

Friedrich-Alexander-Universität Erlangen-Nürnberg

**Lehrstuhl für Multimediakommunikation und
Signalverarbeitung**

Prof. Dr.-Ing. André Kaup

Studienarbeit

**Inter-View-Bewegungsprädiktion für einen
waveletbasierten Multiview Videocoder**

von Johannes Petzold

2009

Betreuer: Dipl.-Ing Jens-Uwe Garbas

Erklärung

Ich versichere, dass ich die vorliegende 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.

Ort, Datum

Unterschrift

Inhaltsverzeichnis

Kurzfassung	III
Abkürzungsverzeichnis	IV
Formelzeichen	V
1 Einleitung	1
2 Multiview Videocodierung - Grundlagen	3
2.1 Aufbau und Funktionsweise des Motion Compensated Wavelet Coders (MCW)	3
2.1.1 Lifting-Struktur	4
2.2 Örtliche Wavelet-Transformation	6
2.3 Bewegungsschätzung in zeitlicher Richtung	7
2.3.1 Blockmodi und deren Kostenfunktion	8
2.3.2 Codierung der Makroblockmodi	9
2.3.3 Codierung der Bewegungsvektoren	11
2.4 Multi-View-Videocodierung bezüglich der Ansichtenachse	12
3 Voruntersuchung zur Inter-View-Bewegungsprädiktion	15
3.1 Kompensation der Disparitäten	16
3.1.1 Globale Phasenkorrelationsmethode	16
3.1.2 Mean Squared Error	17

3.1.3	Interpolation zwischen benachbarten Blöcken bei nicht ganzzahliger Blockverschiebung	18
3.2	Ergebnisse	20
3.3	Reflexion	24
4	Einführung eines neuen Makroblock Modus: VIEWSKIP	25
4.1	Beschreibung des VIEWSKIP Modus	25
4.2	Ergebnisse	26
4.2.1	Grundlagen	26
4.2.2	Simulationsergebnisse mit VIEWSKIP Modus	27
5	Einführung des VIEW-Prädiktor-Verfahren	33
5.1	Beschreibung des VIEW-Prädiktor-Verfahren	33
5.1.1	Berechnung des VIEW-Prädiktors	34
5.1.2	Explizite Signalisierung des VIEW-Prädiktors	35
5.1.3	Implizite Signalisierung des VIEW-Prädiktors	39
6	Kombination von VIEWSKIP Modus und VIEW-Prädiktor Verfahren	45
6.1	Beschreibung	45
6.2	Ergebnisse	45
7	Zusammenfassung	49
	Abbildungsverzeichnis	51
	Tabellenverzeichnis	52
	Literaturverzeichnis	54

Kurzfassung

In dieser Arbeit wird untersucht, inwiefern sich die Datenrate für die Übertragung von Bewegungsinformation in einem waveletbasierten Multiview Videocoder einsparen lässt. Dabei wird die Ähnlichkeit benachbarter Kameraansichten ausgenutzt. Wichtig ist die ausschließliche Betrachtung der zeitlichen Transformation. Zuerst erfolgt eine Voruntersuchung zur Inter-View-Bewegungsprädiktion, wobei vor allem die unterste Stufe der Lifting-Struktur genauer analysiert wird. Bei der Kompensation der Disparität kommt eine globale Phasenkorrelationsmethode zum Einsatz. Anschließend wird mit den Ergebnissen der Voruntersuchung der MCW-Coder um den neuen VIEWSKIP-Makroblockmodus erweitert. Dieser übernimmt die bereits übertragene und disparitätskompensierte Bewegungsinformation aus der Nachbaransicht. Dadurch lässt sich bereits bei niedrigen Datenraten ein Gewinn erzielen. Als Nächstes wird geprüft, ob sich die disparitätskompensierte Information aus der Nachbaransicht als Prädiktor eignet. Dazu wird der zusätzliche Signalisierungsaufwand für die jeweiligen Modi untersucht. Um eine implizite Signalisierung zu umgehen, wird versucht, das Entscheidungskriterium zwischen örtlichem- und VIEW-Prädiktor mit verschiedenen Ansätzen optimal nachzubilden. Zuletzt wird die Kombination zwischen implizierter Signalisierung des Prädiktors und dem VIEWSKIP-Modus betrachtet.

Abkürzungsverzeichnis

DCVF	Diversity Compensated View Filtering
MAP	Motion Aligned Predict
MAU	Motion Aligned Update
MCTF	Motion Compensated Temporal Filtering
MCW	Motion Compensated Wavelet
MSE	Mean Squared Error
PSNR	Peak Signal to Noise Ratio
SAD	Summe der absoluten Differenzen
UVLC	Universal Variable Length Coding

Formelzeichen

$\lfloor \cdot \rfloor$	Abrundung
\ln	Natürliche Logarithmus zur Basis e
\lg_{10}	Logarithmus zur Basis 10
Σ	Summenzeichen
$ \cdot $	Betrag
$\mathcal{F}^{-1} [\cdot]$	Inverse Fouriertransformation
$(\cdot)^*$	Komplex konjugiert
δ	Delta-Funktion
$\text{GDC}\{MVF\}$	Global Disparity Compensated Motion Vector Field
\wedge	Verundung
\vee	Veroderung

Kapitel 1

Einleitung

Die Entwicklung neuerer Bildschirmtechnologien eröffnet neue Möglichkeiten im Bereich des 3D-Sehens. Die Aufnahme einer Sequenz mit mehreren Kameras (Multi-Kameraarray) kann zur Darstellung auf speziellen Bildschirmen genutzt werden, um dem Betrachter eine dreidimensionale Wahrnehmung zu ermöglichen. Ein Beispiel dafür ist das 3D-TV, das in Zukunft unseren Haushalt bereichern wird. Mit den Video-Daten eines Multi-Kameraarrays lässt es sich auch realisieren, die Sequenz auf einem Bildschirm aus verschiedenen Blickwinkeln beobachten zu können.

Beim Aufnehmen mit einem Kamerarray entstehen mehrere Videosequenzen. Damit kommt es bei der Multiview Videocodierung im Gegensatz zur herkömmlichen Videocodierung zu mehr Information, die verarbeitet und übertragen werden muss. Dennoch bieten die benachbarten Kameransichten eine gewisse Ähnlichkeit, da zum Beispiel die Kameras nur um einen gewissen Offset verschoben sind. Die mögliche Verringerung der Datenrate durch die Berücksichtigung der Bewegungsinformation aus der benachbarten Kameransicht wird in dieser Arbeit untersucht.

Im zweiten Kapitel werden zuerst die Grundlagen der Multiview-Videocodierung erläutert. Dabei wird die Funktionsweise des vorhandenen MCW-Coders aufgezeigt und auf die zeitliche Transformation, dem Motion Compensated Temporal Filtering (MCTF), eingegangen. Die ebenfalls vorhandene Transformation hinsichtlich der Ansichtenachse, dem Disparity Compensated View Filtering (DCVF), wird nur kurz beleuchtet, da in dieser Arbeit aus-

schließlich die MCTF verwendet wird.

Kapitel 3 beschäftigt sich mit einer Voruntersuchung zur Inter-View-Bewegungsprädiktion. An dieser Stelle wird eine globale Phasenkorrelationsmethode zur Kompensation der Disparitäten verwendet. Es wird überprüft, inwieweit die Berücksichtigung der Bewegungsinformation der Nachbaransicht von Vorteil ist.

In Kapitel 4 wird der Makroblock Modus "VIEWSKIP" eingeführt. Dieser übernimmt für den aktuellen Makroblock die disparitätskompensierte Bewegungsinformation aus der Nachbaransicht. Der Modus wird genauer beschrieben und die Ergebnisse der Simulation aufgezeigt.

Im folgenden Kapitel wird das neue Verfahren "View-Prädiktor" untersucht. Um die Datenrate weiter zu verbessern erfolgt zunächst eine explizite Signalisierung zwischen örtlichen und VIEW-Prädiktor. Anschließend wird versucht durch eine implizite Signalisierung die optimale Entscheidung nachzubilden.

Zuletzt wird die Kombination von VIEWSKIP-Modus und View-Prädiktor Verfahren beschrieben und anschließend die Simulationsergebnisse dargestellt.

Kapitel 2

Multiview Videocodierung - Grundlagen

2.1 Aufbau und Funktionsweise des Motion Compensated Wavelet Coders (MCW)

Eine bewegte Szene wurde mit einem Kameraarray aus mehreren Ansichten aufgenommen (Abbildung 2.1). Anschließend wird die Szene mit einem skalierbaren waveletbasierten Mul-

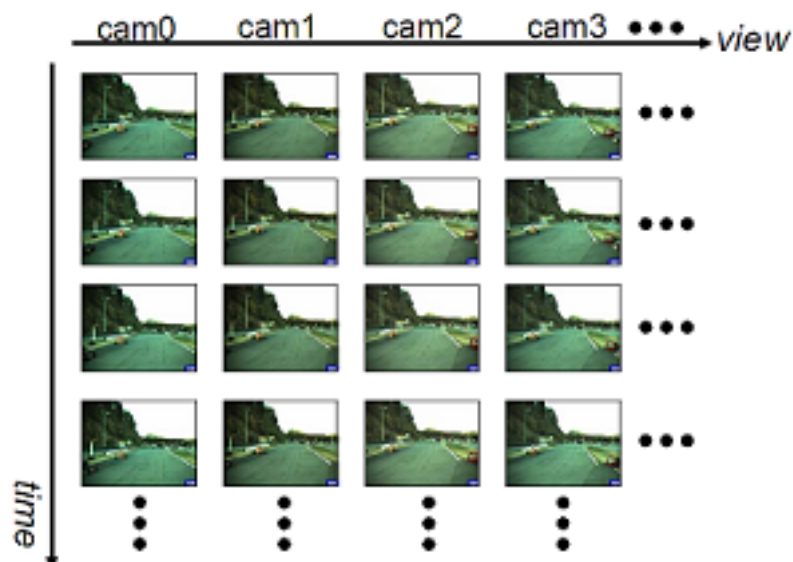


Abbildung 2.1: Multiview Videosequenz, nach [1]

tiview Videocodec codiert. Jede Sequenz einer Kameraansicht wird zuerst zeitlich mit dem Motion Compensated Temporal Filtering (MCTF) zerlegt. Nachfolgend findet eine örtliche örtliche Zerlegung des Videos mit der Wavelet-Transformation statt, um eine Skalierbarkeit bezüglich der Auflösung zu erreichen. Zuletzt werden die Daten entropiecodiert und in einem Bitstream geschrieben, der dann auf Empfängerseite wieder decodiert werden kann.

2.1.1 Lifting-Struktur

Bei der Zerlegung in zeitlicher Richtung kommt eine sogenannte Lifting Struktur (Abbildung 2.2) zum Einsatz. Hier werden die fortlaufenden Bilder einer Videosequenz abwechselnd

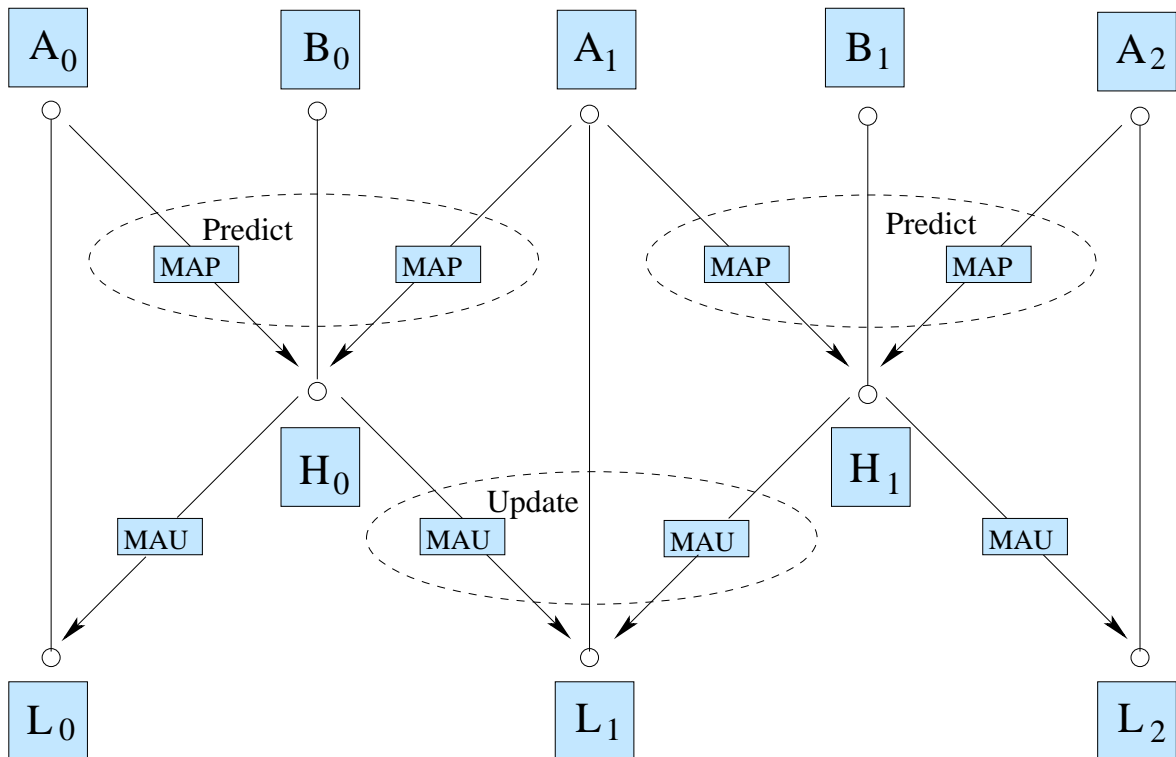


Abbildung 2.2: Lifting-Struktur der MCTF, nach [5]

in A- und B-Bilder aufgeteilt, also $A_0, B_0, A_1, B_1, \dots$. Die Bewegung zwischen den Bildern wird entlang von Bewegungstrajektorien prädiziert. Dies geschieht in der Lifting-Struktur im Motion Aligned Predict-Schritt (MAP). Die B-Bilder werden jeweils aus dem vorherigen

und dem nachfolgenden A -Bild prädiziert. Dabei muss die Bewegung geschätzt werden, um sie dann kompensieren zu können. Wie die verwendete bidirektionale Bewegungsschätzung funktioniert, wird in Abschnitt 2.3 genauer betrachtet. Gemäß der Formel 2.1 entsteht so zu jedem Bild B_i ein Hochpassteilband H_i (H -Bild), welches als Prädiktionsfehlerbild bezeichnet wird.

$$\begin{aligned} H_i &= B_i - \frac{1}{2} [\text{MAP}(A_i) + \text{MAP}(A_{i+1})] \\ L_i &= A_i + \frac{1}{4} [\text{MAU}(H_{i-1}) + \text{MAU}(H_i)] \end{aligned} \quad (2.1)$$

für $i = 0, 1, 2, \dots \left\lfloor \frac{\text{Anzahl Bilder} - 2}{2} \right\rfloor$

Im anschließenden Motion Aligned Update-Schritt (MAU) wird die Bewegungskompensation wieder rückgängig gemacht. Dadurch erhält man die Tiefpassteilbänder L_i (L -Bild). Diese ergeben zusammen die einmal zeitlich skalierte Videosequenz (halbe Bildwiederholrate). Bei einer Szene mit einer Länge von 16 Bildern entstehen somit acht L -Bilder nach der ersten Zerlegung, da sie bezüglich der zeitlichen Richtung um den Faktor $\frac{1}{2}$ skaliert wird. Mit den Tiefpassteilbändern L_i kann wieder ein MAP- und MAU-Schritt erfolgen. Abhängig von der Anzahl der Bilder einer Sequenz können

$$T = \left\lfloor \frac{\ln(\text{Anzahl Bilder})}{\ln(2)} \right\rfloor \quad (2.2)$$

zeitliche Filterungen angewandt werden. Gemäß Abbildung 2.3 erhält man eine vierstufige, zeitliche Zerlegung bei einer Sequenz von 16 Bildern, die in grau dargestellt sind. Die roten Bilder entsprechen den Tiefpass-, die blauen den Hochpassteilbändern. Die Zerlegungen werden als Stufen bezeichnet, wobei die erste Zerlegung der nullten Stufe entspricht.

Da die L -Bilder nach der Entropiecodierung in einem anderen Bitstream Layer als die H -Bilder gespeichert werden können, ist es auf Empfängerseite möglich, zu entscheiden mit welcher Bildwiederholrate wieder decodiert werden soll. So benötigt man für die volle zeitliche Auflösung alle Tiefpass- sowie Hochpassteilbänder, um die A - und B -Bilder wieder vollständig herzustellen. Hingegen werden bei einer skalierten Variante mit dem Faktor $\frac{1}{2}$ die L - und H -Teilbänder die Stufen eins bis drei benötigt.

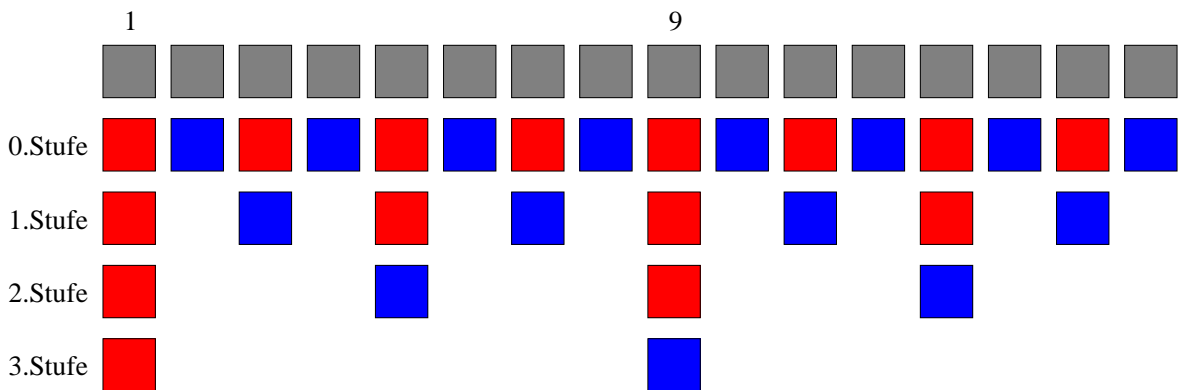


Abbildung 2.3: mehrstufige Lifting Struktur einer Ansicht

Weshalb die mehrstufige Lifting-Struktur bei der Inter-View-Bewegungsschätzung eine wichtige Rolle spielt, behandelt Kapitel 3.

2.2 Örtliche Wavelet-Transformation

Die anschließende örtliche Zerlegung wird mit der Wavelet-Transformation durchgeführt. Wie in Abbildung 2.4 zusehen ist, erhält man nach der Transformation eines Bildes die vier Teilbänder LL , LH , HL und HH . Die letzten drei genannten ergeben die Hochpassanteile und

LL	HL
LH	HH

Abbildung 2.4: Örtliche Zerlegung mit Hilfe der Wavelet-Transformation

können als Zusatzinformation bezeichnet werden. Das Tiefpassanteilband LL entspricht dem in örtlicher Richtung einmal skaliertem originalen Bild mit halber Höhe und halber Breite. Diese Zerlegung kann gewissermaßen beliebig oft durchgeführt werden. Nach der Entropiecodierung der Tiefpass- und Hochpassanteilbänder können diese so im Bitstrom gespeichert

werden, dass man sie wieder getrennt extrahieren kann. Somit kann auf Empfängerseite entschieden werden, mit welcher örtlichen Auflösung decodiert werden soll. Wird allerdings wieder die volle Auflösung benötigt, müssen die Tiefpass- als auch die Hochpassanteile decodiert werden.

Da die Bewegungsschätzung immer bei voller örtlicher Auflösung stattfindet, und die örtliche Zerlegung somit keinen Einfluss auf die Inter-View-Bewegungsschätzung hat, wird sie im nachfolgenden nicht weiter beachtet.

2.3 Bewegungsschätzung in zeitlicher Richtung

Wie bereits erwähnt benutzt die MCTF eine bidirektionale Bewegungsschätzung. Das zu prädizierte Bild B_i und die benachbarten Bilder A_i und A_{i+1} werden dabei in Makroblöcke aufgeteilt. Die Größe derer sollte ein ganzzahliger Teiler der Höhe und Breite des Bildes sein, da dieses sonst erweitert werden muss (padding). Die Standardgröße des MCW-Coders, mit denen später auch die Simulationen durchgeführt werden, sind 16 x 16 Pixel für die nullte bzw. 32 x 32 Pixel für die anderen Stufen der Lifting-Struktur. In Abbildung 2.5 wird

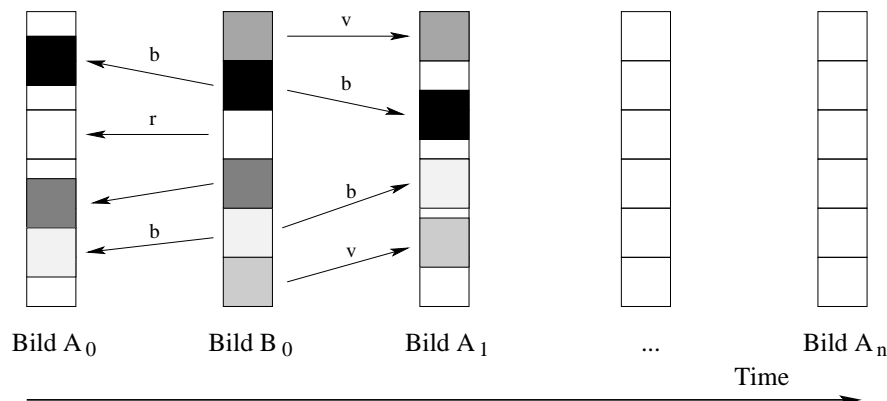


Abbildung 2.5: blockbasierte Bewegungsschätzung, nach [5]

vereinfacht dargestellt, wie eine blockbasierte Bewegungsschätzung funktioniert. So wird zum Beispiel für den ersten Block des Bildes B_0 (von oben gelesen) eine passende Prädiktion im Bild A_1 gefunden. Im Gegensatz zu dieser vorwärtsgerichteten Prädiktion (v) erfolgt

beim dritten Block eine rückwärtsgerichtete (r) Prädiktion zum Bild A_0 . Ebenso kann anhand des schwarzen Blocks gesehen werden, dass ein Block bidirektional geschätzt werden kann. Dabei müssen die gefundenen Blöcke in den Nachbarbildern nicht mit deren Makroblockraster übereinstimmen. Es können also nicht ganzzahlige Blockverschiebungen auftreten. Die bidirektional gefundenen Blöcke werden mit dem Faktor $\frac{1}{2}$ skaliert und dann zusammenaddiert. Bei rückwärts bzw. vorwärts geschätzten Blöcken wird die Gleichung 2.1 so abgeändert, dass A_{i+1} durch A_i oder umgekehrt (je nach Prädiktionsrichtung) ersetzt wird. Sie werden somit nicht skaliert, sondern gehen ganz in die Berechnung ein.

2.3.1 Blockmodi und deren Kostenfunktion

Um die blockbasierte Bewegungsschätzung zwischen den Bildern zu verbessern, kommen im MCW-Coder verschiedene Typen von Makroblock Modi zum Einsatz. Diese sind der DIRECT-, INTER16x16-, INTER16x8-, INTER8x16-, INTER8x8, INTRA- und der mit dieser Arbeit hinzugefügte VIEWSKIP-Modus. Beim DIRECT-Modus wird über die Bewegungsvektoren der bereits codierten Nachbarblöcke gemittelt. Der erhaltene Block wird dann als Prädiktion verwendet. Somit muss nur der Modus signalisiert werden. Der INTER16x16-Modus sucht nach einem komplett passenden Makroblock im vorher- und nachfolgendem Bild. Bei den anderen INTER-Modi, die ebenfalls vorwärts, rückwärts und bidirektionale Bewegungsschätzung erlauben, werden die Makroblöcke in Unterblöcke (Subblöcke) unterteilt. Der INTER8x8 Modus erlaubt es sogar, einen 8x8 Block nochmal in 8x4, 4x8 und 4x4 Blöcke zu unterteilen. Dabei muss jedoch berücksichtigt werden, dass für jeden Block bzw. Subblock mindestens ein Bewegungsvektor (bei bidirektional sogar zwei für vorwärts und rückwärts Prädiktion) übertragen werden muss. Der örtlich arbeitende INTRA-Modus wird standardmäßig nicht verwendet. Als neuer Makroblockmodus wird der VIEWSKIP eingeführt. Dieser übernimmt die bereits vorhandene Makroblockinformation aus der Nachbarsicht. Er wird in Kapitel 4 näher erläutert.

Die Modi müssen miteinander verglichen werden, um den günstigsten Makroblockmodus auswählen zu können. Dies geschieht mit einer Kostenfunktion nach Formel 2.3, wobei für

jeden Modus die Kosten J bestimmt werden.

$$J_{Modus} = SAD + \lambda * R_{Modus} \quad (2.3)$$

Die Summe der absoluten Differenzen (SAD) entspricht dem Maß für die Unterschiedlichkeit des ursprünglichen Blocks O und des prädizierten Blocks P . Die Berechnung erfolgt nach folgender Formel, wobei b und h die Breite und Höhe der Blöcke bezeichnet:

$$SAD = \sum_{x=0}^{b-1} \sum_{y=0}^{h-1} |O(x, y) - P(x, y)| \quad (2.4)$$

R steht für die benötigte Rate der Seiteninformation und enthält die Anzahl der Bits für die Codierung des Makroblockmodus (siehe Abschnitt 2.3.2) und die Anzahl der Bits für die zu übertragende Bewegungsinformation (siehe Abschnitt 2.3.3).

λ ist ein Gewichtungsfaktor, mit dem gesteuert werden kann, inwieweit die Rate mit in die Kostenberechnung eingeht.

2.3.2 Codierung der Makroblockmodi

Nach der Auswahl des günstigsten Modus wird das Universal Variable Length Coding (UVLC) angewandt. Um den entsprechenden Modus zu signalisieren, werden die Modi mit ihrer Prädiktionsrichtung durchnummeriert. Je höher die Nummer N ist, desto größer ist das Codewort und desto mehr Bits werden gebraucht. Ein Codewort beginnt mit einer gewissen Anzahl von Nullen. Diese sind der Signalisierungsanteil und geben an, wie viele binäre In-

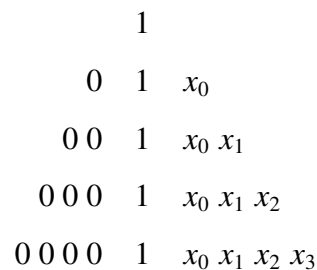


Abbildung 2.6: UVLC Pyramidenform, nach [5]

formationsbits $x_0 \dots x_n$ nach dem Trennungssymbol '1' folgen werden. Das Binärsystem besteht aus einer Kombination von Nullen und Einsen. Gemäß Abbildung 2.6 erhält man bei der Codierung der Nummern eine Pyramidenform.

Die Informationsbits berechnen sich nach folgender Gleichung:

$$Info = N - 2^{\lfloor \frac{L}{2} \rfloor} + 1 \quad (2.5)$$

L steht hier für die Länge des Codeworts. Sollte die Länge des Codeworts 1 sein, wird der Informationsanteil nicht codiert. Zur Veranschaulichung wird es an dem Beispiel der Nummer $N = 4_d$ (Dezimalsystem) aus Tabelle 2.7 verdeutlicht. In Gleichung 2.5 erhalten wir bei $L = 5$, für $Info = 1_d$ oder 01 im Binärsystem. Wird der Binärteil nun zum Signalisierungsanteil hinzugefügt, ergibt das Codewort 00101.

CW	N	L
1	-	0
01	0	3
01	1	3
001	00	5
001	01	5
001	10	5
001	11	5
.	.	.
.	.	.
.	.	.

Abbildung 2.7: UVLC Tabelle, Zusammenhang von Codewort, Nummer und Länge, nach [5]

2.3.3 Codierung der Bewegungsvektoren

Der MCW-Coder verwendet eine differentielle Codierung der Bewegungsvektoren, um diese kostengünstig zu übertragen. Dabei wird vom originalen Bewegungsvektor der örtliche Prädiktor abgezogen. Letzterer kann sowohl auf Encoder- als auch auf Decoder-Seite ohne zusätzliche Bitübertragung mit Hilfe einer Median- und einer richtungsabhängigen Prädiktion berechnet werden.

Median Prädiktion

Für den Block *E* soll mit Hilfe der Median Prädiktion ein Prädiktor gefunden werden (Abbildung 2.8). Der örtliche Prädiktor ist der Median-Wert der drei benachbarten und bereits codierten Bewegungsvektoren der Blöcke *A*, *B* und *C*. Ist der Bewegungsvektor des Blocks *C* nicht vorhanden, wird stattdessen der des Blocks *D* verwendet. Anschließend werden die drei Bewegungsvektoren der Größe nach geordnet und der Mittlere von ihnen (Median) als Prädiktor ausgewählt. Dieses Verfahren wird bei quadratischer Blockaufteilungen der Größe von 16x16, 8x8 oder 4x4 Pixel verwendet.

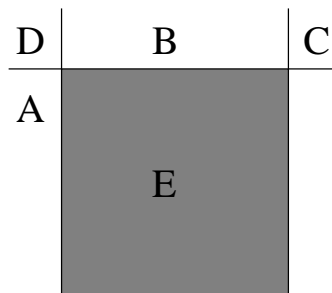


Abbildung 2.8: Median Prädiktion, nach [5]

Richtungsabhängige Prädiktion

Sind die Blöcke nicht quadratisch aufgeteilt (16x8, 8x16, 8x4 oder 4x8), wird dagegen eine richtungsabhängige Prädiktion angewandt. Für einen Block der Pixelgröße 16x8 wird für die obere Teilhälfte der Bewegungsvektor von Block *B* als Prädiktor verwendet. Beim unteren

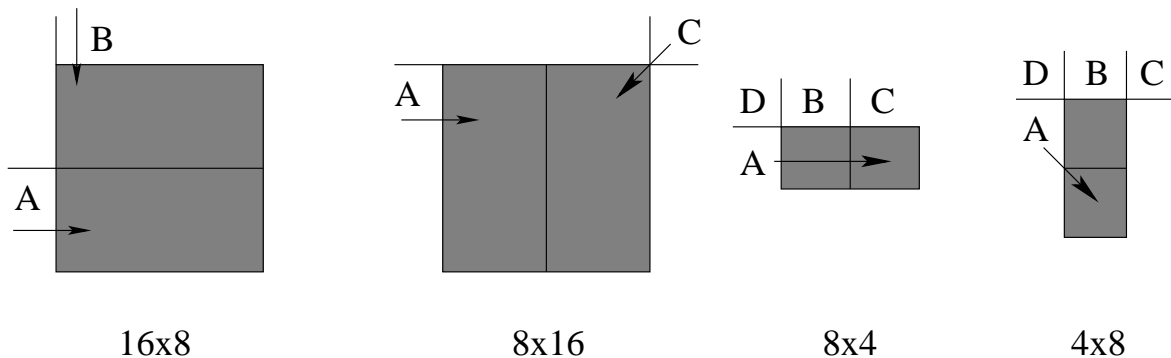


Abbildung 2.9: Richtungsabhängige Prädiktion, nach [5]

Teilblock wird der Bewegungsvektor des Blocks A zum Prädiktor. Ähnlich verhält es sich mit einer Blockaufteilung der Größe von 8x16 Pixel. Für den linken Teilblock wird der Bewegungsvektor von Block A und für den rechten Teilblock von Block C als Prädiktor gesetzt. Sind die genannten Blöcke nicht vorhanden, wird die Median Prädiktion benutzt.

Etwas anders verhält es sich bei kleinsten Blockaufteilungen (8x4, 4x8). Bei der 8x4 Unterteilung wird der Prädiktor des linken 4x4 Subblocks per Median Prädiktion berechnet und der rechte bekommt den Bewegungsvektor des Blocks A zugeteilt. Ähnliches gilt bei der 4x8 Aufteilung: Der Prädiktor für den oberen 4x4 Teilblock wird durch Median-Prädiktion, der untere durch den Vektor des Blocks A bestimmt.

Der örtliche Prädiktor spielt eine bedeutende Rolle in der Lifting-Struktur und beim View-Prädiktor-Verfahren. Dies wird im nächsten und in Kapitel 5 genauer behandelt.

2.4 Multi-View-Videocodierung bezüglich der Ansichtenachse

Bei der Aufnahme einer Szene mit einem Kameraarray entstehen mehrere Videosequenzen. Diese können bei geringen Abständen der Kameras eine große Ähnlichkeit aufweisen. Der MCW-Coder ist deshalb auch in der Lage eine Transformation hinsichtlich der Ansichtenachse durchzuführen (DCVF). Die Sequenz wird im Hinblick auf die verschiedenen

Ansichten mit der Lifting-Struktur zerlegt. Um eine Ansicht aus der anderen zu präzisieren wird eine blockbasierte Prädiktion verwendet. Es werden dafür, wie in der zeitlichen Bewegungsschätzung, die INTER- und INTRA-Blockmodi ausgewählt. Der später zusätzlich eingeführte VIEWSKIP-Modus sowie das View-Prädiktor-Verfahren könnten prinzipiell, mit kleinen Abänderungen in der DCVF eingesetzt werden. Dabei würde dann nicht ein Sprung zur Nachbaransicht stattfinden, sondern es würde die Disparitätsinformation aus verganginem Zeitschritt als zusätzliche Information verwendet werden. Das wird in dieser Arbeit aber nicht untersucht, da nur die zeitliche Zerlegung (MCTF) für die Inter-View-Bewegungsschätzung betrachtet wird.

Kapitel 3

Voruntersuchung zur Inter-View-Bewegungsprädiktion

Bei der Inter-View-Bewegungsprädiktion dient die Bewegungsinformation der Nachbaransicht als zusätzliche Seiteninformation. Durch sie soll die Datenrate zur Übertragung von Bewegungsinformation bei der waveletbasierten Multiview-Videocodierung verringert werden. Dabei wird die Ähnlichkeit benachbarter Bewegungsvektorfelder ausgenutzt.

Zu Beginn der Arbeit wird untersucht, ob die Inter-View Bewegungsprädiktion lohnenswert ist. Bis jetzt wird jede Ansicht zeitlich mit der MCTF transformiert und die Seiteninformation unabhängig voneinander übertragen. Dabei könnte die bereits übertragene Bewegungsinformation aus der Nachbaransicht für die aktuelle Bewegungsschätzung hilfreich sein.

In der in Abbildung 2.3 dargestellten vierstufigen Lifting-Struktur wird deutlich, dass bei einer Sequenzlänge von 16 Bildern in der dritten Stufe im Prinzip eine Bewegungsschätzung zwischen erstem und neuntem Bild stattfindet. Dies ergibt sich durch die zeitliche Skalierung der mehrstufigen Lifting-Struktur. Bei einer großen Distanz der Bewegungsschätzung kann es vorkommen, dass durch die schnelle Bewegung in einer Szene die Bewegungsvektoren benachbarter Blöcke, in unterschiedliche Richtungen zeigen. Da eine differentielle Codierung der Bewegungsvektoren (siehe 2.3.2) zum Einsatz kommt, könnte der örtliche Prädiktor in der untersten Stufe ungenau sein. Die Bewegungsinformation aus der Nachbaransicht steht

bereits zur Verfügung und könnte somit durch günstigere Bewegungsvektoren die Bewegungsschätzung verbessern. Ein Teil der zu übertragenden Seiteninformation könnte eingespart werden und ein Prädiktionsgewinn wäre möglich.

3.1 Kompensation der Disparitäten

3.1.1 Globale Phasenkorrelationsmethode

Die Bewegungsinformation der Nachbaransicht soll als Zusatzinformation für die jetzige Ansicht benutzt werden. Um die Disparität zwischen den Ansichten zu kompensieren, wird die globale Verschiebung der Nachbaransicht zur aktuellen Ansicht in Pixel berechnet. Die Verschiebung wird mit einer globalen Phasenkorrelationsmethode ermittelt. Das Verfahren beruht auf dem Verschiebungssatz der Fouriertransformation (Shift-Theorem), wobei sich die Verschiebung im Orts-, bzw. Frequenzraum nur durch einen Phasenfaktor im korrespondierenden Bereich äußert.

Gegeben sind die zwei Funktionen $f_1(x, y)$ und $f_2(x, y)$, die sich nur durch eine Translation unterscheiden (Gleichung 3.1).

$$\begin{aligned} f_2(x, y) &= f_1(x + x_0, y + y_0) \\ F_2(u, v) &= F_1(u, v)e^{j(ux_0 + vy_0)} \\ \frac{F_2(u, v)F_1(u, v)^*}{|F_2(u, v)F_1(u, v)^*|} &= e^{j(ux_0 + vy_0)} \\ \mathcal{F}^{-1} \left[\frac{F_2(u, v)F_1(u, v)^*}{|F_2(u, v)F_1(u, v)^*|} \right] &= \mathcal{F}^{-1} \left[e^{j(ux_0 + vy_0)} \right] = \delta(x_0, y_0) \end{aligned} \tag{3.1}$$

Es wird die Fouriertransformation und danach das Kreuzleistungsdichtespektrum berechnet, um die Phase zu isolieren. Die Phasenkorrelation ist definiert als das Maximum der inversen Fouriertransformation des Kreuzleistungsdichtespektrums (nach [2] und [3]) und entspricht dem Peak $\delta(x_0, y_0)$. Er wird anhand Abbildung 3.1 verdeutlicht und ist, ausgedrückt in Pixeln, die Translation in x - bzw. y -Richtung. Die Verschiebung muss als Seiteninformation mit übertragen werden, um auf Decoder Seite die entsprechenden Bewegungsvektorfelder aus der Nachbaransicht um den richtigen Faktor zu verschieben.

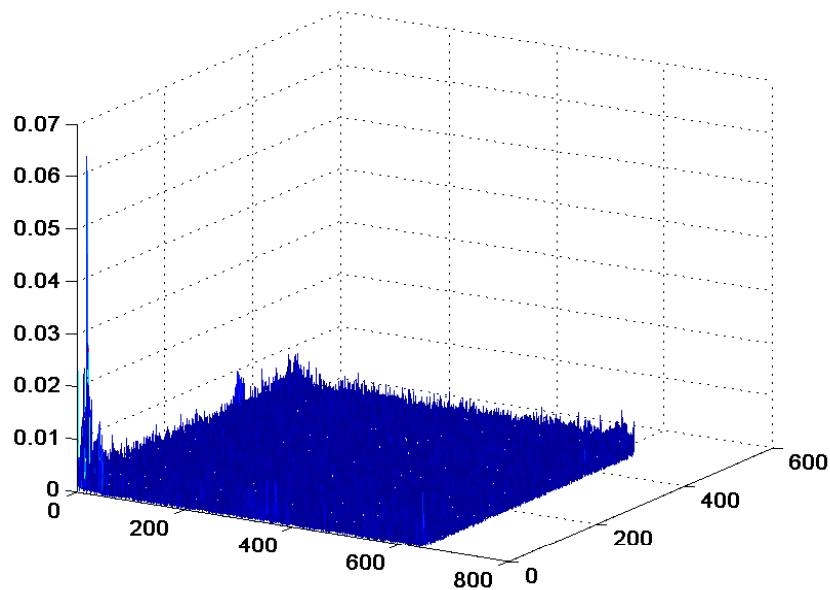


Abbildung 3.1: Peak der Phasenkorrelation bei *ballroom* Sequenz zwischen sechstem Bild von Ansicht 0 und Ansicht 1

Die Phasenkorrelationsmethode wird verwendet, da sie mit wenig Aufwand berechnet werden kann. Ebenso besitzt sie eine geringe Anfälligkeit gegenüber Helligkeitsschwankungen der verschiedenen Ansichten. Somit kann die globale Verschiebung zwischen benachbarten Ansichten sehr gut geschätzt werden.

3.1.2 Mean Squared Error

Eine andere Möglichkeit um die Disparität zwischen zwei Ansichten zu kompensieren ist die Berechnung des mittleren quadratischen Fehlers, dem Mean Squared Error (MSE). Alle Multiview Videosequenzen wurden mit einem Kameraarray aufgenommen, wobei die Kameras nacheinander auf einer horizontalen Ebene angeordnet sind. Daraus folgt, dass sich zwei Bilder (f_1 und f_2) nur um eine Verschiebung bezüglich der x -Richtung unterscheiden. Das MSE ist ein Qualitätsmaß und berechnet sich allgemein nach Gleichung 3.2, wobei f_1

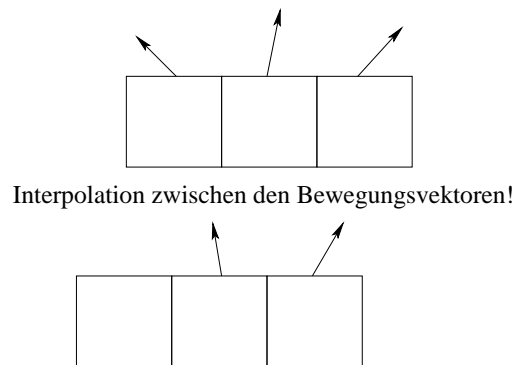


Abbildung 3.3: notwendige Interpolation der Bewegungsvektoren bei nicht ganzzahliger blockverschiebung

Untersucht werden die bilineare, nearest-Neighbour, spline und die kubische Interpolation. Dazu wird die Länge der Vektoren als Maß für eine Kostenberechnung eingeführt (Gleichung 3.3).

$$Q_R = \sqrt{x^2 + y^2} \quad (3.3)$$

Die Kosten Q lassen sich durch die x - und y -Komponente des Vektors R berechnen. Es werden die Kosten für jedes Verfahren pro Bewegungsvektorfeld ermittelt (Addition von Kosten der Bewegungsvektoren) und miteinander verglichen. Die bilineare Interpolation stellt sich als die geeignetste heraus, da sie nicht nur günstig interpoliert, sondern auch sehr einfach nach folgender Formel arbeitet:

$$\begin{aligned} f(R_1) &\approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \text{ mit } R_1 = (x, y_1) \\ f(R_2) &\approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \text{ mit } R_2 = (x, y_2) \\ f(P) &\approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \end{aligned} \quad (3.4)$$

Gegeben ist eine Funktion f , die unserem Bewegungsvektorfeld entspricht. Umgeben von den vier Punkten $Q_{11}(x_1, y_1)$, $Q_{12}(x_2, y_1)$, $Q_{21}(x_1, y_2)$ und $Q_{22}(x_2, y_2)$ soll nun der Wert eines Punktes $P(x, y)$ bestimmt werden. Zuerst wird gemäß Abbildung 3.4 der Wert zwischen den Punkten horizontal und mit den erhaltenen Stützpunkten R_1 und R_2 vertikal interpoliert.

Bei der Interpolation können nicht ganzzahlige Werte entstehen. Diese werden zum nächsten

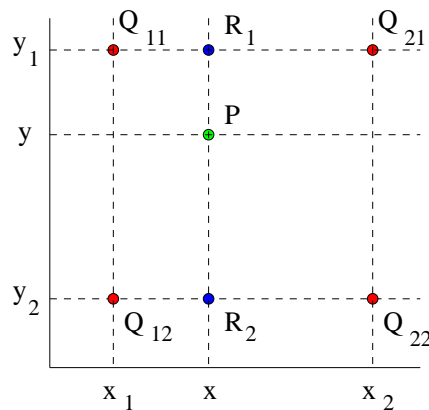


Abbildung 3.4: bilineare Interpolation, nach [4]

ganzzahligen Wert ab- bzw. aufgerundet.

Für weitere Untersuchungen wird bei nicht ganzzahliger Blockverschiebung die bilineare Interpolation verwendet. Die anderen Verfahren werden wegen ihrer komplexeren Berechnung und der Verschlechterung des Interpolationsergebnisses an dieser Stelle nicht weiter erläutert.

3.2 Ergebnisse

Um die Ergiebigkeit der Inter-View-Prädiktion näher zu untersuchen, wird versucht, einen Bezug zwischen benachbarten Ansichten herzustellen. Für das folgende Beispiel werden die Ansichten 0 und 1 der *ballroom* Sequenz (erste 16 Bilder, Auflösung 640x480 Pixel) verwendet und die unterste Stufe der Lifting-Struktur betrachtet. Zwischen den Ansichten wird die Disparität mit Hilfe der Phasenkorrelationsmethode geschätzt. Anschließend wird das entsprechende Bewegungskvektorfeld von Ansicht 0 um $\delta(x_0, y_0)$ verschoben und man erhält ein disparitätskompensiertes Bewegungskvektorfeld $GDC\{MVF_0\}$ (Global Disparity Compensated Motion Vector Field). Das entsprechende Bewegungskvektorfeld von Ansicht 1

wird als MVF_1 bezeichnet.

$$\begin{aligned} \text{Diff}\{MVF\} &= MVF_1 - \text{GDC}\{MVF_0\} \\ \text{Fair}\{MVF_1\} &= MVF_1 - \text{Pred}\{MVF_1\} \end{aligned} \tag{3.5}$$

Um zu sehen, inwieweit sich die beiden Bewegungsvektorfelder ähnlich sind, wird nach Gleichung 3.5 ein Differenz-Bewegungsvektorfeld $\text{Diff}\{MVF\}$ gebildet. Für einen fairen Vergleich zwischen $\text{Diff}\{MVF\}$ und dem Bewegungsvektorfeld aus der aktuellen Ansicht MVF_1 wird von den Bewegungsvektoren von MVF_1 der jeweilige örtliche Prädiktor $\text{Pred}\{MVF_1\}$ abgezogen (differentielle Übertragung).

Anschließend werden die Kosten wieder nach Gleichung 3.3 berechnet. Sie werden für die Bewegungsvektorfelder $\text{Fair}\{MVF_1\}$ und $\text{Diff}\{MVF\}$ im kleinsten Blockraster 4×4 für jeden Block bestimmt und in Abbildung 3.2 dargestellt. Anhand der rechten Farbskala ist zu sehen, wie "teuer" die einzelnen Blöcke sind. Ein dunkelblauer Block verursacht keine, ein dunkelroter dagegen hohe Kosten. Nun werden die Kosten aller Blöcke des Bewegungsvektorfeldes $\text{Fair}\{MVF_1\}$ sowie die des Bewegungsvektorfeldes $\text{Diff}\{MVF\}$ addiert. Vergleicht man die Kosten für beide Bewegungsvektorfelder, so stellt sich heraus, dass $\text{Fair}\{MVF_1\}$ günstiger ist als $\text{Diff}\{MVF\}$. Es kann also, bezüglich des Feldes, kein Gewinn durch die Kompensation der Disparität zwischen Ansicht 0 und Ansicht 1 erzielt werden.

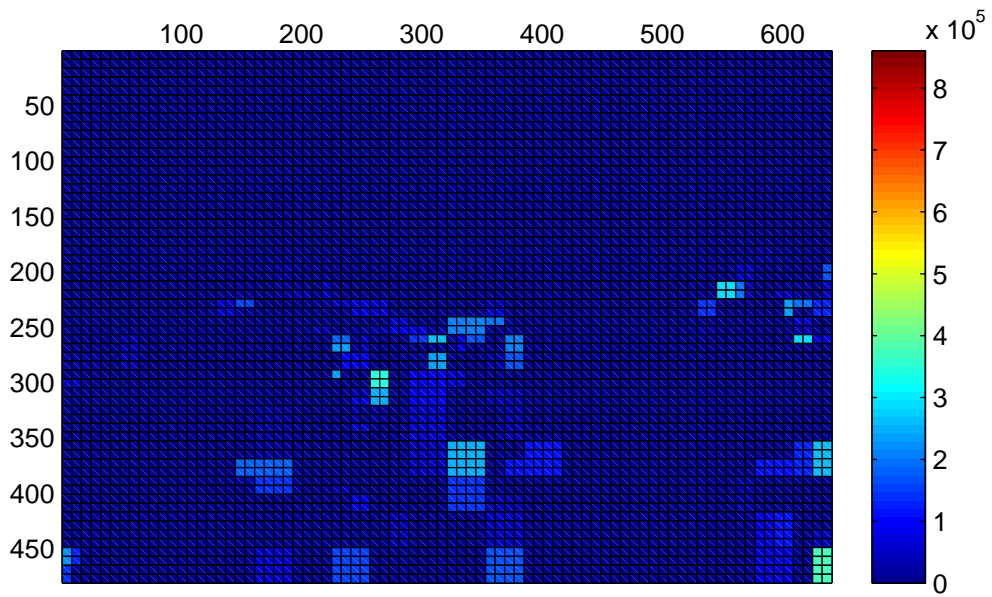
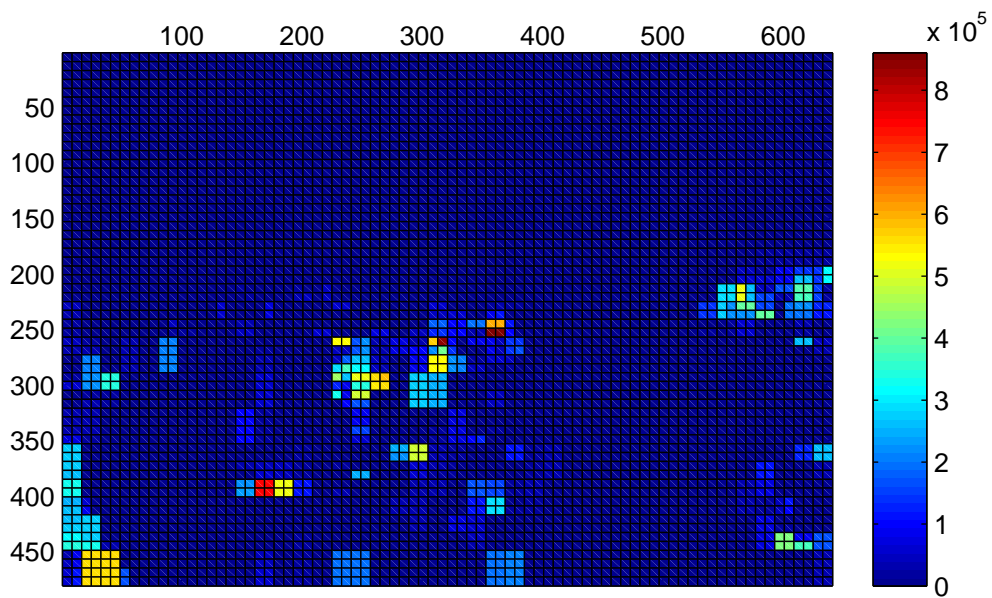
(a) $\text{Fair}\{MVF_1\}$ (b) $\text{Diff}\{MVF\}$

Abbildung 3.5: farblich dargestellte Kosten des im Text genannten Beispiels der *ballroom* Sequenz, x- und y-Achse in Pixel

Dagegen lohnt es sich die Bewegungsvektorfelder in einer blockbasierten Entscheidung miteinander zu vergleichen (Abbildung 3.6).

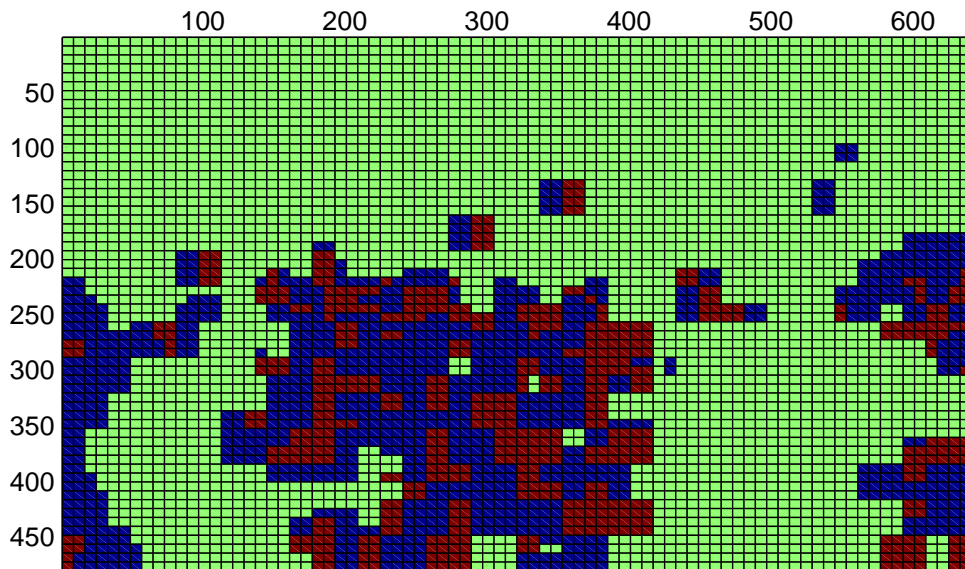


Abbildung 3.6: *ballroom* Sequenz, blockbasierte Entscheidung zwischen $\text{Fair}\{MVF_1\}$ und $\text{Diff}\{MVF\}$, x- und y-Achse in Pixel

Zu sehen sind drei verschiedene Farben. Bei den grünen Blöcken sind die Kosten der beiden Bewegungsvektorfelder gleich. Die blauen Blöcke haben günstigere Kosten im Bewegungsvektorfeld $\text{Fair}\{MVF_1\}$. Rot steht für die geeigneteren Bewegungsvektoren im Bewegungsvektorfeld $\text{Diff}\{MVF\}$. Sie ergeben einen Anteil von $\sim 11.3\%$. Auch wenn in den oberen Stufen der Lifting-Struktur deutlich weniger Blöcke in der disparitätskompensierten Nachbaransicht ausgewählt werden (zum Beispiel $\sim 2\%$ pro Vektorfeld in der nullten Stufe), lohnt es sich gerade in der untersten Stufe den Ansatz der Inter-View-Bewegungsprädiktion genauer zu verfolgen.

Des Weiteren wurden die Sequenzen *race1* (640x480 Pixel) und *ballet* (1024x768 Pixel) untersucht. Mit 16% und 7% günstigeren Kosten im Differenz-Bewegungsvektorfeld in der untersten Stufe der Lifting-Struktur fallen die Ergebnisse ähnlich der *ballroom* Sequenz aus.

3.3 Reflexion

Für die Sequenzen *ballroom*, *race1* und *ballet* wurde zur Kompensation der Disparität zwischen benachbarter Ansichten die globale Phasenkorrelationsmethode verwendet. Sie besitzt ihre Stärken vor allem bei einer globalen x- oder y-Verschiebung. Bei der *ballet* Sequenz ist es so, dass diese nicht von einem linearen, sondern von einem im halbkreis angeordneten Kamerarray aufgenommen wurde. Dadurch unterscheiden sich die benachbarten Ansichten nicht unbedingt durch eine Verschiebung, sondern eher durch eine Scherung. Die Disparität der Ansichten könnte bei dieser Sequenz zum Beispiel mit Hilfe einer affinen Transformation besser kompensiert werden. Dieser Ansatz wird in dieser Arbeit aber nicht untersucht.

Kapitel 4

Einführung eines neuen Makroblock

Modus: VIEWSKIP

4.1 Beschreibung des VIEWSKIP Modus

Dem MCW-Coder wird neben den bekannten Verfahren DIRECT, INTER16x16, INTER16x8, INTER16x16 und INTER8x8 der Makroblock Modus VIEWSKIP hinzugefügt. Wie der Name "VIEWSKIP" schon andeutet, findet bei diesem Verfahren ein Wechsel bzw. ein Sprung zu einer anderen Ansicht statt. Es wird davon ausgegangen, dass die Seiteninformation aus der Nachbaransicht bereits übertragen wurde und für die zeitliche Zerlegung der aktuellen Ansicht zur Verfügung steht.

Der VIEWSKIP Modus übernimmt die Makroblock- und Bewegungsinformation unverändert aus der gleichen Stelle der disparitätskompensierten Nachbaransicht. Die Berechnung der Kosten für das neue Verfahren gleicht der Kostenfunktion der anderen Modi bis auf eine Ausnahme (Gleichung 4.1).

$$J_{VIEWSKIP} = SAD + \lambda * R_{VIEWSKIP} \quad (4.1)$$

Es ändert sich die Rate R . Sie besteht aus der Anzahl der Bits für die Signalisierung des Modus VIEWSKIP und aus null Bits für die Übertragung der Bewegungsinformation (wurde

mit der Nachbaransicht schon übertragen). Der Makroblock wird in vier 8x8 Blöcke aufgeteilt und die Prädiktionsrichtung (vorwärts, rückwärts, bidirektional) aus der Nachbaransicht übernommen.

4.2 Ergebnisse

4.2.1 Grundlagen

Es folgt eine kurze Erklärung zur Interpretation der in Abschnitt 4.2.2 gezeigten Ergebnisse. Um die Qualität zweier Bilder vergleichen zu können wird das bereits in Abschnitt 3.1.2 erwähnte MSE verwendet. Jedoch wird jetzt das Qualitätsmaß zwischen dem originalen Bild A und dem durch den MCW-Coder verzerrten Bild B bestimmt. Anschließend wird das "Peak Signal to Noise Ratio" (PSNR)

$$\text{PSNR} = 10 \cdot \lg_{10}\left(\frac{I_{max}}{\text{MSE}}\right) \quad (4.2)$$

in dB berechnet, wobei I_{max} der maximal mögliche Intensitätswert ist. Je größer der dB-Wert ist, desto ähnlicher sind sich originales und verzerrtes Bild. Da man nun die Qualität einer originalen und einer codierten Sequenz vergleichen möchte, wird das PSNR für jedes einzelne Bild berechnet, aufsummiert und durch die Anzahl der Bilder geteilt (arithmetisches Mittel).

Um die Qualität einer Sequenz mit der benötigten Rate in Verbindung zu setzen, werden so genannte Rate-Distortion-Kurven (kurz: RD-Kurven) betrachtet. Dort wird auf der x -Achse die Bitrate in Kilobit pro Sekunde (kbit/s), auf der y -Achse das mittlere PSNR in dB aufgetragen. Je höher der PSNR-Wert und je niedriger die Bitrate, desto besser arbeitet der Coder. Für die Simulationen in dieser Arbeit wurden die ersten 16 Bilder der Sequenzen *ballet* (1024x768 Pixel), *ballroom* (640x480 Pixel) und *race1* (640x480 Pixel) verwendet. Sie wurden bei voller zeitlicher und örtlicher Auflösung decodiert. Der Kostenfaktor λ wird standardmäßig zu 16, 32, 64, 64 für die nullte bis dritte Stufe gesetzt. Die disparitätskompensierte Seiteninformation der Ansicht 0 wurde für die aktuelle Zerlegung der Ansicht 1 als

Zusatzinformation eingesetzt. Prinzipiell ähneln sich die Ergebnisse der Simulationen für alle Sequenzen, wobei *ballroom* den größten Gewinn erzielt. Anhand dieser Sequenz werden die Ergebnisse ausschließlich für die Luminanz(Grau)-Komponente der Bilder aufgezeigt und interpretiert. Die Chrominanz (Farb)-Komponenten verhalten sich ähnlich.

4.2.2 Simulationsergebnisse mit VIEWSKIP Modus

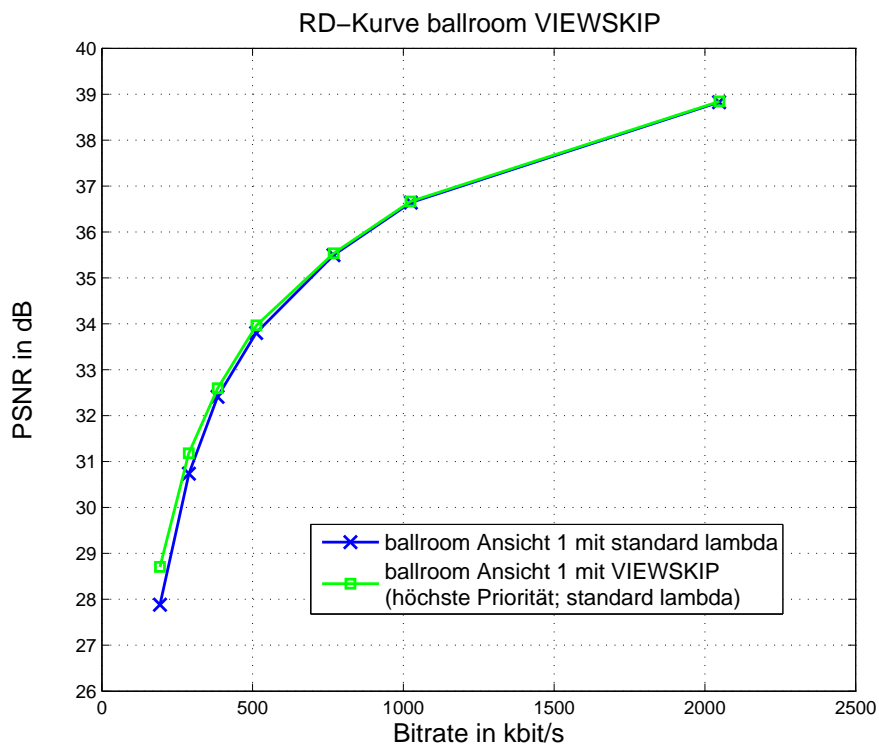


Abbildung 4.1: *ballroom* Sequenz, Vergleich zwischen Referenzkurve und VIEWSKIP-Kurve von Ansicht 1

Die Sequenz *ballroom* wird mit dem MCW codiert und anschließend wieder aus dem Bitstream für verschiedene Bitraten decodiert. Der PSNR-Wert wird berechnet und in einer RD-Kurve verbildlicht. Zunächst wird der VIEWSKIP Modus mit höchster Priorität in den Coder eingebaut und besitzt somit im Vergleich zu anderen Verfahren bei gleichem SAD die höchste Priorität. Abbildung 4.1 zeigt zwei Kurven: Die blaue Kurve entspricht der codierten

Ansicht 1 der *ballroom* Sequenz und dient als Referenz der grünen Kurve. Bei letzter Kurve wurde neben den anderen INTER-Prädiktionsverfahren der VIEWSKIP Modus hinzugefügt. Durch ihn erhält man einen durchschnittlichen Prädiktionsgewinn von ~ 0.24 dB. Auffallend ist, dass die Kurve mit VIEWSKIP nicht mit einem konstanten Wert über der Referenz liegt, sondern dass gerade bei niedrigeren Raten ein großer Gewinn (bis zu ~ 0.83 dB bei circa 250 kbit/s) erzielt wird. Verantwortlich dafür ist die Kostenfunktion, in der nicht nur das SAD sondern auch die Rate mit eingeht. Bei niedrigeren Raten macht sich das Einsparen von einigen Bits der Seiteninformation deutlich bemerkbar, wogegen bei höheren Bitraten ein paar Bits bei der großen Übertragungsrate kaum ins Gewicht fallen. Dagegen bringt ein nicht optimal prädizierter Makroblock einen schlechteren PSNR-Wert mit sich. Deshalb kann es dazu führen, dass bei höheren Raten ($\sim 1000 - 2000$ kbit/s) sogar ein minimaler Verlust im Vergleich zur Referenzkurve entsteht. Der MCW-Coder mit VIEWSKIP Modus ist folglich ein für niedrigere Bitraten optimierter Coder.

in %	Direct	Inter16x16	Inter16x8	Inter8x16	Inter8x8
0.Stufe	85.0	7.5	2.6	2.7	2.2
1.Stufe	72.3	7.7	2.3	3.8	13.9
2.Stufe	69.2	11.5	2.7	4.3	12.3
3.Stufe	64.6	8.7	6.0	5.0	15.7

Tabelle 4.1: Prozentuale Anteile der Modi für die einzelnen Stufen, Referenz

in %	Direct	Inter16x16	Inter16x8	Inter8x16	Inter8x8	VIEWSKIP
0.Stufe	5.2	5.0	1.8	2.2	2.0	83.8
1.Stufe	5.1	4.7	1.3	2.9	10.5	75.5
2.Stufe	5.3	5.3	1.2	2.3	9.0	76.9
3.Stufe	6.0	3.3	3.3	3.7	11.0	72.7

Tabelle 4.2: Prozentuale Anteile der Modi für die einzelnen Stufen, zusätzlich verwendeter VIEWSKIP Modus

Tabelle 4.1 und 4.2 zeigen jeweils die prozentualen Anteile der Modi für die Referenzkurve

und für die Kurve mit VIEWSKIP Modus. Auffallend ist, dass der DIRECT Modus durch den zusätzlichen VIEWSKIP Modus nicht mehr so oft wie in der Referenz verwendet wird. Der VIEWSKIP dagegen wird sehr oft ausgewählt, weil die Übertragungskosten für die anderen Modi durch das Einfügen eines zusätzlichen Modus mit höherer Priorität steigen. So wird bei gleichen Kosten J der VIEWSKIP gegenüber dem DIRECT Modus bevorzugt und vor den anderen INTER-Verfahren verwendet. Obwohl laut Tabelle 4.2 die Auswahl des sehr kostspieligen INTER8x8 Modus ansteigt, gelingt es durch den VIEWSKIP Modus, die Summe der Gesamtkosten aller Stufen um $\sim 1.5\%$ zu verringern, wobei die unterste Stufe mit $\sim 3.3\%$ die größte Einsparung aufweist.

Weiterhin wurde untersucht, wie sich eine Veränderung der Priorität des VIEWSKIP Modus auswirkt. Durch ein Einfügen an niedrigster Stelle kann kein zusätzlicher Prädiktionsgewinn erzielt werden. Da DIRECT und VIEWSKIP Modus prinzipiell die gleichen Kosten für die Rate haben, wurde überprüft, inwieweit ein Ersetzen des DIRECT durch das VIEWSKIP-Verfahren erfolgreich ist. Die Bewegungsinformation aus der disparitätskompensierten Nachbaransicht ist aber oft ungenauer als mit dem DIRECT Modus, weshalb kein zusätzlicher Gewinn erreicht wird.

Wie in Abschnitt 3.1.3 beschrieben, kann es bei der Kompensation der Disparität zu einer nichtganzzahligen Blockverschiebung kommen. An dieser Stelle wird nicht zwischen den Bewegungsvektoren interpoliert, sondern die Verschiebung zur nächsten Blockgröße quantisiert. Untersucht wurde die Quantisierung zum nächsten 4x4/8x8 Sub- und zum ganzzahligen Makroblock. Bei den Simulationen entsteht im Vergleich zur grünen VIEWSKIP Kurve teilweise ein minimaler Prädiktionsgewinn bei niedrigen Bitraten (um die 0.01 dB - 0.05 dB), der dann aber meist zu Verlusten bei hohen Bitraten führt.

Mit der Einführung des VIEWSKIP Modus wurde neben der globalen Phasenkorrelation auch das MSE Verfahren zur Kompensation der Disparität betrachtet. Beide Verfahren ergeben eine fast identische Verschiebung in x-Richtung. Die *ballroom* Sequenz besitzt zudem (laut Phasenkorrelation) eine leichte Verschiebung in y-Richtung. Das MSE unterscheidet jedoch nur den Unterschied der Disparität hinsichtlich der x-Achse. Somit kann auch mit diesem Verfahren keine Verbesserung der VIEWSKIP-RD-Kurve erreicht werden. Deshalb

wird es in den folgenden Simulationen nicht weiter beachtet.

Zuletzt wurde der Standard-Wert für den Kostenfaktor λ verdoppelt, um der Rate ein größeres Gewicht gegenüber dem SAD zu geben. Der VIEWSKIP Modus wird weiterhin mit höchster Priorität eingesetzt. Abbildung 4.2 zeigt, bei doppeltem Kostenfaktor λ , die

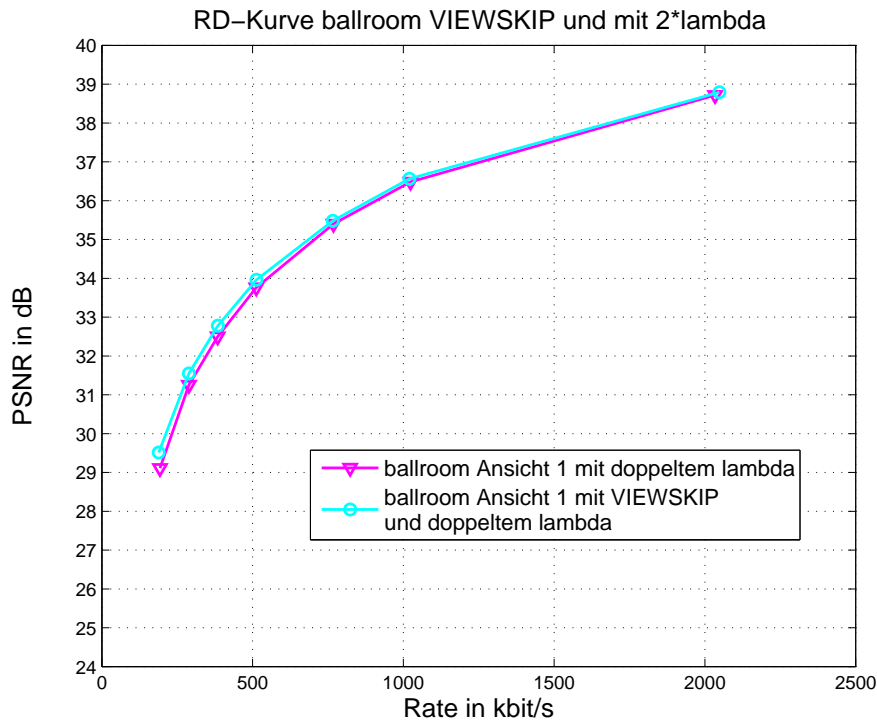


Abbildung 4.2: *ballroom* Sequenz, Vergleich zwischen Referenzkurve und VIEWSKIP-Kurve von Ansicht 1 bei doppeltem Kostenfaktor λ

rosa gefärbte Referenzkurve und die türkise gefärbte VIEWSKIP-Kurve, welche einen durchschnittlichen Gewinn von 0.2 dB erreicht.

Als nächstes wird die türkise Kurve mit der Standard-Referenzkurve und der dazugehörigen VIEWSKIP-Kurve verglichen (Abbildung 4.3). Es fällt auf, dass mit doppeltem λ gerade bei den kleinen Bitraten ein weiterer Gewinn gegenüber der Referenzkurve (+ ~ 1.63 dB) und der VIEWSKIP-Kurve (+ ~ 0.8 dB) mit Standard λ erreicht wird. Dagegen erhält man bei den höheren Raten kaum bzw. nur einen minimalen Verlust von bis zu ~ 0.04 dB. Es kann also durchaus in Betracht gezogen werden, den MCW-Coder mit VIEWSKIP Modus

und doppeltem Kostenfaktor λ einzusetzen.

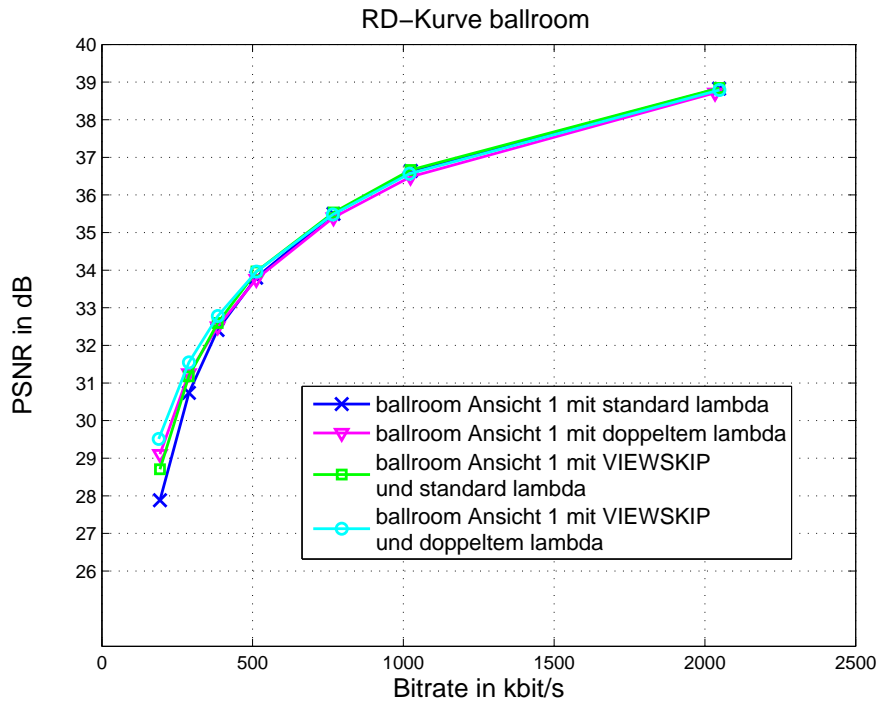


Abbildung 4.3: *ballroom* Sequenz, Vergleich zwischen Referenzkurve (Standard lambda), Referenzkurve (doppeltem lambda), VIEWSKIP-Kurve (standard lambda) und VIEWSKIP-Kurve (doppeltem lambda), Ansicht 1

Kapitel 5

Einführung des

VIEW-Prädiktor-Verfahren

5.1 Beschreibung des VIEW-Prädiktor-Verfahren

Der MCW-Coder verwendet eine differentielle Codierung der Bewegungsvektoren zur Übertragung, wobei der örtliche Prädiktor vom Bewegungsvektor subtrahiert wird. Bei einem großen zeitlichen Abstand zwischen zwei Bildern kann es durch viel Bewegung im Bild vorkommen, dass der Bewegungsvektor eines Blocks zwar gut geschätzt wird, aber ihn sein örtlicher Prädiktor sehr ungenau prädiziert.

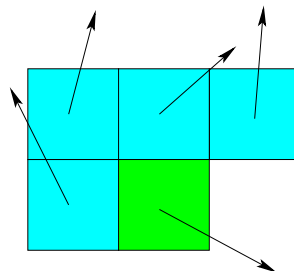


Abbildung 5.1: Situation eines ungeeigneten örtlichen Prädiktors (türkis) für den aktuell zu übertragenden Makroblock (grün)

Abbildung 5.1 veranschaulicht diese Situation, bei der die umliegenden Bewegungsvektoren

in verschiedene Richtung zeigen, und es somit nicht gelingt eine kostengünstige Übertragung der Bewegungsinformation zu gewährleisten. Falls ein ungünstiger örtlicher Prädiktor vorliegt, wird untersucht, ob die disparitätskompensierte Bewegungsinformation der Nachbaransicht besser als Prädiktor geeignet ist. Genau genommen soll der VIEW-Bewegungsvektor den örtlichen Prädiktor ersetzen, um somit die Kosten für die Übertragung weiter zu verringern. An dieser Stelle muss der Signalisierungsaufwand analysiert werden. Der VIEWSKIP Modus wird vorerst nicht eingesetzt. Es werden nur die ursprünglichen Makroblock Verfahren verwendet.

5.1.1 Berechnung des VIEW-Prädiktors

Bei der Berechnung des VIEW-Prädiktors wird der aktuelle Modus, für den die Kosten J_{Modus} ermittelt werden, mit einbezogen. Dessen Makroblock Aufteilung kommt an gleicher Stelle auf den disparitätskompensierten Bewegungsvektoren der Nachbaransicht zum liegen. Der VIEW-Prädiktor wird anschließend als Median-Wert der in diesem Blockraaster vorkommenden Bewegungsvektoren bestimmt, wobei x- und y-Komponente der Bewegungsvektoren getrennt betrachtet werden. Um das Vorgehen zu veranschaulichen, wird die Berechnung der x-Komponente des VIEW-Prädiktors vom INTER16x8 Verfahren als Beispiel betrachtet. Abbildung 5.2 zeigt die Ansicht eins mit der Position des Makroblocks, für den der VIEW-Prädiktor berechnet werden soll. Dieser Makroblock ist in zwei Subblöcke aufgeteilt, die hier grün und blau dargestellt und in einem 4x4 Blockraaster angeordnet sind. Somit muss für jeden Teilblock ein View-Prädiktor gefunden werden. Dazu kommt das Muster an der gleichen Stelle der x-Komponente des disparitätskompensierten Bewegungsvektorfeldes der nullten Ansicht zum liegen. Jeder Wert dort entspricht der x-Komponente eines Bewegungsvektors für einen 4x4 Block. Der Median dieser Werte wird anschließend als x-Komponente des VIEW-Prädiktors für die jeweiligen Subblöcke verwendet. In diesem Beispiel erhält man für den oberen Teilblock den Wert 16, für den unteren den Wert 12 als x-Komponente des VIEW-Prädiktors.

Die y-Komponente des VIEW-Prädiktors wird genauso ermittelt. Für die Berechnung des

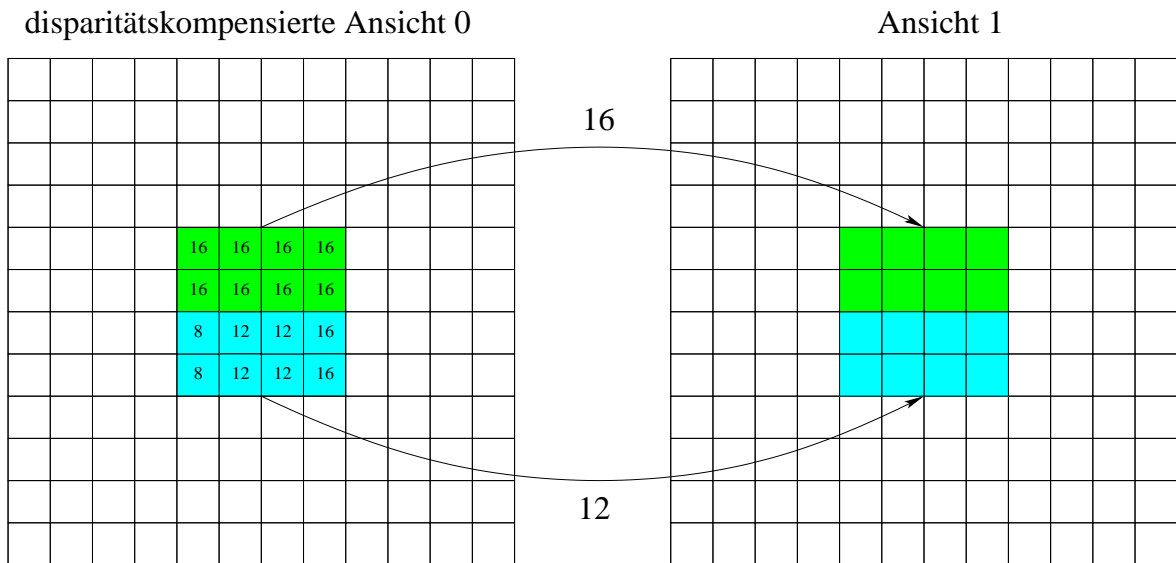


Abbildung 5.2: Veranschaulichung der Berechnung der x-Komponente des VIEW-Prädiktors für einen INTER16x8 Makroblock

VIEW-Prädiktors der anderen Modi wird prinzipiell ebenso vorgegangen.

5.1.2 Explizite Signalisierung des VIEW-Prädiktors

Beschreibung

Als Erstes wird der VIEW-Prädiktor explizit signalisiert. Für jeden Makroblock Modus wird zunächst ein zusätzliches Informationsbit bereitgestellt, um zu entscheiden ob der örtliche Prädiktor oder der VIEW-Prädiktor besser geeignet ist. Es werden die Kosten für jeden Modus jeweils mit beiden Prädiktoren berechnet und anschließend wieder der kostengünstigste von ihnen ausgewählt.

Das zusätzliche Bit für die Signalisierung des VIEW-Prädiktors wird zunächst für alle Stufen der Lifting Struktur angewandt. Bei einer Bildauflösung von 640x480 Pixeln und einer Makroblockgröße von 16x16 Pixeln in der nullten, bzw. 32x32 Pixeln in allen anderen Stufen, müssen 1200 bzw. 300 Bewegungsinformationsbits pro Hochpassbild zusätzlich übertragen

werden. Dies ergibt bei einer Sequenzlänge von 16 Bildern

$$8 * 1200 + 4 * 300 + 2 * 300 + 1 * 300 = 11700$$

Bits und würde ungefähr 20% mehr an Bewegungsinformation entsprechen (Beispiel *ballroom* Sequenz, ca. 56000 Bits an Bewegungsinformation für 16 Bilder und vier stufiger Lifting-Struktur ohne VIEW-Prädiktor Bit). Durch einen sehr gut geeigneten VIEW-Prädiktor kann aber die Anzahl der Bits für die Bewegungsinformation wieder verringert werden.

Ergebnisse

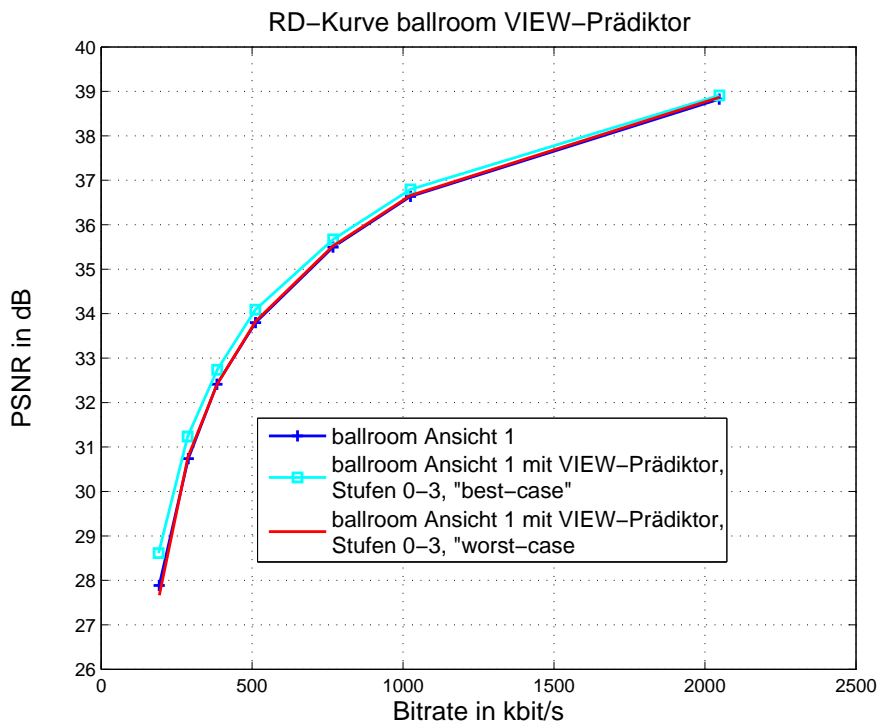


Abbildung 5.3: *ballroom* Sequenz, VIEW-Prädiktor Verfahren mit Stufen 0-3, "worst-case" und "best-case"

Abbildung 5.3 zeigt die blaue Referenz RD-Kurve der Sequenz *ballroom*. Des Weiteren wurde eine "best-case" Kurve (türkis) simuliert, bei der das zusätzliche Bit nicht mit in die Rate

eingeht. Dies ist im Prinzip eine imaginäre Kurve, die im besten Fall erreicht werden kann. Dabei würde man davon ausgehen, dass man auf Empfängerseite ohne eine explizite Signalisierung weiß, wann der VIEW-Prädiktor verwendet werden soll. Dies entspricht einer impliziten Signalisierung und wird im nächsten Abschnitt untersucht. Die rote Kurve dagegen entspricht der sogenannten "worst-case" Kurve. Sie beschreibt den Fall, dass pro Makroblock ein Bit zusätzlich übertragen werden muss. Für alle RD-Kurven wurde der standardmäßige Kostenfaktor λ verwendet.

Die "best-case" Kurve liegt gerade bei niedrigen Raten deutlich über der Referenzkurve (bis zu ~ 0.52 dB) und selbst bei hohen Bitraten kann ein Gewinn von ca. ~ 0.15 dB erzielt werden. Der Durchschnittsgewinn liegt bei ungefähr ~ 0.2 dB. Mit der "worst-case" Kurve kann zwar bei hohen Raten ein minimaler Gewinn erreicht werden, jedoch fällt sie im Bereich der niedrigeren Raten unter ihre Referenz.

Um die "worst-case" RD-Kurve weiter zu verbessern, wird untersucht, inwieweit die Lifting-Struktur mit ihren verschiedenen Stufen eine Rolle spielt. Wie in vorherigen Kapiteln bereits beschrieben wurde, müsste gerade in den unteren Stufen der größte Gewinn möglich sein. Deshalb wird das VIEW-Prädiktor Verfahren nur für die drei unteren Stufen eingesetzt, was sich unmittelbar an den zusätzlich zu übertragenden VIEW-Prädiktor Bits bemerkbar macht. Anstatt 11700 müssen jetzt noch 2100 Bits als zusätzliche Signalisierung übertragen werden. Dies wird in Abbildung 5.4 deutlich. Die "best-case" Kurve aus den Stufen eins bis drei liegt erwartenderweise schlechter als die für "best-case" Kurve für alle Stufen. Wichtiger aber ist, dass mit der "worst-case" RD-Kurve ein durchschnittlicher Gewinn von ~ 0.12 dB gegenüber der Referenzkurve erzielt wird. Ebenso fällt auf, dass die "worst-case" Kurve nur bei den niedrigeren Raten, wo die zusätzlichen Bits mehr ins Gewicht fallen, schlechter gegenüber der "best-case" Kurve ist.

Wird nur in den unteren beiden Stufen der VIEW-Prädiktor eingesetzt, so wird die "worst-case" Kurve minimal schlechter. Aus diesem Grund sind die Stufen eins bis drei der Lifting-Struktur für das VIEW-Prädiktor Verfahren am besten geeignet. Bei den anderen Sequenzen *race1* und *ballet* sind die Ergebnisse ähnlich, wobei es bei letzterer zu erwähnen gibt, dass es durch die höhere Bildauflösung mehr Makroblöcke gibt und somit mehr Bits

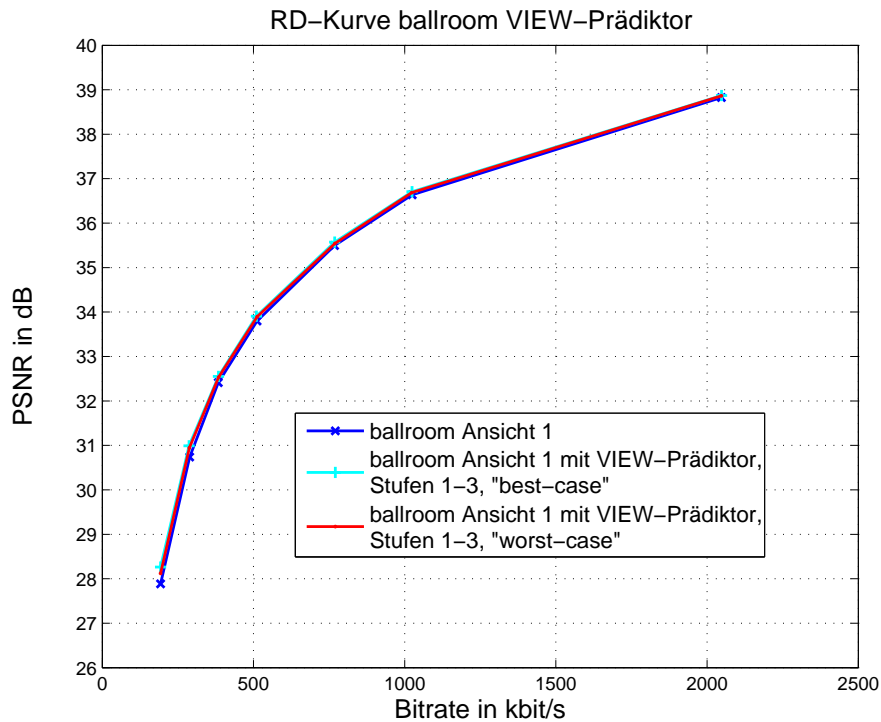


Abbildung 5.4: *ballroom* Sequenz, VIEW-Prädiktor Verfahren mit Stufen 1-3, "worst-case" und "best-case"

zusätzlich übertragen werden müssen.

Für einen Makroblock können natürlich auch weitere Bits eingesetzt werden, um zum Beispiel bei den verschiedenen Subblöcken auch zwischen örtlichem oder VIEW-Prädiktor zu unterscheiden. Ebenso kann ein weiteres Bit zur Berechnung der Kosten mit oder ohne VIEW-Prädiktor in der vorwärts- bzw. rückwärts gerichteten Prädiktion verwendet werden. Je mehr Subblöcke der Makroblock enthält, desto mehr Bits werden zur Signalisierung gebraucht. Die "best-case" RD-Kurve in Abbildung 5.5 wird dadurch weiterhin verbessert (durchschnittlicher PSNR Gewinn von ~ 0.32 dB gegenüber der Referenz). Es würden jedoch je nach Makroblock Modus mehrere zusätzliche Bits benötigt werden, die die Datenrate deutlich ansteigen lassen. Da dieser Ansatz für die "worst-case" Kurve keinen Erfolg verspricht, wurde er aus programmieretechnischen Gründen in dieser Arbeit nicht weiter untersucht.

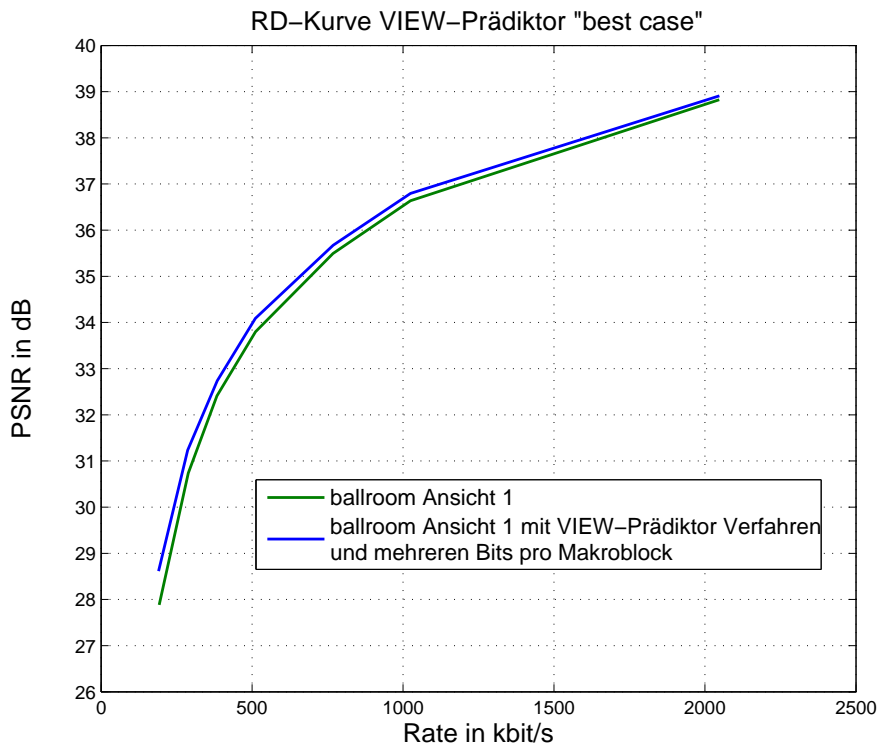


Abbildung 5.5: "VIEW-Prädiktor Verfahren "best-case" mit mehreren Bits Zusatzinformation pro Makroblock, Referenzkurve *ballroom* Sequenz

5.1.3 Implizite Signalisierung des VIEW-Prädiktors

Beschreibung

Wie bereits erwähnt, kann der VIEW-Prädiktor auch implizit signalisiert werden. Die Idee besteht darin, aufgrund der umliegenden und bereits vorhandenen Bewegungsvektoren zu entscheiden, ob der VIEW- oder der örtliche Prädiktor verwendet werden soll. Abbildung 5.6 zeigt die optimale Entscheidung zwischen örtlichem Prädiktor (blau) und VIEW-Prädiktor (rot). Es wird die unterste Stufe der Lifting-Struktur gewählt, da hier nur eine rückwärtsgerichtete Prädiktion stattfindet. Die "best-case" Entscheidung kann so leichter untersucht werden. Der VIEW-Prädiktor wird nur eingesetzt, wenn sowohl x- als auch y-Komponente bessere Prädiktoren in der disparitätskompensierten Nachbaransicht haben.

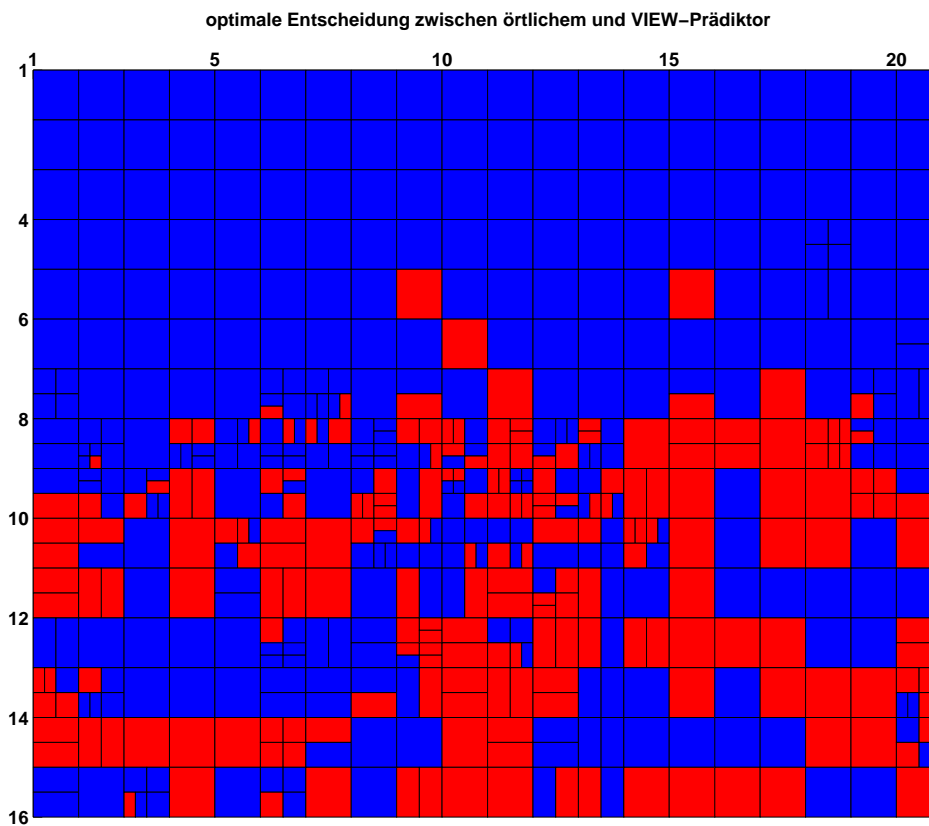


Abbildung 5.6: optimale Entscheidung zwischen örtlichem (blau) und VIEW-Prädiktor (rot), unterste Stufe der Lifting-Struktur, *ballroom* Sequenz

Ergebnisse

Es wurden verschiedene Ansätze untersucht, um die optimale Entscheidungsschwelle der "best-case" Kurve mit mehreren Bits nachzubilden. Sie werden im folgenden Text erläutert.

- Methode 1

Anfangs werden die umliegenden Bewegungsvektoren verwendet, die auch bei der Bestimmung des örtlichen Prädiktors zum Einsatz kommen (siehe Abbildung 2.8).

Zwischen ihnen wird laut Gleichung 5.1

$$\begin{aligned}f_1 &= |mv_a - mv_b| \\f_2 &= |mv_b - mv_c| \\f_3 &= |mv_b - mv_a|\end{aligned}\tag{5.1}$$

$$f_1 > D \wedge f_2 > D \wedge f_3 > D$$

jeweils der Betrag der Differenz gebildet. Diese Berechnung erfolgt sowohl für x- als auch für die y-Komponente der Bewegungsvektoren. Anschließend wird untersucht, ob die Werte f_1 , f_2 und f_3 allesamt größer als eine gewisse Schwelle D sind. Ist dies bei der x- und y-Komponenten der Fall, dann wird für den aktuellen Block (Subblock) der VIEW-Prädiktor verwendet. So wird untersucht, ob die Bewegungsvektoren zumindest von ihrer Länge sich stark unterscheiden oder nicht. Die Schwelle wurde durch weitere Versuche bestimmt. Nach Gleichung 5.1 ist es allerdings so, dass sie keine großen Auswirkungen hat, da allgemein nur sehr wenige Blöcke richtig erkannt werden. Der Entscheidungsalgorithmus wird so abgeändert, dass nicht f_1 sowohl f_2 als auch f_3 größer als die Grenze sein müssen, sondern nur jeweils zwei von ihnen. Die Schwelle wird zunächst so gewählt, dass durch den Entscheidungsalgorithmus der VIEW-Prädiktor für einen Block nicht falsch-positiv ist. Dadurch ist es aber auch nicht möglich, viele Blöcke zu erkennen, bei denen der VIEW-Prädiktor ausgewählt werden soll. Es gibt also noch zu viele falsch-negative Blöcke. Wird dagegen die Grenze kleiner gemacht, so werden zwar einige Blöcke VIEW-Prädiktor-positiv erkannt, dennoch wird auch der Anteil der falsch-positiven Entscheidungen größer. Bei den Simulationsergebnissen ergibt sich aber kein Gewinn zur Referenzkurve. Es ist vielmehr so, dass die falsch-positiv erkannten Blöcke stärker ins Gewicht und somit zu einem Verlust im Vergleich zur Referenz führen.

- Methode 2

Im folgenden wird versucht, den Algorithmus zur impliziten Erkennung des VIEW-Prädiktors weiter zu verbessern. Es wurden Blöcke von Abbildung 5.6, bei denen der

VIEW-Prädiktor gewählt wurde, näher untersucht. Auffallend ist, dass bei einigen von ihnen zwei der umliegenden Bewegungsvektoren keine Bewegung (gleich null) und beim anderen eine starke Bewegung vorhanden ist.

$$\begin{aligned} & (mv_a == 0 \wedge mv_b == 0 \wedge mv_c > D) \\ \vee & (mv_b == 0 \wedge mv_c == 0 \wedge mv_a > D) \\ \vee & (mv_a == 0 \wedge mv_c == 0 \wedge mv_b > D) \end{aligned} \tag{5.2}$$

Diese Gleichung beschreibt den, nach oben erhaltenen Erkenntnissen, neuen Entscheidungsmodus. Sind zwei der umliegenden Bewegungsvektoren null und der andere ist größer als eine bestimmte Grenze, so soll der VIEW-Prädiktor gesetzt werden. Dadurch werden zwar viele neue Blöcke als VIEW-Prädiktor-positiv erkannt, doch das Ergebnis kann durch falsch-positive nicht verbessert werden.

- Methode 3

Des Weiteren gibt es umliegende Bewegungsvektoren mit verschiedenen Vorzeichen. Zunächst wird der VIEW-Prädiktor gesetzt, sobald alle drei örtlichen Bewegungsvektoren nicht das gleiche Vorzeichen haben. Da aber zu viele Blöcke als falsch-positiv gekennzeichnet werden, wird die Entscheidung so angepasst, dass die Differenz der derjenigen mit gleichem Vorzeichen größer als eine bestimmte Grenze sein muss. Leider lässt sich damit wieder kein Erfolg erzielen.

- Methode 4

Eine andere Ansatzweise verwendet nachfolgende Idee. Zu den drei umliegenden Prädiktoren (links, oben, oben rechts) wird der links oben liegende sowie der VIEW-Prädiktor mit zur Entscheidung einbezogen. Sie werden der Größe nach geordnet und der Median-Wert berechnet. Entspricht letzterer dem VIEW-Prädiktor, so wird er eingesetzt, andererseits wird der örtliche Prädiktor verwendet. Es wird davon ausgegangen, dass durch die zusätzliche Information bei der Signalisierungsentscheidung ein besser geeigneter Prädiktor für den Bewegungsvektor gefunden werden kann. Die Simulationsergebnisse aber entsprechen den vorherigen. Es kann mit dieser Nachbildung der

optimalen Entscheidung ebenfalls kein Gewinn gegenüber der Referenz erzielt werden.

- Methode 5

Zuletzt wird untersucht, wie sich eine Kombination der verschiedenen Methoden auswirkt. Dennoch blieb auch dieses Vorgehen ohne erfolgversprechendes Resultat.

- Fazit

Mit den Entscheidungsregeln werden einige VIEW-Prädiktor Blöcke erkannt. Dennoch liegt das Problem darin, dass es auch viele Blöcke gibt, bei denen der örtliche Prädiktor trotz verschiedener Vorzeichen oder großen Richtungsunterschieden der umliegenden Bewegungsvektoren, sehr genau ist. Dadurch wird der VIEW-Prädiktor sehr oft falsch-positiv eingesetzt und somit das Ergebnis wieder verschlechtert. Dies lässt sich auch durch den Grenzwert in der Entscheidung nicht ändern. Zugleich ist aufgefallen, dass bei plötzlicher auftretender Bewegung der VIEW-Prädiktor gewählt wird, da die Bewegung an gleicher Stelle in der disparitätskompensierten Nachbaransicht ähnlich ist. Die umliegenden Bewegungsvektoren sind, wie zum Beispiel im Hintergrund, allesamt null oder sehr klein. Folglich wird mit den umliegenden Bewegungsvektoren der Einsatz des VIEW-Prädiktors unbefriedigend gekennzeichnet.

Abschließend kann erwähnt werden, dass durch die "best-case" Kurve ein großer Gewinn durch den VIEW-Prädiktor möglich wäre. Leider lässt sich im Hinblick auf die umliegenden Bewegungsvektoren eines Makroblocks keine zufriedenstellende Aussage über den richtigen Einsatz des VIEW-Prädiktors treffen. Durch die implizite Signalisierung ist also kein Gewinn gegenüber der Referenzkurve möglich. Bei den anderen untersuchten Sequenzen lassen sich die negativen Ergebnisse ebenfalls bestätigen.

Kapitel 6

Kombination von VIEWSKIP Modus und VIEW-Prädiktor Verfahren

6.1 Beschreibung

Zuletzt wird der VIEWSKIP Modus mit dem VIEW-Prädiktor Verfahren kombiniert. Dies geschieht aus folgenden Gründen:

Im Detail der Prädiktion fällt auf, dass beim VIEWSKIP Modus die Prädiktionsrichtung aus der Nachbaransicht übernommen wird. Im VIEW-Prädiktor Verfahren dagegen wird berechnet, ob eine vorwärts-, rückwärts- oder bidirektional gerichtete Prädiktion am besten geeignet ist. Dies bringt bei der Kombination beider Verfahren einen zusätzlichen Vorteil gegenüber der ausschließlichen Verwendung des VIEWSKIP Modus.

6.2 Ergebnisse

Simuliert wurde die Kombination von VIEWSKIP Modus und VIEW-Prädiktor Verfahren für die Stufen eins bis drei der Lifting-Struktur. Abbildung 6.1 zeigt die blaue Referenzkurve der *ballroom* Sequenz sowie die "worst-case" RD-Kurven des VIEW-Prädiktor Verfahrens (grün) und der Kombination aus oben genannten Verfahren (braun). Durch den zusätzlichen

VIEWSKIP Modus in der Kombination wird gerade bei niedrigeren Bitraten ein Gewinn gegenüber der grünen Kurve, und vor allem gegenüber der Referenzkurve erzielt.

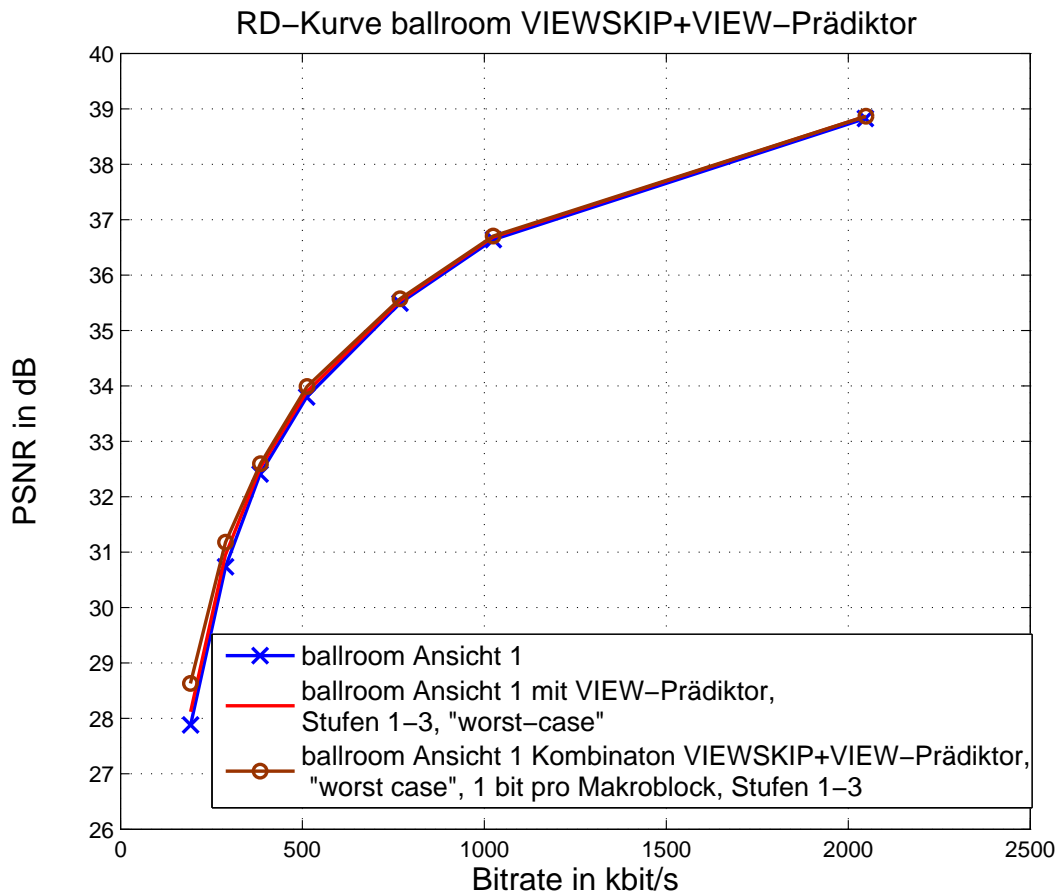


Abbildung 6.1: Kombination aus VIEWSKIP und VIEW-Prädiktor Verfahren, Stufen 1-3 der Lifting-Struktur, *ballroom* Sequenz

Vergleicht man zuletzt die Kombination der Verfahren (Abbildung 6.2) mit der VIEWSKIP RD-Kurve, so liegen beide fast aufeinander. Bei höheren Raten kann dennoch ein kleiner Gewinn erzielt werden, was zu einem zusätzlichen minimalen Erfolg gegenüber der VIEWSKIP Kurve führt. Der Grund hierfür wurde bereits oben beschrieben. Somit ist die Kombination des VIEWSKIP Modus und VIEW-Prädiktor Verfahren durchaus sinnvoll.

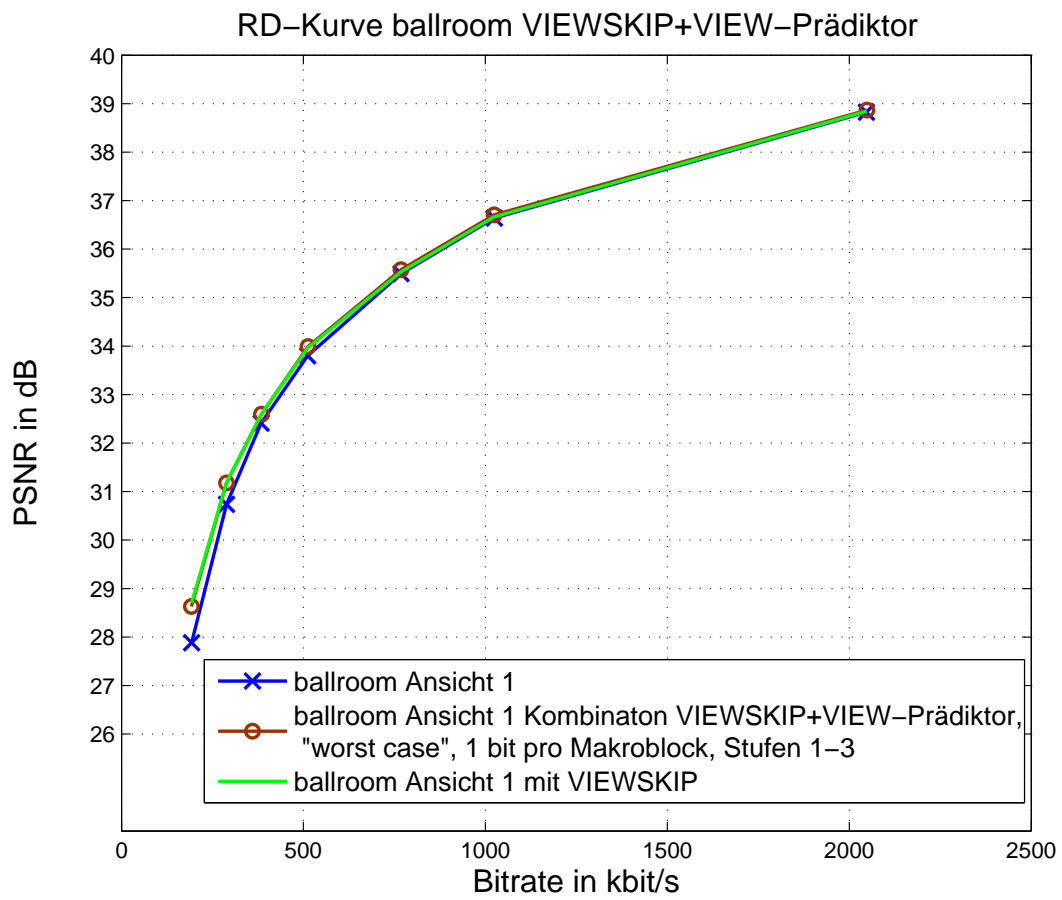


Abbildung 6.2: Kombination aus VIEWSKIP und VIEW-Prädiktor Verfahren im Vergleich mit VIEWSKIP, *ballroom* Sequenz

Kapitel 7

Zusammenfassung

In dieser Arbeit wurden verschiedene Ansätze und Verfahren zur Inter-View-Bewegungsprädiktion in einem waveletbasierten Videocoder vorgestellt und untersucht.

Zu Beginn wurden die Grundlagen des verwendeten Motion Compensated Wavelet Coders erläutert.

Die anschließende Voruntersuchung beschäftigte sich mit der Kompensation der Disparitäten. Die globale Phasenkorrelationsmethode wurde eingeführt und künftig verwendet. Weiterhin wurde der Ansatz zur linearen Interpolation bei nicht ganzzahliger Makroblockverschiebung vorgeschlagen und die Vorteile der unteren Stufen der Lifting-Struktur bei der Inter-View-Bewegungsprädiktion aufgezeigt.

Mit Hilfe der zuvor erhaltenen Ergebnisse wurde der VIEWSKIP Modus eingeführt. Durch ihn konnte bereits ein Gewinn bei niedrigeren Bitraten gegenüber der Referenzkurve erzielt werden. Der neue Modus wurde in Verbindung mit den anderen Modi in Bezug auf die Reihenfolge unterschiedlich eingesetzt. Ebenso wurde der Kostenfaktor λ verändert. Dadurch konnte festgestellt werden, dass es sich durchaus als lohnenswert erweisen könnte, wenn man den MCW-Coder mit VIEWSKIP und doppeltem Kostenfaktor einsetzen würde.

Ein anderer Ansatz, die disparitätskompensierte Bewegungsinformation aus der Nachbaransicht zu verwenden, bietet das eingeführte VIEW-Prädiktor Verfahren. Untersucht wurde zunächst eine expliziten Signalisierung des VIEW-Prädiktors mit einem zusätzlichen Bit pro Makroblock. Mit der Verwendung der unteren Stufen der Lifting-Struktur konnte ein Ge-

winn zur Referenzkurve erzielt werden. Erweitert wurde die Signalisierung für mehrere Bits pro Makroblock. Dieses Verfahren war jedoch nicht erfolgreich, da zu viele Bits als Zusatzinformation für einen Makroblock benötigt wurden.

Um eine weitere Verbesserung erzielen zu können, wurde die implizite Signalisierung eingeführt. Die zusätzliche Seiteninformation für den VIEW-Prädiktor sollte durch eine Entscheidungsregel über die umliegenden Bewegungsvektoren ersetzt werden. Hier wurden einige Ansätze untersucht und simuliert. Ein zufriedenstellendes Ergebnis stellte sich dennoch nicht ein, da der VIEW-Prädiktor oftmals falsch-negativ oder falsch-positiv eingesetzt wurde. Die Verwendung der umliegenden Bewegungsvektoren als Hilfe für eine VIEW-Prädiktorentscheidung erwies sich als nicht erfolgreich.

Zuletzt wurde noch die Kombination aus VIEWSKIP Modus und VIEW-Prädiktor Verfahren betrachtet. Durch den geringen Unterschied in der Prädiktion der beiden Verfahren konnte sich durch ihre Kombination ein zusätzlicher, wenn auch kleiner, Gewinn einstellen.

Abschließend kann gesagt werden, dass sich der Ansatz zur Einsparung an Datenrate durch die Inter-View-Bewegungsschätzung lohnt. Der VIEWSKIP Modus sowie das VIEW-Prädiktor Verfahren können vielseitig eingesetzt werden. Vor allem bei niedrigeren Bitraten lässt sich durch beide ein großer Gewinn gegenüber der Referenzkurve erreichen.

Um erste Ergebnisse über die Inter-View-Bewegungsschätzung zu bekommen, wurden nur die Ansichten null und eins der Beispielsequenzen untersucht. Eine Multi-View Videosequenz besitzt natürlich mehrere Ansichten. So könnte zum Beispiel jede ungerade Ansicht normal codiert und übertragen werden. Bei jeder geraden Ansicht könnte dann die Inter-View-Bewegungsprädiktion eingesetzt werden.

Abbildungsverzeichnis

2.1	Multiview Videosequenz, nach [1]	3
2.2	Lifting-Struktur der MCTF, nach [5]	4
2.3	mehrstufige Lifting Struktur einer Ansicht	6
2.4	Örtliche Zerlegung mit Hilfe der Wavelet-Transformation	6
2.5	blockbasierte Bewegungsschätzung, nach [5]	7
2.6	UVLC Pyramidenform, nach [5]	9
2.7	UVLC Tabelle, Zusammenhang von Codewort, Nummer und Länge, nach [5]	10
2.8	Median Prädiktion, nach [5]	11
2.9	Richtungsabhängige Prädiktion, nach [5]	12
3.1	Peak der Phasenkorrelation bei <i>ballroom</i> Sequenz zwischen sechstem Bild von Ansicht 0 und Ansicht 1	17
3.2	Veranschaulichung des Vorgangs zur Berechnung des MSE	18
3.3	notwendige Interpolation der Bewegungsvektoren bei nicht ganzzahliger block- verschiebung	19
3.4	bilineare Interpolation, nach [4]	20
3.5	farblich dargestellte Kosten des im Text genannten Beispiels der <i>ballroom</i> Sequenz, x- und y-Achse in Pixel	22
3.6	<i>ballroom</i> Sequenz, blockbasierte Entscheidung zwischen Fair{ MVF_1 } und Diff{ MVF }, x- und y-Achse in Pixel	23

4.1	<i>ballroom</i> Sequenz, Vergleich zwischen Referenzkurve und VIEWSKIP-Kurve von Ansicht 1	27
4.2	<i>ballroom</i> Sequenz, Vergleich zwischen Referenzkurve und VIEWSKIP-Kurve von Ansicht 1 bei doppeltem Kostenfaktor λ	30
4.3	<i>ballroom</i> Sequenz, Vergleich zwischen Referenzkurve (Standard lambda), Referenzkurve (doppeltem lambda), VIEWSKIP-Kurve (standard lambda) und VIEWSKIP-Kurve (doppeltem lambda), Ansicht 1	31
5.1	Situation eines ungeeigneten örtlichen Prädiktors (türkis) für den aktuell zu übertragenden Makroblock (grün)	33
5.2	Veranschaulichung der Berechnung der x-Komponente des VIEW-Prädiktors für einen INTER16x8 Makroblock	35
5.3	<i>ballroom</i> Sequenz, VIEW-Prädiktor Verfahren mit Stufen 0-3, "worst-case" und "best-case"	36
5.4	<i>ballroom</i> Sequenz, VIEW-Prädiktor Verfahren mit Stufen 1-3, "worst-case" und "best-case"	38
5.5	"VIEW-Prädiktor Verfahren "best-case" mit mehreren Bits Zusatzinformation pro Makroblock, Referenzkurve <i>ballroom</i> Sequenz	39
5.6	optimale Entscheidung zwischen örtlichem (blau) und VIEW-Prädiktor (rot), unterste Stufe der Lifting-Struktur, <i>ballroom</i> Sequenz	40
6.1	Kombination aus VIEWSKIP und VIEW-Prädiktor Verfahren, Stufen 1-3 der Lifting-Struktur, <i>ballroom</i> Sequenz	46
6.2	Kombination aus VIEWSKIP und VIEW-Prädiktor Verfahren im Vergleich mit VIEWSKIP, <i>ballroom</i> Sequenz	47

Tabellenverzeichnis

4.1	Prozentuale Anteile der Modi für die einzelnen Stufen, Referenz	28
4.2	Prozentuale Anteile der Modi für die einzelnen Stufen, zusätzlich verwendeter VIEWSKIP Modus	28

Literaturverzeichnis

- [1] Jens-Uwe Garbas. Wavelet-based scalable multi-view video coding - an overview. Talk, University of Erlangen.
- [2] Yi Liang. Phase-correlation motion estimation. Technical report, Stanford University, yiliang@stanford.edu.
- [3] Wikipedia. Bildregistrierung, fourier-methoden, phasenkorrelation. Website.
- [4] Wikipedia. Bilineare filterung. Website.
- [5] Ruiqin Xiong, Xiangyang Ji, Dongdong Zhang, and X Jizheng. *MSRA 3D Wavelet Video Coding Specification*. Microsoft Research Asia, t-jzhxu@microsoft.com, 2005.