Zum Einsatz fraktaler Verfahren in der Sprachkompression

Stephan Schneider
BTU Cottbus, Lehrstuhl Interkulturalität
schneide@tu-cottbus.de

Abstract: Fraktale Verfahren, die aus der Bildkompression bekannt sind, werden zur Kompression von Sprachsignalen angewendet. Der Signal/Rauschabstand und die Koeffizienten der affinen Transformationen werden analysiert und optimiert. Ferner sind Verbesserungsvorschläge für den praktischen Einsatz der fraktalen Kompression angegeben.

1 Einleitung

Im Gegensatz zur Bildverarbeitung ([1]) werden fraktale Kompressionsverfahren bei eindimensionalen Signalen im allgemeinen und speziell bei Sprachsignalen bislang eher selten angewendet (z.B. [3], [6]). Das mag daran liegen, daß ihr hoher Rechenaufwand die Fraktaltransformation gegenüber etablierten Verfahren wie z.B. der Prädiktionskodierung unattraktiv erscheinen läßt.

In dem vorliegenden Aufsatz werden einige Ergebnisse einer Untersuchung berichtet, die sich zum Ziel gesetzt hatte, die Fraktaltransformation einmal auf Sprachsignale anzuwenden. Angestrebt war weniger, die vorhandenen Verfahren zu ersetzen, als vielmehr herauszufinden, mit welcher objektiven und subjektiven Qualität sich Sprachsignale fraktal komprimieren lassen und wie die Algorithmen dazu ausgelegt und optimiert werden müssen.

Bild 1 zeigt den grundsätzlichen Aufbau eines fraktalen Kodierers / Dekodierers.

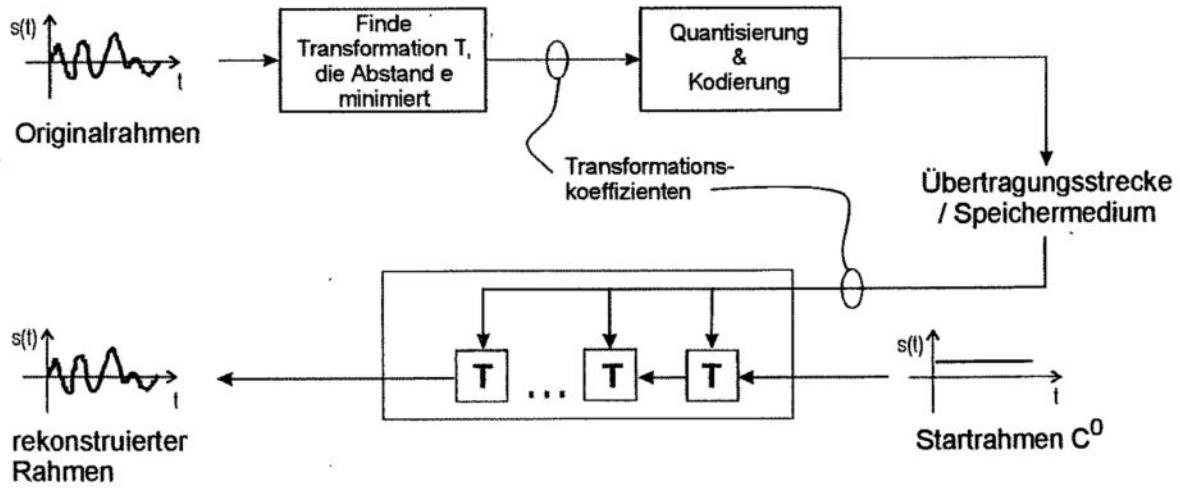

Figure 1: Grundaufbau eines fraktalen Kodierers / Dekodierers

2 Beschreibung des Verfahrens

Ausgangspunkt der Untersuchung bildete die fraktale Kodierung in ihrer einfachsten Form, wie sie im folgenden kurz beschrieben werden soll.

Da das Sprachsignal im Gegensatz zu Bilder, keine natürlichen Grenzen hat, wird es vor der Kompression in Rahmen gleicher Länge eingeteilt. Mit der "Länge" eines Rahmens oder Blockes ist im folgenden immer die Zahl der Abtastwerte gemeint. Die Rahmenlänge geht bei dem hier beschriebenen Verfahren unmittelbar in den erreichbaren Signal/Rauschabstand ein (siehe Abschnitt 3). Die folgenden Ausführungen beziehen sich immer auf einen Rahmen.

2.1 Das Kodierverfahren

Jeder Rahmen wird in Blöcke der Länge N eingeteilt, die den gesamten Rahmen lückenlos und ohne sich zu überlappen überdecken. Diese Blöcke werden im folgenden als "Zielblöcke" a_k bezeichnet, sie können als Vektor mit N Elementen geschrieben werden $a_n = \{a_{n,i}\}_{i=1}^{N}$.

Weiterhin wird aus dem Rahmen eine möglichst große Menge von Blöcken $b_k = \{b_{k,i}\}_{i=1}^{\alpha N}$ der Länge αN gebildet, die im folgenden als "Ursprungsblöcke" bezeichnet werden. Die Ursprungsblöcke dürfen einander überlappen, sie müssen nur aufeinanderfolgende Abtastwerte enthalten und ihre Länge muß um den Faktor α größer sein als die der Zielblöcke.

Das Ziel einer fraktalen Kodierung ist es, für jeden Zielblock a_n eines Rahmens eine affine Transformation T_n zu bestimmen, die einen geeigneten Ursprungsblock b_k so transformiert, daß er den Zielblock im Sinne eines Abstandsmaßes optimal annähert (Bild 2).

Die Transformationen T_n lassen sich in Operatorenschreibweise ([7]) ausdrücken:

$$T_n = d_n P_n D_n F_n + o_n P_n b_0 \tag{1}$$

wobei der Operator

- $F_n : X \rightarrow R^{\alpha N}$ den "besten" Ursprungsblock aus dem Rahmen herausnimmt,
- $D_n : R^{\alpha N} \rightarrow R^N$ den Ursprungsblock kontrahiert, d.h. zeitlich verkürzt, und
- $P_n : R^N \rightarrow X$ den transfomierten Ursprungsblock an die Stelle des entsprechenden Zielblockes einsetzt.

Die beiden Linearfaktoren

- der Amplitudenfaktor d_n und
- der Offsetfaktor o_n

skalieren den transformierten Ursprungsblock b_k bzw. den Offsetblock b_0 mit N konstanten Werten $b_0 = \{\beta, \beta, ...\}, \beta \neq 0$. Jeder Zielblock a_n wird also mit einer linearen Kombination aus einem transformierten Ursprungsblock b_k und einem konstanten (DC-)

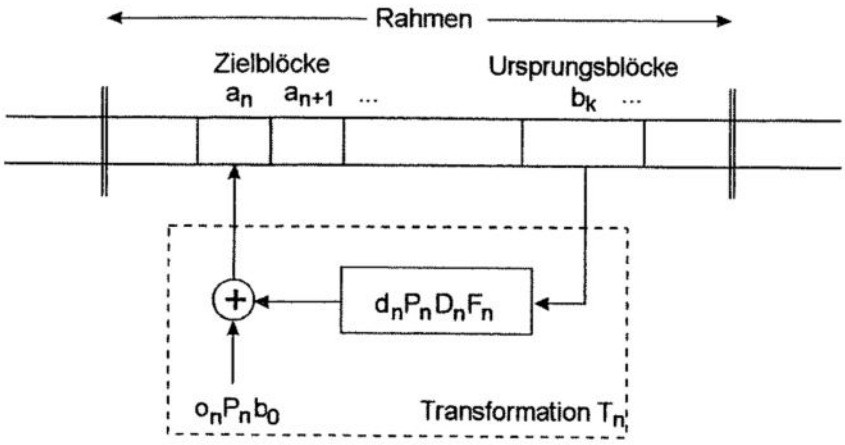

Figure 2: Transformationsvorgang bei der Kodierung und Dekodierung

Block b_0 angenähert. Die Basisvektoren dieser linearen Kombination können zusätzlich orthogonalisiert werden, um die Berechnung der Linearfaktoren zu erleichtern ([7]).

Das Verfahren ähnelt in vielen Aspekten der Vektorquantisierung. Der entscheidende Unterschied besteht darin, daß bei der fraktalen Kompression die Vektoren des Codebuches aus dem Signal selbst genommen werden und nicht aus einem vorher festgelegten, trainierten Codebuch. Das Codebuch, mit dem der Sender sendet, muß bei der fraktalen Kompression dem Empfänger nicht bekannt sein.

2.2 Bestimmung der Parameter der Transformation

Die Hauptaufgabe des Kodierers besteht nun darin, für jeden Zielblock die optimale Transformation T_n zu bestimmen. Als Optimierungskriterium wird ein Abstandsmaß definiert, welches den Abstand zwischen einem Zielblock a_n und einem transformierten Ursprungsblock T_nb_k angibt. Für praktische Anwendungen hat sich das L2-Maß (Euklidisches Maß) e bewährt, welches im folgenden verwendet wird:

$$e_{n,k}^2 = \sum_{i=1}^{N} (a_{n,i} - (T_nb_k)_i)^2 \tag{2}$$

Die Transformationen T_n werden so bestimmt, daß sie dieses Abstandsmaß minimieren. Der optimale Block b_k, d.h. die Adresse für den Operator F_n wird im einfachsten Fall durch bloßes Durchprobieren (lineare Suche) der gesamten Menge der Ursprungsblöcke bestimmt. Geeignete Suchverfahren, die den optimalen Block schneller finden, sind in Abschnitt 4 zitiert. Der Operator D_n, der die Ursprungsblöcke zeitlich verkürzt, ist beim Grundverfahren fest auf einen Kontraktionsfaktor von $\alpha_n = \alpha = 2$ eingestellt. Die Kontraktion um den Faktor α wird dadurch erreicht, indem der Mittelwert aus α Abtastwerten gebildet wird oder aus einem unkontrahierten Ursprungsblock nur jeder α -te Abtastwert genommen wird.

Die beiden Linearfaktoren d_n und o_n lassen sich mit einfacher Optimierungsrechnung bestimmen (die Indizes n des Zielblockes und k des Ursprungsblockes sowie die Grenzen der Summationszeichen sind der Lesbarkeit wegen weggelassen, die Summationen sind immer von $i = 1$ bis $i = N$ zu erstrecken):

$$d = \frac{N \sum_i a_i b_i - \sum_i a_i \sum_i b_i}{N \sum_i b_i^2 - \left(\sum_i b_i\right)^2} \tag{3}$$

$$o = \frac{1}{N}\left(\sum_i a_i - d \sum_i b_i\right) \tag{4}$$

Wird eine Orthogonalisierung durchgeführt und die Ursprungsblöcke mittelwertfrei gemacht, vereinfachen sich die Gleichungen zu

$$d = \frac{\sum_i a_i b_i}{\sum_i b_i^2}, \; o = \frac{1}{N} \sum_i a_i \tag{5}$$

Die Kompression eines Rahmens ist abgeschlossen, wenn für jeden Zielblock a_n eine optimale Transformation T_n bestimmt ist. Anstelle der Abtastwerte werden für einen Rahmen die Vereinigung aller Transformationen $\mathbf{T} = \bigcup_n T_n$, d.h. alle ihre Parameter gespeichert. Der Kompressionseffekt besteht darin, daß mit wenigen Transformationsparametern ein ganzer Block von Abtastwerten beschrieben werden kann.

2.3 Rekonstruktion des Signals

Das ursprüngliche Sprachsignal wird nicht durch eine Umkehrtransformation, sondern durch wiederholtes Ausführen der Transformation $\mathbf{T} = \bigcup_n T_n$ auf den zu rekonstruierenden Rahmen C wiederhergestellt. Der Inhalt des Rahmens C^0, mit dem die Rekonstruktion begonnen wird, ist dabei völlig unerheblich: der Startrahmen kann Abtastwerte von weißem Rauschen, von harmonischen Signalen oder konstante Werte enthalten. Die Transformation $\mathbf{T}$ sei so ausgelegt, daß sie einen eindeutigen Fixpunkt C^∞ besitzt. Dieser Fixpunkt entspricht der vorher ermittelten optimalen Näherung aller Zielblöcke a_n des originalen Rahmens A durch transformierte Ursprungsblöcke $T_n b_k$, d.h.

$$A \approx C^\infty = \lim_{\nu \to \infty} \mathbf{T}^{(\nu)} C^0 \tag{6}$$

Durch geeignete Maßnahmen bei der Bestimmung der Parameter ist sicherzustellen, daß die Transformation $\mathbf{T}$ schnell und sicher in ihren Fixpunkt konvergiert ([5]). So müssen die Kontraktionsfaktoren α_n des Operators D_n stets größer als 1 sein. Die Orthogonalisierung ist eine weitere Maßnahme.

Im allgemeinen reichen 5 - 8 Iterationen aus, um einen Rahmen sicher und innerhalb des Quantisierungsfehlers zu rekonstruieren. Trotz des iterativen Vorgehens ist der Rekonstruktionsalgorithmus wesentlich schneller als der Kodieralgorithmus. Eine Besonderheit der fraktalen Rekonstruktion ist, daß sie das Signal ohne zusätzliche Interpolationen mit beliebigen Auflösungen in der Zeit (Samplingfrequenz) und in der Amplitude wiederherzustellen vermag.

3 Analyse und Optimierung des Verfahrens

3.1 Signal / Rauschabstand

Durch einfache Rechnung läßt sich zeigen, daß das mittlere, auf die Varianz des Zielblockes normierte Fehlerquadrat proportional zum Quadrat des Korrelationskoeffizienten r_{ab} von

Ziel- und kontrahiertem Ursprungsblock ist (die Indizes n, k sind wieder weggelassen):

$$\frac{e^2}{N\sigma_a^2} = 1 - r_{ab}^2 \tag{7}$$

Daraus kann direkt der SNR für jede Kombination aus Ziel- und transformiertem Ursprungsblock bestimmt werden. Durch statistische Rechnung ([9]) konnten folgende bekannten Faustformeln bewiesen und quantifiziert werden:

- der SNR ist umso höher, je kleiner die Blocklänge N gewählt wird, da bei kleineren Blöcken weniger Variationsmöglichkeiten bestehen. Mit kleineren Blöcken sinkt aber der Kompressionfaktor.
- der SNR liegt umso höher, je größer die Menge an (unterschiedlichen) Ursprungsblöcken ist, da die Wahrscheinlichkeit einen passenden Ursprungsblock zu finden, steigt. Eine größere Zahl an Ursprungsblöcken bewirkt aber einen höheren Aufwand bei der Suche.

Durch einen Vergleich von Modellrechnungen mit Meßwerten hat sich gezeigt, daß bei Sprachsignalen Zielblöcke aus stimmhaften Lauten (Vokale) wesentlich besser, d.h. mit höherem SNR, kodiert werden als Zielblöcke aus stimmlosen Lauten. Weiterhin ließ sich zeigen, daß der SNR ohne Berücksichtigung der Quantisierung der Parameter von **T** von den Varianzen der Ziel- und Ursprungsblöcke unabhängig ist.

3.2 Quantisierung und Kodierung der Parameter

Bevor die Parameter der Transformationen T_n digital gespeichert werden können, müssen sie quantisiert und kodiert werden. Die Adresse des optimalen Ursprungsblockes braucht nicht quantisiert zu werden, da sie bereits eine ganze Zahl ist. Da die Ursprungsblöcke statistisch gleichmäßig über den Rahmen verteilt sind, wie eine empirische Untersuchung zeigte, kann die Adresse direkt mit $c_F = \lceil \log_2 M \rceil$ Bits kodiert werden, wobei M die Zahl der Ursprungsblöcke angibt und $\lceil x \rceil$ die nächsthöhere ganze Zahl zu x .

Wie empirische Untersuchungen ergeben haben, sind die Linearfaktoren d_n und o_n für gleiche n meist recht gut voneinander dekorreliert, so daß sie getrennt voneinander quantisiert und kodiert werden können. Eine Orthogonalisierung, besonders eine mit einer optimalen (Eigenvektor-)Basis, kann die Dekorrelation noch verbessern.

Der Amplitudenfaktor d_n wie der Offsetfaktor o_n haben stark ungleichförmige, symmetrisch um 0 angeordnete Häufigkeitsverteilungen. Bild 3a und 3b zeigen zwei typische Beispiele für Häufigkeitsverteilungen (HV) von d_n und o_n. Näherungen der theoretischen Wahrscheinlichkeitsdichtefunktionen (WDF) wurden in [9] hergeleitet. Die Orthogonalisierung wirkte sich auf die WDF's nur marginal aus.

Wegen der ungleichförmigen WDF's wurden für beide Koeffizienten Lloyd-Max-Quantisierer ([4]) mit ungleichförmigen Quantisierungscharakteristika eingesetzt, die anhand der empirischen HV's optimiert wurden. Da sich die HV's der Koeffizienten von Rahmen zu Rahmen ändern, wurden die HV's aus mehreren Rahmen gebildet. Eine adaptive Quantisierungscharakteristik, die von Rahmen zu Rahmen neu angepaßt wird, lohnt den Aufwand nicht.

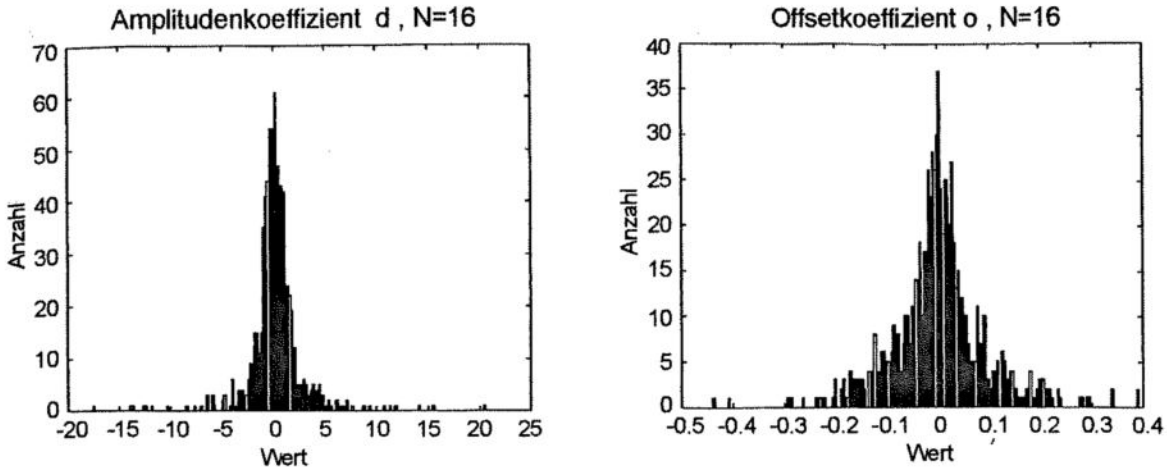

Figure 3: a, b : Häufigkeitsverteilung des Amplituden- und des Offsetkoeffizienten in absoluten Zahlen. Blocklänge N=16, Rahmen mit 12.000 Abtastwerten zu 8 Bit.

Der Fehler, der bei der Quantisierung der beiden Koeffizienten d_n und o_n entsteht, bewirkt eine veränderte Transformation $\tilde{\mathbf{T}}$, deren Fixpunkt $\tilde{C}^{\infty}$ vom Fixpunkt der nicht quantisierten Transformation $\mathbf{T}$ leicht abweicht. Durch die Fixpunkteigenschaften wird der Quantisierungsfehler bei der iterativen Rekonstruktion des Signals nicht akkumuliert, sondern lediglich das Endergebnis ist gegenüber dem Original verändert.

Wieviele Bits notwendig sind, um die Koeffizienten mit hinreichender Genauigkeit zu kodieren, wurde noch nicht exakt untersucht. Erfahrungswerte aus der Bildkompression schlagen $c_A = 5..7$ Bits für den Amplitudenkoeffizienten und $c_O = 6..8$ Bits für den Offsetkoeffizienten vor. Zusammen mit der Adresse des Ursprungsblockes, die mit c_F Bits kodiert wird, ergeben sich etwa 3 - 4 Bytes je Zielblock. Damit beträgt das Kompressionsverhältnis etwa $N/4$ für eine konstante Blocklänge von N , d.h etwa 8 für eine Blocklänge von $N = 32$. Kopf- und Rahmeninformationen der komprimierten Datei sind darin ebensowenig berücksichtigt wie Prüfbits.

Schließlich wurde noch die Autokorrelierte der beiden Koeffizienten d_n und o_n untersucht. Da die normierte Autokorrelationsfunktion schon nach Zeitverschiebungen von 1 -2 Abtastwerten auf Beträge unter 0.2 fällt, bringt eine Prädiktionskodierung kaum einen Gewinn.

4 Verbesserungen

Das Verfahren aus Abschnitt 2 ist grundsätzlich funktionstüchtig und eignet sich wegen seiner Einfachheit für die statistische Analyse. Für den praktischen Einsatz muß das Verfahren jedoch in vielen Punkten verbessert werden. Anregung dazu können die umfangreichen Arbeiten zur fraktalen Bildkompression (z.B. in [1]) geben. Einige Verbesserungen seien hier kurz skizziert.

4.1 Dynamische Blocklängen

Wie aus Abschnitt 3 bekannt, verbessert sich der SNR bei kleineren Blocklängen N . Dieses Verhalten kann dazu ausgenutzt werden, einen bestimmten Mindestwert des SNR für die Blöcke des Sprachsignals zu erzwingen. Das Kompressionsverfahren beginnt mit einer voreingestellten Blocklänge N_0 für die Zielblöcke. Sollte sich ein Zielblock nach Durchsicht aller Urpsrungsblöcke nicht mit dem vorgegebenen SNR kodieren lassen, wird er in einem rekursiven Vorgehen in zwei Teile der Länge $N_0/2$ geteilt, die getrennt kodiert werden. Sollten auch diese Blöcke nicht mit dem geforderten SNR kodiert werden können, werden sie nochmals halbiert usw.. Schwierig zu kodierende Abschnitte des Sprachsignals wie die stimmlosen Laute werden so mit zahlreichen kleineren Blöcken abgedeckt, einfach zu kodierende mit wenigen großen.

Umgekehrt kann dieses Verfahren auch dazu dienen, ein bestimmtes Kompressionsverhältnis oder eine geforderte Länge der komprimierten Datei genau einzustellen. Hierbei werden iterativ die zwei nebeneinanderliegenden Zielblöcke mit dem jeweils besten SNR vereinigt, solange bis eine festgelegte Dateigröße erreicht ist.

4.2 Verbesserte Suchverfahren

Da der bisherige Kodierungsalgorithmus die gesamte Menge der Ursprungsblöcke durchsucht, dauert die Kodierung eines Rahmens je nach Rahmengröße relativ lange. Schon frühzeitig wurden deshalb Verfahren vorgeschlagen, die die lineare Suche durch andere Suchverfahren ersetzen, um den Kodiervorgang zu beschleunigen. Die frühesten Verfahren klassifizieren die Ursprungsblöcke nach verschiedenen Merkmalen und durchsuchen für einen bestimmten Zielblock nur noch eine Klasse von Ursprungsblöcken. Von D. Saupe ([8]) stammt der Vorschlag, die Suche nach dem am besten geeigneten Ursprungsblock als Suche nach dem nächsten Nachbarn aufzufassen. Zu diesem Problem existieren Suchverfahen, wie z.B. der "kd-tree" von Friedman et al. ([2]). Damit ist zumindest theoretisch möglich, de Anzahl der Suchoperationen auf $O\,[\log(M)]$ zu beschränken. Die Operationen, die notwendig sind, um die Ursprungsblöcke entsprechen anzuordnen und den Zugriffsschlüssel zu berechnen, sind bei einer Abschätzung der Kodierzeiten mit zu berücksichtigen.

4.3 Nachbearbeitung mit Wavelets

Durch das stumpfe Aneinandersetzen der Zielblöcke entstehen bei der Rekonstruktion des Signals zahlreiche Amplitudendiskontinuitäten. Da die Grenzen der Ursprungsblöcke nicht mit den Rändern der Zielblöcke übereinstimmen, sind die Amplitudendiskontinuitäten durch die iterativen Kopiervorgänge statistisch über das Signal verteilt. Wegen der statistischen Verteilung werden sie nicht als Ton, sondern als Prasseln oder Knistern wahrgenommen. Nachdem Versuche mit Interpolationen an den Schnittkanten fehlgeschlagen waren, wurde ein Filterverfahren auf der Basis der Wavelettransformation implementiert.

Dazu wird ein ganzer Rahmen, der vorteilhaft eine dyadische[1] Länge hat, mit Hilfe

[1] d.h. einer ganzzahligen Potenz von 2 entsprechend

der diskreten Wavelettransformation in eine Folge von Waveletkoeffizienten entwickelt

$$x(t) \rightarrow x_n^m : \quad x_n^m = \int_{-\infty}^{\infty} x(t)\psi_n^m(t)\, dt \tag{8}$$

, wobei $\psi_n^m(t)$ das um n verschobene, mit m skalierte Basiswavelet $\psi(t)$ ist: $\psi_n^m(t) = 2^{-m/2}\psi(2^{-m}t - n)$. Kurze, sprunghafte Ereignisse im Sprachsignal, wie z.B. die Amplitudendiskontinuitäten, werden von den Koeffizienten x_n^m mit hohen Werten von m erfaßt. Die Koeffizienten werden nun mit einer Filterfunktion $F(m)$ bewertet, die die Skalen $m > m_g$ stark dämpft oder ganz unterdrückt. Bei der nachfolgenden inversen Wavelettransformation sind die Amplitudendiskontinuitäten wirksam herausgefiltert.

References

[1] Yuval Fisher, editor. *Fractal image compression: theory and application.* Springer, Berlin, 1995.

[2] Bentley J. L. Finkel R. A. Friedman, J. H. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. on Mathematical Software*, 3(3):209 – 226, September 1977.

[3] T. Holter. Fraktal koding av tale. Master's thesis, The Norwegian Institute of Technology, Trondheim, 1990.

[4] S. P. Lloyd. Least squares quantization in pcm. *IEEE Trans. on Information Theory*, IT-28(2):129 – 137, March 1982.

[5] L. Lundheim. *Fractal Signal Modelling for Source Coding.* PhD thesis, Department of Electrical and Computer Engineering, Trondheim, September 1992.

[6] Young K. L. Maragos, P. Fractal excitation signals for celp speech coders. In *Proc. ICASSP-90*, pages 669 – 672, 1990.

[7] Lepsoy S. Oien, G.E. On the benefits of basis orthogonalization in fractal compression. In Y. Fisher, editor, *Fractal Image Compression and Analysis*, pages 65 – 88, Berlin, 1998. NATO ASI Series F: Computer and Systems Sciences, Vol. 159, Springer.

[8] D. Saupe. Fractal image compression via nearest neighbor search. In Y. Fisher, editor, *Fractal Image Compression and Analysis*, pages 95 – 116, Berlin, 1998. NATO ASI Series F: Computer and Systems Sciences, Vol. 159, Springer.

[9] S. Schneider. *Kodierung von Sprachsignalen mit iterierten Funktionensystemen (Arbeitstitel).* PhD thesis, BTU Cottbus, in Vorbereitung.