Filterbankstrukturen zur verlustfreien Kompression medizinischer Bilddaten

Andreas Klappenecker, Frank U. May und Thomas Beth

Universität Karlsruhe, IAKS, D-76128 Karlsruhe
wavelet@ira.uka.de

Zusammenfassung In der modernen medizinischen Diagnostik werden eine Vielzahl von bildgebenden Verfahren verwendet. Hierdurch fallen jährlich radiologische Bilddaten in der Größenordnung von mehreren Petabytes an. In Deutschland wird vom Gesetzgeber zur Archivierung nur eine *verlustlose* Kompression dieser Bilddaten erlaubt. Wir stellen zwei neue verlustlose Kompressionsverfahren vor. Beide Verfahren benutzen eine Filterbank zur Reduktion der Signalentropie und verwenden eine Erweiterung der arithmetischen Codierung zur Entfernung der Redundanz.

Schlüsselwörter: Verlustfreie Kompression, Filterbänke, modulare Arithmetik, nichtlineare Rundungsoperationen, Leiterstrukturen.

1 Einleitung

Die bekannten *verlustbehafteten* Kompressionsverfahren (wie z.B. JPEG) lassen sich häufig in die drei Schritte Signaltransformation, Quantisierung und Entropiecodierung unterteilen. Die ersten beiden Schritte reduzieren die Entropie der Eingabedaten und erhöhen damit die Effizienz der Entropiecodierung.

Bei *verlustfreien* Kompressionsverfahren kann ebenfalls eine Signaltransformation (ohne Quantisierung) zur Reduktion der Entropie benutzt werden. Wir stellen zwei verschiedene Verfahren vor, welche auf diesem Prinzip beruhen. Beide benutzen im ersten Schritt eine Filterbankstruktur [1] und im zweiten Schritt eine Erweiterung der arithmetischen Codierung. Die Verfahren unterscheiden sich in den Operationen, die zur Realisierung der Filterbank verwendet werden.

Wir demonstrieren den Nutzen dieser Kompressionsverfahren für CT- und MRT-Schichtbilder. Es handelt sich hierbei um monochromatische Bilder, die typischerweise eine Größe von 256×256 oder 512×512 Pixeln mit einer Tiefe von 12-16 Bit aufweisen.

2 Filterbänke mit modularer Arithmetik

Die Implementierung der Filterbank muß eine perfekte Rekonstruktion des Signals erlauben und soll – trotz exakter Rechnung – eine Explosion der transformierten Koeffizienten vermeiden. Dies legt eine Verwendung von modularer

Arithmetik nahe. Die Pixel der Bilddaten werden hierbei als Elemente aus einem Restklassenring $\mathbf{Z}/2^n\mathbf{Z}$ (etwa mit $n \in [12..16]$) interpretiert. Entsprechend werden in der Filterbank alle Additionen und Multiplikationen modulo 2^n gerechnet. Im folgenden untersuchen wir die Bedingungen, denen eine modulare Filterbank genügen muß, um eine perfekte Rekonstruktion zu garantieren.

Die Eingabedaten interpretieren wir als Elemente des Laurentpolynomrings in zwei Unbestimmten mit Koeffizienten aus dem Restklassenring $A := \mathbf{Z}/2^n\mathbf{Z}$. Aufgrund der Tensorproduktstruktur

$$A[x, x^{-1}] \otimes_A A[y, y^{-1}] \cong A[x, x^{-1}, y, y^{-1}]$$

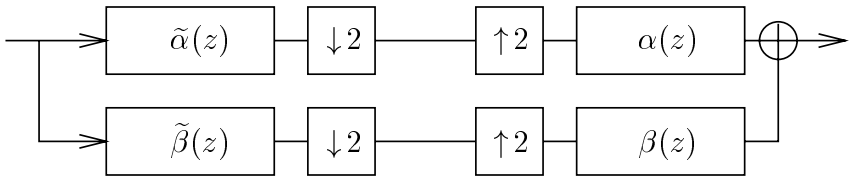
bietet sich eine zeilen- und spaltenweise Verarbeitung der Daten an.

Die Zeilen des Eingabebildes werden mit einer sogenannten Zweikanal-Filterbank horizontal in zwei Teilsignale zerlegt. Danach führen wir dasselbe Verfahren auf den Spalten der erhaltenen Teilbilder durch und erhalten so insgesamt vier Teilbilder. Dieser Zerlegungsschritt wird auf einem Teilbild rekursiv wiederholt. Der Datenfluß ist wie bei der schnellen Wavelettransformation [1, 2], lediglich die verwendeten Operationen unterscheiden sich. Die Funktionsweise der verwendeten Zweikanal-Filterbank erläutern wir in den nächsten beiden Abschnitten.

Analyse. Ein Eingabesignal $s(z) \in A[z, z^{-1}]$ wird mit den Filtern $\widetilde{\alpha}(z)$ und $\widetilde{\beta}(z)$ multipliziert (gefaltet). Danach wird die Abtastrate auf die Hälfte reduziert. Wir erhalten in diesem Schritt zwei Signale $d_\alpha(z) := [\downarrow 2]\,\widetilde{\alpha}(z)s(z)$ und $d_\beta(z) := [\downarrow 2]\,\widetilde{\beta}(z)s(z)$, wobei

$$[\downarrow 2] : \begin{cases} A[z, 1/z] \longrightarrow A[z, 1/z], \\ a(z) \quad \longmapsto \quad a_g(z), \end{cases} \qquad \text{mit} \qquad a(z) = a_g(z^2) + z a_u(z^2).$$

Diese Analysefilterbank ist in der linken Hälfte der folgenden Abbildung skizziert:



Synthese. Im Syntheseschritt wird die Abtastrate der Signale $d_\alpha(z)$ und $d_\beta(z)$ wieder erhöht, mit $\alpha(z)$ bzw. $\beta(z)$ multipliziert und addiert. Wir erhalten hierbei das Ausgabesignal

$$\widehat{s}(z) := \alpha(z)d_\alpha(z^2) + \beta(z)d_\beta(z^2). \tag{1}$$

Auf der rechten Seite der vorigen Abbildung ist diese Synthesefilterbank skizziert.

Perfekte Rekonstruktion. Eine Filterbank wird **perfekt rekonstruierend** genannt, wenn das Ausgabesignal $\widehat{s}(z)$ mit dem Eingabesignal $s(z)$ für alle Signale $s(z) \in A[z, z^{-1}]$ übereinstimmt.

Wenn wir sogenannte Polyphasenmatrizen [1]

$$H_p := \begin{pmatrix} \widetilde{\alpha}_g(z) & \widetilde{\alpha}_u(z) \\ \widetilde{\beta}_g(z) & \widetilde{\beta}_u(z) \end{pmatrix}, \qquad \text{mit} \qquad \begin{array}{l} \widetilde{\alpha}(z) = \widetilde{\alpha}_g(z^2) + z^{-1}\widetilde{\alpha}_u(z^2), \\ \widetilde{\beta}(z) = \widetilde{\beta}_g(z^2) + z^{-1}\widetilde{\beta}_u(z^2), \end{array}$$

und

$$G_p := \begin{pmatrix} \alpha_g(z) & \alpha_u(z) \\ \beta_g(z) & \beta_u(z) \end{pmatrix}, \qquad \text{mit} \qquad \begin{array}{l} \alpha(z) = \alpha_g(z^2) + z\alpha_u(z^2), \\ \beta(z) = \beta_g(z^2) + z\beta_u(z^2), \end{array}$$

definieren, dann erhalten wir mit einer kleinen Rechnung [3] die folgende Charakterisierung der perfekt rekonstruierenden Filterbänke:

Theorem 1. *Eine Zweikanal-Filterbank für Signale aus $A[z, z^{-1}]$ ist genau dann perfekt rekonstruierend, wenn die Polyphasenmatrizen H_p und G_p die Bedingung $G_p^t H_p = I$ erfüllen.*

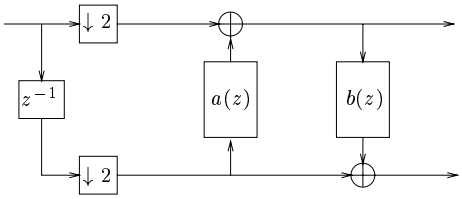
Diese Polyphasenmatrizen lassen sich vollständig parametrisieren [4, Kap. 4]:

Theorem 2. *Jede Polyphasenmatrix einer perfekt rekonstruierenden Zweikanal-Filterbank läßt sich als Produkt von elementaren Transvektionen und einer Diagonalmatrix aus $\mathrm{GL}_2(A[z, z^{-1}])$ schreiben.*
[Eine elementare Transvektion unterscheidet sich von der Identitätsmatrix durch einen Eintrag aus $A[z, z^{-1}]$ auf der Nebendiagonalen.]

3 Filterbänke mit Rundungsoperationen

In der klassischen Filterbanktheorie werden die Koeffizienten A der Signale und Filter als reelle Zahlen interpretiert. Es läßt sich zeigen, daß Theorem 1 und 2 auch für Signale und Filter aus $A[z, z^{-1}] = \mathbf{R}[z, z^{-1}]$ gelten [3]. Nachteilig ist jedoch, daß die Transformationskoeffizienten im allgemeinen mit einer erheblich größeren Genauigkeit als die reellen Signalwerte dargestellt werden müssen, sofern die Filterbank-Implementierung eine perfekte Rekonstruktion des Signals erlauben soll.

Eine andere Vorgehensweise wird durch Theorem 2 (mit $A := \mathbf{R}$) nahegelegt. Die elementaren Transvektionen entsprechen bei einer Polyphasen-Filterbank [1, S. 259] direkt einem Leiterschritt (oder *lifting step* in der Terminologie von Sweldens). Ein kleines Beispiel soll dieses Prinzip verdeutlichen:



Die Struktur der Filterbank zeigt, daß man perfekte Rekonstruktion erreichen kann, indem die Operatoren $a(z)$ und $b(z)$ mit negativem Vorzeichen in umgekehrter Reihenfolge angewendet werden. Eine naheliegende Verallgemeinerung dieser Filterbankstruktur ist, statt der Faltungsoperatoren beliebige Operationen (wie Rundungsoperationen) zuzulassen. Die perfekte Rekonstruktion wird bereits durch die Struktur der Filterbank garantiert [5].

4 Arithmetische Codierung

Die Redundanz in den Bilddaten wird erst durch die Entropiecodierung entfernt. Es erweist sich als vorteilhaft, die Koeffizienten nach einer der vorgestellten Transformationen zu codieren, anstatt die Codierung direkt auf die Bilddaten anzuwenden.

Die arithmetische Codierung ist eine optimale verlustfreie Entropiecodierung. Sie basiert auf einer Idee von Elias, Symbole auf Teilintervalle von $[0,1)$ abzubilden, wobei die Intervallbreite proportional zur Auftrittswahrscheinlichkeit des Symbols gewählt wird. Eine Symbolfolge wird gliedweise codiert, wobei jedes codierte Symbol das Ausgangsintervall $[0,1)$ weiter verkleinert. Die Intervallbreite entspricht nach jedem Schritt der Wahrscheinlichkeit des bis dahin codierten Folgenpräfixes. Der Codierer überträgt dann eine beliebige Zahl aus dem letzten Intervall. Der Decodierer kann anhand dieser Zahl die Operationen des Codierers nachvollziehen, und dabei die Symbolfolge gliedweise rekonstruieren.

Abgesehen von der Einschränkung, daß Codierer und Decodierer dasselbe Modell benutzen müssen (d. h. zur Codierung und Decodierung eines Symbols muß jeweils dieselbe Wahrscheinlichkeitsverteilung benutzt werden), sind die Verteilungen in jedem Schritt frei wählbar. Ein adaptiver arithmetischer Codierer wird realisiert, indem man nach der (De-)Codierung eines Symbols das Modell mit einem vorgegebenen Algorithmus anpaßt.

Die Flexibilität des Modells erlaubt es, die Abhängigkeiten zwischen benachbarten Transformationskoeffizienten zu berücksichtigen. Dazu berechnet man aus den bereits codierten Koeffizienten einen *Codierungskontext*, der die tatsächlich benutzte Wahrscheinlichkeitsverteilung aus einer vorgegebenen Menge auswählt. Das zugehörige Modell enthält für jeden möglichen Kontext eine eigene Wahrscheinlichkeitsverteilung. Die einzelnen Verteilungen werden unabhängig voneinander adaptiert.

Eine adaptive Codierung arbeitet erst dann wirklich effizient, wenn das Modell die tatsächliche Statistik der Koeffizienten ungefähr widerspiegelt. Das ist zu Beginn der Codierung jedoch meist nicht der Fall. Die Adaption dauert umso länger, je mehr unterschiedliche Symbole und Codierungskontexte das Modell enthält, denn die Anzahl der adaptierten Wahrscheinlichkeiten im Modell entspricht gerade dem Produkt aus der Anzahl der Symbole und der Anzahl der Kontexte. Für eine effiziente Codierung muß man sich also auf eine möglichst kleine Anzahl von Kontexten beschränken.

Um einerseits möglichst viele benachbarte Koeffizienten berücksichtigen zu können, andererseits aber die Anzahl der Kontexte möglichst klein zu halten, teilt man den Wertebereich der Koeffizienten in mehrere Klassen (sogenannte *buckets*) ein und berücksichtigt bei der Berechnung des Codierungskontexts nur die Klassenzugehörigkeit der benachbarten Koeffizenten.

5 Erste Ergebnisse

In der folgenden Tabelle werden unsere beiden Verfahren (Modular und Rundung) mit bekannten Standardkompressionsverfahren für ein MRT- und ein CT-

Bild mit jeweils 512×512 Pixeln und 12 Bit Tiefe verglichen. Die Tabelle zeigt das Verhältnis von der Größe der Rohdaten (384 KByte) zur komprimierten Dateigröße und die Bits pro Pixel. Mit den weitverbreiteten universellen Kompressionsverfahren GZIP und BZIP2 wurden die Bilddaten (ohne *header*-Information) komprimiert. Hierfür wurden zwei Dateiformate verwendet: eine Datei enthielt die Bilddaten in gepackter 12-Bit-Form, bei der anderen wurde jeder 12-Bit-Wert in zwei Bytes eingebettet (16 Bit). Das GIF-Bildformat ist lediglich für Bilder mit bis zu 8 Bit Tiefe definiert. Wir haben zum Vergleich das Nachfolgeformat PNG herangezogen, das deutlich besser komprimiert als GIF und auch für Bilder mit 12 Bit Tiefe definiert ist. Leider stand uns keine Implementierung des „lossless JPEG"-Standards oder des kommenden JPEG-LS-Standards zur Verfügung, die unsere 12 Bit Bilder korrekt komprimieren konnte.

	CT			MRT		
Verfahren	Rate	BpP	Größe (KB)	Rate	BpP	Größe (KB)
GZIP 16 Bit	1.042	11.519	368.62	1.041	11.526	368.83
GZIP 12 Bit	1.096	10.945	350.25	1.089	11.021	352.66
BZIP2 12 Bit	1.202	9.981	319.39	1.247	9.619	307.82
BZIP2 16 Bit	1.365	8.794	281.40	1.453	8.262	264.37
PNG	1.368	8.772	280.69	1.489	8.061	257.97
Modular	1.398	8.583	274.64	1.546	7.764	248.43
Rundung	1.519	7.902	252.87	1.753	6.845	219.04

Aus der Tabelle wird ersichtlich, daß die vorgeschlagenen Kompressionsverfahren im Vergleich zu den Standardverfahren deutlich besser abschneiden. Die Parametrisierung der Codierung und der Filter bietet noch viele Möglichkeiten zur weiteren Optimierung der Verfahren.

Danksagung. Wir danken der Deutschen Forschungsgemeinschaft für die Unterstützung durch den Sonderforschungsbereich SFB 414 „Informationstechnik in der Medizin – Rechner und sensorgestützte Chirurgie" (Projekte Q1 und Q6).

Literatur

1. N. Fliege: *Multiraten-Signalverarbeitung.* B. G. Teubner, Stuttgart, 1993.
2. A. K. Louis, P. Maaß und A. Rieder: *Wavelets – Theorie und Anwendungen.* B.G. Teubner, Stuttgart, 1994.
3. A. Klappenecker, A. Nückel und F. U. May: *Lossless image compression using wavelets over finite rings and related architectures.* In: A. Aldroubi, A. F. Laine und M. A. Unser (Herausgeber): *Wavelet Applications in Signal and Image Processing V*, Band 3169, Seiten 139–147. SPIE, 1997.
4. A. Klappenecker: *Algebraische Wavelets.* (In Vorbereitung), Universität Karlsruhe, 1998.
5. A. R. Calderbank, I. Daubechies, W. Sweldens und B. L. Yeo: *Wavelet transforms that map integer to integers.* Erscheint in ACHA, 1996.