

Noise Cancellation Verfahren unter Verwendung von Wavelets

Projektarbeit

durchgeführt von

Henning Petersen

Institut für elektronische Musik
der Universität für Musik und darstellende Kunst Graz

Leiter: Prof. Dr. Höldrich

Begutachter: Prof. Dr. Robert Höldrich

Betreuer: Dipl-Ing. Markus Noisternig

Graz, im April 2005

Kurzbeschreibung

Mit Hilfe von Wavelets soll ein Noise Cancellation Verfahren entwickelt werden. Das zu entstörende Signal wird mittels einer *Wavelet Packet Transformation* in die Waveletdomäne transformiert. Es soll hier eine für die Entrauschung geeignete Basis für das Signal gefunden werden. Zur Unterdrückung des Störsignals werden simple Thresholding - Algorithmen in der Waveletdomäne verwendet. Um semistationären Störsignalen gerecht zu werden, müssen in den Sprachpausen die Störgeräuschpegel geschätzt und die Thresholds dementsprechend adaptiert werden. Hierzu ist eine Voice Activity Detection, unter Verwendung eines geeigneten Noise Estimators, notwendig. Die Programmierung erfolgt unter MATLAB.

Abstract

Using wavelets, a noise cancellation technique has to be developed. The signal which is to be denoised, is transformed into the wavelet domain, using the *wavelet packet transform*. An appropriate base tree for best representing the signal has to be found. Simple Thresholding algorithms are used to suppress the interfering signal in wavelet domain. Because of dealing with semi stationary interference signals, setting the thresholds has to work adaptive using *Voice Activity Detection* (VAD). Therefore, the noise / interference level has to be estimated during non-speech periods. The algorithms are implemented in MATLAB.

Inhaltsverzeichnis

1	Einleitung	5
2	Wavelet Grundlagen	6
2.1	Mutter-Wavelet und Wavelet-Familien	6
2.2	Anforderungen an ein Mutter-Wavelet	7
3	Wavelet-Transformationen	9
3.1	Die kontinuierliche Wavelet-Transformation (CWT)	9
3.1.1	Interpretation der Wavelet-Koeffizienten	10
3.1.2	Das kontinuierliche an der kontinuierlichen Wavelet-Transformation	11
3.2	Die diskrete Wavelet-Transformation (DWT)	11
3.3	Mehrfachauflösungen	12
3.3.1	Die Haar Wavelet-Transformation	12
3.3.2	Die schnelle Wavelet-Transformation (FWT)	15
3.3.3	Lokalität	16
3.3.4	Verschwinde Momente	16
3.4	Stationary Wavelet-Transformation (Cicle-Spinning)	17
3.5	Wavelet Pakete	18
4	Noise cancellation durch Wavelet-Koeffizienten thresholding	19
4.1	Wahl des Wavelets	20
4.2	Threshold Methoden	21
4.2.1	Hard-Thresholding	21
4.2.2	Soft-Thresholding	22
4.2.3	Supersoft Thresholding	22
5	Der Denoising Algorithmus	24
5.1	VAD	25
5.2	Noise Estimator	28
5.3	Wahl der Basis des Wavelet Pakets	29

6	Beispiele	33
----------	------------------	-----------

	Literaturverzeichnis	34
--	-----------------------------	-----------

1 Einleitung

In dieser Arbeit werden zunächst grundlegende Dinge über Wavelets erläutert. Begriffe wie Mutter-Wavelet, Wavelet-Familie, Skalierungsfunktion u.a. werden erklärt. Im dritten Kapitel werden verschiedene Wavelet-Transformationen, wie die kontinuierliche, die diskrete, die Haar, die schnelle und die Stationary Wavelet- Transformation vorgestellt. Abschließend wird der Begriff des Wavelet Pakets erläutert. Das vierte Kapitel befasst sich mit verschiedenen Threshold-Methoden, die in einem Noise Cancellation Verfahren eingesetzt werden können. Im speziellen wird die Supersoft-Methode vorgestellt, die auch in der Implementation, die in MATLAB realisiert wurde, Anwendung findet. Der verwendete Denoising Algorithmus wird Kapitel fünf behandelt. Hier wird auch die verwendete Voice Activity Detection und der Noise Estimator ausführlich erklärt. Abschließend wird erläutert wie eine günstige Basis für ein Wavelet Paket gefunden werden kann. Einige Beispiele befinden sich in Kapitel sechs. Ein kurzer Ausblick wird in Kapitel sieben gegeben.

2 Wavelet Grundlagen

2.1 Mutter-Wavelet und Wavelet-Familien

Wavelets („kleine Wellen“) sind mathematische Funktionen, welche, ähnlich den Sinus- und Kosinus Funktionen der Fourier-Transformation, dazu benutzt werden, eine gegebene Funktion (Signal) auf die in ihr enthaltenen Frequenzbestandteile hin zu untersuchen. Abbildung 1 zeigt einige der häufig verwendeten „Mutter-Wavelets“

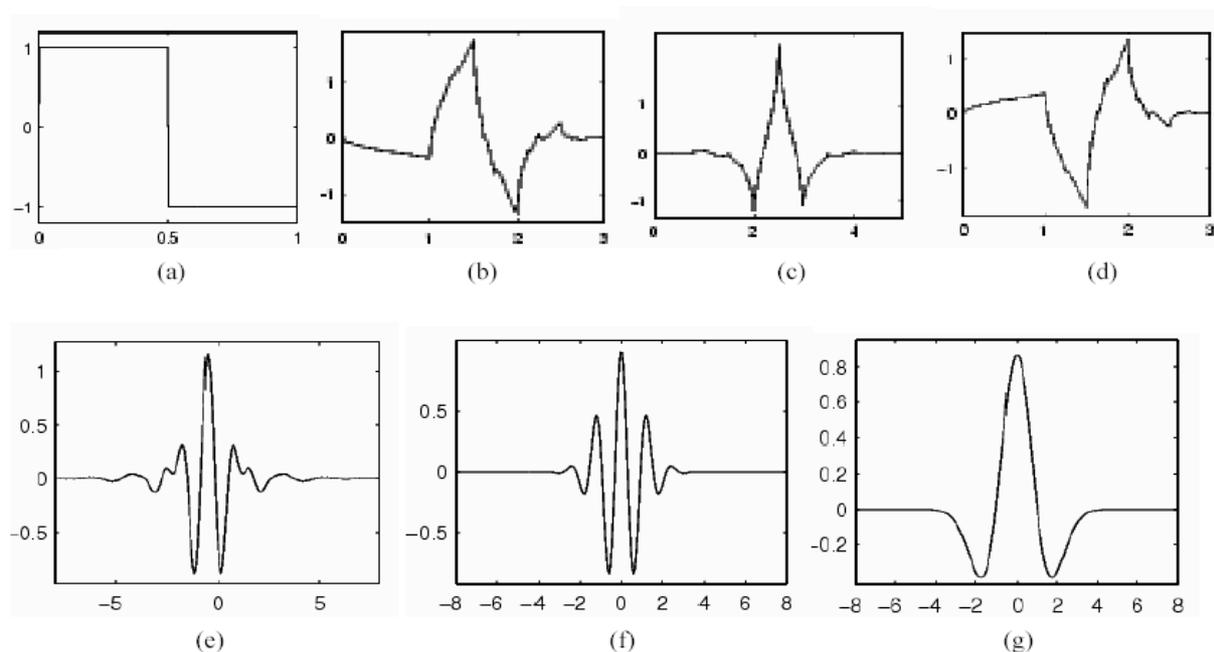


Abbildung 1: einige Mutter-Wavelets a) Haar b) Daubechies4 c) Coiflet d) Symlet2 e) Meyer f) Morlet g) Mexican hat

Zu diesem Zweck können Wavelets gestreckt bzw. gestaucht (skaliert) werden oder an eine bestimmte Stelle des Signals verschoben werden. Ausgangspunkt einer Wavelet-Transformation ist ein Basis- bzw. Mutter-Wavelet. (s. Abb.1) In der Transformation selbst kommen die entsprechend skalierten und örtlich verschobenen Varianten des Basis-Wavelets zum Einsatz. Diese Varianten werden auch allgemein als Waveletfamilie bezeichnet.

Wavelets sind also eine Erweiterung der Fourier-Analyse. Die Vorgehensweise bei der Wavelet-Analyse ist aber grundsätzlich die gleiche wie die, bei der Fourier-Analyse. Bei der Fourier-Analyse wird das zu analysierende Signal mit unterschiedlich gewichteten und phasenverschobenen Sinus- und Kosinusschwingungen verglichen. Dies geschieht durch Multiplikation der Abtastfunktion (also dem Sinus oder Kosinus) mit dem Signal und

anschließender Integration über das Produkt. Hieraus ergeben sich die Fourier-Koeffizienten. Je höher ein Koeffizient ist, desto höher ist die Ähnlichkeit dieser Grundschwingung mit dem zu analysierenden Signal.

Bei der Wavelet-Analyse wird das Signal nicht mit Sinus- oder Kosinusschwingungen verglichen sondern mit der Waveletfamilie; die Abtastfunktion ist hier also ein Wavelet. Die Wavelet-Koeffizienten ergeben sich in gleicher Weise wie die Fourier-Koeffizienten, also durch Multiplikation des Wavelets mit dem Signal und anschließender Integration über das Produkt. Auch hier wird der Wavelet-Koeffizient umso höher sein, je höher die Ähnlichkeit des gedehnten oder gestauchten Wavelets an der analysierten Stelle ist.

Abbildung 2 zeigt beispielhaft die „Mexican Hat“- Waveletfamilie.

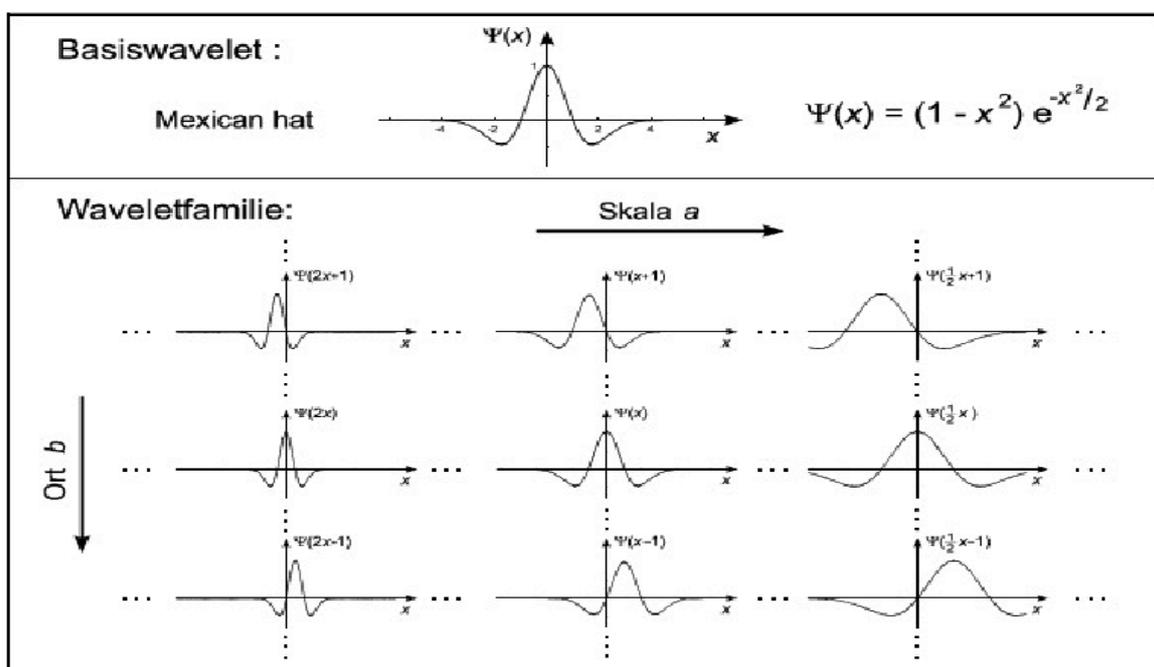


Abbildung 2: Die „Mexican Hat“- Wavelet-Familie aus [3]

2.2 Anforderungen an ein Mutter-Wavelet

Ein Mutter-Wavelet Ψ muss der Zulässigkeitsbedingung genügen:

$$\int_{-\infty}^{+\infty} \frac{|\hat{\Psi}(\omega)|^2}{|\omega|} d\omega < \infty \quad \text{aus [9]}$$

erfüllen, wobei $\hat{\Psi}(\omega)$ die Fourier-Transformierte von $\Psi(\omega)$ ist.

In den meisten Fällen ist diese Bedingung erfüllt, wenn, wie in [4] S.25f beschrieben, der Pol im Integranden durch

$$\hat{\Psi}(0) = 0 \quad \text{d.h.} \quad \int_{-\infty}^{+\infty} \Psi(t) dt = 0$$

kompensiert wird.

Das Integral über die Wavelet-Funktion muss also gleich Null sein, d.h. die Flächen, die oberhalb der x-Achse vom Wavelet eingeschlossen sind müssen genauso groß sein, wie die Flächen, die unterhalb der x-Achse vom Wavelet eingeschlossen sind. Funktionen deren Integral Null ergibt haben typischer Weise oszillierenden Charakter. Daher kommt auch der Name „Wavelet“, also „Wellchen“ oder „kleine Welle“.

Ein Wavelet $\Psi_{a,b}$ aus einer Wavelet-Familie kann beschrieben durch:

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

wobei a der Skalierungsfaktor (also der Grad der Streckung oder Stauchung) und b der Verschiebungsfaktor ist.

3 Wavelet-Transformationen

3.1 Die kontinuierliche Wavelet-Transformation (CWT)

Die kontinuierliche Wavelet-Transformation ist definiert als die Summe aller Produkte aus dem Signal und den unterschiedlich skalierten und verschobenen Wavelets über die gesamte Zeit.

$$C(\text{scale}, \text{position}) = \int_{-\infty}^{+\infty} f(t) \cdot \Psi(\text{scale}, \text{position}, t) dt$$

Das Ergebnis der kontinuierlichen Wavelet-Transformation sind also viele Wavelet-Koeffizienten, die eine Funktion von *scale* und *position* sind. Durch Multiplikation der Wavelet-Koeffizienten mit den zugehörigen verschobenen und skalierten Wavelets erhält man wieder das Originalsignal.

Die kontinuierliche Wavelet-Transformation lässt sich einfach in fünf Schritte unterteilen:

- 1) Man nimmt ein Wavelet und vergleicht es mit einem Abschnitt am Beginn des Signals
- 2) Man berechnet eine Zahl, *C*, die angibt wie stark das Wavelet mit diesem Signalabschnitt korreliert. Je höher *C* ist, desto höher ist die Ähnlichkeit. Nimmt man für die Signalenergie und die Waveletenergie den Wert eins an, so kann *C* als Korrelationskoeffizient interpretiert werden. Selbstverständlich hängt dieses Resultat von der Form des gewählten Wavelets ab.
- 3) Nun schiebt man das Wavelet nach rechts und wiederholt die Schritte 1 und 2 solange, bis man das ganze Signal abgedeckt hat.
- 4) Als nächste skaliert (dehnt) man das Wavelet und wiederholt die Schritte 1 – 3.
- 5) Schließlich wiederholt man die Schritte 1 – 4 für alle Skalierungen.

Nach dieser Prozedur hat man die Koeffizienten für alle Positionen und alle Skalierungen des Wavelets.

3.1.1 Interpretation der Wavelet-Koeffizienten

Man kann diese Koeffizienten graphisch darstellen. Die x-Achse in folgender Abbildung repräsentiert die Position des Wavelets entlang des Signals, also die Zeit. Die y-Achse stellt die Skalierung des Wavelets dar und die Farbe an jedem x,y-Punkt bildet den Betrag des jeweiligen Koeffizienten C ab.

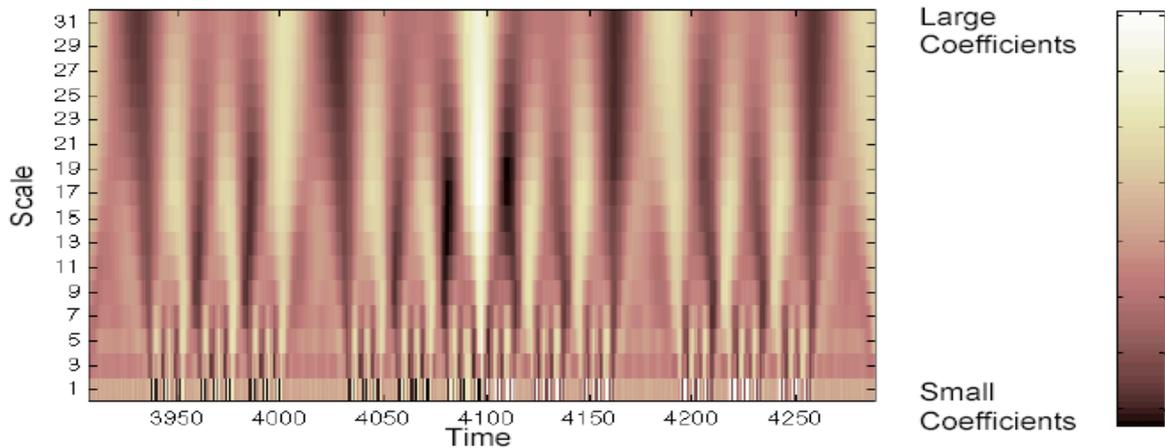


Abbildung 3: Wavelet-Koeffizienten

Die Skalierung des Wavelets reicht in obiger Abbildung von 1 – 31. Je größer das Wavelet skaliert ist, also je größer es gedehnt ist, desto größer ist der Abschnitt des Signals mit dem das Wavelet verglichen wird. Es werden also gröbere Signaleigenschaften gemessen, je weiter das Wavelet gedehnt wird. Abbildung 4 veranschaulicht dies.



Abbildung 4: unterschiedlich skalierte Wavelets im Vergleich zum Signal

Es besteht offenbar ein Zusammenhang zwischen der Skalierung des Wavelets und der Frequenz. Ein schmales Wavelet misst die schnell veränderlichen Eigenschaften des Signals, also die hohen Frequenzen, ein gedehntes, breites Wavelet misst die groben Eigenschaften des Signals, also die niedrigen Frequenzen.

3.1.2 Das kontinuierliche an der kontinuierlichen Wavelet-Transformation

Jedes natürliche Signal, das an einem Computer verarbeitet wird, muss diskretisiert werden. Das kontinuierliche an der kontinuierlichen Wavelet-Transformation ist die Möglichkeit auf jeder möglichen Skalierung und jeder möglichen Position des Wavelets gegenüber dem Signal zu agieren. Während der Berechnung wird das Wavelet also ganz langsam und mit allen Skalierungen über den zu analysierenden Bereich geschoben.

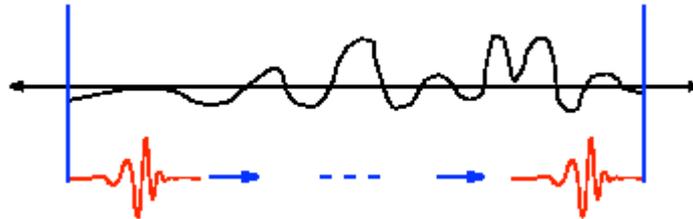


Abbildung 5: Das Wavelet wird langsam über den zu analysierenden Bereich geschoben

3.2 Die diskrete Wavelet-Transformation (DWT)

Die Berechnung der Wavelet-Koeffizienten an jeder möglichen Stelle und mit jeder möglichen Skalierung produziert eine riesige Datenmenge und erfordert einen sehr hohen Rechenaufwand. Daher kann man versuchen die Skalen- und Zeitparameter der kontinuierlichen Wavelet-Transformation einzuschränken und somit zur diskreten Wavelet-Transformation überzugehen. Die Berechnung der DWT erfolgt analog zu der der CWT, mit dem Unterschied, dass nun für die Skalen- und Zeitparameter a und b deutlich weniger diskrete Werte verwendet werden. Dabei gilt es, die verbleibenden Parameter a und b so geschickt zu wählen, dass einerseits möglichst wenig Redundanz (die CWT ist in hohem Maße redundant) entsteht und dass andererseits das Originalsignal noch vollständig beschrieben werden kann, also rekonstruierbar ist. Eine sehr gute Variante, die beide Bedingungen erfüllt, ist beim Strecken der Wavelets den (dyadischen) Faktor 2 zu verwenden. Es ergeben sich dann ausschließlich Wavelets der Form:

$$\Psi(2^k t + l) \quad \text{mit } k, l \in \mathbb{Z}$$

Abbildung 6 zeigt die Anwendung der DWT auf ein Signal. Das Wavelet wird zunächst entlang des Signals verschoben anschließend gestreckt, erneut entlang des Signals verschoben usw.

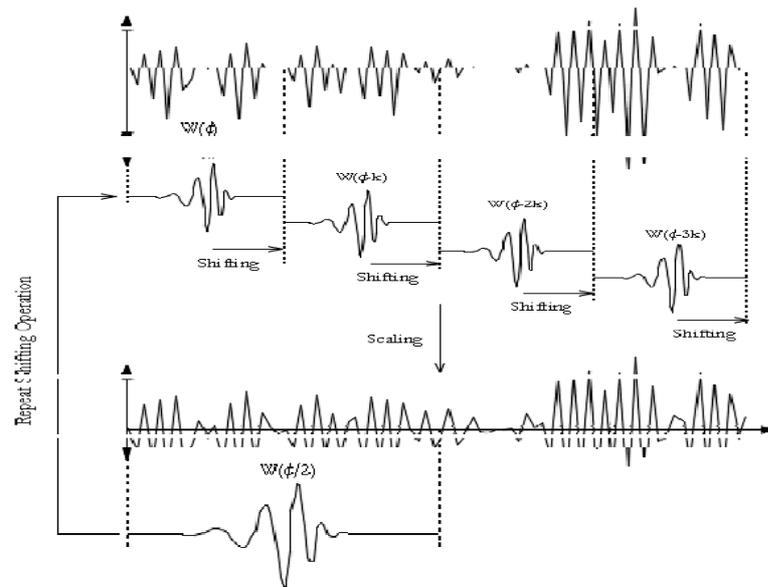


Abbildung 6: Verschieben und Strecken eines Wavelets bei der DWT aus [7]

3.3 Mehrfachauflösungen

3.3.1 Die Haar Wavelet-Transformation

Die schnelle Wavelet-Transformation (FWT) basiert auf einer etwas anderen Idee. Gegeben sei ein diskretes Signal f . Abbildung 7 oben zeigt wie ein solches Signal als „kontinuierliches Signal“ durch eine abschnittsweise konstante Funktion dargestellt werden kann.

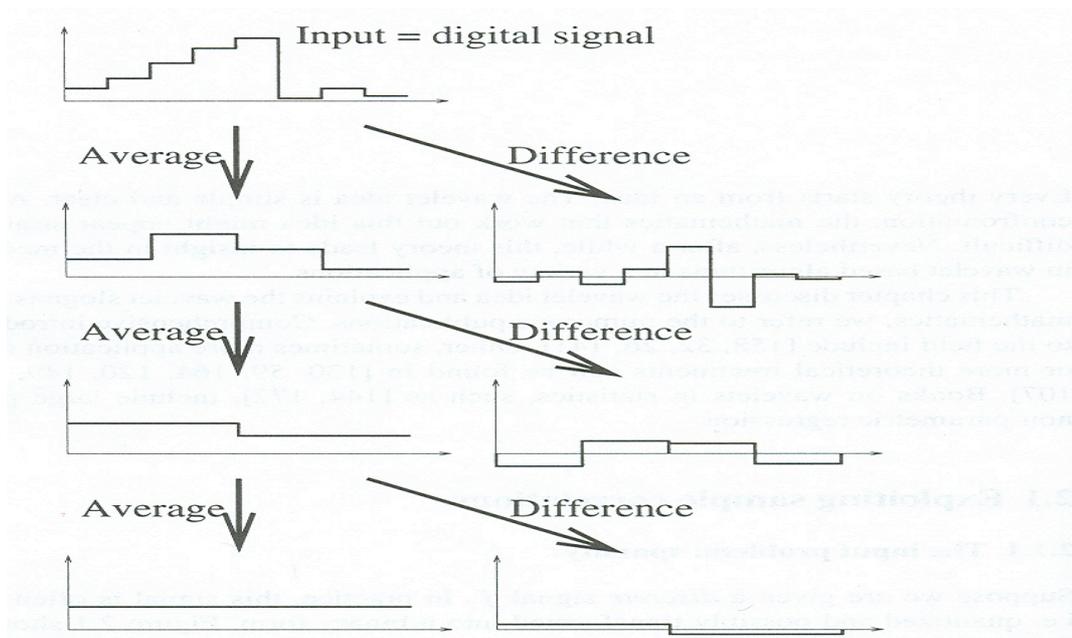


Abbildung 7: Die Haar-Wavelettransformation aus [4]

Normaler Weise korrelieren benachbarte Punkte stark miteinander. Nur an sehr wenigen Punkten sind große Sprünge zu erwarten. Alle diese Werte zu speichern wäre also eine unnötige Verschwendung von Speicherkapazität. Deswegen nimmt man zwei benachbarte Datenpunkte f_1 und f_2 und berechnet den Durchschnitts- und den Differenz-Koeffizienten.

$$a_1 = \frac{f_1 + f_2}{2} \quad d_1 = \frac{f_1 - f_2}{2}$$

In obiger Abbildung ist auf der linken Seite der Durchschnitt und auf der rechten Seite die Differenz zu sehen. Der Differenz-Koeffizient gibt an, wie viel kleiner der Wert eines Datenpunkts verglichen mit dem Durchschnitt ist. Durch Addition des linken und rechten Plots erhält man wieder das darüberliegende Eingangssignal. Das Durchschnittssignal ist eine „verschwommene“ oder „verwischte“ Version des Eingangssignals. Diese Durchschnittsbildung kann nun solange wiederholt werden, bis ein Gesamtdurchschnitt und die Unterschiede in unterschiedlichen Auflösungen übrigbleiben. Weil jeder Schritt umkehrbar ist, kann das Signal verlustlos rekonstruiert werden. Diese Transformation wird Haar-Transformation genannt. Man sieht in Abbildung 7, dass die meisten Differenz-Koeffizienten sehr klein sind. Die größten Koeffizienten treten an den Stellen auf, an denen das Eingangssignal die größten Sprünge macht. Deutlicher zu sehen ist dies an einem etwas realistischerem Beispiel, das Abbildung 8 veranschaulicht. Alle Koeffizienten sind in der Grafik hintereinander dargestellt. Die gestrichelten Linien zeigen die Grenzen zwischen den unterschiedlichen Skalierungen an. Nur einige wenige Koeffizienten heben sich deutlich hervor; sie zeigen die großen Sprünge bzw. die Singularitäten im Eingangssignal an. Diese Seltenheit, mit der Koeffizienten mit hohen Werten auftreten, ist eine Eigenschaft die allen Wavelet-Transformationen gemeinsam ist.

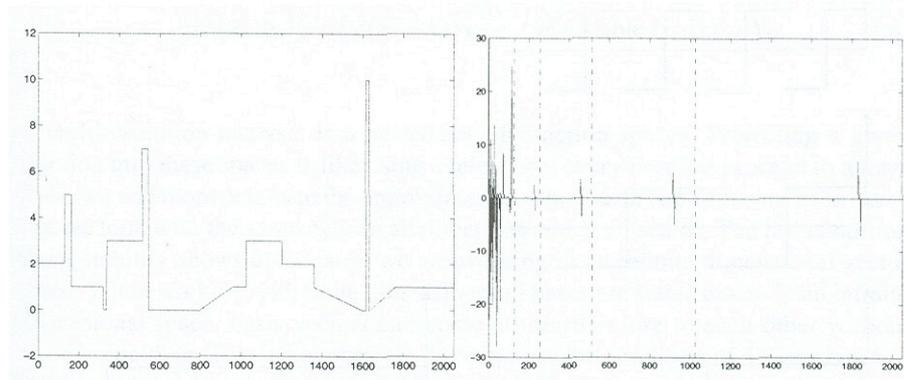


Abbildung 8: Testsignal (links) und Haar-Transformierte(rechts)

Der Eingangsvektor f (s.o.) kann als Koeffizientenvektor für eine Basis von charakteristischen Funktionen angesehen werden.

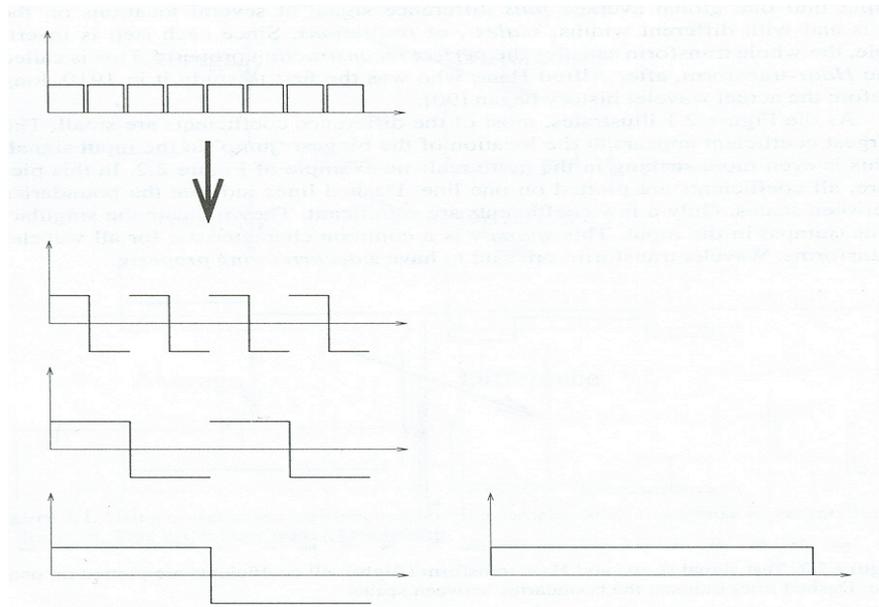


Abbildung 9: Basis-Funktion einer Waveletzerlegung

Man kann die kontinuierliche Darstellung $f(x)$ des diskreten Eingangssignals f als Linearkombinationen dieser Basis-Funktionen $\varphi_k(x)$ darstellen:

$$f(x) = \sum_k f_k \varphi_k(x)$$

Alle diese Funktionen sind Verschiebungen von einer „Vater-Funktion“, sie heißen „Skalierungs-Funktionen“. Die Unterschiede, die während des Algorithmus berechnet werden, können durch die Basis-Funktionen im unteren Teil von Abbildung 9 dargestellt werden. All diese Funktionen sind verschobene und gestreckte Versionen einer „Mutter-Funktion“.

Diese Zerlegung ist ein Beispiel für eine multiresolution analysis MRA („Mehrfachauflösungs-Analyse“). Eine MRA ist eine verschachteltes Set von

Funktionsräumen. Wenn man eine gegebene Funktion in diese Funktionsäume projiziert, ist das wie wenn man ein Fernglas benutzt. Jedes Mal wenn man eine feinere Auflösung wählt, fügt man der Approximation mehr Details hinzu.

3.3.2 Die schnelle Wavelet-Transformation (FWT)

Das Ziel der FWT ist es, ein Signal mittels einer Skalierungsfunktion in verschiedene Auflösungen zu approximieren. Die Skalierungsfunktion wird hierbei durch einen Tiefpass-Filter realisiert, d.h. dass Signal wird geglättet. Die durch Anwendung dieses Filters wegfallenden Details, werden durch geeignete Wavelets aus einer Wavelet-Familie beschrieben und als Wavelet-Koeffizienten gespeichert. Dies geschieht durch eine Hochpass-Filterung. Nach der Filterung wird das Signal um den Faktor 2 downgesampled. Danach wird das Signal in der nächst tieferen Auflösungsstufe bearbeitet. Als Eingangsgröße dient nun der dezimierte Tiefpassanteil des Signals. Dieser Prozess kann solange wiederholt werden bis das Signal kürzer als die Länge des Filters ist. Abbildung 10 zeigt einen Schritt der FWT-Analyse und einen Schritt der Synthese.

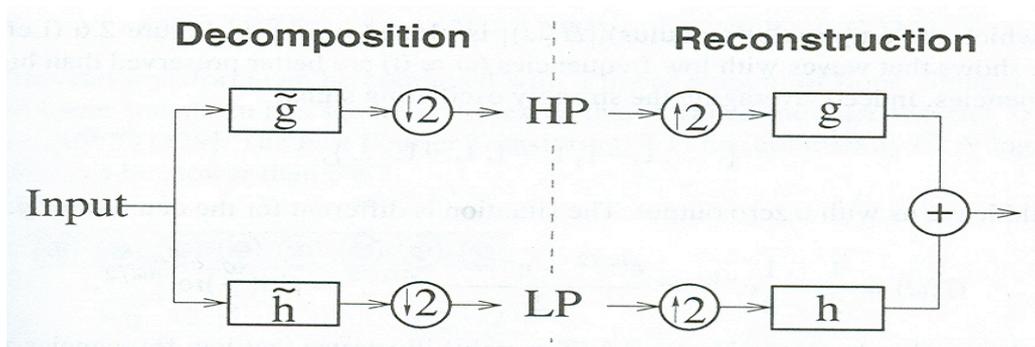


Abbildung 10: Ein Schritt einer FWT-Zerlegung und die zugehörige Rekonstruktion

Um die Synthese durchzuführen muss das Signal einem upsampling unterzogen werden und kann anschließend durch geeignete Hoch- bzw. Tiefpassfilterung zurückgewonnen werden.

Da das Signal nach jeder Filterung downgesampled wird, schrumpft die Anzahl der Koeffizienten mit der Skalierung. Abbildung 11 zeigt das Schema einer FWT.

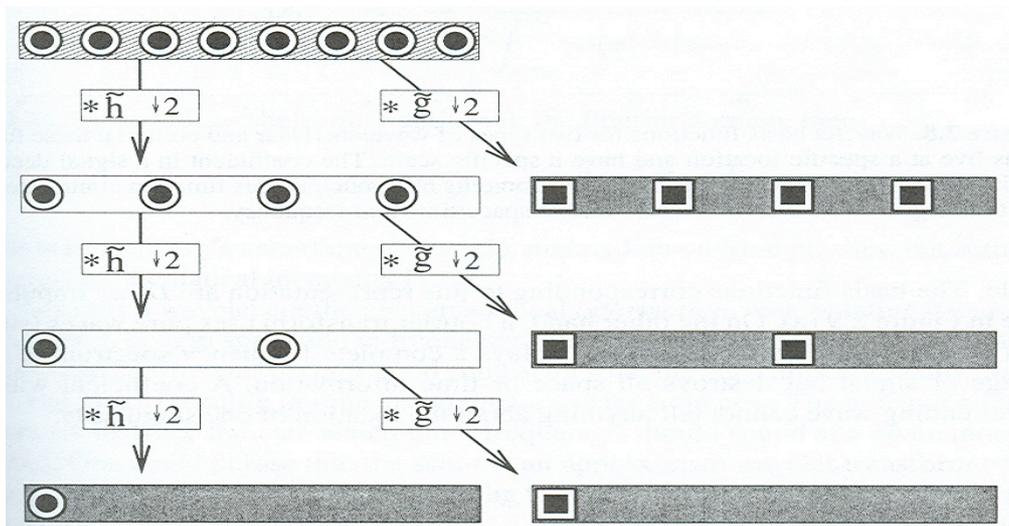


Abbildung 11: Schema einer FWT

3.3.3 Lokalität

Bei einer Fourier-Transformation werden reine Sinus- und Kosinusschwingungen als Basisfunktionen verwendet. Es kann anhand dieser Transformation das komplette Frequenzspektrum des Signals dargestellt werden, gleichzeitig wird aber jegliche Zeitinformation zerstört; d.h. es ist völlig unklar wann eine bestimmte Frequenz im Signal auftritt. Natürlich geht bei einer Blockverarbeitung nur die Zeitinformation innerhalb eines Blocks verloren, macht man aber die Wavelettransformation blockweise, so geht auch die Zeitinformation innerhalb eines Blockes nicht völlig verloren. Wavelet-Basisfunktionen sind „kleine Wellen“, sie haben also anders als die Analyse-Funktionen der Fourier-Transformation nur in einem kleinem Intervall einen Wert, der von Null verschieden ist. Der Wert eines Wavelet-Koeffizienten sagt aus, wie sehr ein Wavelet zu einem bestimmten Zeitpunkt mit einer bestimmten Skalierung mit dem Signal übereinstimmt, er enthält also Information über Zeit und Frequenz. Diesen Vorteil der Wavelet-Transformation gilt es gegenüber der Fourier-Transformation auszunützen. Natürlich kann wegen der Ort/Zeit und Frequenz Unschärfe keine Basisfunktion gleichzeitig die exakte Information über Ort/Zeit und Frequenz geben.

3.3.4. Verschwinde Momente

Wie bereits gesagt, ist die „Seltenheit“, mit der Wavelet Koeffizienten mit hohen Werten auftreten, eine Eigenschaft die allen Wavelet-Transformationen gemeinsam ist. Diese „Seltenheit“ wird durch verschwindende Momente verstärkt. Um solch eine seltene „Darstellung“ zu bekommen, versucht man sehr kleine Koeffizienten in den Bereichen zwischen Singularitäten zu bekommen. Da es in diesen Bereichen, in denen sich das Signal nur langsam verändert, sehr gut möglich ist das Signal durch ein Polynom anzunähern, ist man an Polynomen interessiert, die einige Koeffizienten mit dem Wert Null aufweisen. Wenn jedes Monom bis zum Grad $\tilde{p}-1$ die Bedingung:

$$\langle x^n, \tilde{\Psi} \rangle = 0, n = 0, \dots, \tilde{p}-1 \quad \text{aus [4]}$$

erfüllt, dann tragen die ersten \tilde{p} Terme in einer Taylor-Näherung nicht zu den Wavelet Koeffizienten bei.

Das größte \tilde{p} , das obige Bedingung erfüllt, nennt man die Anzahl der verschwindenden Momente.

3.4 Stationary Wavelet-Transformation (Cicle-Spinning)

Die DWT hat einen großen Nachteil. Sie ist keine Zeit-invariante Transformation. Das bedeutet, die DWT einer verschobenen Version des Signals X , ist nicht zwangsläufig die verschobene Version der DWT des Signals X . Beim Entrauschen treten einige unerwünschte Artefakte an Singularitäten, also an Sprüngen des Signals auf. Diese Artefakte heißen pseudo-Gibbs Phänomene, Über- und Unterschwingen an den Sprungstellen. Das Ausmaß, mit dem dieses Phänomen auftritt, hängt stark mit dem Ort zusammen, an dem sich der Sprung befindet. Verwendet man beispielsweise das Haar-Wavelet, so führt eine Unstetigkeit am Ort $n/2$ zu nahezu gar keinen pseudo-Gibbs Schwingungen. Befindet sich die Unstetigkeit jedoch am Ort $n/3$, so wird man erhebliche pseudo-Gibbs Schwingungen feststellen. Das einfachste wäre, das Signal immer so zu verschieben, dass die Unstetigkeiten an einer Stelle zu liegen kommen, an denen sie keine Probleme verursachen. Da ein Signal aber mehrere Unstetigkeiten besitzen kann, und die günstigste Verschiebung für eine Unstetigkeit gleichzeitig die schlechteste für eine andere Unstetigkeit sein kann, muss man auf andere Mittel zurückgreifen:

T sei die Analysetransformation. Anstatt von T berechnet man die zeitlich verschobene Version von T :

$$\tilde{T}(x; S_h) = S_{-h}(T(S_h(x)))$$

Das Problem ist nun, dass man nicht weiß für welchen Shift h man das beste Ergebnis erzielt.

Dieses Problem kann behoben werden, indem über verschiedene, verschobene DWTs gemittelt wird. Dieser Vorgang heißt „cicle spinning“ und wird in [6] näher beschrieben.

$$\bar{T}(x; (S_h)_{h \in H}) = Ave_{h \in H} S_{-h}(T(S_h(x)))$$

Man berechnet also verschiedene verschobene DWTs, denoised, verschiebt zurück und mittelt über alle Ergebnisse. Das gemittelte Ergebnis weist zumindest keine sehr hohen pseudo-Gibbs Schwingungen auf.

$$Average[Shift - DeNoise - Unshift]$$

mit:

T : Analyse-Transformation

$$x_t : 0 \leq t \leq n$$

$$S_h : \text{zirkulärer Shift um } h, (S_h x)_t = x_{(t+h) \bmod n}$$

H : Anzahl der Shifts

Führt man diese Durchschnittsbildung über alle n zirkulären Shift durch, so ist dies identisch mit der „stationary“ oder „undecimated“ Wavelet-Transformation.

3.5 Wavelet Pakete

Bei einer Wavelet Zerlegung wie unter Punkt 3.3.2 wird jeweils der dezimierte Tiefpassanteil weiterverarbeitet, d.h. er wird wiederum Hoch- bzw. Tiefpassgefiltert. Dies kann auch mit dem Hochpassanteil geschehen. Nach jedem Schritt kann entschieden werden ob mit der Zerlegung (Näherung) an beiden, einem oder keinem der beiden Filterausgängen fortgesetzt werden soll. Die Zerlegung fortzusetzen bedeutet, eine bessere Frequenzauflösung zu bekommen, gleichzeitig aber Zeitinformation einzubüßen. Es gibt also unterschiedliche Zerlegungsmöglichkeiten für ein und dasselbe Eingangssignal. Ein möglicher Vertreter aus dieser „Familie“ heißt „Wavelet Paket“. Der Vorteil der Zerlegung in ein Wavelet Paket ist, dass man sich den Baum so aussuchen kann, dass er genau auf das zu analysierende Signal passt, d.h. man zerlegt die Zeit-Frequenz Ebene so, dass sie optimal an das Signal angepasst ist. Abbildung 12 zeigt eine komplette Wavelet-Paket-Zerlegung, Abbildung 13 eine spezielles Wavelet-Paket mit der zugehörigen Zeit-Frequenz-Zerlegung.

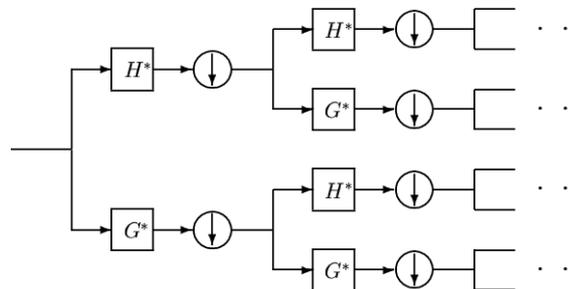


Abbildung 12: eine komplette Wavelet-Paket-Zerlegung

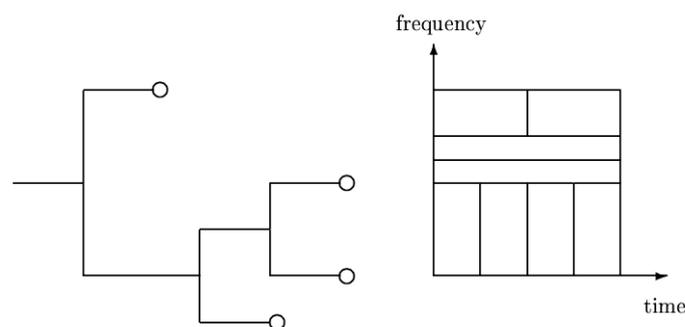


Abbildung 13: eine spezielle Wavelet-Paket-Zerlegung mit der zugehörigen Zeit-Frequenz-Zerlegung

Wenn man die Näherungen so lang wie möglich durchführt (Abb. 13), erhält man eine Zerlegung, die der Zeit-Frequenz-Auflösung der Fourier-Zerlegung sehr ähnlich ist.

4 Noise cancellation durch Wavelet-Koeffizienten thresholding

Geräuschreduktion durch Nullsetzen oder Verringern von Wavelet-Koeffizienten nennt man Thresholding. Bei vielen Signalen, wie zum Beispiel Sprache, ist die Energie in einigen wenigen Wavelet-Koeffizienten konzentriert. Bei Rauschen ist die Energie über alle Wavelet-Koeffizienten verteilt; alle Koeffizienten haben hier relativ kleine Werte. Ist also einem Nutzsignal Rauschen überlagert, so werden die Koeffizienten, die Energie des Nutzsymbols gespeichert haben, deutlich größer sein als jene Koeffizienten, die nur Energie des Rauschens gespeichert haben; vorausgesetzt der SNR ist groß genug.

Die meisten Geräuschreduktionsalgorithmen gehen von folgendem additiven Model aus:

$$y = x + n$$

x ist hierbei ein endlicher Eingangsvektor, der von einem zero-mean, weißen gaußverteilten Rauschen mit der Varianz σ^2 gestört wird. Das Ziel ist es, aus dem beobachteten Signal y , das ungestörte Signal x zurückzugewinnen.

Wenn W eine Wavelet-Transformations-Matrix darstellt, kann obige Gleichung von der Zeitdomäne in die Waveletdomäne transformiert werden:

$$Y = X + N$$

mit

$$Y = W \cdot y \quad X = W \cdot x \quad N = W \cdot n$$

Wenn $X_{\text{est}} = E\{X\}$ eine Schätzung des ungestörten Signals X ist, die aus dem beobachteten Signal Y in der Waveletdomäne hervorgeht, dann kann das entrauschte Signal nach [5] folgendermaßen geschätzt werden:

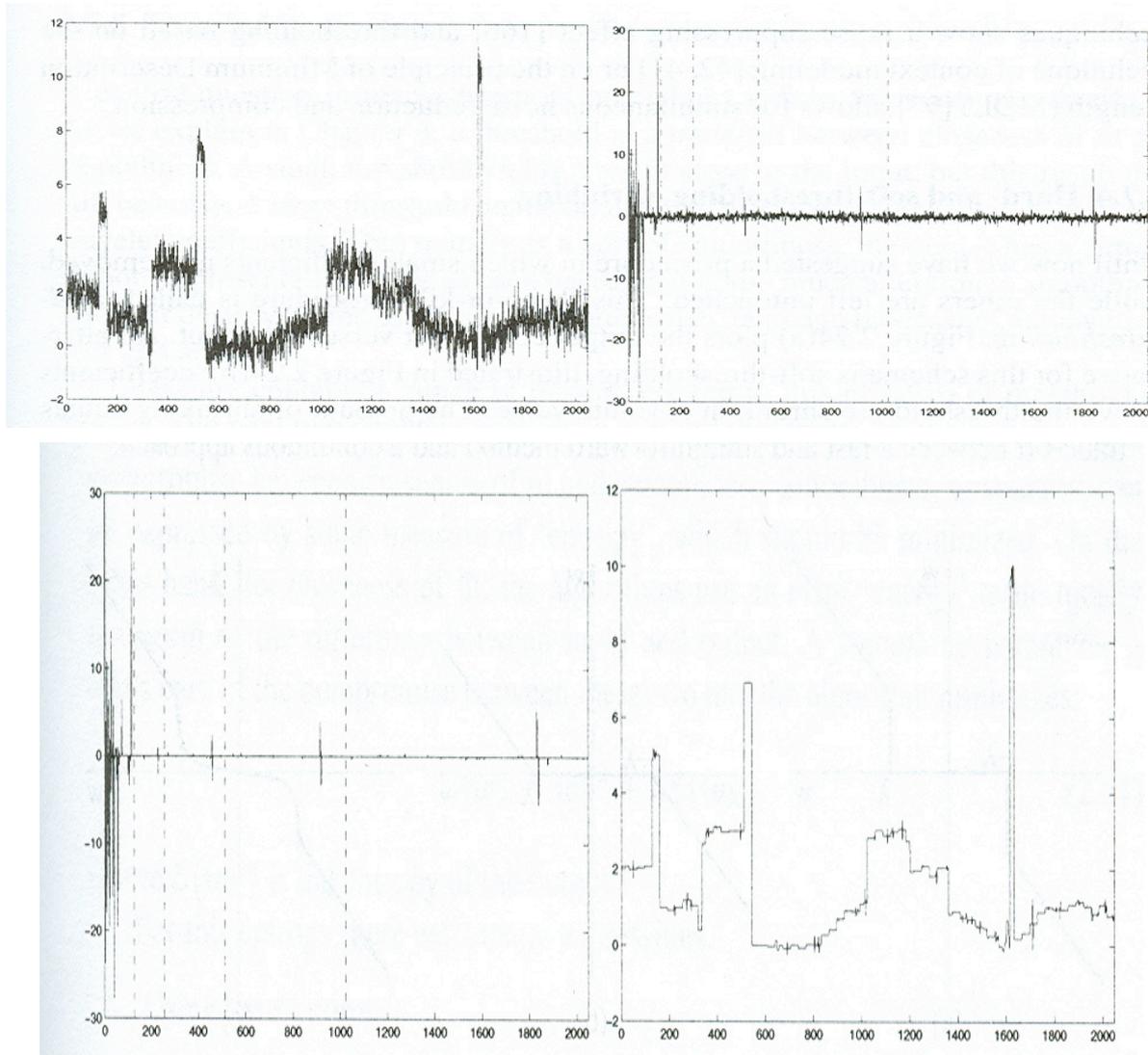
$$x = W^{-1} \cdot E\{X\} = W^{-1} Y_{\text{thr}}$$

Y_{thr} stellt hierbei die durch thresholding gewonnen Wavelet-Koeffizienten dar.

Abbildung 14 oben zeigt, dass das Rauschen auf alle Koeffizienten verteilt ist und die Singularitäten (Sprünge) des Signals anhand der größeren Wavelet-Koeffizienten gut unterscheidbar sind.

Indem man beispielsweise die kleinsten Koeffizienten, in diesem Fall alle, die eine Amplitude haben, die kleiner als 1 ist, Null setzt und danach eine Rücktransformation vornimmt, erhält man ein Signal, das im wesentlichen die gleichen Eigenschaften wie das Eingangssignal hat, aber weniger verrauscht ist. Das Ergebnis einer solchen Prozedur ist in Abbildung 14 ersichtlich. Das gewonnene Ausgangssignal in Abbildung 14 rechts unten enthält deutlich

weniger Rauschanteil als das Ursprungssignal in Abbildung 14 links oben. Allerdings stellen die dabei auftretenden akustischen Artefakte eine erheblich Minderung der Signalqualität dar.



**Abbildung 14 oben: Testsignal mit weißem Rauschen (links) und deren Haar-Transformierte (rechts)
unten: Wavelet-Koeffizienten nach soft-thresholding mit threshold $\lambda=1$ (links) und Rekonstruktion des
Signals (rechts)**

Um die Signalqualität des entstörten Signals zu verbessern verwendet man verschiedene, verbesserte Threshold Methoden. (s.4.2)

4.1 Wahl des Wavelets

Bei der Wahl des richtigen Wavelets für die Zerlegung des Signals ist es wichtig, dass das Wavelet gut zum Signal passt, das entrauscht werden soll. Da man beim thresholding die seltenen, großen, zum Signal gehörigen Koeffizienten behalten will, und die kleinen zum Rauschen gehörigen Koeffizienten schrumpfen oder weglassen will, wird dies um so besser

funktionieren, je ähnlicher das Wavelet in seiner Art dem Signal kommt. Hätte man es beispielsweise mit einem Signal zu tun, welches große und viele Unstetigkeiten (Sprünge) beinhaltet, so wäre das Haar-Wavelet wohl das geeignetste. Für ein Sprachsignal schlägt H. Storm in [1] das weich oszillierende Vaidyanathan-Wavelet vor. (s. Abb. 15)

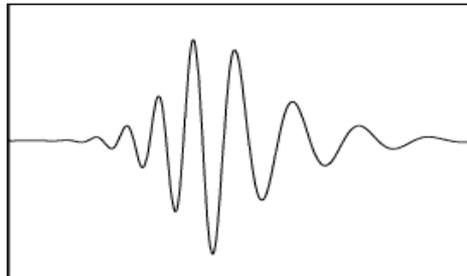


Abbildung 15: Vaidyanathan's Wavelet

Für das Rauschen ist die Wahl des Wavelets egal, da beispielsweise weißes gaußsches Rauschen in jeder orthogonalen Basis, also auch in einer Wavelet-Basis wieder weißes Rauschen sein wird. Die Amplitude der Koeffizienten wird sich also nicht ändern. Leider war die genaue mathematische Beschreibung des Vaidyanathan-Wavelets nicht zugänglich und es musste für diese Arbeit auf die in Matlab existierenden Wavelets zurückgegriffen werden.

4.2 Threshold Methoden

4.2.1 Hard-Thresholding

Im letzten Abschnitt wurde für die Thresholding-Methode vorgeschlagen, alle Koeffizienten, die unter einem bestimmten Wert liegen, Null zu setzen, und die übrigen Koeffizienten beizubehalten. Diese „keep or kill“ Methode nennt man Hard-Thresholding. In Abbildung 16 a) ist diese Methode graphisch veranschaulicht.

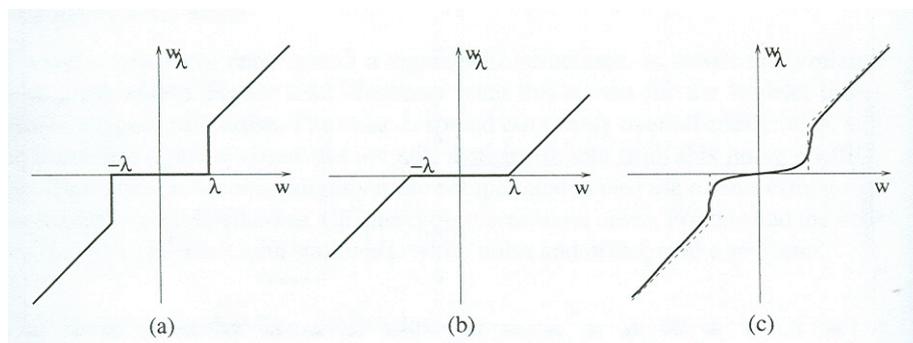


Abbildung 16: a) hard-thresholding b) soft-thresholding c) eine verfeinerte thresholding-Methode

Donoho schlägt in [2] folgende Formel für die Berechnung des Thresholds vor:

$$T = \sigma \sqrt{2 \cdot \log(N)},$$

wobei σ die Standardabweichung und N die Länge des analysierten Signalabschnitts ist. Die Koeffizienten ergeben sich somit zu:

$$THR_H(Y, T) = \begin{cases} Y, & |Y| > T \\ 0, & |Y| < T \end{cases}$$

4.2.2 Soft-Thresholding

Eine andere Thresholding-Methode ist in Abbildung 17 b) zu sehen. Hier werden zusätzlich diejenigen Koeffizienten, die oberhalb des Thresholds liegen geschrumpft. Der Betrag, um den jeder dieser Koeffizienten geschrumpft wird, ist gleich dem Threshold-Wert, so dass die Funktion in 17 b) kontinuierlich wird.

$$THR_S(Y, T) = \begin{cases} \text{sgn}(Y)(|Y| - T), & |Y| > T \\ 0, & |Y| < T \end{cases}$$

T berechnet sich wie unter 4.1.1.

Ein Kompromiss der beiden Methoden ist in Abbildung 15 c) zu sehen. Bei dieser Methode bleiben die großen Koeffizienten erhalten und der Übergang von verrauschten zu wichtigen Koeffizienten ist sanfter. Solche Methoden sind rechenintensiver und meist von mehr als einem Threshold abhängig.

4.2.3 Supersoft Thresholding

Man kann die Ziele beim Entfernen von Hintergrundrauschen von einem Sprachsignal wie folgt zusammenfassen:

- 1) Die Koeffizienten, die durch das Rauschen entstehen sollen entfernt werden.
- 2) Die Koeffizienten, die vom Sprachsignal stammen sollen möglichst unverändert bleiben, so dass die Sprachqualität nicht hörbar verschlechtert wird.
- 3) Das Hintergrundrauschen soll seinen Charakter nicht wesentlich ändern, da der Klang sonst unnatürlich würde. Es soll also ungefähr gleich klingen und nur in der Lautstärke abgeschwächt werden.

Diese Ziele sind zwar schwer vereinbar, es wird aber im folgenden gezeigt wie sie mit der Supersoft Thresholding Methode annähernd erreicht werden können.

Die Thresholding Vorschrift ist hier:

$$THR_s(Y, T) = \begin{cases} \text{sgn}(Y)(|Y| - T) \cdot (1 - \alpha), & |Y| > T \\ \alpha \cdot Y, & |Y| < T \end{cases}$$

mit, $0 \leq \alpha \leq 1$

Man sieht, dass Soft nur ein Spezialfall von Supersoft ist, nämlich wenn $\alpha = 0$ ist. Abbildung 17 zeigt wie die Koeffizienten mit der Supersoft Methode gethresholdet werden.

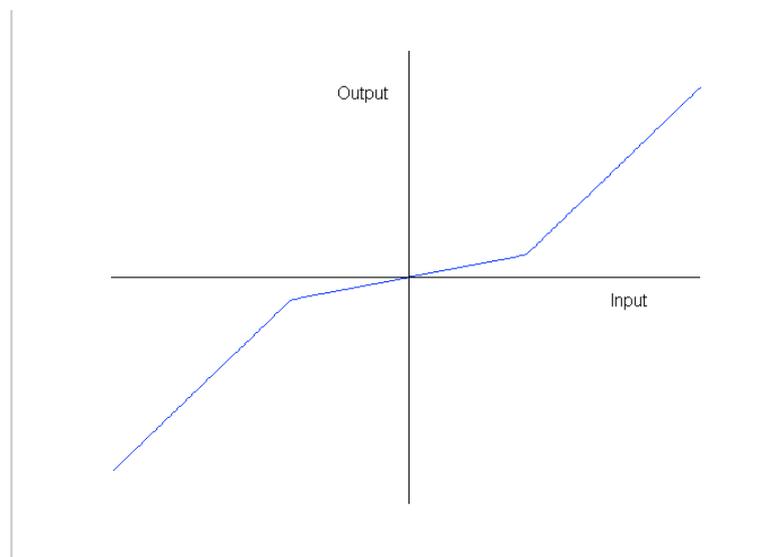


Abbildung 17: Verhalten des Supersoft Threshold Operators

Kleine Koeffizienten, also jene von denen man annimmt, dass sie vom Rauschen stammen, werden bei dieser Methode mit dem Faktor α abgeschwächt. Große Koeffizienten werden ebenfalls abgeschwächt, aber nicht in so hohem Maße wie bei der Soft Thresholding Methode. Die Charakteristik des Hintergrundrauschens wird weniger verändert als in den beiden anderen Fällen. Da die großen, der Sprache zugehörigen Koeffizienten weniger abgeschwächt werden, ist leicht einzusehen, dass die Sprache weniger verändert wird als im Soft Thresholding Fall.

5 Der Denoising Algorithmus

- 1) Die Verarbeitung findet blockweise statt, damit eventuell eine Echtzeimplementation möglich ist. Die einzelnen Blöcke bestehen aus 128 Samples.
- 2) Bestimmung, ob der Block aus reinem Rauschen besteht, oder ob Sprache in ihm enthalten ist. Dies geschieht mit einem VAD. (Voice-Activity-Detector)
- 3) Handelt es sich um einen reinen „Rauschblock“, so wird eine Schätzung des „noise levels“ für alle Knoten des Wavelet-Paket-Baumes durchgeführt.
- 4) Der Block wird einer kompletten Wavelet-Paket-Transformation unterworfen.
- 5) Auswählen einer passenden Basis aus dem Wavelet-Paket-Baum, d.h. es muss ein Wavelet-Paket gefunden werden, dass Sprache und Rauschen so gut wie möglich trennt. (Nur wenn Sprache in diesem Block vorhanden ist.)
- 6) Koeffizienten-Thresholding in der ausgewählten Basis mit der Supersoft-Threshold-Methode.
- 7) Rekonstruktion des entrauschten Blocks mit Hilfe der inversen Wavelet-Paket-Transformation.
- 8) Bearbeitung des nächsten Blocks.

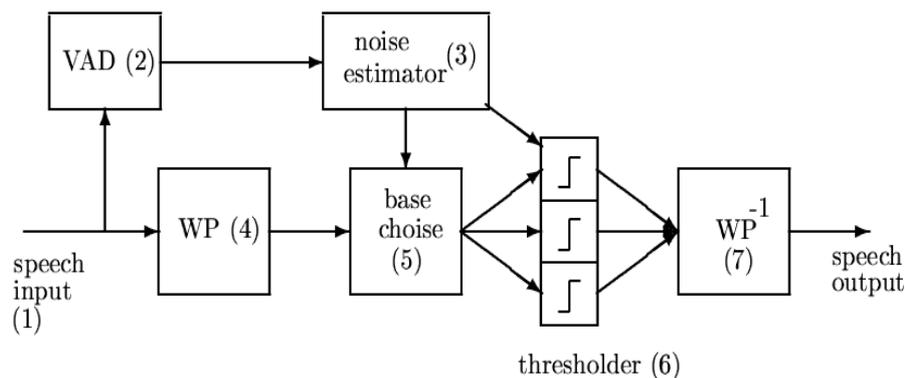


Abbildung 18: Schema des Denoising Algorithmus

5.1 VAD

Die Voice Activity Detection (VAD) wird für jeden Block berechnet. Sie entscheidet ob in einem Block nur Rauschen oder Sprache und Rauschen enthalten ist. Der Ausgang der VAD ist 1, wenn der Block Sprache und Rauschen enthält, 0, wenn er nur Rauschen enthält. Das Blockschaltbild ist Abbildung 19 zu entnehmen. Die hier verwendete VAD arbeitet mit zwei Thresholds T_{n1} und T_{n2} . Für jeden Block wird eine Energieschätzung durchgeführt. Dies wurde zunächst mit dem Teager Energy Operator (TEO) realisiert. Da dies aber zu schlechteren Ergebnissen führte wie eine einfache quadratische Energieschätzung, wurde dieser Ansatz wieder verworfen. Die Version mit der einfachen quadratischen Energieschätzung liefert Fehldetektionen eher im reinen „Noise-Bereich“, was besser ist als wenn die VAD die Sprache nicht „erkennen“ würde, also eher im „Sprach-Bereich“ die Fehldetektionen aufweisen würde, da bei einem nachgeschalteten Denoising sonst das Sprachspektrum abgezogen würde. Die akustischen Artefakte sind bei dieser Version also kleiner.

Ist die geschätzte Energie E_r in einem Block kleiner als der Threshold T_{n1} , so handelt es sich um einen reinen Rauschblock, die VAD gibt also 0 aus und die Thresholds T_{n1} und T_{n2} bzw. eine gemittelte Rauschenergie E_n wird upgedated. Eine Sprachanregung hat laut [8] eine maximale Dauer von 300 ms. Daher kann man davon ausgehen, dass es sich nicht um Sprache, sondern um eine Anhebung des Rauschlevels handelt, wenn die VAD über einen Zeitraum von mehr als 300 ms eine 1 liefert, also eigentlich Sprachblöcke detektiert. Es wird also ein Zähler c gestartet, wenn E_r den Threshold T_{n1} überschreitet. Dieser Zähler sorgt dafür, dass eine Anhebung des Rauschlevels erkannt wird, und zwar genau dann, wenn der Zählwert T_c von 300 ms überschritten wird. In diesem Fall erfolgt eine Anhebung des Rauschlevels und ein Update der Thresholds. Der alte Rauschlevel wird dabei mit einem Skalierungsfaktor (z.B. 2) multipliziert. Damit eine Adaption des Algorithmus möglich ist, wird zusätzlich ein kleiner Skalierungswert d (z.B. $d=0.0001$) addiert, falls der alte Wert von E_n Null war. Wird der Zählwert T_c nicht überschritten, so werden die alten Thresholdwerte beibehalten. Im folgenden wird überprüft ob die geschätzte Energie den Threshold T_{n2} überschreitet. Ist dies der Fall so handelt es sich um einen Sprachblock, die VAD gibt 1 aus, ist dies nicht der Fall handelt es sich um einen Rauschblock, die VAD gibt 0 aus und der Zähler wird resetiert. Danach wird der nächste Block überprüft und der Algorithmus beginnt wieder dem Vergleich zwischen E_r und T_{n1} .

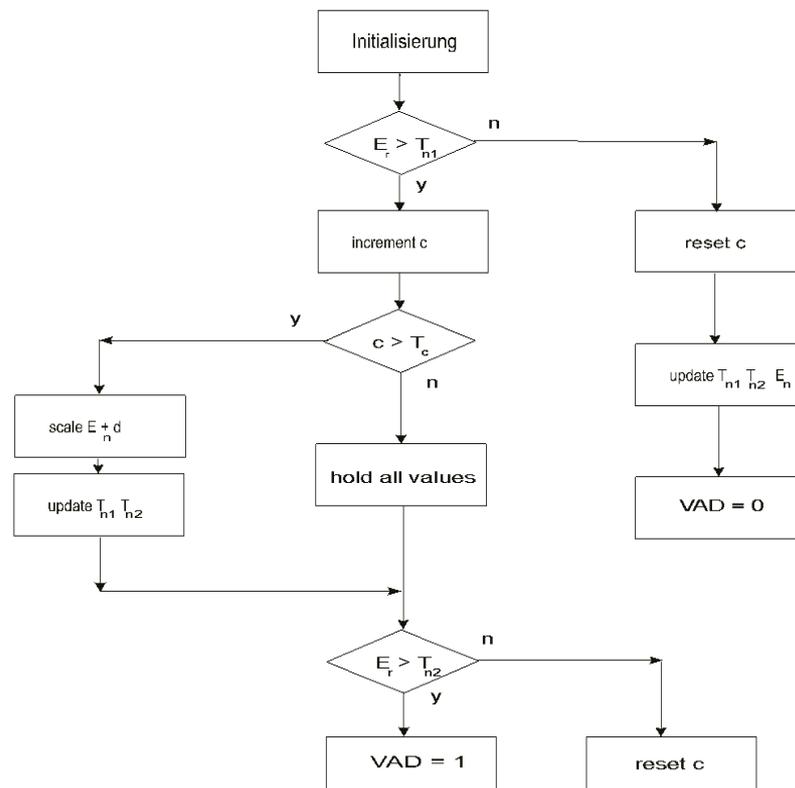


Abbildung 19: Blockschaltbild der VAD

In folgender Abbildung 20 ist ein Testlauf für die VAD zu sehen. Man sieht, dass nur Sprache detektiert wird, wenn die quadratische Energieschätzung über dem Threshold T_{n2} liegt. Liegt die Energie oberhalb des Thresholds T_{n1} so werden die Thresholds eingefroren, liegt sie unterhalb von T_{n1} , so werden die Thresholds ständig aktualisiert.

Ungefähr nach 20000 Samples hat der Ausgang der VAD länger als 300ms den Wert 1, der Zählwert T_c wird also erreicht. In diesem Moment springen die gemittelte Energie und die Thresholds nach oben (hier ungefähr um den Faktor 2), da nun eine Rauschlevelanhebung vorliegt. Die VAD hat danach wieder den Wert Null.

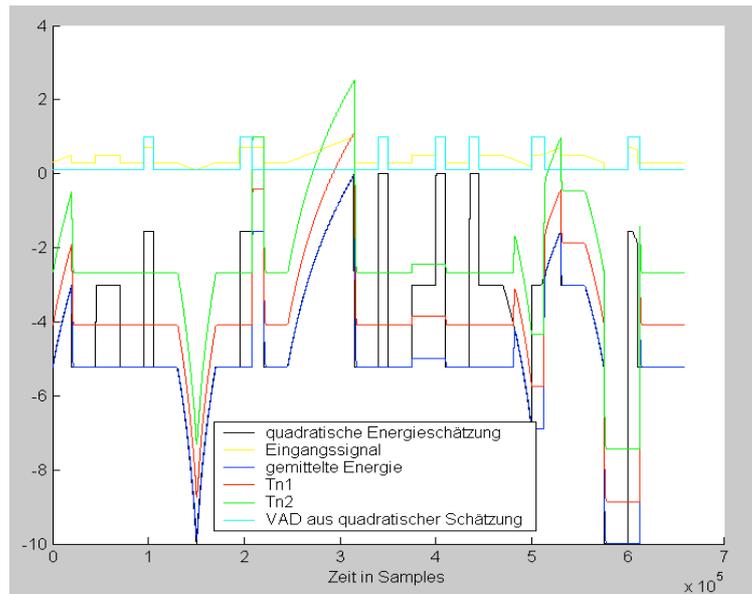


Abbildung 20: Testlauf für die VAD

Führt man keinen Skalierungswert d ein, und das Eingangssignal hat ist am Anfang auf Grund eines Kabelbruchs oder einer anderen Störung Null, dann wird die VAD die gemittelte Energie E_n und die beiden Thresholds T_{n1} und T_{n2} nicht updaten können und somit, wie in Abbildung 21 zu sehen, immer Sprache detektieren.

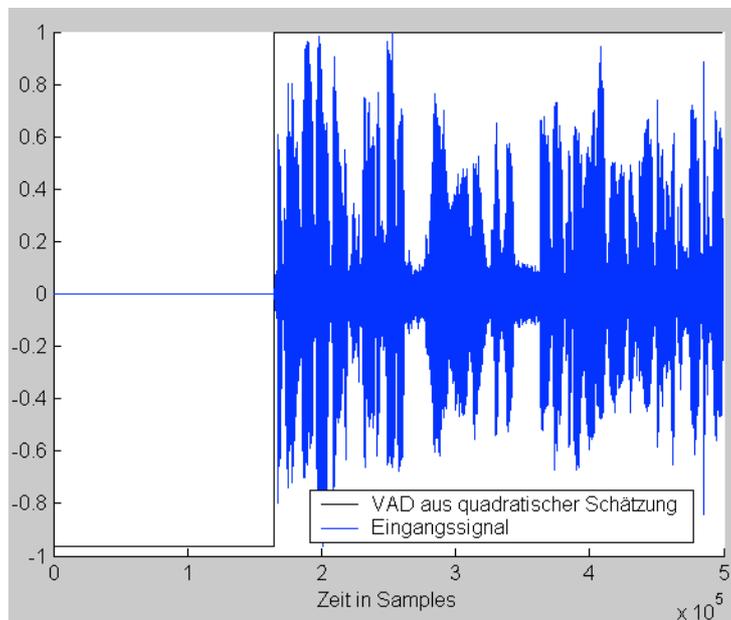


Abbildung 21: Testsignal mit „digital Null“ am Anfang und $d=0$

Verwendet man hingegen einen Skalierungswert $d=0.0001$ funktioniert die VAD nach einer kurzen Adaptionsphase wieder. (s. Abbildung 22)

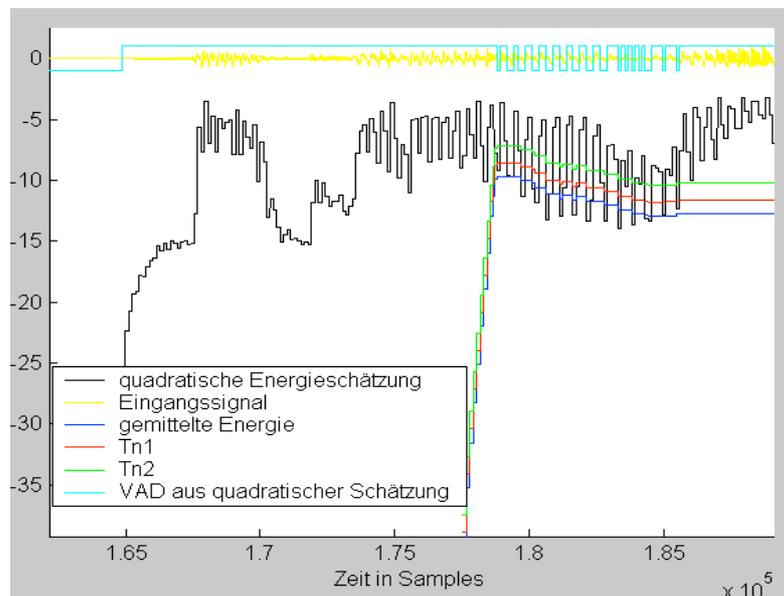


Abbildung 22: Testsignal mit „digital Null“ am Anfang und $d=0.0001$

5.2 Noise Estimator

Wenn die VAD einen reinen „Rauschblock“ detektiert hat, so wird der Noiselevel für jeden Knoten i des Waveletpakets geschätzt. Dies wurde auf drei Arten implementiert. Eine Möglichkeit ist, den betragsmäßig höchsten Koeffizienten in jedem Knoten als Noiselevel anzunehmen. Der Nachteil ist, dass dieser Wert möglicher Weise etwas zu hoch liegt. Durch die Multiplikation dieses Wertes mit einem empirisch ermittelten Faktor, der kleiner als eins ist kann dies kompensiert werden. Allerdings ist diese Methode sehr anfällig für statistische Ausreißer und wird daher nicht verwendet.

Eine andere Möglichkeit ist eine Varianzschätzung des Noiselevels.

$$n_i = \frac{1}{k} \sqrt{\sum_k (coeffs)^2} \quad \text{mit } k: \text{Anzahl der Koeffizienten im Knoten } i$$

Tests haben ergeben, dass der berechnete Noiselevel n_i deutlich zu niedrig ist. Lässt man die Normierung $\frac{1}{k}$ bei der Berechnung weg, so ergibt sich ein viel zu hoher Wert, so dass eine dritte Methode implementiert wurde.

Die beste Methode ist, den Mittelwert aus den Beträgen der Koeffizienten zu berechnen und diesen Wert als Noislevel zu verwenden.

$$n_i = \frac{1}{k} \sum_k |coeffs| \quad \text{mit } k: \text{ Anzahl der Koeffizienten im Knoten } i$$

Zwar ergibt sich auch hier ein etwas zu niedriger Wert, allerdings ist dieser nicht so weit von einer guten Schätzung entfernt, so dass mit einem empirisch ermittelten Faktor $p = 1.5$ ein guter Noislevel $N_i = p \cdot n_i$ geschätzt werden kann.

Abbildung 23 zeigt die Koeffizienten für den Knoten 8 und deren zugehörigen Threshold.

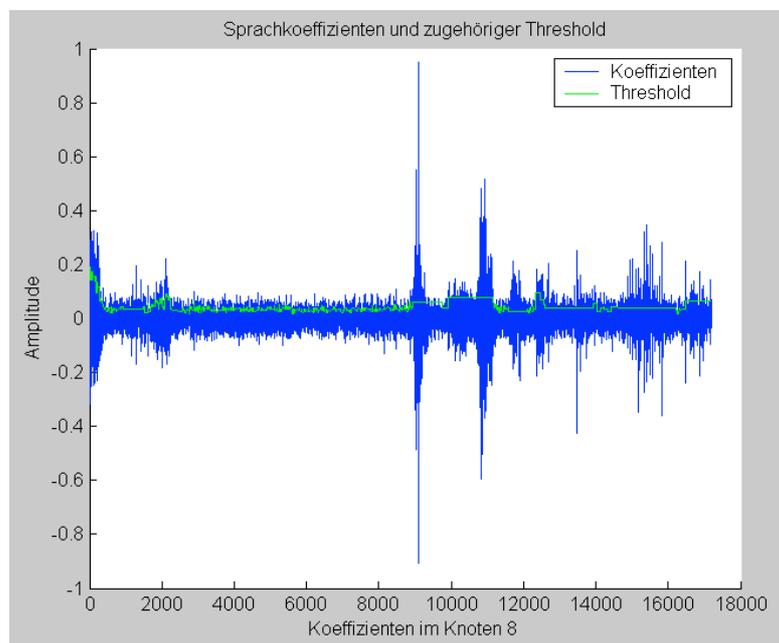


Abbildung 23: Koeffizienten und zugehöriger Threshold

Wenn die VAD Rauschen detektiert wird der Threshold ständig aktualisiert. Sobald Sprache detektiert wird, wird der Threshold „eingefroren“.

5.3 Wahl der Basis des Wavelet Pakets

Wenn ein Block von 128 Samples in ein Wavelet Paket transformiert worden ist, gilt es die günstigste Basis, also den besten Transformationsbaum zu finden, der Sprache und Rauschen möglichst gut trennt. Es ist nicht notwendig alle Hoch- und Tiefpassfilter der Transformation zu verwenden. Der Baum muss nicht vollständig sein, d.h. es müssen nicht alle Zweige die selbe Länge haben. Die einzige Bedingung ist, dass der gesamte Frequenzbereich mit den gewählten Knoten abgedeckt sein muss. Natürlich dürfen sich die Bereiche auch nicht überlappen, damit eine Rekonstruktion möglich ist. Das Ziel ist, die Basis so zu wählen, dass ein paar große, zur Sprache gehörende Koeffizienten und viele kleine, dem Rauschen

zuzuordnende Koeffizienten zur Verfügung stehen. Ist dies erfüllt, so wird die Supersoft Threshold Methode das Signal wirkungsvoll entrauschen können.

Um eine Basis aus der Wavelet Paket Zerlegung zu wählen wird die Transformation als Baumstruktur betrachtet. Das Originalsignal entspricht dabei der „Wurzel“ des Baumes, also dem Punkt $(0,0)$ in Abbildung 24. Die Wavelet Paket Koeffizienten befinden sich in den Knoten des Baumes. Um das Signal wieder rekonstruieren zu können muss eine „zulässige“ Teilmenge der Knoten ausgewählt werden. Eine gewählte Basis muss den gesamten Frequenzbereich ohne Überlappung abdecken. Ist ein Knoten ausgewählt, so darf keiner seiner „Kinder“ oder „Enkel“ ausgewählt werden, da dieser Frequenzbereich bereits abgedeckt ist. Abbildung 24 a) zeigt einen vollständigen Baum einer Wavelet Paket Zerlegung mit der Tiefe $N=3$. Die Abbildungen 24 b) und 24 c) zeigen zwei „zulässige“ Basen für den Baum.

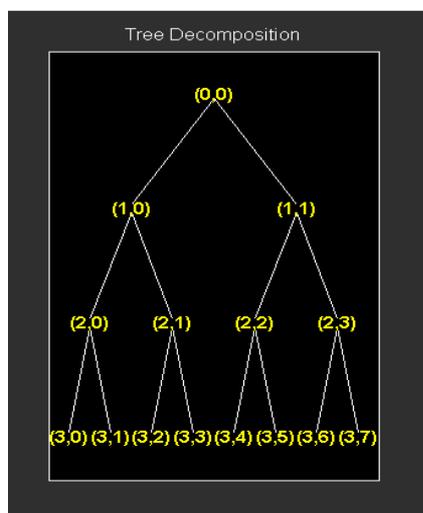


Abbildung 24: a) vollständiger Baum

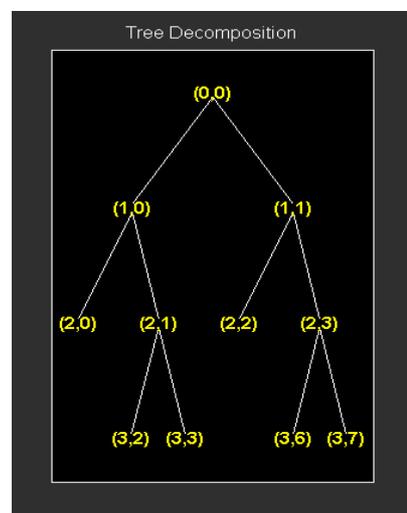


Abbildung 24 b) zulässige Basis

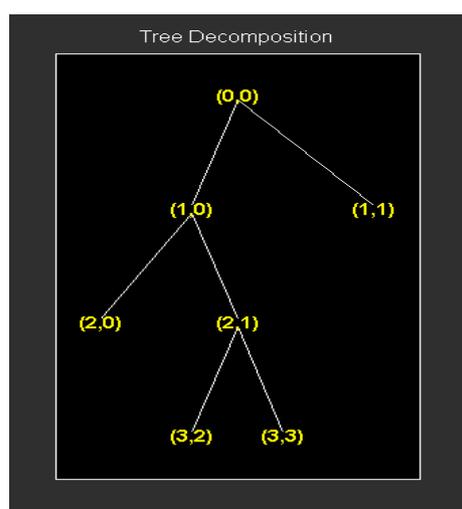


Abbildung 24: c) zulässige Basis

Wenn man davon ausgeht, dass die linken Zweige die Tiefpass- und die rechten Zweige die Hochpassfilterung darstellen, so besteht der Unterschied der beiden Basen aus Abbildung 24b) und 24c) in der geringeren hochfrequenten Auflösung der Basis aus Abbildung 24c).

Es wurde für dieses Projekt, die in der Arbeit „Noise Reduction of Speech Signals with Wavelets“ von Henrik Storm vorgeschlagene Wahl der besten Basis verwendet [1]. Das Maß für die „Güte“ einer ausgewählten Basis ist hierbei die enthaltene Information. Unter der Voraussetzung, dass die Knoten derart durchnummeriert sind, dass die Wurzel des Baumes dem Knoten 0 entspricht, seine beiden Kinder die Knoten 1 und 2 (von links nach rechts) sind, im nächsten Level sich die Knoten 3 bis 6 befinden usw., und das transformierte Signal v genannt wird, soll die Information im Knoten i mit $m_i(v)$ bezeichnet werden. Das Ziel ist es, eine Basis zu wählen, die so wenig Information wie möglich enthält. (In Analogie zur Entropie, dem oft verwendeten Maß für Information, die es häufig zu minimieren gilt.) Wie bereits gesagt, ist es wünschenswert, das Signal durch einige wenige große und viele kleine Koeffizienten darzustellen. Die kleinen Koeffizienten werden dann entfernt, unter der Annahme, dass sie aus dem Rauschen herrühren. Für diesen Fall ist ein Maß, das in einem bestimmten Knoten sowohl niedriger Resultate für kleine, als auch für große Koeffizienten liefert, ein gutes Maß. Das Optimierungsproblem für die Wahl der besten zulässigen Basis kann also wie folgt zusammengefasst werden:

Es gilt, diejenige Teilmenge von Knoten I zu finden, die ein Minimum an Information enthält.

$$M(v) = \sum_{i \in I} m_i(v)$$

Die Auswahl der Knoten findet folgendermaßen statt. Man startet die Auswahl am unteren Ende des Baumes und vergleicht die Information eines gegebenen Knotens mit der Summe der Informationen seiner Kinder. Voraussetzung dafür ist, dass das Informationsmaß additiv ist. Die „Kinderknoten“ haben auf Grund des Downsamplingprozesses genau halb so viele Koeffizienten wie der „Elternknoten“. Ist die Voraussetzung der Additivität erfüllt, so ist ein Vergleich der beiden Informationen möglich. Wenn die Summe der Informationen der Kinder kleiner ist als die Information des Elternknotens, dann werden die Knoten behalten und der Wert der Information des Elternknotens wird auf den, seiner beiden Kinder gesetzt. Wenn die Elterninformation kleiner ist, dann werden die beiden Kinder verworfen, der Baum wird an dieser Stelle also geschrumpft. Diesen Vorgang setzt man so lange fort, bis man ganz oben im Baum angekommen ist. Dieser Algorithmus wurde von M. V. Wickerhauser entwickelt und wird in [1] beschrieben.

Natürlich gibt es verschiedene Möglichkeiten eine Funktion zu entwickeln, die kleine Werte, für entweder sehr große oder sehr kleine Koeffizienten liefert. Eine Möglichkeit ist die Coifman-Wickerhauser Entropie:

$$m_i(v) = - \sum_{k=1}^{n^{2^{-d}}} (v_k^i)^2 \ln(v_k^i)^2$$

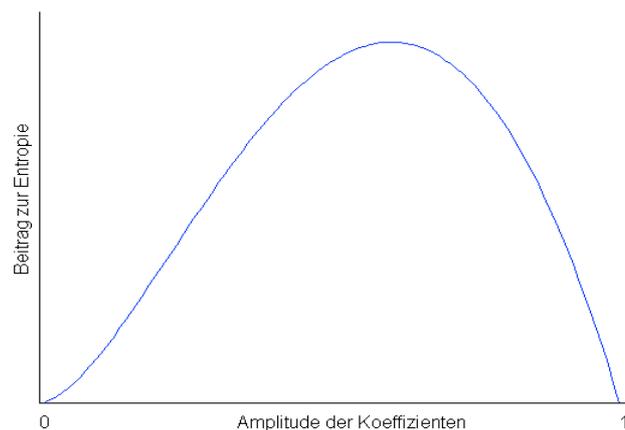


Abbildung 25: Coifman-Wickerhauser Entropie

Man sieht in Abbildung 25 deutlich, dass die Coifman-Wickerhauser Entropie kleine Werte liefert, wenn die Koeffizienten Werte nahe bei 1 oder 0 annehmen.

Da bei diesem Algorithmus aber eine Schätzung des durchschnittlichen Rauschlevels zur Verfügung steht, greift Storm auf ein selbstentwickeltes Informationsmaß zurück, das auch in dieser Arbeit verwendet wird. Angenommen der geschätzte Noisedurchschnitt im Knoten i sei n^i und die transformierte Funktion sei v , dann wird eine Funktion $m_i(v, n)$ gesucht, die einen niedrigen Wert hat wenn die geforderten Bedingungen erfüllt sind, einen hohen wenn die Bedingungen nicht erfüllt sind. Die Coifman-Wickerhauser Entropie wählt Basen aus, die hauptsächlich aus hohen und niedrigen Koeffizienten bestehen. Dies könnte für einen Denoisingprozess nicht optimal sein, da sich auch die mittleren Koeffizienten schlecht auf das Ergebnis auswirken können, wenn sie über dem Threshold liegen. Außerdem werden kleine Sprachkoeffizienten mit hoher Wahrscheinlichkeit gethresholdet und dies führt zu Sprachverzerrungen. Storm benutzt daher ein besser angepasstes Informationsmaß, das hauptsächlich Basen mit großen Koeffizienten (groß im Verhältnis zum Noiselevel, der ja zur Verfügung steht) auswählt:

$$m_i(v, n) = \sum_k \chi_{[|v_k^i| < n^i]} (v_k^i)^2$$

Man sieht in Abbildung 26, dass diese Vorschrift die Summe der Quadrate minimiert, die unterhalb des thresholds n_i liegen.

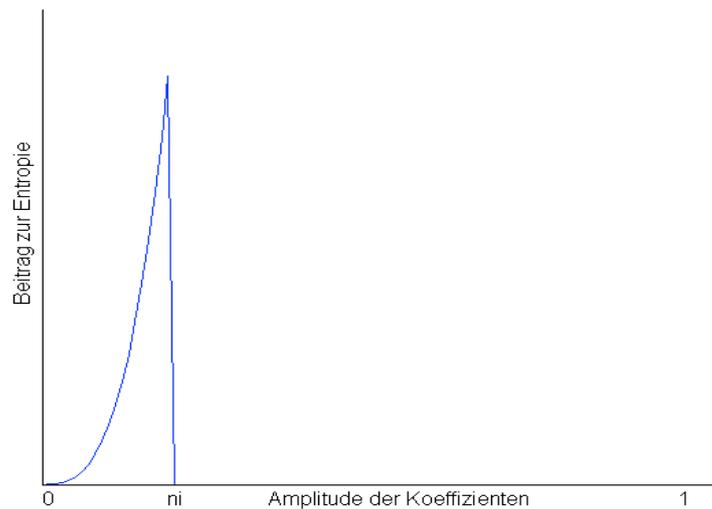


Abbildung: 26 DNE-Maß

Es werden mit dieser Vorschrift Basen ausgewählt, die hauptsächlich aus großen Koeffizienten bestehen. Der Thresholding-Vorgang wird also so wenig Energie wie möglich vom Signal entfernen. Storm nennt dieses Informationsmaß Denoising-Energie-Maß (DNE-Maß).

6 Beispiele

Die Hörbeispiele befinden sich im Präsentationsordner.

Literaturverzeichnis

- [1] Henrik Storm: „Noise Reduction of Speech Signals with Wavelets“
- [2] David L. Donoho: “De-Noising by Soft-Thresholding”
- [3] Christian Rauch: „Bildkompression mit der Wavelet-Transformation“
<http://home.arcor-online.de/chr.rauch/wavelet/kap2.htm>
- [4] Maarten Jansen: “Noise Reduction by Wavelet Thresholding”
- [5] Hamid Sheikhzadeh, Hamid Reza Abutalebi: „An Improved Wavelet-Based Speech Enhancement System”
- [6] R.R. Coifman, D.L. Donoho: “Translation-Invariant De-Noising”
- [7] <http://www.wavelet.org/tutorial/wbasic.htm>
- [8] Prof. Dr. Wolfgang Hess: „Grundlagen der Sprachsignalverarbeitung“
- [9] J.-C. Hong and Y.Y. Kim: “Determination of the Optimal Gabor Wavelet Shape for the Best Time–Frequency Localization Using the Entropy Concept”