqualität bei mittleren und hohen Kompressionsraten (durch Blockartefakte), der unbefriedigende progressive Bildaufbau v.a. bei Datenübertragung über das Internet, sowie mangelnde Toleranz bei Übertragungsfehlern (bspw. im Online- oder Mobilfunkbereich) [BGH99, S. 184].

Aus diesen Gründen wird seit 1997 unter Federführung der ISO (International Organization for Standardization), der IEC (International Electrotechnical Commission) sowie zahlreichen wissenschaftlichen Einrichtungen und namhafter Hersteller aus der Wirtschaft, an dem Nachfolger für JPEG, dem JPEG2000 Verfahren gearbeitet. Die offizielle Verabschiedung des Standards ist für 2001 geplant.

JPEG2000 basiert im Kern auf der Wavelet-Transformation, da schon recht früh erkannt wurde, daß dieses Verfahren viele der Anforderungen an den Nachfolger von JPEG in idealer Weise erfüllt. Dies sind insbesondere (vgl. [BGH99, S. 185]):

- bessere Bildqualität als JPEG bei mittleren und hohen Kompressionsraten
- wahlweise verlustbehaftete oder verlustfreie Kompression
- progressive Übertragung von Bildinformationen (Vorschaubilder)
- freier Zugriff auf Teilbereiche eines kodierten Bildes (zur Erhöhung der Fehlertoleranz bei Informationsverlusten) → wird durch Blockbildung nach Durchführung der Wavelet-Transformation umgesetzt
- offene Architektur, die Optimierungen auf spezielle Bildklassen erlaubt

Mehr zum Pflichtenheft von JPEG2000 sowie der technischen Umsetzung finden sich u.a. auf der offiziellen JPEG2000 Homepage unter www.jpeg.org/JPEG2000.htm.

Literaturverzeichnis

- ALT96 Altman, Joshua: *Surfing the Wavelets*, 1996
<http://www.wavelet.org/wavelet/tutorial/wavelet.htm>
- BGH99 Berghorn, Wilhelm et al.: *Schlanke Bilder, Winzig mit Wavelets*, c't (Magazin für Computer-Technik), Heise-Verlag, Ausgabe 26/99
- BOE00 Bömers, Florian: *Wavelets in real time digital audio processing: Analysis and sample implementations*, Diplomarbeit, Universität Mannheim, 2000
- BUR97 Burke Hubbard, Barbara: *Wavelets – Die Mathematik der kleinen Wellen*, Birkhäuser Verlag, 1997
- CLE99 Clemens, Michael: *Wavelet Tutorial*, Universität Kaiserslautern, 1999
<http://nt.eit.uni-kl.de/wavelet/index.html>
- EFF00 Effelsberg, Wolfgang: *Vorlesung Multimediatechnik WS00/01*, Universität Mannheim, 2000
<http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/ws200001/multimedia/index.html>
- LUE97 Lüdders, Peter J.: *EEG-Segmentierung mit Hilfe der diskreten Wavelet-Transformation*, Diplomarbeit, TU Berlin, 1997
<http://ntife.ee.tu-berlin.de/personen/bert/research/dwt/W0.htm>
- POL95 Polikar, Robi: *The Wavelet Tutorial*, 1995
<http://www.public.iastate.edu/~rpolikar/WAVELETS/WTpreface.html>
- RAU99 Rauch, Christian: *Bildkompression mit der Wavelet-Transformation*, Lst. für Datenverarbeitung TU München, 1999
<http://home.arcor-online.de/chr.rauch/wavelet/wavelet.htm>
- STO95 Stollnitz, Eric J. et al: *Wavelets for computer graphics – A primer, Part 1*, IEEE Computer Graphics and Applications, 15(3): 76-84, May 1995.
<http://www.cs.washington.edu/research/graphics/projects/wavelets/article/index.html>
- SUN99 Hrsg. Sun Microsystems: *Programming in Java Advanced Imaging*, 1999
<http://java.sun.com/products/java-media/jai/index.html>
- RUS00 Russel, Christopher T.: *ESS 265: Instrumentation, Data Processing and Data Analysis in Space Physics*, UCLA Institute of Geophysics and Planetary Physics
<http://www-ssc.igpp.ucla.edu/personnel/russell/ESS265/Ch8/>