Statistische Untersuchung blockbasierter Langzeitprädiktion

Diplomarbeit von

Xiaozheng Zhang

Betreuer: Dipl. -Ing. Thomas Wiegand

Hochschullehrer: Prof. Dr. -Ing. Bernd Girod

Lehrstuhl für Nachrichtentechnik Universität Erlangen-Nürnberg

Dezember 1996

Erklärung

Ich versichere, daß ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und daß die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 6.12.1996

Xiaozheng Zhang

Adresse: Walter-Flex-Str. 1/Zi.363, 91052 Erlangen

Inhaltsverzeichnis

1	Einleitung	5
2	 Blockbasierte Bewegungsschätzung mit Langzeitprädiktion 2.1 Block-Matching-Verfahren	8 8 10 13
3	 Codierung der Bewegungsparameter bei der Langzeitprädiktion 3.1 Das Prinzip der Vektorquantisierung mit Entropie-Nebenbedingung und der iterative Entwurfsalgorithmus	16 16 19 24 24 36 36 36 41
4	Gemeinsame Codierung der drei Bewegungsparameter	45
5	Vergleich von Langzeitprädiktion zur blockbasierten Bewegungskom pensation mit 8×8 Blöcke	- 50
6	Zusammenfassung und Ausblick	52

Kurzfassung

Die vorliegende Arbeit behandelt eine neue Methode zur blockbasierten bewegungskompensierten Prädiktion: die bewegungskompensierte Langzeitprädiktion für die Anwendung zur Übertragung von Bewegtbildsequenzen mit niedrigen Bitraten.

Zunächst wird das Modell der bewegungskompensierten Langzeitprädiktion vorgestellt und die dadurch verbesserte Bewegungskompensation anhand von PSNR-Werten für ausgewählte Testsequenzen dargestellt. Die Wirksamkeit der Langzeitprädiktion wird für unterschiedliche Bildsequenzen, Bildwiederholraten und Langzeitspeichergrößen gezeigt.

Anschließend wird die bewegungskompensierte Langzeitprädiktion mit Vektorquantisierung mit Entropie-Nebenbedingung vereint betrachtet und das Entwurfskonzept für die Langzeitprädiktion aufbauend auf dem aus ECVQ bekannten iterativen Algorithmus entwickelt. Umfangreiche Untersuchungen der verschiedenen Codierverfahren, u.a. verschiedene lineare und nichtlineare Prädiktionsmodelle und der Finite-State-Mechanismus für die Codierung des Zeitparameters, werden durchgeführt, um die Codierverfahren im Rate-Distortion-Sinn zu optimieren. Die Codierergebnisse der verschiedenen Verfahren werden präsentiert und miteinander verglichen.

Zum Schluß wird ein Vergleich der Codierergebnisse der bewegungskompensierten Langzeitprädiktion mit anderen Bewegungsschätzverfahren, wie die Bewegungsschätzung mit unmittelbar vorangegangenem Bild und die Bewegungsschätzung mit der Blockgröße 8×8 , gegeben. Die Ergebnisse zeigen, daß das vorgestellte Bewegungsschätzmodell mit Langzeitprädiktion im Rate-Distortion-Sinn bessere Resultate erzielt.

Die Implementierungen finden in der Programmiersprache C unter Verwendung der CIDS (Common_image_data_structure)-Library statt.

Liste der verwendeten Formelzeichen

$E\{\cdot\}$:	Erwartungswertoperator
$r_{xx}(m)$:	Autokorrelationsfolge von $x(n), r_{xx} = E\{x^*(n) \cdot x(n+m)\}$
$r_{xy}(m)$:	Kreuzkorrelationsfolge von $x(n)$ und $y(n), r_{xy} = E\{x^*(n) \cdot y(n+m)\}$
$min\{\cdot\}$:	Minimalwertoperator
€:	Element von
[]:	beidseits geschlossenes Intervall
\mathbf{a}^T :	Transponierter Vektor
\mathbf{A}^T :	Transponierte Matrix
A^{-1} :	Inverse Matrix
$k_1(\Delta k)k_2$:	Variation einer diskreten Variablen in Schritt Δk von k_1 bis k_2
$ld(\cdot)$:	Logarithmus mit Basis 2
$ \operatorname*{arg\ min}_{a \in \mathbf{A}} $	Argument des Minimalwertes aus Menge ${\bf A}$
$MED(\cdot):$	Medianfilterung

4

Kapitel 1 Einleitung

Videocodierung für niedrige Bitraten findet Anwendung z.B. in der Bildtelefonie, bei Videokonferenzen und verschiedenen interaktiven mobilen multimedia Kommunikationssystemen. Um die Übertragung der Bildsignale, die eine hohe Bitrate von mehreren Megabit pro Sekunde besitzen, über z.B. herkömmliche analoge Telefonleitungen von einigen Kilobit pro Sekunde zu ermöglichen, ist eine hohe Kompressionsrate erforderlich.

Die Datenkompressionsalgorithmen basieren auf der Tatsache, daß Bildsignale innerhalb einer Bildsequenz sowohl eine hohe örtliche als auch eine hohe zeitliche statistische Abhängigkeit aufweisen. Es ist daher möglich, bei Ausnutzung dieser Redundanz, das Bildsignal mit einer geringeren Datenrate zu übertragen.

Die derzeit standardisierten Videocodierverfahren für niedrige Bitraten beruhen auf dem Prinzip der "Hybridcodierung"- eine Kombination aus bewegungskompensierter Prädiktion und Transformationscodierung des Prädiktionsfehlerbildes. Eine der am häufigsten verwendeten Verfahren zur Bewegungskompensation stellt die blockbasierte Methode mit translatorischem Bewegungsmodell dar. Dabei wird zuerst ein Bild in Blöcke konstanter Größe unterteilt. Für jeden Block des zu codierenden Bildes wird eine temporale Prädiktion mit Hilfe des zeitlich unmittelbar vorangegangenen Bildes vorgenommen. Da die Änderungen zwischen Bildern einer Folgesequenz meistens durch die Bewegung der Objekte oder der Kamera verursacht werden, läßt sich mit Hilfe der bewegungskompensierten Prädiktion eine große Reduktion der Datenrate erzielen.

Bisher wurde in diesem Verfahren nur das zuletzt decodierete Bild zur Prädiktion herangezogen, dessen Verschiebung zum aktuellen Bild anhand des translatorischen Bewegungsmodells mit je zwei diskreten Bewegungsvektoren in horizontale und vertikale Richtung bestimmt wird. Jedoch wird diese Bewegungsschätzung ungenauer bei geringen Bildfolgefrequenzen, die bei niedrigen Bitraten auftreten. Die Bewegungen können nicht exakt durch die Verschiebung zweier Bilder kompensiert werden, da die Ähnlichkeit zwischen aufeinanderfolgenden Bildern gering ist. Um dieses Problem zu umgehen, wird in der vorliegenden Diplomarbeit ein Codierverfahren mit bewegungskompensierter Langzeitprädiktion entwickelt.

In einem Langzeitspeicher werden mehrere vorangegangene Bilder aufbewahrt, die als Prädiktionssignal zur Verfügung stehen. Durch Ausnutzung der Abhängigkeit mehrerer aufeinanderfolgender Bilder im Langzeitspeicher, im Gegensatz zu dem Fall, bei dem nur das vorangegangene Bild bei der Codierung des aktuellen Bildes berücksichtigt wird, ist eine verbesserte bewegungskompensierte Prädiktion zu erwarten. Ausgehend von vorangegangenen Bildern läßt sich die Prädiktion für sich wiederholenden Bildinhalt in Bildfolgesequenzen, bei Verdeckung und Aufdeckung zwischen Objekten und Hintergrund, sowie beim Hin- und Herschwenken der Kamera, sehr gut durchführen.

Die zusätzliche Information für die Addressierung im Langzeitspeicher erhöht andererseits aber die Bitrate für die Bewegungsparameter, die besonders bei der Codierung mit niedrigen Bitraten beschränkt ist. Es ist daher ein Kompromiß zwischen einem verringerten Prädiktionsfehler D und der zur Übertragung notwendigen Rate R zu finden. Das Hauptziel der Codierung läßt sich daher so formulieren, daß der mittlere Prädiktionsfehler Dzwischen Original- und rekonstruiertem Bildsignal bei Vorgabe einer Gesamtbitrate R^* möglichst gering gehalten wird

$$\min \{D\} \qquad \text{mit der Nebenbedingung} \quad R < R^*, \tag{1.1}$$

wobe
i R^{\ast} die maximale erlaubte Gesamtbitrate darstellt.

Die untere Grenze der Gesamtbitrate R ist hierbei durch eine optimale Rate-Distortion-Kurve R(D) gegeben. Eine typische minimale Rate-Distortion-Kurve ist in Abb. 1 gezeigt. Das Minimierungsproblem mit Nebenbedingung aus Gleichung 1.1 läßt sich in eine Minimierung der Lagrange-Kostenfunktion

$$J = D + \lambda \cdot R \tag{1.2}$$

überführen. Hierbei ist λ der Lagrange-Multiplikator, der das relative Gewicht zwischen R und D angibt. Durch Variation von λ werden verschiedene Punkte an der optimalen Rate-Distortion-Kurve erreicht, die zur Tangente $R = -\frac{1}{\lambda}D + k$ korrespondieren.



Abbildung 1.1: Rate-Distortion-Funktion

Shannon's Quellencodierungstheorem besagt, daß zwischen einer gewünschten Genauigkeit der Approximation eines Quellensignals und der dafür minimal zur Übertragung notwendigen Bitrate ein Zusammenhang existiert, und diese minimal notwendige Bitrate R(D) beliebig nahe approximiert werden kann, wenn die Signale zu Vektoren zusammengefaßt und gemeinsam codiert werden. Dieses bietet die theoretische Grundlage für die Anwendung der Vektorquantisierung, die sich als eine effiziente Methode zur Datenkomprimierung erweist und derzeit in der Bildcodierung eine immer größere Rolle spielt. Bei der Vektorquantisierung werden mehrere Quellensymbole zusammengefaßt, für die ein gemeinsames Codesymbol zur Übertragung gefunden wird. Würde bei der Auswahl des Codesymbols neben der Minimierung der Verzerrung D auch die zu übertragende Rate berücksichtigt, so könnte eine bessere Approximation von R(D) erzielt werden. Die auf diesem Prinzip basierende Methode mit gleichzeitiger Entropieminimierung wird als Entropy-constrained vector quantization (ECVQ) [CLG89] bezeichnet. Es ist zu beachten, daß diese ebenfalls mit Gleichung 1.2 formuliert werden kann.

In der vorliegenden Arbeit wird die bewegungskompensierte Langzeitprädiktion im Kontext von Vektorquantisierung mit Entropie-Nebenbedingung betrachtet, und das Codierverfahren so entwickelt, daß es im Sinne der Rate-Distortion-Theorie optimal ist, d.h. eine Optimierung der Prädiktion durch eine gemeinsame Minimierung von D und R mit Hilfe der Vektorquantisierung durchgeführt wird.

Im Rahmen dieser Arbeit wird nur die bewegungskompensierte Prädiktion berücksichtigt und auf die Codierung des Prädiktionsfehlers verzichtet. Aus Rechenzeitgründen wird nur die Helligkeitskomponente des Bildsignals berücksichtigt.

Kapitel 2

Blockbasierte Bewegungsschätzung mit Langzeitprädiktion

2.1 Block-Matching-Verfahren

Das Block-Matching-Verfahren ist das am häufigsten verwendete Verfahren für die Bewegungsschätzung. Wegen seines einfachen und intuitiven Konzeptes ist es vielen anderen komplizierteren Verfahren im Codierergebnis überlegen.

Beim Block-Matching-Verfahren wird ein Bild zuerst in quadratische Blöcke der Größe $N \times N$ Abtastwerte zerlegt. Unter der Annahme des translatorisch bewegten Bewegungsmodelles (alle Pixel innerhalb des Blocks haben die gleiche Bewegung, und zwar paralell zur Bildebene), wird für jeden Block eine horizontale und eine vertikale Verschiebung (v_x, v_y) bestimmt, die auf alle Bildpunkte des Blocks angewendet wird. Für jeden Block $S_k(x, y)$ im aktuellen Bild wird dann ein Block im Vorgängerbild $S_{k-1}(x + v_x, y + v_y)$ innerhalb eines zugelassenen Suchbereiches Ω mit bestem Match gefunden, der eine bestimmte Zielfunktion minimiert. $S_k(x, y)$ stellt hierbei das Luminanzsignal an der Position (x, y) dar mit k als Zeitindex. Verschiedene Zielfunktionen sind möglich. Häufig wird als Verzerrungsmaß Mean Square Error (MSE) verwendet, welches wie folgt definiert ist:

$$D_{\text{MSE}}(\mathbf{v}) = \sum_{\mathbf{r} \in \text{Block}} [S_k(\mathbf{r}) - S_{k-1}(\mathbf{r} + \mathbf{v})]^2$$
(2.1)

mit $\mathbf{r} = (x, y), \mathbf{v} = (v_x, v_y)$ und $\mathbf{v} \in \Omega = [-u, w] \times [-u, w].$

Wie in der Einleitung schon beschrieben, wird im Rahmen dieser Arbeit anstatt der Minimierung der Verzerrung, Rate R und Verzerrung D gemeinsam minimiert. Daher wird die obige Zielfunktion noch durch einen Term erweitert, der die Rate berücksichtigt:

$$J(\lambda) = D(\mathbf{v}) + \lambda \cdot R(\mathbf{v}) \tag{2.2}$$

 $R(\mathbf{v})$ stellt hierbei die für die Übertragung der Bewegungsparameter benötigte Rate dar. Der zu einer bestimmten Rate korrespondierende Lagrange-Multiplikator λ ist für eine bestimmte Videosequenz abhängig von der Rate-Distortion-Kurve. Durch ein großes λ wird im Lagrange-Funktional die Rate stärker gewichtet und die optimale Lösung hat eine relativ geringe Rate und große Verzerrung. Umgekehrt hat die optimale Lösung eine große Rate und eine geringe Verzerrung. In dem speziellen Fall mit $\lambda = 0$ entartet dies zur Optimierung der Verzerrung ohne Beschränkung der Rate.

Der optimale Bewegungsparameter wird demnach gefunden, indem die Zielfunktion für jede mögliche Position des Blocks innerhalb des Suchgebietes überprüft und das Minimum wie folgt gewählt wird

$$\mathbf{v} = \underset{\mathbf{v}\in\Omega}{\arg\min}\{D(\mathbf{v}) + \lambda \cdot R(\mathbf{v})\}.$$
(2.3)

In unserer Implementierung wird eine sog. "Spiralsuche" vorgenommen. Dabei werden die Suchpositionen spiralförmig von der Mitte des Suchgebietes ausgehend untersucht. Die Berechnung der Zielfunktion erfolgt zeilenweise und wird abgebrochen, sobald die akkumulierte Lagrange-Summe das bisherige Optimum überschritten hat. Der Gedanke dahinter ist die Vorstellung, daß das Optimum häufig in der Nähe des Nullvektors liegt und demzufolge die Berechnung bei allen weiteren Suchpositionen frühzeitig abgebrochen werden kann.

Die Arbeitsweise der Bewegungsschätzung ist in Abb. 2.1 veranschaulicht dargestellt.



Abbildung 2.1: Blockbasierte Bewegungsschätzung

Die Genauigkeit der Bewegungsschätzung wird durch die folgenden Parameter beeinflußt:

- **Block Größe N:** Bei kleiner Blockgröße arbeitet das Verfahren aufgrund der Rauscheinflüsse unzuverlässig, außerdem tritt das Bildfensterproblem auf. Bei einem zu großen Block wird die Bewegungsschätzung bei stark bewegten und stark detaillierten Bildbereichen ungenauer, da die Pixel innerhalb des Blocks keine einheitliche Bewegung mehr aufweisen.
- Suchbereich Ω : Dieser legt die Größe der maximal möglichen und noch schätzbaren Bewegung fest, $\mathbf{v} \in \Omega$, bzw. $v_x, v_y \in [-u, w]$. Während ein zu kleiner Suchbereich

einer schnellen und großen Bewegung nicht folgen kann und daher zu einer schlechten Prädiktion führt, ergibt ein großer Suchbereich bei kleiner Bewegung eine unnötige Komplexität.

Suchschrittweite: Diese gibt an, wie v_x und v_y während der Suche inkrementiert werden. Die Sub-Pel-Suche wird gegenüber der Integer-Pel-Suche wegen der besseren Genauigkeit bevorzugt. Aufgrund ihrer großen Komplexität wird in dieser Arbeit jedoch auf Sub-Pel-Suche verzichtet.

In dieser Arbeit wird das QCIF(quater common intermediate format)-Format mit 176 × 144 Pixel für die Luminanz pro Bild verwendet und die Blockgröße auf 16 × 16 Pixel gesetzt. Entsprechend dem H.261-Standard [ITU93] (Internationaler Standard für Bild-telefone bzw. für Videokonferenzsysteme) werden die maximal zulässigen Verschiebungen v_x und v_y auf den Bereich [-16, 15] begrenzt und die Suchschrittweite auf ganze Pixelgenauigkeit gesetzt.

2.2 Einbauen eines Langzeitspeichers in Bewegungsschätzverfahren

In dem vorgestellten Block-Matching-Verfahren wird das unmittelbar vorangegangene Bild für die bewegungskompensierte Prädiktion herangezogen. Jedoch ist der Ansatz des translatorischen Bewegungsmodells bei dem verwendeten Verfahren für niedrige Bitraten aufgrund der großen Abweichung zwischen aufeinanderfolgenden Bildern problematisch. Vielversprechend wäre es, wenn mehrere Vorgängerbilder für die Prädiktion herangezogen würden. Der dadurch vergrößerte Suchraum ermöglicht ein besseres Matching des Blocks, wenn der Bildinhalt sich bei einem der Vorgängerbilder im Suchraum verschobend wieder findet.

Daher wird mit der Langzeitprädiktion ein neues Verfahren untersucht, bei der mehrere Vorgängerbilder zur Prädiktion herangezogen werden. Eine verbesserte bewegungskompensierte Langzeitprädiktion im Vergleich zur Bewegungskompensation mit dem unmittelbar vorhergehenden Bild ist zu erwarten.

Um die vorangegangenen Bilder aufzubewahren, wird ein Langzeitspeicher mit einer ringförmigen Struktur verwendet. Der Langzeitspeicher wird am Anfang mit MVorgängern des aktuellen Bildes initialisiert. Mit Hilfe der Ringstruktur läßt sich die Aktuallisierung des Langzeitspeichers der jeweiligen nachfolgenden Bilder durch das Ersetzen des zu aktualisierenden Bildes an Stelle des aus dem Zeitfenster herausfallende Bildes realisieren. Wie in Abb. 2.2 gezeigt, stehen M Vorgängerbilder F_{k-i} , $(i = 1, 2, \dots, M)$ hintereinander im Ringbuffer. Die Suche fängt mit dem unmittelbaren Vorgänger des aktuellen Bildes an, auf den der Pfeil zeigt, und folgt in umgekehrter zeitlicher Richtung, d.h. der Pfeil bewegt sich immer in negativer zeitlicher Richtung und wird auf das Ende des Ringbuffers gesetzt, wenn der Anfang erreicht wird. Nach dem Durchsuchen aller Bilder im Ringbuffer wird das am weitesten zurückliegende Bild durch das aktuelle Bild ersetzt. Dadurch ist eine kontinuierliche Suche von Bild zu Bild mit Hilfe der Ringstruktur



Abbildung 2.2: Langzeitspeicher mit Ringstruktur

möglich.

Der Langzeitspeicher ist dementsprechend charakterisiert durch einen 3-dim. Suchraum $\Omega = \{v_x, v_y, v_t\}, v_x, v_y \in [-u, w], v_t \in [1, M]$, wobei der zusätzliche Zeitparameter v_t eine Zeitverzögerung darstellt, und M die Langzeitspeichergröße bedeutet. Die Bewegungsschätzung erfolgt durch das beschriebene Block-Matching-Verfahren, angewendet für alle Bilder im Langzeitspeicher. Die Lagrange-Kostenfunktion in Gleichung 2.2 ist erweitert durch

$$J = D(\mathbf{v}) + \lambda \cdot R(\mathbf{v}) \tag{2.4}$$

 mit

$$D(\mathbf{v}) = \sum_{x,y \in \text{Block}} [S_k(x,y) - S_{k-v_t}(x+v_x,y+v_y)]^2,$$
(2.5)

Dabei ist $\mathbf{v} = (v_x, v_y, v_t).$

Den Ablauf der Bewegungsschätzung mit Langzeitprädiktion soll das folgende Struktogramm verdeutlichen.

Struktogramm — für das Programm zur Bewegungsschätzung mit Langzeitprädiktion

```
 \begin{array}{l} \mbox{Gegeben: die Langzeitspeichergröße $M$,} \\ \mbox{der Langrange-Multiplikator $\lambda$,} \\ \mbox{die zu codierende Bildsequenzlänge $L$;} \\ \mbox{Initialisieren: $k=0, $J^* = MAX, $MSE = 0$} \end{array}
```

Initialisieren des Ringbuffers mit den Bildern S_{k-M}, \cdots, S_{k-1} , und Erzeugen des entsprechenden Suchbereichs nach UMV-Mode¹

While (k < L)

1:Im UMV-Mode (Unrestricted Motion Vector Mode) dürfen die Bewegungsvektoren für die Prädiktion auch außerhalb des codierten Bildbereiches liegen. Dabei wird für die Prädiktion der außerhalb liegenden Pixel ein Kantenpixel verwendet, der durch die Begrenzung des Bewegungsvektors auf die letzte Pixelposition innerhalb des codierten Bildbereiches ermittelt wird.

2.3 Beurteilung der Langzeitprädiktion mit dem PSNR-Maß

Um eine Aussage über die vorgeschlagene Methode der bewegungskompensierten Prädiktion mit Langzeitspeicher zu machen, sollen nun die PSNR-Werte der dadurch codierten Bilder betrachtet werden.

Der PSNR (peak signal to noise ratio)-Wert ist ein häufig verwendetes Maß für die objektive Beurteilung der Bildqualität. Er ist definiert durch

$$PSNR \ [dB] = 10 \cdot \log_{10} \frac{A^2}{D}.$$
(2.6)

A stellt die maximal mögliche Amplitude dar und ist gleich 255 bei 8 bit Auflösung. D ist der mittlere quadratische Fehler des Luminanzsignals zwischen dem Orignalbild und dem rekonstruierten Bild, der nach Gleichung 2.5 definiert ist. Dementsprechend korrespondiert ein großer PSNR-Wert mit einem kleinen mittleren quadratischen Fehler.

Zuerst wird die bewegungskompensierte Langzeitprädiktion an zwei Sequenzen, Foremanund Mother-and-Daughter-Sequenz, die in der ganzen Arbeit als Testsequenzen verwendet werden, vorgenommen und die Codiergebnisse miteinander verglichen. Dabei wird nur die Minimierung der Verzerrung berücksichtigt, d.h. eine Optimierung der Zielfunktion in Gleichung 2.4 mit $\lambda = 0$ durchgeführt.

In Abb. 2.3 und 2.4 sind die PSNR-Werte für die verschiedenen Langzeitspeichergrößen über unterschiedlichen Frameraten eingetragen. Die Kurven stehen von oben bis unten jeweils für die abnehmenden Langzeitspeichergrößen von 8 bis 1. Wie erwartet, erzielt die bewegungskompensierte Langzeitprädiktion ein besseres PSNR-Ergebnis als die konventionelle Bewegungsschätzung nur mit dem unmittelbar vorhergehenden Bild. Diese PSNR-Werte ergeben einen zunehmenden Gewinn bei niedrigeren Frameraten. Der Gewinn verringert sich jedoch mit dem Zuwachs der Langzeitspeichergröße. Auch ist zu beachten, daß der Langzeitspeicher bei der Foreman-Sequenz mit mehreren schwankenden Bildinhalten vorteilhafter ist als bei der Mother-and-Daughter-Sequenz, bei der ein großer Teil des Bildes aus dem ruhenden Hintergrund besteht.

Die Abhängigkeit zwischen der PSNR-Verbesserung und der Langzeitspeichergröße zeigt Abb. 2.5. Dabei sind 10 Trainingssequenzen bei einer Bildfolgefrequenz von 10 Bildern pro Sekunde verwendet worden. Die eingetragenen PSNR-Werte sind die gemittelten Werten von Frame 500 bis 550, jeweils für die Langzeitspeichergröße von 1 bis 50. Man sieht eine große Steigung der Kurve am Anfang, die jedoch mit der erhöhten Langzeitspeichergröße verringert wird. Während durch die Verwendung des Langzeitspeichers von 10 die PSNR-Verbesserung von 1.8 dB gegenüber die Prädiktion mit dem unmittelbar vorangegangenen Bild erreicht wird, bringt eine weitere Erhöhung der Langzeitspeichergröße bis zum 50 eine Verbesserung um nur 0.5 dB auf 2.3 dB.



Abbildung 2.3: Codierergebnis mit Langzeitspeicher für Foreman-Sequenz



Abbildung 2.4: Codierergebnis mit Langzeitspeicher für Mother-and-Daughter-Sequenz



Abbildung 2.5: PSNR vs. Langzeitspeichergröße

Kapitel 3

Codierung der Bewegungsparameter bei der Langzeitprädiktion

Im letzten Kapitel wurde die bewegungskompensierte Bewegungsschätzung mit Langzeitprädiktion vorgestellt und die erzielbaren Codierergebnisse anhand von PSNR-Werten demonstriert. Dabei wurde jedoch die Erhöhung der Bitrate verschwiegen, die bei Verwendung des Langzeitspeichers auftritt, da der zur Adressierung im Langzeitspeicher benötigte Zeitparameter als Seiteninformation übertragen werden muß. Die zentrale Frage ist daher, ob die erhöhte Bitrate für die verbesserte Bewegungskompensation gerechtfertig ist, da die Bewegungskompensation mit dem Langzeitspeicher erst dann der konventionellen Bewegungskompensation überlegen ist, wenn sie im Rate-Distortion-Sinn eine Verbesserung erzielt. Dafür ist eine effiziente Codierung der Bewegungsparameter notwendig. Damit beschäftigen sich die folgenden Abschnitte.

Zuerst wird der iterative Algorithmus der Vektorquantisierung mit Entropie-Nebenbedingung vorgestellt und der Zusammenhang zwischen Vektorquantisierung und der Bewegungsschätzung mit Langzeitprädiktion gezeigt. Das Rate-Distortion-Verhalten der Codierung unter Verwendung des genannten Algorithmus wird präsentiert. Anschließend werden durch die genaue Untersuchung der Bewegungsparameter verschiedene Prädiktionsmodelle vorgeschlagen, um die Codierung der Bewegungsparameter zu optimieren. Das Rate-Distortion-Verhalten verschiedener Prädiktionsmodelle wird dargestellt und miteinander verglichen.

3.1 Das Prinzip der Vektorquantisierung mit Entropie-Nebenbedingung und der iterative Entwurfsalgorithmus

Vektorquantisierung mit Entropie-Nebenbedingung weist ein gutes Rate-Distortion-Verhalten auf und stellt sich daher als eine effiziente Datenkompressionsmethode dar [CLG89]. Bevor wir die Bewegungsschätzung mit der Vektorquantisierung im Zusammenhang betrachten, wird das Prinzip der Vektorquantisierung kurz erläutert.



Abbildung 3.1: Blockschaltbild eines Vektorquantisierungssystems

Abb. 3.1 zeigt ein Blockschaltbild zur digitalen Nachrichtenübertragung unter der Verwendung eines Vektorquantisierungssystems. Zuerst werden k Quellensymbole x(i) zu kdimensionalen Vektoren $\mathbf{x}_n = [x(nk), x(nk+1), \cdots, x(nk+k-1)]^T, n = 0, 1, 2, \cdots$ zusammengefaßt. Die \mathbf{x}_n bilden dann Eingangsvektoren des Coders α , der über ein Codebuch bestehend aus J k-dimensionalen Codewörtern $\mathbf{y}_j = [y_j(0), y_j(1), \cdots, y_j(k-1)]^T, j = 1(1)J$, verfügt. Der Prozeß $\alpha(\cdot)$ ordnet \mathbf{x}_n ein Codewort \mathbf{y}_j aus dem Codebuch mit dem Index jzu, der von der anschließenden Zuordnung von Codes variabler Länge $\gamma(\cdot)$ entropiecodiert wird. Die dadurch erzeugten binären Kanalsymbole $U_n, n = 0, 1, 2 \cdots$ mit variabler Länge werden über den Kanal übertragen. Auf der Decodiererseite wird durch $\gamma^{-1}(\cdot)$ (Inverseoperation zu $\gamma(\cdot)$) jedes Kanalsymbol \hat{U}_n zurück in den Index zugeordnet. Durch die Zuordnung $\beta(\cdot)$, die über ein identisches Codebuch wie $\alpha(\cdot)$ auf der Codiererseite verfügt, werden schließlich die Reproduktionsvektoren $\hat{\mathbf{X}}_n$ mittels einfacher Tabellendecodierung anhand von Indizes gewonnen.

Das Ziel beim Entwurf solcher Vektorquantisierungssysteme ist es, Coder bzw. Decoder α, γ, β so zu entwerfen, daß die Verzerrung zwischen dem Quellensymbol \mathbf{x}_n und seinem Repräsentanten $\hat{\mathbf{x}}_n$ zusammen mit der zur Übertragung benötigten Codewortlänge minimiert wird, nämlich die Minimierung der folgenden Lagrange-Kostenfunktion:

$$J_{\lambda}(\alpha, \gamma, \beta) = D(\mathbf{x}_n, \beta(\alpha(\mathbf{x}_n))) + \lambda \cdot |\gamma(\alpha(\mathbf{x}_n))|$$
(3.1)

dabei bedeutet $|\gamma(\alpha(\mathbf{x}_n))| = |U_n|$ die Länge des Kanalcodesymbols. Es wird angenommen, daß der Kanal störungsfrei ist, nämlich $U_n = \hat{U}_n, n = 0, 1, 2, \cdots$.

Einen Algorithmus zur Lösung der obigen Aufgabe stellt der bekannte iterative Entwurfsalgorithmus für Entropy-constrained vector quantization (ECVQ) von CHOU, LOOKA-BAUGH und GRAY [CLG89] dar, der im folgenden beschrieben wird.

Gegeben seien die Trainingsvektoren $\{\mathbf{x}_n, n = 0, 1, 2, \dots, N\}$, das Anfangscodebuch $\mathbf{C}^0 = \{\mathbf{y}_j^0\} = \{\beta^0(j)\}$ mit der Codewortlänge $\{|\gamma^0(j)|\}, j = 1, 2, \dots, J$, der Lagrange-Multiplikator λ und die Konvergenzschwelle ϵ . Setze: $k = 0, J^0 = \infty$ 1. $n(j) = 0, \mathbf{y}_{j}^{k+1} = \mathbf{0}, j = 1, \cdots, J$ Für n = 0 : N,

Suche für den Eingangsvektor \mathbf{x}_n ein Codewort \mathbf{y}_j^k aus dem Codebuch \mathbf{C}^k zur Minimierung der Lagrange-Kostenfunktion nach 3.1. Der Index des ausgewählten Codewortes y_j ist demnach gleich

$$j = \alpha(\mathbf{x}_n) = \arg\min_{\mathbf{y}_j^k \in \mathbf{C}^k} (D(\mathbf{x}_n, \mathbf{y}_j^k) + \lambda |\gamma^k(j)|)$$

Inkrementiere die Anzahl der entsprechenden Indizes:

$$n(j) = n(j) + 1$$

und summiere die zu dem Index j gehörenden Vektoren

$$\mathbf{y}_j^{k+1} = \mathbf{y}_j^{k+1} + \mathbf{x}_n$$

2. Berechne die Auftrittswahrscheinlichkeit p(j) des Codewortes \mathbf{y}_j aus n(j) und aktualisiere die jeweilige Codewortlänge. Für j = 1 : J und $p(j) \neq 0$:

$$|\gamma^{k+1}(j)| = -\log_2 p(j) = -\log_2 \frac{n(j)}{\sum_{i=1}^J n(i)}$$

3. Optimiere das Codebuch durch Neuberechnung der Clusterschwerpunkte. Für j = 1 : J und $n(j) \neq 0$:

$$\mathbf{y}_j^{k+1} = \frac{\mathbf{y}_j^{k+1}}{n(j)}$$

4. Berechne die Lagrange-Kostenfunktion neu.

$$J^{k+1} = \sum_{i=0}^{N} (D(\mathbf{x}_i, \mathbf{y}_j^{k+1}) + \lambda \cdot |\gamma^{k+1}(j)|)$$

5. Ist $\frac{J^k - J^{k+1}}{J^{k+1}} < \epsilon$, wird die Iteration abgebrochen. Sonst wird k = k + 1 gesetzt, und mit Schritt 1 fortgefahren.

In dem 2. Schritt wird die Auftrittswahrscheinlichkeit des jeweiligen Codewortes \mathbf{y}_j in jeder Iteration neu berechnet. Die Codewortlänge ergibt sich aus dem Zweierlogarithmus der Wahrscheinlichkeit des Auftretens von \mathbf{y}_j . Demzufolge hat das selten auftretende Codewort eine große Codewortlänge und daher eine große Lagrange-Kostenfunktion J, wodurch dieses bei der nächsten Iteration mit geringerer Wahrscheinlichkeit gewählt wird.

Der Clusterschwerpunkt im 3. Schritt ergibt sich als der Mittelwert aus aller zum Cluster (Häufungen) zugeordneten Vektoren \mathbf{x}_j mit dem Index j. Die Codewörter werden durch die Schwerpunktbildung des Clusters in jeder Iteration optimiert. Die beiden Schritte garantieren, daß jede Iteration konvergiert. Nach der Konvergenz erreicht das Codebuch ein lokales Minimum.

Es ist an dieser Stelle auf die Bedeutung der Trainingssequenzen hinzuweisen. Im Allgemeinen ist in der Entwurfsphase die zu codierende Videosequenz noch unbekannt. Außerdem ist es in der Praxis kaum möglich, die Wahrscheinlichkeitsverteilung der in der Regel instationären Bildsignale zu ermitteln. Einen Ausweg dafür zeigt das folgende pragmatische Vorgehen: Man benutzt genügend lange Trainingssequenzen, deren Bildsignalkomponenten in ihrer Statistik etwa denen entsprechen, die später codiert werden sollen, und entwirft ein optimales Codebuch für die gegebenen Trainingssequenzen. Wenn die Trainingssequenzen repräsentativ sind, sollte der Coder genauso gut bei einer Sequenz funktionieren, die nicht zu den Trainingssequenzen gehört. Daher ist die Codierung einer Videosequenz mit Vektorquantisierung möglich, auch wenn die statistischen Eigenschaften des Bildes unbekannt sind. In dieser Arbeit sind insgesamt 10 Bildsequenzen, die jeweils 1200 Frames in QCIF-Auflösung bei einer Bildfolgefrequenz von 10 Bildern pro Sekunde enthalten, als Trainingssequenzen benutzt worden.

3.2 Entwurf der Langzeitprädiktion mit dem iterativen Algorithmus

In diesem Abschnitt wird die Bewegungsschätzung mit Langzeitprädiktion im Zusammenhang mit der Vektorquantisierung nochmals betrachtet.

Für einen bestimmten Block im Bild sucht das Block-Matching-Verfahren aus dem gegebenen Suchraum $\Omega = \{v_x, v_y, v_t\}$ mit $v_x, v_y = -u(1)w, v_t = 1(1)M$ nach den optimalen Bewegungsparametern, die die Zielfunktion aus Gleichung 2.4 minimieren. Wenn man sich den Inhalt des Langzeitspeichers als ein Codebuch vorstellt und die 3 Bewegungsparameter als Indizes interpretiert, die auf die rekonstruierten Vektoren im Codebuch zeigen, dann kann man die Bewegungsschätzung als Vektorquantisierung mit Entropie-Nebenbedingung interpretieren. Dabei entspricht der Block, für den die Bewegungsschätzung durchgeführt wird, dem Eingangssymbol \mathbf{x}_n , $\alpha(\cdot)$ entspricht der Bewegungsschätzung und $\beta(\cdot)$ der Bewegungskompensation. Zu minimieren ist die Lagrange-Kostenfunktion in Gleichung 3.1. Das vorgestellte iterative Entwurfsverfahren für ECVQ kann daher für die Langzeitprädiktion zur Anwendung kommen.

Da der gesamte Inhalt des Langzeitspeichers zur Prädiktion herangezogen wird, entfällt der 3. Schritt des iterativen Algorithmus, d.h. das vollständige VQ-Codebuch wird in jeder Iteration verwendet. Außerdem wird die Entropiecodierung des Indizes mittels Huffmancodierung realisiert. Dabei wird zuerst für die drei Bewegungsparameter je eine Huffman-Tabelle angewendet. Insgesamt ergibt sich das folgende Flußdiagramm. In dem Schritt der Durchführung der Bewegungsschätzung wird die aus dem Kapitel 2.2 anhand des Struktogramms beschriebene Vorgehensweise vorgenommen.

In Tabelle 3.1 sind die Simulationsergebnisse unter Verwendung des beschriebenen iterativen Algorithmus zusammengestellt. Dabei wird λ in der Langrange-Kostenfunktion als 150 gewählt. Die Konvergenzschwelle ϵ wird zu 0.005 gesetzt, der Langzeitspeicher hat eine Größe von 10 Bildern und jeder Block hat die Größe 16 × 16. Die Spalte Rate und die PSNR-Werte in der Tabelle geben die durchschnittlichen Werte pro Makroblock an,



Abbildung 3.2: Entwurf der Langzeitprädiktion mit dem iterativen Algorithmus

ermittelt aus 10 Trainingssequenzen, die jeweils 500 Frames umfassen. In der Klammer sind die Entropie-Werte der jeweiligen Bewegungsparameter angegeben, die eine untere Grenze der Codewortlänge darstellen.

$$H = -\sum_{i} p(v_i) \cdot ld(p(v_i))$$
(3.2)

mit ld dem Logarithmus zur Basis 2 und $p(v_i)$ der Wahrscheinlichkeit des Auftretens von v_i .

Zum Vergleich sind auch die Ergebnisse für die Bewegungschätzung ohne Langzeitprädiktion in Tabelle 3.2 angegeben. Dabei wird der gleiche Algorithmus wie für die bewegungskompensierte Langzeitprädiktion verwendet, wobei sich die gesamte Bitrate aus den zwei Anteilen der Verschiebungsvektoren v_x und v_y zusammensetzt. In der Bewegungsschätzung wird nur das unmittelbar vorangegangene Bild zur Prädiktion herangezogen. Es ist zu sehen, daß die Werte ziemlich schnell konvergieren und nach 3 Iterationsschritten bereits innerhalb der Konvergenzschwelle liegen.

Tabelle 3.1: Iterationsergebnisse für Langzeitspeichergröße 10 ($\lambda = 150$)

Iteration	Rate(X)[bits]	Rate(Y)[bits]	Rate(T)[bits]	PSNR[dB]	$J = D + \lambda \cdot R$
1	5(2.7778)	5(2.4511)	5(2.7203)	31.0121	1.5436e + 04
2	$2.3063 \ (2.2673)$	2.0467 (1.9579)	2.0771(1.9874)	30.9979	1.3575e+04
3	2.2904(2.2651)	$2.0441 \ (1.9595)$	$2.0766 \ (1.9869)$	30.9978	1.3573e+04

Tabelle 3.2: Iterationsergebnisse für die Bewegungsschätzung ohne Langzeitspeicher ($\lambda = 150$)

Iteration	$\operatorname{Rate}(X)[\operatorname{bits}]$	Rate(Y)[bits]	Rate(T)[bits]	PSNR[dB]	$J = D + \lambda \cdot R$
1	5(2.2750)	5(2.0896)	0	29.3377	2.0889e + 04
2	$2.0005\ (1.8340)$	1.8421(1.6768)	0	29.3360	1.9973e+04
3	1.9680(1.8323)	$1.8376\ (1.6755)$	0	29.3359	1.9968e+04

Das aus dem iterativen Entwurf mittels Trainingssequenzen gewonnene Codebuch wird dann an zwei Testsequenzen: Foreman- und Mother-and-Daughter-Sequenz mit 8.33 Bilder pro Sekunde ausprobiert. In Abb. 3.3 und Abb. 3.4 sind die Rate-Distortion-Kurve der jeweiligen Sequenz gezeigt, berechnet aus 20 logarithmisch gleichverteilten λ - Werten zwischen 0 und 10⁵. Zum Vergleich sind auch die Ergebnisse für die bewegungskompensierten Prädiktion mit dem unmittelbar vorangegangenen Bild herangezogen.

An Abb. 3.3 und 3.4 erkennt man, daß im kleinen λ -Bereich von 0 bis 20 die zwei Kurven fast konstant laufen. Die Mother-and-Daughter-Sequenz hat bei $\lambda = 0$ den besten PSNR-Wert von 37.5380[dB] bei der Langzeitprädiktion. Dabei wird jedoch eine Bitratenerhöhung um 108 Bits/Frame im Vergleich zu $\lambda = 10$ benötigt, wobei die PSNR-Verbesserung nur 0.006[dB] beträgt. Das gleiche gilt auch für Foreman-Sequenz. Eine Bitratenersparnis von 65 Bits/Frame bei $\lambda = 40$ gegenüber $\lambda = 0$ kann durch einen geringeren PSNR-Verlust von nur 0.003[dB] erzielt werden. Diese zeigt, daß die Minimierung der Zielfunktion mit $\lambda = 0$ sehr ungünstig ist, da hohe Bitrate für nur geringe PSNR-Verbesserung benötigt wird.

Der Vergleich des Rate-Distortion-Verhaltens der bewegungskompensierten Prädiktion zwischen dem herkömmlichen Verfahren und dem Langzeitspeicher-Verfahren zeigt, daß die Verbesserung der bewegungskompensierten Langzeitprädiktion gegenüber der Prädiktion mit dem unmittelbar vorangegangenen Bild durch eine größere Bitrate erkauft werden muß.



Abbildung 3.3: Codierergebnis für Foreman-Sequenz



Abbildung 3.4: Codierergebnis für Mother-and-Daughter-Sequenz

3.3 Prädiktive Codierung der Bewegungsparameter

Bisher wurde die blockbasierte Bewegungsschätzung unabhängig voneinander durchgeführt und die dadurch gewonnenen Bewegungsparameter anschließend codiert. Es besteht jedoch eine hohe Korrelation zwischen den benachbarten Blöcken und dem entsprechenden Bewegungsparameter. Um diese Redundanz zu verringern und damit die zu übertragende Rate der Bewegungsparameter möglichst klein zu halten, erfolgt eine prädiktive Codierung der Bewegungsparameter. Das Ziel der Prädiktion ist es, durch die Differenzbildung aus dem eigentlichen Wert und seinem entsprechenden Prädiktor eine bessere Wertekonzentration zu erreichen. Dadurch läßt sich mittels eines Codes variabler Länge bei der Codierung der Differenz zum vorher gesagten Zentrum der Wertkonzentration des Prädiktionsfehlers, eine große Datenreduktion erzielen.

Im folgenden werden verschiedene Prädiktionsmodelle untersucht.

3.3.1 Lineare prädiktive Codierung der Bewegungsparameter

Ein geschlossenes Objekt bewegt sich i.allg. nicht ruckartig, sondern kontinuierlich. Daher kann eine Korrelation der Bewegungsparameter sowohl in den beiden örtlichen Richtungen, als auch entlang der Zeitachse erwartet werden. Aufgrund dieser Betrachtung wird ein 3-dim. Prädiktor gebildet, so daß sich die Bewegungsparameter aus den Parametern der örtlich links und oberhalb liegenden Blöcke, bzw. der zeitlich vergangenen Blöcke kausal prädizieren lassen. Diese Blöcke werden als *region of support* (ROS) bezeichnet. In Abb. 3.5 ist eine 3-dim. ROS dargestellt. Diese erstreckt über 3 Bilder mit dem aktuellen und den zwei zeitlich vorhergehenden Bildern. Block G in Bild k-1 und T in Bild k-2 liegen an der gleichen Position wie der aktuelle Block in Bild k. Es ist möglich, noch mehere Blöcke zu ROS hinzu zu nehmen. Da die weiter entfernten Blöcke eine geringe Korrelation zu dem aktuellen Block haben, ist deren Einfluß auf die Prädiktion geringer.

Nun stellt sich die Aufgabe, die Bewegungsparameter des aktuellen Blocks v_i^{-1} von den Bewegungsparametern in ROS linear zu prädizieren (sollte ROS außerhalb des Bildes liegen, werden deren Bewegungsparameter zu 0 gesetzt):

$$\hat{v}_i = \sum_{k=1}^n a_k \cdot m_{i-k} = \mathbf{a}^T \mathbf{m}_i, \qquad (3.3)$$

mit $\mathbf{a} = [a_1, \cdots, a_n]^T$ und $\mathbf{m}_i = [v_{i-1}, \cdots, v_{i-n}]^T$, wobei *n* der Prädiktionsgrad ist.

Die Koeffizienten a_k , auch Modellparameter genannt, werden derartig bestimmt, daß eine möglichst gute Übereinstimmung zwischen dem Bewegungsparameter v_i und seinem Schätzwert \hat{v}_i erzielt wird. Hier wird diese im Least-Square-Sinne bestimmt, nämlich durch die Minimierung des quadratischen Fehlers zwischen v_i und \hat{v}_i :

$$d_i = v_i - \hat{v}_i = v_i - \mathbf{a}^T \mathbf{m}_i \tag{3.4}$$

 $^{^1 {\}rm einfachheitshalber}$ wird hier nur eine Komponente der Bewegungsparameter ${\bf v}$ untersucht, der v bezeichnet wird.



Abbildung 3.5: ROS

$$\varepsilon(\mathbf{a}) = E\{d_i^2\}$$

$$= E\{v_i^2\} - 2\mathbf{a}^T E\{v_i \cdot \mathbf{m}_i\} + \mathbf{a}^T E\{\mathbf{m}_i \mathbf{m}_i^T\} \mathbf{a} \stackrel{!}{=} \min .$$
(3.5)
(3.6)

Mit

$$\mathbf{p} = E\{v_i \cdot \mathbf{m}_i\} = \begin{bmatrix} r_{vm}[1] & r_{vm}[2] \dots r_{vm}[n] \end{bmatrix}^T$$
(3.7)

und

$$\mathbf{R} = E\{\mathbf{m}_{i} \cdot \mathbf{m}_{i}^{T}\} = \begin{pmatrix} r_{mm}[1] & r_{mm}[2] & \dots & r_{mm}[n] \\ r_{mm}[2] & r_{mm}[1] & \dots & r_{mm}[n-1] \\ \vdots & & \dots & \vdots \\ r_{mm}[n] & r_{mm}[n-1] & \dots & r_{mm}[1] \end{pmatrix},$$
(3.8)

läßt sich Gleichung 3.6 umschreiben als:

$$\varepsilon(\mathbf{a}) = E\{v_i^2\} - 2\mathbf{a}^T \mathbf{p} + \mathbf{a}^T \mathbf{R} \mathbf{a}$$
(3.9)

Das Minimum bestimmt sich dann durch die folgende Gleichung:

$$\nabla(\varepsilon(\mathbf{a})) = \left[\frac{\partial\varepsilon}{\partial a_1}, \dots, \frac{\partial\varepsilon}{\partial a_n}\right]^T \stackrel{!}{=} \mathbf{0}$$
 (3.10)

$$\Rightarrow \mathbf{Ra} = \mathbf{p} \tag{3.11}$$

Man erhält schließlich

$$\mathbf{a} = \mathbf{R}^{-1}\mathbf{p} \tag{3.12}$$

Die Anwendung dieser Gleichung benötigt die Autokorrelierte $r_{mm}[n]$ und die Kreuzkorrelierte $r_{vm}[n]$ in Gleichung 3.7 und 3.8. Die zwei korrelierten Folgen sind jedoch i.allg. nicht gegeben, da die Prädiktion auf eine unbekannte Sequenz angewendet wird. Damit der Algorithmus ohne die a-priori Kenntnisse der Signalstatistik auskommt, wird ein iterativer Entwurfsvorgang entwickelt, in dem jedesmal die Modellparameter in **a** von den gegebenen Trainingssequenzen "lernen" und dann aktualisiert werden.

Zum Anfang der Iteration werden die Modellparameter in \mathbf{a} zu 0 gesetzt. Ausgehend von dem Prädiktor \hat{v}_i , der sich aus innerem Produkt der Modellparameter \mathbf{a} und der Bewegungsparameter in ROS \mathbf{m} nach Gleichung 3.3 ergibt, startet der spiralförmige Suchvorgang. Die optimalen Bewegungsparameter werden gefunden, indem die folgende Zielfunktion minimiert wird:

$$J = D(\mathbf{v}) + \lambda \cdot R(\mathbf{d}) \tag{3.13}$$

mit

$$\mathbf{d} = (d_x, d_y, d_t) = (v_x - \hat{v}_x, v_y - \hat{v}_y, v_t - \hat{v}_t)$$

Wie bereits erklärt, wird der Prädiktionsfehler **d**, die Differenz zwischen aktuellem Bewegungsparamter und dem Prädiktor, anstatt des richtigen Bewegungsparameters **v** codiert. $R(\mathbf{d})$ stellt die Summe der Rate dar, mit der d_x, d_y und d_t codiert wird.

Während dieses Vorgangs werden \mathbf{p} in Gleichung 3.7 und \mathbf{R} in Gleichung 3.8 blockweise akkumuliert:

$$\mathbf{p} = \sum_{i=n}^{L} v_i \cdot \mathbf{m}_i \tag{3.14}$$

und

$$\mathbf{R} = \sum_{i=n}^{L} \mathbf{m}_i \cdot \mathbf{m}_i^T, \qquad (3.15)$$

Dabei ist L die Gesamtanzahl der betrachteten Blöcke.

Die Modellparameter in **a** werden nach Gleichung 3.12 aktualisiert und in der nächsten Iteration zur Berechnung des Prädiktors nach Gleichung 3.3 angewandt und wieder aktualisiert bis zur Konvergenz, d.h. bis die Verbesserung der Zielfunktion aufhört bzw. die Zielfunktion nicht mehr wesentlich verbessert wird.

Der oben beschriebene iterative Algorithmus zur Optimierung der Modellparameter wird eingebettet in dem im Kapitel 3.2 erläuterten iterativen Algorithmus zum Entwurf von Vektorquantisierung mit Entropie-Nebenbedingung.

Insgesamt ergibt sich der folgende Algorithmus:

1. Initialisierung: Gegeben: Lagrange-Parameter λ : Prädiktionsgrad n, bzw. Festlegung der ROS; Huffman-Tabelle mit Codewörtern fester Länge, nämlich $R(d_i) = \text{const.} \forall d_i \in \Omega$ Gesamte Anzahl der betrachteten Blöcke LKonvergenzschwelle: ϵ Setzen: $J_0 = \infty, \quad k = 0, \quad \mathbf{p} = \mathbf{0}, \quad \mathbf{R} = \mathbf{0}$ $a_0 = 0$,

2. Blockweise Codierung mit Minimierung der Zielfunktion in Gleichung 3.13 für alle Bilder und Sequenzen. Akkumulieren von **p** und **R**.

$$J_k = \sum_{i=0}^{L} (D(\mathbf{v}_i) + \lambda \cdot R(\mathbf{d}_i))$$

$$\mathbf{p} = \sum_{i=0}^{L} (\mathbf{v}_i \mathbf{m}_i)$$

$$\mathbf{R} = \sum_{i=0}^{L} (\mathbf{m}_i \mathbf{m}_i^T)$$

3. Aktualisierung der Modellparameter

Setzen:

$$k = k + 1,$$

 $\mathbf{a}_k = \mathbf{R}^{-1}\mathbf{p}$

4. Neue Berechnung der Bewegungsparameter mit Modellparameter und Aktualisierung der Huffman-Tabelle anhand des Histogramms.

 $n(d_i) = 0, \forall d_i \in \Omega$ Für i = 1(1)L, $\{\hat{v}_i = \mathbf{a}_k^T \mathbf{m}_i\}$ $n(d_i) = n(v_i - \hat{v}_i) + 1$ $R(d_i) = h(n(d_i))$, wobei h eine Huffmancodierung bedeutet.

)

5. Konvergenzkontrolle Ist $(\frac{J_k - J_{k-1}}{J_{k-1}} < \epsilon)$, wird die Iteration abgebrochen. Sonst wird k = k + 1 gesetzt, und mit Schritt 2 fortgefahren.

Im folgenden werden mit dem vorgestellten iterativen Algorithmus die prädiktive Codierung der Bewegungsparameter jeweils bei der Langzeitprädiktion und bei der Prädiktion mit dem unmittelbar vorangegangen Bild untersucht.

Lineare prädiktive Codierung der Bewegungsparameter bei der Bewegungskompensation ohne Langzeitspeicher

Die Bewegungsschätzung wird durch die blockbasierte Suche innerhalb eines bestimmten Suchgebietes des unmittelbar vorangegangenen Bildes nach den optimalen Bewegungsparametern vorgenommen, der die Lagrange-Kostenfunktion $J = D + \lambda \cdot R$ minimiert.

Anstatt die Suche in der unmittelbaren Mitte des Suchbereichs zu starten, wie es üblicherweise bei Bewegungsschätzverfahren der Fall ist, wird hier zuvor ein Prädiktor verwendet. Dieser bestimmt den Startpunkt des Spiralförmigen Suchvorgangs.

Nach dem beschriebenen Algorithmus werden die Modellparameter des linearen Prädiktors und die Wahrscheinlichkeits-Tabelle der Bewegungsparameter anhand von 10 Trainingssequenzen iterativ trainiert und dann an den zwei Testsequenzen Foreman- und Mother-and-Daughter-Sequenz ausprobiert. Die Konvergenzschwelle ϵ wird zu 0.005 gesetzt, und jeder Block hat die Größe 16 × 16.

In Tabelle 3.3, 3.4 und 3.5 sind die Simulationsergebnisse zusammengestellt, jeweils für $\lambda = 150$, $\lambda = 0$ und 10000. Die Rate und die Verzerrung geben die durchschnittlichen Werte pro Makroblock an, ermittelt aus allen Trainingssequenzen, die jeweils 500 Frames umfassen. Die Rate ergibt sich aus der Huffmancodierung mittels der Wahrscheinlichkeits-Tabelle aus dem letzten Iterationsschritt. Die Entropiewerte, die sich aus Gleichung 3.2 berechnen lassen, sind in der Klammer angegeben.

Die Modellparameter und die Verteilung der Bewegungsparameter (x-Komponente wird betrachtet) anhand der Simulationsergebnisse der 3. Iteration sind in Abb. 3.6 und 3.7 dargestellt.

Tabelle 3.3: Iterationsergebnis für das lineare Prädiktionsmodell ohne Langzeitspeicher($\lambda = 150$)

Iteration	Rate(X)[bits]	Rate(Y)[bits]	distortion[dB]	$J = D + \lambda \cdot R$
1	5(2.1222)	5(1.9330)	29.3377	2.0889e + 04
2	1.6359(1.3563)	1.5405(1.2736)	29.3281	1.8445e+04
3	1.5368(1.3341)	1.4892(1.2640)	29.3253	1.8310e + 04

Tabelle 3.4: Iterations
ergebnis für das lineare Prädiktionsmodell ohne Langzeitspeicher
($\lambda=0)$

Iteration	Rate(X)[bits]	Rate(Y)[bits]	distortion[dB]	$J = D + \lambda \cdot R$
1	5(2.1222)	5(1.9330)	29.3377	1.9389e + 04
2	1.9331(1.8092)	1.8156(1.6830)	29.3377	1.9389e + 04

Tabelle 3.5: Iterations
ergebnis für das lineare Prädiktionsmodell ohne Langzeitspeicher
($\lambda=10000)$

Iteration	Rate(X)[bits]	Rate(Y)[bits]	distortion[dB]	$J = D + \lambda \cdot R$
1	5(2.1222)	5(1.9330)	29.3377	1.1939e+05
2	1.3936(1.0001)	1.3199(0.8774)	28.7423	4.9372e+04
3	1.3275(0.9735)	1.2784(0.8623)	28.7971	4.8018e+04

		0.0639		
		0.1010		
		0.0166		
	0.0338	0.2094	0.0819	
	0.0178	0.3427	0.1813	
	0521	0.4443	0.1695	
0.0349	0.2273			
0.0460	0.2583	MV		
0.0703	0.3342		Bild	1 k

$\lambda = 0$
$\lambda = 150$
λ=1000



Abbildung 3.6: Modellparameter für ROS



Abbildung 3.7: Wahrscheinlichkeitsverteilung der Bewegungsparameter (x-komponente)

Aus den obigen Simulationen wurden die Modellparameter des linearen Prädiktors und die Huffman-Tabelle für die Bewegungsparameter jeweils für $\lambda = 0,150$ und 10000 gewonnen. Diese werden dann an zwei Testsequenzen ausprobiert. Ein Vergleich des Codierergebnisses zu dem Fall, daß die Bewegungsparameter nicht modelliert werden, zeigt Abb. 3.8 und 3.9. Eine Verbesserung des Rate-Distortion-Verhaltens nach der Prädiktion ist zu erkennen. Dabei ist die Verbesserung bei der Foreman-Sequenz deutlicher als bei der Mother-and-Daughter-Sequenz.

Die Verbesserung der PSNR-Werte nach der Prädiktion bei der Foreman-Sequenz im kleinen λ -Bereich läßt sich darauf zurückführen, daß durch die Prädiktion der Bewegungsparameter vor der Bewegungsschätzung eine Vergrößerung des Suchbereiches erreicht wird, da die Beweugngsschätzung ausgehend von dem Prädiktor durchgeführt wird. Demzufolge ergibt sich eine verbesserte Bewegungskompensation durch die Prädiktion der Bewegungsparameter vor der Bwegungsschätzung, was insbesondere bei stark bewegten Bildsequenzen wie Foreman-Sequenz deutlich wird.

Die Ergebnisse aus dem linearen Prädiktionsmodell mit den Modellparametern aus dem Training mit verschiedenen λ zeigen, daß ihr Rate-Distortion-Verhalten keine großen Unterschiede aufweisen. Es ist daher zu vermuten, daß der Prädiktor nicht sehr empfindlich gegenüber der Prädiktionsgenauigkeit ist.



Abbildung 3.8: Codierergebnis für Foreman-Sequenz



Abbildung 3.9: Codierergebnis für Mother-and-Daughter-Sequenz

Lineare prädiktive Codierung der Bewegungsparameter bei der Langzeitprädiktion

Bei der Langzeitprädiktion wird noch ein t-Parameter hinzugefügt, der die Adressierung im Langzeitspeicher beschreibt. Die gesamte Seiteninformation der Bewegungsparameter besteht daher aus den drei Anteilen: x, y für die translartorische Bewegungsverschiebung und t für den Zeitparameter.

Dem letzten Abschnitt entsprechend, werden die Modellparameter des linearen Prädiktors und die Huffman-Tabelle der Bewegungsparameter jeweils für die x-, y- und t-Parameter mittels Training anhand der 10 Trainingssequenzen für $\lambda = 150$ bestimmt und dann anhand der Testsequenzen ausprobiert. In Tabelle 3.6 sind die Iterationsergebnisse für die Bewegungsschätzung mit einer Langzeitspeichergröße 10 dargestellt.

Tabelle 3.6: Prädiktive Coc	dierung der Bewegungsparan	neter mit linearem Modell
-----------------------------	----------------------------	---------------------------

Iteration	Rate(X)[bits]	Rate(Y)[bits]	Rate(T)[bits]	PSNR[dB]	$J = D + \lambda \cdot R$
1	5(1.9764)	5(1.7932)	5(2.6467)	31.0160	1.3924e+05
2	2.0021(1.8560)	1.8482(1.6801)	2.7927(2.5690)	30.9499	1.4372e+04
3	1.8927(1.7917)	1.7594(1.6211)	2.4403(2.3931)	30.9043	1.3796e+04

Die Entropiewerte der Bewegungsparameter v_x und v_y sind nach der prädiktiven Codierung stark reduziert, dieses macht das Histogramms deutlich. Die prädiktive Codierung des Zeitparameters ist jedoch unzufriedenstellend. Die Rate erhöht sich nach der Prädiktion.

Die Modellparameter aus der 3. Iteration sind in Abb. 3.12 dargestellt. Man beobachtet, daß die direkt oberhalb und vor dem aktuellen Block liegenden Blöcke, A und B nach der Bezeichnung in Abb. 3.5, den größten Einfluß auf die Prädiktion haben. Der zeitlich direkt davor liegende Block hat auch einen wichtigen Beitrag. Je weiter die Blöcke von dem aktuellen Block entfernt sind, desto geringer ist ihre Wirkung auf die Prädiktion.

Die durch das Training erhaltenen Modellparameter und die Huffmann-Tabelle wurden ebenfalls anhand der Foreman- und Mother-and-Daughter-Sequenz getestet. Abb. 3.13 und Abb. 3.14 zeigen die Codierergebnisse, zum Vergleich ist die Codierung ohne Prädiktion aus Kapitel 3.2 herangezogen. Die prädiktive Codierung zeigt im Bereich niedriger Bitrate (großer λ - Werte) aber ein schlechteres Ergebnis. Einerseites wird wegen der schlechteren Codierung des Zeitparameters die Bitrate immer höher. Andererseits sind die Modellparameter und die Huffman-Tabelle aus dem Training mit $\lambda = 150$ gewonnen. Diese sind vermutlich für große λ - Werte nicht geeignet.



Abbildung 3.10: Histogramm der Bewegungsparameter vor der prädiktiven Codierung



Abbildung 3.11: Histogramm der Bewegungsparameter nach der prädiktiven Codierung



Abbildung 3.12: Modellparameter für ROS



Abbildung 3.13: Codierergebnis mit vs. ohne Prädiktion für Foreman-Sequenz



Abbildung 3.14: Codierergebnis mit vs. ohne Prädiktion für Mother-and-Daughter-Sequenz

3.3.2 Nichtlineare prädiktive Codierung der Bewegungsparameter

Neben dem linearen Prädiktionsmodell sind andere nichtlineare Prädiktoren möglich, z.B. der Medianfilter. Bei einem Medianfilter werden p Elemente (p wird als ungerade Zahl angenommen) der Größe nach geordnet, das resultierende Ausgangselement ist der mit (p + 1)/2 - indizierte Wert dieser Reihe.

Anhand der in Abb. 3.6 und Abb. 3.12 eingetragenen Ergebnisse der Modellparameter aus der linearen Prädiktion, sind die benachbarten Blöcke des aktuellen Blocks, nämlich A, B und D nach Abb. 3.5 und die an der gleichen Position sich befindenden Blöcke wie der aktuelle Block der vorangegangenen Bilder, nämlich G und T in ROS für die Prädiktion maßgebend. Daher werden im folgenden 2 Arten von Medianfilter Prädiktor untersucht, für die p jeweils als 3 und 5 genommen wird.

- 1. Modell 1: Blöcke A, B, D werden gewählt. (Dies entspricht dem Prädiktor, wie er auch im H.263-Standard vorgesehen ist). Prädiktor = MED(A, B, D).
- 2. Modell 2: Blöcke G und T werden zu A, B, D hinzugenommen und bilden einen Medianfilter 5. Grades. Prädiktor = MED(A, B, D, G, T).

Sollte der aktuelle Block an der Bildkante liegen, so wird eine Regel ähnlich zu der im H.263-Standard verwendet:

- 1. Die Bewegungsparameter von A werden zu Null gesetzt, falls A außerhalb des Bildes liegt
- 2. Die Bewegungsparameter von B und von D werden gleich dem aktuellen Bewegungsparameter gesetzt, wenn B und D außerhalb des Bildes liegen
- 3. Die Bewegungsparameter von D werden zu Null gesetzt, falls D außerhalb des Bildes liegt.

3.3.3 Vergleich verschiedener Prädiktionsmodelle

Im folgenden werden die aus den letzten zwei Abschnitte beschriebenen verschiedenen linearen und nichtlinearen Prädiktionsmodelle jeweils für die bewegungskompensierte Prädiktion mit und ohne Langzeitspeicher miteinander verglichen.

Vergleich verschiedener Prädiktionsmodelle für die bewegungskompensierte Prädiktion ohne Langzeitspeicher

In Tabelle 3.7 und 3.8 sind die Ergebnisse verschiedener Prädiktionsmodelle jeweils für die Foreman- und Mother-and-Daughter-Sequenz aufgelistet. Die Rate-Distortion-Kurven für die beiden Testsequenzen, berechnet an 20 logarithmisch gleichverteilten λ - Werten zwischen 0 und 10⁵, sind in Abb. 3.15 und 3.16 zu finden.

Tabelle 3.7: Codierungsperformance mit verschiedenen Prädiktionsmodellen für Foreman-Seq. ($\lambda = 150$)

Prädiktionsmodell	Rate(X)[bits]	Rate(Y)[bits]	PSNR[dB]
Ohne Prädiktion	4.3000	4.3812	28.2633
lin. Prädiktor(trainiert mit $\lambda = 150$)	2.9458	2.9242	28.5499
H.263-artiger Medianfilter	2.4378	2.4776	28.4881
Medianfilter 5. Grades	2.6127	2.5915	28.5105

Tabelle 3.8: Codierungsperformance mit verschiedenen Prädiktionsmodellen für Motherand-Daughter-Seq. ($\lambda = 150$)

Prädiktionsmodell	Rate(X)[bits]	Rate(Y)[bits]	PSNR[dB]
Ohne Prädiktion	1.2190	1.2523	36.5109
lin. Prädiktor(trainiert mit $\lambda = 150$)	1.1759	1.2050	36.5108
H.263-artiger Medianfilter	1.1953	1.2320	36.5110
Medianfilter 5. Grades	1.1816	1.1901	36.5105

Durch die prädiktive Codierung der Bewegungsparameter sind die benötigten Bitraten reduziert worden, was insbesondere bei der Foreman-Sequenz deutlich wird. Im Gegensatz dazu ist die Verbesserung bei der Mother-and-Daughter-Sequenz nur gering, da die Bewegungsparameter vor der Prädiktion bereits mit geringer Rate codiert werden können und sich durch die weitere Modellierung nicht mehr wesentlich verbessern lassen.

Während der lineare Prädiktor bei Mother-and-Daughter-Sequenz eine geringere Verbesserung gegenüber den H.263-artigen Prädiktor zeigt, weist der letztere bei Foreman-Sequenz aber ein besseres Codierergebnis im Vergleich zum linearen Prädiktor auf. Es ist zu vermuten, daß Foreman-Sequenz eine stärkere räumliche Abhängigkeit zwischen benachbarten Blöcke aufweist, wobei Mother-and-Daughter-Sequenz mehr zeitliche Kontinuität zeigt.

Für die Codierung mit Codewörter fester Länge benötigt jeder Verschiebungsvektor v_x, v_y mit dem Suchraum $\Omega = [-16, 15] \times [-16, 15]$ jeweils 5 Bits. Die obigen Ergebnisse zeigen, daß durch die Entropiecodierung bei anschließender prädiktiven Codierung viele Bits gespart werden können. Dabei ist eine Datenreduktion von 40% bis zum 80% für die beiden Testsequenzen erreichbar.



Abbildung 3.15: Codierergebnis für Foreman Seq.



Abbildung 3.16: Codierergebnis für Mother-and-Daughter Seq.

Vergleich verschiedener Prädiktionsmodelle für die bewegungskompensierte Prädiktion mit Langzeitspeicher

Ein Vergleich der Ergebnisse zwischen Medianfilter-Prädiktor und dem linearen Prädiktor ist in folgender Tabelle zusammengefaßt. Dabei ist der Rate in bits/Block angegeben. Der

Parameter λ in der Lagrange-Kostenfunktion wird zu 150 gesetzt. Der Langzeitspeicher hat eine Größe von 10 Bildern.

Tabelle 3.9: Codierungsergebnis mit verschiedenen Prädiktionsmodellen für die Trainingssequenzen. $(\lambda = 150)$

Prädiktionsmodell	Rate(X)[bits]	Rate(Y)[bits]	Rate(T)[bits]	PSNR[dB]
Ohne Prädiktion	2.2904(2.2651)	2.0441(1.9595)	2.0766(1.9869)	30.9978
lin. Prädiktor	1.8927(1.7917)	1.7594(1.6211)	2.4403(2.3931)	30.9043
Medianfilter 3. Grades	1.7218(1.5139)	1.6306(1.3847)	1.9522(1.8039)	30.7389
Medianfilter 5. Grades	1.8461(1.7134)	1.7265(1.5466)	2.0762(1.9764)	31.0001

Tabelle 3.10: Codierungsergebnis mit verschiedenen Prädiktionsmodellen für Foreman-Seq. $(\lambda = 150)$

Prädiktionsmodell	Rate(X)[bits]	Rate(Y)[bits]	Rate(T)[bits]	PSNR[dB]
Ohne Prädiktion	4.6950	4.6948	2.6210	30.4326
lin. Prädiktor	3.3730	3.3377	2.7300	30.7154
Medianfilter 3. Grades	3.0861	2.9678	2.2879	30.7102
Medianfilter 5. Grades	3.0963	2.9737	2.2840	30.7108

Tabelle 3.11: Codierungsergebnis mit verschiedenen Prädiktionsmodellen für Mother-and-Daughter-Seq. $(\lambda = 150)$

Prädiktionsmodell	Rate(X)[bits]	Rate(Y)[bits]	$\operatorname{Rate}(T)[\operatorname{bits}]$	PSNR[dB]
Ohne Prädiktion	1.4265	1.3519	1.4850	37.4934
lin. Prädiktor	1.2724	1.2241	1.7944	37.4982
Medianfilter 3. Grades	1.2958	1.2238	1.5020	37.4860
Medianfilter 5. Grades	1.2956	1.2241	1.5020	37.4860

Die Rate-Distortion-Kurven der zwei Testsequenzen sind in Abb. 3.17 und 3.18 dargestellt.

Aus den vorliegenden Ergebnisse ist zu erkennen, daß für die Codierung der zwei Verschiebungsvektoren v_x und v_y eine große Bitratenreduktion infolge der Prädiktion erzielbar ist. Dabei ist die Verbesserung bei der Foreman-Sequenz deutlicher. Unter allen Prädiktionsmodellen ergibt der Medianfilter-Prädiktor 3. Grades ein relativ gutes Ergebnis trotz seiner einfachen Implementierung. Daher wird er meistens unter verschiedenen Prädiktionsmodellen bevorzugt. Im Gegensatz zu der Codierung der zwei Verschiebungsvektoren sind die bisherigen Codierresultate für den Zeitparameter nicht zufriedenstellend, wenn



Abbildung 3.17: Codierergebnis für Foreman-Seq.



Abbildung 3.18: Codierergebnis für Mother-and-Daughter-Seq.

man bedenkt, daß bei der Codierung des Zeitparameters mit Codewörtern fester Länge eine durchschnittliche Rate von etwa 3 bits/Block benötigt wird. Außerdem sind duch die prädiktive Codierung die benötigte Bitrate für den Zeitparameter zum Teil erhöht worden. Daher wird im folgenden Abschnitt das Verhalten des Zeitparameters genauer untersucht und nach einem besseren Codierverfahren gesucht.

3.4 Codierung des Zeitparameters unter Verwendung des Finite-State-Mechanismus

Bisher wurde in der Bewegungsschätzung mit Vektorquantisierung ein einziges Codebuch für den Zeitparameter sämtlicher Blöcke berücksichtigt und die Codierung aller Zeitparameter als gedächtnislos durchgeführt. Es ist jedoch zu erkennen, daß die Bewegungsparameter benachbarten Blöcken hohe Korrelation haben. Es ist wahrscheinlich, daß ein bestimmter Block in aufeinanderfolgenden Bildern ähnliche Zeitparameter aufweist. Wenn man das Gedächtniss bzw. die Korrelation zwischen aufeinanderfolgenden Vektoren ausnutzt, läßt sich eine bessere Optimierung der Codierung erzielen. Eine bekannte Stragie dafür stellt der Finite-state-Mechanismus [FGD85] [DG85] dar.

Bei dem Finite-State-Mechanismus korrespondiert jedes mögliche Symbol, hier der Zeitparameter, mit einem der endlichen Zustände (davon leitet sich der Begriff Finite-State ab). Je nach dem Zustand wird ein Codewort aus dem Zustandscodebuch zur Codierung des aktuellen Symbols verwendet. Die Codewörter bestimmen zusammen mit dem aktuellen Zustand wieder den Folgezustand. Der Vorteil dieser Vorgehensweise liegt darin, daß eine Adaption der Codierung an die Änderung des Bildsignals möglich ist. Das Zustandscodebuch kann besser an den jeweilig zu codierenden Vektor angepaßt werden wie bei der normalen VQ ohne Gedächtnis, bei der nur ein einziges Codebuch für alle Vektoren verwendet wird.



Abbildung 3.19: Arbeitsweise der FSVQ

Eine Finite-State-Vektorquantisierung ist demnach charakterisiert durch einen Zustandsraum **S** mit S endlichen Zuständen, Zustandscodebücher $\{\mathbf{C}_s, s \in \mathbf{S}\}$ und eine Folgezustandsfunktion $f(U_n, s)$. Der Decoder hat den gleichen Zustandsraum **S** wie der Coder. Unter der Annahme, daß die Folgezustandsfunktion f nur vom letzten Zustand und den gewählten Kanalcodesymbolen U_n abhängig ist, kann der Decoder den Zustand verfolgen, ohne zusätzliche Seiteninformation für die Mitteilung des Zustands zu übertragen, falls ein Anfangszustand s_0 beiden Seiten bekannt ist und der Kanal störungsfrei ist. Nach der Terminologie aus Kapitel 3.1 sind bei der Vektorquantisierung mit Eingangsvektoren { $\mathbf{x}_n, n = 0, 1, 2, ...$ }, gegebenem Anfangszustand $s_0 \in \mathbf{S}$ und Kanalcodesymbolsequenzen { $U_n, n = 0, 1, 2, ...$ } die Rekonstruktionsvektoren $\hat{\mathbf{x}}_n$ rekursiv definiert:

$$U_n = \gamma(\alpha(\mathbf{x}_n, s_n)), \qquad s_{n+1} = f(U_n, s_n),$$
$$\hat{\mathbf{x}}_n = \beta(\gamma^{-1}(U_n, s_n)) = \beta(\alpha(\mathbf{x}_n, s_n), s_n), \quad n = 0, 1, 2, \dots$$

Die Arbeitsweise der Coder und Decoder bei der Finite-State-VQ zeigt das obenstehende Blockschaltbild.

Zwei Aspekte sind für die Anwendung des Finite-State-Mechanismus wichtig: Entwurf des Zustandscodebuchs und Festlegung der Folgezustandsfunktion.

Im folgenden wird jedem möglichen Zeitparameter ein Zustand zugeordnet. Daher ergeben sich insgesamt M(= Größe der Langzeitspeicher) Zustände, zu denen jeweils ein Zustandscodebuch gehört. Das Zustandscodebuch wird folgendermaßen entworfen. Zuerst werden alle M Zeitparameter in jedem Zustandscodebuch als gleichwahrscheinlich angenommen. Mit diesen Anfangszustandscodebüchern werden die Trainingssequenzen derart codiert, daß die Lagrange-Kostenfunktion nach Gleichung 2.4 minimiert wird (diese entspricht $\lambda = 0$). Dabei werden alle Zeitparameter gezählt, die nach dem Auftreten eines Zustandes gewählt werden. Dadurch bekommt man für je einen Zustand ein Histogramm. Anhand dieser Histogramme werden die Anfangszustandscodebücher aktualisiert und in der nächsten Iteration der Codierung verwendet.

Da die Zustandscodebücher anhand des jeweiligen Codierzustandes entworfen werden und besser an die statistischen Abhängigkeit des Signals angepaßt sind, genügt für die Übertragung der Codesymbole eine geringere Bitrate als im bisherigen Fall, bei dem nur ein Codebuch verwendet wurde. Für die Wirkungsweise des Finite-State-Mechanismus ist auch das Festlegen der Folgezustandsfunktion maßgebend. Diese entscheidet über den jeweiligen Zustand, mit dem ein bestimmtes Zustandscodebuch gewählt wird.

Es werden im folgenden zwei Arten der Folgezustandsfunktionen vorgestellt:

- 1. der gewählte Zeitparameter des entsprechenden Blocks legt zugleich den Folgezustand fest,
- 2. der Prädiktor des Zeitparameters des jeweiligen Blocks bestimmt den Folgezustand.

In dem 1. Modell entspricht der Zeitparameter einer Makovquelle 1. Ordnug. Im 2. Modell gehen wir von der Vorstellung aus, daß jeder Zustand als eine grobe Schätzung des codierten Zeitparameters gesehen werden kann und daher eine Art Prädiktor des Zeitparameters darstellt. Dabei wird der H.263-artige Prädiktor eingesetzt.

Die Zustandscodebücher werden zuerst anhand von 10 Trainingssequenzen iterativ entworfen und dann anhand der zwei Testsequenzen ausprobiert. In Tabelle 3.12 sind die zur Codierung des Zeitparameters benötigte Bitrate in bits/Block für verschiedene Codiermodelle angegeben. Bei Finite-State(I) bzw. (II) werden die Folgezustandsfunktionen durch

Prädiktionsmodell	Trainingssequenzen	Foreman-Sequenz	M-D-Sequenz
Ohne Prädiktion	2.0766	2.6210	1.4850
lin. Prädiktor	2.4403	2.7300	1.7944
Medianfilter 3. Grades	1.9522	2.2879	1.5020
Medianfilter 5. Grades	2.0762	2.2840	1.5020
Finite-State (I)	1.9408	2.0737	1.5284
Finite-State (II)	$1.8\overline{252}$	2.0028	1.4737

Tabelle 3.12: Rate für den Zeitparameter ($\lambda = 150$)

den oben beschriebenen Methoden festgelegt. Die beiden verwenden den H.263-artigen Prädiktor. Es ist zu sehen, daß das 2. Modell der Festlegung der Folgezustandsfunktion durch den Prädiktor des Zeitparameters ein besseres Ergebnis erzielt.

In Abb. 3.20 und 3.21 sind die Rate-Distortion-Kurven der beiden Testsequenzen für die beschriebenen Codierungsmethoden angegeben. Wie man sieht, durch die Verwendung des Finite-State-Mechanismus ist eine große Bitratenreduktion bei der Foreman-Sequenz erreicht worden. Im Gegensatz dazu ergeben sich die verschiedenen Codierverfahren bei der Mother-and-Daughter-Sequenz keine wesentlichen Unterschiede. Wie bereits erwähnt, läßt sich das Codierergebnis bei den bereits leicht codierbaren Bewegungsparametern durch die weitere Verfahren nicht mehr verbessern.



RD Vergleich zwischen verschiedenen Praediktionsmodellen mit Langzeitspeicher fuer Foreman-Sec

Abbildung 3.20: Codierergebnis für Foreman-Seq.



RD Vergleich zwischen verschiedenen Praediktionsmodellen mit Langzeitspeicher fuer M-D-Seq.

Abbildung 3.21: Codierergebnis für Mother-and-Daughter-Seq.

Kapitel 4

Gemeinsame Codierung der drei Bewegungsparameter

Nachdem die drei Bewegungsparameter im letzten Kapitel einzeln untersucht wurden, werden sie nun zusammengefaßt und gemeinsam codiert. Unter Verwendung des Finite-State-Mechanismus ergeben sich daher insgesamt M (= Langzeitspeichergröße) Zustandscodebücher für alle drei Parameter.

In der Tabelle 4.1 sind Zustandscodebücher abgebildet. In jeder Zeile steht ein Zustandscodebuch $\{\mathbf{C}_s, s \in \mathbf{S}\}$ mit der Länge l. Der Index jedes Codewortes ist einem aus drei Bewegungsparametern zusammengesetzten Vektor $[v_x, v_y, v_t]^T$ zugeordnet.

Tabelle 4.1: Zustandscodebücher

S_1 :	y_1^1	y_2^1	 y_l^1
S_2 :	y_1^2	y_2^2	 y_l^2
÷			
S_M :	y_1^M	y_2^M	 y_l^M

Tabelle 4.2, 4.3 und 4.4 zeigen einen Vergleich der Codierergebnisse zwischen den beiden Fällen, jeweils für die Trainingssequenzen, Foreman- und Mother-and-Daughter-Sequenz. Im ersten Fall (A) wird für die drei Bewegungsparameter je ein gesondertes Codebuch verwendet, die gesamten Bitraten ergeben sich aus $R = R_x(v_x) + R_y(v_y) + R_t(v_t)$. Im zweiten Fall (B) werden alle Parameter zusammengefaßt und sie verwenden ein gemeinsames Codebuch, $R = R_{xyt}(v_x, v_y, v_t)$. Gleichzeitig wird auch ein Vergleich der Codierung der Bewegungsparameter mit und ohne prädiktion sowie mit und ohne Finite-State-Mechanismus angegeben. In dem Fall, daß die Bewegungsparameter prädiktiv codiert werden, erfolgt die H.263-artige Prädiktion. Für die Verwendung des Finite-State-Mechanismus wird die Folgezustandsfunktion durch den H.263-artigen Prädiktor des Zeitparameters festgelegt. In der Klammer sind die zu der Rate korrespondierenden PSNR-Werte mit angegeben. Dabei hat der Langzeitspeicher eine Größe von 10 Bildern.

Aus den vorliegenden Ergebnissen kann man folgendes feststellen: Während die verschiedenen prädiktiven Codierverfahren für die Mother-and-Daughter-Sequenz keine wesentliche

	getrennte Codierung (A)	gemeinsame Codierung (B)
ohne Prädiktion	6.4111 (30.9978[dB])	6.1312 (30.9968[dB])
mit Prädiktion	5.3046 (30.9987[dB])	4.4345 (30.9894[dB])
mit Finite-State	5.1755 (30.9988[dB])	4.1673 (30.9919[dB])

Tabelle 4.2: Rate für die Bewegungsparameter bei verschiedenen Codierverfahren für die Trainingssequenzen($\lambda = 150$)

Tabelle 4.3: Rate für die Bewegungsparameter bei verschiedenen Codierverfahren für Foreman-Sequenz($\lambda = 150$)

	getrennte Codierung (A)	gemeinsame Codierung (B)
ohne Prädiktion	11.3957 (30.4162[dB])	11.5115 (30.4010[dB])
mit Prädiktion	8.0254 (30.6564[dB])	7.5170 (30.6382[dB])
mit Finite-State	7.6941 (30.6469[dB])	7.2833 (30.6443[dB])

Tabelle 4.4: Rate für die Bewegungsparameter bei verschiedenen Codierverfahren für Mother-and-Daughter-Sequenz $(\lambda = 150)$

	getrennte Codierung (A)	gemeinsame Codierung (B)
ohne Prädiktion	4.2633 (37.4934[dB])	2.8402 (37.4462[dB])
mit Prädiktion	4.1603 (37.5130[dB])	2.7257 (37.4608[dB])
mit Finite-State	4.0695 (37.4935[dB])	2.6217 (37.4597[dB])

Verbesserung ergeben, sind sie sehr effektiv für die Codierung der Foreman-Sequenz. Andererseits ist die gemeinsame Codierung der drei Bewegungsparameter bei der Mother-and-Daughter-Sequenz sehr wirkungsvoll. Insgesamt läßt sich durch die prädiktive Codierung der Bewegungsparameter unter Verwendung des Finite-State-Mechanismus mit einer Zusammenfassung der Bewegungsparameter das Codierverfahren optimieren. Nicht zu verschweigen ist jedoch, daß sich das Codebuch durch die Zusammenfassung der Bewegungsparameter und hohen Speicherbedarf aufweist. Bei dem Suchraum $[-u, w] \times [-u, w] \times [1, M]$ ist jedes Zustandscodebuch von der Größe $l = (u+w) \cdot (u+w) \cdot M$ statt der bisherigen Größe (u+w) + (u+w) + M.

In Abb. 4.1 und Abb. 4.2 sind die Rate-Distortion-Kurven der oben genannten sechs Codierverfahren dargestellt.

An dieser Stelle kommen wir zurück zu der ursprünglichen Frage, ob die Bewegungsschätzung mit der Langzeitprädiktion im Rate-Distortion-Sinn gerechtfertig ist. Um diese Frage zu beantworten, wird die Bewegungsschätzung mit unmittelbar vorangegangenem Bild ebenfalls mit dem iterativen Algorithmus durchgeführt und ein gemeinsames Codebuch der zusammengefaßten Bewegungsparameter v_x und v_y verwendet. Daher kann



Abbildung 4.1: Codierergebnis für Foreman-Sequenz



Abbildung 4.2: Codierergebnis für Mother-and-Daughter-Sequenz

ein Vergleich durchgeführt werden.

In Abb. 4.3 und 4.4 sind die Rate-Distortion-Kurven aus der Codierung mit Langzeitspeicher der Größe 10 und aus der bewegungskompensierten Prädiktion mit dem unmittelbar vorangegangenen Bild zusammengestellt. Für den letzteren Fall werden die linearen und

H.263-artigen Prädiktionsmodelle herangezogen. Für die Langzeitprädiktion werden verschiedene prädiktive Codierverfahren unter Verwendung des Finite-State-Mechanismus untersucht. Die in der Klammer angegebenen Zahlenwerte geben an, wie die Folgezustandsfunktion gewählt wird, wobei (I) der Zustand durch den letzten gewählten Zeitparameter und (II) der Zustand durch den Medianfilter-Prädiktor 3. Grades des Zeitparameters bestimmt wird.

Bei der Foreman-Sequenz weist der Medianfilter-Prädiktor 3. Grades unter allen anderen Codierungsmethoden das beste Rate-Distortion-Verhalten auf, sowohl bei der Langzeitprädiktion als auch bei der Bewegungsschätzung mit nur dem vorhergehenden Bild. Ein besseres PSNR-Ergebnis bei gleichen Bitraten ab 340 Bits/Frame wird bei der Langzeitprädiktion erzielt. Bei der Mother-and-Daughter-Sequenz fallen die Kurven für die beiden Fälle im kleinen Bitraten-Bereich fast zusammen, und daher ist das Rate-Distortion-Verhalten beim Langzeitspeicher in dem gesamten Bitraten-Bereich fast nie schlechter als bei der Prädiktion nur mit dem vorhergehenden Bild. Insgesamt ist das Rate-Distortion-Verhalten bei Langzeitprädiktion für den größeren Bitraten-Bereich überlegen.



Abbildung 4.3: Codierergebnis für Foreman-Sequenz



Abbildung 4.4: Codierergebnis für Mother-and-Daughter-Sequenz

Kapitel 5

Vergleich von Langzeitprädiktion zur blockbasierten Bewegungskompensation mit 8 × 8 Blöcke

Eine andere Möglichkeit, die Verzerrung zu verringern bzw. die PSNR-Werte zu verbessern, stellt die Methode der Verkleinerung der für die Bewegungsschätzung verwendeten Blockgröße dar. Daher wird abschließend noch die Bewegungsschätzung mit der Blockgröße 8×8 statt der bisherigen Blockgröße 16×16 untersucht und diese mit der Langzeitprädiktion verglichen. Dabei bleibt der Suchbereich für die beiden Verschiebungsvektoren v_x, v_y unverändert, nämlich $v_x, v_y \in [-u, w]$, wobei u = 16 und w = 15 gilt.

Abb. 5.1 und 5.2 zeigen die Rate-Distortion-Kurven der beiden Testsequenzen unter Verwendung der beiden Methoden: Bewegungsschätzung mit Langzeitspeicher der Größe 10 und mit der verkleinerten Blockgröße 8×8 . Man erkennt, daß die Methode mit der Langzeitprädiktion eine deutliche Verbesserung für das Rate-Distortion-Verhalten gegenüber der Blockgröße 8×8 zeigt. Während für die großen λ -Werte die Rate-Distortion-Kurve der ersteren gegen 99 Bits/Frame strebt, strebt die Kurve bei der Blockgröße 8×8 gegen 396 Bits/Frame (in diesem Fall gibt es insgesamt 396 Blöcke in einem Bild, für die das Block-Matching-Verfahren durchgeführt wird). Für die kleineren λ -Werte benötigt die Langzeitprädiktion bei gleichen PSNR-Werten wesentlich weniger Bits. Bei der Motherand-Daughter-Sequenz zeigt sie sogar bessere PSNR-Werte bei gleichzeitig niedriger Bitrate.



Abbildung 5.1: Codierungsergebnis für Foreman-Sequenz



Abbildung 5.2: Codierungsergebnis für Mother-and-Daughter-Sequenz

Kapitel 6

Zusammenfassung und Ausblick

Gegenstand der vorliegenden Arbeit war, die blockbasierte bewegungskompensierte Langzeitprädiktion zu untersuchen und ihre Wirksamkeit und Effektivität zu überprüfen. Dabei sollte der Entwurf der neuen Langzeitprädiktoren so durchgeführt werden, daß er im Rate-Distortion-Sinn optimal ist.

Zunächst wurde das Prinzip des Block-Matching-Verfahrens bei der konventionellen bewegungskompensierten Prädiktion mit dem unmittelbar vorhergehenden Bild geschildert. Das ausgehend von diesem Verfahren entwickelte Modell der Langzeitprädiktion wurde anschließend vorgestellt und die dadurch verbesserte bewegungskompensierte Prädiktion anhand des PSNR-Maßes präsentiert.

Anschließend wurde eingehend das Codierverfahren mit der Langzeitprädiktion entwickelt und im Rate-Distortion-Sinn optimiert. Nach einer Einführung in die Vektorquantisierung mit Entropie-Nebenbedingung wurde der Zusammenhang mit der bewegungskompensierten Langzeitprädiktion gezeigt. Aufbauend auf dem aus ECVQ bekannten iterativen Algorithmus wurde das Entwurfskonzept für die Langzeitprädiktion entwickelt und das Rate-Distortion-Verhalten der Codierergebnisse aufgezeigt. Eine weitere Optimierung des Codierverfahrens ließ sich durch die prädiktive Codierung erzielen. Dabei sind verschiedene lineare und nichtlineare Prädiktionsmodelle einzeln untersucht und die Codierergebnisse miteinander verglichen worden. Daraus wurde festgestellt, daß der H.263-artige Prädiktor sich wegen der einfacheren Implementierung und des besseren Codierergebnisses hervorhebt.

Weiterhin wurde durch die genaue Untersuchung des statistischen Verhaltens des Zeitparameters der Finite-State-Mechanismus vorgestellt und für die Codierung angewendet. Außerdem konnte die Zusammenfassung der drei Bewegungsparameter und die gemeinsame Codierung zur weiteren Verbesserung der Codierergebnisse beitragen. Die Codierergebnisse an zwei Testsequenzen zeigen, daß die verschiedenen Codierverfahren unterschiedliche Wirksamkeit auf die Bildsequenzen haben. Während die prädiktive Codierung für Foreman-Sequenz sehr effektiv ist, läßt sich für die Mother-and-Daughter-Sequenz nur duch die Zusammenfassung der Bewegungsparameter eine große Datenreduktion erreichen.

Die in dieser Arbeit vorgestellten verschiedenen Verfahren sollen als Grundlagen für die weitere Entwicklung der Langzeitprädiktion und Vorgehensweisen dienen. Erweiterungen

und Verbesserungen wären:

Schneller Such-Algorithmus

In dieser Arbeit wurde das Blockmatching mit vollständiger Suche vorgenommen, um ein globales Optimum zu erzielen. Die vollständige Suche ist für die bewegungskompensierte Langzeitprädiktion sehr rechenaufwendig, da für alle Pixelpositionen in dem entsprechenden Suchbereich im Langzeitspeicher die Kostenfunktion berechnet und das Minimum ermittelt wird. Das Profiling-Tool zeigt, daß die etwa 7-fache Rechenzeit für die Durchführung der Bewegungsschätzung mit Langzeitspeichergröße von 10 Bildern im Gegensatz zur Bewegungsschätzung mit dem unmittelbar vorangegangen Bild benötigt wird. Daher ist ein schneller Such-Algorithmus für die Ermittlung der optimalen Bewegungsparameter notwendig.

Unterschiedliche Behandlung der Blöcke

Bisher wurden alle Blöcke einer Bildesequenz gleich behandelt und mit einem Codierverfahren bearbeitet. Es ist jedoch zu beachten, daß das bewegte Objekt und der starre Hintergrund starke Unterschiede in den statistischen Eigenschaften zeigen. Wenn der Hintergrund vollkommen statisch ist, wie es beim Bildtelefon mit fest aufgestellter Kamera öfters der Fall ist, ist für die Hintergrundsblöcke eine Langzeitprädiktion völlig überflüssig. Außerdem wurde in der Arbeit beobachtet, daß die Bewegungsparameter eines einzelnen Blocks ausgeprägte Cluster aufweisen, die zur Codierung nützlich sein können, jedoch nach der Mittelung aller Blöcke verschwunden sind. Es ist daher sinnvoll, ein Kriterium zur Unterscheidung verschiedener Blöcke zu finden, um das Codierverfahren weiter zu optimieren.

Untersuchung des Rate-Distortion-Verhaltens unterschiedlicher Langzeitspeichergrößen In dieser Arbeit wurde das Rate-Distortion-Verhalten unterschiedlicher Codierverfahren mit der Langzeitspeichergröße von 10 Bildern untersucht. Anhand der in Abb. 2.5 gezeigten Kurve der Abhängigkeit zwischen PSNR-Werte und der Langzeitspeichergröße ist jedoch eine weitere Erhöhung der PSNR-Werte zu erreichen, die andererseits auch mehr Bits benötigt. Die Abhängigkeit zwischen der Langzeitspeichergröße und dem Rate-Distortion-Verhalten sollte weiter untersucht werden.

Gemeinsame Optimierung

In dieser Arbeit wurden die optimalen Bewegungsparameter des jeweiligen Blocks nach dem Kriterium der Minimierung der Lagrange-Kostenfunktion unabhängig voneinander sukzessiv berechnet. Die ermittelten Bewegungsparameter haben jedoch Einfluß auf die Ergebnisse der nachfolgenden bewegungskompensierten Prädiktion, da die Bewegungsschätzung der nachfolgenden Blöcke ausgehend von dem Prädiktor, der aus den vorangegangenen Bewegungsparametern ermitteltet wird, durchgeführt wird. Im Laufe der Arbeit wurde öfters beobachtet, daß die optimalen Bewegungsparameter eines bestimmten Blocks ein ungeeigneter Startwert zur Bewegungssuche der nachfolgenden Bewegungsparameter sind und ein schlechteres Gesamtergebnis verursachen können. Um ein globales Optimum zu erzielen, damit die optimalen Bewegungsparameter nicht nur für die bewegungskompensierte Prädiktion des jeweiligen Blocks, sondern auch für die nachfolgenden Blöcke optimal ist, ist das Trelliscodierverfahren mit Viterbi-Algorithmus eine mögliche Lösung. In diesem Verfahren werden alle k Blöcke gleichzeitig codiert und die gesamte Kostenfunktion innerhalb dieser k Blöcke minimiert. Das genannte Verfahren kann jedoch die Komplexität und den Rechenaufwand weiter erhöhen.

Variable Blockgröße

Der Ansatz des Block-Matching-Verfahrens ist, daß alle Pixel innerhalb des Blocks die gleiche Bewegung aufweisen. Große Prädiktionsfehler werden daher durch die Interpretation des nicht einheitlichen Blocks, z.B. Blockinhalte mit Objektkannten, stark detaillierten Bildbereich oder unterschiedlich bewegte Objekte, mit einem Bewegungsvektor verursacht. In diesem Fall ist eine variable Blockgröße sinnvoll. Dabei läßt sich die Blockgrößen-Struktur in effizienter Weise mittels eines quadtrees codieren.

Sub-Pel-Genauigkeit

Aus Rechenzeitgründen wurden die bisherigen Untersuchungen der Langzeitprädiktion in Integer-Pel-Genauigkeit vorgenommen. Für die Abwägung zwischen der verbesserten Genauigkeit aus der Sub-Pel-Suche und der dadurch erhöhten Bitrate wegen der Mitteilung der Sub-Pel-Position sollten weitere Untersuchungen angestellt werden.

Literaturverzeichnis

- [Abu90] Hüseyin Abut, editor. Vector Quantization. IEEE Press, 1990.
- [AS94] R. Armitano and R. W. Schafer. Linear Predictive Coding of Motion Vectors. In Proceedings of the Picture Coding Symposium, pages 233-236, Sacramento, USA, September 1994.
- [CEG96] P.A. Chou, M. Effros, and R.M. Gray. A vector quantization approach to universal noiseless coding and quantization. *IEEE Transactions on Information Theory*, pages 1109–1138, July 1996.
- [CKS96] W. Chung, F. Kossentini, and M. Smith. An Efficient Motion Estimation Technique Based on a Rate-Distortion Criterion. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Atlanta, USA, May 1996.
- [CLG89] P.A. Chou, T. Lookabaugh, and R.M. Gray. Entropy-constrained vector quantization. IEEE Transactions on Acoustics, Speech and Signal Processing, pages 31-42, January 1989.
- [Dav72] L.D. Davisson. Rate-Distortion-Theory and Application. *Proceedings of the IEEE*, July 1972.
- [DG85] M.O. Dunham and R.M. Gray. An Algorithm for the Design of Labeled-Transition Finite-State Vector Quantizers. *IEEE Transactions on Communi*cations, 33, January 1985.
- [FGD85] J. Foster, R.M. Gray, and M.O. Dunham. Finite-State Vector Quantization for Waveform Coding. *IEEE Transactions on Information Theory*, 31, May 1985.
- [Gir94] Bernd Girod. Rate-constrained motion estimation. In SPIE Symposium on Visual Communications and Image Processing, pages 1026–1034, September 1994.
- [Gir95] Bernd Girod. Bildkommunikation 1,2; Skript zur Vorlesung. 1995.
- [Gra84] R.M Gray. Vector quantization. *IEEE ASSP Mag.*, pages 4–29, 1984.
- [ITU93] ITU-T Recommendation H.261. Video Codec for Audiovisual Services at p*64kbits. Draft, December 1993.

- [ITU95] ITU-T Recommendation H.263. Video Coding for Low Bitrate Communication. Draft, December 1995.
- [JJ81] J.R. Jain and A.K. Jain. Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communications*, pages 1799– 1808, 1981.
- [KLSW96] F. Kossentini, Y.-W. Lee, M. J. T. Smith, and R. Ward. Predictive RD-Constrained Motion Estimation for Very Low Bit Rate Video Coding. IEEE Journal on Selected Areas in Communications, 1996. Submitted for publication.
- [KSB95] F. Kossentini, M. J. T. Smith, and C. F. Barnes. Image Coding Using Entropy-Constrained Residual Vector Quantization. *IEEE Transactions on Image Pro*cessing, pages 1349–1357, October 1995.
- [LBG80] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, pages 84–95, 1980.
- [LL96] S.J. Lee and C.W. Lee. Conditional-entropy-constrained trellis-searched vector quantization for image compression. Signal Processing: Image Communication, pages 99–104, August 1996.
- [LLF94] H. Li, A. Lundmark, and R. Forchheimer. Image sequence coding at very low bitrates: a review. *IEEE Transactions on Image Processing*, pages 589–609, 1994.
- [LMM94] M. Lightstone, D. Miller, and S. K. Mitra. Entropy-Constrained Product Code Vector Quantization with Application to Image Coding. Proceedings of the IEEE International Conference on Image Processing, pages 623-627, November 1994.
- [LS95] W. Li and E. Salari. Successive elimination algorithm for motion estimation. *IEEE Transactions on Image Processing*, pages 105–107, Jan 1995.
- [MFRT94] R.M. Mersereau, S.H. Fatemi, C.H. Richardson, and K.K. Truong. Methods for low bit-rate video compression: some issues and answers. In SPIE Conference on Video Communication and Image Processing, pages 2–13, 1994.
- [Ohm95] J. Ohm. Digitale Bildcodierung. Springer Verlag, 1995.
- [OS94] M.T. Orchard and G.J. Sullivan. Overlapped block motion compensation: An estimation-theoretic approach. *IEEE Transactions on Image Processing*, pages 693–699, September 1994.
- [SB91] G.J. Sullivan and R.L. Baker. Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks. In *IEEE Global Telecomm. Conf. (GLOBECOM)*, volume 3, pages 85–90, 1991.