

FOURIER- UND WAVELET-TRANSFORMATIONEN  
VON BILDDATEIEN MIT DEM  
ZIEL DER DATENREDUKTION

**DIPLOMARBEIT**

zum

Diplom-Mathematiker

im Studiengang Mathematische Systemanalyse

des Lehrgebiets  
Numerische Mathematik  
der  
FernUniversität -Gesamthochschule- Hagen

von

Dipl.-Ing. (FH)

Marcus Bäckmann

Wörth a. Main, August 1998

<http://www.baeckmann.de>

Referent: Prof. Dr. Locher

## KURZFASSUNG

Bilddaten gehören zu den Daten, die im Bereich der Computertechnik mit am häufigsten vorkommen. Gerade diese Daten sind durch eine nicht unerhebliche Dateigröße gekennzeichnet, wodurch Speicherung und Datenübertragung wesentlich erschwert werden. Mit Hilfe mathematischer Transformationen sollen in dieser Arbeit Methoden und Hilfsmittel dargelegt werden, die zu einer Reduktion von Bilddateien, insbesondere von Bildern, geeignet sind. Die vorgestellten Verfahren lassen ohne weitergehende Optimierungen eine Reduktion bis zu 90% der ursprünglichen Dateigröße zu, was am Ende dieser Arbeit in zahlreichen Abbildungen belegt wird.

Zunächst werden die Verfahren der diskreten Fourier- und Cosinustransformation im Eindimensionalen vorgestellt und ihre Anwendungsmöglichkeiten für die Datenreduktion aufgezeigt. Durch eine Übertragung auf den zweidimensionalen Bereich wird die Methode für die Bilddatenreduktion anwendbar gemacht. Während heute bereits Fourier- und Cosinustransformationen für diesen Zweck in existierenden Dateiformaten vorhanden sind, wird im Anschluß - von der diskreten Fourier- und Cosinustransformation ausgehend - das Problem behandelt, wie die feste Auflösung dieser Operationen durch eine noch sehr junge Transformation, der sogenannten Wavelettransformation, ersetzt werden kann. Die Wavelettransformation bietet eine variablere Auflösung von Details in Signalen an, so daß ebenfalls die Anwendung von Wavelets für die Datenreduktion von Bildern aufgezeigt wird. Ein Kapitel über die Implementierung der verwendeten Algorithmen und ihren Laufzeitbedarf sowie eine ausführliche Gegenüberstellung der hier vorgestellten Verfahren in zahlreichen Bildern runden das theoretische Fundament ab.

Mit der gefensterter diskreten Cosinustransformation und der Wavelettransformation liegen brauchbare Methoden für eine Datenreduktion vor, wobei man allerdings erkennt, daß die Wavelettransformation wesentlich vielseitiger, aber auch komplizierter zu handhaben ist.

Die beiliegende CD enthält ein Programm für Windows-Systeme, mit dem sich die Verfahren sehr leicht vom Leser nachprüfen lassen, sowie die Quelltexte aller verwendeten Algorithmen.

# INHALTSVERZEICHNIS

1	Notwendigkeit und Bedeutung der Datenreduktion .....	1
2	Mathematische Grundlagen.....	6
2.1	Die diskrete Fouriertransformation.....	6
2.1.1	Überblick über die kontinuierliche Fouriertransformation.....	6
2.1.2	Der eindimensionale diskrete Fall.....	11
2.1.3	Die Inverse der diskreten Fouriertransformierten.....	14
2.1.4	Physikalische Interpretation der DFT .....	15
2.1.5	Idee der Datenreduktion.....	17
2.1.5.1	Tiefpaßfilter .....	22
2.1.5.2	Hochpaßfilter.....	25
2.1.6	Erweiterung auf den zweidimensionalen Fall.....	26
2.1.6.1	Die diskrete Fouriertransformierte einer Matrix .....	27
2.1.6.2	Die inverse diskrete Fouriertransformierte einer Matrix.....	30
2.1.6.3	Tensorierte Filterungen bei Matrizen .....	30
2.2	Die diskrete Cosinustransformation .....	33
2.2.1	Zusammenhang mit der diskreten Fouriertransformation .....	33
2.2.2	Filterungen bei der DCT .....	37
2.2.2.1	Tiefpaßfilter .....	37
2.2.2.2	Hochpaßfilter.....	38
2.2.3	Erweiterung auf den zweidimensionalen Fall.....	40
2.2.3.1	Die Transformation und ihre inverse Transformation .....	40
2.2.3.2	Filterungen der DC-transformierten Matrix .....	41
2.3	Die Fensterung von Fourier- und Cosinustransformationen .....	43
2.3.1	Prinzip und Motivation für gefensterte Transformationen .....	43
2.3.2	Die Fensterung im zweidimensionalen Fall.....	46
2.3.3	Filterung gefensterter Transformation.....	47
2.4	Die diskrete Wavelettransformation.....	47
2.4.1	Unterschiede und Gemeinsamkeiten zur Fouriertransformation.....	47
2.4.2	Der eindimensionale diskrete Fall.....	49
2.4.3	Der zweidimensionale diskrete Fall.....	53
2.4.4	Idee der Datenreduktion.....	54
2.4.5	Anwendung der Datenreduktion im zweidimensionalen Fall .....	56
2.5	Entwicklung eines Reduktionsalgorithmus.....	57
3	Programmtechnische Implementation .....	59
3.1	Die Entwicklungsumgebung.....	59
3.2	Unterstützte Grafikformate .....	59

3.2.1	Das Bitmap-Format (BMP).....	60
3.2.2	Das Graphics Interchange Format (GIF).....	61
3.2.3	Das ZSoft Paintbrush Format (PCX) .....	61
3.2.4	Das Tag Image File Format (TIFF) .....	62
3.2.5	Das Joint Photographic Experts Group Format (JPEG) .....	62
3.2.6	Übersicht über alle unterstützten Grafikformate.....	63
3.3	Bildverarbeitung und Speicherung im Rechner.....	64
3.3.1	Aufteilung eines Farbbildes auf Farbauszüge.....	65
3.3.2	Speicherung und Speicherplatzbedarf von Bildmatrizen .....	66
3.3.3	Die Berechnungs- und Verarbeitungskette .....	67
3.3.4	Modifikation der Farbwertverteilung.....	68
3.4	Die Basis: Implementation der Algorithmen .....	69
3.4.1	Die Idee der schnellen diskreten Fouriertransformation.....	70
3.4.2	Der DCT-Algorithmus.....	71
3.4.3	Der FWT-Algorithmus zur diskreten Wavelet-Transformation.....	72
3.4.4	Die Implementation der Matrix-Transformationen .....	72
3.4.5	Feintuning und Effizienzsteigerung .....	73
3.5	Programmbedienung .....	74
4	Anwendung auf Echtfarbbilder.....	76
4.1	Beurteilung der Reduktionsverfahren.....	76
4.1.1	Die Reduktionsrate.....	76
4.1.2	Die Reduktionsraten der verschiedenen Filter.....	78
4.2	Die Vergleichsbilder und Vergleichsverfahren .....	79
4.3	Untersuchung mit Hilfe der diskreten Fouriertransformation .....	81
4.3.1	Hochpaßfilterungen .....	81
4.3.2	Tiefpaßfilterungen bei verschiedenen Reduktionsfaktoren.....	82
4.3.3	Auswirkung der Fensterung bei verschiedenen Fenstergrößen.....	83
4.4	Untersuchung mit Hilfe der diskreten Cosinustransformation .....	86
4.4.1	Tiefpaßfilterungen .....	86
4.4.2	Auswirkung der Fensterung bei verschiedenen Fenstergrößen.....	87
4.5	Untersuchung mit Hilfe der diskreten Wavelettransformation .....	90
4.5.1	Unterdrückung des Wavelet-Hochpaßanteils.....	90
4.5.2	Auswirkungen der Filterlänge .....	92
4.6	Original oder Fälschung – wie man das Auge täuscht .....	93
5	Ergebnisse.....	96

## DANKSAGUNG UND VORWORT

Großer Dank gilt meiner Freundin Sabine, die mit unendlicher Geduld meine zahlreichen Sitzungen am Schreibtisch und vor dem Monitor ertrug, an Wochenenden, am Abend und an den „freien“ Tagen. Ich hoffe, daß ich mein ihr gegebenes Versprechen, keine dritte Diplomarbeit mehr zu schreiben, halten kann.

Weiterhin gilt der Dank auch meinen Eltern, die mir mit Rat und Tat zur Seite standen, wenn die Arbeit über meinem Kopf zusammenschlug.

Bedanken möchte ich mich auch bei Herrn Prof. Dr. Locher dafür, daß er über die lange Bearbeitungszeit neben meinen beruflichen Pflichten nicht die Geduld verlor und sehr großzügig das Ende und die Auslieferung abwartete.

Eine Entschuldigung geht an alle meine Bekannten und Freunde, die ich mit schlechter Laune und wenig Zeit vernachlässigt habe.

Beim Schreiben habe ich versucht, mich an einen schönen Satz zu halten, den ich erstmals in Harro Heusers großartigem Werk über *Gewöhnliche Differentialgleichungen* gefunden habe:

*„So nahmen wir uns vor, diejenigen, die gerne lesen, zu unterhalten, denen, die mit Eifer auswendig lernen, zu helfen, allen aber, die das Buch auf irgendeine Weise in die Hand bekommen, zu nützen.“ (Zweites Buch Makkabäer, 2,25)*

## ERKLÄRUNG

Hiermit versichere ich, daß ich die vorliegende Diplomarbeit selbständig erarbeitet habe, und daß dabei keine anderen als die angegebenen Quellen und Hilfsmittel verwendet wurden, und daß alle Zitate als solche kenntlich gemacht wurden.

Wörth a. Main, den 10.08.1998

---

Marcus Bäckmann

## ABKÜRZUNGS- UND SYMBOLVERZEICHNIS

### Abkürzungen, mathematische Symbole und Definitionen:

$\mathcal{R}$  Die Menge der reellen Zahlen

$\mathcal{C}$  Die Menge der komplexen Zahlen

$\mathcal{N}$  Die Menge der natürlichen Zahlen

$\mathcal{Z}$  Die Menge der ganzen Zahlen

$j$  Die imaginäre Einheit<sup>1</sup>

$L_1(\mathcal{R})$  Der Raum der komplexwertigen absolut integralen Funktionen:

$$L_1(\mathcal{R}) = \left\{ f : \mathcal{R} \rightarrow \mathcal{C} \mid f \text{ meßbar} \wedge \int_{-\infty}^{\infty} |f(t)| < \infty \right\}$$

$L_2(\mathcal{R})$  Der Raum der komplexwertigen absolut quadratintegralen Funktionen:

$$L_2(\mathcal{R}) = \left\{ f : \mathcal{R} \rightarrow \mathcal{C} \mid f \text{ meßbar} \wedge \int_{-\infty}^{\infty} |f(t)|^2 < \infty \right\}$$

*Hinweis in eigener Sache:* Falls sich in dieser Arbeit in Gleichungen und Definitionen noch Fehler befinden, so hat der Leser immerhin den Trost, daß sich diese Fehler nicht im Programm befinden – denn die Programmierung konnte mehrfach durch die praktische Anwendung auf Herz und Nieren überprüft werden, während sich vielleicht ein falscher Index im gedruckten Teil der Arbeit durchaus hartnäckig festgesetzt haben kann. In diesem Falle wird sich der Autor auf die Ausrede berufen, daß er besser programmieren als schreiben könne.

---

<sup>1</sup> Der Autor bittet um Entschuldigung, daß für die imaginäre Einheit der Buchstabe  $j$  und nicht der Buchstabe  $i$  verwendet wird. Dies liegt daran, daß der Autor seine ersten Schritte auf dem Gebiet der Signalverarbeitung in der Elektrotechnik unternommen hat und ihm daher die Benutzung des Buchstaben  $j$  in Fleisch und Blut übergegangen ist.

## GLOSSAR

**Bild:** Als Bild wird hier eine Bilddatei bezeichnet, die die digitale Darstellung einer sichtbaren Projektion oder Darstellung von Umwelt und Objekten auf einer (rechteckigen) Oberfläche realisiert, s.a. →Zeichnung.

**Bildbereich:** Bei einer Transformation, v.a. Fourier- und Laplacetransformation, die Bezeichnung für den Funktionenraum, dessen Basisvektoren für die Darstellung der transformierten Funktion verwendet werden, siehe auch →Frequenzbereich, →Ortsbereich.

**Frequenzbereich:** siehe auch →Bildbereich, in der Elektrotechnik ist diese Bezeichnung bei der Fouriertransformation üblich, weil die physikalische Einheit in dieser Darstellung eine Frequenz ist.

**Gleichanteil:** In der Signalverarbeitung, vor allem der Elektrotechnik, die Bezeichnung für den gewöhnlichen Mittelwert über alle Werte des Signals, wobei die Mittelwertbildung bei Funktionen über Integration, bei Reihen über eine Summation durchgeführt wird.

**Matrix:** Sind  $m \cdot n$  Ausdrücke in einem rechteckigen Schema von  $m$  Zeilen und  $n$  Spalten angeordnet,

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} .$$

so spricht man von einer Matrix vom Typ  $(m,n)$ . Entgegen dieser üblichen Definition werden in der folgenden Abhandlung die Koeffizienten *nicht* ab 1 numeriert, sondern ab 0.

$$\begin{pmatrix} a_{0,0} & \cdots & a_{0,n-1} \\ \vdots & & \vdots \\ a_{m-1,0} & \cdots & a_{m-1,n-1} \end{pmatrix}$$

Dies hat mehrere Gründe, unter anderem den, daß sich einige physikalische Zusammenhänge besser mit einer Koeffizientennumerierung ab der 0 darstellen lassen. Weiterhin ergeben sich einige programmtechnische Vorteile, da Vektoren und Matrizen bei einer Numerierung ab der 0 effizienter zu implementieren sind. Dies ist ein wichtiger Vorteil, da das Ziel dieser Arbeit letztendlich ein Berechnungsprogramm für den Computer ist, und bei den großen Zahlenmengen, die zu verarbeiten sind, ist jedes eingesparte Quentchen Rechenzeit ein wertvoller Gewinn.

**Ortsfrequenzbereich:** eine Bezeichnung für →Frequenzbereich und →Bildbereich bei einer Transformation, die auf Bilder angewandt wird.

**Zeichnung:** Eine grafische Darstellung, die in erster Linie aus Linien, Rasterungen und Texten besteht, also insbesondere keine Farbverläufe aufweist und keine reale Szene darstellt, z.B. eine technische Zeichnung, oder ein digitalisierter Text, siehe im Gegensatz dazu →Bild.

## 1 Notwendigkeit und Bedeutung der Datenreduktion

Betrachtet man die Welt aus der Sicht der Computeranwender und der Computerprogrammierer, so scheint es wirklich nur zwei Probleme zu geben:

- der Speicher ist zu knapp und
- die Datenübertragung ist zu langsam.

Interessanterweise stehen diese beiden Probleme nun schon seit (Computer-)Urzeiten auf der Tagesordnung. Der Bordrechner der Mondlandefähre Eagle von 1969 hatte einen RAM-Speicher von 4096 Zeichen, in den 80er Jahren war ein Heimcomputer bereits mit stattlichen 65535 Zeichen ausgestattet –und in vielen Wohnungen trifft man inzwischen auf PCs, die über eine Speicherkapazität von über 67 Millionen Zeichen verfügen: 64MB. Scheinbar hat sich die Speicherkapazität auf fast das 20000-fache erhöht.

Wie sieht es bei der Datenübertragung aus? Mitte der 80er Jahre feierte man es als Erfolg, wenn man die Datenrate seines Akkustikkopplers von 150 Baud<sup>2</sup> auf 300 Baud erhöhen konnte, ohne daß Probleme beim Transfer auftraten. Heute gilt ISDN, mit einer Datenrate von 64.000 Bit pro Sekunde<sup>3</sup>, als Standard für den Heimnutzer, die 200-fache Geschwindigkeit. Und ADSL, eine neuer Standard, winkt bereits am Horizont.

Wo sind also die Probleme? Eigentlich sollten aufgrund dieser Zahlen mangelnder Speicherplatz und Geschwindigkeitsprobleme bei Datenübertragungen der Vergangenheit angehören. Doch dem ist leider nicht so.

In der Tat hat sich die Leistungsfähigkeit der Rechner in den letzten Jahren nicht nur vermehrt, sondern sogar potenziert. Unglücklicherweise gilt dies auch für die Ansprüche der Benutzer und Anwender. Man denke an eines der ersten berühmten Computerspiele, das „Ping-Pong“. Ein Tennisspiel, reduziert auf zwei weiße Balken und einen weißen Punkt, die angeblich Spieler und Ball darstellen sollten –sagte die Verpackung des Spiels. Heute dagegen hat eine Tennissimulation eine

---

<sup>2</sup> Baud = Informationseinheiten pro Sekunde. Eine Maßzahl zur Bestimmung der Übertragungsgeschwindigkeit von Daten. Nicht zu verwechseln mit der Angabe von Bits pro Sekunde: mit Baud wird die tatsächliche Übertragungsgeschwindigkeit der Daten gemessen, nicht die physikalische Geschwindigkeit.

<sup>3</sup> Dies entspricht bei entsprechendem Protokoll einer Baudrate von *mehr* als 64000 Baud.

3D-Grafik in Echtzeit, es gilt als Selbstverständlichkeit, daß der animierte Computergegner Gesichtszüge von leibhaftigen Stars trägt –ein ruckelndes Bild wird nicht akzeptiert.

Die gewachsenen Ansprüche der Nutzer machen sich gerade an der Grafik oder der grafischen Repräsentation von Daten oder Zuständen fest. Der Mensch ist ein Augentier und fordert daher von den Systemen immer realistischere Bilder und Präsentationen.

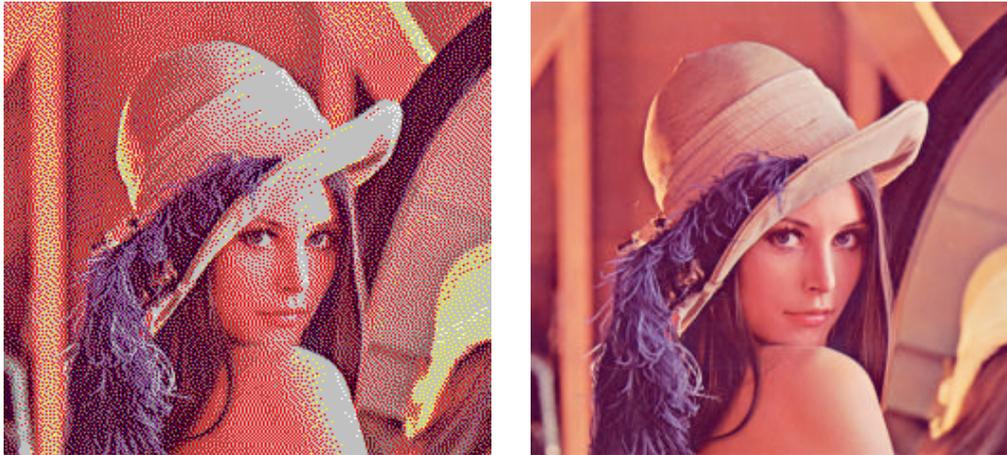


Abbildung 1: Computergrafik einst und jetzt. Während das linke Bild vor 15 Jahren noch leicht einen Grafikwettbewerb gewonnen hätte, zeigt das rechte Bild die heutigen Ansprüche an eine Computergrafik. Die Unterschiede sind deutlich sichtbar.

Kommt man wieder auf die zwei Grundprobleme der Softwareentwicklung zurück, so muß man also etwas entmutigt feststellen, daß der Zuwachs an Speicher und an Datenraten zum gleichen Wachstum der Ansprüche geführt hat –der Gewinn war gleich Null.

Ein anderes Problem tritt noch auf: Geld. In jeder Phase der Computerentwicklung konnte man gegen Geld sein System aus der Masse herausheben. Aber der Massenwunsch war und ist es, ohne zusätzlichen Hardwareaufwand die Leistung zu erhöhen. Auch ist eine Hardwareerweiterung gar nicht immer möglich: man denke an eine der zahlreichen Raumsonden, die gestartet wurden, z.B. die Voyager-Sonden. Die Sonden starteten mit Hardware und Software aus den 70ern, liefert aber bis heute Bilder. Zugleich nimmt die Entfernung zur Erde immer weiter ab, die Dauer für die Übertragung von Bildern nimmt also ständig zu. Die einzige Einflußmöglichkeit, um dieses Problem zu beheben, ist die Veränderung der Software an Bord der Sonde.

Und dies ist auch der Ansatzpunkt dieser Arbeit: es sollen Mittel und Wege untersucht werden, wie man ausschließlich durch Algorithmen die Menge der Daten, die ein Bild repräsentiert, so geschickt vermindert, daß im Idealfall

- keiner etwas davon bemerkt und
- eine spürbare Verkleinerung der Datenmenge erreicht wird.

Diese Vorgehensweise hebt sich tatsächlich von den bekannten Verfahren der Datenkompression ab. Bei einem Kompressionsverfahren wird die eigentliche Information nicht beeinflußt, sondern nur die Art der Darstellung verändert, mit der Hoffnung, daß die neue Darstellung eine geringere Datenmenge zur Folge hat.

Im Gegensatz dazu steht die Datenreduktion, bei der gezielt die Menge der Informationen vermindert wird, ohne daß dies bemerkt wird, oder so, daß die verlorene Information wieder rekonstruiert werden kann. Die folgende Abbildung zeigt dies in einfacher Form für die Zahl  $\pi$  - durch Weglassen von Ziffern am Ende läßt sich die Datenmenge reduzieren, und der Informationsgehalt wird nur unwesentlich verändert. In vielen Anwendungsfällen würde man die Reduktion nicht spürbar bemerken.

3,1415 oder 3,14159265359 ?

Abbildung 2: Ein simples Beispiel für eine Datenreduktion. Die Näherung an die Zahl  $\pi$  nimmt in beiden Fällen unterschiedlich viele Symbole = Speicherstellen ein.

Um das Beispiel eines Bildes wieder aufzugreifen: feine Farbverläufe könnte man durch eine einzige Farbe ersetzen, bei kleinen Änderungen fällt das Auge auf diesen Trick herein. Noch ein letztes simples Beispiel: eine Verkleinerung eines Bildes auf ein Sechzehntel der Größe durch Viertelung der Seitenlängen wäre ebenfalls ein Reduktionsverfahren. Doch wären diese Methoden nicht sonderlich glücklich gewählt, da der aufmerksame Beobachter derartige Tricks recht rasch durchschaut, wie Abbildung 3 zeigt. Hier wurde das Bild durch Datenreduktion um 94% der Punkte reduziert, allerdings macht sich bei diesem Verfahren die Einbuße der Bildinformation als sehr störend bemerkbar, da der „Klötzcheneffekt“ das Auge des Betrachters stark irritiert.



Abbildung 3: Eine einfache Datenreduktion durch Verkleinerung des Originalbildes auf 1/16 der ursprünglichen Größe, wodurch 94% der Bildpunkte wegfallen. Die Rekonstruktion auf die Originalgröße rechts zeigt deutlich sichtbar die Einbußen der Bildinformation.

Wie man sieht, wird man auch auf den Gesichtspunkt der Treue der Informationen zum Original erhebliche Aufmerksamkeit schenken müssen, wenn man ein Verfahren entwickeln will, das auch später akzeptiert wird.

Neben diesem bekannten Verfahren soll hier die Anwendung der Fourier- und Wavelettransformation zur Bilddatenreduktion heraus gearbeitet werden.

Die Grundlage für die hier vorgestellten Methoden bilden die Arbeiten von Jean Baptiste Joseph Fourier, der im Jahre 1768 geboren wurde, wobei sich dieser natürlich noch nicht mit der Bildverarbeitung beschäftigte. Vielmehr beschäftigte sich Fourier mit der Lösung bestimmter Klassen von linearen Differentialgleichungen und kam dabei auf einige Ideen, die er erstmals 1807 in einer Abhandlung und 1822 in seinem Buch „Analytische Theorie der Wärme“ ausgesprochen hatte. Ausgehend von seinen Ideen werden die Theorien auf die Welt der Bilder übertragen, wobei die entsprechenden Verfahren recht junger Natur sind und aus den 50er Jahren dieses Jahrhunderts stammen.



Abbildung 4: Jean Baptiste Joseph Fourier, 1768 - 1830, Mathematiker, Physiker und Ägyptologe

Anschließend werden auch die Grenzen von Fouriers Methoden aufgezeigt und mit Hilfe der Theorie der Wavelets mathematisches Neuland betreten, denn die entsprechenden Methoden basieren auf Verfahren, die überwiegend erst in den letzten 15 Jahren systematisch erforscht wurden. Die Forschung auf diesem Gebiet ist sehr stark mit dem Namen Ingrid Daubechies verbunden, die heute an der Universität Princeton einen Lehrstuhl besitzt ([HUB-97], [PRI-98]).



Abbildung 5: Ingrid Daubechies, Professor für angewandte und numerische Mathematik an der Universität Princeton (geb. 1954)

Die folgende Abbildung zeigt erneut eine Bilddatenreduktion um 94%, die allerdings diesmal mit einem der hier entwickelten Verfahren durchgeführt wurde. Der Unterschied zu Abbildung 3 ist offensichtlich und zeigt, daß der zu gehende Weg ein lohnenswertes Ziel hat.



Abbildung 6: Eine Datenreduktion um 94% durch Anwendung einer diskreten gefensternten Cosinustransformation und Tiefpaßfilterung.

## 2 Mathematische Grundlagen

### 2.1 Die diskrete Fouriertransformation

#### 2.1.1 Überblick über die kontinuierliche Fouriertransformation

Vor allem in den Ingenieurwissenschaften trifft man allenthalben auf den Begriff der Fouriertransformation, der seine Domäne vor allem in der Signalverarbeitung und Systemanalyse hat. Bekanntermaßen definiert man für eine Funktion  $f \in L_1(\mathbb{R})$  eine Funktion  $f^\wedge$ , die als Fouriertransformierte von  $f$  bezeichnet wird:

$$f^\wedge(x) := \int_{-\infty}^{\infty} f(t)e^{-jxt} dt, x \in \mathbb{R} \quad (1)$$

Umgekehrt bezeichnet man eine Funktion  $f^\vee$  als Fourier-Kotransformierte von  $f^\wedge$ .

$$f^\vee(t) := \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x)e^{jxt} dt, t \in \mathbb{R} \quad (2)$$

Man bezeichnet den Definitionsbereich der Funktion  $f$  auch als *Zeitbereich*, den Definitionsbereich der Fouriertransformierten  $f^\wedge$  als *Frequenzbereich* und die Fouriertransformierte als *Spektrum der Funktion*.

Interessant ist dabei die Frage, wann die Fourier-Kotransformierte einer Fouriertransformierten wieder die Ausgangsfunktion darstellt, also wann eigentlich

$$f^{\wedge\vee}(t) = f(t), t \in \mathbb{R} \quad (3)$$

gilt. Umfassendere Abhandlungen dazu finden sich z.B. in [LOC-94], sollen hier aber nicht näher erläutert werden, bis auf die Tatsache, daß folgendes gilt:

Satz (Identitätssatz):

Es sei  $f \in L_2(\mathcal{R})$  beliebig gegeben. Dann gilt für fast alle  $t \in \mathcal{R}$ :

$$f(t) = \frac{1}{2\mathbf{p}} \int_{-\infty}^{\infty} f^{\wedge}(x) e^{jxt} dx \quad (4)$$

Weiterhin gilt für die Funktion  $f$  und ihre Fouriertransformierte  $f^{\wedge}$ :

Satz (Parsevalsche Gleichung):

Sei  $f \in L_2(\mathcal{R})$  beliebig gegeben. Dann gilt:

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\mathbf{p}} \int_{-\infty}^{\infty} |f^{\wedge}(x)|^2 dx \quad (5)$$

(der Beweis hierzu befindet sich z.B. in [LOC-94])

Weiterhin trifft man folgende

Def. (Signalenergie):

Sei  $f \in L_2(\mathcal{R})$  beliebig gegeben. Dann ist  $E$  die Signalenergie von  $f$  mit

$$E = \int_{-\infty}^{\infty} |f(t)|^2 dt \quad (6)$$

Eine physikalische Interpretation dieser beiden letzten Aussagen ist, daß die Energie eines Signals gleich ist, unabhängig davon, ob man es im Zeitbereich oder im Frequenzbereich darstellt. Die Wahl der Darstellung Zeit- oder Frequenzbereich beeinflußt also die Energie des Signals nicht. Allerdings wird bei dieser physikalischen Betrachtung häufig die Voraussetzung  $f \in L_2(\mathcal{R})$  unter-

schlagen – dies ist bei realen Signalen in der Tat der Fall, da reale Signale zwangsläufig eine endliche Energie besitzen und das Energieintegral auch folgerichtig konvergieren muß<sup>4</sup>.

Ein solches Signal, auch *Energiesignal* genannt, enthält eine Information, die sich in der Energieverteilung auf verschiedene Frequenzen ausdrückt. Damit erhält man mit Hilfe der Fouriertransformation ein Hilfsmittel, um den Informationsgehalt eines Signals hinsichtlich der Verteilung über der Frequenz zu untersuchen.

Für eine stetige Funktion  $f$  und ihre Fouriertransformierte gilt insbesondere auch, daß die Darstellung der Funktion durch ihre Transformierte lediglich einen Basiswechsel darstellt, die identische Funktion wird nur bezüglich eines anderen Systems von unendlich vielen Basisvektoren dargestellt ([LOC-94]). Zweckmäßig daran ist, daß es damit egal wird, ob man Funktion oder Transformierte betrachtet, die Information ist gleich, nur die Darstellung unterschiedlich. Für die Analyse eines Signals, das mit Hilfe einer Funktion dargestellt wird, kann man also jeweils die geeignete Darstellung, Zeit- oder Frequenzbereich, wählen.

Wichtiger im vorliegenden Fall ist die Frage, was eigentlich der Grund ist, warum die Fouriertransformation für Signaluntersuchungen so gut geeignet ist. In technischen Anwendungen wird das Integral, das bei der inversen Fouriertransformierten bestimmt wird, als die Summation einer unendlichen Anzahl von komplexen Schwingungen  $e^{-j\omega t}$  mit der Frequenz  $\omega$  betrachtet – dies ist die physikalische Interpretation der Basisvektoren. Das bedeutet, daß die Fouriertransformierte in der Umkehrung eine Möglichkeit bietet, alle in einem Signal vorhandenen Frequenzen zu betrachten und festzustellen, mit welchem Betrag die Frequenz in dem Signal vorhanden ist.

Es wird im weiteren Verlauf noch einmal darauf eingegangen, daß die Beträge der Fouriertransformierten für große Werte verschwinden. Dies führt dazu, daß man betragsgroße Frequenzen und betragskleine Frequenzen finden wird, die Energie ist nicht gleichmäßig auf alle Frequenzen verteilt, sondern bevorzugt bestimmte Frequenzbänder, in denen sich ein Großteil der Signalenergie befindet.

Die folgende Abbildung zeigt dies für ein einfaches Beispiel, als Funktion im Zeitbereich wurde die Rechteckfunktion

---

<sup>4</sup> Bei dieser Argumentation wurde zwar die mathematische Sichtweise „Voraussetzung – Beweis – Satz“ von hinten aufgeklärt, für Physiker oder Ingenieure ist diese Vorgehensweise aber völlig normal und nicht weiter verwerflich. Häufig geht man hier her und verwendet einen Satz ohne Prüfung der Voraussetzungen mit der Argumentation, wenn der Satz erfüllt ist, müssen ja auch die Voraussetzungen gestimmt haben. Diese Auffassung ist techniküblich und soll hier nicht weiter diskutiert werden.

$$\text{rect}\left(\frac{t}{T}\right) = \begin{cases} 1 & , -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & , \text{sonst} \end{cases} \quad (7)$$

mit ihrer Fouriertransformierten

$$\text{rect}^{\wedge}(x) = T \cdot \text{sinc } pTx := \frac{\sin pTx}{px} \quad (8)$$

gewählt. Man sieht deutlich, daß sich im Spektrum ein Großteil der Energie auf ein Intervall  $[-\frac{2}{T}, \frac{2}{T}]$  konzentriert. Außerdem verschwindet für größer werdende Frequenzen die Fouriertransformierte der Rechteckfunktion, da

$$\lim_{x \rightarrow \pm\infty} \text{rect}^{\wedge}(x) = 0 \quad (9)$$

gilt. Ohne dies zunächst weiter auf den allgemeinen Fall auszudehnen, sieht man also, daß es Funktionen gibt, deren Energie im wesentlichen auf einen bestimmten Bereich des gesamten Frequenzbereichs konzentriert ist, diese wesentliche Auffälligkeit wird später noch einmal aufgegriffen werden.

Es gibt nur zwei wesentliche Ausnahmen von diesem Verhalten: zum einen die Nullfunktion, bei der die Fouriertransformierte ebenfalls die Nullfunktion darstellt. Hier gibt es keine Betragsunterschiede bei verschiedenen Frequenzen, allerdings ist auch keine Information in dieser Funktion enthalten, so daß die Energiebetrachtung ohnehin nicht sinnvoll ist. Die andere Ausnahme sind Fouriertransformierten, die konstante Funktionen darstellen. Allerdings sind die zugehörigen Zeitfunktionen die sogenannten Delta-Distributionen, sogenannte verallgemeinerte Funktionen, die in realen Systemen nicht erzeugbar sind, insbesondere also auch nicht in Bildern auftreten können.

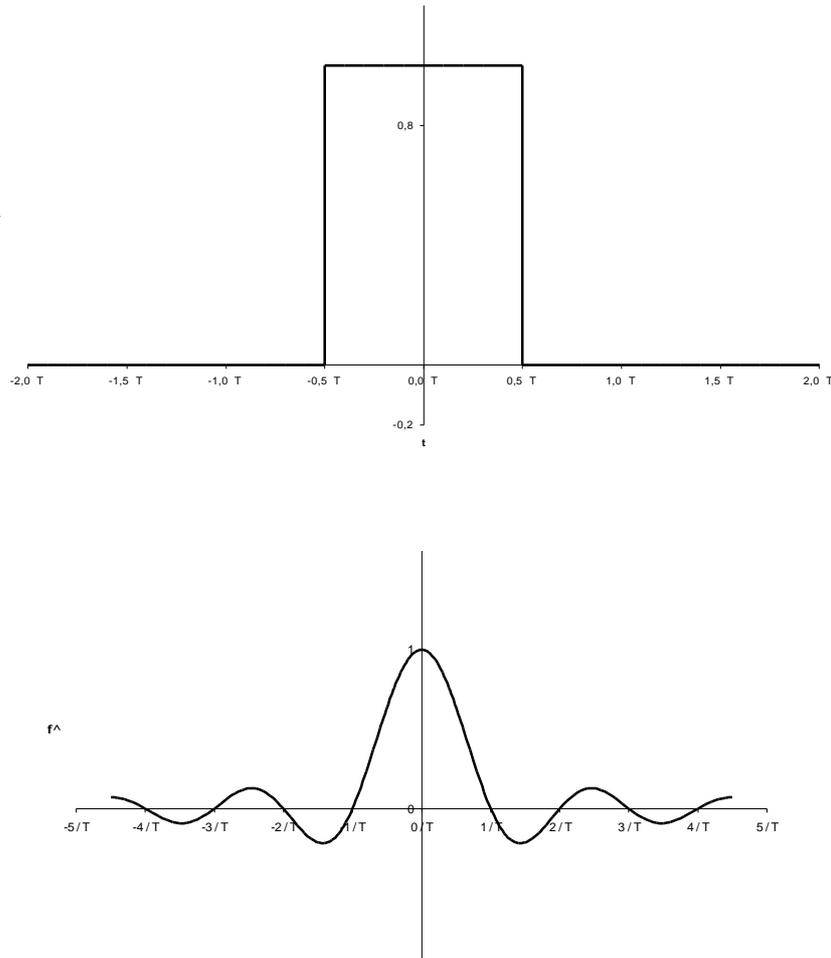


Abbildung 7: Die Rechteckfunktion *rect* und ihre Fouriertransformierte, die Sinus-Cardinalis Funktion (*sinc*- oder auch *si-Funktion*)

Der Gedanke liegt jetzt nahe, dieses Hilfsmittel nicht nur zur Untersuchung von eindimensionalen Signalen zu nutzen, wie dies häufig geschieht, sondern auch zur Untersuchung von Bildern zu verwenden. Betrachtet man die Gleichung der Fouriertransformierten, so sieht man aber zunächst drei wesentliche Unterschiede:

- Die Fouriertransformierte wird über den Signalbereich in ganz  $\mathcal{R}$  gebildet, ein reales Bild ist aber nur in einem festen Intervall definiert.
- Ein Computerbild besteht aus Punkten, lässt sich also im allgemeinen nicht als stetige Funktion interpretieren.
- Bilder haben eine Matrixstruktur, besitzen also zwei Dimensionen, im Gegensatz zur Fouriertransformation, die nur für Funktionen einer Dimension definiert ist.

Bilder haben für jeden Bildpunkt einen Farbwert, der sich nicht in einer Skala darstellen läßt, sondern der selbst nur als Vektor in einem Farbraum darstellbar ist.

Das erste Problem kann man beheben, wenn man das Intervall auf  $\mathcal{R}$  fortsetzt und alle Werte außerhalb des Intervalls identisch 0 setzt. Das Problem mit dem Farbwert läßt sich dadurch beheben, daß man keine Farbbilder, sondern Grauwertbilder betrachtet. Da sich jedes Farbbild durch 3 Grauwertbilder für die Farbauszüge Rot, Grün und Blau darstellen läßt, erhält man die Möglichkeit, jeden Bildpunkt nach Farbauszügen getrennt in einer einzigen Skala darzustellen. Man zerlegt das Bild in 3 Grauwertbilder, führt eine getrennte Bearbeitung durch und kombiniert die Bilder wieder.

Die anderen Einwände sollen in den nächsten Abschnitten näher untersucht und mit entsprechenden neuen Hilfsmitteln behandelt werden.

### 2.1.2 Der eindimensionale diskrete Fall

Bekanntlich läßt sich jeder periodischen Funktion, die über eine Periode absolut integrierbar ist, eine komplexe Fourierreihe zuordnen. Diese Zuordnung ist allgemein bekannt und in der Literatur auch sehr oft ausführlich dargestellt (z.B. [LOC-94], [OPP-92]). Etwas weniger bekannt ist die Tatsache, daß sich auch periodischen Folgen eine *zeitdiskrete Fourierreihe* zuordnen läßt. Der Wunsch dies zu tun ist verständlich, wenn man an die zunehmende Verbreitung von Abtastsystemen denkt, wo als Signale ausschließlich Folgen verarbeitet werden.

#### Def. Abtastfolge

Sei  $f$  eine Funktion auf  $\mathcal{C}$  und  $x$  eine Folge auf  $\mathcal{C}$ . Dann definiert man die Abtastfolge  $x$  zur Funktion  $f$  mit der Abtastrate  $T_{ab}$  (Abtastintervall) zu

$$x[k] = f(kT_{ab}) \quad , k \in \mathbf{Z} \quad (10)$$

#### Def. zeitdiskrete Fourierreihe

Sei  $x$  eine periodische Folge auf  $\mathcal{C}$  mit der Periode  $T$ , d.h. für die gelte  $x[n] = x[n + kT] \quad \forall k \in \mathbf{Z}$  für ein festes  $n \in \mathbf{Z}$ , dann heißen  $c_m \in \mathcal{C}, m \in \mathbf{Z}$  die Fourierkoeffizienten zur Folge  $x$  mit

$$c_m = \frac{1}{N} \sum_{\substack{n=0+nT \\ n \in \mathbf{Z} \text{ fest}}}^{(N-1)+nT} x[n] e^{-j2\pi \frac{n}{N} m} \quad , m \in \mathbf{Z} \quad (11)$$

Ebenso existiert eine Darstellung der Folge  $x$  aus den Fourierkoeffizienten

$$x[n] = \sum_{\substack{m=0+nT, \\ n \in \mathbf{Z} \text{ fest}}}^{(N-1)+nT} c_m e^{j2\pi \frac{m}{N}n}, n \in \mathbf{Z} \quad (12)$$

Für die relativ umfangreiche Herleitung der Darstellung der zeitdiskreten Fourierreihe, sowie den entsprechenden Beweisen für die Gültigkeit der Gleichungen sei auf [OPP-92] verwiesen.

Interessant ist die Feststellung, daß Fourierkoeffizienten  $c_m$  ebenfalls eine Periode besitzen, es gilt nämlich

$$c_m = c_{m+nN}, n \in \mathbf{Z} \quad (13)$$

Diese Tatsache hängt letztendlich damit zusammen, daß die zugehörige Folge aus einer Abtastung entstanden ist. Das Abtasttheorem (z.B. in [OPP-92], [LOC-96]) sagt aus, daß nach einer Abtastung einer bandbegrenzten Funktion das Spektrum bzw. die Fouriertransformierte der Funktion periodisch ist. Damit folgt auch die zeitdiskrete Fourierreihe dem Zusammenhang, daß die Periodisierung einer Funktion zu einem diskreten Spektrum, und daß eine Abtastung zu einer Periodisierung des Spektrums führt.

Ein weiterer wichtiger Gegenstand läßt sich aus der Periodizität der Fourierkoeffizienten ableiten: es gibt nur maximal  $N$  unterschiedliche Fourierkoeffizienten. Man ist also in der Lage, für eine Folge der Länge  $N$  nach der Bildung einer zeitdiskreten Fourierreihe die Funktion mit ebenfalls wieder  $N$  Koeffizienten zu beschreiben.

Dies ist eine bedeutsame Voraussetzung für den folgenden Abschnitt.

Man betrachte ein Signal  $x[n]$  mit endlicher Dauer bzw. der Länge  $N$ , für die Folge gilt also

$$x[n] = 0 \quad \forall n \in [0, N-1] \quad (14)$$

Nun bildet man eine periodische Folge  $\tilde{x}[n]$  mit der Periode  $N$ , die über eine Periode identisch ist mit der Originalfolge  $x[n]$ , so daß gilt:

$$\tilde{x}[n] = x[n], \quad 0 \leq n \leq N-1 \quad (15)$$

Für eine periodische Folge erhält man nach [OPP-92] die Fourierkoeffizienten zu

$$c_m = \frac{1}{N} \sum_{\substack{n=0+nN, \\ n \in \mathbf{Z}}}^{(N-1)+nN} \tilde{x}[n] e^{-j2\pi \frac{mn}{N}}, m \in \mathbf{Z} \quad (16)$$

Dabei ist es gleichgültig, über welche Periode die Summation ausgeführt wird. Wählt man die Periode so, daß sie mit der Ausgangsfolge übereinstimmt, so gilt:

$$c_m = \frac{1}{N} \sum_{m=0}^{N-1} \tilde{x}[n] e^{-j2\pi \frac{mn}{N}} = \frac{1}{N} \sum_{m=0}^{N-1} x[n] e^{-j2\pi \frac{mn}{N}} \quad (17)$$

Die durch diese Gleichung definierten Koeffizienten beziehen sich auf die diskrete Fouriertransformierte, kurz *DFT*, die wie folgt definiert ist:

Def. Diskrete Fouriertransformierte

Seien  $F \in \mathbb{C}^N$ ,  $f \in \mathbb{C}^N$ ,  $N \in \mathbb{N}$  Vektoren. Dann gilt für die diskrete Fouriertransformation

$$DFT : \mathbb{C}^N \rightarrow \mathbb{C}^N$$

$$f \mapsto F : F_n = \frac{1}{N} \sum_{m=0}^{N-1} f_m e^{-j2\pi \frac{mn}{N}} \quad (18)$$

Auf den ersten Blick sieht man, daß die diskrete Fouriertransformierte eine lineare Operation ist, da auch die Summation linear ist.

Für praktische Anwendungen viel wichtiger ist die Tatsache, daß bei der DFT keinerlei Konvergenzprobleme auftreten können: alle Summationen sind endlich, da die Intervalle endlich sind. Gerade Konvergenzbetrachtungen stellen aber in der Theorie der kontinuierlichen Fouriertransformation die größten Probleme bei der praktischen Umsetzung der mathematischen Aussagen dar. Daher werden Konvergenzbetrachtungen im Folgenden keinerlei Rolle mehr spielen.

### 2.1.3 Die Inverse der diskreten Fouriertransformierten

Nach [OPP-92] ist die Inverse der diskreten Fouriertransformierten eines Vektors durch folgende Beziehung gegeben:

#### Def. Inverse der diskreten Fouriertransformierten

Seien  $F \in \mathbb{C}^N$ ,  $f \in \mathbb{C}^N$ ,  $N \in \mathbb{N}$  Vektoren. Dann gilt für die inverse, diskrete Fouriertransformation

$$iDFT : \mathbb{C}^N \rightarrow \mathbb{C}^N$$

$$F \mapsto f : f_n = \sum_{m=0}^{N-1} F_m e^{j2\pi \frac{nm}{N}} \quad (19)$$

Wie man leicht durch einsetzen zeigt, gilt:

$$iDFT(DFT(f)) = f \quad (20)$$

#### Beweis:

$$\begin{aligned} f_n &= \sum_{m=0}^{N-1} F_m e^{j2\pi \frac{nm}{N}} \\ &= \sum_{m=0}^{N-1} \frac{1}{N} \underbrace{\sum_{n=0}^{N-1} f_n e^{-j2\pi \frac{nm}{N}}}_{DFT} e^{j2\pi \frac{nm}{N}} \\ &= \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_n e^{j2\pi \frac{m}{N}(n-n)} \\ &= \frac{1}{N} \cdot N \cdot f_n \\ &= f_n \end{aligned}$$

Die Summation läßt sich wegen den Orthogonalitätsrelationen für die komplexen trigonometrischen Funktionen auflösen.

Rechentechisch interessant ist ein weiterer Zusammenhang zwischen der DFT und der inversen DFT. Die Berechnung der inversen DFT lässt sich sehr leicht auf die Berechnung einer DFT zurückführen, was algorithmisch interessant ist: eine Implementation der DFT in einer einmal gewählten Form ist ausreichend, um auch die inverse DFT zu bestimmen:

$$\begin{aligned}
 f_n &= \sum_{m=0}^{N-1} F_m e^{j2\pi \frac{nm}{N}} = \overline{\overline{\sum_{m=0}^{N-1} F_m e^{j2\pi \frac{nm}{N}}} } = \overline{\sum_{m=0}^{N-1} \overline{F_m} e^{-j2\pi \frac{nm}{N}}} \\
 &= N \cdot \overline{DFT(\overline{F})}
 \end{aligned}
 \tag{21}$$

Das entsprechende Mittel ist die Konjugation der beteiligten Vektoren, eine Rechenoperation, die sich in der Größenordnung von  $O(N)$  ausführen lässt.

#### 2.1.4 Physikalische Interpretation der DFT

Zum Entwurf von Filtern benötigt man den Zusammenhang des Eingangs- und Ausgangsvektors der DFT mit den physikalischen Frequenzen. Bei der kontinuierlichen Fouriertransformation war dies relativ einfach, zu jeder komplexen Frequenz  $e^{-jx}$  lässt sich mit  $f^{\wedge}(x)$  eindeutig die komplexe Amplitude dieser Frequenz ermitteln.

Zunächst wird der Zeitbereich für einen Vektor der Länge  $N$  betrachtet.

#### Satz: Koeffizienten im Zeitbereich

Sei  $x$  ein komplexer Vektor der Länge  $N \in \mathbb{N}$ , sowie  $T_{ab} \in \mathbb{R}$  die zugehörige Abtastzeit. Dann gehört zum  $k$ -ten Koeffizienten  $x_k$  die Zeit  $t$

$$t = k \cdot T_{ab}, 0 \leq k < N
 \tag{22}$$

Setzt man den Vektor  $x$  nun periodisch auf  $\mathbb{Z}$  fort, so kann man den Zusammenhang zwischen Index und Zeit sogar noch erweitern:

$$t = k \cdot T_{ab} = (k \bmod N) T_{ab}, k \in \mathbb{Z}
 \tag{23}$$

Dieser Zusammenhang folgt unmittelbar aus der Entstehung der Folge durch Abtastung.

Durch die Modulo-Operation wird der Periodizität des Zeitsignals Rechnung getragen. Im Frequenzbereich hängt die Zuordnung zwischen Frequenz und dem Index der Fourierkoeffizienten von der spektralen Auflösung der Fourierreihe ab. Für eine Fourierreihe einer  $T$ -periodischen Funktion besitzen die Koeffizienten der Fourierreihe den Abstand

$$\Delta c = \frac{1}{T} \quad (24)$$

voneinander. Da die DFT aus einer Fourierreihendarstellung abgeleitet ist, gilt die Gleichung ebenfalls für die DFT. Die Periode  $T$  des Zeitsignals ist hier insbesondere  $T = N \cdot T_{ab}$ . Insgesamt erhält man dann für den Zusammenhang zwischen dem Index der Koeffizienten der DFT im Frequenzbereich und der tatsächlichen Frequenz die folgende

#### Def. Koeffizienten im Frequenzbereich

Seien  $X$  und  $x$  komplexe Vektoren der Länge  $N \in \mathbb{N}$ , für die gilt  $X = DFT(x)$ , sowie  $T_{ab} \in \mathbb{R}$  die zugehörige Abtastzeit. Dann gehört zum  $k$ -ten Koeffizienten  $X_n$  die Frequenz  $f$

$$f = k \cdot \frac{\Delta c}{N} = \frac{k}{N \cdot T_{ab}}, 0 \leq k < N \quad (25)$$

Setzt man den Vektor  $X$  nun periodisch auf  $\mathbb{Z}$  fort, so kann man den Zusammenhang zwischen Index und Frequenz ebenfalls noch erweitern:

$$f = k \cdot \frac{\Delta c}{N} = \frac{k \bmod N}{N \cdot T_{ab}}, k \in \mathbb{Z} \quad (26)$$

Auch hier spiegelt die Modulo-Operation die Periodizität des Spektrums wieder, die durch die Abtastung im Zeitbereich bedingt ist. Wichtig ist die letzte Definition in erster Linie für den Sprachgebrauch, damit der Begriff der Frequenz<sup>5</sup> auch für die diskrete Fouriertransformation eindeutig mit den Koeffizienten in Zusammenhang gebracht werden kann (sehr ausführlich dargestellt in [OPP-92]).

---

<sup>5</sup> Um eine wichtige Fehlinterpretation zu vermeiden, muß man eigentlich von einem Frequenzband sprechen, da jeder Koeffizient alle Frequenzen in einem Intervall  $[f - \Delta c / 2, f + \Delta c / 2]$  repräsentiert. Dies ist die physikalische Interpretation, die aber ebenfalls in der Bezeichnung häufig nicht so genau genommen wird.

### 2.1.5 Idee der Datenreduktion

Als elementarer Vektor<sup>6</sup> soll eine *Sprungfolge* betrachtet werden. Für diesen Vektor wurde in der folgenden Abbildung die Anzahl der Stützstellen auf  $N = 32$  gesetzt.

Vektor mit einem Sprung:

$$y[n] = \begin{cases} 0 & , n = 0.. \frac{N}{2} - 1 = 0..15 \\ 1 & , n = \frac{N}{2} .. N - 1 = 16..31 \end{cases}$$

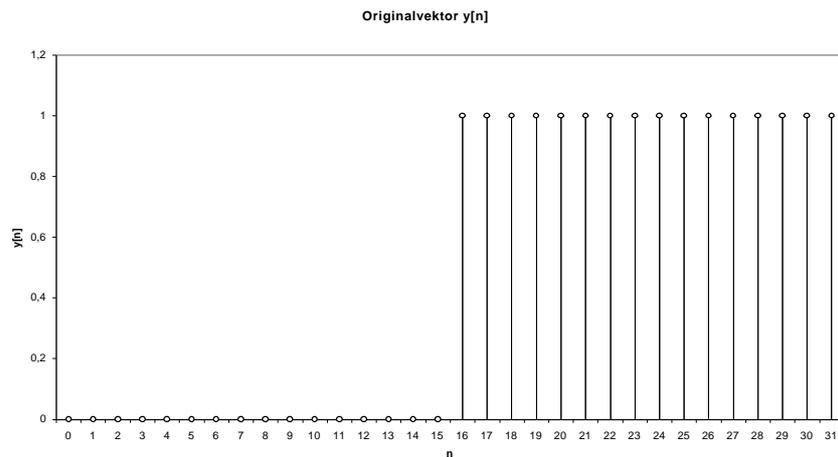


Abbildung 8: Ein Sprung mit 32 Stützstellen

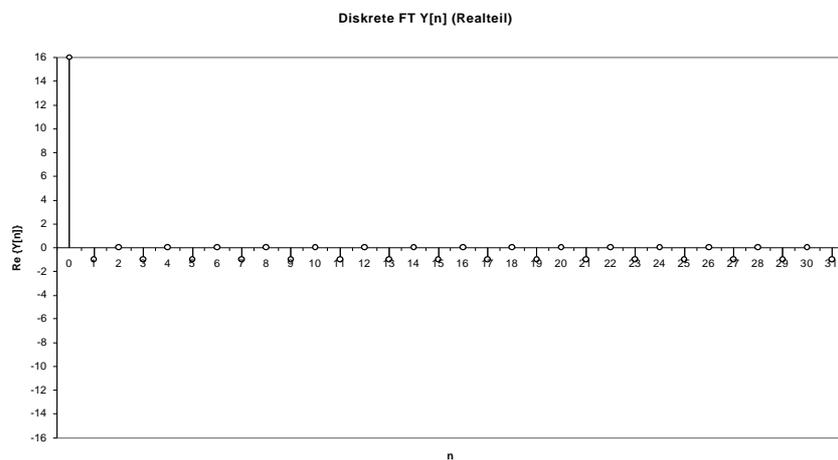


Abbildung 9: Der Realteil der diskreten Fouriertransformierten eines Sprunges mit 32 Stützstellen

<sup>6</sup> In diesem Zusammenhang ist der Begriff „elementarer Vektor“ im Sinne der Informationsdarstellung zu verstehen, nicht im mathematischen Sinne als Basisvektor.

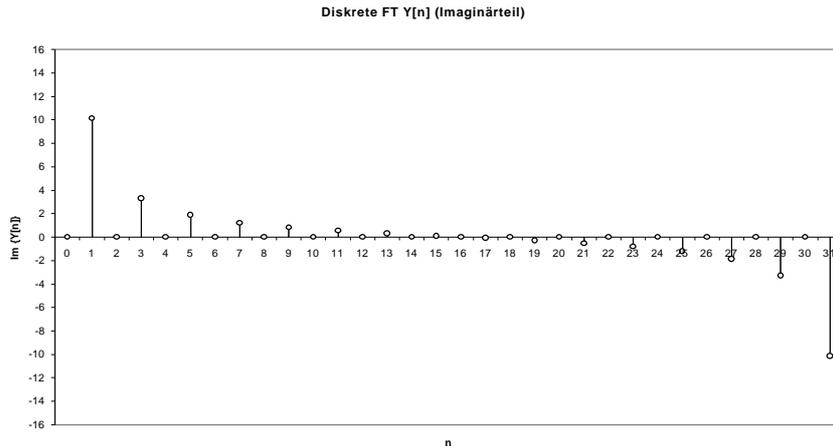


Abbildung 10: Der Imaginärteil der diskreten Fouriertransformierten eines Sprunges mit 32 Stützstellen

Schon eine solch einfach Betrachtung wie in Abbildung 9 und Abbildung 10 zeigt, daß die Beträge der Koeffizienten zu höheren Frequenzanteilen hin abnehmen. Die etwas verwirrende Zuordnung der tiefen und hohen Frequenzen liegt am Spiegelungseffekt der Abtastung, die höchste vorkommende Frequenz in der Abtastfolge ist die halbe Abtastfrequenz, die bei  $N/2$  liegt.

Die Frage für eine allgemeine Anwendung ist, ob die betragskleinen Koeffizienten immer auf bestimmte Frequenzbereiche beschränkt sind. Falls dies der Fall ist, so könnte man diese betragskleinen Koeffizienten durch eine 0 ersetzen. Für eine Datenreduktion ergibt sich dazu die Überlegung, daß man einen Vektor mit  $N$  Koeffizienten einer diskreten Fouriertransformation unterwirft, betragskleine Koeffizienten in einem bestimmten Frequenzband unterdrückt, was den Vektor natürlich um eine bestimmte Anzahl von Koeffizienten vermindert.

Der Effekt: der Speicherbedarf ist geringer geworden.

Zur Rekonstruktion führt man eine inverse diskrete Fouriertransformation aus, und erhält einen Vektor, der sich hoffentlich nicht so stark vom Original unterscheidet. Es ist klar, daß hierbei Informationen verloren gehen, denn die betragskleinen Koeffizienten wären natürlich zur vollständigen Rekonstruktion des Originals notwendig. Es ist hierbei das Ziel, nur so viele Koeffizienten wegzulassen, daß die Energie innerhalb dieses Bandbereichs klein ist gegenüber der Gesamtenergie des Vektors.

Interessant ist nun, ob die betragskleinen Koeffizienten auf ein bestimmtes Frequenzband beschränkt sind, oder ob diese Koeffizienten mehr oder weniger zufällig (bzw. abhängig von den repräsentierten Informationen) verteilt sind. Dazu soll zuerst noch einmal kurz die kontinuierliche

Fouriertransformation für Funktionen betrachtet werden, bei denen die Bildung der Transformierten und der inversen Transformierten wieder auf die Ausgangsfunktion zurückführen.

### Satz: Riemann-Lebesgue-Theorem

Es sei  $f \in L_1(\mathbb{R})$  beliebig gegeben. Dann gilt:

$$\lim_{|x| \rightarrow \infty} f^\wedge(x) = 0 \quad (27)$$

Der Beweis dazu findet sich in [LOC-94].

Daraus folgt unmittelbar, daß es für jedes  $\epsilon > 0$  eine Zahl  $d > 0$  gibt, so daß gilt

$$|f^\wedge(x)| < \epsilon \quad \forall |x| > d \quad (28)$$

Mit anderen Worten: es läßt sich für jede kontinuierliche Fouriertransformierte eine Frequenz finden, ab der die Beträge der Transformierte so klein werden, daß man die Transformierte nur für die Frequenzen unterhalb dieser Grenzfrequenz berücksichtigen kann, ohne einen wesentlichen Teil der Signalenergie zu verlieren. Dabei wird  $\epsilon$  nach der gewünschten Genauigkeit gewählt. Ein solche Näherung führt automatisch dazu, daß man vor allem sehr hohe Frequenzen vernachlässigen wird, wenn man eine Reduktion der Datenmenge erzielen will<sup>7</sup>. Diese Aussage gilt ebenfalls für die Koeffizienten der Fourierreihe, wie an verschiedener Stelle gezeigt wird.

Leider läßt sich dieses Erkenntnis nicht unmittelbar auf die diskrete Fouriertransformation übertragen, da hier das insgesamt betrachtete Frequenzband nicht ganz  $[0, \infty)$  umfaßt, sondern nur ein Frequenzband  $[0, f_{ab}]$ . Und in diesem Frequenzband darf man für den allgemeinen Fall keinerlei Verhalten erwarten, das dem des Grenzwertes der kontinuierlichen Transformation entspricht. Man muß hier je nach Anwendungsgebiet auf einige Annahmen ausweichen, in der Hoffnung, daß diese für möglichst viele Fälle zutreffen.

Geht man zurück zum Ausgangspunkt - dem Versuch, Bilddatenreduktion zu betreiben - so sollte man die Bilder in zwei große Gruppen einteilen:

---

<sup>7</sup> Diese Näherung führt bei realen Systemen in weiten Grenzen nicht einmal zu einem meßbaren Näherungsfehler, da reale Systeme ohnehin immer bandbegrenzt sind und hohe Frequenzen ab einer systemabhängigen Grenzfrequenz nicht übertragen.

- Grauwertbilder (die, wie angesprochen, auch zu Farbbildern kombinierbar sind) und
- Zeichnungen und Texte

In Grauwertbildern werden selten starke Helligkeits- oder Kontrastwechsel auftreten, es wird immer Übergänge geben. Diese können zwar mehr oder weniger rasch erfolgen, werden aber immer vorhanden sein. Anders dagegen eine Zeichnung: hier wird man überwiegend starke Wechsel finden. Das Problem an dieser Aussage ist, daß es sich hier in erster Linie um Erfahrungswerte handelt, die man sich im Laufe der Zeit angeeignet hat. Die Theorie der Fourieranalyse zeigt eindeutig, daß aus Sprüngen starke *Oberwellen*, also Energieanteile in höheren Frequenzbereichen, folgen. Dies gilt auch dann, wenn das Signal wie bei einer Abtastung bandbegrenzt ist.

Man kann also von der Überlegung ausgehen, daß ein Schnitt durch Grauwertbilder, die reale Szenen darstellen, in erster Linie tiefere Frequenzanteile besitzt, während Zeichnungen und Texte zu einem Schnitt führen, der höhere Frequenzen stärker betont. Leider ist dies in erster Linie eine qualitative Aussage, man kann keine quantitativen Aussagen machen, wie die Frequenzverteilung im allgemeinen Fall tatsächlich verlaufen wird (dazu [HAB-91]).

Mathematisch gesehen ist es so, daß das Ausgangssignal durch den Originalvektor exakt dargestellt wird und daß die inverse Transformation gegen dieses Original konvergiert. Bricht man nun die Summation vorzeitig ab, was ja im wesentlichen dem Nullsetzen entspricht, so erhält man eine Näherung an das Original. Unglücklicherweise lassen sich Folgen in der Form konstruieren, so daß man keine allgemeine Aussage über eine Konvergenzgeschwindigkeit machen kann.

Die folgenden Abbildungen zeigen einen Längsschnitt<sup>8</sup> durch eine angenommene Helligkeitsverteilung, wie sie in einem Grauwertbild vorkommen kann, und die zugehörigen diskreten Fouriertransformierten.

---

<sup>8</sup> Der Längsschnitt wurde gewählt, weil im Moment noch nicht auf die Matrixstruktur der Bilder eingegangen werden soll. Durch den Schnitt kann im Moment noch mit Vektoren gearbeitet werden.

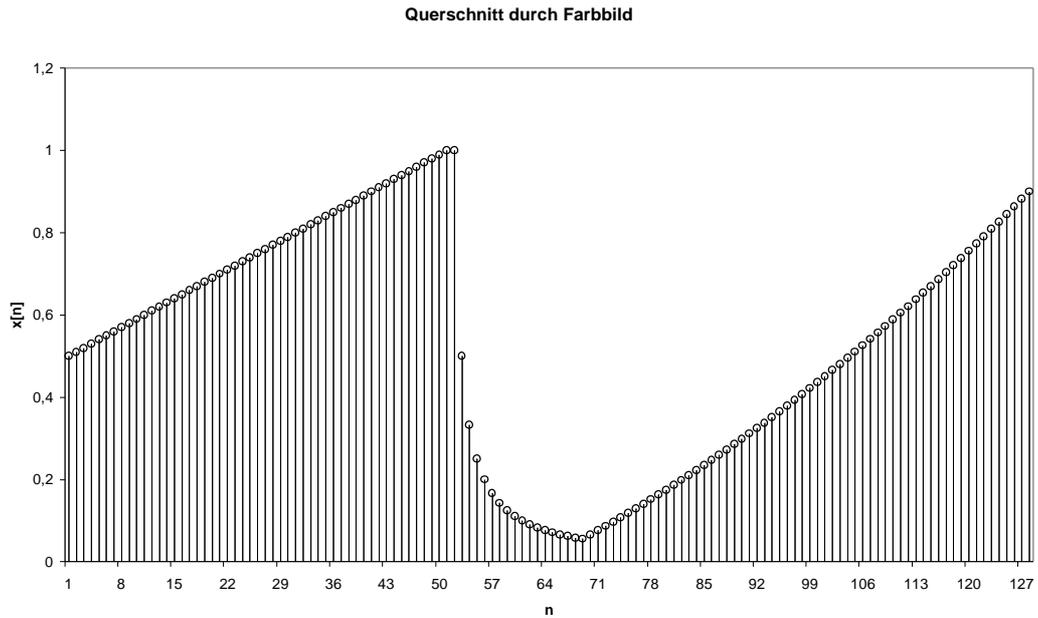


Abbildung 11: Ein Querschnitt mit 128 Punkten durch ein fiktives Grauwertbild mit glatten Grauwertverläufen, wie man sie für ein reales Bild annehmen kann

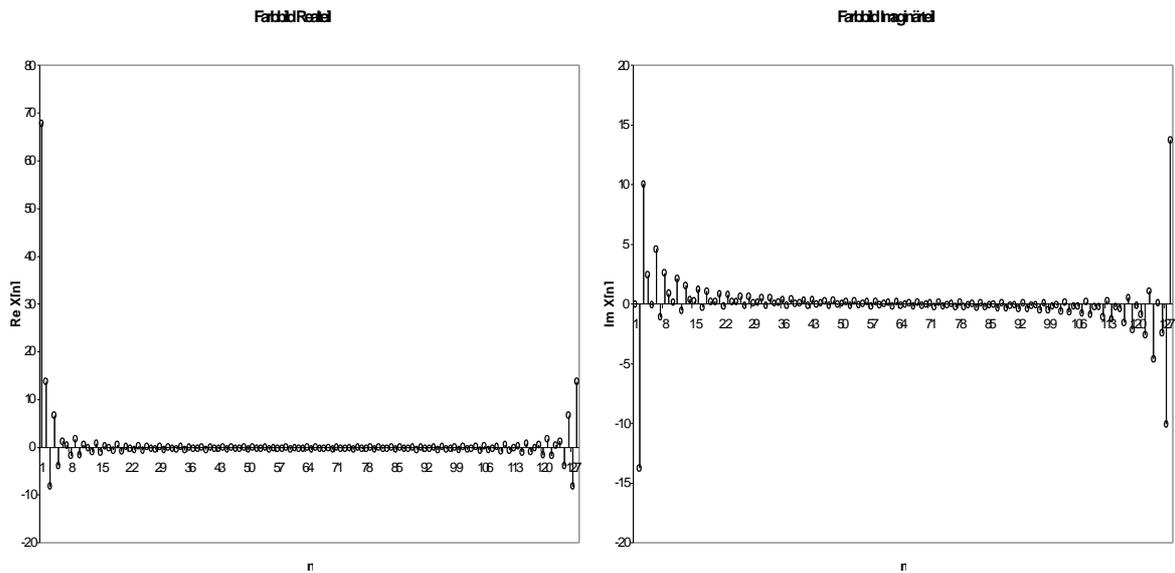


Abbildung 12: Die diskrete Fouriertransformierte für das Grauwertbild, nach Real- und Imaginärteil

Es bietet sich nun an mit Filtern entweder hohe oder tiefe Frequenzen zu unterdrücken und somit die Speicherung des transformierten Vektors in einem kürzeren Vektor zu ermöglichen. Die folgende Abbildung zeigt für das vorige Beispiel des Weglassens eines Teils der Koeffizienten (sie werden auf 0 gesetzt) und das Ergebnis einer anschließenden inversen diskreten Fouriertransformation.

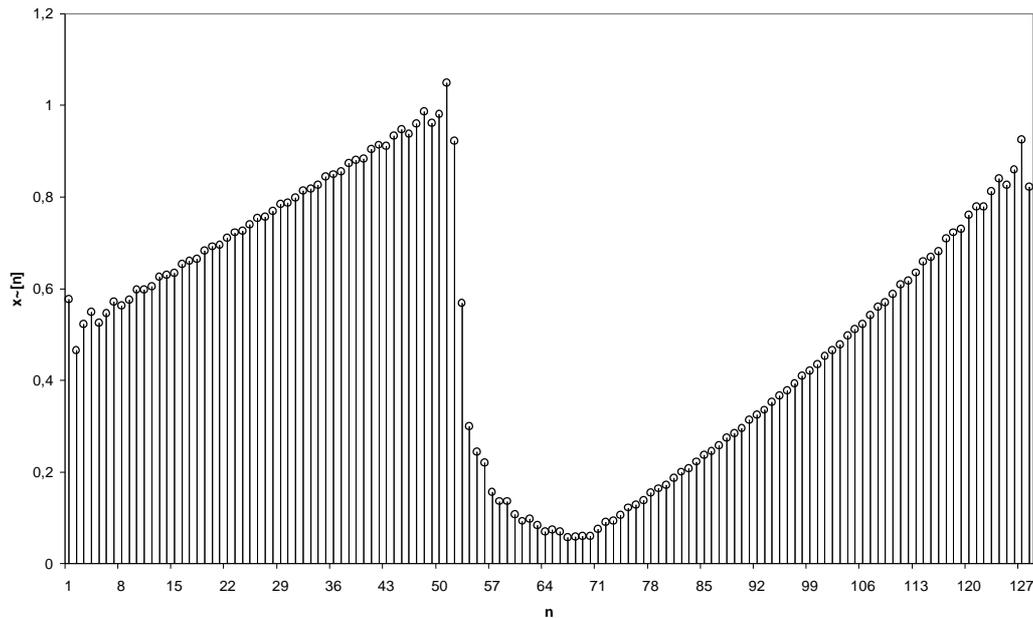


Abbildung 13: Das Grauwertbild nach Nullsetzen der diskreten Fourierkoeffizienten  $X[40]..X[80]$

Offensichtlich entspricht das Ergebnis nicht mehr vollständig dem Original, allerdings ist doch festzustellen, daß eine Ähnlichkeit vorhanden ist. Der Schnitt durch das Grauwertbild in Abbildung 11 und Abbildung 13 ist nur leicht verändert, obwohl 40 von 128 Fourierkoeffizienten weggelassen wurden, dies sind immerhin 30% der Koeffizienten. Inwieweit die Ähnlichkeit auch für die Anwendung auf Bilder gegeben ist, werden die folgenden Abschnitte noch zeigen, aber die Idee, wie man Daten mit Hilfe der diskreten Fouriertransformation reduzieren kann, sollte nun plausibel sein.

### 2.1.5.1 Tiefpaßfilter

Ein *Filter* ist in der Systemtheorie ein bekanntes System mit einem bestimmten Übertragungsverhalten, bei dem Energieanteile eines Eingangssignals auf der Ausgangsseite nicht mehr vorhanden sind. Prinzipiell gibt es zwei Möglichkeiten, wie ein Filter beschrieben werden kann. Zum einen kann das Übertragungsverhalten im Zeitbereich beschrieben werden, hier wird das Ergebnis der Filterung durch eine diskrete Faltung von Filterfunktion und Eingangsfolge ermittelt. Die andere Beschreibungsmöglichkeit ist die Darstellung des Filters durch seine diskrete Fouriertransformierte, hier wird die Ausgangsfolge durch Multiplikation der transformierten Eingangsfolge und der transformierten Filterfolge gewonnen. Speziell handelt es sich hierbei um eine Abbildung, bei der bestimmte Frequenzbereiche, also die entsprechenden Koeffizienten, mit einer *Gewichtsfunktion* beaufschlagt werden. Dieser Sachverhalt ist auch aus der Theorie der kontinuierlichen Fouriertransformation unter dem Begriff der *Faltungssätze* bekannt.

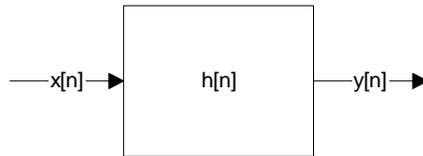


Abbildung 14: Allgemeine Darstellung eines Systems mit Filter

Für das System nach Abbildung 14 erhält man daher folgende Möglichkeiten, um die Ausgangsfolge  $x[n]$  zu berechnen:

Satz: Systemübertragungsverhalten

Sei  $x \in \mathcal{C}^N$  der Eingangsvektor eines diskreten Systems,  $y \in \mathcal{C}^N$  der Ausgangsvektor des Systems und  $h \in \mathcal{C}^N$  ein Filtervektor, weiterhin gilt  $N \in \mathbb{N}$ . Dann ergibt sich der Ausgangsvektor zu

$$y[n] = x[n] * h[n] = \sum_{n=0}^{N-1} x[n-n] \cdot h[n], 0 \leq n \leq N-1 \quad (29)$$

oder mit Hilfe der diskreten Fouriertransformation zu

$$Y[n] = N \cdot X[n] \cdot H[n], 0 \leq n \leq N-1 \quad (30)$$

wobei für die Vektoren gilt

$$Y = DFT(y), X = DFT(x), H = DFT(h) \quad (31)$$

(aus [LOC-92]).

Von besonderem Interesse ist hier die Methode, den Filter im Frequenzbereich zu bilden, da dann mit einem entsprechendem Filter direkt Koeffizienten ausgeblendet werden können, das Filterdesign wird besonders einfach und übersichtlich.

Es wurde bereits zuvor gezeigt, wie der Index der diskreten Fourierkoeffizienten mit der physikalischen Frequenz zusammenhängt. Daher muß man auch beim Filterentwurf den Spiegeleffekt beachten, was einige kleine Zusatzüberlegungen notwendig macht.

Für den Entwurf eines einfachen Tiefpaßfilters, der nur Koeffizienten aus  $\{0,1\}$  besitzt (vollständiger Durchlaß oder vollständige Sperrung eines Frequenzbereichs), muß daher bedacht werden, daß

auch die Koeffizienten mit hohem Index zum Durchlaßbereich gehören: auf den ersten Blick würde man eher annehmen, daß diese Koeffizienten unterdrückt werden.

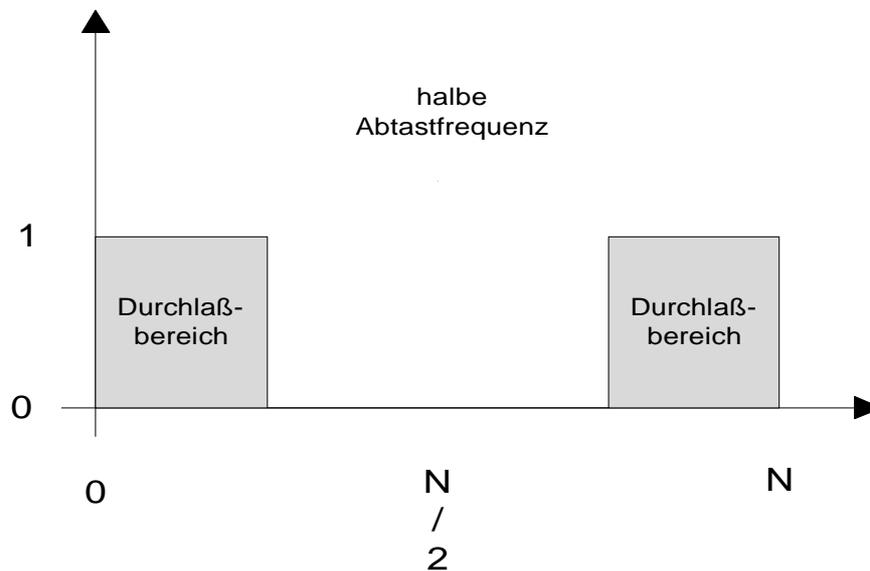


Abbildung 15: Prinzipdarstellung des Durchlaßbereichs eines Tiefpaßfilters für die DFT

### Def. Tiefpaßfilter im Frequenzbereich für DFT

Sei  $x \in \mathbb{C}^N$  der Eingangsvektor eines diskreten Systems,  $y \in \mathbb{C}^N$  der Ausgangsvektor des Systems nach Abbildung 14, weiterhin gilt  $N \in \mathbb{N}$  und  $N$  ist gerade<sup>9</sup>. Dann heißt ein Vektor  $H_{LP} \in \mathbb{C}^N$  ein *idealer Tiefpaß*<sup>10</sup> mit der Grenzfrequenz  $g, g \in \mathbb{N}, 0 \leq g \leq \frac{N}{2}$ , wenn gilt:

$$H_{LP}[n] = \begin{cases} 0 & , g < n < N + 1 - g \\ 1 & , \text{sonst} \end{cases} \quad (32)$$

Man sieht, daß die Definition des Tiefpaßfilters in der Tat einen Durchlaßbereich gemäß Abbildung 15 ermöglicht.

<sup>9</sup> Die Einschränkung, daß der Vektor gerade ist, hat nur eine Bedeutung im Rahmen der vorgestellten Anwendungen. Es ist möglich, die Definition auch auf ungerades  $N$  zu erweitern, allerdings wird später ohnehin davon ausgegangen, daß  $N$  eine Potenz von 2, also insbesondere auch gerade ist.

<sup>10</sup> Dieser Begriff mag irritieren: ein ideales Filter gilt allgemein als nicht realisierbar, dennoch wird hier scheinbar ein ideales Filter mit idealem Durchlaßbereich und idealem Sperrbereich erzeugt. Es handelt sich im vorliegenden Fall aber nicht um kausale Systeme, daher ist diese Einschränkung nicht zutreffend. Und die Sprungantwort dieses Filters setzt in der Tat bereits vor einer Systemanregung ein, da durch die periodische Fortsetzung des Signals im Zeitbereich bereits vor dem Zeitpunkt  $t=0$  eine Systemantwort erscheint.

### 2.1.5.2 Hochpaßfilter

Ein komplementäres Filterverhalten zum Tiefpaß stellt der Hochpaß dar, bei dem die niedrigen Frequenzen gesperrt werden. Auch bei diesem Entwurf ist wieder der Spiegelungseffekt zu beachten.

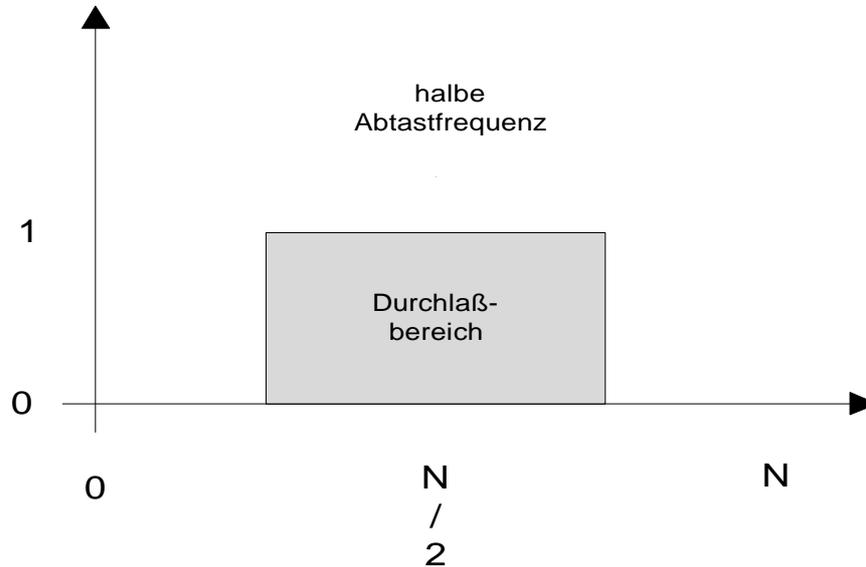


Abbildung 16: Prinzipdarstellung des Durchlaßbereichs eines Hochpaßfilters für die DFT

#### Def. Hochpaßfilter im Frequenzbereich für DFT

Sei  $x \in \mathcal{C}^N$  der Eingangsvektor eines diskreten Systems,  $y \in \mathcal{C}^N$  der Ausgangsvektor des Systems nach Abbildung 14, weiterhin gilt  $N \in \mathbb{N}$  und  $N$  ist gerade. Dann heißt ein Vektor  $H_{HP} \in \mathcal{C}^N$  ein *idealer Hochpaß* mit der Grenzfrequenz  $g, g \in \mathbb{N}, 0 \leq g \leq \frac{N}{2}$ , wenn gilt:

$$H_{HP}[n] = \begin{cases} 1 & , g \leq n \leq N + 1 - g \\ 0 & , \text{sonst} \end{cases} \quad (33)$$

Blickt man kurz in die Systemtheorie, so sieht man, daß in der kontinuierlichen Theorie ein Hochpaßfilter für bestimmte Frequenzen differenzierend wirkt. Dies gilt in ähnlicher Weise auch für den diskreten Fall, allerdings muß hier der Begriff des Differenzierens durch eine Differenzbildung ersetzt werden. Ähnlich wie beim Differenzieren einer Funktion der Absolutwert der Funktion verloren geht und nur noch die Änderungen vorhanden sind, passiert dies auch bei der Differenzbildung. Bei der Umkehrung fehlt daher, analog zur Integration, der richtige Anfangswert. Daher

kommt man zur Idee, den Gleichanteil, dargestellt durch den Fourierkoeffizienten mit Index  $0$ , zum Durchlaßbereich des Hochpaßfilters hinzu zu nehmen – es wird quasi der Aufhängepunkt von der Ausblendung verschont. Dies wird sich später noch als zweckmäßig erweisen. Die folgende Abbildung zeigt den modifizierten Durchlaßbereich.

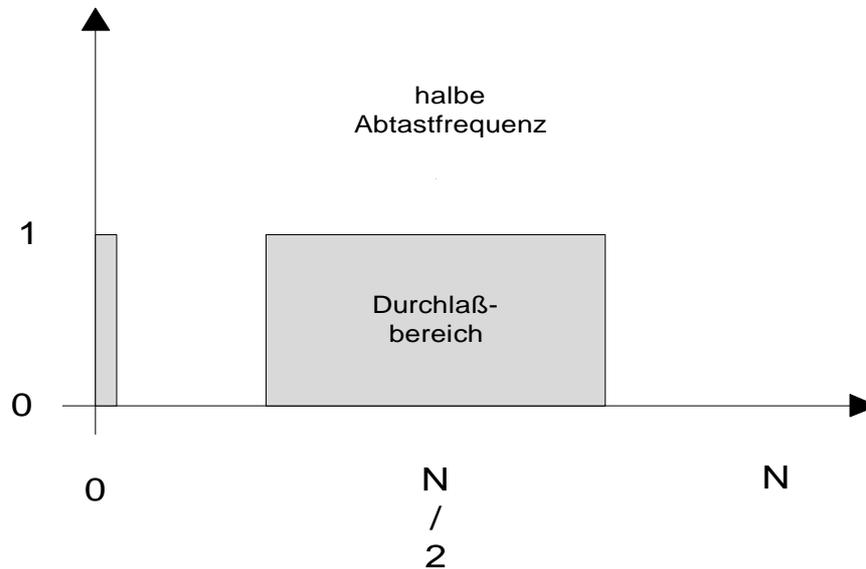


Abbildung 17: Prinzipdarstellung des Durchlaßbereichs eines Hochpaßfilters für die DFT, bei dem auch der Gleichanteil zum Durchlaßbereich gehört

### Def. Hochpaßfilter mit Gleichanteil im Frequenzbereich für DFT

Sei  $x \in \mathcal{C}^N$  der Eingangsvektor eines diskreten Systems,  $y \in \mathcal{C}^N$  der Ausgangsvektor des Systems nach Abbildung 14, weiterhin gilt  $N \in \mathbb{N}$  und  $N$  ist gerade. Dann heißt ein Vektor  $H_{HPDC} \in \mathcal{C}^N$  ein *idealer Hochpaß mit zugeschaltetem Gleichanteil* mit der Grenzfrequenz  $g, g \in \mathbb{N}, 0 \leq g \leq \frac{N}{2}$ , wenn gilt:

$$H_{HPDC}[n] = \begin{cases} 1 & , g \leq n \leq N + 1 - g \vee n = 0 \\ 0 & , \text{sonst} \end{cases} \quad (34)$$

#### 2.1.6 Erweiterung auf den zweidimensionalen Fall

Die vorausgegangenen Überlegungen zur Transformation von Vektoren müssen nun entsprechend auf die Bildverarbeitung übertragen werden. Es ist naheliegend, daß man zur mathematischen Beschreibung eines Bildes eine Matrixform verwendet. Um die elementaren Berechnungen – auch im Hinblick auf die spätere Programmierung – zu vereinfachen, wird davon ausgegangen, daß alle

Matrizen quadratisch sind. Dies hat bei der Implementation den Vorteil, daß man in Zeilen- und Spaltenrichtung mit den gleichen vordefinierten Operationen und Tabellen arbeiten kann, eine wesentliche Vereinfachung bei der Implementation und ein nicht unerheblicher Zeitgewinn für die Ausführung.

### Def. Bildmatrix

Sei  $N \in \mathbb{N}$  und sei  $\mathbf{C}$  eine Matrix mit  $\mathbf{C} \in \mathbb{R}^N \times \mathbb{R}^N$ . Dann wird  $\mathbf{C}$  als eine Bildmatrix bezeichnet, die mit einem entsprechenden Grauwertbild identisch ist. Dabei entspricht jedes Element der Matrix  $c_{yx}$  einem Bildpunkt mit dem Grauwert  $c_{yx}$  an der Koordinate  $(x, y)$ .<sup>11</sup>

#### 2.1.6.1 Die diskrete Fouriertransformierte einer Matrix

Nach [HAB-91] läßt sich eine zweidimensionale diskrete Fouriertransformierte einer Matrix definieren:

### Def. Diskrete zweidimensionale Fouriertransformierte

Seien  $\mathbf{X}$  und  $\mathbf{X}^\wedge$  Matrizen mit  $\mathbf{X} \in \mathbb{C}^N \times \mathbb{C}^N, \mathbf{X}^\wedge \in \mathbb{C}^N \times \mathbb{C}^N, N \in \mathbb{N}$ . Dann lautet die diskrete, zweidimensionale Fouriertransformierte  $\mathbf{X}^\wedge$  der Matrix  $\mathbf{X}$  mit  $\mathbf{X} = (c_{yx}), \mathbf{X}^\wedge = (c_{vu}^\wedge)$  (mit  $0 \leq x \leq N-1, 0 \leq y \leq N-1$  und  $0 \leq u \leq N-1, 0 \leq v \leq N-1$ ):

$$DFT : \mathbb{C}^N \times \mathbb{C}^N \rightarrow \mathbb{C}^N \times \mathbb{C}^N$$

$$\mathbf{X} \mapsto \mathbf{X}^\wedge, c_{vu}^\wedge = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} c_{yx} e^{-j2\pi \frac{xu+yv}{N}} \quad (35)$$

Vorteilhafterweise läßt sich die Berechnung der diskreten, zweidimensionalen Fouriertransformierten einer Matrix auf die Berechnung der diskreten, eindimensionalen Fouriertransformierten der Zeilen- und Spaltenvektoren zurückführen, wie folgende einfache Umformung zeigt:

---

<sup>11</sup> Es sollte darauf hingewiesen werden, daß hier davon ausgegangen wird, daß die Elemente der Matrix reelle Zahlen sind, da nur hier eine sinnvoll Identifikation mit einem Farbwert möglich ist. Für die allgemeine Theorie der diskreten Fouriertransformierten ist dies aber ohne Belang, die Theorie bleibt also auch für komplexe Matrixelemente gültig.

Man betrachtet die diskrete Fouriertransformierte der Matrix  $\hat{\mathbf{X}}$

$$\begin{aligned} \hat{c}_{vu} &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} c_{yx} e^{j2p \frac{xu+yv}{N}} \\ &= \frac{1}{N} \sum_{x=0}^{N-1} \underbrace{\frac{1}{N} \sum_{y=0}^{N-1} c_{yx} e^{-j2p \frac{yv}{N}}}_{\text{dies ist die DFT des jeweiligen Spaltenvektors zur Spalte } x} e^{-j2p \frac{xu}{N}} \\ &= \frac{1}{N} \sum_{x=0}^{N-1} d_{vx} e^{-j2p \frac{xu}{N}} \end{aligned}$$

und erkennt, daß zunächst die DFT  $d_{vx}$  der Spaltenvektoren berechnet werden kann. Danach wird die DFT über die Zeilenvektoren der Matrix

$$(d_{v0} \quad \dots \quad d_{v,N-1})$$

gebildet. Ebenso ist auch der umgekehrte Weg möglich, d.h. zunächst die Zeilen zu transformieren, und danach die Spalten.

Offensichtlich ist also auch die diskrete Fouriertransformation einer Matrix eine lineare Operation.

Die obige Vorgehensweise gilt auch bei anderen linearen Transformationen von Matrizen, auch hier kann die Transformation auf die Transformation der Zeilen- und Spaltenvektoren zurückgeführt werden. Zwei Vorteile ergeben sich daraus: zum einen können die bereits zahlreich vorhandenen Sätze, Theorien und Rechenverfahren – insbesondere der Filterentwurf - der „eindimensionalen“ Welt weiter verwendet werden, zum anderen ergibt sich dadurch eine extrem effiziente Rechnerimplementation, wie später noch gezeigt werden wird<sup>12</sup>. Man bezeichnet diese Art der Berechnung auch als Tensorierung.

Weiterhin wird bei Betrachtung der Tensorierung klar, daß ebenfalls bei der Matrix eines Bildes eine periodische Fortsetzung vorgenommen wird, allerdings eine periodische Fortsetzung nach Zeilen und Spalten. Die folgende Abbildung verdeutlicht dies.

---

<sup>12</sup> Für Schnelleser: die Matrix kann transformiert werden, indem man den FFT-Algorithmus zuerst auf die Zeilen, danach auf die Spalten der Matrix anwendet. Und optimierte FFT-Algorithmen für den eindimensionalen Fall sind in der Informatik und der Numerik ein bekanntes Hilfsmittel.

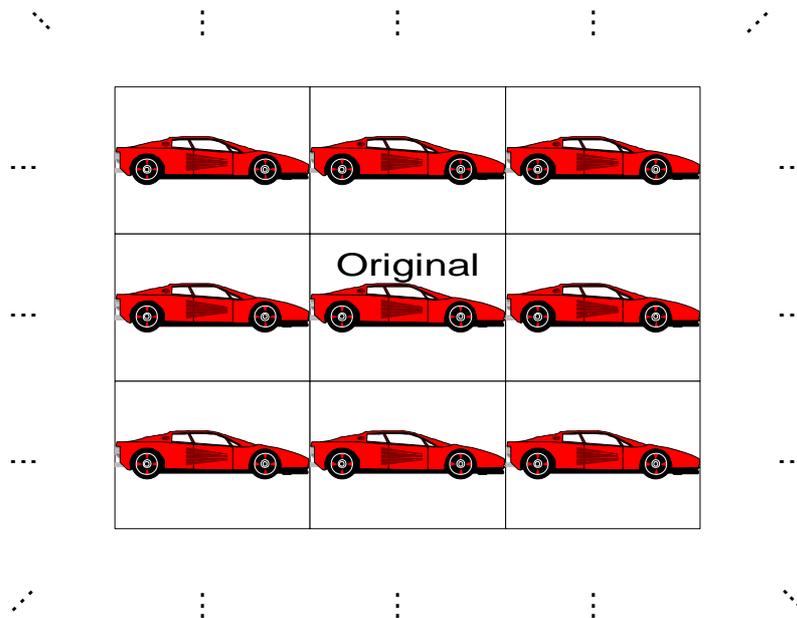


Abbildung 18: Periodische Fortsetzung des Originalbildes nach Zeilen und Spalten

Dieser Effekt der periodischen Fortsetzung wird zwar bei der Berechnung ausgenutzt und muß auch bei Anwendung von Filtern berücksichtigt werden, ist aber nicht wirklich sichtbar, da die Matrizen  $\mathbf{X}$  und  $\hat{\mathbf{X}}$  immer nur auf den Bereich des Originalbildes beschränkt sind.

Ein Problem taucht noch auf, wenn man sich die DFT für Matrizen betrachtet. Die Ausgangsmatrix ist bei einem Bild zwar rein reell, für die transformierte Matrix muß dies aber nicht gelten, diese wird in der Regel komplexe Zahlen enthalten. Um die Matrix mit einem Grauwertbild zu identifizieren, ist aber eine Ordnung der Matrixelemente auf einer Skala notwendig, die bei komplexen Werten nicht gegeben ist. Dieses Problem läßt sich aus der Welt schaffen, indem man jeder transformierten Matrix zwei Grauwertbilder zuordnet. Zum einen kann man für den Realteil und den Imaginärteil der Matrix, gebildet aus den Real- bzw. Imaginärteilen aller Elemente, jeweils eine eigene Bildmatrix erzeugen. Hier ergibt sich wieder eine eindeutige Zuordnung von Matrixelement und Grauwert. Die andere Variante ist die Bildung einer Betrags- und Phasenmatrix, d.h. zwei Matrizen, bei der die eine die Beträge aller Matrixelemente, die andere die Argumente<sup>13</sup> aller Matrixelemente enthält.

<sup>13</sup> Ist ein Matrixelement  $0$ , so wird man das entsprechende Element der Argument-Matrix ebenfalls mit dem Wert  $0$  füllen. Eigentlich wäre dieses Matrixelement natürlich undefiniert, da  $\text{Arg } 0$  nicht definiert ist. Bei der Rechnerimplementation muß aber natürlich ein Wert eingesetzt werden. Für die Rücktransformation ist dies ohne Einfluß, da hier das Argument mit dem Betrag  $0$  multipliziert wird, und wieder der korrekte Ausgangswert  $0$  als Ergebnis entsteht.

Die Bildmatrix, die als Ausgangspunkt für die Transformation dient, ist rein reell, kann aber in eine komplexe Matrix eingebettet werden, wodurch die DFT nach obiger Gleichung angewandt werden kann.

### 2.1.6.2 Die inverse diskrete Fouriertransformierte einer Matrix

Wegen der Linearität der diskreten Fouriertransformaten läßt sich die Inverse der diskreten Fouriertransformaten einer Matrix ebenfalls durch Verwendung von zwei Konjugationen und Berechnung der diskreten Fouriertransformaten bilden.

#### Def. Inverse zweidimensionale diskrete Fouriertransformierte

Seien  $\mathbf{X}$  und  $\mathbf{X}^\wedge$  Matrizen mit  $\mathbf{X} \in \mathcal{C}^N \times \mathcal{C}^N, \mathbf{X}^\wedge \in \mathcal{C}^N \times \mathcal{C}^N, N \in \mathbb{N}$ . Dann lautet die inverse diskrete, zweidimensionale Fouriertransformierte zur transformierten Matrix  $\mathbf{X}^\wedge$  mit  $\mathbf{X} = (c_{yx}), \mathbf{X}^\wedge = (c_{vu}^\wedge)$  (mit  $0 \leq x \leq N-1, 0 \leq y \leq N-1$  und  $0 \leq u \leq N-1, 0 \leq v \leq N-1$ ):

$$\begin{aligned} iDFT : \mathcal{C}^N \times \mathcal{C}^N &\rightarrow \mathcal{C}^N \times \mathcal{C}^N \\ \mathbf{X}^\wedge &\mapsto \mathbf{X}, \mathbf{X} = N^2 \cdot \overline{\overline{DFT(\mathbf{X}^\wedge)}} \end{aligned} \quad (36)$$

Inbesondere läßt sich also auch die Methode der Tensorierung bei der Berechnung der Matrix der inversen, diskreten Fouriertransformaten anwenden. Ebenfalls folgt hieraus:

$$iDFT(DFT(\mathbf{X})) = \mathbf{X} \quad (37)$$

was sich aus der Tensorierung der Matrix, der Linearität der eindimensionalen DFT und der Existenz der inversen Abbildung zur DFT ergibt.

Die Zuordnung der Matrix  $\mathbf{X}$  zu einer Bildmatrix kann wie zuvor beschrieben erfolgen.

### 2.1.6.3 Tensorierte Filterungen bei Matrizen

Durch den Vorteil, daß sich die Transformation einer Matrix auf die Transformation der Zeilen- und Spaltenvektoren zurückführen läßt, ergibt sich ein ähnlicher Vorteil bei der Filterung. Es ist nämlich möglich, alle zuvor definierten Filterfunktionen unmittelbar auf die Matrizen anzuwenden –mathematisch und algorithmisch betrachtet ein sehr zweckm äßiger Umstand.

Dies liegt daran, daß in einer transformierten Matrix ebenfalls tiefe und hohe Frequenzen vertreten sind, allerdings gibt es jetzt diese Frequenzen in Zeilen- und Spaltenrichtung. Die Filterung der

Zeilen- und Spaltenfrequenzen ist jeweils aber ein eindimensionaler Vorgang, so daß die bereits vorgestellten Filter zur Anwendung kommen können.

Üblicherweise verbindet man mit dem Begriff der *Frequenz* eine zeitliche periodische Änderung einer Größe. Bei einem statischen Bild kann diese Bezeichnung in die Irre führen, da es hier keine zeitlichen Änderungen, sondern im Frequenzbereich nur Änderungen pro Pixel bzw. Längeneinheit gibt. Um Verwechslungen zu vermeiden, führen [HAB-91] und [KOR-82] den Begriff der *Ortsfrequenz* ein. Die Ortsfrequenz beschreibt in diesem Zusammenhang eine Änderung der Bildinformation pro Zeilen- oder Spalteneinheit. Da im hier vorgestellten Kontext Verwechslungen nicht möglich sind, werden die Begriffe der Frequenz und Ortsfrequenz teilweise synonym verwendet.

### Def. Algorithmus für Filterung von Bildmatrizen

Seien  $\mathbf{X}$  und  $\mathbf{Y}$  Bildmatrizen mit  $\mathbf{X} \in \mathcal{C}^N \times \mathcal{C}^N$ ,  $\mathbf{Y} \in \mathcal{C}^N \times \mathcal{C}^N$  und  $\mathbf{X}'$  eine Hilfsmatrix mit  $\mathbf{X}' \in \mathcal{C}^N \times \mathcal{C}^N$ , sowie  $N \in \mathbb{N}$ . Außerdem sei  $H \in \mathbb{R}^N$  ein Filtervektor, wie er für die eindimensionale Filterung definiert wurde.

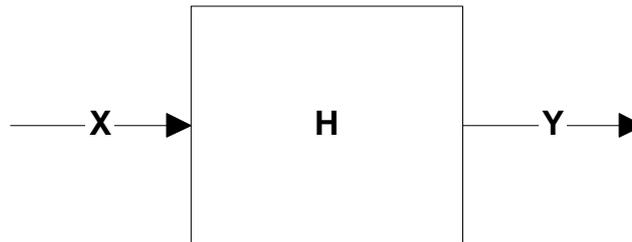


Abbildung 19: Filtersystem für Bildmatrizen

Für die nach Abbildung 19 gefilterte Bildmatrix  $\mathbf{Y}$  gilt dann:

$$x'_{mn} = x_{mn} \cdot H(m) \quad \forall 0 \leq n \leq N-1 \wedge 0 \leq m \leq N-1$$

$$y_{mn} = x'_{mn} \cdot H(n) \quad \forall 0 \leq n \leq N-1 \wedge 0 \leq m \leq N-1$$

Da für die diskrete Fouriertransformation eines Vektors bereits Filtervektoren definiert wurden, können nun direkt Tiefpaß- und Hochpaßfilter für transformierte Bildmatrizen angegeben werden, für den Hochpaß gibt es ebenfalls wieder die Variante mit Gleichanteil im Durchlaßbereich. Verwendung finden dabei die Filtervektoren  $H_{LP}$ ,  $H_{HP}$  und  $H_{HPDC}$ , wie sie zuvor definiert wurden.

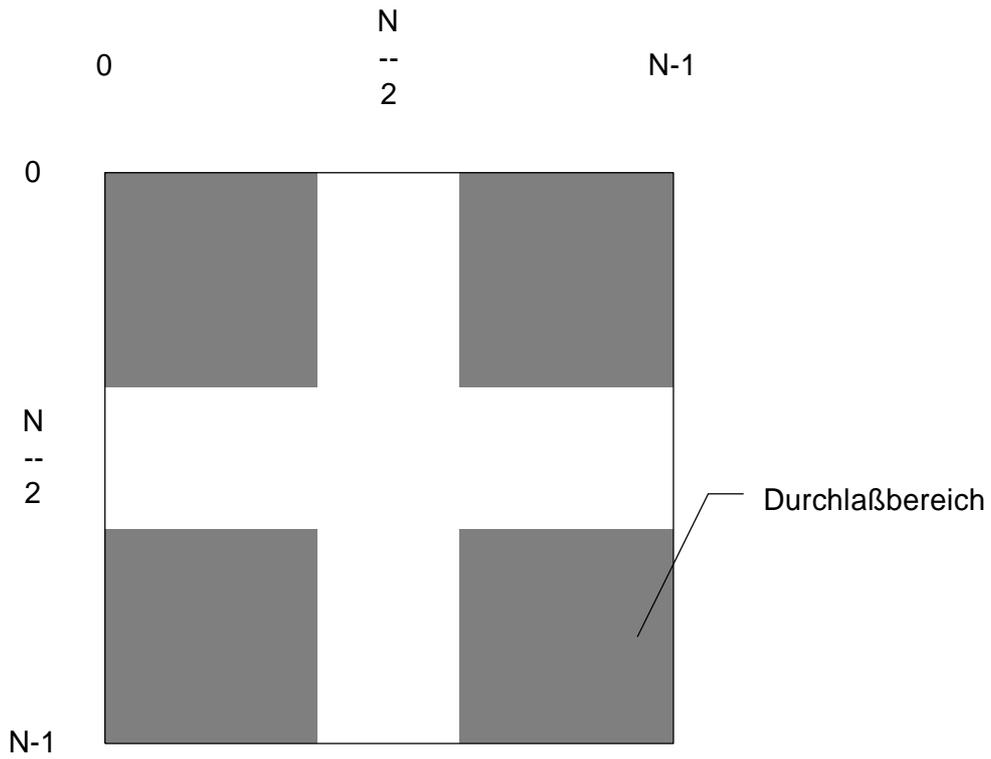


Abbildung 20: Durchlaßbereich des Tiefpaßfilters für Bildmatrizen

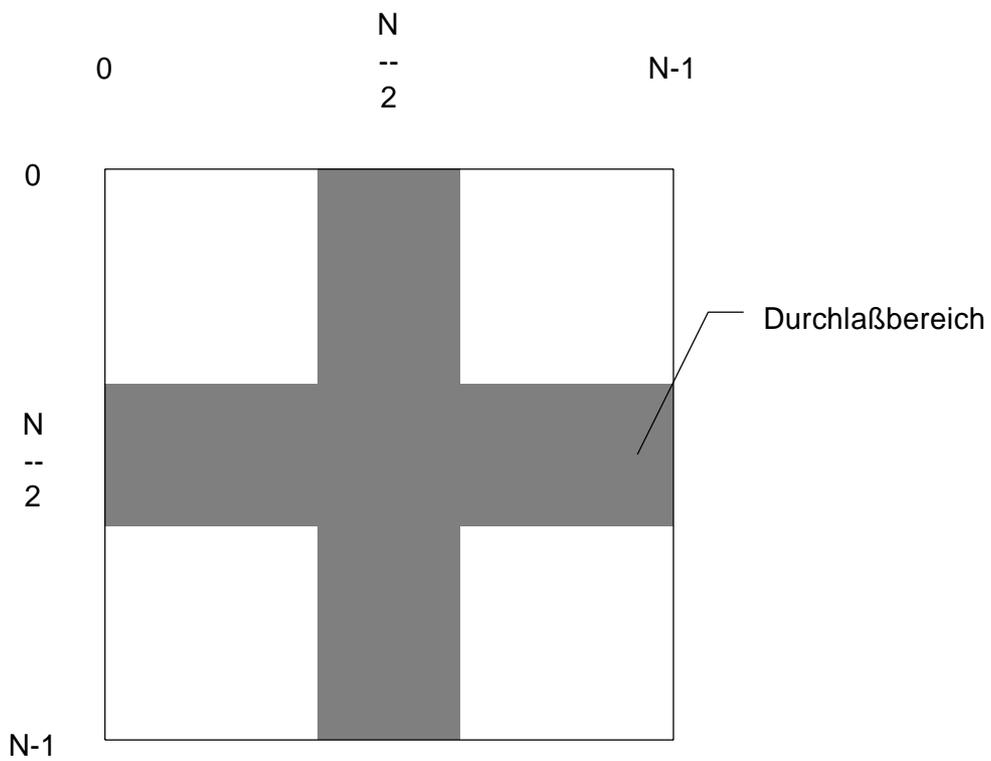


Abbildung 21: Durchlaßbereich des Hochpaßfilters für Bildmatrizen

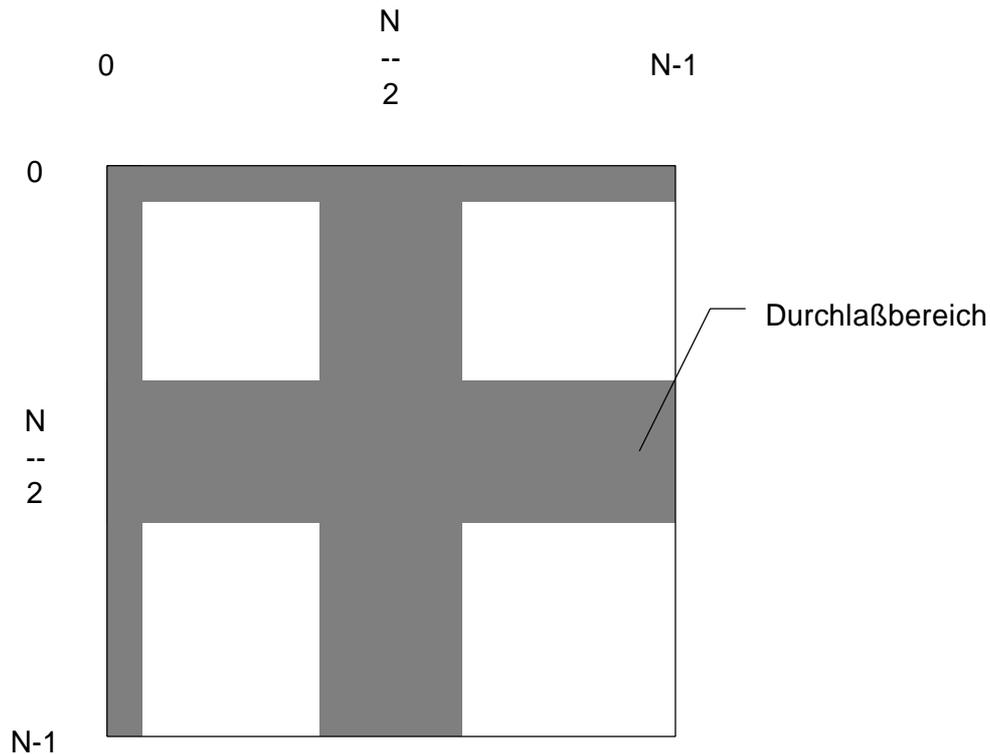


Abbildung 22: Durchlaßbereich des Hochpaßfilters für Bildmatrizen mit Durchlaß des Gleichanteils

Wie man in den Abbildungen erkennt, muß auch hier dem Aliaseffekt Rechnung getragen werden, die hohen Frequenzen liegen in der „Mitte“ der Matrix. Überhaupt ist die Interpretation der Matrixkoeffizienten etwas schwieriger, dies sieht man auch am Gleichanteil in Abbildung 22. Der Gleichanteil ist hier nicht nur ein einziger Koeffizient wie beim Vektor, sondern der Rand der Matrix. Alle Koeffizienten, bei denen eine der beiden Koordinaten gleich  $0$  ist, gehören zum Gleichanteil. Dies liegt daran, daß man bei der Matrix Frequenzen nach Spalten und Zeilen unterscheiden muß, es gibt also eine zusätzliche Richtungsabhängigkeit der Frequenz und dementsprechend auch einen Gleichanteil in jede Richtung (siehe auch [HAB-91] oder [STE-95]).

## 2.2 Die diskrete Cosinustransformation

### 2.2.1 Zusammenhang mit der diskreten Fouriertransformation

Bei der komplexen Fourierreihe einer auf einem Intervall definierten und periodisch fortgesetzten Funktion erhält man im allgemeinen Fall auch tatsächlich komplexe Fourierkoeffizienten. Setzt man die Funktion jetzt so periodisch fort, daß insgesamt eine gerade periodische Funktion entsteht, so werden alle Fourierkoeffizienten der komplexen Fourierreihe rein reell, d.h. für alle Koeffizienten verschwindet der Imaginärteil. Diese Vorgehensweise kann man sich auch für die diskrete Fouriertransformation vorstellen, da hier die Matrix nach der Transformation ebenfalls komplexe

Werte aufweisen wird. Dies hat zwar keinen Einfluß auf die Berechnungen, verschlechtert aber die Visualisierung: das Spektrum kann immer nur in Form von mindestens zwei Bildern dargestellt werden, nach Real- und Imaginärteil oder nach Betrag und Phase. Günstiger wäre es, wenn man unmittelbar nur eine reelle Matrix als Ergebnis der Transformation erhält. Das dazu notwendige Hilfsmittel ist die *diskrete Cosinustransformation*.

Man muß bei der diskreten Cosinustransformation, kurz *DCT*, vier verschiedene Variationen unterscheiden (z.B. [WIC-96]): die sogenannten DCT-I, DCT-II, DCT-III und DCT-IV. Diese Variationen unterscheiden sich im wesentlichen durch die Wahl der Basisvektoren der Transformation, lassen sich aber alle ineinander überführen. In dieser Arbeit wurde die DCT-II verwendet, vor allem daher, weil das JPEG-Dateiformat ebenfalls mit der DCT-II arbeitet und die üblicherweise in der Literatur zu findende DCT eigentlich eine DCT-II ist. Im folgenden werden die Bezeichnungen DCT und DCT-II synonym verwendet.

Ebenfalls trifft man in der Literatur (ebenfalls [WIC-96]) noch auf eine *diskrete Sinustransformation*, die sich aber von der Cosinustransformation nur durch eine andere Phasenlage unterscheidet – hier wird das Ergebnis rein imaginär, was für die Verarbeitung also keinerlei Vorteile bringt. Daher wird hierauf nicht eingegangen.

#### Def. Diskrete Cosinustransformation

Seien  $F \in \mathbb{R}^N$ ,  $f \in \mathbb{R}^N$ ,  $N \in \mathbb{N}$  Vektoren, sowie  $N$  gerade. Weiter gelte für die Hilfsvektoren  $q \in \mathbb{R}^{2N}$ ,  $Q \in \mathbb{R}^{2N}$ :

$$q_k = \begin{cases} f_k & , 0 \leq k < N \\ f_{2N-k-1} & , N \leq k < 2N \end{cases}$$

und

$$Q = DFT(q)$$

Dann gilt für die diskrete Cosinustransformation

$$DCT : \mathbb{R}^N \rightarrow \mathbb{R}^N$$

$$f \mapsto F : F_n = \begin{cases} \sqrt{2} \operatorname{Re}\{Q_0\} & , k = 0 \\ \operatorname{Re}\left\{Q_k e^{-j\frac{k}{2N}p}\right\} + \operatorname{Re}\left\{Q_{2N-k-1} e^{j\frac{k}{2N}p}\right\} & , 1 \leq k < N \end{cases}$$

(38)

(siehe dazu [WIC-96] im Abschnitt über Algorithmen)

Diese Darstellung ist evtl. etwas ungewöhnlich, da man meistens nur die Definition mit Hilfe der Basisvektoren findet, z.B. in [STE-95]. In Anbetracht der geplanten Anwendung ist obige Darstellung allerdings interessanter, da die DCT unmittelbar auf die DFT zurückgeführt werden kann – eine effektive Implementation der DFT kann somit unmittelbar auch für eine effektive Implementation der DCT verwendet werden.

Betrachtet man den Aufbau der DCT, so sieht man, daß im Gegensatz zur DFT nur der halbe Frequenzbereich abgedeckt wird, es gilt nämlich:

Satz: Koeffizienten im Frequenzbereich

Seien  $X$  und  $x$  reelle Vektoren der Länge  $N \in \mathbb{N}$ , für die gilt  $X = DCT(x)$ , sowie  $T_{ab} \in \mathbb{R}$  die zugehörige Abtastzeit. Dann gehört zum  $k$ -ten Koeffizienten  $X_n$  die Frequenz  $f$

$$f = \frac{2k}{N \cdot T_{ab}}, 0 \leq k < N$$

(39)

Setzt man den Vektor  $X$  nun periodisch auf  $\mathbb{Z}$  fort, so kann man den Zusammenhang zwischen Index und Frequenz wieder erweitern:

$$f = \frac{2(k \bmod N)}{N \cdot T_{ab}}, k \in \mathbb{Z}$$

(40)

Der Beweis ergibt sich unmittelbar aus der Herleitung der DCT-Bestimmungsgleichungen.

Wie man sieht, resultiert aus dieser Darstellung eine Verdoppelung der Auflösung im Frequenzbereich, da bei der DFT  $N$  Punkte für das Frequenzband  $[0, f_{ab})$  zur Verfügung stehen, während bei der DCT ebenfalls  $N$  Punkte für die „Hälfte“, also den Bereich  $[0, \frac{f_{ab}}{2})$  verwendet werden. Dadurch ergibt sich bei der DCT zusätzlich der Vorteil, daß der Aliaseffekt in der unteren und oberen Vektorhälfte nicht zum tragen kommt, da keine Spiegeeffekte im Bereich  $[0, \frac{f_{ab}}{2})$  auftreten.

Eine weitere Eigenschaft wird sich positiv bemerkbar machen, nämlich daß der transformierte Vektor rein reell ist – dies wird bei der Anwendung auf Matrizen eine einfachere grafische Darstellung ermöglichen.

#### Def. Inverse diskrete Cosinustransformation

Seien  $F \in \mathbb{R}^N$ ,  $f \in \mathbb{R}^N$ ,  $N \in \mathbb{N}$  Vektoren, sowie  $N$  gerade. Weiter gelte für die Hilfsvektoren  $q \in \mathbb{R}^{2N}$ ,  $Q \in \mathbb{R}^{2N}$ :

$$q_k = \frac{1}{2} \begin{cases} \frac{2}{\sqrt{2}} F_0 & , k = 0 \vee k = N \\ F_k e^{j \frac{k}{2N} p} & , 1 \leq k < N \\ F_{2N-k} e^{-j \frac{2N-k}{2N} p} & , N < k < 2N \end{cases}$$

und

$$q = iDFT(Q)$$

Dann gilt für die inverse diskrete Cosinustransformation

$$iDCT : \mathbb{R}^N \rightarrow \mathbb{R}^N$$

$$F \mapsto f : f_k = q_k, 0 \leq k < N \quad (41)$$

([WIC-96] im Abschnitt über Algorithmen)

Verkettet man die Definitionsgleichungen der DCT und der iDCT miteinander, so folgt nach einigen kleinen Umformungen die wichtige Gleichung

$$iDCT(DCT(f)) = f \quad (42)$$

die im Zweidimensionalen natürlich genauso wie im Eindimensionalen gelten muß.

### 2.2.2 Filterungen bei der DCT

Im Grunde gilt für die Filterung bei der DCT das gleiche, wie es für die DFT bereits gesagt wurde. Allerdings vereinfacht sich die Darstellung von Durchlaß- und Sperrbereich, da bei der DCT die Koeffizienten mit höherem Index tatsächlich auch zu den physikalisch hohen Frequenzen gehören – der Aliaseffekt fällt ja nun weg. Wichtig ist ebenso, daß sich die Gültigkeit der Faltungssätze auch auf diskrete Systeme überträgt, die mit Hilfe der diskreten Cosinustransformation beschrieben werden. Dies ergibt sich im wesentlichen aus dem Zusammenhang zwischen DCT und DFT und ist Grundvoraussetzung für die Art und Weise, wie bei der DCT die Filterung eines Vektors vorgenommen werden kann.

#### 2.2.2.1 Tiefpaßfilter

Für den Durchlaßbereich eines idealen Tiefpaßfilters erhält man bei der DCT das in der folgenden Abbildung gezeigte Übertragungsverhalten.

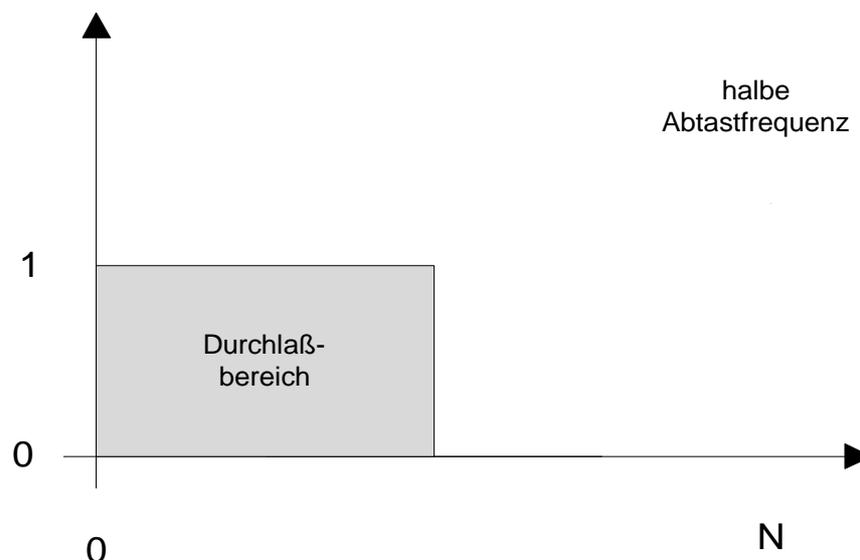


Abbildung 23: Prinzipdarstellung des Durchlaßbereichs eines Tiefpaßfilters für die DCT

### Def. Tiefpaßfilter im Frequenzbereich für DCT

Sei  $x \in \mathbb{R}^N$  der Eingangsvektor eines diskreten Systems,  $y \in \mathbb{R}^N$  der Ausgangsvektor, weiterhin gilt  $N \in \mathbb{N}$  und  $N$  ist gerade. Dann heißt ein Vektor  $H_{LP} \in \mathbb{R}^N$  ein *idealer Tiefpaß* mit der Grenzfrequenz  $g, g \in \mathbb{N}, 0 \leq g \leq N$ , wenn gilt:

$$H_{LP}[n] = \begin{cases} 1 & , n \leq g \\ 0 & , \text{sonst} \end{cases} \quad (43)$$

Man sieht, daß die Definition des Tiefpaßfilters einen Durchlaßbereich gemäß Abbildung 23 beschreibt.

#### 2.2.2.2 Hochpaßfilter

Ein komplementäres Filterverhalten zum Tiefpaß stellt wieder der Hochpaß dar, bei dem die niedrigen Frequenzen gesperrt und die hohen Frequenzen durchgelassen werden.

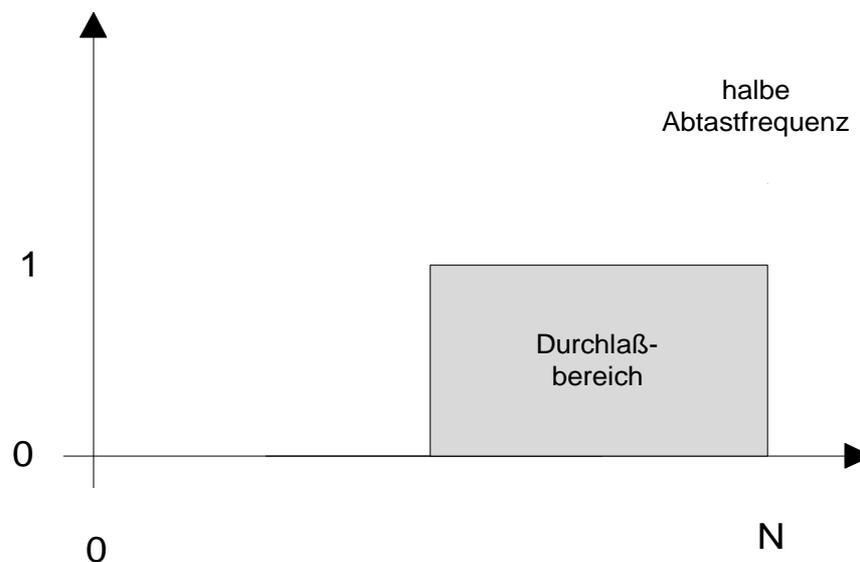


Abbildung 24: Prinzipdarstellung des Durchlaßbereichs eines Hochpaßfilters für die DCT

### Def. Hochpaßfilter im Frequenzbereich für DCT

Sei  $x \in \mathcal{R}^N$  der Eingangsvektor eines diskreten Systems,  $y \in \mathcal{R}^N$  der Ausgangsvektor, weiterhin gilt  $N \in \mathcal{N}$  und  $N$  ist gerade. Dann heißt ein Vektor  $H_{HP} \in \mathcal{R}^N$  ein *idealer Hochpaß* mit der Grenzfrequenz  $g, g \in \mathcal{N}, 0 \leq g \leq N$ , wenn gilt:

$$H_{HP}[n] = \begin{cases} 1 & , n \geq g \\ 0 & , \text{sonst} \end{cases} \quad (44)$$

Aus den gleichen Gründen wie bei der DFT ist es sinnvoll, für den Hochpaß noch zusätzlich einen Filter einzuführen, bei dem der Gleichanteil zusätzlich zum Durchlaßbereich der hohen Frequenzen hinzugenommen wird.

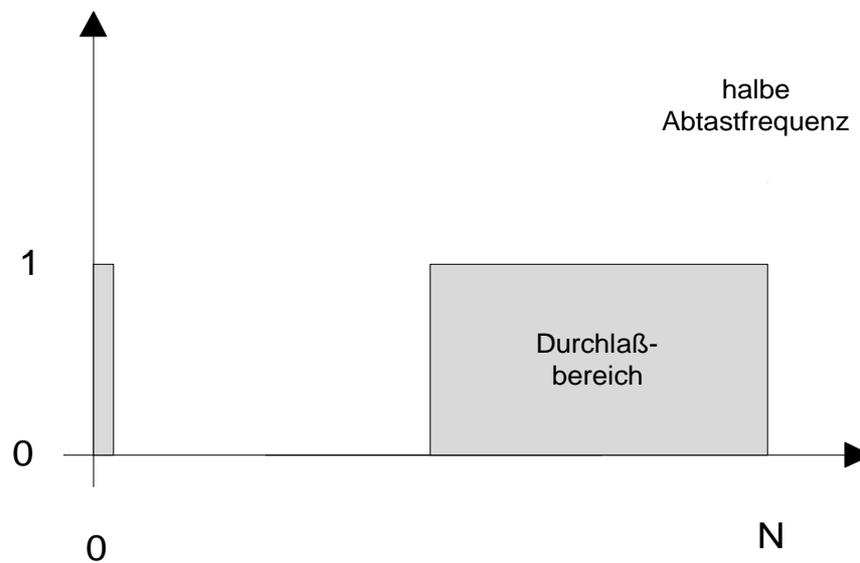


Abbildung 25: Prinzipdarstellung des Durchlaßbereichs eines Hochpaßfilters für die DCT, bei dem auch der Gleichanteil zum Durchlaßbereich gehört

### Def. Hochpaßfilter mit Gleichanteil im Frequenzbereich für DCT

Sei  $x \in \mathcal{R}^N$  der Eingangsvektor eines diskreten Systems,  $y \in \mathcal{R}^N$  der Ausgangsvektor, weiterhin gilt  $N \in \mathcal{N}$  und  $N$  ist gerade. Dann heißt ein Vektor  $H_{HPDC} \in \mathcal{R}^N$  ein *idealer Hochpaß mit zugeschaltetem Gleichanteil* mit der Grenzfrequenz  $g, g \in \mathcal{N}, 0 \leq g \leq N$ , wenn gilt:

$$H_{HPDC}[n] = \begin{cases} 1 & , n \geq g \vee n = 0 \\ 0 & , \text{sonst} \end{cases} \quad (45)$$

### 2.2.3 Erweiterung auf den zweidimensionalen Fall

Der Zusammenhang mit der diskreten Fouriertransformation ermöglicht eine direkte Übertragung der Erkenntnisse aus der zweidimensionalen DFT und der zugehörigen Filterungen auf die DCT, in diesem Abschnitt werden keine neuen Aussagen gemacht, sondern nur die entsprechenden Methoden der DFT auf die DCT übertragen.

#### 2.2.3.1 Die Transformation und ihre inverse Transformation

Sowohl die diskrete, als auch die inverse diskrete Cosinustransformation können wieder durch eine Tensorierung der Matrix ausgeführt werden, also durch eine aufeinanderfolgende Transformation der Spalten- und Zeilenvektoren.

#### Def. Zweidimensionale (inverse) diskrete Cosinustransformation

Seien  $\mathbf{X}$  und  $\mathbf{Y}$  Matrizen mit  $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^N$ ,  $\mathbf{Y} \in \mathbb{R}^N \times \mathbb{R}^N$  und  $\mathbf{X}'$  eine Hilfsmatrix mit  $\mathbf{X}' \in \mathbb{R}^N \times \mathbb{R}^N$ , sowie  $N \in \mathbb{N}$ . Dann gilt für die zweidimensionale diskrete Cosinustransformierte von  $\mathbf{X}$

$$\begin{aligned} DCT : \mathbb{R}^N \times \mathbb{R}^N &\rightarrow \mathbb{R}^N \times \mathbb{R}^N \\ \mathbf{X} &\mapsto \mathbf{Y} : \mathbf{Y} = DCT(\mathbf{X}) \end{aligned} \quad (46)$$

wobei für die Zeilen- und Spaltenvektoren der Matrizen gilt:

$$\begin{aligned} \begin{pmatrix} x'_{0,n} \\ \vdots \\ x'_{N-1,n} \end{pmatrix} &= DCT \left( \begin{pmatrix} x_{0,n} \\ \vdots \\ x_{N-1,n} \end{pmatrix} \right) \quad \forall 0 \leq n \leq N-1 \\ (y_{m,0} \quad \cdots \quad y_{m,N-1}) &= DCT((x'_{m,0} \quad \cdots \quad x'_{m,N-1})) \quad \forall 0 \leq m \leq N-1 \end{aligned}$$

Umgekehrt gilt für die inverse Transformation

$$\begin{aligned} iDCT : \mathbb{R}^N \times \mathbb{R}^N &\rightarrow \mathbb{R}^N \times \mathbb{R}^N \\ \mathbf{Y} &\mapsto \mathbf{X} : \mathbf{X} = iDCT(\mathbf{Y}) \end{aligned} \quad (47)$$

mit den Zeilen- und Spaltenvektoren:

$$\begin{pmatrix} x'_{0,n} \\ \vdots \\ x'_{N-1,n} \end{pmatrix} = iDCT \begin{pmatrix} y_{0,n} \\ \vdots \\ y_{N-1,n} \end{pmatrix} \quad \forall 0 \leq n \leq N-1$$

$$(x_{m,0} \quad \cdots \quad x_{m,N-1}) = iDCT((x'_{m,0} \quad \cdots \quad x'_{m,N-1})) \quad \forall 0 \leq m \leq N-1$$

### 2.2.3.2 Filterungen der DC-transformierten Matrix

Für eine DC-transformierte Matrix kann eine Filterung in der gleichen Art und Weise ausgeführt werden, wie im Abschnitt 2.1.6.3 (Seite 30) dies bereits für die DFT beschrieben wurde. Daher ist es ausreichend, wenn zur Verdeutlichung nur noch die Durchlaßbereich der verschiedenen Filter für die DCT dargestellt werden. Einziger Unterschied ist, daß die Berechnung mit reellen Zahlen erfolgt, und nicht mit komplexen, was aber vom Ablauf her nichts an der Berechnung und am Algorithmus ändert.

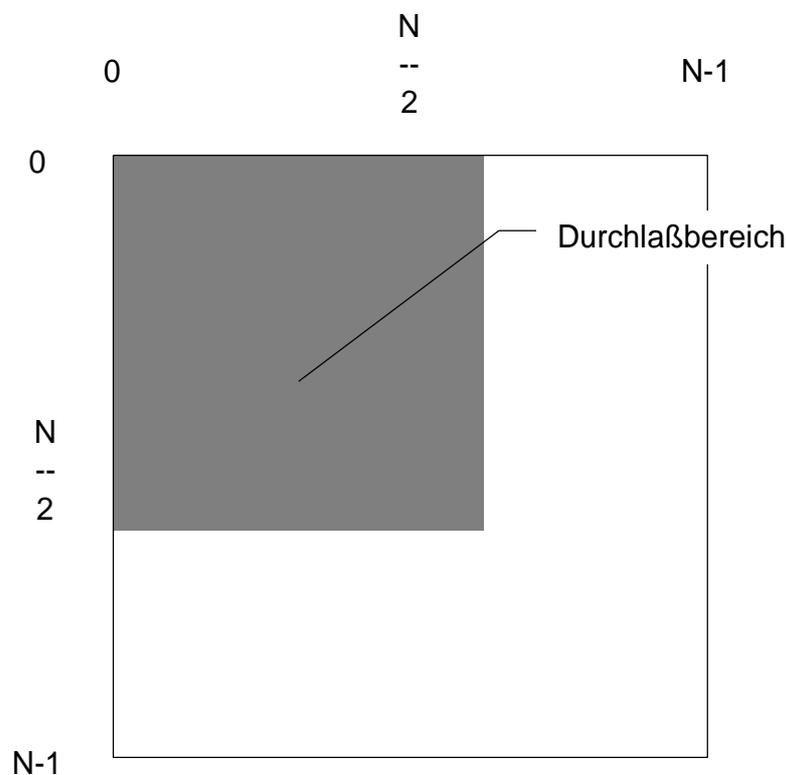


Abbildung 26: Durchlaßbereich des Tiefpaßfilters für Bildmatrizen bei der DCT

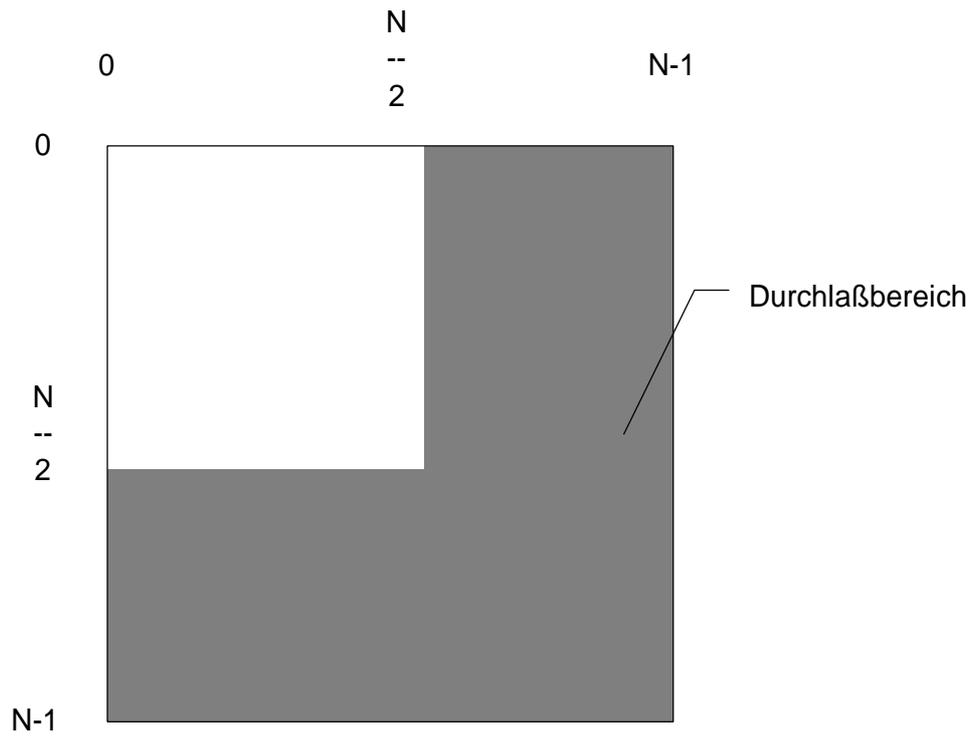


Abbildung 27: Durchlaßbereich des Hochpaßfilters für Bildmatrizen bei der DCT

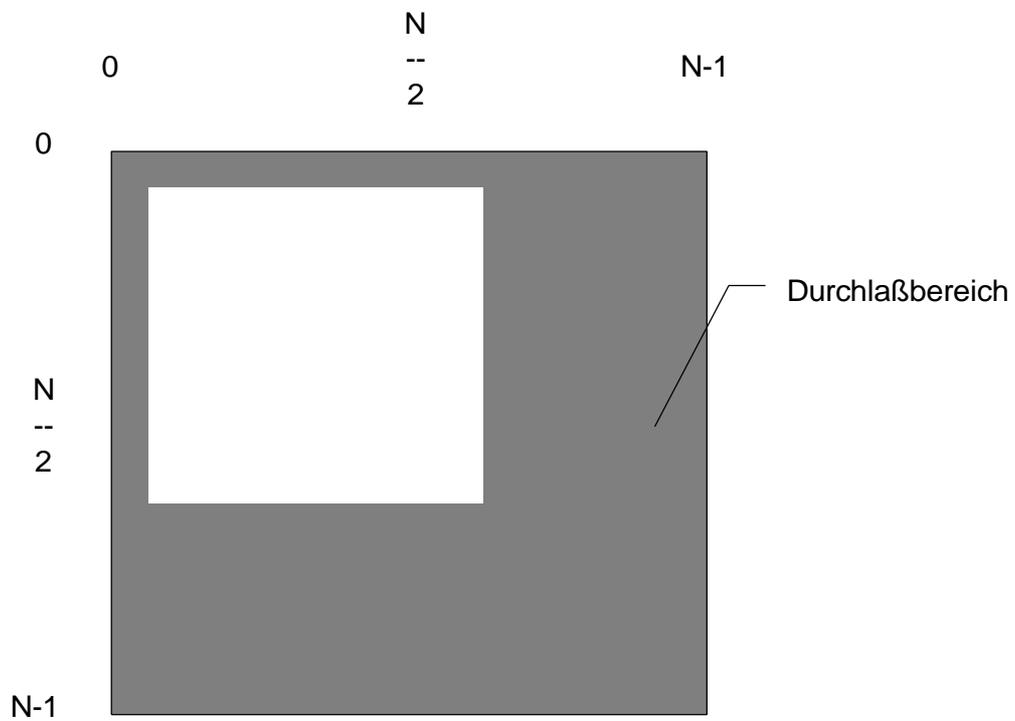


Abbildung 28: Durchlaßbereich des Hochpaßfilters mit Gleichanteil für Bildmatrizen bei der DCT

## 2.3 Die Fensterung von Fourier- und Cosinustransformationen

### 2.3.1 Prinzip und Motivation für gefensterete Transformationen

Betrachtet man einen Vektor mit einer größeren Anzahl von Elementen, so wird es immer wieder Vektoren geben, die – aus Sicht der enthaltenen Informationen – aus mehreren Teilen bestehen: scharfe Übergänge der Werte in dem einen Teil und nur geringe Schwankungen der Werte in einem anderen Teil.

Aus Sicht der Fouriertransformation sieht dies stark vereinfacht so aus, daß an einem bestimmten Punkt im Zeitbereich scharfe Übergänge vorkommen, die sich im Frequenzbereich als hohe Frequenzen ausdrücken. Betrachte man eine Zeitfunktion, die „fast“ einen verschobenen Delta-Impuls darstellt, also noch einen endlichen Wert annimmt, und transformiert diese Funktion, so ist die Energie im Frequenzbereich über alle Frequenzen „fast gleichmäßig“ verteilt. Und dies, obwohl die Zeitfunktion lokal begrenzt und außerhalb des Peaks überall identisch  $0$  ist. Dies liegt daran, daß die Sinus- und Cosinusfunktionen eine „unendliche Länge“ besitzen, das darzustellende Signal aber beliebig kurz sein kann. In einem solchen Fall sind eben sehr viele Sinus- und Cosinusfunktionen mit entsprechend großer Amplitude notwendig, um für dieses scharfe und kurze Signal Konvergenz zu erzielen. Abbildung 30 verdeutlicht diesen Effekt, der hier allerdings nur stark vereinfacht dargestellt wurde, um die Probleme zu verdeutlichen.

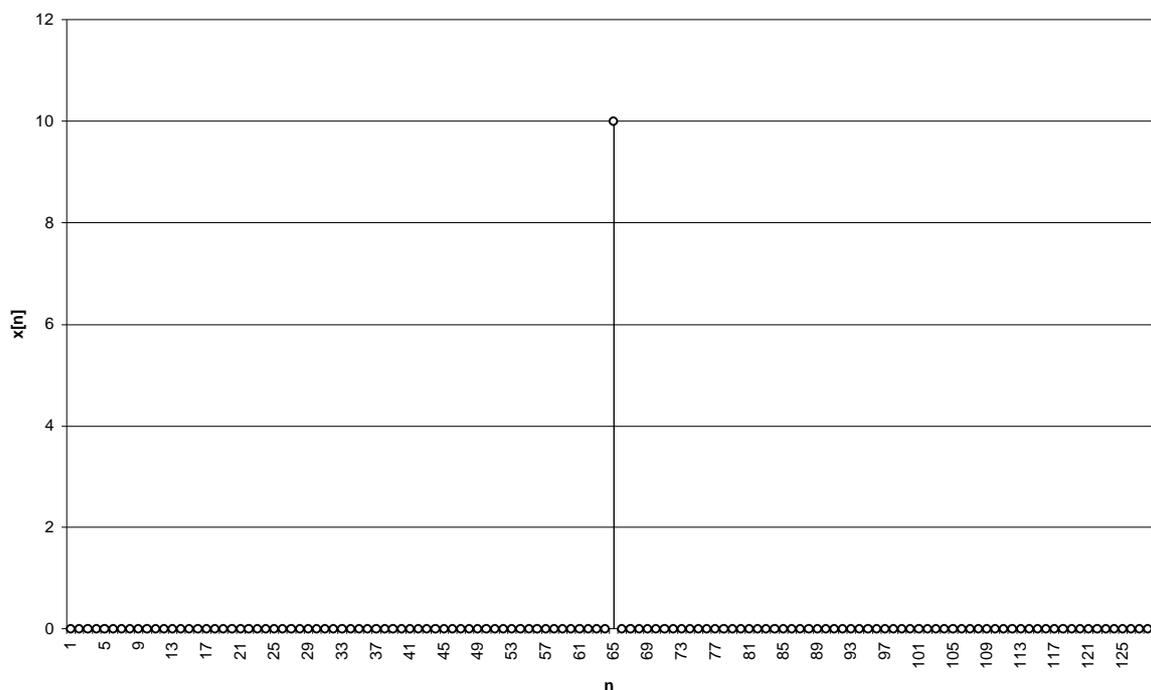


Abbildung 29: Ein „Peak“, also eine Impulsfolge, im Zeitbereich

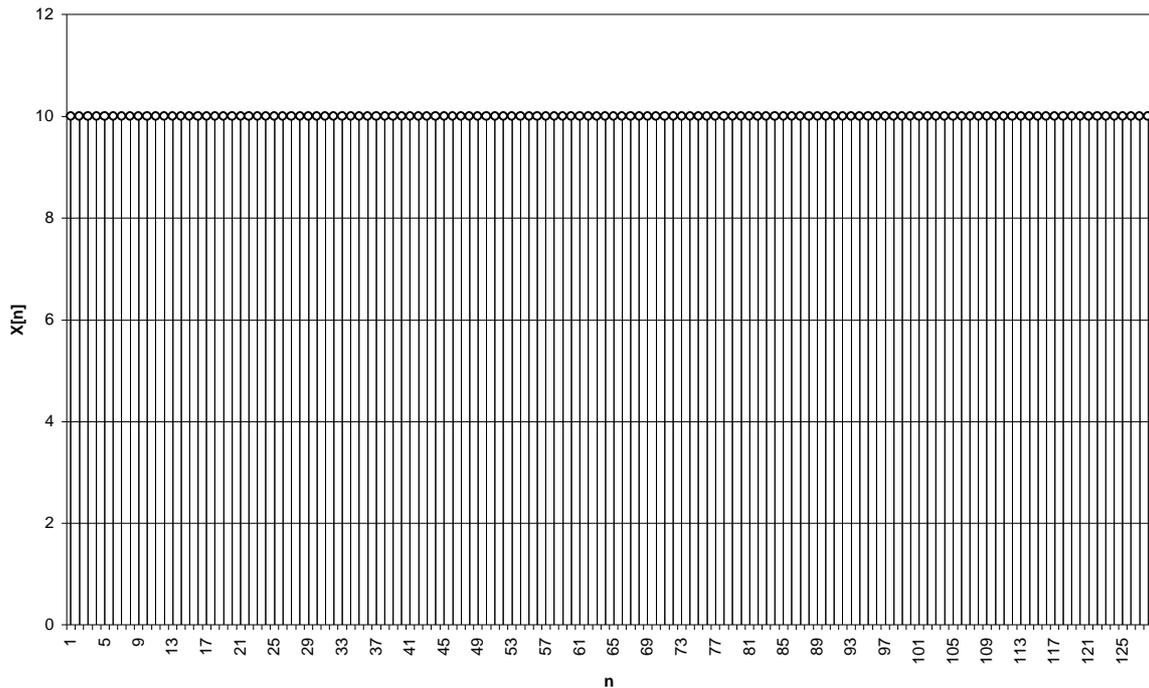


Abbildung 30: Der Betrag der diskreten Fouriertransformierten der Impulsfolge

Übertragen auf die Transformation von Bildern lautet die Frage, wieso man eigentlich für ein ganzes Bild nur einen einzigen Gleichanteil erhält, schließlich besteht ein Bild im Normalfall ja aus verschiedenen Farbbereichen, die alle einen anderen Gleichanteil besitzen. Wird die Darstellung nicht besser, wenn man diese Farbbereiche auch bei der Filterung trennt, so daß den lokalen Bildeigenschaften mehr Rechnung getragen wird?

Ein alternativer Lösungsansatz ist, das Zeitsignal in verschiedene Segment aufzuteilen und die Frequenzen nach Segmenten getrennt zu analysieren. Dadurch werden nur in Abschnitten, in denen tatsächlich scharfe Übergänge und Peaks vorliegen, die hohen Frequenzen erscheinen, die Abschnitte enthalten jeweils nur die Frequenzen, die auch tatsächlich in diesem Bereich vorkommen. Um dies im Vergleich zur kompletten Transformation zu verdeutlichen, ist die Transformation der Folge aus Abbildung 29 erneut durchgeführt worden, allerdings wurden nun vier Transformationen für jeweils ein Viertel der Ausgangsfolge durchgeführt.

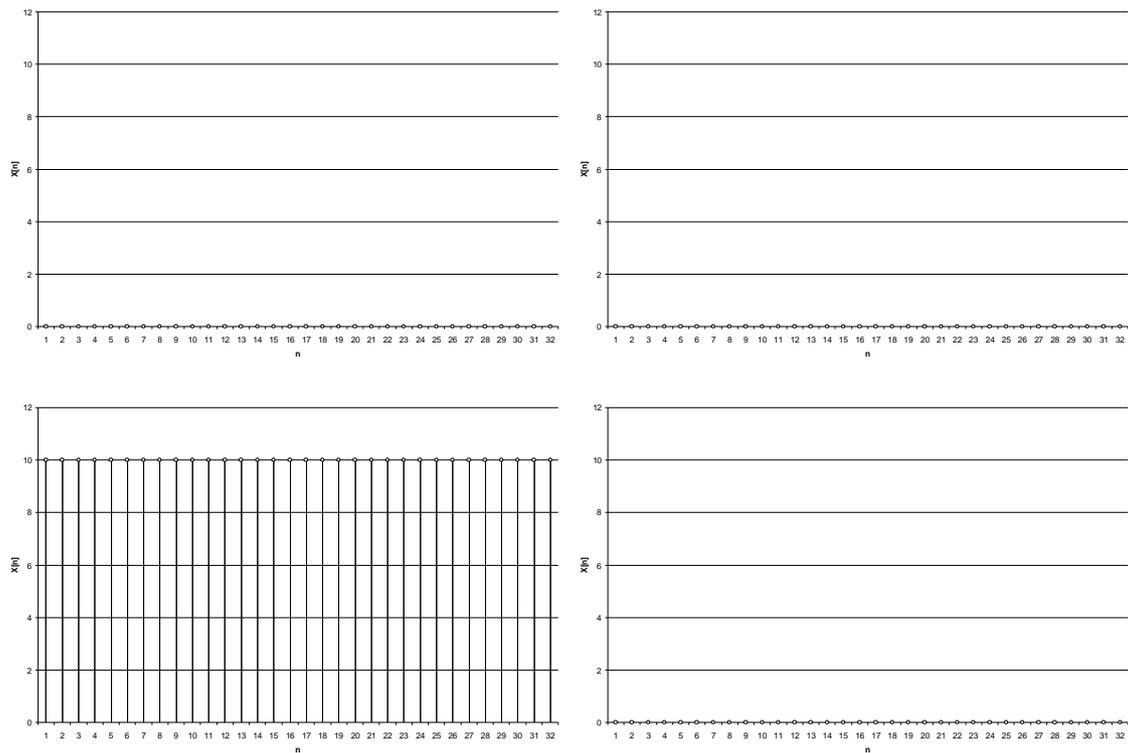


Abbildung 31: Wird die Folge aus Abbildung 29 in Viertel zerlegt und getrennt transformiert, so entstehen die vier gezeigten Fouriertransformierten (es sind nur die Beträge dargestellt).

Wie man sieht, ist es wohl tatsächlich möglich, die Frequenzen zu lokalisieren und somit bestimmte Teile der Signale im Zeitbereich getrennt zu analysieren. Man sollte sich allerdings auch über die Nachteile bewußt sein: es ist zu erwarten, daß das Zusammensetzen nach der Rücktransformation und getrennten Verarbeitung zu einem Vektor der ursprünglichen Länge an den Nahtstellen für den allgemeinen Fall einige Probleme aufwerfen wird. Schließlich setzen die diskrete Fourier- und Cosinustransformation ein Zeitsignal mit positivem Träger periodisch fort, was aber für den Fall der Transformation des ganzen Vektors in anderer Form geschieht als bei der Transformation der Teilvektoren. Für ein beliebiges Signal wird man an den Übergangsstellen nach einer Filterung wohl Sprünge und Störungen erhalten, die bei der Transformation des gesamten Vektors nicht auftreten.

Weiterhin verliert man im Frequenzbereich an Auflösung, da das gleiche Frequenzband nun auf weniger Koeffizienten verteilt ist. Das Frequenzband bleibt durch die Aufteilung im Zeitbereich unverändert, da sich die Abtastzeit nicht ändert, aber die spektrale Auflösung wird entsprechend der Anzahl der *Fenster* im Zeitbereich vermindert.

Die Aufteilung des Vektors im Zeitbereich in gleich große Teilvektoren nennt man auch *Fensterung* die Teilvektoren werden als *Fenster* oder auch *Zeitfenster* bezeichnet. Ein großer Anwendungsbereich

dieser Methode ergibt sich vor allem in der Echtzeitverarbeitung von Audiosignalen, wo diese Methode hauptsächlich eingesetzt wird.

Weitere Informationen hierzu findet man auch in [AMA-98] und [HUB-97], wo auch auf das grundsätzliche Problem eingegangen wird, daß die Ausschnittgröße im Zeitbereich und die spektrale Auflösung miteinander gegensätzlich verknüpft sind, eine Aussage, die ganz wesentlich mit der *Heisenbergschen Unschärferelation* zusammenhängt.

### 2.3.2 Die Fensterung im zweidimensionalen Fall

Die Fensterung einer Matrix wird so ausgeführt, daß man das Bild in entsprechende Teilmatrizen zerlegt und dies jeweils einer zweidimensionalen Transformation unterwirft. Es ergibt sich

#### Def. Gefensterter Transformation im zweidimensionalen Fall

Seien  $\mathbf{X}$  und  $\mathbf{Y}$  Matrizen mit  $\mathbf{X} \in \mathcal{C}^N \times \mathcal{C}^N$ ,  $\mathbf{Y} \in \mathcal{C}^N \times \mathcal{C}^N$ , sowie  $N \in \mathcal{N}$ . Außerdem sei  $F \in \mathcal{N}$  die Fenstergröße, wobei  $N_f = \frac{N}{F} \in \mathcal{N}$  gelte. Die Abbildung  $T$  sei entweder eine zweidimensionale diskrete Fourier- oder Cosinustransformation, die Abbildung  $iT$  ihre zugehörige Inverse. Dann gilt für die zweidimensionale gefensterter Transformierter von  $\mathbf{X} = (x_{m,n})$ ,  $0 \leq n < N$ ,  $0 \leq m < N$ :

$$\begin{aligned} wT : \mathcal{C}^N \times \mathcal{C}^N &\rightarrow \mathcal{C}^N \times \mathcal{C}^N \\ \mathbf{X} &\mapsto \mathbf{Y} : \mathbf{Y} = wT(\mathbf{X}) \end{aligned} \quad (48)$$

mit

$$\begin{pmatrix} y_{0+nF,0+mF} & \cdots & y_{0+nF,F-1+mF} \\ \vdots & & \vdots \\ y_{F-1+nF,0+mF} & \cdots & y_{F-1+nF,F-1+mF} \end{pmatrix} = T \left( \begin{pmatrix} x_{0+nF,0+mF} & \cdots & x_{0+nF,F-1+mF} \\ \vdots & & \vdots \\ x_{F-1+nF,0+mF} & \cdots & x_{F-1+nF,F-1+mF} \end{pmatrix} \right)$$

$$, 0 \leq n < N_f, 0 \leq m < N_f$$

Umgekehrt gilt für die Inverse

$$\begin{aligned} iwT : \mathcal{C}^N \times \mathcal{C}^N &\rightarrow \mathcal{C}^N \times \mathcal{C}^N \\ \mathbf{Y} &\mapsto \mathbf{X} : \mathbf{X} = iwT(\mathbf{Y}) \end{aligned} \quad (49)$$

mit

$$\begin{pmatrix} x_{0+nF,0+mF} & \cdots & x_{0+nF,F-1+mF} \\ \vdots & & \vdots \\ x_{F-1+nF,0+mF} & \cdots & x_{F-1+nF,F-1+mF} \end{pmatrix} = iT \begin{pmatrix} y_{0+nF,0+mF} & \cdots & y_{0+nF,F-1+mF} \\ \vdots & & \vdots \\ y_{F-1+nF,0+mF} & \cdots & y_{F-1+nF,F-1+mF} \end{pmatrix}$$

$$, 0 \leq \mathbf{n} < N_f, 0 \leq \mathbf{m} < N_f$$

### 2.3.3 Filterung gefensterter Transformation

Aus dem bisher gesagten folgt unmittelbar, wie man die vorgestellten Filter sowohl im ein- als auch zweidimensionalen nach einer gefensterter Transformation anwendet. Die Fenster werden getrennt transformiert und getrennt gefiltert, im Anschluß daran wieder zusammengesetzt. Der einzige Unterschied ergibt sich darin, daß die Filterschwelle bei einer nicht gefensterter Transformation als  $0 \leq g \leq N$  (DCT) bzw.  $0 \leq g \leq \frac{N}{2}$  (DFT) gewählt werden muß. Bei einer gefensterter Transformation wird die Filterschwelle durch  $0 \leq g \leq f$  (DCT) bzw.  $0 \leq g \leq \frac{f}{2}$  (DFT) beschränkt, da hier die Fenstergröße  $f$  die Anzahl der Elemente während der Transformation festlegt. Es sollte aber keine Probleme bereiten, dies entsprechend anzuwenden.

## 2.4 Die diskrete Wavelettransformation

*Wavelets* sind Funktionen, die Daten in verschiedene Frequenzen zerlegen und jeden Frequenzanteil mit einer passenden Auflösung untersuchen. Zuvor wurde bereits auf das Problem hingewiesen, daß bei einer Funktion im Zeitbereich mit einem „scharfen“ Übergang mit der Fourier-Methode eine sehr niedrige Konvergenzgeschwindigkeit vorliegt. *Wavelets* dagegen sind in der Lage, auch Sprungstellen und Spitzen in Signalen zu untersuchen. *Wavelets* wurden fast gleichzeitig unabhängig voneinander von Mathematikern, Quantenphysikern, Elektrotechnikern und Seismologen entdeckt.

### 2.4.1 Unterschiede und Gemeinsamkeiten zur Fouriertransformation

Die zugrunde liegende Idee bei der *Wavelet-Analyse* ist die Einführung einer Skala, die eine verschiedene Auflösung bei der Untersuchung der Funktionen einführt. Es ist möglich, ein Signal mit einem großen Zeitfenster zu untersuchen, um globale Eigenschaften des Signals zu erhalten. Untersucht man das Signal mit einem kleinen Zeitfenster, so lassen sich feine lokale Eigenschaften als Ergebnis erhalten. In [AMA-98] heißt es sehr schön, daß die *Wavelet-Analyse* sowohl „den Wald als auch die Bäume“ zeigt.

Ebenso wie die diskrete Fouriertransformation ist die diskrete Wavelettransformation eine lineare Operation, die aus einem Vektor einen Vektor gleicher Länge erzeugt. Beide Transformationen lassen sich als Transformation in einen anderen Funktionenraum betrachten, wobei die DFT Sinus- und Cosinusfunktionen als Basisvektoren für diesen Raum verwendet. Die diskrete Wavelettransformation, kurz *DWT*, verwendet kompliziertere Funktionen als Basis, die sogenannten Wavelets, die auch als „*mother wavelets*“, „*Mutterwavelets*“ oder „*analyzing wavelets*“ bezeichnet werden. Wichtig ist dabei die Selbstähnlichkeit der *Waveletbasis*, da sich jede Funktion der Waveletbasis durch *Dilatation* und *Translation* aus einer *Mutterfunktion* erzeugen läßt. Mit Kenntnis dieser Mutterfunktion läßt sich also die ganze Basis erzeugen. Eine Gemeinsamkeit zwischen DFT und DWT ist ebenfalls, daß die Basisfunktionen im Frequenzbereich lokalisiert sind, also eine Energie besitzen, die auf ein bestimmtes Frequenzband vollständig konzentriert ist.

Einen wesentlichen Unterschied besitzen die Basisfunktionen im Zeitbereich, die Waveletbasen sind im Zeitbereich ebenfalls lokalisiert und besitzen einen *positiven Träger*, im Gegensatz zu den auf ganz  $\mathbb{R}$  vorhandenen trigonometrischen Basisfunktionen der DFT. Bei einer gefensterten DFT wird zum Ausschneiden einer Teilfolge ein Rechteckfenster benutzt, so daß man im Frequenzbereich nur eine einzige Auflösung besitzt. Durch den positiven Träger und die Selbstähnlichkeit der Waveletbasisfunktionen variiert bei der DWT die Fensterbreite, so daß man im Frequenzbereich bei einer einzigen Transformation mit verschiedenen Auflösungen arbeitet.

Im Gegensatz zur DFT gibt es bei der DWT nicht nur eine feste Anzahl von Basisfunktionen, sondern es gibt eine unendliche Menge geeigneter Basisfunktionen, die eine Waveletbasis erzeugen können. Dies hat den Vorteil, daß sich für bestimmte Anwendungen auch entsprechend optimierte Basen finden lassen.

Weitere Informationen und spezifische Anwendungen in Physik und Technik sind in [AMA-98], [HUB-97], [POL-98], [WAV-98] und [LOU-94] beschrieben, wo auch die theoretischen Hintergründe und Herleitungen dargelegt sind.

### 2.4.2 Der eindimensionale diskrete Fall

Zunächst beginnt man auch hier mit der Verbindung zwischen diskreter und kontinuierlicher Welt. Für Funktionen gilt

#### Def. Kontinuierliche Wavelettransformation

Seien  $a, b \in \mathbb{R}, a \neq 0$ , sowie eine Funktion  $f \in L^2(\mathbb{R})$  gegeben. Dann heißt die Transformation

$$(W_y f)(b, a) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) \overline{y\left(\frac{t-b}{a}\right)} dt \quad (50)$$

*Integral-Wavelettransformation* von  $f$ . Der Parameter  $a$  gibt die *Skala*, also die Ausdehnung der Basisfunktion  $y$  an, und  $b$  ist der *Ort*. Die Funktion  $y$  heißt *Wavelet*, wenn es möglich ist, aus der *Wavelettransformation*  $(W_y f)$  wieder die Ausgangsfunktion  $f$  zu berechnen, also

$$f(t) = \frac{1}{\sqrt{c_y}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (W_y f)(b, a) y\left(\frac{t-b}{a}\right) \frac{1}{\sqrt{|a|}} \frac{da db}{a^2} \quad (51)$$

Dabei ist  $c_y$  eine vom Wavelet abhängige Konstante ([LOU-94]).

Die Wavelettransformation ist eine lineare Transformation, deren Basis aus einer einzigen Funktion  $y$  durch Verschiebung und Stauchung gewonnen wird. Die obige Darstellung ist redundant, und man kann das Integral ohne Informationseinbußen durch eine Doppelsumme ersetzen. In der diskreten Wavelettransformation verwendet man für die Streckung in der Regel nur den Faktor 2 und beschränkt sich auf Wavelets der Form

$$y(2^k t + l) \quad (52)$$

mit ganzzahligen Werten für  $k$  und  $l$ . Die diskrete Wavelettransformation vermeidet daher die Redundanz in der Darstellung der Funktion ([HUB-97]).

Betrachtet wird die diskrete Transformation von periodischen Folgen, wobei eine Periodisierung wie bei der DFT bereits gezeigt durchgeführt wird.

Für eine Gewinnung des DWT-Algorithmus benötigt man die Definition einer *Multiskalenanalyse* von  $L_2(\mathbb{R})$ , oder kurz *MRA* (*multiresolution analysis*).

### Def. Multiskalenanalyse

Eine *Multiskalenanalyse* ist eine Kette von Teilräumen  $\{V_j | j \in \mathbb{Z}\}$ , die folgende Bedingungen erfüllt:

*Verkettung*  $V_j \subset V_{j-1} \subset L^2 \quad \forall j \in \mathbb{Z}$

*Ausdünnung*  $\lim_{j \rightarrow \infty} V_j = 0$

*Ausschöpfung*  $\lim_{j \rightarrow -\infty} V_j = L^2$

*Dilatation*  $v(2t) \in V_{j-1} \Leftrightarrow v(t) \in V_j$

*Riesz-Basis-Eigenschaft*: es gibt eine Funktion  $f \in V_0$ , deren Verschiebungen  $\{f(t-k) | k \in \mathbb{Z}\}$  eine Riesz-Basis von  $V_0$  bilden.

Aus diesen Eigenschaft folgt nach [WIC-96] die Existenz einer normierten Funktion  $f$ , die die Zwei-Skalen-Relation erfüllt:

$$f(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h(k) f(2t - k) =: Hf(t) \tag{53}$$

Weiterhin definiert man komplementäre Unterräume

$$V_{j-1} = W_j + V_j \tag{54}$$

woraus sich ebenfalls die Eigenschaft

$$L^2 = \sum_{j \in \mathbb{Z}} W_j \tag{55}$$

ergibt.

Die Räume  $W_j$  nennt man *Waveleträume*, wobei die Waveleträume eine sogenannte *Wavelet-Zerlegung* von  $L^2$  bilden. Um zu einer geeigneten Riesz-Basis von  $W_0$  zu kommen, muß eine sogenannte *Waveletgleichung* erfüllt sein:

$$y(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g(k) f(2t - k) =: Gf(t) \quad (56)$$

$$\text{mit } g(k) = (-1)^k \bar{h}(1 - k)$$

Für geeignete Funktionen  $f$  und  $H$  erhält man eine orthonormale Waveletbasis für  $L^2$ , die das Fundament für den Basiswechsel der Wavelettransformation darstellt.

Die Operationen wurden bisher auf Funktionen angewandt, in [WIC-96] wird gezeigt, daß die Bedingungen auch für diskrete Folgen gelten, wenn  $x$  und  $f$  stetige Ableitungen bis zu einer Ordnung  $d$  besitzen.

Die Operationen nach den Gleichungen ( 53 ) und ( 56 ) im diskreten Fall sind auch als *diskrete Faltungsoperation* bekannt, eine Operation, die einer Filterung von Signalen entspricht. Ganz speziell gilt hier, daß die Faltung nicht mit jedem Element von  $f$  ausgeführt wird, sondern nur mit jedem zweiten Element. In der Signalverarbeitung bezeichnet man derartige Filter als *Dezimationsfilter*, da die Länge der Eingangsfolge nicht mehr der Länge der Ausgangsfolge entspricht. Ganz speziell wird hier die Ausgangsfolge wegen dem Faktor 2 nur noch halbe Länge besitzen (z.B. [LOC-96]).

Sucht man nun nach Folgen  $H$  und  $G$ , die eine orthonormale Riesz-Basis ergeben, so wird man bei den sogenannten orthonormalen *Quadraturfiltern* fündig. Ein solches Filterpaar besteht aus einem Hochpaß  $G$  und einem Tiefpaß  $H$ . Die wesentliche Eigenschaft von einem Paar solcher Quadraturfilter ist, daß man keine Informationen verliert, da die Operationen zueinander komplementär sind. Es ist möglich, mit entsprechenden Rekonstruktionsfiltern wieder die Ausgangsfolge zu erhalten, wobei allerdings die Konstruktion der Rekonstruktionsfilter relativ aufwendig und nur für *IIR-Filter*, also Filter mit unendlicher Sprungantwort (*infinite impulse response*), möglich ist. Die Elemente der Zerlegung mit orthogonalen Filtern sind wieder orthogonal.

Man kann auf die Bedingung der Orthogonalität der Zerlegung verzichten und fordert nur eine *Biorthogonalität* der Quadraturfilter, wodurch man auch *FIR-Filter*, also Filter mit Sprungantworten endlicher Länge (*finite impulse response*), verwenden kann. Damit ist natürlich auch eine algorithmische Umsetzung ohne Verluste durch den Abbruch der Folgen möglich. Im Falle von periodischen Eingangsfolgen periodisiert man ebenfalls die Filterfolgen, so daß keine neuen Überlegungen notwendig sind.

In [WIC-96] sind die Koeffizienten zahlreicher biorthogonaler Quadraturfilter aufgeführt, die zur Berechnung entsprechender Wavelettransformationen verwendet werden können. Die Filterkoeffi-

zienten sind dabei so berechnet worden, daß sich eine Transformation unter Verwendung des jeweiligen Mutterwavelet ergibt:

- Beylkin 18. Ordnung
- Coifman 6. Ordnung
- Coifman 12. Ordnung
- Coifman 18. Ordnung
- Coifman 24. Ordnung
- Coifman 30. Ordnung
- Daubechies 2. Ordnung, auch bekannt als Haar-Walsh
- Daubechies 4. Ordnung
- Daubechies 6. Ordnung
- Daubechies 8. Ordnung
- Daubechies 10. Ordnung
- Daubechies 12. Ordnung
- Daubechies 14. Ordnung
- Daubechies 16. Ordnung
- Daubechies 18. Ordnung
- Daubechies 20. Ordnung
- Vaidyanathan 24. Ordnung

Die Filter sind dabei durch zwei Paare charakterisiert - das Paar der analysierenden Filter - womit das Signal in zwei Teile aufgeteilt wird, und das Paar der synthetisierenden Filter, womit sich das Signal wieder gewinnen läßt.

Die folgende Abbildung 32 zeigt dabei die Anordnung der Koeffizienten bezüglich der Teilräume durch Filterung, nachdem die Transformation durchgeführt wurde. Aus dieser Anordnung läßt sich später die Idee für die Datenreduktion ableiten.

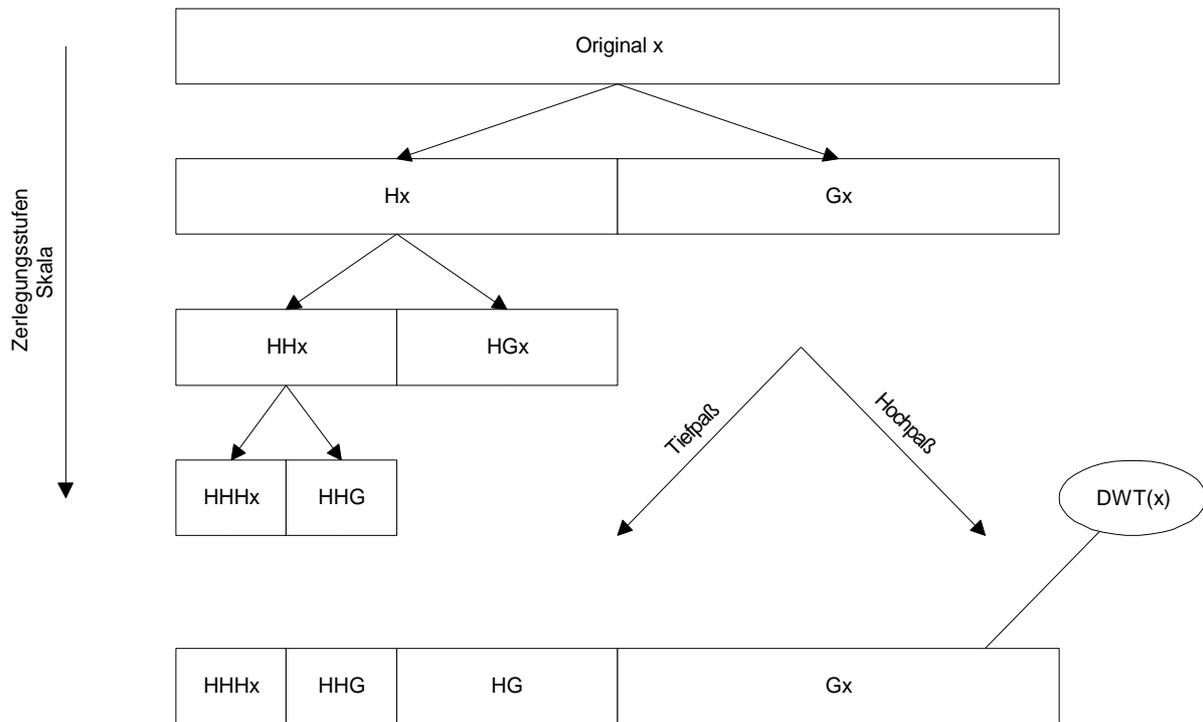


Abbildung 32: Diskrete Wavelet-Transformation und die Zerlegung über mehrere Skalen, H ist eine Operation zur Tiefpaßfilterung, G eine Operation zur Hochpaßfilterung

### 2.4.3 Der zweidimensionale diskrete Fall

Nach [LOU-94] kann auch die zweidimensionale diskrete Wavelettransformation durch eine Tensorisierung der Matrix und Anwendung der eindimensionalen Transformation auf die Zeilen – und Spaltenvektoren durchgeführt werden. Man erhält die

#### Def. Zweidimensionale (inverse) diskrete Wavelettransformation

Seien  $\mathbf{X}$  und  $\mathbf{Y}$  Matrizen mit  $\mathbf{X} \in \mathbb{R}^N \times \mathbb{R}^N$ ,  $\mathbf{Y} \in \mathbb{R}^N \times \mathbb{R}^N$  und  $\mathbf{X}'$  eine Hilfsmatrix mit  $\mathbf{X}' \in \mathbb{R}^N \times \mathbb{R}^N$ , sowie  $N \in \mathbb{N}$ . Dann gilt für die zweidimensionale diskrete Wavelettransformierte von  $\mathbf{X}$

$$DWT : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^N \times \mathbb{R}^N$$

$$\mathbf{X} \mapsto \mathbf{Y} : \mathbf{Y} = DWT(\mathbf{X}) \quad (57)$$

wobei für die Zeilen- und Spaltenvektoren der Matrizen gilt:

$$\begin{pmatrix} x'_{0,n} \\ \vdots \\ x'_{N-1,n} \end{pmatrix} = DWT \left( \begin{pmatrix} x_{0,n} \\ \vdots \\ x_{N-1,n} \end{pmatrix} \right) \quad \forall 0 \leq n \leq N-1$$

$$(y_{m,0} \quad \cdots \quad y_{m,N-1}) = DWT \left( (x'_{m,0} \quad \cdots \quad x'_{m,N-1}) \right) \quad \forall 0 \leq m \leq N-1$$

Umgekehrt gilt dies analog auch für die inverse diskrete Wavelettransformation, die entsprechenden Gleichungen lassen sich leicht selbst herleiten, wenn man die eindimensionale  $DWT$  durch die  $iDWT$  ersetzt.

Auch im zweidimensionalen gilt die Invertierbarkeit der diskreten Wavelettransformation, also

$$iDWT(DWT(\mathbf{X})) = \mathbf{X} \quad (58)$$

#### 2.4.4 Idee der Datenreduktion

Bei der DWT erhält man ebenso wie bei der DFT oder DCT einen Vektor mit den Koeffizienten, bezogen auf eine Darstellung mit Hilfe der Basisvektoren. Bei DFT und DCT wird zur Datenreduktion die Tatsache eingesetzt, daß betragskleine Koeffizienten existieren, die weggelassen werden können. Praktischerweise fallen bei realen Farbbildern die betragskleinen Koeffizienten mit den hohen Frequenzen zusammen, da hohe Frequenzen in diesen Bildern nur einen geringeren Anteil der Gesamtenergie enthalten. Bei der DWT kann man zur Datenreduktion die gleiche Idee nutzen, da jede Zerlegung des Eingangsvektors die Informationen in einen höherfrequenten und einen niederfrequenten Anteil zerlegt. Läßt man nun in jeder Zerlegungsstufe immer den höherfrequenten Anteil weg bzw. setzt diesen Teil des Vektors identisch 0, so läßt sich eine zunehmende Datenreduktion erreichen, je nachdem, auf wie vielen Stufen man diese Reduktion durchführt. Durch eine inverse Transformation erhält man wieder ein Bild nach der Datenreduktion.

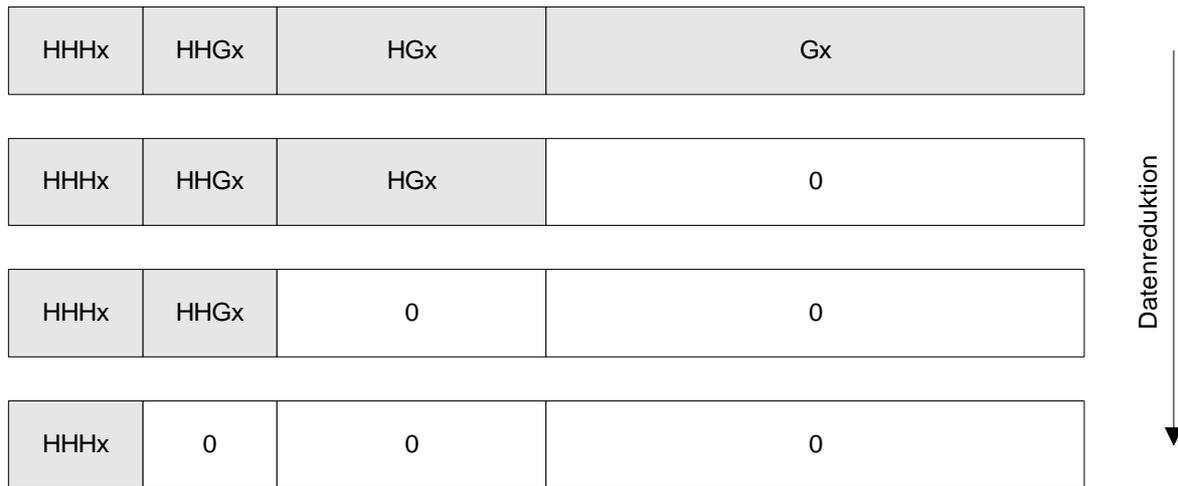


Abbildung 33: Datenreduktion mit zunehmendem Reduktionsfaktor über verschiedene Skalen

Zur Filterung definiert man einen Waveletfilter

Def. Wavelettiefpaßfilter

Sei  $X \in \mathbb{R}^N$  der Eingangsvektor eines diskreten Systems im diskreten Waveletraum,  $Y \in \mathbb{R}^N$  der entsprechende Ausgangsvektor, weiterhin gilt  $N = 2^k, k \in \mathbb{N}$ . Dann heißt ein Vektor  $H_{WLP_g} \in \mathbb{R}^N$  ein Wavelettiefpaß mit dem Schwellwert  $g, 0 \leq g \leq k, g \in \mathbb{N}$  wenn gilt:

$$\begin{aligned}
 H_{WLP_g}[n] &= 1, g = 0 \\
 H_{WLP_g}[n] &= \begin{cases} 1 & , n < 2^{k-g} \\ 0 & , sonst \end{cases}, g > 0
 \end{aligned}
 \tag{59}$$

Für den Ausgangsvektor des Systems gilt:

$$Y[n] = H_{WLP_g}[n] \cdot X[n] \quad , 0 \leq n < N
 \tag{60}$$

Die Filterung wurde dabei wie bei der DFT und DCT in der Form definiert, daß im Waveletraum eine Filterung durch ein Hadamard-Produkt von Eingangsvektor und Filtervektor durchgeführt werden kann.

### 2.4.5 Anwendung der Datenreduktion im zweidimensionalen Fall

Die Art und Weise der Filterung wavelettransformierter Matrizen bietet keine Neuerungen gegenüber der Filterung bei der zweidimensionalen DFT und DCT. Es gilt auch hier wieder der

#### Satz: Algorithmus für Filterung von zweidimensionalen diskreten Wavelettransformierten

Seien  $\mathbf{X}$  und  $\mathbf{Y}$  Matrizen mit  $\mathbf{X} \in \mathcal{R}^N \times \mathcal{R}^N$ ,  $\mathbf{Y} \in \mathcal{R}^N \times \mathcal{R}^N$  und  $\mathbf{X}'$  eine Hilfsmatrix mit  $\mathbf{X}' \in \mathcal{R}^N \times \mathcal{R}^N$ , sowie  $N = 2^k, k \in \mathcal{N}$ . Außerdem sei  $H_{WLP_g} \in \mathcal{R}^N, 0 \leq g \leq k$  ein Wavelettieflpaß mit dem Schwellwert  $g$ , wie er für die eindimensionale Waveletfilterung definiert wurde.

Für die gefilterte Matrix  $\mathbf{Y}$  gilt dann:

$$x'_{mn} = x_{mn} \cdot H_{WLP_g}(m) \quad \forall 0 \leq n \leq N-1 \wedge 0 \leq m \leq N-1$$

$$y_{mn} = x'_{mn} \cdot H_{WLP_g}(n) \quad \forall 0 \leq n \leq N-1 \wedge 0 \leq m \leq N-1$$

Zum besseren Verständnis ist in Abbildung 34 für ein Bild der Größe  $64 \times 64$  Punkten die Schrittweise Zerlegung der Matrix in die verschiedenen Skalenräume dargestellt, wobei in der jeweiligen Teilmatrix der zugehörige Schwellwert eingetragen ist, bei dem die entsprechende Teilmatrix noch nicht identisch 0 gesetzt wird.

$5_H$	$5_H$	5H	5H	5H	5H
$5_H$	$4_G$	4G	4G	4G	4G
$5_H$	$4_G$	3 (2x2)	3 (4x2)	3 (8x2)	3 (16x2)
$5_H$	$4_G$	3 (2x4)	2 (4x4)	2 (8x4)	2 (16x4)
$5_H$	$4_G$	3 (2x8)	2 (4x8)	1 (8x8)	1 (16x8)
$5_H$	$4_G$	3 (2x16)	2 (4x16)	1 (8x16)	0 (16x16)

Abbildung 34: Bilddatenreduktion bei der zweidimensionalen diskreten Wavelettransformation bei einer Matrix der Größe  $64 \times 64$  Elementen

## 2.5 Entwicklung eines Reduktionsalgorithmus

Die Idee der Datenreduktion bei der Bildverarbeitung läuft darauf raus, einen Basiswechsel des Bildes durchzuführen. Man beschreibt das Bild nicht mehr mit den Farbwerten der Pixels, sondern mit den Werten der Koeffizienten bezogen auf eine andere Basis. Wie gesehen, kann der Basiswechsel mit verschiedenen Transformationen durchgeführt werden. Idealerweise ist die Basis so gewählt, daß möglichst viele Koeffizienten sehr klein und damit auch vernachlässigbar werden. Ein entsprechender Algorithmus arbeitet im wesentlichen die folgenden Schritte ab ([LOU-94]):

1. **Transformation:** Aus einem Ausgangsbild wird eine Matrix erstellt, die einer Transformation unterworfen wird. Als Transformation kommen nur solche in Frage, die invertierbar sind, um das Originalbild wieder herstellen zu können.
2. **Quantisierung:** Die Koeffizienten der transformierten Matrix sind reell oder komplex und müssen nun für die Speicherung auf ganzzahlige Werte abgebildet werden, sehr kleine Koeffizienten unterhalb eines bestimmten Schwellwertes werden dabei auf 0 abgebildet. Es ist durchaus

möglich, für verschiedene Indizes der Koeffizienten unterschiedliche Genauigkeiten bei der Quantisierung anzuwenden, also bestimmte Frequenzen genauer oder weniger genau abzubilden. Dazu dienen sogenannte *Quantisierungstableaus*, die den Quantisierungskennlinien der Audioverarbeitung entsprechen. Die Auswahl der entsprechenden Tableaus ist der wesentliche Schritt bei der Entwicklung eines anwendbaren Algorithmus.

3. **Codierung:** Die quantisierten Koeffizienten werden binär abgespeichert, wobei hier entsprechende Codierungsverfahren (z.B. Huffman-Codierung) eingesetzt werden.

Zur Darstellung und Gewinnung der Originalinformationen werden die Schritte in umgekehrter Reihenfolge ausgeführt, wobei durch Weglassung und Näherung ein identisches Abbild der Originalinformation in der Regel nicht mehr möglich ist. Wie groß dieser Näherungsfehler tatsächlich wird, hängt stark von der Auflösung und Qualität der Quantisierung ab.

Im vorliegenden Fall wird kein vollständiger Kompressions- bzw. Reduktionsalgorithmus entwickelt, der Schritt der Codierung wird nicht berücksichtigt, der Quantisierungsschritt wird nur soweit berücksichtigt, wie das Weglassen von betragskleinen Koeffizienten betroffen ist. Hiermit sind bereits wesentliche Fehlerabschätzungen der Verfahren möglich.

### 3 Programmtechnische Implementation

#### 3.1 Die Entwicklungsumgebung

Das Programm wurde für die Betriebssysteme Windows 95 und Windows NT4.0 von Microsoft entwickelt. Die Implementation erfolgte durchgängig in der Sprache C++ auf dem Entwicklungssystem *Visual C++ 5.0* der Firma Microsoft, zum Aufbau eines fensterbasierten Programms wurde dabei die mitgelieferte Bibliothek *MFC – Microsoft Foundation Classes 4.2* benutzt.

Um den Entwicklungsaufwand für den Datenaustausch auf Festplatte und Diskette so gering wie möglich zu halten, wurden für den Import und Export von Grafikdateien die mit [LIP-97] mitgelieferten Konvertierungsmodule verwendet. Verlag und Autor erlauben die private und kommerzielle Nutzung der Konvertierungsmodule, solange die Konvertierungsmodule nur in kompilierter Form und mit entsprechenden Autorenhinweisen weitergegeben werden.

#### 3.2 Unterstützte Grafikformate

Seitdem es die ersten Heimcomputer gibt, beschäftigen sich die Anwender mit der Computergrafik. Doch bereits seit den ersten Ursprüngen taucht ein immer wiederkehrendes Problem auf: wie speichert man die Bilder, so daß sie auf einem elektronischen Speichermedium weitergegeben werden können? Leider hat sich hier seit den ersten Heimcomputertagen beinahe zuviel getan: so kennt zum Beispiel das bekannte Microsoft Word in der neuesten Version über 20 verschiedene Grafikformate. Es gibt hier nicht nur unterschiedliche Grafikformate für verschiedene Betriebssysteme, sondern auch für verschiedene Hersteller, Softwarepakete und Anwendungen. Auch die zunehmende Verbreitung des Internet in Form des World-Wide-Web, das in vielen Bereichen ja doch eine Standardisierung von Dateiformaten bewirkte, wird in den letzten Monaten durch immer mehr proprietäre Datenformate überflutet, obwohl hier immerhin im Moment (noch?) die Verwendung von nur zwei Grafikformaten am gängigsten ist.

Prinzipiell lassen sich zur Datenspeicherung von Bildern *vektorbasierte* und *rasterbasierte* Verfahren unterscheiden. Bei den Vektorverfahren wird die Information in der Art abgelegt, daß im wesentlichen Anfangs- und Endpunkte eines definierten Koordinatensystems, Linienstärken und Farben gespeichert werden. Diese Formate besitzen ihre Stärken bei der Speicherung von linienbasierten oder technischen Zeichnungen, wie sie im CAD-Bereich Verwendung finden. Zur Speicherung von „flächigen“ Bildern, also Bildern, bei denen ganze Bereiche mit Farben oder Farbverläufen eingefärbt sind, verwendet man in der Regel rasterbasierte Verfahren: ein Bild besteht hier aus einer

rechteckigen Anordnung von Bildpunkten (sogenannten Pixeln), wobei jedem Punkte eine Zahl zugeordnet wird, die einem definierten Wert einer Farbskala entspricht. Diese Methode entspricht im wesentlichen der Ablage eines Bildes in Form einer Matrix  $\mathbf{M} \in \mathcal{Q}^n \times \mathcal{Q}^m$ . Aus naheliegenden Gründen beschränkt sich daher die Implementation auf einige verbreitete Dateiformate für Rasterbilder für das Betriebssystem Windows.

Unterschiede hinsichtlich der verwendbaren Farbskala bieten alle Formate. Im wesentlichen unterscheidet man hier eine maximale Farbanzahl von  $2^{24}$  (= 16,8 Millionen) Werten, sogenannte True-Color-Bilder, die einen realistischen Fotoeindruck vermitteln können, und Formate, die zwischen 2 und 256 Farbwerte aufweisen. Letztere bieten sich weniger für Fotos oder gescannte Vorlagen an, sondern für Logos oder Symbole, die eine genau bestimmbare Anzahl von Farben bzw. eine genau definierbare Farbpalette besitzen.

Außerdem wird die Vielzahl von Formaten noch durch die unterschiedlichen Komprimierungsverfahren bei der Speicherung bereichert. Es handelt sich hierbei in der Regel um verschiedene Umcodierungsverfahren, die mehr oder weniger auf der Huffman-Codierung beruhen (eine Beschreibung findet man z.B. in [MIL-90]). Dadurch wird, teilweise sehr trickreich, eine Verringerung der Dateigröße bei der Speicherung bewirkt.

Grundlagen eines jeden Grafikformates sind die sogenannten *Planes*. Ein Farbwert eines Pixels wird dabei im Speicher entsprechend seiner binären Wertigkeit über mehrere Planes verteilt. Angenommen, ein Bild besitzt eine maximale Farbzahl von 16 Farben, dann sind zur Speicherung 4 Bit notwendig. Das Bild besitzt daher 4 Planes oder Ebenen, die in der Reihenfolge der Bitwertigkeit von 0 bis 3 numeriert werden. Soll ein Pixel an der Stelle  $(x,y)$  mit dem Farbwert 5 (binär: 0101) gespeichert werden, so wird an der Stelle  $(x,y)$  in den Planes 0 und 2 das Bit gesetzt, in den Planes 1 und 3 das Bit gelöscht. Daher wird von jedem Pixel in jeder Plane tatsächlich nur jeweils ein einziges Bit gespeichert, die Information aber über alle Planes verteilt.

### 3.2.1 Das Bitmap-Format (BMP)

Das BMP-Format ist das Standard-Dateiformat für Rasterbilder auf allen Windows-Systemen. Hauptvorteil dieses Formats ist, daß es von jedem Windows-Programm unterstützt wird, es bildet daher den kleinsten gemeinsamen Nenner beim Datenaustausch auf Windows-Systemen. Das BMP-Format kann Bilder jeweils mit einer maximalen Farbanzahl von 2, 16, 256 oder 16,8 Millionen Farben aus einer Palette von 16,8 Millionen Werten darstellen. Bei Bildern mit einer Farbanzahl von 16 oder 256 Werten kann ein einfaches Komprimierungsverfahren verwendet werden, das sogenannte *RLE-Verfahren* (*Run Length Encoding*, Lauflängenkodierung). Hierbei werden in jeder

Plane benachbarte Bits mit dem gleichen Wert nicht mehr einzeln gespeichert, sondern es wird gespeichert, wie oft der gleiche Wert vorkommt. Damit läßt sich in der Regel nur eine mäßige Verringerung des Speicherplatzbedarfs erzielen. Es gibt allerdings auch bestimmte Bilder, z.B. bei großen, gleichmäßig eingefärbten Flächen wie sie in Linien- und Strichzeichnungen vorkommen, bei denen die Kompression extrem effektiv ist. Vorteil ist aber, daß dieses Verfahren sehr leicht implementierbar ist.

Außer Windows unterstützten ursprünglich keine anderen Plattformen dieses Format, durch die weite Verbreitung von Windows in den letzten Jahren bieten aber inzwischen auch zahlreiche Fremdanbieter die Möglichkeit an, BMP-Dateien zu lesen oder zu speichern [LIP-97].

### 3.2.2 *Das Graphics Interchange Format (GIF)*

Gerade den Nachteil des BMP-Formats, besonders plattformspezifisch zu sein, vermeidet das GIF-Format. Ursprünglich wurde dieses Format von der Firma CompuServe als geräteunabhängiges Speicherformat eingeführt, vor allem für die Datenübertragung zwischen Rechnern und Mailboxen. Entsprechend wurde bei diesem Format Wert auf eine sehr gute Kompressionsrate gelegt. Maximal kann eine GIF-Datei 256 Farben aus einer Palette von 16,8 Millionen Farben aufnehmen. Wegen dieser Einschränkung ist dieses Format nicht besonders gut für die Speicherung von Fotos oder gescannten Echtfarbvorlagen geeignet, es findet heute seine wesentliche Anwendung bei der Datenübertragung von Symbolen, Icons und Logos auf Internet-Seiten. Als Kompressionsalgorithmus findet der *LZW-Algorithmus (Lempel-Ziv-Welch-Algorithmus)* Anwendung. Grundlegende Idee hierbei ist, anders als beim RLE-Algorithmus, nicht nach gleichen Bitwerten, sondern nach gleichen Mustern Ausschau zu halten. Alle gefundenen Muster, die sich mehrfach wiederholen, werden in einer Tabelle eingetragen und erhalten eine eindeutige Identifikation. Im Bild wird nun nicht mehr die Bitfolge des Musters, sondern nur noch die Nummer des Musters übertragen, sowie die Mustertabelle. Laut Angaben lassen sich hier auch bei komplexeren Bildern durchaus Kompressionsraten von bis zu 60% erreichen [LIP-97].

### 3.2.3 *Das ZSoft Paintbrush Format (PCX)*

Das PCX-Format kommt ursprünglich noch aus dem Bereich der DOS-Anwendungen und wurde von der Firma ZSoft Corporation für das Programm Paintbrush entwickelt. Dieses Format hat sich trotz der zunehmend schwindenden Bedeutung von DOS bis heute gehalten und wurde immer wieder an die neuen Möglichkeiten der Hardware angepaßt. Das PCX-Format unterstützt inzwischen Bilder mit 2, 16, 256 und 16,8 Millionen Farben. Als Kompressionsverfahren wird ähnlich wie beim BMP-Format eine Lauflängenkodierung eingesetzt, die ähnliche Einschränkungen hinsichtlich der Effektivität aufweist. Schwerpunkt bei der Entwicklung dieses Grafikformats war die

Geschwindigkeit des Bildaufbaus, weniger die Effizienz hinsichtlich der Kompressionsrate ([LIP-97]).

### 3.2.4 *Das Tag Image File Format (TIFF)*

Das TIFF-Format wurde von der Firma Aldus als Standard für die Datenspeicherung von gescannten und digitalisierten Bildern definiert und eingeführt. Schwerpunkt bei der Definition war unter anderem die Erweiterbarkeit des Formats durch weitere Eigenschaften, was in der Vergangenheit auch häufig genutzt wurde. Es gibt daher verschiedene Versionen des TIFF-Formats, was natürlich die Austauschbarkeit von Dateien etwas einschränkt. Definitionsgemäß ignoriert nämlich ein Importfilter ihm unbekannte Erweiterungen des Formats einfach, was in der Praxis zu unschönen Effekten führen kann. Das TIFF-Format unterstützt Bilder mit 2, 16, 256 und 16,8 Millionen Farben und verschiedene Kompressionsalgorithmen wie den bereits erwähnten LZW, sowie *CCITT* und *PackBits*. In der neuesten Spezifikation kann ein TIFF-Bild ebenfalls mit dem JPEG-Verfahren (dazu im nächsten Abschnitt mehr) bearbeitet sein. Das *CCITT*-Komprimierungsverfahren kann nur bei Schwarzweißbildern (2 Farben) verwendet werden, es handelt sich hierbei um eine modifizierte Huffman-Lauflängencodierung, die vor allem zur Speicherung von gescannten Textvorlagen gedacht ist (u.a. Fax). Das *PackBits*-Verfahren ist im wesentlichen die Apple Macintosh Variante des RLE-Verfahrens. Das LZW-Verfahren, das beim GIF-Format ja nur mit einer eingeschränkten Farbanzahl zur Verfügung steht, kann nun auch auf Echtfarbbildern mit 16,8 Millionen Farben angewandt werden, Kompressionsraten von 20% und mehr sind hiermit durchaus erreichbar. Beim TIFF-Format handelt es sich um das einzige vom Programm unterstützte Verfahren, daß mit annehmbarer Kompressionsrate eine Echtfarbbild verlustfrei speichern kann. Dies sollte beim Speichern von Berechnungsergebnissen des Programms beachtet werden.

### 3.2.5 *Das Joint Photographic Experts Group Format (JPEG)*

Kein Grafikformat ist so geheimnisumwittert und wird so bestaunt wie das JPEG-Grafikformat. Dies zeigt sich bereits am Namen: eigentlich ist JPEG nur eine Expertengruppe der ISO, die einen Standard für die Komprimierung von Echtfarbbildern (hier mit 16,8 Millionen Farben) entwickelte. Dennoch wird im normalen Gebrauch mit JPEG immer das JPEG-Dateiformat identifiziert. Seinen großen Bekanntheitsgrad hat dieses Format durch den Einsatz für die Datenübertragung von Echtfarbbildern im World-Wide-Web erlangt, viele Anwender sind fasziniert von der großen „Kompressionsrate“ von bis zu 90% (bezogen auf die Dateigröße der Ausgangsdatei).

Allerdings ist der Begriff Kompressionsrate im Zusammenhang mit dem JPEG-Format nicht korrekt, das JPEG-Format führt nämlich keine Umcodierung der Daten durch, wie ein Kompressi-

onsverfahren, sondern führt eine für den Nutzer (hoffentlich) nicht sichtbare Datenreduktion durch. Basis der Datenreduktion ist eine *diskrete Cosinustransformation (DCT)*, wie sie auch im Abschnitt „*Die diskrete Cosinustransformation*“ (Seite 33ff) beschrieben wird. Das Ausgangsbild wird dabei in Matrizen von 8x8 Punkten zerlegt, für die eine DCT durchgeführt wird. Anschließend wird die transformierte Matrix mit Hilfe einer Normalisierungstabelle normalisiert, wobei kleine Koeffizienten auf 0 gesetzt werden. Danach wird der Gleichanteil dieser Matrix von allen Matrixelementen subtrahiert und es findet ein Durchlauf durch die sich ergebenden Matrixelemente statt, wobei dieser Datenstrom nun noch einer Huffman-Codierung zufließt. Das JPEG-Verfahren ist weitgehend frei gehalten, so ist zum Beispiel die Normalisierungstabelle frei wählbar (es gibt allerdings auch Vorschläge von der JPEG), muß allerdings mit dem Bild übertragen werden. Weiterhin ist nicht definiert, wie und mit welcher Genauigkeit der DCT-Algorithmus zu implementieren ist. Dadurch kann die gleiche Datei mit verschiedenen Implementierungen durchaus eine geringfügige Abweichung bei der Darstellung aufweisen. Die qualitativen Nachteile dieses Verfahrens, aber auch die Vorzüge, werden im wesentlichen im Abschnitt „*Untersuchung mit Hilfe der diskreten Cosinustransformation*“ (Seite 86ff) diskutiert. Das JPEG-Verfahren verwendet die gleichen Ideen, wie sie auch im Rahmen dieser Arbeit diskutiert werden, allerdings noch mit zusätzlichen Vorgaben bezüglich einer effizienten Einbettung in den Rahmen eines konkret vorliegenden Dateiformates. Genauere Details zum Aufbau finden sich in [LIP-97] und auch in [STE-95].

### 3.2.6 Übersicht über alle unterstützten Grafikformate

Die folgende Tabelle gibt noch einmal in gekürzter Form die wesentlichen Eigenschaften der unterstützten Grafikformate wieder.

Tabelle 1: Ein Überblick über die verwendeten Grafikformate

Format	Standard im WWW	Datenkompression	Datenreduktion	Max. Farbanzahl in der Datei
BMP	Nein	Ja (bei 16 und 256 Farben)	Nein	2, 16, 256, 16,8 Mio.
GIF	Ja	Ja	Nein	2, 4, 8, 16, 256
PCX	Nein	Ja	Nein	2, 16, 256, 16,8 Mio.
TIFF	Nein	Ja	Ja (in neuester Definition)	2, 16, 256, 16,8 Mio.
JPEG	Ja	Ja (nach Reduktionsschritt)	Ja	16,8 Mio.

### 3.3 Bildverarbeitung und Speicherung im Rechner

Um eine gewisse Vereinfachung der Algorithmen zu erreichen, wurden einige Festlegungen für das Programm getroffen. Folgende Bedingungen müssen die Bilder erfüllen, um für die Verarbeitung geeignet zu sein:

- Die Bilder müssen quadratisch sein.
- Die Kantenlänge  $n$  der Bilder muß eine Potenz von 2 sein, d.h.  $n = 2^k, k \in \mathbb{N}$ .

Diese Forderungen ergeben sich aus den eingesetzten Algorithmen, die zur Durchführung einer schnellen Transformation der Bilder notwendig sind. Nähere Ausführungen dazu folgen in Abschnitt 3.4 (ab Seite 69).

Der Nachteil, daß Bilder nicht in beliebiger Größe transformierbar sind, wird gezielt in Kauf genommen, da der Schwerpunkt der Arbeit nicht auf der Umsetzung eines einsatzbereiten Algorithmus liegt, sondern auf dem Vergleich verschiedener Methoden.

Um dennoch eine gewisse Freiheit bei der Auswahl der Bilder zu besitzen, wurde es ermöglicht, Bilder innerhalb einer Matrix, die die obigen Bedingungen erfüllt, einzubetten. Zum Beispiel kann damit ein Bild der Größe  $640 \times 480$  Punkte geladen werden – es wird danach in einer Matrix der Größe  $1024 \times 1024$  positioniert. Die Lage kann dabei gemäß der folgenden Grafik frei gewählt werden, die neu hinzukommenden Bildpunkte werden auf den Farbwert 0, also schwarz, gesetzt.

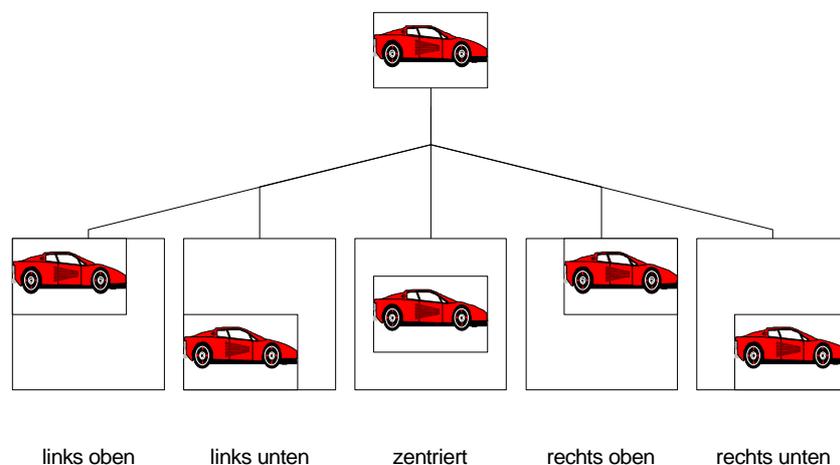


Abbildung 35: Positionierung eines beliebigen Bildes in einer quadratischen Matrix, bei der die Kantenlänge eine Potenz von 2 ist.

Es wurde auch in Betracht gezogen, Bilder mit einem anderen Format durch eine Skalierung auf das notwendige Format zu bringen. Dies wurde allerdings verworfen, da die Skalierung nicht für alle Größen ohne weiteres Konstanz der Energie gewährleisten kann<sup>14</sup>. Und es sollte auf jeden Fall vermieden werden, daß parasitäre Effekte anderer Operationen die Qualität der Transformationen beeinträchtigen.

### 3.3.1 Aufteilung eines Farbbildes auf Farbauszüge

Bei einem Echtfarbbild besitzt jeder Punkt einen Farbwert, der sich allerdings nicht auf einer Skala darstellen läßt. Zur Farbbeschreibung existieren verschiedene Farbmodelle, die alle einen Farbwert durch  $n$ -Tupel mehrerer Skalen beschreiben. In der Computergrafik<sup>15</sup> wird in der Regel das RGB-Farbmodell verwendet, d.h. jeder Farbwert wird in einen Rot-, Grün- und Blauanteil zerlegt. Dabei kann jeder Anteil Werte zwischen 0 und 255 annehmen. Da die Transformationen nur für Vektoren skalarer Werte erklärt sind, ist es klar, daß eine Transformation nicht den vollen Farbwert in einer einzigen Transformation bearbeiten kann.

Üblicherweise sind die Transformationen nur für Grauwertbilder erklärt, eine Transformation eines Farbbildes muß daher in drei Teilen durchgeführt werden. Da jeder Farbauszug (rot, grün, blau) für sich genommen ein Grauwertbild darstellt, wird jeder Farbauszug getrennt in eine eigene Matrix geschrieben und transformiert. Im Anschluß daran wird aus den transformierten Farbauszügen wieder ein Farbbild rekonstruiert.

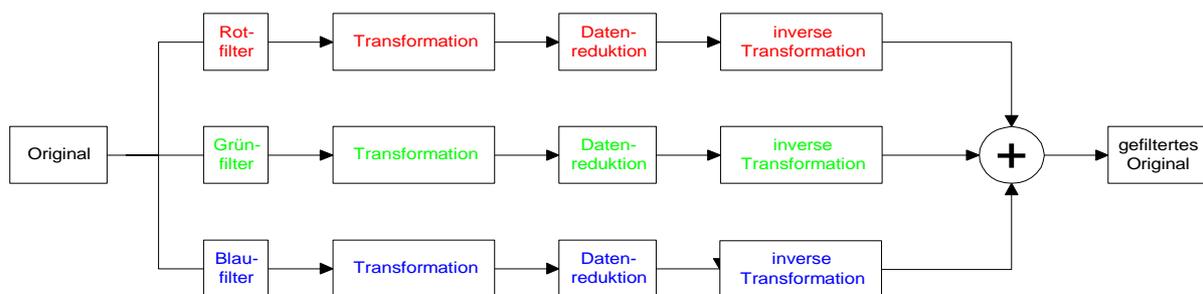


Abbildung 36: Beispiel für die Transformation eines Bildes durch Zerlegung, getrennte Transformation, Datenreduktion und anschließende Rekombination zu einem neuen Farbbild

<sup>14</sup> Ein Beispiel hierfür: ein Bild der Größe  $511 \times 1023$  läßt sich nur auf die Größe von  $1024 \times 1024$  Punkten bringen, sollen die aufgestellten Forderungen erfüllt sein. Dies führt aber dazu, daß das ursprünglich rechteckige Bild extrem verzerrt wird, zum anderen entsteht im Bild ein „Bruch“: ein Punkt muß „doppelt“ genommen werden (oder gemittelt aus Nachbarpunkten) und stört das Aussehen des Bildes erheblich, es handelt sich hier um einen sehr auffälligen Effekt, siehe z.B. [HAB-91] zu dieser Thematik.

<sup>15</sup> Man sollte hinzufügen: in der Bildschirmwelt. Für Computergrafiken, die für Druckenwendungen verwendet werden, wird in der Regel das CMY-Farbmodell (Cyan, Magenta, Yellow) eingesetzt.

### 3.3.2 Speicherung und Speicherplatzbedarf von Bildmatrizen

Die Speicherung eines Bildes erfolgt in einer Matrixform, dies ergibt sich aus den Betrachtungen des 2. Kapitels. Um eine effiziente Implementation der Algorithmen zu erreichen, ist es sinnvoll, noch eine Verbindung zur Vektorstruktur zu haben. Hierdurch lassen sich die Tensorierungen leichter berechnen. Eine Matrix wird als ein Vektor aus Zeilenvektoren im Speicher abgelegt. Dies hat den Vorteil, daß zur Berechnung der Transformation die Zeilenvektoren direkt ohne Umkopieren transformiert werden können. Zur Transformierung der Spalten ist dann allerdings ein Umkopieren notwendig, aus jedem Zeilenvektor wird ein Element entnommen und hieraus ein Spaltenvektor gebildet.

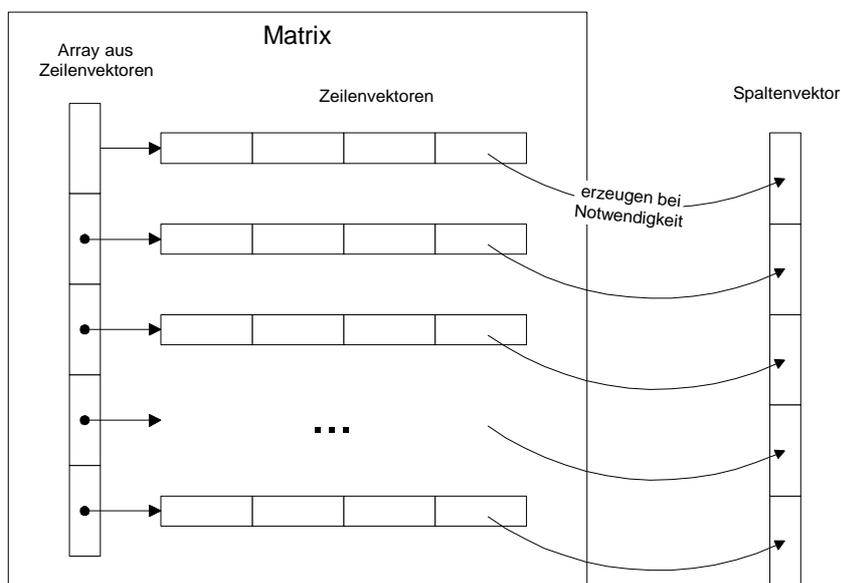


Abbildung 37: Aufbau einer Matrix im Speicher

Der Speicherplatzbedarf  $m$  eines Bildes der Größe  $n \times n$  in Matrixdarstellung ergibt sich aus folgender Gleichung:

$$m = 3 \cdot n^2 \cdot \text{sizeof}(\text{Datentyp}) \quad (61)$$

wobei die Funktion *sizeof* die Größe des Datentyps in Byte zurückliefert, der für die Speicherung eines Matrixelementes verwendet wird. Im Programm wird hier der Typ *double* verwendet, eine Fließkommazahl, für die gilt  $\text{sizeof}(\text{double}) = 4$  Bytes.

Um eine DFT auszuführen, muß noch berücksichtigt werden, daß eine Matrix in diesem Falle komplexe Elemente enthält. Hier gilt für den Speicherbedarf einer Matrix die Gleichung:

$$m = 3 \cdot 2 \cdot n^2 \cdot \text{sizeof}(\text{Datentyp}) \quad (62)$$

Um davon eine Vorstellung zu gewinnen:

Ein Bild der Größe  $1024 \times 1024$  benötigt bei einer DFT für alle 3 Farbauszüge den stattlichen Speicherbedarf von immerhin rund  $25 \text{ MB}$ . Immerhin ist dieser Speicher nur einmal notwendig, da die Transformation *in-place* durchgeführt wird: die Daten werden unmittelbar in der Matrix verarbeitet und es ist keine zusätzliche Matrix für Zwischenwerte notwendig.

### 3.3.3 Die Berechnungs- und Verarbeitungskette

Um das Bild auf den Bildschirm übertragen zu können, ist eine spezielle Speicherdarstellung notwendig, das sogenannte DIB-Format: *Device Independent Bitmap* (Beschreibung in [LIP-97]). In diesem Format werden die Bildinformationen auf Bitebene gepackt – ideal für eine schnelle Darstellung am Bildschirm, aber ungeeignet für Rechnungen. Daher muß vor Beginn der Datenreduktion aus einer DIB die entsprechende Matrix im Speicher aufgebaut werden. Nach dem Ende der Verarbeitung wird aus der Information in Matrixform wieder eine DIB erstellt, die vom Betriebssystem gezeichnet werden kann. Zudem ist die Packungsdichte in einer DIB wesentlich höher, da für jeden Bildpunkt nur tatsächlich so viele Bits reserviert werden, wie für die gewünschte Farbtiefe notwendig sind.

Um den zuvor aufgezeigten Speicherplatzbedarf etwas zu reduzieren, wird ein Trick angewandt: es wird immer nur für einen Farbauszug eine Matrix in Fließkommadarstellung erzeugt. Diese wird direkt aus der DIB erzeugt, indem nur die Informationen des zugehörigen Farbauszuges gelesen werden. Danach wird die Datenreduktion, also Transformation, Filterung und inverse Transformation, ausgeführt. Das Ergebnis wird aus der Matrix in eine neue DIB kopiert, wobei jeweils nur die Informationen der zugehörigen Farbebene überschrieben werden. Damit kann aber der Speicherplatzbedarf auf rund ein Drittel reduziert werden.

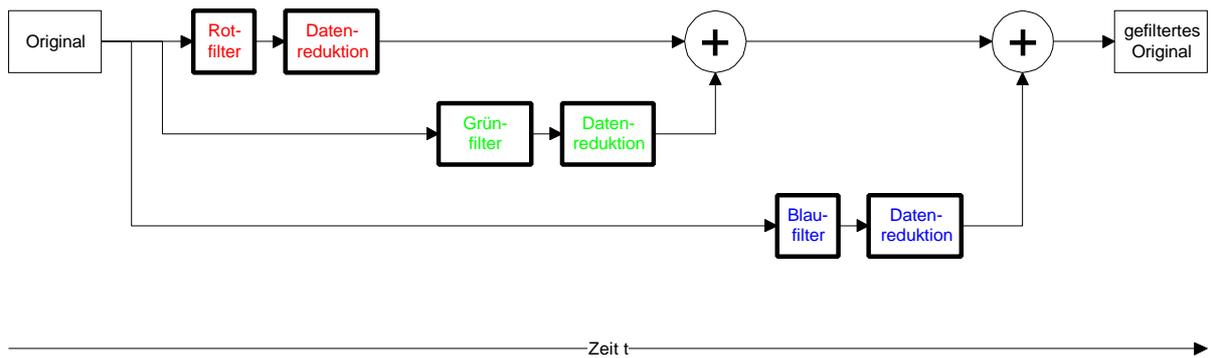


Abbildung 38: Verarbeitung eines Farbbildes in zeitlicher Reihenfolge. Dick umrandet die Operationen, die mit der Fließkomma-Matrix ausgeführt werden, dünn umrandet die Operationen mit der bitorientierten Bitmap (DIB).

### 3.3.4 Modifikation der Farbwertverteilung

Betrachtet man die Eingangsbilder vor und nach der Filterung, so ist die Farbwertverteilung so, daß sich die Skala nicht ändert, d.h. einem bestimmten Farbwert der Matrix wird nach der Filterung wieder der gleiche Grauwert zugeordnet wie zuvor. Dies ergibt sich schon aus der Forderung, daß ein Filter, der die Identität repräsentiert, die Matrix unverändert läßt – also müssen die Grauwerte ebenfalls unverändert bleiben.

#### Def. Farbtreue Skalierung eines Matrixelements

Sei  $z$  ein beliebiges Element der Bildmatrix  $\mathbf{M} \in \mathcal{Q}^m \times \mathcal{Q}^n$  mit  $m$  Zeilen und  $n$  Spalten. Dann gilt für die Abbildung  $s(z): \mathcal{Q} \rightarrow [0,255] \subset \mathcal{N}_0$  eines Matrixelementes  $z$  in die Bildschirmdarstellung bei bildtreuer Skalierung

$$s(z) = \begin{cases} 255 & , \quad z > 1 \\ \lfloor 255 \cdot z \rfloor & , \quad 0 \leq z \leq 1 \\ 0 & , \quad z < 0 \end{cases} \quad (63)$$

Da die Transformationen alle eine Inverse und alle Filterkoeffizienten einen Betrag kleiner 1 besitzen, liegen die Werte nach der Transformation, Filterung und inversen Transformation im gleichen Wertebereich  $[0, 255]$  wie zuvor.

Anders sieht dies aus, wenn man die Matrizen der Transformierten betrachtet. Hier ist es nicht unbedingt offensichtlich, wie man die Werte der Matrix der Grauwertskala zuordnet. Betrachtet man die Matrix einer Transformierten, so fällt auf, daß der Wertebereich wesentlich größer ist als die zur Verfügung stehenden 256 Graustufen, und daß weiterhin die Masse der Koeffizienten einen klei-

nen Betrag hat. Um hier Details etwas besser herauszustellen, werden die Graustufen der Matrix logarithmisch zugeordnet –dadurch werden die betragskleinen Koeffizienten etwas stärker betont.

#### Def. Minimum und Maximum einer Bildmatrix

Sei  $\mathbf{M} \in \mathbb{Z}^m \times \mathbb{Z}^n$  die Bildmatrix mit  $m$  Zeilen und  $n$  Spalten, dann sind die Minimum- und Maximumoperatoren wie folgt definiert:

$$\min\{\mathbf{M}\} = \min(m_{ij}), m_{ij} \in M, 0 \leq i \leq m, 0 \leq j \leq n \quad (64)$$

$$\max\{\mathbf{M}\} = \max(m_{ij}), m_{ij} \in M, 0 \leq i \leq m, 0 \leq j \leq n \quad (65)$$

Mit Hilfe dieser Definition erhält man die Abbildung eines Matrixelements in einen Grauwert nach der folgenden

#### Def. Logarithmische Skalierung eines Matrixelements

Sei  $z$  ein beliebiges Element der Bildmatrix  $\mathbf{M} \in \mathcal{Q}^m \times \mathcal{Q}^n$  mit  $m$  Zeilen und  $n$  Spalten. Dann gilt für die Abbildung  $s(z): \mathcal{Q} \rightarrow [0,255] \subset \mathcal{N}_0$  eines Matrixelementes  $z$  in die Bildschirmdarstellung

$$s(z) = \begin{cases} \left\lfloor \frac{255}{\log 2} \cdot \log \left( 1 + \frac{z - \min\{\mathbf{M}\}}{\max\{\mathbf{M}\} - \min\{\mathbf{M}\}} \right) \right\rfloor, & \min\{\mathbf{M}\} \neq \max\{\mathbf{M}\} \\ 149, & \text{sonst} \end{cases} \quad (66)$$

Bei der Skalierung muß der Sonderfall beachtet werden, daß Minimum und Maximum einer Matrix zusammenfallen. In diesem Fall wird als Wert der Logarithmus des halben Fullscale-Wertes angenommen.

### **3.4 Die Basis: Implementation der Algorithmen**

In der numerischen Mathematik gibt es zusätzlich zur natürlichen Forderung, daß alle Zusammenhänge korrekt sind, auch die verständliche Forderung, daß alle Rechnungen doch bitte möglichst schnell ausgeführt werden. Dem Autor war dies bei der Entwicklung ein wichtiges Anliegen, denn die Programmierung einer „Werkbank“, mit deren Hilfe man alle Ideen ausprobieren kann, wird wohl nur dann akzeptiert, wenn der Nutzer nicht ewig auf die Ergebnisse warten muß. Nur mit Schaudern erinnere man sich an die Zeiten, als man für ein Bild der berühmten „Apfelmännchen“ 10 oder 12 Stunden warten mußte. Das vorliegende Programm dürfte auf einem modernen Rech-

ner hier doch einige Geschwindigkeitsvorteile haben: eine DFT eines  $1024 \times 1024$  Punkte großen Farbbildes dauert nur knapp 2 Minuten<sup>16</sup>.

Bei Optimierungen von Algorithmen wurde dabei die Devise „Speicher ist billiger als Rechenzeit“ angewandt, im Zweifel wurde bei alternativen Implementationsstrategien die schnellere gewählt, ohne Rücksicht darauf, ob diese Variante mehr Speicher für die Rechnung benötigt<sup>17</sup>.

### 3.4.1 Die Idee der schnellen diskreten Fouriertransformation

Der sogenannte *FFT-Algorithmus* - *fast fourier transform* - dürfte wohl mit zu den bekanntesten mathematischen Algorithmen gehören. Zahlreiche Quellen befassen sich mit der Idee, wie dieser Algorithmus arbeitet (z.B. aus theoretischer Sicht [BET-84], stärker auf die Anwendung bezogen [BRI-95]), unzählige Implementationsvarianten kursieren durch das Internet.

Um nicht viele bereits bekannte Fakten erneut zu wiederholen, sollen nur noch einmal kurz die wesentlichen Dinge aufgelistet werden.

- Der FFT-Algorithmus ist kein Näherungsverfahren an die DFT, wie oft behauptet wird, sondern nur eine schnelle Berechnungsvariante zur direkten Definition der DFT, die Ergebnisse sind identisch.
- Der Algorithmus ist ein sogenannter *Divide-And-Conquer-Algorithmus*, bei dem die Menge der Eingangswerte halbiert wird, der Algorithmus wird nun auf der halben Menge erneut ausgeführt. Dies wird wiederholt, bis die Menge nur noch eine geringe Größe hat (hier: 2 Elemente) und der Rechenschritt trivial wird.
- Im Normalfall kann der Algorithmus nur Vektoren mit  $2^n, n \in \mathbb{N}$  Elementen verarbeiten, dies ist auch der Grund für einige im 2. Kapitel gemachte Einschränkungen bezüglich der Eigenschaften der Eingangsvektoren.
- Eine naive direkte Implementation der DFT benötigt  $O(n^2)$  Multiplikationen, der FFT-Algorithmus dagegen nur  $O(n \log n)$  Operationen. Dies beruht im wesentlichen auf der Ausnutzung der Periodizität der komplexen Exponentialfunktion, die in der Definition der DFT auftaucht. Hiermit wird die Berechnung der DFT als Divide-And-Conquer-Algorithmus beschreibbar.

---

<sup>16</sup> Gemessen auf einem Rechner mit einem Prozessor AMD K6 233MHz, 96MB SD-RAM, Chipsatz VIA 590VP3 AGP, und Betriebssystem Windows NT 4.0.

<sup>17</sup> Im allgemeinen Fall sind die Forderungen geringer Laufzeit und wenig Speicherplatz nicht gleichzeitig zu erfüllen und konkurrieren sogar miteinander (sog. *Time-space-trade-off*).

Ursprünglich wurde die erste Implementation des Autors für eine Anwendung in der Signalverarbeitung geschrieben, der Algorithmus wurde dann später für die hier vorliegenden Problembereiche entsprechend angepaßt und in die verwendete Programmiersprache übertragen ([BÄC-95]<sup>18</sup>). Basis ist die sogenannte klassische „*radix-2-decimation in-place*“-Variante nach Cooley und Tukey, vorgestellt 1965 ([BRI-95]). Hierbei wird für Eingabe und Ausgabe des Algorithmus der gleiche Vektor verwendet, es gibt auch Varianten, die 2 verschiedene Vektoren benutzen.

Für die inverse Transformation wird der gleiche Algorithmus verwendet, wobei hier der genannte Implementationstrick nach Gleichung ( 21 ) angewandt wird.

#### Satz: Komplexität des FFT-Algorithmus

Die diskrete Fouriertransformation und die inverse diskrete Fouriertransformation eines Vektors  $x \in \mathbb{C}^n, n = 2^k, k \in \mathbb{N}$  lassen sich mit der Komplexität

$$O(n) = n \log n \quad (67)$$

realisieren. Beweise dafür findet man z.B. in [BRI-95].

#### 3.4.2 Der DCT-Algorithmus

Die diskrete Cosinustransformation wird mit Hilfe von Gleichung ( 38 ) implementiert, dabei wird der FFT-Algorithmus für die diskrete Fouriertransformation benutzt. Es gibt z.B. auch in [WIC-96] ein Beispiel einer Entwicklung eines schnellen Algorithmus speziell für die DCT, dies bringt allerdings einen nur unwesentlichen Komplexitätsgewinn, verschlechtert aber die Testbarkeit der Implementation.

#### Satz: Komplexität der diskreten Cosinustransformation

Die diskrete Cosinustransformation eines Vektors  $x \in \mathbb{R}^n, n = 2^k, k \in \mathbb{N}$  und ihre Inverse lassen sich mit der Komplexität

$$O(n) = 2n \cdot \log(2n) \quad (68)$$

realisieren, wenn für die dabei notwendige DFT der FFT-Algorithmus verwendet wird. Der Beweis ergibt sich aus Gleichung ( 38 ) unter Benutzung des vorigen Satzes.

---

<sup>18</sup> Der Text des Vortrags des Autors ist auf der beigelegten CD als PDF-Datei vorhanden. Es handelt sich hierbei um einen vom

### 3.4.3 Der FWT-Algorithmus zur diskreten Wavelet-Transformation

Als Implementation des DWT-Algorithmus wird der „schnelle“ Transformationsalgorithmus nach Mallat und Daubechies verwendet, wie er in [WIC-96] zu finden ist.

#### Satz: Komplexität der diskreten Wavelettransformation

Die diskrete Wavelettransformation eines Vektors  $x \in \mathbb{R}^n$ ,  $n = 2^k$ ,  $k \in \mathbb{N}$  und ihre Inverse lassen sich mit der Komplexität

$$O(n) = n \quad (69)$$

berechnen.

### 3.4.4 Die Implementation der Matrix-Transformationen

Im 2. Kapitel wurde bereits festgestellt, daß alle Transformationen von Matrizen durch eine nacheinander ausgeführte Transformation aller Zeilen- und Spaltenvektoren realisiert werden kann. Damit ist die Implementation der Matrix-Transformationen unmittelbar auf eine wiederholte Anwendung von Vektor-Transformationen zurück zu führen.

Interessant ist in diesem Zusammenhang, welche Komplexität die notwendigen Algorithmen haben. Allgemein gilt der folgende Zusammenhang:

#### Satz: Komplexität von Matrix-Transformationen

Sei  $T(n)$  die Komplexität einer Transformation (DFT, DCT, DWT) eines Vektors der Länge  $n = 2^k$ ,  $k \in \mathbb{N}$ . Dann gilt für die Komplexität  $O(n)$  der Transformation einer Matrix  $\mathbf{M}$  der Größe  $n \times n$  bei Ausführung der Transformation durch zeilen- und spaltenweise vektorweise Transformation

$$O(n) = 2n \cdot T(n) \quad (70)$$

Für gefensterter Transformationen gilt

### Satz: Komplexität von gefensterter Matrix-Transformation

Seien die Voraussetzungen wie beim vorigen Satz gegeben, und sei zusätzlich  $f = 2^l, l \in \mathbb{N}, f \leq n$  die Fenstergröße mit der quadratischen Kantenlänge  $f$ . Dann gilt für die Komplexität  $O(n, f)$  der Transformation

$$O(n, f) = 2f \cdot T(f) \cdot \left(\frac{n}{f}\right)^2 \quad (71)$$

Der Beweis der beiden letzten Sätze ergibt sich unmittelbar aus einfachen geometrischen Überlegungen zum Aufbau eines Quadrats aus Zeilen, Spalten und kleineren Quadraten.

Insbesondere sieht man, daß sich eine effiziente Realisierung des Transformations-Algorithmus unmittelbar auch auf die Komplexität der Matrixtransformation übertragen läßt. Die gefensterter Transformationen sind einfach durch mehrfache Anwendung der Transformation auf Teilmatrizen durchführbar, wobei diese Teilmatrizen jeweils die Fenstergröße besitzen. Dies ist algorithmisch leicht zu realisieren und verringert die Rechenzeit, weil der Divide-And-Conquer-Algorithmus um eine entsprechende Anzahl von Rekursionsstufen verkürzt wird.

#### 3.4.5 Feintuning und Effizienzsteigerung

Der FFT-Algorithmus läßt sich zusätzlich noch beschleunigen, wenn man einige Zwischenwerte in Tabellen ablegt. Besonders interessant wird dies, weil für ein Bild  $2n$  FFT-Auswertungen notwendig sind. Werden die Tabellen bereits vor der ersten FFT-Anwendung initialisiert und für alle weiteren Rechnungen unverändert verwendet, erreicht diese Methode ein besonders hohe Effizienz.

Tabellen werden für die komplexe Exponentialfunktion für alle vorkommenden Koeffizienten angelegt – bei gegebener Vektorlänge sind vorher alle Koeffizienten genau bestimmbar und somit tabellierbar. Weiterhin wird für das sogenannte *Bit-Reverse-Shuffling* eine Tabelle angelegt, hierbei handelt es sich um eine Zuordnungstabelle für eine Vertauschung der Koeffizienten, die beim FFT-Algorithmus notwendig wird. Die dafür notwendige Rechenoperation ist von einem normalen Prozessor nur ineffektiv ausführbar und gewinnt durch die Tabellierung einen nicht unerheblichen Geschwindigkeitszuwachs.

Auch bei der DCT kann mit einer Tabelle der Drehfaktoren, wie sie in der Definition ( 38 ) vorkommen, aus den gleichen Gründen wesentlich beschleunigt werden. Der für die Tabellen notwendige Speicherbedarf ist im Vergleich zur Größe der Bilddateien nicht relevant.

Weiterhin läßt sich der FFT-Algorithmus beschleunigen, wenn rein reelle Eingangsvektoren verwendet werden, dies ist hier bei der Transformation der Fall – allerdings nicht bei der Rücktransformation. In diesem Fall kann in der ersten Rechenstufe die Addition von Koeffizienten rein reell ausgeführt werden, da die Imaginärteile identisch 0 sind.

Bei der Zerlegung des Eingangsvektors nach der Divide-And-Conquer-Methode kann in der ersten Berechnungsstufe eine komplexe Multiplikation eingespart werden, da der Koeffizient der komplexen Exponentialfunktion immer identisch 0 ist, also  $e^0 = 1$  ergibt.

Die beiden letztgenannten Optimierungen wirken bei der Anwendung eines FFT-Algorithmus auf einen Vektor nicht so dramatisch. Bei einer mehrfachen Anwendung, wie sie bei der Matrix-Transformation notwendig ist, ist allerdings der Geschwindigkeitsgewinn meßbar.

Der Algorithmus zur diskreten Wavelettransformation kann durch derartige Maßnahmen nicht beschleunigt werden, die in [WIC-96] vorgestellte Implementation ist weitgehend optimal<sup>19</sup>.

Weiterhin wurde ganz allgemein in allen Routinen, die sich mit der Verarbeitung der Matrizen befassen, innerhalb von Schleifen mit Zwischenvariablen gearbeitet, um Mehrfachberechnungen von Hilfswerten zu vermeiden. Es wurde allerdings aus Gründen einer überschaubaren Softwarearchitektur nicht dazu übergegangen, Aufrufe von Hilfsrechnungen, etc. in irgendeiner Form einzubetten oder besonders trickreich aufzurufen.

### **3.5 Programmbedienung**

Das Programm wurde als Multi-Dokument-Applikation ausgeführt, es können also mehrere Bilder gleichzeitig geöffnet sein. Die Bedienung erfolgt mit Hilfe der unter Windows üblichen Menüs, Dialoge und Fenster. Die Oberfläche besitzt die Standard-Symbolik, wie sie auch von Microsoft-Programmen verwendet wird. Dem einigermaßen geübten Windows-Benutzer ist dadurch ein schneller Einstieg in die Bedienung des Programms möglich.

---

<sup>19</sup> Sie hat allerdings in der deutschen Ausgabe in der verwendeten Auflage einen schlimmen Druckfehler auf Seite 189, in der letzten Zeile des Pseudocodes muß das [J\*STEP] durch ein [I\*STEP] ersetzt werden. Ob der Fehler auch in der englischen Fassung vorliegt, ist dem Autor nicht bekannt.

Die Transformationen sind auf nur 3 Dialogfenster verteilt, für jede Transformation existiert ein eigenes Fenster, in dem die Einstellungen der Filter gewählt werden können. Um den Umfang der vorliegenden Arbeit nicht unnötig aufzublähen, wurde auf eine gedruckte Bedienungsanleitung verzichtet, statt dessen wird eine Online-Hilfe für das Programm angeboten. Für jedes Menü und jeden Dialog erreicht man über die Taste **F1** einen Hilfetext, der die momentan verfügbaren Funktionen beschreibt.

## 4 Anwendung auf Echtfarbbilder

Zum Test der Verfahren wird immer das gleiche Bild verwendet, um eine Vergleichbarkeit zu erreichen. Die Wahl fiel auf das relativ bekannte Testbild „Lena“ in einer Größe von 512x512 Punkten bei einer Echtfarbendarstellung ([LEN-98]). Die folgende Übersicht zeigt zusätzlich eine Zusammenfassung aller Reduktionsverfahren, die in den Vergleich mit einbezogen wurden. Als Vergleichsmaßstab wurde ein einfaches Reduktionsverfahren mit aufgenommen, bei dem einfach aus einer entsprechenden Anzahl von Punkten der farbliche Mittelwert gebildet wird, wodurch die Anzahl der Punkte verringert werden kann. Diese Verfahren sind mit jedem Grafikprogramm möglich.

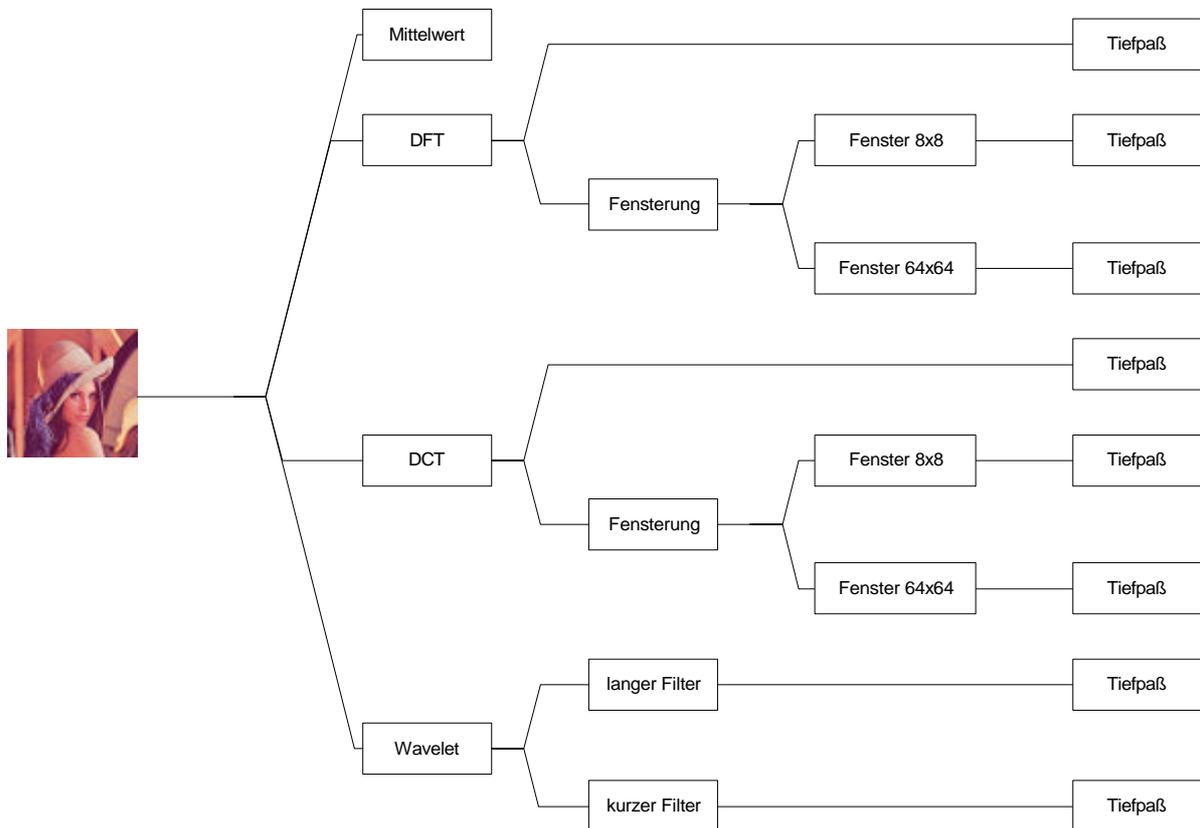


Abbildung 39: Übersicht über die verwendeten Verfahren zur Bildreduktion eines Echtfarbbildes

### 4.1 Beurteilung der Reduktionsverfahren

#### 4.1.1 Die Reduktionsrate

Der Vergleich der verschiedenen Reduktionsverfahren setzt voraus, daß eine Maßzahl definiert wird, die übergreifend über alle Verfahren einen einheitlichen Maßstab vorgibt. Analog zur Defini-

tion einer Kompressionsrate (wie z.B. in [LIP-97], S. 249) wird eine Reduktionsrate definiert, die angibt, wie das Verhältnis des reduzierten Bildes zum Originalbild ist. Dadurch lassen sich Bilder mit der gleichen Reduktionsrate miteinander hinsichtlich der optischen Qualität vergleichen. Als Bildgröße des reduzierten Bildes werden dabei alle Punkte berücksichtigt, die durch das Reduktionsverfahren nicht zu 0 gesetzt wurden.

### Def. Charakteristische Funktion einer Filterung

Sei  $f$  eine der definierten zweidimensionalen Filterabbildungen für die Filterung einer Matrix  $\mathbf{M} \in \mathcal{C}^N \times \mathcal{C}^N$ , die als Ergebnis die Matrix  $\mathbf{M}' \in \mathcal{C}^N \times \mathcal{C}^N$  liefert, also  $\mathbf{M}' = f(\mathbf{M})$ . Dann gilt für die charakteristische Funktion  $\mathbf{k}_f : [0, N-1] \times [0, N-1] \rightarrow \{0,1\}$  der Filterung

$$\mathbf{k}_f(m,n) = \begin{cases} 0 & , m'_{m,n} \neq m_{m,n} \\ 1 & , \text{sonst} \end{cases} \quad , 0 \leq n < N , 0 \leq m < N \quad (72)$$

### Def. Reduktionsrate

Sei  $\mathbf{M} \in \mathcal{C}^N \times \mathcal{C}^N$  eine Matrix eines Originalbildes,  $\mathbf{M}' \in \mathcal{C}^N \times \mathcal{C}^N$  die Matrix, die durch die Filterung mit der zweidimensionalen Filterabbildung  $f$  entsteht, also  $\mathbf{M}' = f(\mathbf{M})$ . Es gilt außerdem  $N = 2^k, k \in \mathbb{N}$ . Außerdem sei  $\mathbf{k}_f$  die charakteristische Funktion der Filterabbildung  $f$ . Dann gilt für die Reduktionsrate  $R_R$  die Definition:

$$R_R = 1 - \frac{\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \mathbf{k}_f(m,n)}{N^2} \quad (73)$$

Die Reduktionsrate setzt also die Anzahl der nach der Reduktion noch vorhandenen Bildpunkte zur Anzahl der vor der Reduktion vorhandenen Bildpunkte in ein Verhältnis. Findet keine Reduktion statt, so ist die Reduktionsrate gleich 0, bei einer nahezu vollständigen Reduktion gilt  $R_R \rightarrow 1$ .

Es sei darauf hingewiesen, daß die Reduktionsrate nicht zwangsläufig identisch mit der tatsächlichen Größenänderung von Bilddateien ist, durch zusätzliche Codierungsverfahren lassen sich noch Speicherplatzverminderungen durch Umcodierungen oder ähnliches erreichen, dies soll aber hier nicht näher betrachtet werden.

### Beispiel:

Man betrachte eine Bildmatrix mit  $8 \times 8$  Punkten bei Anwendung einer DCT und anschließender Tiefpaßfilterung, wobei die Filterfunktion wie folgt gegeben sei:

$$f(m,n) = \begin{cases} 1 & , (0 \leq m \leq 3) \wedge (0 \leq n \leq 3) \\ 0 & , \text{sonst} \end{cases}$$

Dann gilt  $R_R = 1 - \frac{4 \cdot 4}{8 \cdot 8} = 0,75$ . Das heißt, es würde ausreichen, nach der Transformation nur noch 25% der Bildpunkte zu speichern oder zu übertragen. Die folgende Abbildung verdeutlicht noch einmal die zugrunde liegenden Überlegungen zur Definition der Reduktionsrate.

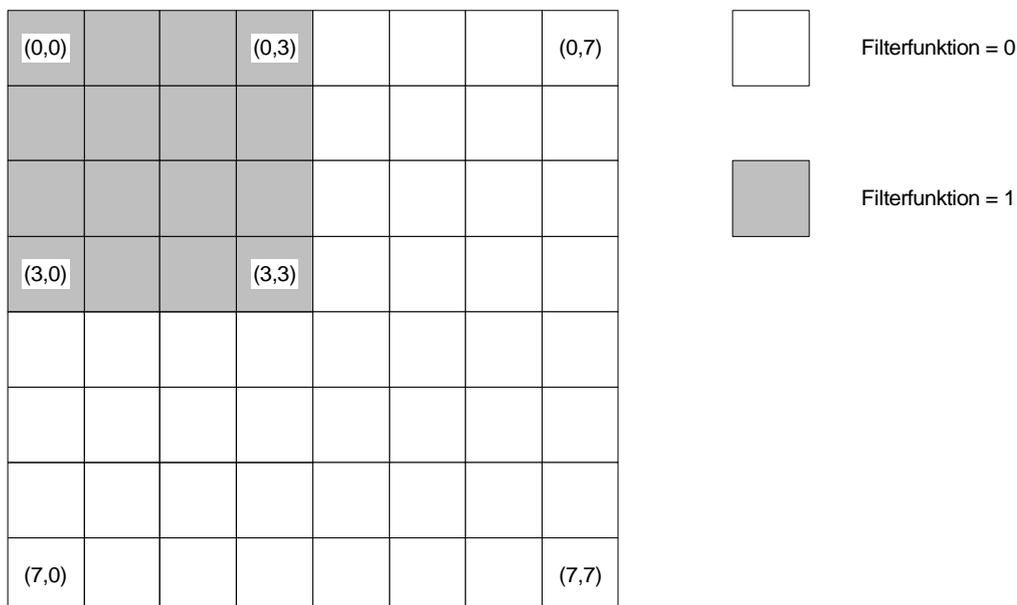


Abbildung 40: Beispiel für Anzahl der übertragenden Punkte nach Datenreduktion

#### 4.1.2 Die Reduktionsraten der verschiedenen Filter

Die Gleichungen für die Reduktionsraten der vorgestellten Transformationen mit nachgeschalteten Filtern lassen sich durch geometrische Überlegungen relativ leicht bestimmen und mit Hilfe der vollständigen Induktion auch für beliebige Bildgrößen beweisen. Aus Platzgründen wurde hier auf den Abdruck aller Gleichungen für die Reduktionsraten der verschiedenen Filter bei den entsprechenden Transformationen verzichtet, das Programm berechnet in den Dialogen bei der Einstellung der Filterschwellen automatisch die zugehörige Reduktionsrate, so daß man dies sofort ablesen kann.

Als interessantes Detail am Rande sollte erwähnt werden, daß bei gefensterten Transformationen bei der relativ gleichen Anzahl von weggelassenen Koeffizienten sich die gleiche Reduktionsrate ergibt. Man könnte erwarten, daß die Fenstergröße die Reduktionsrate beeinflusst, dem ist aber nicht so, da die Fenstergröße bei der Bestimmung der Reduktionsrate herausfällt. Ein Beispiel zur Verdeutlichung: läßt man bei einer nicht gefensterten Transformation mit Tiefpaß die Hälfte der Koeffizienten weg, so ergibt sich mit Fensterung die gleiche Reduktionsrate, wenn man hier in jedem Fenster ebenfalls die Hälfte der Koeffizienten wegläßt.

## 4.2 Die Vergleichsbilder und Vergleichsverfahren

In der folgenden Abbildung ist das Originalbild, das für die Untersuchungen verwendet wurde, dargestellt.

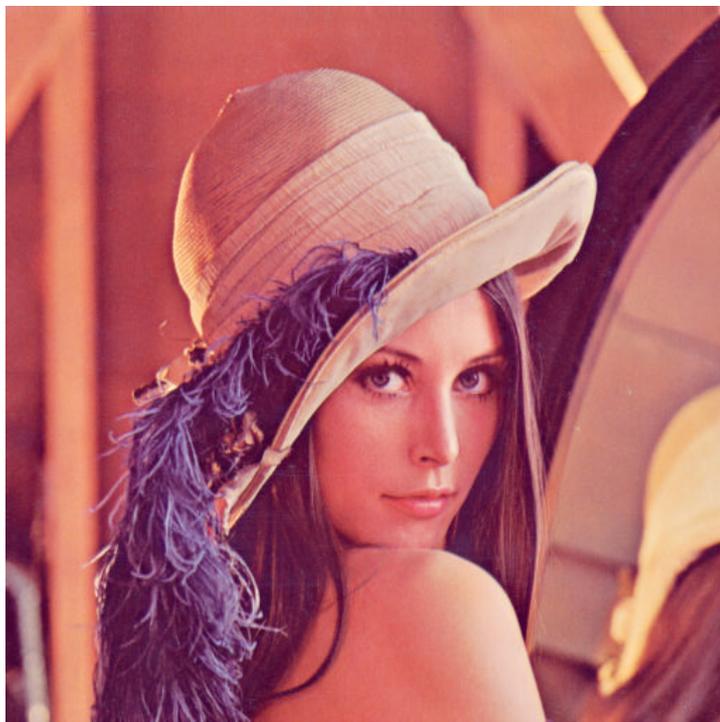


Abbildung 41: Lena im Original mit 512x512 Punkten

Weiterhin wurden zum Vergleich mit einem Grafikprogramm noch zwei Bilder für Reduktionsraten von 60% und 90% berechnet, bei denen die Bildpunkte durch eine Mittelwertbildung farblich zu einer einzigen Farbe zusammengefaßt wurden. Zunächst wurden die Bilder entsprechend der Reduktionsrate verkleinert und im Anschluß wieder auf die Originalgröße vergrößert, wo nun mehrere Bildpunkte die gleiche Farbe besitzen. Die Einbuße der Information ist deutlich sichtbar, da der Klötzchen-Effekt die im Bild notwendigen Kanten zerstört. Die Bilder wirken auf den Betrachter unwirklich und vermitteln nicht mehr den Eindruck eines „normalen“ Fotos, sondern e i-

nes Effektbildes, einer Computergrafik. Die beiden nächsten Abbildungen zeigen diese beiden Bilder<sup>20</sup>.



Abbildung 42: Datenreduktion durch Mittelwertbildung benachbarter Bildpunkte und 60% Datenreduktion

---

<sup>20</sup> Der Autor hofft, daß man die angesprochenen Unterschiede der Bilder auch nachher im Druck deutlich erkennen kann, alle Beurteilungen beziehen sich auf die Bildschirmdarstellung. Durch die Rasterung beim Druck können natürlich neue Störungen auftreten, aber vor allem bestimmte angesprochene Bildfehler eliminiert werden. In diesem Fall wird der Leser gebeten, sich die Originale der Bilder auf der beiliegenden CD anzuschauen, um das Gesagte nachzuvollziehen.



Abbildung 43: Datenreduktion durch Mittelwertbildung benachbarter Bildpunkte und 90% Datenreduktion

### **4.3 Untersuchung mit Hilfe der diskreten Fouriertransformation**

#### *4.3.1 Hochpaßfilterungen*

Ursprünglich war geplant, auch Untersuchungen mit Hochpaß-Filterungen an Linien- und Strichzeichnungen durchzuführen. Aus Platzgründen wurde jedoch auf den Abdruck und die weitere Untersuchung dieser Filterung verzichtet, da der Umfang der Arbeit sonst auf ein zu großes Format angewachsen wäre. Entsprechende theoretische Vorarbeit wurde bereits im 2. Kapitel geleistet und das Programm bietet die entsprechenden Berechnungsmöglichkeiten auch an. Es ist dem Leser also möglich, entsprechend eigene Untersuchungen für Hochpaßfilterungen durchzuführen.

Allgemein ist die Hochpaßfilterung in den meisten Fällen nicht so interessant bei Fotos von Objekten und realen Szenen, da diese in der Regel wesentlich mehr tiefe als hohe Frequenzen enthalten. Hochpaßfilterungen sind interessanter bei Strichzeichnungen, bei denen der hohe Frequenzanteil überwiegt und die einfarbigen Flächenteile als Bildhintergrund keine wesentlichen Informationen enthält.

#### 4.3.2 Tiefpaßfilterungen bei verschiedenen Reduktionsfaktoren



Abbildung 44: DFT mit anschließender Tiefpaßfilterung bei 60% Datenreduktion



Abbildung 45: DFT mit anschließender Tiefpaßfilterung bei 90% Datenreduktion

Betrachtet man Abbildung 45, so sieht man bei einer 90-prozentigen Datenreduktion ein Muster über dem Bild liegen, ähnlich dem Bild von Wellen oder eines Beugungseffektes. Dieser Effekt

stellt sich an den Farbkanten ein. Je größer der Helligkeitsunterschied, desto stärker die Wellen. Es handelt sich hierbei um das *Gibbsche Phänomen*, allerdings in etwas anderer Sichtweise als üblich. Normalerweise wird das Gibbsche Phänomen an periodischen oder aperiodischen Rechteckfunktionen gezeigt. Hier liegt der gleiche Effekt wie bei dem Sprung des Rechtecks vor, nur daß man „von oben“ auf die Sprungstelle sieht. Die Begründung ist die, daß mit abnehmender Grenzfrequenz des Filters immer mehr Oberwellen ausgeblendet werden, so daß die Kante nach der Reduktion nicht mehr gegen das Original konvergiert.

#### 4.3.3 Auswirkung der Fensterung bei verschiedenen Fenstergrößen



Abbildung 46: Gefensterte DFT bei einem Fenster 64x64 Punkte mit anschließender Tiefpaßfilterung bei 60% Datenreduktion



Abbildung 47: Gefensterte DFT bei einem Fenster 8x8 Punkte mit anschließender Tiefpaßfilterung bei 60% Datenreduktion

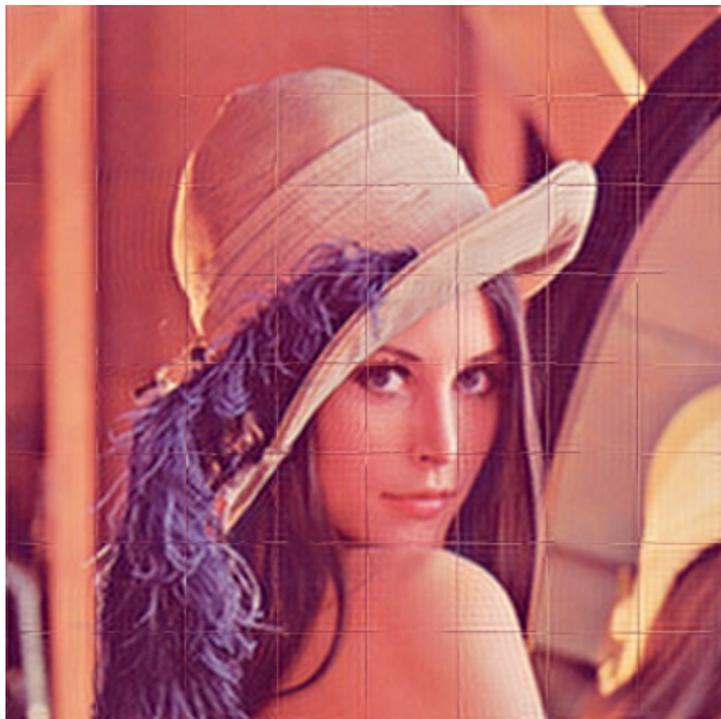


Abbildung 48: Gefensterte DFT bei einem Fenster 64x64 Punkte mit anschließender Tiefpaßfilterung bei 90% Datenreduktion



Abbildung 49: Gefensterte DFT bei einem Fenster 8x8 Punkte mit anschließender Tiefpaßfilterung bei 90% Datenreduktion

Bei der gefensterten DFT stellt sich in allen Fällen eine Störung durch ein überlagertes Gittermuster ein, das die Größe des gewählten Fensters besitzt. Dies hat seinen Grund in der Periodisierung der Eingangsfolgen, man kann z.B. sehr gut in Abbildung 49 sehen, daß bei einem Hell-Dunkel-Übergang im hellen Bereich noch dunkle Punkte auftauchen. Die Begründung läßt sich so formulieren, daß die DFT mit Filterung „glaubt“, daß tatsächlich dunkle Punkte an den hellen Bereich angrenzen --dies ist auch richtig, wenn man die Periodisierung berücksichtigt. Da eine Tiefpaßbildung Kanten „verschmiert“, tauchen also dunkle Punkte im hellen Farbbereich auf. Dieser Effekt macht sich vor allem bei einem größeren Gitter stärker bemerkbar, da hier die Bildstörung in größerer Entfernung zur Kante auftritt, die der Verursacher ist. Zusätzlich kann man bei einem größeren Gitter innerhalb des Fensters auch den zuvor bereits angesprochenen Beugungseffekt erkennen, was das Ergebnis insgesamt als nicht brauchbar erscheinen läßt.

## 4.4 Untersuchung mit Hilfe der diskreten Cosinustransformation

### 4.4.1 Tiefpaßfilterungen



Abbildung 50: DCT mit anschließender Tiefpaßfilterung bei 60% Datenreduktion

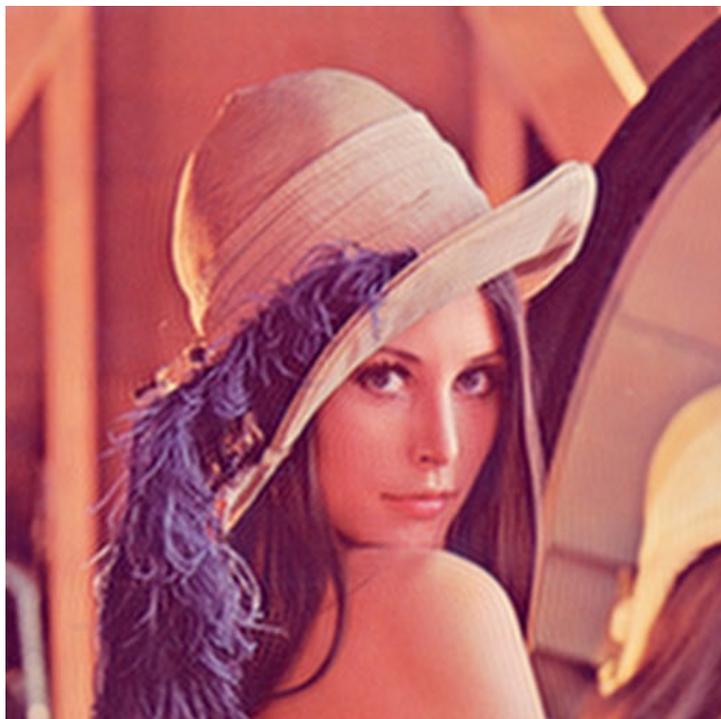


Abbildung 51: DCT mit anschließender Tiefpaßfilterung bei 90% Datenreduktion

Die Veränderung der Qualität bei der DCT mit Tiefpaßfilterung läßt keinen wesentlichen Unterschied zur DFT erkennen, allerdings ist der Beugungseffekt bei der DCT bei gleicher Reduktionsrate deutlich schwächer ausgeprägt. Der Autor vermutet, daß der Grund hierfür die höhere Auflösung der DCT im Frequenzbereich ist und die höhere Auflösung zu einem besseren Ergebnis führt.

#### 4.4.2 *Auswirkung der Fensterung bei verschiedenen Fenstergrößen*



Abbildung 52: Gefensterte DCT bei einem Fenster 64x64 Punkte mit anschließender Tiefpaßfilterung bei 60% Datenreduktion



Abbildung 53: Gefensterte DCT bei einem Fenster 8x8 Punkte mit anschließender Tiefpaßfilterung bei 60% Datenreduktion



Abbildung 54: Gefensterte DCT bei einem Fenster 64x64 Punkte mit anschließender Tiefpaßfilterung bei 90% Datenreduktion

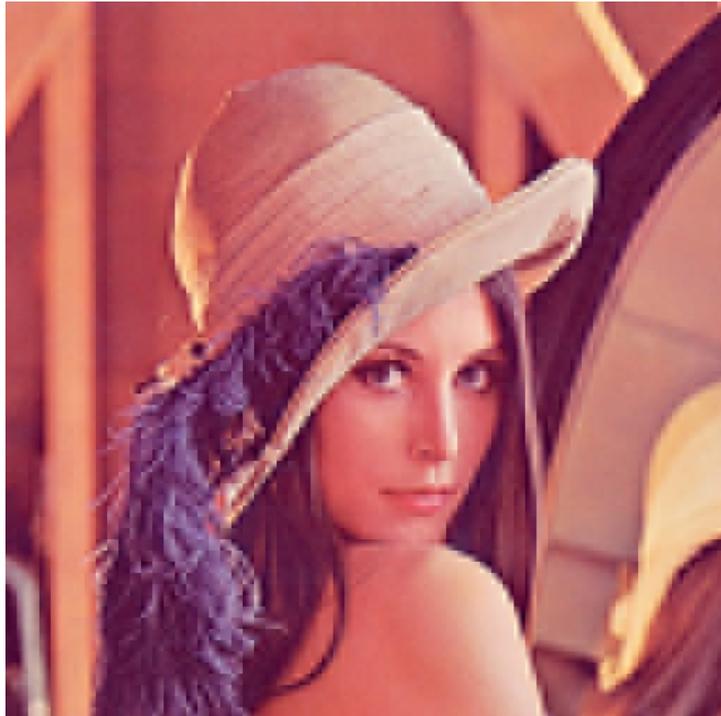


Abbildung 55: Gefensterte DCT bei einem Fenster 8x8 Punkte mit anschließender Tiefpaßfilterung bei 90% Datenreduktion

Betrachtet man die Bilder der gefensterten DCT, so sucht man zuerst nach dem störenden Gittereffekt, der das Ergebnis der gefensterten DFT ziemlich unbrauchbar machte. In der Tat ist das Gitter bei der gefensterten DCT aber verschwunden.

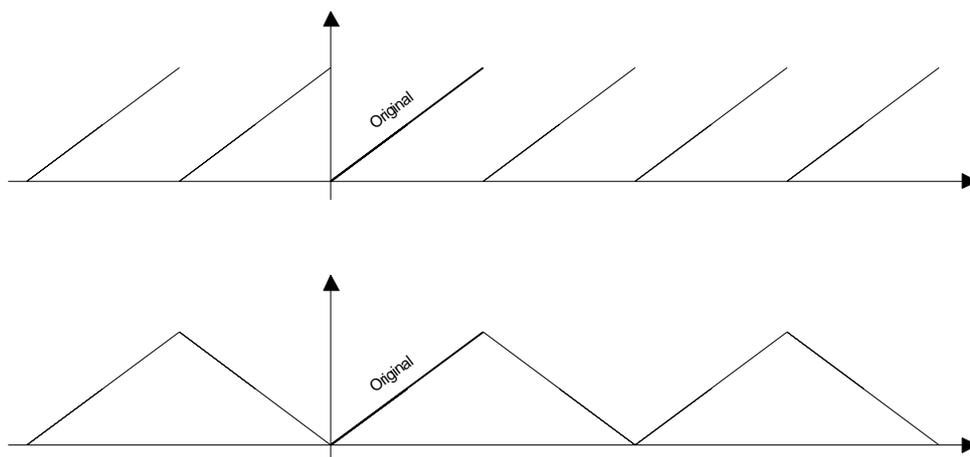


Abbildung 56: Unterschiedliche Auswirkung des Fenstereffektes bei DFT und DCT durch eine andere Art der Periodisierung

Dies liegt daran, daß der Fehler bei der DFT durch die Periodisierung der Teilfolgen entsteht, da hier im allgemeinen eine Sprungstelle erzeugt wird. Im Gegensatz dazu wird bei der DCT die Periodisierung erst durchgeführt, nachdem man aus der Originalfolge eine gerade Folge konstruiert,

und diese kann per Definition niemals eine Sprungstelle aufweisen. Daher kann bei der DCT diese auch als *Fenster-* oder *Leckeffekt* (*leakage effect*) bezeichnete Störung niemals auftreten.

Bei einem sehr kleinen Gitter und starker Reduktion durch eine niedrige Grenzfrequenz des Tiefpaßfilters tritt die Eigenschaft von Tiefpässen, auch als Mittelwertbildner zu arbeiten, in den Vordergrund. Das Bild besteht von der Tendenz her aus mehr oder wenige einfarbigen Kästchen, da nur noch der Gleichanteil und niederfrequente Informationen enthalten sind. Dies macht sich bei einem kleinen Gitter stark bemerkbar, da hier der Mittelwert von kleinen Bildteilen dargestellt wird. Der Mittelwert ist bei einer inhomogenen Farbverteilung und einer kleinen Anzahl von Punkten stärkeren Schwankungen ausgesetzt, als bei einer größeren Anzahl von Punkten.

## **4.5 Untersuchung mit Hilfe der diskreten Wavelettransformation**

### *4.5.1 Unterdrückung des Wavelet-Hochpaßanteils*

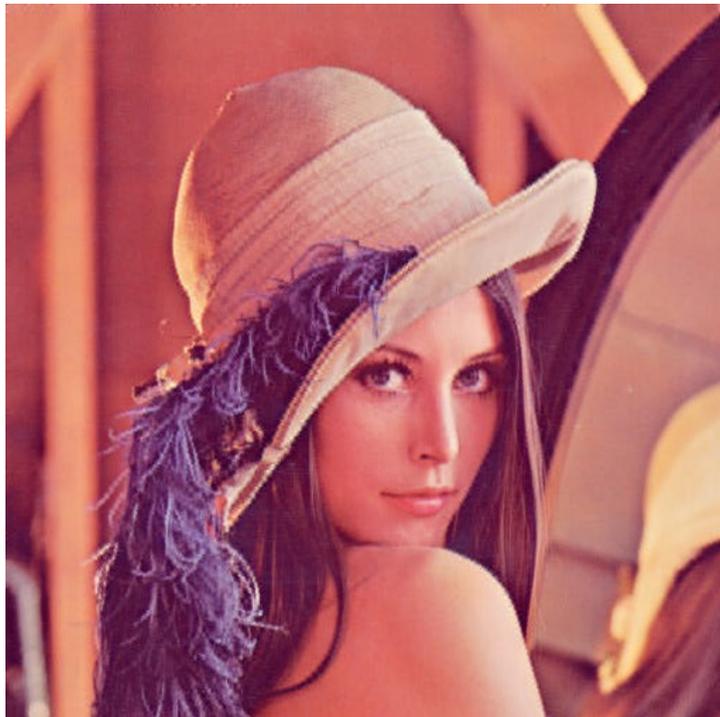


Abbildung 57: DWT mit erfolgter Unterdrückung der Wavelet-Anteile bei einer Datenreduktion von 60% und Verwendung eines Daubechies-Filters 2. Ordnung (auch als Haar-Walsh-Filter bekannt)

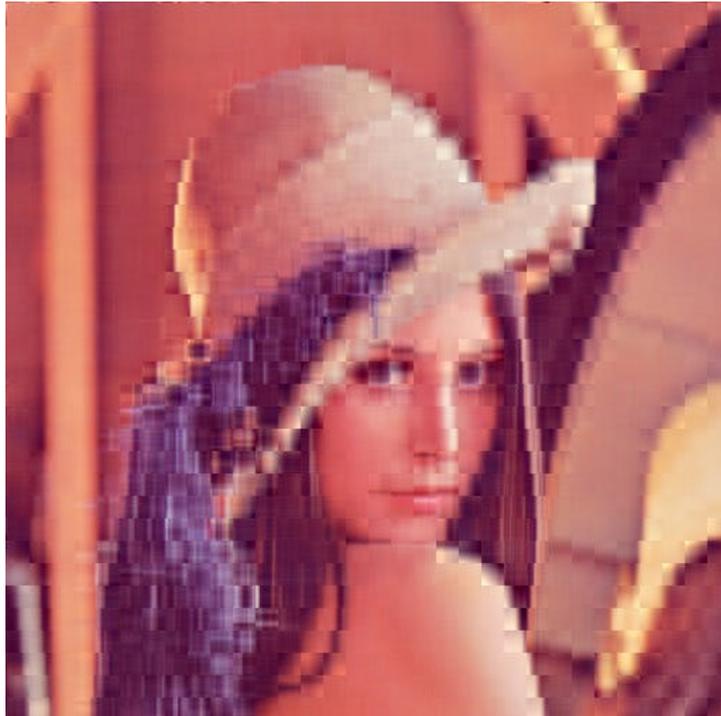


Abbildung 58: DWT mit erfolgter Unterdrückung der Wavelet-Anteile bei einer Datenreduktion von 90% und Verwendung eines Daubechies-Filters 2. Ordnung (auch als Haar-Walsh-Filter bekannt)

Bei einer höheren Reduktionsrate sieht man sehr rasch, daß das Bild aus Rasterflächen besteht, die ähnlich wie bei der gefensterten DCT mit kleinem Fenster aussehen. Der Autor vermutet, daß dies an der Gestalt des Haar-Walsh-Wavelets liegt, das in diesem Fall verwendet wird. Dieses Mutterwavelet, und somit auch seine Translate, nimmt nur diskrete Werte an, im Gegensatz zu den Zwischenwerten, die von den anderen Wavelets angenommen werden. Daher tritt der Effekt einer Art Mittelwertbildung bezüglich der verschiedenen Skalen ein, so daß nach einer Weglassung der höherfrequenten Waveletanteile nur noch ein Mittelwert als Ergebnis bleibt – und dessen Einfluß auf das Aussehen des Bildes wurde bereits zuvor erläutert.

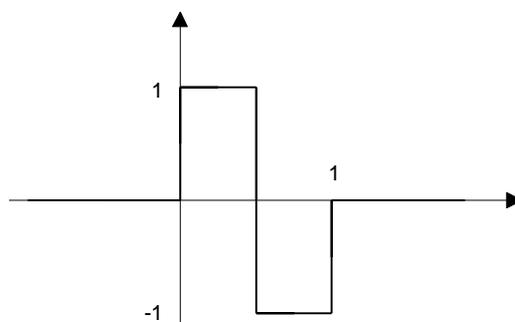


Abbildung 59: Das Haar-Walsh-Wavelet (auch als Daubechies-2-Wavelet bekannt)

#### 4.5.2 Auswirkungen der Filterlänge

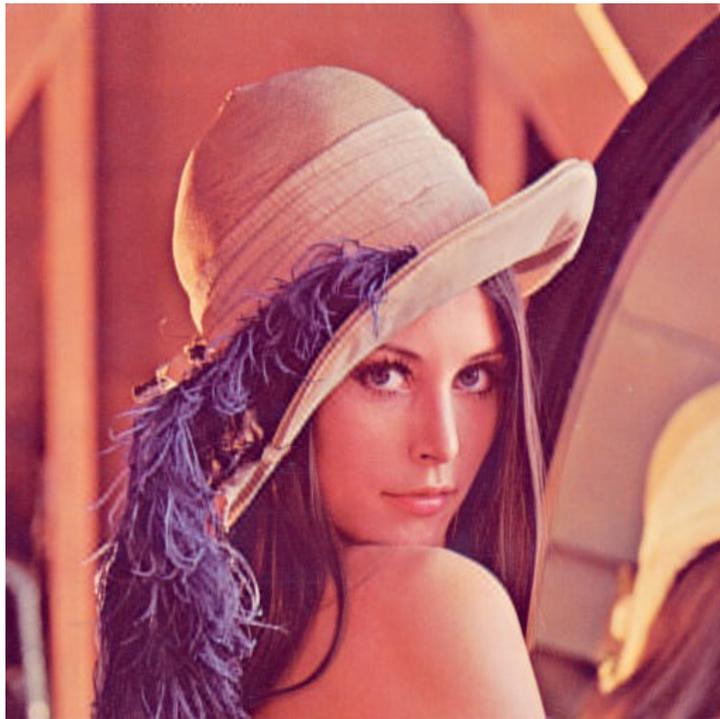


Abbildung 60: DWT mit erfolgter Unterdrückung der Wavelet-Anteile bei einer Datenreduktion von 60% und Verwendung eines Daubechies-Filters 20. Ordnung

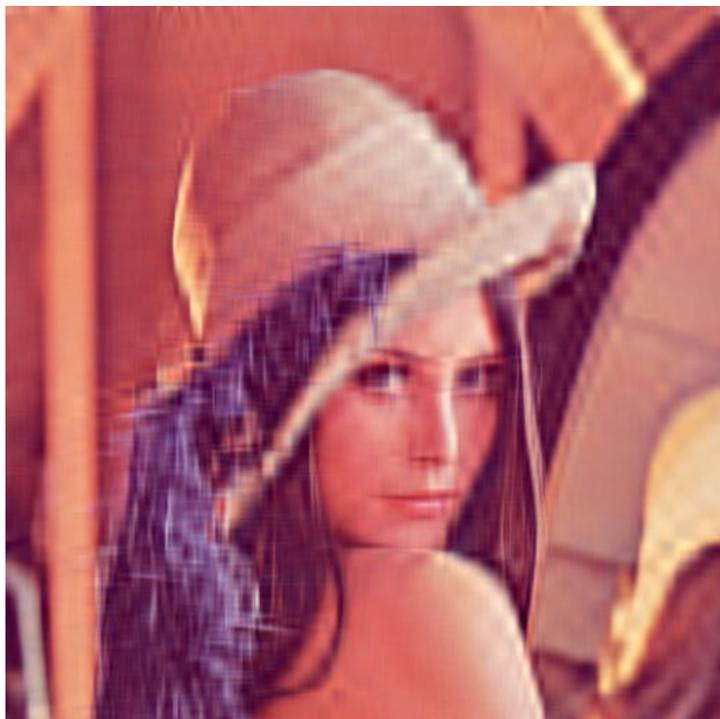


Abbildung 61: DWT mit erfolgter Unterdrückung der Wavelet-Anteile bei einer Datenreduktion von 90% und Verwendung eines Daubechies-Filters 20. Ordnung

Die Verwendung eines Filters höherer Ordnung zeigt deutlich eine Verbesserung, da nun der starke Rastereffekt des Filters niedrigerer Ordnung verschwindet. Allerdings wirkt das Bild insgesamt etwas unscharf, wobei die Störungen über die gesamte Bildfläche verteilt sind und nicht lokal gehäuft auftreten. Dieser Effekt tritt bei allen im Programm zu Verfügung stehenden Filtern auf, sobald eine Ordnung höher 4 gewählt wird. Daraus läßt sich folgern, daß man wohl durch eine Veränderung der Ausblendbereiche (siehe Abbildung 34 auf Seite 57) bei der Waveletfilterung zusätzliche Schärfe ins Bild bringen läßt, was in von [LOU-94] bestätigt wird.

#### **4.6 Original oder Fälschung – wie man das Auge täuscht**

An diesem Punkt ist der Leser nun an einer Stelle angekommen, wo die theoretischen Möglichkeiten der Datenreduktion aufgezeigt wurden, auch sollte klar sein, wie die Methoden mathematisch realisiert werden. Die Beispiele dieses Kapitels zeigen sehr gut, daß Bilder mit nur 90% der Informationen immer noch aussehen können wie das Original.

Eine Frage bleibt trotz der bisherigen Ausführungen noch offen: *warum* funktionieren diese Methoden eigentlich? Wieso fällt das Auge –oder das Gehirn –auf diesen Trick herein?

Zwei Dinge bleiben auf jeden Fall festzuhalten:

- Die Bilder sind nicht mehr identisch zum Original und
- Das Auge bemerkt den Unterschied bis zu einer gewissen Grenze praktisch nicht.

Um diesen Sachverhalt aufzuklären, wären einige neurobiologischen Untersuchungen notwendig, die diese Arbeit sowohl thematisch als auch vom Umfang her sprengen würden. Dennoch soll auf einen Ansatz einer Erklärung mit einigen beispielhaften Erklärungsversuchen nicht verzichtet werden. Die Ausführungen fassen einige in [KOR-82] gemachte Aussagen zusammen.

Alle realen System der Meßtechnik sind bandbegrenzt und weisen eine Tiefpaß- oder Bandpaßstruktur auf. Ein allgemein auftretender Effekt ist der, daß ab einer gewissen Frequenz das System nicht mehr in der Lage ist, sein Übertragungsverhalten beizubehalten und die Dämpfung der Eingangswerte gegen  $\infty$  geht. Je nach Systemtyp (mechanisch, elektrisch, optisch, biologisch) sind unterschiedliche physikalische Gründe hierfür maßgeblich, die jedoch immer wieder auf das gleiche Systemverhalten führen. Auch das Auge mit der zugehörigen Auswertelogik – es soll hier nicht biologisch diskutiert werden, wo und wie Auge, Nerven und Hirn zu trennen sind – unterliegt diesen physikalischen Gesetzen.

[KOR-82] führt dazu aus, daß beim Auge die „Trennschärfe“ von einzelnen Linien und Punkten der Übertragung von Ortsfrequenzen eine *cut-off-frequency*<sup>21</sup> entgegengesetzt. Dies hat in erster Linie geometrische Gründe, die mit der Verteilung und Dichte der Sehzellen auf der Netzhaut zusammenhängen. Das System „Auge – Nerven – Gehirn“ weist also ebenfalls eine Tiefpaßcharakteristik auf und kann bestimmte Ortsfrequenzen nicht mehr übertragen. Wichtig ist dabei, daß es sich hier um eine Tiefpaßcharakteristik bezogen auf die Ortsfrequenzen handelt – das System besitzt auch ein Tiefpaßverhalten bezogen auf die Bildwiederholfrequenz von bewegten Bildern, das hier aber *nicht* gemeint ist.

Mit diesem Verhalten läßt sich in groben Zügen erklären, warum das Auge die fehlenden höheren Frequenzen nicht bemerkt. Man betrachte hierzu die folgende Abbildung 62. Betrachtet man die Bildinformationen im Ortsfrequenzbereich, so liegen auch Informationen außerhalb der *cut-off-frequency* vor. In der rechten Bildhälfte ist dabei jeweils die Information dargestellt, die nach Gewichtung mit der Systemverstärkung übrig bleibt. Es wird vom Prinzip her deutlich, daß ein Weglassen von nur einem kleinen Teil der hohen Frequenzen zunächst nicht bemerkt wird, da dieser Frequenzbereich bereits im Bereich der abfallenden Systemverstärkung liegt. Erst wenn Frequenzen weggelassen werden, die im Bereich des konstanten Übertragungsverhaltens liegen, kann das Auge dies bemerken. Allerdings sollte man hier beachten, daß dieses Beispiel nur die prinzipiellen Vorgänge verdeutlicht, insgesamt hängt dies natürlich auch von weiteren Faktoren ab, die einen Einfluß ausüben können, z.B. Umgebungshelligkeit, Auflösung und Größe des Bildes, usw.

Weiterhin wurde hier auch noch nicht berücksichtigt, daß im Bild bei Farbbildern wie Fotos in der Regel die Energie stärker im Bereich der niedrigen Frequenzen als im Bereich der hohen Frequenzen liegt – die Dämpfung des Auges wird hier zusätzlich durch die Abnahme der Energie des Eingangssignals in diesem Frequenzbereich unterstützt, so daß die fehlenden Informationen noch schlechter bemerkt werden.

---

<sup>21</sup> Frequenz, ab der ein Eingangssignal nur noch mit einer Verstärkung kleiner 1 übertragen wird.

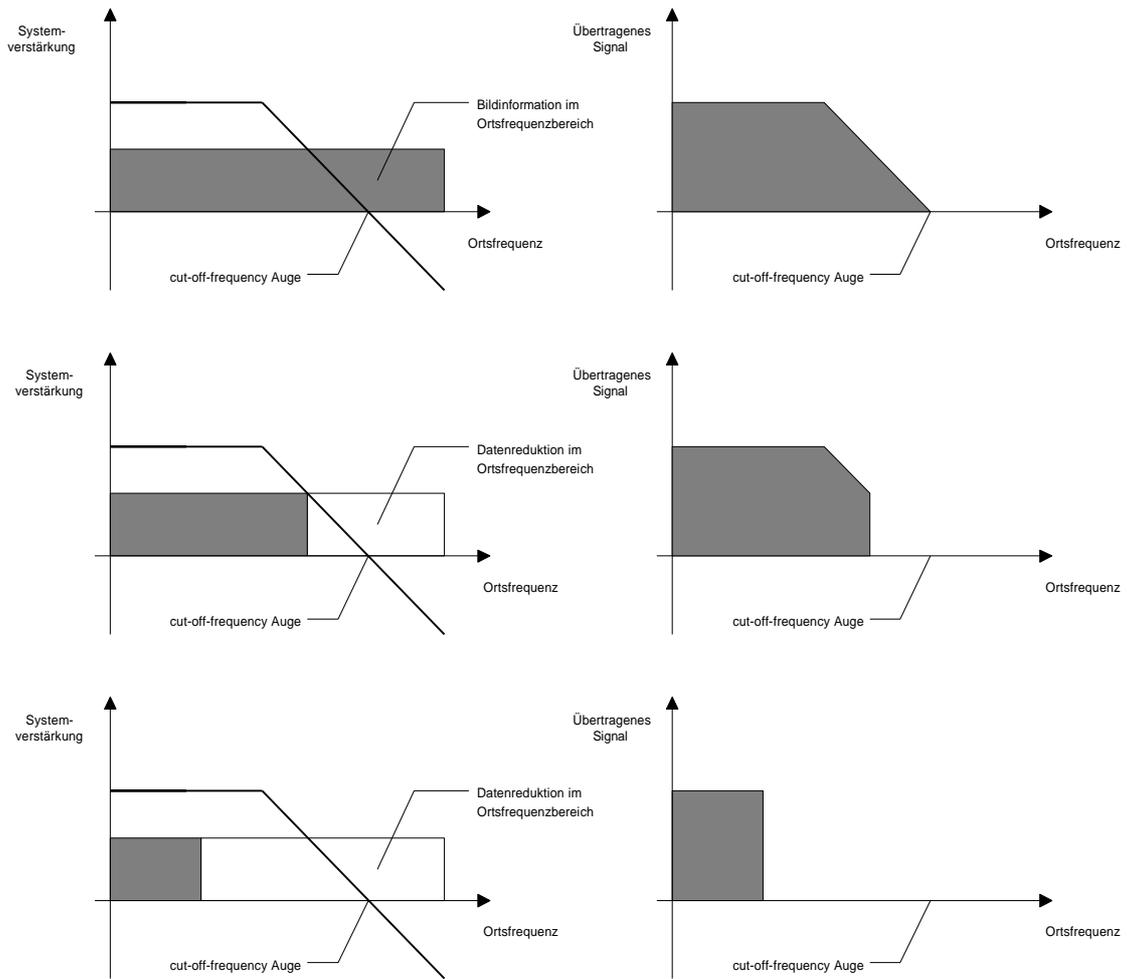


Abbildung 62: Zusammenhang zwischen dem Übertragungsverhalten des Auges und dem Weglassen von Bildinformationen im Ortsfrequenzbereich.

## 5 Ergebnisse

Vergleicht man die Bilder, die im vorigen Abschnitt dargestellt wurden, so ist man sicherlich überrascht, wie wenig Informationen eigentlich notwendig sind, um ein Bild auch ohne sinnentstellende Verzerrungen noch erkennen zu können. Mit allen vorgestellten Verfahren sind problemlos Reduktionsraten um die Hälfte erreichbar, ohne daß dem Betrachter ein Qualitätsverlust deutlich wird.

Erst bei wesentlich härteren Anforderungen an die Reduktionsrate werden die Nachteile einzelner Verfahren erkennbar. Führt man die Berechnungen außerdem auch direkt am Computer durch, so kommt auch noch der Faktor Rechenzeit zum tragen. Dabei gehören die diskrete Wavelettransformation und die gefensterten Fouriertransformationen (DFT und DCT) zu den schnelleren Methoden. Aber gerade die gefensterten Transformationen zeigen sehr schnell die Grenzen dieser Idee: an den Nahtstellen der Fenster sind bei hohen Reduktionsraten sehr deutliche „Beugungsgitter“ erkennbar, das Bild scheint mit einem Gitter überlegt zu sein, was die Brauchbarkeit dieser Verfahren doch erheblich einschränkt. Ebenfalls macht sich das Gibbsche Phänomen als Geisterbild an kontrastreichen Kanten störend bemerkbar. Diese Effekte sind dabei bei der DFT wesentlich stärker ausgeprägt als bei der DCT. Ähnliches kann der aufmerksame Computernutzer auch bei der Anwendung des JPEG-Dateiformats beobachten, wenn die Reduktionsrate hier zu hoch eingestellt wird, Beispiele finden sich hierfür auch häufig auf Seiten im World-Wide-Web, wo die Autoren der Seiten eine zu hohe Reduktionsrate eingestellt haben, um Speicher zu sparen.

Unter diesen Umständen scheinen sich zwei Gruppen von Verfahren als gute Datenreduktionsverfahren anzubieten: die diskrete gefensterte Cosinustransformation mit einer nicht zu kleinen Fenstergröße, sowie die diskrete Wavelet-Transformation mit einem Filter höherer Ordnung.

Für die gefensterte DCT spricht die gute Beherrschbarkeit der Methode, da man sehr leicht die Einflüsse von Filtern elementar erklären kann. Dagegen besitzt die diskrete Wavelettransformation die Vorteile, daß

- die numerische Durchführung der Transformation sehr schnell ist und
- es werden die Informationen des ganzen Bildes verarbeitet, es treten keine lokalen Fehler an künstlich erzeugten Übergangsstellen auf.

Erschwerend wirkt sich hier aus, welche Koeffizientenbereiche der Transformierten eines Bildes als vernachlässigbar einzustufen sind, das hier vorgestellte Verfahren scheint hier durchaus noch ver-

besserungswürdig zu sein, es erscheint sinnvoll, auch einen Bereich mit Kanteninformationen von der Ausblendung zu verschonen. Die jetzige Strategie der Weglassung von Details (Abbildung 34 auf Seite 57) ist relativ naiv und sicherlich noch optimierbar, [LOU-94] nennt hierfür einige weitere Ansätze, die die Verstärkung der Kanten betrifft und die vorhandenen Fehler verringern kann.

Zusätzlich notwendig ist ebenfalls der Einsatz von speziell für die Bildverarbeitung entwickelten Filtern. Die bekannten biorthogonalen Filter, wie sie auch hier verwendet wurden, sind in erster Linie für Anwendungen der eindimensionalen Welt entwickelt worden, und sollten durch für die Bildverarbeitung und Datenreduktion optimierte Filter, also durch entsprechende Waveletbasen, weiter verbessert werden, um das starke Verwischen des Bildes zu vermindern.

Zudem fallen hier auch weitere Optimierungsmöglichkeiten ins Auge. Im Moment wird bei den in dieser Arbeit gezeigten Verfahren immer ein ganzer Bereich der Transformierten ausgeblendet, unter der Annahme, daß die betragskleinen Koeffizienten tatsächlich in diesem Bereich liegen – was in guter Näherung auch stimmt, aber nicht zwangsläufig ist. Bei allen Verfahren wird es also bei praktischer Anwendung vorkommen, daß man Koeffizienten eliminiert, die besser im Bild geblieben wären. Der starre Ausblendefilter in der jetzigen Form sollte also durch eine adaptive Filterung ersetzt werden, bei der ein Filter sich die Teilbereiche der Transformierten mit betragskleinen Koeffizienten selbst sucht und nur diese Teilbereiche verschwinden läßt.

Im Extremfall würde dies sogar bedeuten, daß man alle Punkte der Transformierten nach ihren Beträgen sortiert, und „von hinten her“ die kleinsten Koeffizienten mit einer einstellbaren Schwelle auf 0 setzt. Dies ist der Grenzfall, der nicht praktikabel sein wird, da in diesem Falle für jeden Koeffizienten eine Ortsinformation mitgeführt werden muß, da der Punkt ja bei der Rekonstruktion wieder an der richtigen Stelle im Ortsfrequenzbereich eingefügt werden soll. In diesem Grenzfall wird der Overhead für die Koordinatenverwaltung den eingesparten Speicher der verschwindenden Koeffizienten wieder verbrauchen, so daß effektiv keine Reduktion stattfindet. Die folgende Abbildung läßt aber ahnen, daß es irgendwo zwischen den beiden Extremen der starren Filterung und der idealen, tatsächlich betragsabhängigen Filterung, einen minimalen Speicherplatzbedarf geben wird.

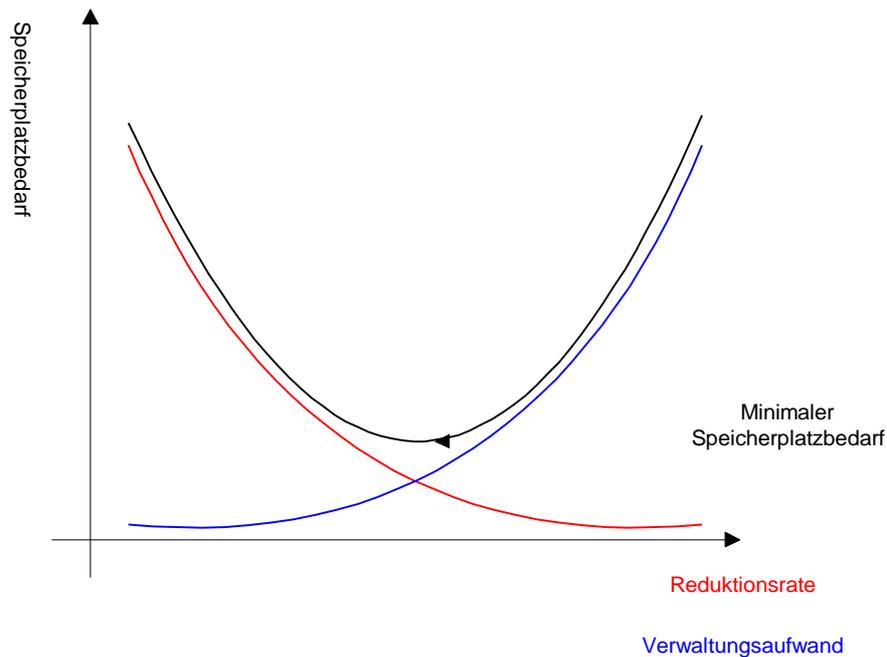


Abbildung 63: Darstellung des prinzipiellen Verlaufs des Speicherplatzbedarfs eines Bildes bei Betrachtung von Reduktionsrate und Verwaltungsaufwand zur Speicherung der Restinformationen

Außerdem ist die Datenreduktion nur ein Teilbereich des Problems, wie man den Speicherbedarf eines Bildes verringern kann. Zum anderen gibt es auch noch den Teilbereich, der sich mit besonders trickreichen Codierungsverfahren beschäftigt, um die nach der Reduktion immer noch vorhandenen Informationen in einer Form darzustellen, die möglichst wenig Speicher benötigt. Dazu gehört unter anderem auch eine andere Separation der Farbebenen, wo nicht nach den Grundfarben der Farbmischung, sondern nach Differenz- und Helligkeitsinformationen getrennt wird. Mit diesem Problem fand in dieser Arbeit keine Auseinandersetzung statt – dies bedeutet aber auch, daß sich bei konsequenter Nutzung dieser Verfahren ebenfalls weitere der begehrten Speicherplätze einsparen lassen.

Sind heute mit dem bereits auf dem Markt eingeführten bekannten JPEG-Verfahren Kompressions- und Reduktionsraten von auf ein Zehntel oder noch weniger erreichbar, so darf man gespannt sein, welche Reduktionsraten sich in einigen Jahren bei einer weiteren Nutzung der Wavelet-Transformation erreichen lassen, wenn man sowohl Reduktions- als auch Kompressionsverfahren miteinander kombiniert. Reduktionsraten um bei der Waveletkompression um den Faktor 350 sind mit Unterstützung effizienter Codierungsverfahren realistisch, entsprechende elektronische Bausteine sind am Markt erhältlich und werden bereits in handelsüblicher Grafikhardware für den Heimbereich vereinzelt eingesetzt ([ANA-98]).

Vielleicht gelingt uns Ingenieuren, Mathematikern und Wissenschaftlern dann endlich das, wovon der Computernutzer heute noch träumt:

–daß der Download der Mona Lisa nicht mehr Minuten, sondern nur noch einen „Augenblick“ dauert – wir werden es sehen.

## ABBILDUNGSVERZEICHNIS

- Abbildung 1: Computergrafik einst und jetzt. Während das linke Bild vor 15 Jahren noch leicht einen Grafikkwettbewerb gewonnen hätte, zeigt das rechte Bild die heutigen Ansprüche an eine Computergrafik. Die Unterschiede sind deutlich sichtbar.
- Abbildung 2: Ein simples Beispiel für eine Datenreduktion. Die Näherung an die Zahl  $\pi$  nimmt in beiden Fällen unterschiedlich viele Symbole = Speicherstellen ein.
- Abbildung 3: Eine einfache Datenreduktion durch Verkleinerung des Originalbildes auf 1/16 der ursprünglichen Größe, wodurch 94% der Bildpunkte wegfallen. Die Rekonstruktion auf die Originalgröße rechts zeigt deutlich sichtbar die Einbußen der Bildinformation.
- Abbildung 4: Jean Baptiste Joseph Fourier, 1768 - 1830, Mathematiker, Physiker und Ägyptologe
- Abbildung 5: Ingrid Daubechies, Professor für angewandte und numerische Mathematik an der Universität Princeton (geb. 1954)
- Abbildung 6: Eine Datenreduktion um 94% durch Anwendung einer diskreten gefensternten Cosinustransformation und Tiefpaßfilterung.
- Abbildung 7: Die Rechteckfunktion *rect* und ihre Fouriertransformierte, die Sinus-Cardinalis Funktion (*sinc*- oder auch *si-Funktion*)
- Abbildung 8: Ein Sprung mit 32 Stützstellen
- Abbildung 9: Der Realteil der diskreten Fouriertransformierten eines Sprunges mit 32 Stützstellen
- Abbildung 10: Der Imaginärteil der diskreten Fouriertransformierten eines Sprunges mit 32 Stützstellen
- Abbildung 11: Ein Querschnitt mit 128 Punkten durch ein fiktives Grauwertbild mit glatten Grauwertverläufen, wie man sie für ein reales Bild annehmen kann
- Abbildung 12: Die diskrete Fouriertransformierte für das Grauwertbild, nach Real- und Imaginärteil
- Abbildung 13: Das Grauwertbild nach Nullsetzen der diskreten Fourierkoeffizienten  $X[40]..X[80]$
- Abbildung 14: Allgemeine Darstellung eines Systems mit Filter
- Abbildung 15: Prinzipdarstellung des Durchlaßbereichs eines Tiefpaßfilters für die DFT
- Abbildung 16: Prinzipdarstellung des Durchlaßbereichs eines Hochpaßfilters für die DFT
- Abbildung 17: Prinzipdarstellung des Durchlaßbereichs eines Hochpaßfilters für die DFT, bei dem auch der Gleichanteil zum Durchlaßbereich gehört
- Abbildung 18: Periodische Fortsetzung des Originalbildes nach Zeilen und Spalten
- Abbildung 19: Filtersystem für Bildmatrizen
- Abbildung 20: Durchlaßbereich des Tiefpaßfilters für Bildmatrizen
- Abbildung 21: Durchlaßbereich des Hochpaßfilters für Bildmatrizen
- Abbildung 22: Durchlaßbereich des Hochpaßfilters für Bildmatrizen mit Durchlaß des Gleichanteils
- Abbildung 23: Prinzipdarstellung des Durchlaßbereichs eines Tiefpaßfilters für die DCT
- Abbildung 24: Prinzipdarstellung des Durchlaßbereichs eines Hochpaßfilters für die DCT
- Abbildung 25: Prinzipdarstellung des Durchlaßbereichs eines Hochpaßfilters für die DCT, bei dem auch der Gleichanteil zum Durchlaßbereich gehört
- Abbildung 26: Durchlaßbereich des Tiefpaßfilters für Bildmatrizen bei der DCT
- Abbildung 27: Durchlaßbereich des Hochpaßfilters für Bildmatrizen bei der DCT
- Abbildung 28: Durchlaßbereich des Hochpaßfilters mit Gleichanteil für Bildmatrizen bei der DCT
- Abbildung 29: Ein „Peak“, also eine Impulsfolge, im Zeitbereich
- Abbildung 30: Der Betrag der diskreten Fouriertransformierten der Impulsfolge
- Abbildung 31: Wird die Folge aus Abbildung 29 in Viertel zerlegt und getrennt transformiert, so entstehen die vier gezeigten Fouriertransformierten (es sind nur die Beträge dargestellt).

- Abbildung 32: Diskrete Wavelet-Transformation und die Zerlegung über mehrere Skalen, H ist eine Operation zur Tiefpaßfilterung, G eine Operation zur Hochpaßfilterung
- Abbildung 33: Datenreduktion mit zunehmendem Reduktionsfaktor über verschiedene Skalen
- Abbildung 34: Bilddatenreduktion bei der zweidimensionalen diskreten Wavelettransformation bei einer Matrix der Größe  $64 \times 64$  Elementen
- Abbildung 35: Positionierung eines beliebigen Bildes in einer quadratischen Matrix, bei der die Kantenlänge eine Potenz von 2 ist.
- Abbildung 36: Beispiel für die Transformation eines Bildes durch Zerlegung, getrennte Transformation, Datenreduktion und anschließende Rekombination zu einem neuen Farbbild
- Abbildung 37: Aufbau einer Matrix im Speicher
- Abbildung 38: Verarbeitung eines Farbbildes in zeitlicher Reihenfolge. Dick umrandet die Operationen, die mit der Fließkomma-Matrix ausgeführt werden, dünn umrandet die Operationen mit der bitorientierten Bitmap (DIB).
- Abbildung 39: Übersicht über die verwendeten Verfahren zur Bildreduktion eines Echtfarbbildes
- Abbildung 40: Beispiel für Anzahl der übertragenden Punkte nach Datenreduktion
- Abbildung 41: Lena im Original mit  $512 \times 512$  Punkten
- Abbildung 42: Datenreduktion durch Mittelwertbildung benachbarter Bildpunkte und 60% Datenreduktion
- Abbildung 43: Datenreduktion durch Mittelwertbildung benachbarter Bildpunkte und 90% Datenreduktion
- Abbildung 44: DFT mit anschließender Tiefpaßfilterung bei 60% Datenreduktion
- Abbildung 45: DFT mit anschließender Tiefpaßfilterung bei 90% Datenreduktion
- Abbildung 46: Gefensterter DFT bei einem Fenster  $64 \times 64$  Punkte mit anschließender Tiefpaßfilterung bei 60% Datenreduktion
- Abbildung 47: Gefensterter DFT bei einem Fenster  $8 \times 8$  Punkte mit anschließender Tiefpaßfilterung bei 60% Datenreduktion
- Abbildung 48: Gefensterter DFT bei einem Fenster  $64 \times 64$  Punkte mit anschließender Tiefpaßfilterung bei 90% Datenreduktion
- Abbildung 49: Gefensterter DFT bei einem Fenster  $8 \times 8$  Punkte mit anschließender Tiefpaßfilterung bei 90% Datenreduktion
- Abbildung 50: DCT mit anschließender Tiefpaßfilterung bei 60% Datenreduktion
- Abbildung 51: DCT mit anschließender Tiefpaßfilterung bei 90% Datenreduktion
- Abbildung 52: Gefensterter DCT bei einem Fenster  $64 \times 64$  Punkte mit anschließender Tiefpaßfilterung bei 60% Datenreduktion
- Abbildung 53: Gefensterter DCT bei einem Fenster  $8 \times 8$  Punkte mit anschließender Tiefpaßfilterung bei 60% Datenreduktion
- Abbildung 54: Gefensterter DCT bei einem Fenster  $64 \times 64$  Punkte mit anschließender Tiefpaßfilterung bei 90% Datenreduktion
- Abbildung 55: Gefensterter DCT bei einem Fenster  $8 \times 8$  Punkte mit anschließender Tiefpaßfilterung bei 90% Datenreduktion
- Abbildung 56: Unterschiedliche Auswirkung des Fenstereffektes bei DFT und DCT durch eine andere Art der Periodisierung
- Abbildung 57: DWT mit erfolgter Unterdrückung der Wavelet-Anteile bei einer Datenreduktion von 60% und Verwendung eines Daubechies-Filters 2. Ordnung (auch als Haar-Walsh-Filter bekannt)
- Abbildung 58: DWT mit erfolgter Unterdrückung der Wavelet-Anteile bei einer Datenreduktion von 90% und Verwendung eines Daubechies-Filters 2. Ordnung (auch als Haar-Walsh-Filter bekannt)
- Abbildung 59: Das Haar-Walsh-Wavelet (auch als Daubechies-2-Wavelet bekannt)
- Abbildung 60: DWT mit erfolgter Unterdrückung der Wavelet-Anteile bei einer Datenreduktion von 60% und Verwendung eines Daubechies-Filters 20. Ordnung

- Abbildung 61: DWT mit erfolgter Unterdrückung der Wavelet-Anteile bei einer Datenreduktion von 90% und Verwendung eines Daubechies-Filters 20. Ordnung
- Abbildung 62: Zusammenhang zwischen dem Übertragungsverhalten des Auges und dem Weglassen von Bildinformationen im Ortsfrequenzbereich.
- Abbildung 63: Darstellung des prinzipiellen Verlaufs des Speicherplatzbedarfs eines Bildes bei Betrachtung von Reduktionsrate und Verwaltungsaufwand zur Speicherung der Restinformationen

Bildnachweis: Abbildung 4 stammt aus [KES-98], Abbildung 5 stammt aus [PRI-98], alle anderen Abbildungen sind von Marcus Bäckmann, August 1998

## LITERATURVERZEICHNIS

- [AMA-98] Amara's Wavelet Page. <http://www.amara.com/current/wavelet.html>, Version der Seite von 1998
- [ANA-98] Analog Devices. Low Cost Multiformat Video Codec. Datenblatt, 1998<sup>22</sup>
- [BÄC-95] Bäckmann, Marcus. *Vortrag: Diskrete Fouriertransformation und Anwendungen*. Fachhochschule Darmstadt, SS 1995, Vortrag für die Vorlesung "Signalprozessoren" bei Herrn Prof. Dr. Münter<sup>23</sup>
- [BET-84] Beth, Thomas. *Verfahren der schnellen Fourier-Transformation*. 1. Aufl. Stuttgart: Teubner, 1984
- [BRI-95] Brigham, Elbert Oran. *FFT: schnelle Fouriertransformation*. 6. Aufl. München: Oldenburg, 1995
- [HAB-91] Haberäcker, Peter. *Digitale Bildverarbeitung*. 4. Aufl. München; Wien: Hanser, 1991
- [HUB-97] Hubbard, Barbara Burke: *Wavelets: Die Mathematik der kleinen Wellen*. 1. Aufl. Berlin: Birkhäuser, 1997
- [KES-98] Keston, David. *Joseph Fourier - Politician & Scientist*.  
<http://www.astro.gla.ac.uk/~davidk/fourier.htm>, Version der Seite von 1998
- [KOR-82] Korn, Axel: *Fachberichte Messen – Steuern – Regeln. Bildverarbeitung durch das visuelle System*. 1. Aufl. Berlin; Heidelberg; New York: Springer, 1982
- [LEN-98] Waterloo Brag Zone. <http://links.uwaterloo.ca/bragzone.base.html>, Testbild "Lena.tif" direkt <ftp://links.uwaterloo.ca/pub/BragZone/ColorSet/Lena/>, Version der Seite von 1998
- [LIP-97] Lipp, Thomas: *Grafikformate*. 1. Aufl. Unterschleißheim: Microsoft Press, 1997
- [LOC-92] Locher, Franz: *Mathematik III*. FernUniversität – Gesamthochschule – Hagen, Studienbrief zu Kurs Nr. 1173, 1992
- [LOC-94] Locher, Franz; Lenze, Burkhard: *Cebyshev-Polynome und Fourier-Analysis*. FernUniversität – Gesamthochschule- Hagen, Studienbrief zu Kurs Nr. 1378, 1994
- [LOC-96] Locher, Franz: *Mathematische Systemtheorie*. FernUniversität – Gesamthochschule- Hagen, Studienbrief zu Kurs Nr. 1370, 1996
- [LOU-94] Louis, Alfred K.; Maass, Peter; Rieder, Andreas: *Wavelets: Theorie und Anwendungen*. 1. Aufl. Stuttgart: Teubner, 1994
- [MIL-90] Mildenberger, Otto: *Informationstheorie und Codierung*. 1. Aufl. Braunschweig: Friedr. Vieweg & Sohn, 1990
- [OPP-92] Oppenheim, Alan; Willsky, Alan. *Signale und Systeme*. 2. Aufl. Weinheim: VCH-Verlagsgesellschaft, 1992
- [POL-98] Polikar, Robi: *The Wavelet Tutorial. The engineer's ultimate guide to wavelet analysis*.  
<http://www.public.iastate.edu/~rpolikar/WAVELETS/WTtutorial.html>,  
Version der Seite von 1998

---

<sup>22</sup> Dieses Datenblatt –sowie einige andere dieser Firma –befinden sich auf der beigefügten CD als PDF-Datei.

<sup>23</sup> Der Text des Vortrags ist auf der beigefügten CD als PDF-Datei vorhanden.

- [PRI-98] University Princeton. *Curriculum Vitae of Ingrid Daubechies*.  
<http://www.princeton.edu/~icd>, Version der Seite von 1998
- [STE-95] Steinmetz, Ralf: *Multimedia-Technologie*. 1. Aufl. Berlin; Heidelberg; New York: Springer, 1995
- [SUR-98] *Surfing the wavelets*. <http://www.monash.edu.au/cmcm/wavelet/wavelet.htm>,  
Version der Seite von 1998
- [WAV-98] Wavelet.Org. *The Wavelet Digest*.<sup>24</sup> <http://www.wavelet.org/wavelet/index.html>,  
Version der Seite von 1998
- [WIC-96] Wickerhauser, Mladen Victor: *Adaptive Wavelet-Analysis. Theorie und Software*. 1. Aufl.  
Braunschweig/Wiesbaden: Friedr. Vieweg & Sohn, 1996

---

<sup>24</sup> Hier ist unter anderem das Abonnement des monatlich per Email erscheinenden „The Wavelet Digest“ möglich.

## INDEX

		DWT-Algorithmus	72
— A—		— E—	
Abtastfolge	11, 18	Energiesignal	8
Abtastfrequenz	18	— F—	
Abtastintervall	11	Faltungssätze	22
Abtastrate	11	Fenster	45
Abtastsystem	11	Fenstereffekt	90
Abtasttheorem	12	Fensterung	45
Abtastung	18	FFT-Algorithmus	70
Aliaseffekt	36	Filter	22
analyzing wavelets	48	FIR-Filter	51
— B—		Fourier-Kotransformierte	6
Bandbegrenzung	12	Fourierreihe	33
Bildwiederholfrequenz	94	Fouriertransformation	6
Biorthogonalität	51	Fouriertransformierte	6, 7
Bitmap-Format	60	Frequenz	31
Bit-Reverse-Shuffling	73	Frequenzbereich	6, 16, 31, 35
BMP	Siehe Bitmap-Format	FWT-Algorithmus	72
— C—		— G—	
C++	59	Gewichtsfunktion	22
CCITT	62	Gibbsches Phänomen	83, 96
cut-off-frequency	94	GIF	Siehe Graphics Interchange Format
— D—		Gleichanteil	26, 39
Datenkompression	3	Grafikformat	
Datenreduktion	3, 17	rasterbasiert	59
Daubechies	5, 72	vektorbasiert	59
DCT	34, 63	Graphics Interchange Format	61
Delta-Distributionen	9	— H—	
Device Independent Bitmap	67	Heisenbergschen Unschärferelation	46
Dezimationsfilter	51	Hochpaßfilter	25, 26, 39
DFT	13	Huffman-Codierung	58, 60, 63
DIB	67	— I—	
Dilatation	48	Idealer Hochpaß	25, 26, 39
Diskrete Cosinustransformation	34, 63, 71	Idealer Tiefpaß	24, 38
Diskrete Faltung	51	Identitätssatz	7
Diskrete Fouriertransformation	71	IIR-Filter	51
Diskrete Fouriertransformierte	13	Informationsgehalt	8
Diskrete Sinustransformation	34	in-place	67
Diskrete Wavelettransformation	47, 48	Integral-Wavelettransformation	49
Diskrete zweidimensionale		Inverse diskrete Cosinustransformation	36
Fouriertransformierte	27		
Divide-And-Conquer	70		
DWT	48		

— J—

Joint Photographic Experts Group Format	62
JPEG34 Siehe Joint Photographic Experts Group Format	
JPEG-Verfahren	98

— K—

Koeffizienten	35
Komplexität	71, 72, 73
Kompressionsverfahren	3

— L—

leakage effect	90
Leckeffekt	90
Lempel-Ziv-Welch-Algorithmus	61
Lena	76
LZW Siehe Lempel-Ziv-Welch-Algorithmus	

— M—

Mallat	72
MFC Siehe Microsoft Foundation Classes	
Microsoft Foundation Classes	59
mother wavelets	48
MRA	50
Multi-Dokument-Applikation	74
multiresolution analysis	50
Multiskalenanalyse	50
Mutterfunktion	48
Mutterwavelets	48

— O—

Oberwellen	20
Online-Hilfe	75
Ortsfrequenz	31, 94

— P—

PackBits	62
Parsevalsche Gleichung	7
PCX Siehe ZSoft Paintbrush Format	
Pixel	60
Plane	60
Positiver Träger	48

— Q—

Quadraturfilter	51
Quantisierung	57
Quantisierungstableau	58

— R—

Rechteckfenster	48
Rechteckfunktion	8
Reduktionsabbildung	
Charakteristische Funktion	77
Reduktionsalgorithmus	57
Reduktionsrate	77, 78
Riemann-Lebesgue-Theorem	19
RLE-Algorithmus	60, 61

— S—

Skala	49
Spektrum	6
Spiegeleffekt	23, 36
Spiegelungseffekt	18
Sprungfolge	17
Systemübertragungsverhalten	23

— T—

Tag Image File Format	62
Tensorierte Filterungen	30
Tensorierung	28
Tiefpaßcharakteristik	94
Tiefpaßfilter	24, 38
TIFF Siehe Tag Image File Format	
Translation	48

— U—

Umcodierung	60
Unschärferelation	46

— V—

Verallgemeinerte Funktionen	9
-----------------------------	---

— W—

Wavelet	49
Wavelet-Analyse	47
Waveletbasis	48
Waveletraum	50

Wavelets	47	Zeitdiskrete Fourierreihe	11
Wavelettieffpaßfilter	55	Zeitfenster	45
Wavelettransformation	49	ZSoft Paintbrush Format	61
Wavelet-Zerlegung	50	Zweidimensionale diskrete Cosinustransformation	40
World-Wide-Web	59, 96	Zweidimensionale diskrete Wavelettransformation	53
— Z —			
Zeitbereich	6, 15		