Subband Coding

Andreas Sellmeier

4. April 1997

Proseminarvortrag am Lehrstuhl für effiziente Algorithmen
(Prof. Mayr)
Institut für Informatik
Technische Universität München

Inhaltsverzeichnis

1 Einführung

Bei diesem Verfahren wird das Quellsignal in seine einzelnen Bestandteile zerlegt, um es dann mit einem oder mehreren der in vorhergehenden Vorträgen behandelten Komprimierungsverfahren zu codieren. Es gibt zwei unterschiedliche Verfahren, die Subband Codierung anzugehen.

- Das erste Verfahren teilt die Quelle in unterschiedliche Frequenzbänder mit der Hilfe von digitalen Filtern.
- Das zweite, auch als *multiresolution analysis* bekannt, geht von einem anderen Startpunkt bei der Zerlegung aus. Die Grundidee hier ist, daß man das Signal mit Hilfe sogenannter *Wavelets* aufteilt. Ich werde allerdings nicht auf dieses Verfahren eingehen , da die Beschreibung sehr kompilziert ist und den Rahmen des Vortrages sprengen würde.

2 Motivation und Hintergrund

Nachdem das Eingangssignal in seine Bestandteile zerlegt wurde, kann man das Komprimierungsverfahren für jeden einzelnen Teil des Signals so auswählen, daß man die höchste Komprimierungsrate erreicht. Außerdem hat jede Komponente einen unterschiedlichen Informationsgehalt. Oft haben Signale zwar eine hohe Varianz zwischen zwei nebeneinander liegenden Werten, doch weisen sie einen allgemeinen Trend auf. Man betrachtet Werte innerhalb eines veränderbaren Fensters und mittelt diese, um die Funktion zu glätten. Generierung einer neuen Frequenz $\{ y_n \}$ mit der Fensterbreite von zwei durch mitteln der benachbarten Werte von $\{ x_n \}$:

$$y_n = \frac{x_n + x_{n-1}}{2}, \qquad x_{-1} = x_0$$

Diese neue Funktion $\{ y_n \}$ ist natürlich viel schöner zu codieren, als die vorherige. Nun wollen wir aber nicht diese, sondern $\{ x_n \}$ codieren. Deswegen führen wir noch die Differenz-Sequenz $\{ z_n \}$ ein:

$$z_n = x_n - y_n = \frac{x_n - x_{n-1}}{2}$$

2.1 Beispiel

Es ist folgende Sequenz von Werten $\{ x_n \}$ gegeben:

10	14	10	12	14	8	14	12	10	8	10	12

Hieraus folgt dann für $\{ y_n \}$:

10	12	12	11	13	11	11	13	11	9	9	11

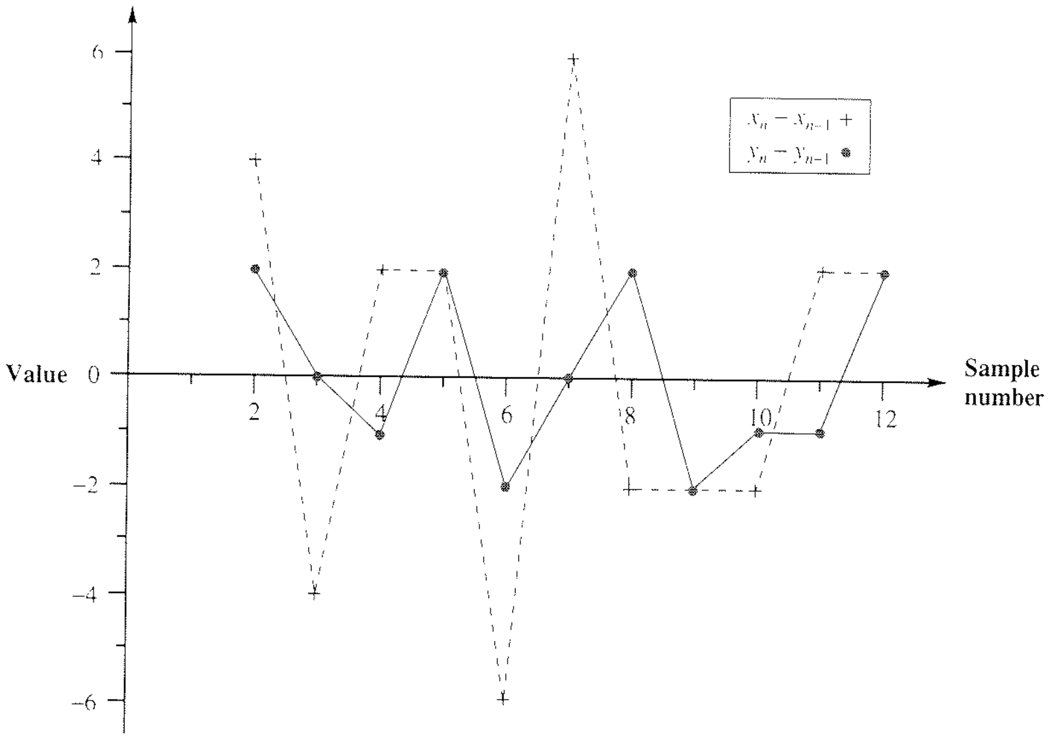

Abbildung 1: Differenz Sequenzen, die aus dem orginal- und dem durchschnitts-Signal gebildet wurden

Nun können wir noch die Differenzen-Sequenz $\{ z_n \}$ bilden:

$$0 \quad 2 \quad -2 \quad 1 \quad 1 \quad -3 \quad 3 \quad -1 \quad -1 \quad -1 \quad 1 \quad 1$$

Hier sieht man ganz deutlich, daß sich die Werte nur noch in einem Intervall der Länge 6 bei z_n bewegen, im Gegensatz zu dem bei x_n, wo dieses Intervall die Länge 14 hat.

2.2 Fehlerberechnung

Nun sehen wir uns an, was passieren würde, wenn wir nach einer herkömmlichen Methode komprimieren:
Da hier eine deutliche Beziehung zwischen denn einzelnen Werten besteht erwägen wir mit der Methode der *differential pulse code modulation (DPCM)* zu komprimieren. Dabei ergibt sich bei der Differenz der einzelnen Werte $\{ x_n - x_{n-1} \}$ zueinander eine Bereich von -6 bis 6. Wenn wir jetzt m Bits zum quantisieren dieser Werte benutzen, folgt daraus ein Quantisierer mit $M = 2^m$ Wiederaufbau-Werten. Daraus folgt ein Quantisierungs-Intervall Δ von:

$$\Delta = \frac{12}{M}$$

Daraus ergibt sich ein maximaler Quantisierungs-Fehler von $\frac{\Delta}{2}$ oder $\frac{6}{M}$. So folgt für den Quantisierungs-Fehler bei:

- $\{ y_n \}$: $\frac{2}{M}$
- $\{ z_n \}$: $\frac{3}{M}$

Somit erhält man einen maximalen Fehler von $\frac{5}{M}$, der obwohl der Empfänger das Signal erst wieder rekonstruieren muß besser ist. Damit können wir für die selbe Anzahl an Bits pro Sequenz $\{ y_n \}$ und $\{ z_n \}$ codieren und erzielen auch noch ein besseres Ergebnis. Das Hauptproblem ist nur, daß wir die Bitrate verdoppeln. Dies kann vermieden werden, indem man jeweils immer nur einen veränderten Wert von $\{ y_n \}$ und $\{ z_n \}$ sendet. Dazu teilt man nun die Sequenz in gerade und ungerade Untersequenzen und überträgt z.B. nur $\{ y_{2n} \}$ und $\{ z_{2n} \}$. Die Wiederherstellung läuft nun nach folgenden Gleichnungen ab:

$$y_{2n} + z_{2n} = x_{2n} \qquad y_{2n} - z_{2n} = x_{2n-1}$$

3 Frequenzbereich und Filterung

3.1 Fourier Reihen und Transformationen

Dieser mathematischer Hintergrund wird für die spätere Konstruktion von Filtern benötigt, die uns helfen das Signal in seine verschiedene Bestandteile zu zerlegen. Bis jetzt haben wir unser Signal in eine langsame und schnelle Komponente aufgeteilt. Nun brauchen wir eine Größe, die ausdrückt, wie sich das Signal mit der Zeit verändert. Dazu führen wir die Frequenz ein. Die Frequenz wird in Hertz (Hz) gemessen und ist ein Ausdruck für die Änderung des Signal mit der Zeit. Eine sinusförmiges Signal, welches 60 Schwingungen pro Sekunde ausführt, hat eine Frequenz von 60 Hz. Bevor man mit Frequenz immer nur eine sinusförmige Funktion in Verbindung gebracht hat, zerlegte Alexander Graham

Bell's Telegraph schon Sprache in eine rechteckige Welle. Erst später wurden Kondensatoren und Induktoren (Spulen), die sinusförmige Funktionen erzeugen können, erfunden.
Wir können jedes Signal $f(t)$, das periodisch mit der Periode T ist, in eine reale oder komplexe sinusförmige Funktion umwandeln:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(n\frac{2\pi}{T}t\right) + \sum_{n=1}^{\infty} b_n \sin\left(n\frac{2\pi}{T}t\right)$$

Diese Darstellung wird *Fourier-Reihe* genannt. Wir können komplexe Fourier-Reihen erweitern, so daß sie auch nichtperiodische Funktionen $f(t)$ darstellen können, indem man zuerst eine periodische Erweiterung $f_p(t)$ schafft:

$$f_p = \sum_{k=-\infty}^{\infty} f(t - kT)$$

Nun läßt man T nach unendlich gehen und bekommt die Darstellung für $f(t)$. Dies nennt man dann die *Fourier Transformation*, und da wir hier mit diskreten Signalen arbeiten, benutzen wir die *diskrete Fourier Transformation (DFT)*. Eine Sequenz $\{ x_n \}$ der Länge N wird durch die DFT so dargestellt:

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{\frac{-i2\pi kn}{N}}$$

Das Orginal kann ausgehend von der transformierten Sequenz $\{ c_k \}$ durch die inverse Transformation wiederhergestellt werden:

$$x_k = \sum_{k=0}^{N-1} c_n e^{\frac{i2\pi kn}{N}}$$

Wenn man nun ein Signal komprimieren will, das hohe wie auch niedrige Frequenzen beinhaltet, dann sollte man es vorher in zwei Signale aufteilen, um eine bessere Ausbeute zu erreichen, da die meisten Komprimierungsverfahren größere Sprünge weniger gut vorhersagen können. Außerdem kann man damit viel mehr auf die Eigenschaften des menschlichen Gehörs eingehen und demnach die Kodierungstechnik auswählen.

3.2 Filter

Ein Sytem, welches bestimmte Frequenzkomponeten isoliert, nennt man *Filter*. Es gibt *low-pass* und *high-pass Filter*, die entweder alles überhalb oder unterhalb einer bestimmten Frequenz f_0 (*cutoff frequency*) abschneiden.
Wenn man davon ausgeht, daß die höchste Frequenz eines Signals f_0 ist, dann muß man dies mehr als $2f_0$ mal samplen. Diese Entdeckung von Harry Nyquist,

der Mathematiker an den Bell Laboratories war, legte den Grundstein für weite Teile der digitalen Kommunikation. Sie wird als *Nyquist theorem* oder *Nyquist rule* bezeichnet.
Bestandteile mit Frequenzen, die höher als die halbe sampling Rate sind, zeigen sich bei niedrigeren Frequenzen. Dies wird *aliasing* genant. Um dies zu vermeiden, beinhalten die meisten Systeme *antialiasing filter*, die den Eingang zum Sampler auf die halbe sampling Frequenz beschränken. Die allgemeine Form der Ein- und Ausgangs Beziehung der Filter ist gegeben durch:

$$y_n = \sum_{i=0}^{N-1} a_i x_{n-i} + \sum_{i=1}^{M} b_i y_{n-i}$$

Hierbei ist $\{ x_n \}$ der Eingang und $\{ y_n \}$ der Ausgang des Filters. $\{ a_i \}$ und $\{ b_i \}$ sind die Filterkoeffizienten.
Einige Definitionen:

- Die *impulse response* ist die Ausgangssequenz eines Filters, wenn die Eingangsequenz aus einer 1 gefolgt von lauter Nullen besteht.
- Sind alle b_i Null, dann stirbt die impulse response nach N samples aus, wobei N ist die Anzahl der *taps* in einem Filter ist.
- Filter bei denen die impulse response ausstirbt nennt man *finite impulse response filters* oder FIR Filter. Daraus kann man zeigen, daß die impulse response einen FIR Filter eindeutig charakterisiert.
- Wenn $\{b_i\}$ nicht nur aus lauter Nullen besteht, dann kann die impulse response theoretisch ewig weiter gehen. Diese Filter nennt man dann *infinite impulse filters* oder IIR Filter.

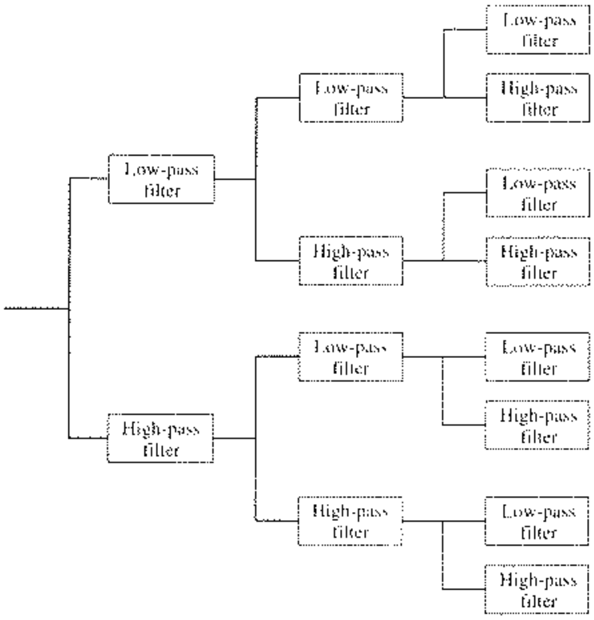

Abbildung 2: Acht Band Filter Reihe

3.3 Beipiele für Filter Systeme

Die meisten Filtersysteme bestehen aus stufenförmig angeordnete Filter Reihen, die auf jeder Stufe einen low-pass und einen high-pass Filter enthalten.
Beliebte Filter, die zum Subband Coding benutzt werden sind die *Quadrature mirror filters (QMF)*. Diese Filter besitzen die Eigenschaft, wenn die Impulse Response der Tiefpassfilter gleich $\{h_n\}$ ist, dann ist dei Impulse Response der Hochpassfilter Gleich $\{(-1)^n h_{N-1-n}\}$ Hierunter gibt es wieder zwei gern benutzte:

- Johnston Filter als 8, 16 oder 32 tab Ausführung
- Smith-Barnwell als 8 oder 16 tab Ausführung

Diese Filter sind natürlich nur Beispiele der beliebtesten Filter. Es gibt aber noch viele andere.

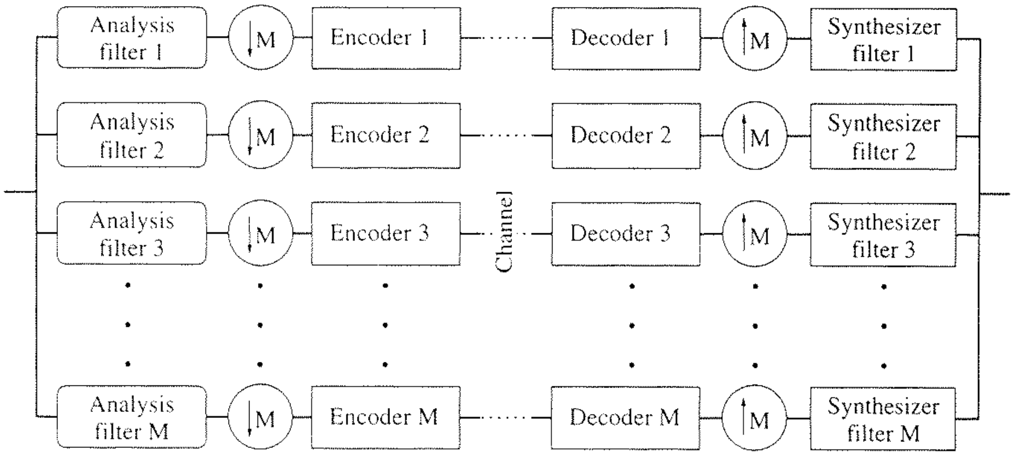

Abbildung 3: Blockdiagramm des Subband Coding Systems

4 Subband Coding Algorithmus

4.1 Algorithmus

Beim Subband Coding wird das Quellsignal zuerst durch ein Filter-System, das man auch *analysis filter bank* nennt geschickt. Diese besitzt die Eigenschaft, daß sie das komplette Spektrum der Frequenzen des Quellsignals abdeckt. Die Bänder, die durchgelassen werden können überlappend oder auch nicht überlappend sein. Aufgrund des Nyquist Gesetzes kann man nun das Signal *subsamplen*. D.h. wir vermindern die Anzahl der Sample. Dies nennt man auch *Dezimierung* oder *Downsampling*. Nun kann man die einzelnen Sample mit den bekannten Komprimierungsverfahren, wie z. B. adatives DPCM, PCM und Vektor Quantisierung bearbeiten.

4.2 Bit Allocation

Jetzt ist aber nicht gleich viel Information des endgültige Signals in den einzelnen Samples enthalten. Deswegen muß man sich eine Art Zuteilungverfahren für die Bits überlegen. Es gibt zwei einfache Formen der bit allocation.
Dafür nehmen wir an, das das Eingangsignal in M Bänder zerlegt, dann um M dezimiert und zum Schluß mit einem skalaren Quantisierer komprimiert wurde.

- Die durchschnittliche Zahl der Bits pro Sample die das Subband Coding System benutzt ist R.
- Die Bitzahl, die der Ausgang des $k$ten benutzt sei R_k.

- Die Wiederherstellungsfehlervarianz sei für den $k$ten Quantizierer $\sigma_{r_k}^2$
- Die Eingangsvarianz des $k$ten Quantizierer $\sigma_{y_k}^2$.

Man versucht nun die Wiederherstellungsfehlervarianz $\sigma_{r_k}^2$ zu minimieren:

$$\sigma_{r_k}^2 = \alpha_k 2^{-2R_k} \sigma_{y_k}^2$$

Die komplette Wiederherstellungsfehler ist damit:

$$\sigma_r^2 = \sum_{k=1}^{M} \alpha_k 2^{-2R_k} \sigma_{y_k}^2$$

Die Minimierung wird dann mit Hilfe der Lagrange Multiplikatoren durchgeführt. Dazu benötigt man noch die Nebenbedingung:

$$R = \frac{1}{M} \sum_{k=1}^{M} R_k$$

Daraus folgt:

$$R_k = R + \frac{1}{2} \log_2 \frac{\sigma_{y_k}^2}{\prod_{k=1}^{M} \left(\sigma_{y_k}^2\right)^{\frac{1}{M}}}$$

Dies ist eine notwendige, aber nicht hinreichende Bedingung für ein Minimum von R_k. Es ist zusätzlich nicht sicher, ob hier überhaupt R_k ganzzahlig oder sogar positv ist. Standardmäßig werden dann alle negativen R_k Null gesetzt. Nun erhöht sich dadurch die Bitzahl, was dazu führt, daß man die positiven R_k gleichmässig reduziert bis die Bitzahl gleich der Geforderten ist.
Es gibt allerdings auch noch eine rekursive Annäherung, die in der Praxis gute Resultate liefert ist:

1. Berechne σ_k^2 am Ausgang jedes Filters
2. Setze $R_k = 0$ für alle k und setze $R_b = MR$. R_b ist die komplette Anzahl der Bits die zur Verteilung vorhanden sind.
3. Sortiere die Varianzen $\sigma_{y_k}^2$ absteigend. $\sigma_{y_l}^2$ sei das Maximum.
4. Erhöhe R_l um eins und teile $\sigma_{y_l}^2$ durch 2.
5. Erniedrige R_b um eins, wenn R_b gleich eins ist, dann stoppe anderenfalls gehe zu 3.

5 Bedeutung für die Sprachkomprimierung

G.722 ist eine Technik für die Weitbandkodierung von Sprachsignalen und wurde von der ITU (International Telecommunication Union) eingeführt. Hier will man Sprache mit hoher Qualität bei 64 kbits/s übertragen. Hier wird das Audio Signal mit 7kHz gefiltert, um *aliasing* zu verhindern, und dann mit 16000 Sample pro Sekunde digitalisiert. Dies geschieht mit 7kHz, weil analoge Filter nicht so genau an den Flanken abschneiden. Der Ausgang wird dann um einen Faktor von zwei dezimiert und darauf mit der Methode der *adaptiven differential pulse code modulation (ADPCM)* komprimiert.

6 Bedeutung für die Audiokomprimierung

MPEG Audio wurde von der Moving Pictures Experts Group eingeführt. Eigentlich ist die Musikkomprimierung hier nur ein Art Abfallprodukt der Videokomprimierung, die ähnlich, nur komplexer, funktioniert. Inzwischen gibt es 3 verschiedene Typen, Layer, die alle aufwärts kompatibel sind.
Layer 1 und 2 teilen das Signal in 32 Bänder auf. Jedes Band mit einer Weite von $f_x/64$. Erlaubte Sampling Frequenzen sind 32000, 44100 und 48000 Samples pro Sekunde.

7 Bedeutung für die Bildkomprimierung

Hier stellt sich die Frage, wie kann man mit Sequenzen, die zwei-dimensionale Abhängigkeiten, wie Bilder, enthalten umgehen? Die recht einfache Antwort ist, daß man nur zwei-dimensionale Filter braucht, die das Signal in die vertikale und die horizontale Komponente zerlegen. Meist kann man hierfür ein-dimensionale Filter verwenden, die man zuerst in die eine und dann in die andere Richtung anwendet. Zwei-dimensionale Filter mit dieser Eigenschaft nennt man *separable filter*. Es gibt aber auch zwei-dimensionale nicht separable Filter, die jedoch viel komplexer sind und deswegen selten Anwendung finden.

8 Zusammenfassung

- Allgemeines Verfahren der Subbandkodierung:
  - Suche eines geeigneten Filters zur Zerlegung des Quellsignals
  - Bei der Benutzung des ausgewählten Filters entstehen die einzelnen Subband Signale $\{ y_{k,n} \}$

$$y_{k,n} = \sum_{i=0}^{N-1} h_{k,i} x_{n-1}$$

wobei $\{ \ h_{k,i} \ \}$ die Koeffizienten des k-ten Filters sind

  - Dezimiere den Ausgang des Filters
  - Codiere den dezimierten Ausgang

- Das Decodierungsverfahren läuft analog dazu in inverser Form ab.