Decoderseitige Bewegungsschätzung in der Videocodierung

Der Fakultät für Elektrotechnik und Informatik

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte

Dissertation

von

Dipl.-Ing. Sven Klomp

geboren am 08.11.1979 in Nordhorn.

2011

Hauptreferent:	Prof. Dr.-Ing. J. Ostermann
Korreferent:	Prof. Dr.-Ing. A. Kaup
Vorsitzender:	Prof. Dr.-Ing. H.G. Musmann

Tag der Promotion: 15. Dezember 2011

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Informationsverarbeitung der Leibniz Universität Hannover. An dieser Stelle möchte ich all denen danken, die es mir ermöglicht haben, diese Dissertation zu verfassen.

Mein besonderer Dank gilt Professor Dr.-Ing. Jörn Ostermann für die Betreuung der Arbeit und die Übernahme des Hauptreferats. Er hat durch seine Anregungen in zahlreichen Fachgesprächen zum Gelingen dieser Arbeit beigetragen. Auch für die mir vom Institut gebotene Möglichkeit, aktiv an Standardisierungstätigkeiten mitzuwirken, bin ich sehr dankbar.

Professor Dr.-Ing. André Kaup danke ich für die Übernahme des Korreferats. Des Weiteren möchte ich Professor Dr.-Ing. Hans-Georg Musmann für die Übernahme des Vorsitzes der Prüfungskommission danken.

Ebenfalls danken möchte ich meinen ehemaligen Kollegen für die sehr gute Arbeitsatmosphäre und stets entgegengebrachte Hilfsbereitschaft. Dazu zählen unter anderem Christian Becker, der sehr zum Zusammenhalt innerhalb des Instituts beigetragen hat, sowie Marco Munderloh, Yuri Vatis, Holger Meuel und Julia Schmidt für die fachliche Unterstützung meiner Arbeit. Auch möchte ich meinem ehemaligen Kollegen und sehr guten Freund Torsten Büschenfeld für seine Ausdauer bei der Korrektur diverser Veröffentlichungen – einschließlich dieser Arbeit – herzlichst danken. Martin Pahl und Matthias Schuh gilt mein Dank für die hervorragende Computerinfrastruktur und die interessanten, fachfremden Diskussionen. Nicht zuletzt danke ich Doris Jaspers-Göring, Ursula Kemner und Silvia Scholl für die administrative Unterstützung.

Paul Cochrane und seinen Kollegen vom Regionalen Rechenzentrum für Niedersachsen danke ich für die Bereitstellung des Clustersystems zur Verarbeitung der großen Datenmengen und die stets freundliche und schnelle Unterstützung bei Problemen.

Ich danke außerdem meiner Freundin Beatrix Hohmann für die unermüdliche Unterstützung, ihr Verständnis und ihre große Geduld insbesondere in der Endphase der Promotionszeit.

Mein spezieller Dank gilt meinen Eltern Mina und Herbert Klomp, die mich während meines gesamten Werdegangs auf vielfältige und unersetzliche Weise unterstützt haben. Ihnen ist diese Arbeit gewidmet.

Inhaltsverzeichnis

Abkürzungen und Formelzeichen

Abkürzungen:

3DRS	3D Recursive Search
720p	HDTV mit einer Auflösung von 1280×720 Bildpunkten progressiv
1080p	HDTV mit einer Auflösung von 1920×1080 Bildpunkten progressiv
AMVP	Advanced Motion Vector Prediction
AOBMC	Adaptive Overlapped Block Motion Compensation
AVC	Advanced Video Coding
BD	Bjøntegaard Delta
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CBP	Coded Block Pattern
CCD	Charge Coupled Device
CIF	Common Intermediate Format
CU	Coding Unit
DCT	Diskrete Cosinustransformation
DMVD	Decoder-side Motion Vector Derivation
DPCM	Differentielle Pulsecodemodulation
DSL	Digital Subscriber Line
DSME	Decoder-side Motion Estimation
DVB-T	Digital Video Broadcasting - Terrestrial
DVC	Distributed Video Coding
DVD	Digital Versatile Disc
FRUC	Frame Rate Up Conversion
GPU	Gaphics Processing Unit
HDTV	High Definition Television
HEVC	High Efficiency Video Coding
HM	HEVC Testmodell
IEC	International Electrotechnical Commission
ISO	International Standardization Organization
ITU-R	International Telecommunication Union - Recommendation
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
JCT-VC	Joint Collaborative Team on Video Coding
JM	Joint Model

LCD	Liquid Crystal Display
LTE	Long Term Evolution
MAD	Mean of the Absolute Differences
MPEG	Moving Picture Experts Group
MSD	Mean of the Squared Differences
MVC	Motion Vector Competition
MVD	Motion Vector Difference
MVP	Motion Vector Prediction
OBME	Overlapped Block Motion Estimation
PCM	Pulsecodemodulation
PSNR	Peak Signal to Noise Ratio
PU	Prediction Unit
QCIF	Quarter Common Intermediate Format
RD	Rate Distortion
SAD	Sum of the Absolute Differences
STAR	Spatio-Temporal Autoregressive
TB	Tree Block
UHDTV	Ultra High Definition Television
VCEG	Video Coding Experts Group
WVMF	Weighted Vector Median Filter

Formelzeichen:

a	Beschleunigung
B	Blockgröße
b	Bitstrom
c	Korrelationskoeffizient
$\tilde{c}$	Gemessener Korrelationskoeffizient
$\vec{d}$	Bewegungsvektor
$\vec{d'}$	Zu übertragender Bewegungsvektor mit begrenzter Genauigkeit
Δ	Genauigkeit der Bewegungsvektoren
$E[.]$	Erwartungswert
e	Prädiktionsfehlersignal
e'	Residuum (Quantisiertes Prädiktionsfehlersignal)
f_t	Bildwiederholfrequenz
G	Maß für die Strukturierung eines Bildes basierend auf den Gradienten
H_i	i-te Hierarchiestufe
$H(.)$	Entropie eines diskreten Signals
$h(.)$	Differentielle Entropie eines kontinuierlichen Signals
k	Konstante zur Substitution
M	Anzahl von Bildpunkten
m	Steigung einer linearen Funktion
μ	Varianz des Rauschsignals

N	Gestamtanzahl von Bildpunkten in einem Bild
N_x	Breite eines Bildes
N_y	Höhe eines Bildes
n_t	Diskrete zeitliche Koordinate
n_x	Diskrete horizontale Koordinate
n_y	Diskrete vertikale Koordinate
$\mathcal{N}(\bar{a},\sigma^2)$	Normalverteilte Zufallsvariable mit Mittelwert $\bar{a}$ und Varianz σ^2
O	Im Videosignal dargestelltes physikalisches Objekt
P_B	Wahrscheinlichkeit für Änderungen im Versatz bei eine Blockgröße B
$\vec{p}$	Position eines Blockes
$\hat{p}$	Prädizierte Position eines Blockes
$\frac{\partial(.)}{\partial B}$	Partielle Ableitung nach der Blockgröße
Q	Quantisierungsstufenbreite
R	Gesamtdatenrate
$\check{R}$	Berechnete Gesamtdatenrate bei feiner Quantisierung
$\hat{R}$	Berechnete Gesamtdatenrate bei grober Quantisierung
S	Zustand bei der prädiktiven Codierung
s	Videosignal
$\check{s}$	Mit Hilfe von DSME interpoliertes Signal
$\hat{s}$	Prädiktionssignal
s'	Rekonstruiertes Videosignal
σ^2	Varianz
T_t	Zeitliche Abtastperiode ($T_t = \frac{1}{f}$)
T_x	Örtliche Abtastperiode in horizontaler Richtung
T_y	Örtliche Abtastperiode in vertikaler Richtung
t	Kontinuierliche zeitliche Koordinate
$\vec{V}$	Bewegung zwischen zwei Referenzbildern
w	Gewichtung
x	Kontinuierliche horizontale Koordinate
y	Kontinuierliche vertikale Koordinate

Zusätzliche Indizes:

-1	Kennzeichnet zeitlich vorrangegangenes Bild
$+1$	Kennzeichnet zeitlich folgendes Bild
A	Kennzeichnet einen autoregressiven Zufallsprozess
i	Allgemeiner Zählindex
j	Allgemeiner Zählindex
n	Kennzeichnet das Rauschen
R	Kennzeichnet das Residuum
V	Kennzeichnet die Bewegungsvektoren
WVM	Kennzeichnet den gesuchten Vektor des WVMF

Kurzfassung

In aktuellen Videocodierstandards werden zeitliche Abhängigkeiten innerhalb einer Sequenz ausgenutzt, um die benötigte Datenrate zu verringern. Dabei wird ein zu codierendes Bild mit Hilfe einer bewegungskompensierten Prädiktion geschätzt und lediglich der entstandene Prädiktionsfehler übertragen. Jedoch werden die benötigten Bewegungsvektoren nur am Encoder geschätzt, sodass diese ebenfalls zum Decoder übertragen werden müssen.

Um die Codiereffizienz durch Einsparung der Bewegungsvektoren zu erhöhen, wird eine decoderseitige Bewegungsschätzung untersucht. Es wird ein Modell zur Bestimmung der notwendigen Datenrate hergeleitet und anhand experimenteller Ergebnisse verifiziert. Mit Hilfe dieses Modells wird gezeigt, dass die Datenrate deutlich gesenkt werden kann, wenn decoderseitige Bewegungsschätzung genutzt wird.

Aufbauend auf den Ergebnissen der Analyse wird eine neue Coderarchitektur vorgestellt, welche die Bewegung auch am Decoder schätzt und diese zur Prädiktion nutzt. Es wird gezeigt, dass die Einschränkung durch die notwendige Annahme unbeschleunigter Bewegung zwischen zwei Bildern mit Hilfe dieser Architektur umgangen werden kann. Ein detaillierter Vergleich der experimentellen Ergebnisse mit dem Referenzverfahren HEVC, welches von der ISO/IEC und ITU-T bei der Entwicklung eines neuen Videocodierstandards genutzt wird, zeigt, dass mit Hilfe der vorgestellten Technik die benötigte Datenrate zur Codierung der untersuchten Sequenzen im Mittel um 3,3 % reduziert wird.

Stichworte: Videocodierung, Redundanzreduktion, decoderseitige Bewegungsschätzung, Bewegungskompensation, H.264 / MPEG-4 Part 10, HEVC

Abstract

In current video coding standards, the encoder takes advantage of temporal dependencies within the video sequence by performing motion compensated prediction. By only transmitting the prediction error, the data rate can be significantly reduced. However, the motion estimation is only performed at the encoder. Therefore, the motion vectors have to be transmitted as well.

To improve the coding efficiency by reducing the amount of motion vectors, a decoder-side motion estimation is evaluated. A model for estimating the rate is proposed and verified with experimental results. Using this model, it is shown that the data rate can be significantly reduced by using decoder-side motion estimation.

Based on this analysis, a new coder architecture is introduced, which estimates the motion at the decoder to predict the current frame. It is shown that the limitation caused by the essential assumption of constant motion between two frames can be avoided with this architecture. A detailed comparison of the experimental results with the HEVC reference, which is used by the ISO/IEC and ITU-T for developing a new video coding standard, shows that decoder-side motion estimation reduces the required rate for coding several sequences by 3.3 % on average.

Keywords: video coding, redundancy reduction, decoder-side motion estimation, motion compensation, H.264 / MPEG-4 Part 10, HEVC

1 Einleitung

In heutigen Medien sind Videosequenzen zu einem wichtigen Bestandteil beim Wissensaustausch geworden. Neben dem klassischen Rundfunk von Fernsehsendern über Funk und Kabel wird das Internet zu einem immer wichtigeren Medium in der Verbreitung und im Austausch von Videodaten. Es ist zu erwarten, dass im Jahr 2014 über 50 % des von Endkunden erzeugten Internetverkehrs durch Videodaten verursacht werden wird [14]. Eine wirtschaftliche Übertragung oder eine Speicherung dieser Videodaten auf digitalen Medien wäre ohne Komprimierung mit effizienten Codieralgorithmen nicht möglich. So benötigt zum Beispiel ein unkomprimiertes digitales Fernsehsignal, das dem Studioformat ITU-R BT.601 [29] entspricht, eine Datenrate von 166 $^{\text{Mbit}}/_{\text{s}}$. Auf einer DVD (*Digital Versatile Disc*) mit 8,5 GB Speicherplatz könnten somit nur knapp sieben Minuten unkomprimierte Videodaten gespeichert werden.

Daher wird sowohl bei der Speicherung auf DVD als auch zur Fernsehübertragung über DVB-T (*Digital Video Broadcasting - Terrestrial*) meist der Standard MPEG-2 [26] zur Komprimierung der Videodaten verwendet, welcher 1994 durch die *International Standardization Organization* (ISO) standardisiert wurde. Um das Videosignal visuell verlustfrei zu codieren, sodass für den menschlichen Betrachter keine Unterschiede zum unkomprimierten Signal erkennbar sind, benötigt dieser Standard abhängig vom Videoinhalt 4-6 $^{\text{Mbit}}/_{\text{s}}$ Datenrate. Wird der 2003 standardisierte Nachfolger ISO/IEC 14496-10 / ITU-T H.264 (AVC) [25] verwendet, kann die Datenrate nochmals um die Hälfte reduziert werden [69]. Bei hochaufgelösten Fernsehsignalen (*High Definition Television*, HDTV) [30] liegt die Rate der unkomprimierten Daten zwischen 553 und 1244 $^{\text{Mbit}}/_{\text{s}}$. Wird solch ein Signal mit Hilfe von AVC codiert, kann die Datenrate auf etwa 10-20 $^{\text{Mbit}}/_{\text{s}}$ verringert werden.

Derzeit stellen jedoch weder Internetzugänge für Endkunden, wie zum Beispiel DSL (*Digital Subscriber Line*) oder die mobile Alternative LTE (*Long Term Evolution*), noch DVB-T die benötigte Datenrate für die Echtzeitübertragung von HDTV zur Verfügung. Obwohl zu erwarten ist, dass Internetzugänge mit höheren Datenraten in der Zukunft verfügbar sein werden, wird eine effiziente Nutzung der Übertragungskanäle und Speichermedien auch weiterhin eine wichtige Rolle spielen. In vielen Bereichen – wie etwa bei *Ultra High Definition Television* (UHDTV) [54], das die 16-fache Auflösung von HDTV besitzt – werden immer höhere Auflösungen verlangt. Es ist zu erwarten, dass die benötigte Datenrate für Videosignale, die mit heutigen Standards codiert sind, schneller steigt als die Kapazität der Netzwerke [27].

Daher haben ISO/IEC und ITU-T gemeinsam das *Joint Collaborative Team on*

Video Coding (JCT-VC) ins Leben gerufen, welches sich mit der Entwicklung eines neuen Videocodierstandards unter dem Arbeitstitel *High Efficiency Video Coding* (HEVC) [70] beschäftigt.

Stand der Technik

Die meisten standardisierten Videocodierverfahren [26, 25, 31, 23, 32, 24] greifen das Prinzip der Hybridcodierung auf, welches zeitliche und örtliche Korrelationen innerhalb der Videosequenz zur Datenreduktion nutzt. Die zeitlichen Abhängigkeiten werden mit Hilfe einer bewegungskompensierenden Prädiktion ausgenutzt. Dafür wird das zu codierende Bild zunächst in Blöcke unterteilt. Zu jedem Block wird ein ähnlicher Block in ausgewählten, bereits codierten Bildern, den sogenannten Referenzbildern, gesucht. Um den Aufwand bei der Bewegungsschätzung in Grenzen zu halten, werden nicht alle bereits codierten Bilder verwendet, sondern nur eine begrenzte Anzahl, die in einer Referenzbildliste gespeichert sind. Der Versatz des ermittelten Blockes zum zu codierenden Block wird als Bewegungsvektor oder auch *Displacement Vector* bezeichnet. Mit Hilfe dieser Vektoren und der Referenzbilder kann die Prädiktion des aktuellen Bildes erfolgen. Werden zur Codierung des aktuellen Bildes lediglich zeitlich vorangegangene Referenzbilder zur Prädiktion genutzt, spricht man von einem P-Bild. Mit der Standardisierung von MPEG-2 wurden zum ersten Mal sogenannte bidirektional prädizierte Bilder (B-Bilder) eingeführt, die auch Referenzbilder aus der Zukunft nutzen. In diesem Fall ist die Codierreihenfolge der einzelnen Bilder eine andere als bei der Darstellung auf dem Bildschirm.

Die blockbasierte Bewegungsschätzung erlaubt jedoch lediglich die Kompensation von translatorischer Bewegung. Rotation und Verzerrungen auf Grund der perspektivischen Abbildung können nicht kompensiert werden. Die durch die Schätzung entstandene Differenz zum Originalbild wird vom Encoder komprimiert und übertragen. Dazu wird dieser sogenannte Prädiktionsfehler in den Frequenzbereich transformiert, um die verbliebenen örtlichen Korrelationen auszunutzen und eine an die Wahrnehmung angepasste Quantisierung zu ermöglichen. Hohe Frequenzen können dabei gröber quantisiert werden, ohne dass Unterschiede für einen menschlichen Betrachter erkennbar sind [60]. Die noch vorhandenen Redundanzen innerhalb des quantisierten Prädiktionsfehlers – auch Residuum genannt – werden mit Hilfe der Entropiecodierung minimiert und die verbleibenden Daten anschließend zum Decoder übertragen.

Da dem Decoder das Originalbild zur Bewegungsschätzung nicht zur Verfügung steht, müssen zusätzlich die Bewegungsvektoren übertragen werden, um das Prädiktionsbild des aktuellen Bildes erstellen zu können. Die Anzahl der zu übertragenen Bewegungsvektoren hängt dabei direkt von der gewählten Größe der Blöcke ab. Diese ist in den meisten Videocodierstandards variabel, um sich an die Charakteristiken der Videosequenz anpassen zu können. So erlaubt zum Beispiel AVC Blöckgrößen

zwischen 4×4 und 16×16 Bildpunkten.

Die Bewegungsvektoren haben im Allgemeinen eine hohe Varianz und besitzen somit eine hohe Entropie. Aus diesem Grund ist auch eine hohe Datenrate zur verlustfreien Codierung der Vektoren nötig. Es kann jedoch davon ausgegangen werden, dass benachbarte Blöcke ähnliche Bewegungen vollziehen. Daher werden in AVC bis zu drei bereits übertragene Bewegungsvektoren von benachbarten Blöcken genutzt, um den aktuellen Vektor zu prädizieren (*Motion Vector Prediction*, MVP) [59]. Es muss nun lediglich die Differenz übertragen werden, welche eine kleinere Entropie als der ursprüngliche Bewegungsvektor aufweist und folglich effizienter zu codieren ist. Aktuelle Forschungsergebnisse belegen, dass die Datenrate der codierten Bewegungsvektoren weiter verringert werden kann, indem mehr Informationen als nur örtlich benachbarte Bewegungsvektoren zur Prädiktion genutzt werden. So berücksichtigt *Motion Vector Competition* (MVC) [44] zusätzlich auch Bewegungen von zeitlich benachbarten Blöcken, um eine genauere Prädiktion des aktuellen Bewegungsvektors zu erstellen. Im Gegensatz dazu werden bei *Decoder-side Motion Vector Derivation* (DMVD) [35] bereits decodierte Bildbereiche zur Bewegungsvektorprädiktion hinzugezogen.

Nachdem der Decoder die Bewegungsvektoren erhalten hat, wird mit Hilfe der Referenzbilder die bewegungskompensierte Prädiktion des aktuellen Bildes durchgeführt. Anschließend werden die übertragene Residuumsinformation hinzu addiert, um das decodierte Bild der Videosequenz zu erhalten. Ein Codiergewinn wird erzielt, da die zur Übertragung der Bewegungsvektoren und des Residuums benötigte Datenrate geringer ist als die Datenrate bei unkomprimierter Pulsecodemodulation (PCM).

Grenzen aktueller Verfahren

Es ist offensichtlich, dass der Kompressionsgewinn in aktuellen Standards direkt von der Güte der Prädiktion des zu codierenden Bildes abhängt. Existieren große Differenzen zum Originalbild, steigt die Datenrate der Residuumcodierung an.

Große Einschränkungen bei der Prädiktion werden insbesondere durch die blockbasierte Bewegungskompensation verursacht, welche lediglich Translationen in der Bildebene modellieren kann. So bereiten Rotationen und Bewegungen entlang der optischen Achse der Kamera (Zoom) erhebliche Probleme bei der Kompensation. Auch Bewegungen der zugrunde liegenden Objekte im dreidimensionalen Raum erzeugen Verzerrungen bei der perspektivischen Abbildung in der Kamera, die aufgrund des rein translatorischen Bewegungsmodells nicht kompensiert werden können.

Des Weiteren trägt die Codierung der Bewegungsvektoren zu der Gesamtdatenrate bei und beeinflusst somit den Codiergewinn. Bei großen Objekten können wenige größere Blöcke verwendet werden, um die Bewegung zu beschreiben. Problematisch

wird die Bewegungskompensation an Objektgrenzen, da der frei werdende Hintergrund eine andere Bewegung als das Objekt besitzt und somit unterschiedliche Bewegungen innerhalb eines Blockes auftreten können. Da jedem Block nur ein Vektor zugeordnet ist, kann in diesen Fällen die Bewegung nicht korrekt kompensiert werden. Außerdem wird bei der Bewegungskompensation eines größeren Blockes die Prädiktion an dessen Rändern ungenauer [41], auch wenn es sich um ein einzelnes Objekt handelt. Daher sollte die Blockgröße klein gewählt werden, was jedoch zu einer größeren Anzahl der zu codierenden Bewegungsvektoren führt. In Abbildung 1.1 ist die benötigte Datenrate der Bewegungsvektoren sowie des Residuums für verschiedene Blockgrößen gezeigt. Dabei wurde der Coder aus Abschnitt 3.2.1 verwendet. Es ist zu erkennen, dass die Kosten für die Übertragung von Bewegungsvektoren bei sehr kleinen Blöcken deutlich höher sind als der Nutzen durch die verbesserte Prädiktion. Dies ist der Grund für die minimale Blockgröße von 4×4 Bildpunkten bei AVC.

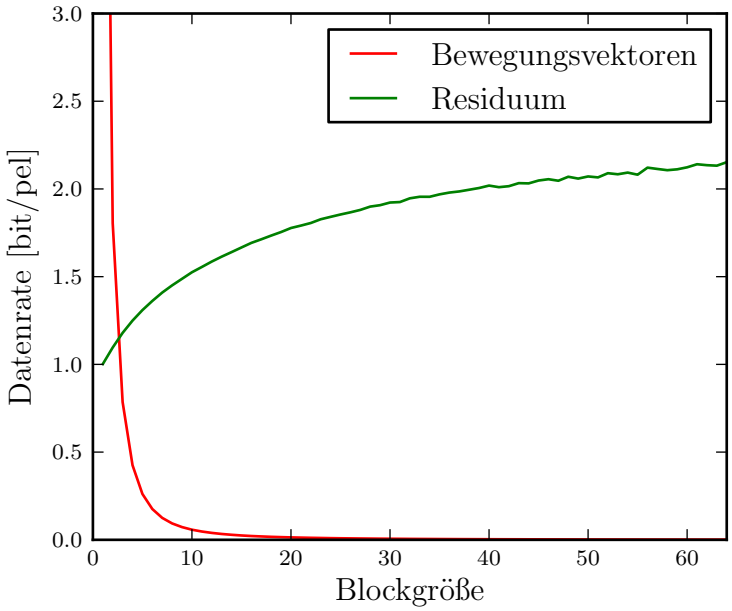


Abbildung 1.1: Datenraten zur Codierung der Bewegungsvektoren und der Residuuminformation für verschiedene Blockgrößen.

Auch hat die Genauigkeit, mit der die Bewegung geschätzt wird, einen großen Einfluss auf das Prädiktionsergebnis. Bewegungsvektoren mit Subpel-Auflösung verbessern die Prädiktion, da die zugrunde liegende Bewegung der Objekte unabhängig von dem Abtastraster der Kamera ist. Jedoch wächst auch die benötigte Datenrate der Bewegungsvektoren bei steigender Genauigkeit. Es wurde gezeigt, dass der optimale Kompromiss zwischen der Datenrate zur Codierung des Prädiktionsfehlers und

der Datenrate für die Bewegungsvektoren zwischen 1/4-pel- und 1/8-pel-Auflösung liegt [17, 67]. In dem aktuellen Standard AVC ist die Genauigkeit für Bewegungsvektoren auf 1/4-pel beschränkt.

Lösungsansatz

Es wird deutlich, dass die Bewegungsvektoren einen großen Einfluss auf den Codiergewinn haben. Mit Hilfe von MVP und MVC wird versucht, örtliche und zeitliche Homogenitäten der Bewegungsvektoren auszunutzen, um die benötigte Datenrate zu verringern. DMVD verfolgt einen anderen Ansatz und schätzt die Bewegung anhand bereits decodierter Bildbereiche. Ähnlich wie bei MVP wird davon ausgegangen, dass das Bewegungsvektorfeld örtlich homogen ist und somit benachbarte Bildpunkte derselben Bewegung folgen. Wie bei der Zuordnung in Tabelle 1.1 ersichtlich wird, verwendet keiner dieser Ansätze das bereits decodierte Bildsignal um zeitliche Homogenitäten zur besseren Codierung auszunutzen. Daher ist das Ziel dieser Arbeit, mit Hilfe einer decoderseitigen Bewegungsschätzung (*Decoder-side Motion Estimation*, DSME), zeitliche Korrelationen innerhalb der Videosequenz zu bestimmen und somit den Codiergewinn weiter zu erhöhen. Außerdem lässt sich dieses Verfahren effizient mit DMVD kombinieren, welches örtliche Korrelationen berücksichtigt.

Tabelle 1.1: Einordnung verschiedener Algorithmen zur Reduzierung der Bewegungsinformation.

	Bewegungshomogenität	
	örtlich	**zeitlich**
Bewegungsvektoren	Motion Vector Prediction	Motion Vector Competition
Bildinhalt	Decoder-side Motion Vector Derivation	Decoder-side Motion Estimation

Bewegungsschätzung am Decoder ist bereits aus dem Gebiet *Distributed Video Coding* (DVC) bekannt. Im Gegensatz zur beschriebenen konventionellen Hybridcodierung versucht DVC jedoch nicht, die Abhängigkeiten innerhalb einer Videosequenz mit Hilfe von komplexen Algorithmen am Encoder zu bestimmen. Stattdessen soll die Komplexität aufbauend auf den Theoremen von Slepian-Wolf [63] und Wyner-Ziv [74] zum Decoder verschoben werden [55], um somit ganz neue Anwendungsgebiete zu erschließen. Architekturbedingt wird hierbei die Bewegung allein auf der Decoderseite geschätzt, sodass keine Bewegungsinformationen übertragen werden. Jedoch konnte auch nach mehr als zehn Jahren Forschung die Codiereffizienz von aktuellen Standards nicht erreicht werden.

Daher wird in dieser Arbeit ein Ansatz vorgestellt, welcher Teile von DVC in die konventionelle prädiktive Videocodierung überführt. Hierdurch soll eine bessere Kompression erzielt werden. Unter der Annahme, dass die Bewegung über mehrere Einzelbilder hinweg homogen ist, kann aus zeitlich benachbarten Bildern das aktuell zu codierende Bild prädiziert werden. Dieses Vorgehen ist auch bekannt als *Frame Rate Up Conversion* (FRUC) und findet zum Beispiel bei der Zwischenbildberechnung in LCD-TVs [13] Anwendung. Ist dieses prädizierte Bild am Encoder sowie am Decoder bekannt, muss lediglich der Prädiktionsfehler, wie zuvor beschrieben, übertragen werden. Die Codierung von Bewegungsinformationen kann hingegen reduziert werden oder ganz entfallen.

Durch die Integration innerhalb der Prädiktionsfehlercodierung werden jedoch ganz andere Anforderungen an die Bewegungsschätzung gestellt als bei FRUC. Ein prädiziertes Zwischenbild muss bei der Darstellung auf einem LCD-TV eine gleichmäßige Qualität besitzen, damit keine störenden Artefakte auftreten. Bei DSME sind lokale Artefakte innerhalb der Prädiktionsbildes nicht störend, da diese durch die Residuencodierung eliminiert werden können. Solange andere Bereiche genauer prädiziert werden konnten, kann im Mittel ein Codierungsgewinn erzielt werden. Des Weiteren muss das zu Grunde liegende Bewegungsmodell angepasst werden, da bei DSME die zeitlichen Abstände zwischen zwei Referenzbildern sehr groß sein können.

Aufbau der Arbeit

Die Grundlagen der hybriden Videocodierung werden in Kapitel 2 vermittelt und die Referenzverfahren vorgestellt. Dabei wird der Schwerpunkt auf die bewegungskompensierende Prädiktion und die Codierung zugehöriger Bewegungsvektoren gelegt. Die Grenzen der mit Hilfe der decoderseitigen Bewegungsschätzung erreichbaren Codiereffizienz werden in Kapitel 3 ermittelt. Dazu wird ein Modell aus der Literatur erweitert und anhand von experimentellen Ergebnissen eines vereinfachten Coders verifiziert. In Kapitel 4 wird eine Architektur zur decoderseitigen Bewegungsschätzung vorgestellt, welche auf einer modifizierten Referenzbildliste basiert. Dabei wird insbesondere auf die Einschränkungen bei der Bewegungsschätzung durch die Annahme zeitlich konstanter Bewegung eingegangen, und es werden angepasste Algorithmen vorgestellt. Am Ende dieses Kapitels wird ein Verfahren aus der Literatur vorgestellt, welches örtlich konstante Bewegung voraussetzt und anschließend beide Verfahren für eine noch bessere Kompression kombiniert. Kapitel 5 gibt eine subjektive sowie objektive Bewertung der vorgestellten Architektur anhand experimenteller Ergebnisse. Dabei wird zunächst die bewegungskompensierende Interpolation mit dem aktuellen Stand der Technik verglichen. Anschließend wird die Codiereffizienz ausgewertet und mit zwei Referenzverfahren verglichen. Es folgt eine detaillierte Auswertung der genutzten Codiermodi für das aktuellere Referenzvefahren. Die Ergebnisse dieser Arbeit werden in Kapitel 6 noch einmal zusammengefasst.

2 Grundlagen der Videocodierung

In diesem Kapitel werden die grundlegenden Prinzipien der hybriden Videocodierung erläutert. Der folgende Abschnitt beschreibt den Aufbau eines allgemeinen Videosignals. Anschließend wird der Aufbau eines Videocoders, wie er in aktuellen Standards Verwendung findet, erläutert. Anhand des AVC-Standards wird die bewegungskompensierte Prädiktion genauer beschrieben, da diese innerhalb dieser Arbeit modifiziert wird. Außerdem werden die wesentlichen Unterschiede des HEVC-Testmodells [46], welches ebenfalls als Referenz in dieser Arbeit dient, gegenüber AVC aufgezeigt.

2.1 Mathematische Beschreibung von Videosignalen

Die Umwandlung von Licht in elektrische Ladung während der Aufnahme eines Bildes erfolgt in heutigen Videokameras meist mit Hilfe von CCD-Elementen (*Charge Coupled Device*). Dabei wird das einfallende Licht zeitlich und örtlich abgetastet. Die gespeicherte Ladung ist proportional zur aufgenommenen Lichtenergie und somit auch zur Helligkeit. Die Ladungswerte der CCD-Elemente werden ausgelesen, gleichförmig quantisiert und mit einer PCM codiert. In dieser Arbeit kommen Sequenzen zur Verwendung, welche auf 256 und 1024 Werte quantisiert wurden, was einer PCM-Codierung mit 8 beziehungsweise 10 bit pro Abtastwert entspricht.

In Abbildung 2.1 ist solch ein abgetastetes Videosignal schematisch dargestellt [66]. Die Abstände zwischen den einzelnen CCD-Elementen in horizontaler und vertikaler Richtung werden dabei mit T_x und T_y bezeichnet. Mit Hilfe der zeitlichen Abtastperiode T_t kann die Bildwiederholfrequenz $f = \frac{1}{T_t}$ des Videosignals berechnet werden. Wie in Abbildung 2.1 veranschaulicht, spiegelt die Funktion $s(n_x, n_y, n_t)$ den Zusammenhang zwischen den diskreten Koordinaten n_x, n_y, n_t und dem dazugehörigen Abtastwert des CCD-Elements wider.

Der Abstand zwischen zwei benachbarten Abtastwerten wird im Folgenden mit der Einheit pel (*Picture Element*) beschrieben. Demnach entspricht der Abstand 1 pel der physikalischen Länge T_x in horizontaler und T_y in vertikaler Richtung. Ein Block der Größe M_x pel$\times M_y$ pel besteht aus einem rechteckigen Bereich mit insgesamt $M_x M_y$ Abtastwerten. Bei quadratischen Blöcken ist es ausreichend, wenn lediglich die Kantenlänge angegeben wird. Eine Blockgröße von M pel entspricht somit einem Quadrat von M^2 Bildpunkten.

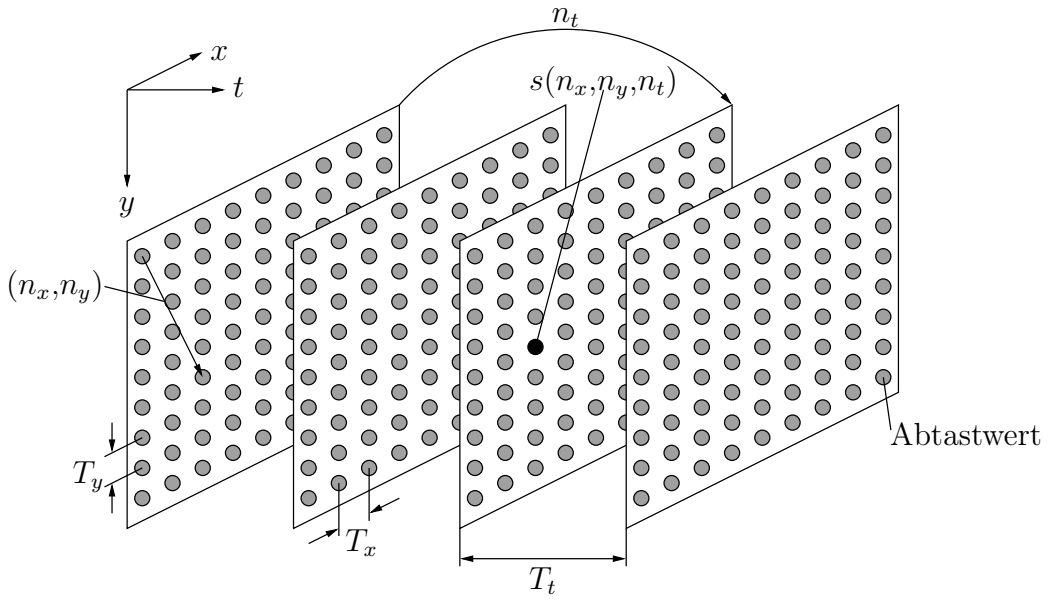


Abbildung 2.1: Darstellung des horizontalen, vertikalen und zeitlichen Abtastrasters eines Videosignals.

Bei der Codierung des Videosignals ist der absolute Zeitpunkt t eines Bildes, oder n_t in diskreten Koordinaten, nicht von großer Bedeutung. Viel interessanter sind die relativen Bezüge von zeitlich benachbarten Bildern. Daher werden, unter der Annahme, dass ein Bild zum Zeitpunkt n_t betrachtet wird, folgende vereinfachte Schreibweisen eingeführt:

$$\begin{aligned} s &\mathrel{\hat{=}} s(n_x, n_y, n_t), \\ s_{-1} &\mathrel{\hat{=}} s(n_x, n_y, n_t - 1), \\ s_{-2} &\mathrel{\hat{=}} s(n_x, n_y, n_t - 2), \\ s_{+1} &\mathrel{\hat{=}} s(n_x, n_y, n_t + 1). \end{aligned}$$

Die einzelnen Bilder eines typischen Videosignals haben hohe Abhängigkeiten untereinander. Daher werden bei der hybriden Videocodierung die Signaländerungen von Bild zu Bild berücksichtigt. Das erste Bild wird durch die Luminanzamplituden aller Abtastwerte beschrieben. Bei folgenden Bildern können die zeitlichen Änderungen des Bildsignals zur Beschreibung verwendet werden [52]. Das Prinzip der hybriden Videocodierung wird im folgendem Abschnitt genauer erläutert.

2.2 Hybride Videocodierung

Die hybride Videocodierung basiert auf zwei Ansätzen. Zum einen wird das zu codierende Bild s aus bereits decodierten Daten prädiziert. Da der Decoder ebenfalls in

der Lage ist, dieses prädizierte Bild zu erstellen, muss lediglich dessen Differenz zum Originalbild – auch Prädiktionsfehler genannt – übertragen werden. Zum anderen wird der Prädiktionsfehler transformiert, um eventuelle Korrelationen benachbarter Bildpunkte zur Datenreduktion auszunutzen und eine wahrnehmungsangepasste Quantisierung zu ermöglichen.

Üblicherweise wird ein Bild in kleinere Blöcke unterteilt und diese Blöcke werden sequentiell codiert. Basiert die Prädiktion eines Blockes lediglich auf bereits codierten Informationen des aktuellen Bildes, wie zum Beispiel benachbarte Blöcke, spricht man von einem Intra-Bild, oder abgekürzt I-Bild. Im Gegensatz dazu nutzt eine Interprädiktion auch andere, bereits codierte Bilder. Hier kann man zwischen unidirektional prädizierten Bildern (P-Bilder) und bidirektional prädizierten Bildern (B-Bilder) unterscheiden. Blöcke in P-Bilder nutzen lediglich Informationen aus einem bereits decodiertem Bild, während bei B-Bilder mehrere decodierte Bilder zur Prädiktion eines Blockes verwendet werden können.

In Abbildung 2.2 ist das vereinfachte Blockdiagramm eines Encoders gezeigt, welcher eine Bewegungskompensation zu Interprädiktion nutzt. Die Intraprädiktion wird in dieser Abbildung nicht weiter berücksichtigt, da diese im Rahmen dieser Arbeit ohne Bedeutung ist.

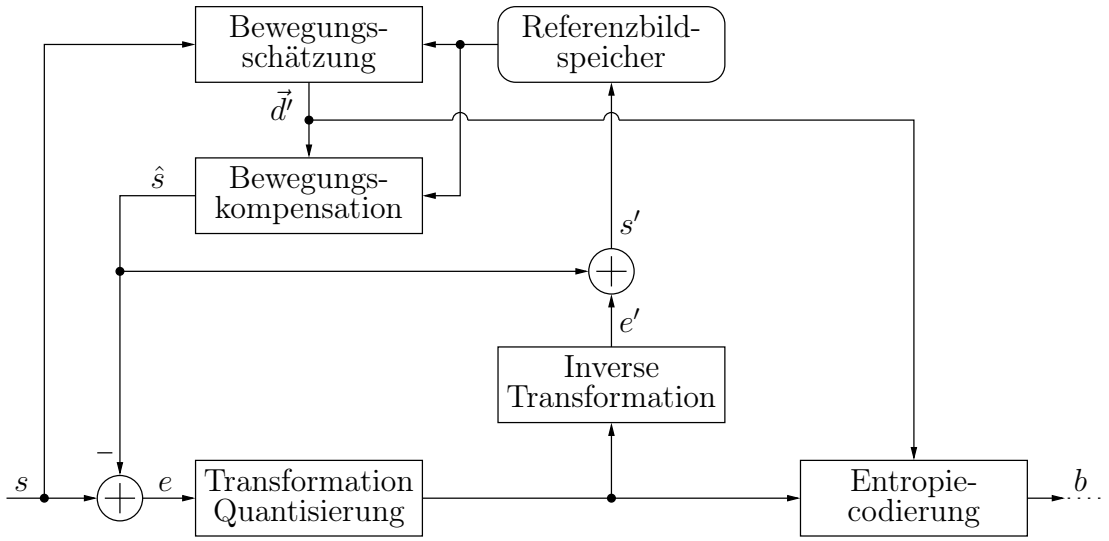


Abbildung 2.2: Vereinfachtes Blockdiagramm eines Encoders zur hybriden Videocodierung.

Im ersten Schritt wird für jeden Block die Bewegung $\vec{d}$ zwischen dem zu codierenden Bild s und bereits codierten Bildern aus dem Referenzbildspeicher geschätzt. Mit Hilfe der so gewonnenen Bewegungsvektoren $\vec{d}'$ wird ein bewegungskompensiertes Bild $\hat{s}$ aus den Referenzbildern berechnet. Da dem Decoder das Originalbild s zur Bewegungsschätzung nicht vorliegt, müssen die am Encoder ermittelten Bewegungs-

vektoren $\vec{d}'$ codiert und übertragen werden. Anschließend wird der aus der Differenz von s und $\hat{s}$ bestimmte Prädiktionsfehler e transformiert und quantisiert. Im letzten Schritt vor der Übertragung werden die quantisierten Transformationskoeffizienten codiert.

Wie in Abbildung 2.3 gezeigt, prädiziert der Decoder das aktuell zu decodierende Bild mit Hilfe der empfangenen Bewegungsvektoren $\vec{d}'$ und den Bildern aus dem Referenzbildspeicher. Aufgrund der Quantisierung am Encoder entspricht e' am Deco-

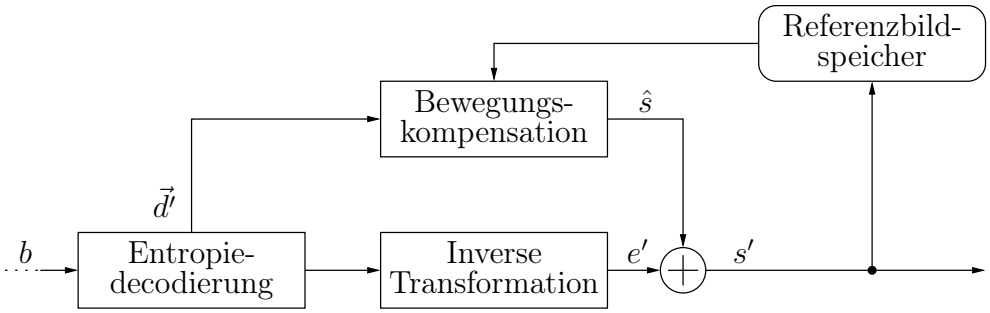


Abbildung 2.3: Vereinfachtes Blockdiagramm eines Decoders zur hybriden Videocodierung.

der nicht exakt dem Prädiktionsfehler e. Daher wird e' zur besseren Unterscheidung als Residuum bezeichnet. Die Kombination des Residuums mit dem bewegungskompensierten Bild $\hat{s}$ ergibt das rekonstruierte Bild s', welches zur Decodierung von folgenden Bildern ebenfalls als Referenzbild gespeichert wird.

Da die Referenzbildspeicher am Encoder und Decoder identischen Inhalt besitzen müssen, wird auch am Encoder, wie in Abbildung 2.2 gezeigt, das decodierte Bild s' aus dem prädizierten Bild $\hat{s}$ und dem Residuum e' erstellt und dem Speicher zugeführt.

Im folgenden Abschnitt werden die einzelnen Komponenten der hybriden Videocodierung am Beispiel des AVC-Standards [25] genauer erläutert. Dem aktuellen Stand der Forschung entspricht das sogenannte HEVC-Testmodell, welches von der ISO/IEC und der ITU-T als Grundlage eines neuen Videocodierstandards genutzt wird. Daher werden im Abschnitt 2.2.2 die wesentlichen Unterschiede zwischen AVC und HEVC hervorgehoben.

2.2.1 AVC-Standard

In AVC wird jedes Bild in Blöcke mit der Größe von 16×16 Bildpunkten – die sogenannten Makroblöcke – unterteilt. Die einzelnen Makroblöcke werden zeilenweise codiert, sodass im Allgemeinen bereits codierte Blöcke über und links des aktuellen Blockes zur Verfügung stehen. Diese können zur Intraprädiktion verwendet werden,

oder wie in den folgenden Abschnitten beschrieben, bei der Codierung der Bewegungsvektoren genutzt werden.

Zur Minimierung der Redundanzen innerhalb des codierten Bitstroms stehen bei AVC zwei verschiedene Codiermethoden zur Verfügung. Die kontext-adaptive Codierung mit variabler Codewortlänge (*Context-Adaptive Variable Length Coding*, CAVLC) und die kontext-adaptive binäre arithmetische Codierung (*Context-Adaptive Binary Arithmetic Coding*, CABAC). Trotz der größeren Komplexität von CABAC wird diese Methode aufgrund der höheren Effizienz sehr häufig bevorzugt. Daher beziehen sich alle folgenden Erläuterungen und auch die experimentellen Ergebnisse in Kapitel 5 auf die Codierung mit Hilfe von CABAC.

Bewegungskompensierende Prädiktion

Im Gegensatz zur Intracodierung, bei der ein Makroblock nur aus bereits decodierten Daten des aktuell zu codierenden Bildes prädiziert wird, werden bei der Intercodierung Referenzbilder zur bewegungskompensierten Prädiktion verwendet (Abbildung 2.4). Zur Codierung eines Blockes sendet der Encoder die Position des Blockes

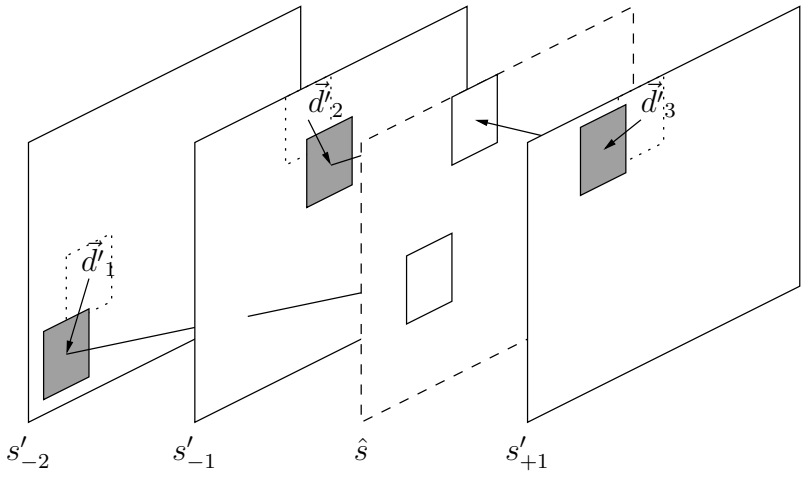


Abbildung 2.4: Interprädiktion von $\hat{s}$ aus den grau hinterlegten Blöcken bereits codierter Referenzbilder. Die gestrichelten Blöcke bilden die Positionen der zu prädizierenden Blöcke in die Referenzbilder ab.

innerhalb des Referenzbildes, welcher zur Prädiktion verwendet wird. Neben diesem Bewegungsvektor $\vec{d}'_i$ muss dem Decoder auch signalisiert werden, welches Referenzbild verwendet wurde. Anschließend wird noch das Residuum übertragen, um den entstandenen Prädiktionsfehler e zu kompensieren. Wie bereits erwähnt, wird das zu codierende Bild als P-Bild bezeichnet, solange zur Prädiktion eines Blockes lediglich ein Referenzbild verwendet wird. Da in Abbildung 2.4 jedoch mindestens ein

Block aus zwei Referenzbildern prädiziert wird, ist dort die Prädiktion eines B-Bildes abgebildet.

In AVC ist es möglich, die Bewegung mit 1/4-pel-Genauigkeit zu schätzen. Die benötigten Zwischenwerte werden mit Hilfe einer zweistufigen Interpolation berechnet. Im ersten Schritt werden die 1/2-pel-Positionen mit Hilfe eines Wienerfilters [68] mit sechs Koeffizienten interpoliert. Anschließend werden die 1/4-pel-Positionen mit einer einfachen bilinearen Interpolation berechnet.

Da zu jedem Block mindestens ein Bewegungsvektor übertragen werden muss, sollte die Blockgröße möglichst groß gewählt werden, um die Gesamtzahl der Bewegungsvektoren gering zu halten. Jedoch wird die Prädiktion für große Blöcke ungenauer, sodass das zu übertragene Residuum ansteigt. Um den besten Kompromiss zwischen der Rate zur Codierung der Residuumsinformation und der Rate für die Bewegungsvektoren zu erreichen, kann ein Makroblock, wie in Abbildung 2.5 gezeigt, in kleinere Partitionen unterteilt werden. Eine 8×8 Partition kann außerdem noch in 4×8, 8×4 und 4×4 Subpartitionen aufgeteilt werden.

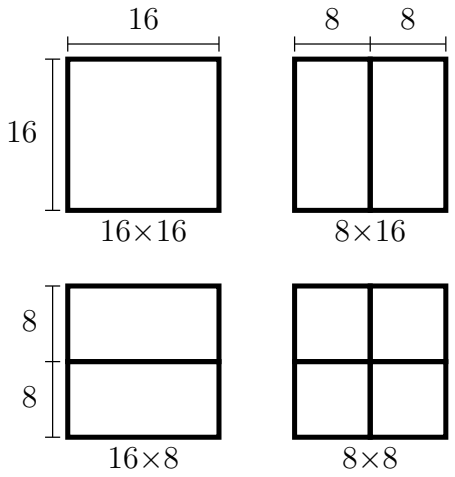


Abbildung 2.5: In der Intercodierung verwendeten Makroblockpartitionen.

Codierung des Referenzbildindex

Wie bereits beschrieben, muss zu jedem Block signalisiert werden, welches Referenzbild zur bewegungskompensierten Prädiktion verwendet werden soll. Dazu werden alle Referenzbilder in einer sogenannten Referenzbildliste angeordnet und können somit über die Position in dieser Liste ausgewählt werden. Bei B-Bildern existiert für die bidirektionale Prädiktion eine zusätzlich Liste, in der die Referenzbilder anders angeordnet sind. Um den Speicherbedarf zu begrenzen, wird die maximale Anzahl der Referenzbilder zu Beginn der Codierung festgelegt. Die Referenzbildliste

ist derart aufgebaut, dass Bilder mit einem geringen zeitlichen Abstand zum zu codierenden Bild am Anfang der Liste stehen. Es kann davon ausgegangen werden, dass diese Bilder eine größere Korrelation mit dem zu codierenden Bild haben und somit häufiger ausgewählt werden. Daher wird bei AVC der Referenzbildindex mit unterschiedlichen Codewortlängen codiert. Der Index wird mit Hilfe der sogenannten *Unary Binarization* in ein binäres Codewort umgewandelt und anschließend mit Hilfe von CABAC codiert. Bei der *Unary Binarization* wird eine zu codierende, positive Zahl n durch n Einsen und eine abschließende Null repräsentiert. In Tabelle 2.1 ist diese Umwandlung in binäre Codewörter beispielhaft erläutert.

Tabelle 2.1: Umwandlung des Referenzbildindex in ein binäres Codewort wie in [25] beschrieben.

Referenzbildindex	Codewort
0	0
1	1 0
2	1 1 0
3	1 1 1 0
...	...

Codierung der Bewegungsvektoren

Da die zu codierenden Blöcke im Vergleich zu den Objektgrößen in einer Sequenz sehr klein sind, wird davon ausgegangen, dass das Bewegungsvektorfeld in einer begrenzten Nachbarschaft weitestgehend homogen ist. Daher kann mit Hilfe der Bewegungsvektoren benachbarter Blöcke die Bewegung des aktuellen Blockes prädiziert werden, wodurch lediglich der Prädiktionsfehler übertragen werden muss. In AVC werden, wie in Abbildung 2.6 angedeutet, die Vektoren von bis zu drei benachbarten Blöcken (A, B und C) zur Prädiktion verwendet. Bei diesem, als *Motion Vector*

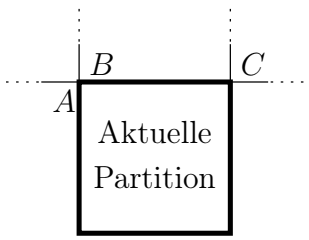


Abbildung 2.6: Bewegungsvektoren der bereits codierten Blöcke A, B und C werden verwendet, um die Bewegung der aktuellen Partition zu schätzen.

Prediction (MVP) bezeichneten Verfahren, wird der Median jeweils für die x- und y-Komponente berechnet und als prädizierter Bewegungsvektor verwendet. Für den Fall, dass nicht alle drei benachbarten Blöcke zur Verfügung stehen, wurden in AVC weitere Berechnungsvorschriften definiert [59].

Der prädizierte Bewegungsvektor wird von dem aktuell zu codierenden Bewegungsvektor abgezogen. Die Differenz (*Motion Vector Differenz*, MVD) wird mit Hilfe einer Exp-Golomb-Binarisierung [25] in ein binäres Codewort umgewandelt und ebenfalls mit CABAC codiert.

Jedoch wird nicht für jeden intercodierten Block ein Bewegungsvektor übertragen. Bei dem sogenannten Skipmodus wird keine Bewegungsinformation übertragen und stattdessen der mit Hilfe von MVP prädizierte Vektor zur Bewegungskompensation genutzt.

Codierung des Prädiktionsfehlers

Örtliche Korrelationen innerhalb des Prädiktionsfehlers werden mit Hilfe einer speziellen diskreten Cosinustransformation (DCT) zur Reduktion der Datenrate ausgenutzt. Diese sogenannte Integer-DCT bietet im Gegensatz zur konventionellen DCT [1] den Vorteil, dass die Rechenoperationen auf ganzen Zahlen und nicht auf Fließkommazahlen durchgeführt werden. Somit wird eine Abweichung zwischen Encoder und Decoder aufgrund unterschiedlicher Rechengenauigkeiten bei der Verarbeitung von Fließkommazahlen vermieden. Außerdem lässt sich die Integertransformation durch einfache Additions- und Bitverschiebeoperationen implementieren. Lediglich zur Erhaltung der Energie ist eine Multiplikation der einzelnen Koeffizienten nötig [53]. In AVC stehen zwei Transformationen mit einer Größe von 4×4 beziehungsweise 8×8 Werten zur Verfügung.

Die Qualität der codierten Videosequenz sowie die benötigte Datenrate lassen sich über die Quantisierung der Transformationskoeffizienten steuern. Dabei werden die einzelnen Frequenzkomponenten unterschiedlich grob quantisiert, um der frequenzabhängigen Wahrnehmung des Betrachters Rechnung zu tragen. So wirken sich Quantisierungsfehler bei hohen Frequenzen weniger störend aus, als bei niedrigen Frequenzen.

Die CABAC-Codierung des Residuums erfolgt in vier Schritten. Als erstes werden die quantisierten Koeffizienten eines Blockes, wie in Abbildung 2.7 für einen 4×4-Block dargestellt, mit Hilfe der sogenannten Zickzackabtastung in eine eindimensionale Reihe abgebildet. Dann werden die Positionen aller Koeffizienten, welche nicht Null sind, codiert. Anschließend werden die entsprechenden Beträge und zuletzt die Vorzeichen codiert.

23	7	-8	0
16	3	0	0
0	1	0	0
0	0	0	0

Abbildung 2.7: Die Zickzackabtastung der dargestellten Koeffizienten erzeugt die Reihe $23, 7, 16, 0, 3, -8, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0$.

Steuerung des Encoders

AVC bietet aufgrund der verschiedenen Blockpartitionierungen, Referenzbilder und Transformationsgrößen eine Vielzahl von Möglichkeiten zur Codierung eines Makroblockes. Allein für die Partitionierung eines Makroblockes gibt es 259 verschiedene Aufteilungen. Jedoch ist die Wahl einer geeigneten Codierung nicht teil des Standards sondern Aufgabe des Encoders. In der AVC-Referenzsoftware JM [34], welche auch in dieser Arbeit Verwendung findet, wird mit Hilfe der Lagrange-Multiplikatorenregel der beste Kompromiss zwischen Datenrate und Verzerrung bestimmt [65]. Dazu testet der Encoder jede mögliche Kombination der Parameter und wählt die Konfiguration, welche die geringsten Lagrangekosten verursacht.

2.2.2 Unterschiede zum HEVC-Testmodell

Das *Joint Collaborative Team on Video Coding* (JCT-VC) der ISO/IEC und der ITU-T hat sich zum Ziel gesetzt, bis 2013 einen neuen Videocodierstandard zu entwickeln. Als Referenz dient das als HM bezeichnete HEVC-Testmodell [33]. Die für diese Arbeit interessanten Unterschiede zu AVC sollen hier kurz erläutert werden. Als Basis dient die Version HM 2.0 [46].

In HM wurde die starre Partitionierung der Makroblöcke fallen gelassen und durch eine allgemeine baumbasierte Struktur ersetzt. Dazu wird das Bild in sogenannten *Tree Blocks* (TB) aufgeteilt, deren Größe vor der Codierung festgelegt wird und maximal 64×64 Bildpunkte beträgt. Jeder dieser TBs wird mit Hilfe eines *Quadtrees*, wie am Beispiel in Abbildung 2.8 gezeigt, in *Coding Units* (CU) aufgeteilt, wodurch nur quadratische CUs möglich sind. Die minimale Größe einer CU ist auf 8×8 Bildpunkte begrenzt. Jede CU kann analog zu der Partitionierung in Abbildung 2.5 in bis zu vier *Prediction Units* (PUs) aufgeteilt werden.

Außerdem wurden weitere Transformationen für Blöcke von bis zu 32×32 Werten eingeführt, wodurch insbesondere für hochaufgelöste Videosequenzen Codiergewinne erzielt und Blockartefakte verringert werden.

Die Bewegungsvektorprädiktion (MVP) wurde in HM erweitert, um zeitliche Korrelationen der Bewegungsvektoren zu berücksichtigen: Bei der als *Advanced Motion Vector Prediction* (AMVP) bezeichneten Technik wird eine Liste von Kandidatenvek-

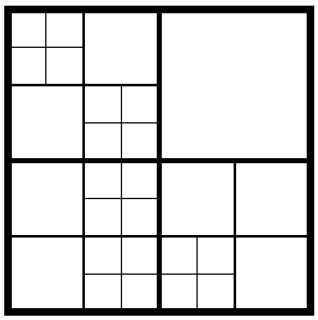


Abbildung 2.8: Beispielhafte Aufteilung eines *Tree Blocks* in *Coding Units*.

toren aus den Bewegungsvektoren zeitlich und örtlich benachbarter Blöcke erstellt und der Index des besten Kandidaten übertragen.

Des Weiteren wurden viele Neuerungen bei der Intraprädiktion eingeführt, welche für diese Arbeit jedoch nicht von Interesse sind.

3 Analyse der bewegungskompensierenden Prädiktion am Decoder

Wie im vorangegangenen Kapitel beschrieben, setzt sich die Datenrate zur Codierung eines Bildes aus den quantisierten Prädiktionsfehlern der Luminanz- und Chrominanzkomponenten, den zur Erstellung der Prädiktion benötigten Bewegungsvektoren, zusätzlichen Daten zur Signalisierung der gewählten Codiermodi und weiterer Parameter zusammen. Abbildung 3.1 zeigt die prozentuale Aufteilung der Datenrate am Beispiel der PeopleOnStreet-Sequenz, welche mit Hilfe eines AVC-Encoders bei unterschiedlichen Quantisierungsparametern codiert wurde. Die Sequenz PeopleOnStreet wird in Abschnitt 5.1 genauer beschrieben.

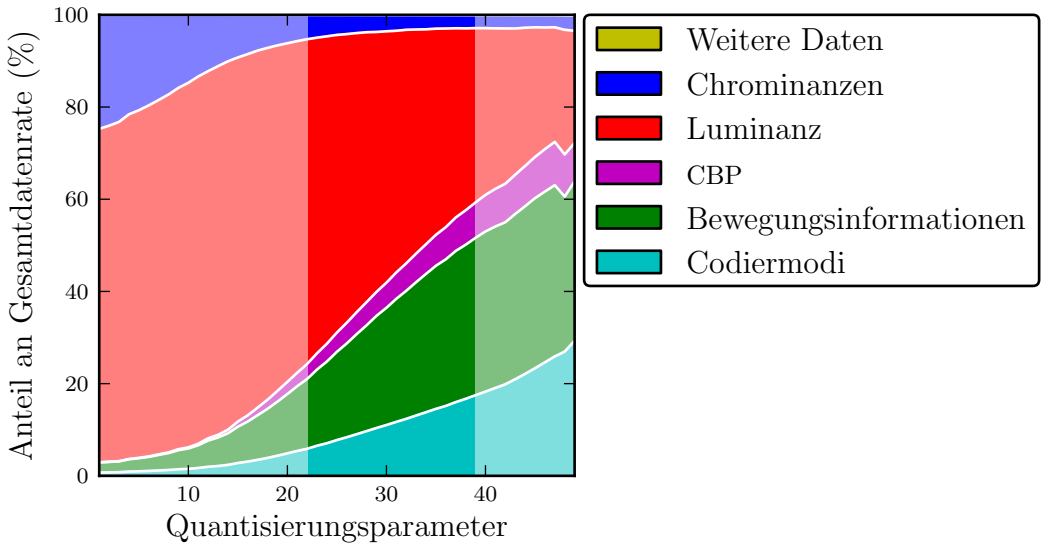


Abbildung 3.1: Aufteilung der Datenrate bei unterschiedlicher Quantisierung eines AVC codierten Bitstroms für die PeopleOnStreet-Sequenz. Die Qualität im hervorgehobene Abschnitt entspricht in etwa Rundfunkqualität.

Den größten Anteil zur Gesamtdatenrate verursacht die Codierung der Prädiktionsfehler für die Luminanz und Chrominanzen. Die Codierung der Bewegungsinformation trägt jedoch ebenfalls signifikant zur Gesamtdatenrate bei. Es ist deutlich zu

erkennen, dass der Anteil der Bewegungsvektoren bei geringen Datenraten (grobe Quantisierung mit großem Quantisierungsparameter) sehr hoch ist. Da nur eine geringe Qualität gefordert ist, muss eine niedrigere Rate zur Kompensation des Prädiktionsfehlers aufgewendet werden. Bei höherer Qualität sinkt der prozentuale Anteil der Bewegungsvektorrate, da zusätzliche Daten nötig sind, um den Prädiktionsfehler präzise zu kompensieren. Hierbei verringert sich jedoch lediglich der relative Anteil der Bewegungsvektorrate an der Gesamtdatenrate. Die absolute Rate für die Bewegungsvektoren steigt ebenfalls bei höherer Qualität. Dies ist darauf zurückzuführen, dass bei geringer Qualität häufiger der sogenannte Skipmodus Verwendung findet, bei dem kein Bewegungsvektor übertragen wird. Außerdem sind in Abbildung 3.1 der Anteil zur Signalisierung der verschiedenen Codiermodi und des Coded Block Pattern (CBP) gezeigt. CBP wird genutzt, um Bereiche innerhalb eines Blockes zu markieren, für welche Transformationskoeffizienten codiert worden sind. Obwohl alle weiteren Daten innerhalb des Bitstroms zusammengefasst wurden, sind deren Anteile sehr gering und kaum in Abbildung 3.1 zu erkennen.

Experimente haben ergeben, dass ein Spitzensignal-Rausch-Verhältnis (Peak Signal to Noise Ratio, PSNR) zwischen 30 dB bis 40 dB einer – für den subjektiven Betrachter – angenehmen Qualität entspricht. Diese Qualität wird bei Codierung der PeopleOnStreet-Sequenz mit Hilfe von AVC für einen Quantisierungsparameter zwischen 22 und 39 erreicht. Innerhalb dieses Bereichs liegt der durchschnittliche Anteil der Datenrate für die Bewegungsvektoren bei über 25 %. Wenn es möglich ist, die Bewegung am Decoder zu schätzen und somit keine Bewegungsvektoren übertragen werden müssen, kann die Gesamtdatenrate in diesem Qualitätsbereich theoretisch um diese 25 % verringert werden.

Die starke Absenkung der Bewegungsvektorrate bei einem Quantisierungsparameter von 48 ist auf den Skipmode zurück zu führen. Die gewünschte Qualität ist so gering, dass für viele Bereiche die Prädiktion des Skipmodes ausreichend ist, wodurch weniger Bewegungsvektoren codiert werden müssen. Auch bei einem Quantisierungsparameter von 49 nimmt die Datenrate für die Bewegungsvektoren weiter ab. Lediglich der prozentuale Anteil steigt aufgrund sehr geringer Luminanz- und Chrominanzraten leicht an.

3.1 Modellierung der Datenrate

Eine ideale Bewegungsschätzung am Decoder, mit deren Hilfe die vollständige Datenrate der Bewegungsinformationen eingespart werden könnte, ist nicht möglich, da nicht alle benötigten Informationen am Decoder vorhanden sind. Um jedoch die möglichen Gewinne bei der Verwendung einer decoderseitigen Bewegungsschätzung abschätzen zu können, soll aufbauend auf den Arbeiten [56, 57, 58] ein Modell dieser Datenrate erstellt werden. Im folgendem Abschnitt werden zunächst die Einschränkungen durch das verwendete Bewegungsmodell beschrieben. Anschließend wird die

Rate zur Kompensation des daraus entstandenen Prädiktionsfehlers modelliert. Mit Hilfe des erstellten Modells wird die optimale Blockgröße bei der Verwendung von decoderseitiger Bewegungsschätzung hergeleitet.

3.1.1 Einschränkungen durch das lineare Bewegungsmodell

Zur Bestimmung der Bewegung am Decoder wird im Folgenden davon ausgegangen, dass bereits decodierte Bilder vorhanden sind, von denen mindestens eines zeitlich vor (s'_{-1}) und eines nach (s'_{+1}) dem aktuell zu codierenden Bild s liegt. Die Bewegung zwischen diesen sogenannten Referenzbildern kann mit Hilfe eines Blockmatching-Algorithmus oder eines anderen Bewegungsschätzers ermittelt werden. Unter der Annahme konstanter Geschwindigkeit kann der entsprechende Block im zu codierenden Bild durch bidirektionale Interpolation prädiziert werden. In diesem Fall spricht man auch von einer linearen Bewegung. Wie in Abbildung 3.2 zu sehen, ergibt sich die Position des Blockes innerhalb des zu codierenden Bildes aus dem halbierten Bewegungsvektor zwischen den Referenzbildern. $\check{s}$ ist dabei das mit Hilfe decoderseitiger Bewegungsschätzung ermittelte Interpolationsbild. Eine genaue Beschreibung der Algorithmen zur Bewegungschätzung am Decoder ist in Kapitel 4 nachzulesen.

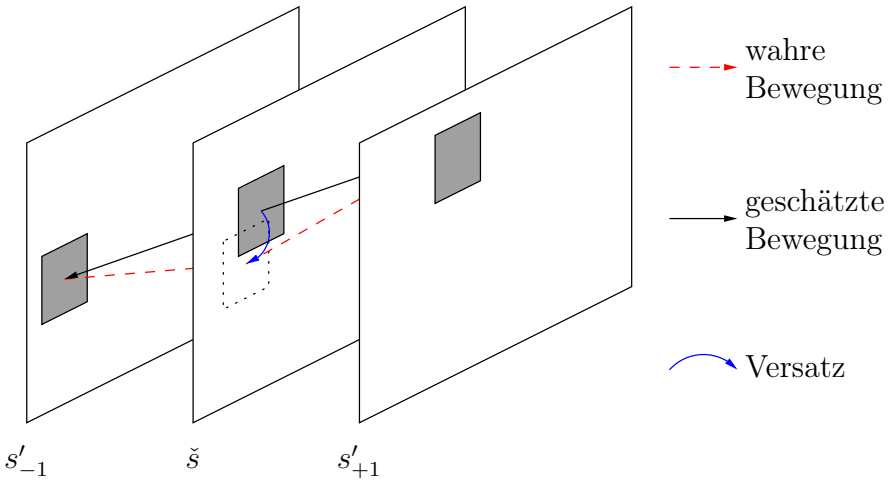


Abbildung 3.2: Bewegungskompensation unter der Annahme linearer Bewegung. Beschleunigte Bewegung zwischen den Referenzbildern erzeugt einen Versatz.

Jedoch ist die Annahme der konstanten Geschwindigkeit zwischen den Referenzbildern nicht immer korrekt. Eine beschleunigte Bewegung, wie sie beispielhaft in Abbildung 3.2 skizziert ist, verursacht einen Versatz zwischen dem interpolierten Block in $\check{s}$ und dem Originalbild s. Diese fehlerhafte Interpolation kann auf zwei

Arten kompensiert werden: Zum einen kann die Datenrate zur Codierung des Fehlers zwischen dem mit Hilfe der decoderseitigen Prädiktion erstelltem Bild $\check{s}$ und dem Originalbild s angepasst werden, um den zusätzlich entstandenen Fehler zu kompensieren. Zum anderen kann der entstandene Versatz durch einen zusätzlich zu übertragenden Bewegungsvektor kompensiert werden. In diesem Fall wird das bewegungs- und versatzkompensierte Bild als Prädiktionsbild verwendet und der verbleibende Fehler codiert. Da der Versatz eine kleinere Entropie als die konventionellen Bewegungsvektoren besitzt, kann dieser mit einer geringeren Datenrate übertragen werden.

Ein leichter Versatz hat bereits einen starken Anstieg des Prädiktionsfehlers zur Folge, weshalb es im Allgemeinen günstiger ist, einen zusätzlichen Bewegungsvektor zu codieren. Daher wird in der folgenden Modellierung die Kompensation nichtlinearer Bewegung durch zusätzliche Bewegungsvektoren angenommen.

Die Gesamtdatenrate R setzt sich somit aus der Rate für die Bewegungsvektoren zur Kompensation von nichtlinearer Bewegung R_V und der benötigten Rate zur Codierung des Residuums R_R zusammen:

$$R = R_V + R_R. \tag{3.1}$$

In den folgenden Abschnitten werden geeignete Modelle für die Raten zur Codierung des Versatzes durch nichtlineare Bewegung und des Prädiktionsfehlers entworfen.

3.1.2 Modellierung der Datenrate zur Übertragung des Versatzes

Der in Abbildung 3.2 gezeigte Versatz wird durch beschleunigte Bewegung zwischen den Referenzbildern verursacht. Die zugrunde liegenden physikalischen Objekte können entweder selbst eine beschleunigte Bewegung ausführen oder aber durch die perspektivische Abbildung in der Kamera zu einer nichtlinearen Bewegung führen. Diese beiden Ursachen werden gemeinsam betrachtet und durch die Beschleunigung a modelliert. In Abbildung 3.3 ist die beschleunigte – und somit nichtlineare – Bewegung an einem eindimensionalen Beispiel gezeigt. T_t ist dabei die Abtastzeit der einzelnen Bilder und kann aus dem Kehrwert der Bildwiederholungsfrequenz f_t berechnet werden.

Ein Block im ersten Referenzbild zum Zeitpunkt $t = 0T_t$ hat sich von der Position p_0 an die Stelle p_2 im zweiten Referenzbild ($t = 2T_t$) verschoben. Der ermittelte Bewegungsvektor ist somit $V = p_2 - p_0$. Durch die Annahme der linearen Bewegung wird das Objekt im zu codierenden Bild zum Zeitpunkt $1T$ an der Stelle

$$\hat{p}(t) = \frac{V}{2T_t}t + p_0 \tag{3.2}$$

$$\Rightarrow \hat{p}(T_t) = \frac{V}{2} + p_0 \tag{3.3}$$

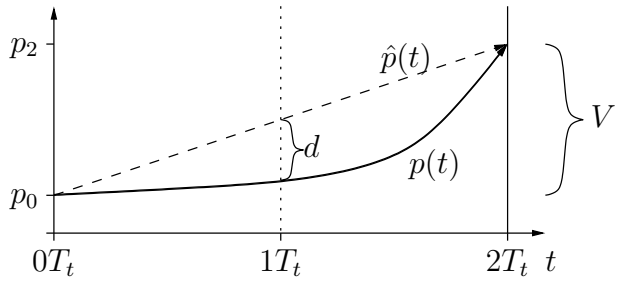


Abbildung 3.3: Beschleunigte Bewegung eines Blockes von der Position p_0 im ersten Referenzbild an die Position p_2 im folgenden Referenzbild.

bidirektional interpoliert. Jedoch wurde das Objekt um a konstant beschleunigt, wodurch die wahre Position

$$p(t) = \frac{1}{2}at^2 + \left(\frac{V}{2T_t} - aT_t\right)t + p_0 \tag{3.4}$$

$$\Rightarrow p(T_t) = \frac{V}{2} + p_0 - \frac{1}{2}aT_t^2 \tag{3.5}$$

ist. Der Versatz d, also die Differenz zwischen geschätzter Bewegung und wahrer Bewegung des Blockes, ist somit

$$\begin{aligned} d &= \hat{p}(T_t) - p(T_t) \\ &= \frac{1}{2}aT_t^2. \end{aligned} \tag{3.6}$$

Da der Versatz durch die Geschwindigkeitsänderung physikalischer Objekte hervorgerufen wird, treten sehr hohe Beschleunigungen nur selten auf. Kleine Beschleunigungen sind dagegen häufiger und können außerdem durch leichtes Wackeln der Kamera verursacht werden. Daher wird a als mittelwertfreie, normalverteilte Zufallsvariable angenommen:

$$a \sim \mathcal{N}(0,\sigma_a^2). \tag{3.7}$$

Die Varianz der Beschleunigung hängt von der jeweiligen Videosequenz ab und kann mit Hilfe der realen Bewegung zwischen den Referenzbildern und der Bewegung zwischen Referenzbild und dem zu codierendem Bild berechnet werden.

Im Folgenden wird davon ausgegangen, dass sich die Abbildung eines physikalischen Objekts in der Kamera aus mehreren Blöcken zusammensetzen lässt. Daher können zur Codierung des Versatzes Nachbarschaftsbeziehungen der einzelnen Blöcke ausgenutzt werden. In Abbildung 3.4 ist beispielhaft ein Teil eines Bildes mit fünf Objekten O_1 bis O_5 skizziert. Jedes dieser Objekte hat eine unterschiedliche Beschleunigung und damit einen Versatz d_i $(i = 1 \ldots 5)$ im interpoliertem Bild im Vergleich zur wahren Bewegung.

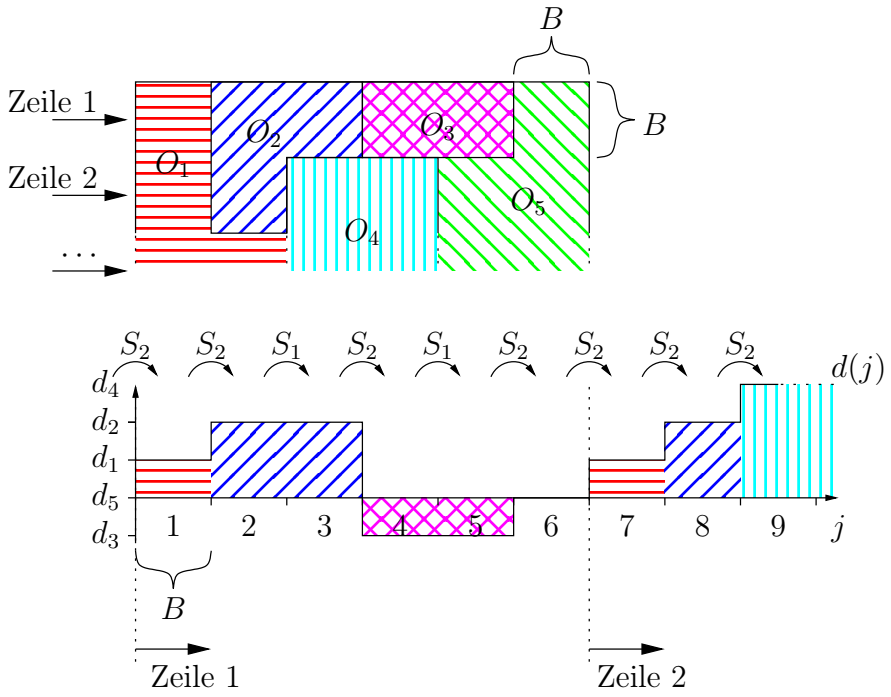


Abbildung 3.4: Verlauf des Versatzes durch nichtlineare Bewegung bei einer Blockgröße von $B \times B$ Bildpunkten.

Neben der Aufteilung des Bildes in verschiedene Objekte ist auch die Größe des Versatzes bei zeilenweisem Ablesen skizziert. Die Funktion $d(j)$ soll diese skizzierte Kurve beschreiben und entspricht dem Versatz des j-ten Blockes. Zur einfacheren Darstellung wird nur eine Richtungskomponente berücksichtigt. In der Praxis besteht der Versatzvektor $\vec{d_i}$ jedoch aus einer x- und einer y-Komponente.

Zur Codierung des Versatzes kann zwischen zwei Zuständen unterschieden werden. Im Zustand S_1 gehört ein Block zu dem selben Objekt wie der vorangegangene Block. Somit ist auch der der Versatz $d(j)$ des aktuellen Blockes identisch mit dem Versatz $d(j-1)$ des vorangegangenen Blockes. Im zweiten Zustand S_2, wo benachbarte Blöcke zu unterschiedlichen Objekten gehören, steht der Versatz dieser Blöcke in keinerlei Verbindung. Dies lässt sich mit der Annahme begründen, dass die zugehörigen Objekte unterschiedlichen Beschleunigungen ausgesetzt sind.

$$S_1: \quad d(j) = d(j-1), \tag{3.8}$$

$$S_2: \quad d(j) = \frac{1}{2} a T_t^2. \tag{3.9}$$

Der aktuelle Zustand eines Blockes kann aus dem decoderseitig berechnetem Vektorfeld bestimmt werden. Ermittelt die Bewegungsschätzung für benachbarte Blöcke gleiche Bewegungsvektoren, kann davon ausgegangen werden, dass diese Blöcke zu

dem selben Objekt gehören und somit den selben Versatz aufweisen werden. Der Zustand ist demnach S_1. Unterschiedliche Bewegungsvektoren können in diesem Modell nur in Zustand S_2 auftreten. Daher kann die Codierung des Versatzes $d(j)$ an die beiden Zustände angepasst werden. Im Zustand S_1 muss der Versatz nicht signalisiert werden, da dieser bereits vom vorangegangenen Block bekannt ist. Der Versatz im Zustand S_2 kann nicht anhand des vorangegangenen Blockes bestimmt werden und muss somit codiert werden. Einer differentiellen PCM-Codierung, die den Versatz des vorangegangenen Blockes als Prädiktion für den aktuellen Block verwendet und lediglich die Differenz codiert, ist nicht sinnvoll, da $d(j)$ und $d(j-1)$ nicht korreliert sind.

Für die Bestimmung der benötigten Datenrate zur Übertragung des Versatzes $d(j)$ im Zustand S_2 wird zunächst die Varianz σ_d^2 berechnet:

$$\begin{aligned}\sigma_d^2 &= E\left[d^2(j)\right] \\ &= E\left[\frac{1}{4}a^2 T_t^4\right] \\ &= \frac{1}{4}T_t^4 E\left[a^2\right] \\ &= \frac{1}{4}T_t^4\sigma_a^2. \end{aligned} \tag{3.10}$$

Die Entropie einer Komponente des gaußverteilten Versatzes $d(j)$ für den zweiten Zustand ergibt sich nach [22] zu

$$h(d) = \frac{1}{2}\log_2 2\pi e\frac{1}{4}T_t^4\sigma_a^2. \tag{3.11}$$

Jedoch soll der Versatz $d(j)$ nicht mit der vollen Genauigkeit bestimmt und übertragen werden, sondern analog zu den konventionellen Bewegungsvektoren mit der Stufenbreite Δ quantisiert werden. Wie in [15] gezeigt, ergibt sich die Entropie durch die verringerte Vektorauflösung zu

$$H(\vec{d'}) = 2\left(h(d) - \log_2\Delta\right) \tag{3.12}$$

$$\begin{aligned} &= \log_2 2\pi e\frac{1}{4}T_t^4\sigma_a^2 - \log_2\Delta^2 \\ &= \log_2\frac{\frac{1}{2}\pi e T_t^4\sigma_a^2}{\Delta^2}. \end{aligned} \tag{3.13}$$

Der Faktor 2 in Gleichung 3.12 resultiert aus der Codierung der x- und y-Komponente des Versatzes.

Die mittlere Rate zur Codierung des Versatzes kann nun aus der Mittlung über alle Blöcke bestimmt werden. Dabei muss unterschieden werden, in welchem Zustand (S_1 oder S_2) sich der jeweilige Block befindet:

$$R_V = \frac{1}{N}\left(\sum_{S_2}\log_2\frac{\frac{1}{2}\pi e T_t^4\sigma_a^2}{\Delta^2}\right), \tag{3.14}$$

wobei N die Anzahl der Punkte im Bild repräsentiert. Wie bereits beschrieben, ist die Rate zur Codierung des Versatzes im Zustand S_1 Null, da der Versatz aus dem vorangegangenen Block bekannt ist und keine Information übertragen werden muss.

Im Folgenden soll $P_B(S_2)$ die Wahrscheinlichkeit sein, mit der sich ein Block der Größe $B \times B$ in Zustand S_2 befindet. Analog dazu ist $P_B(S_1) = 1 - P_B(S_2)$ die Wahrscheinlichkeit für Zustand S_1. Zur Berechnung der mittleren Rate pro Bildpunkt für die Codierung des Versatzes, wird die Gesamtzahl der Blöcke $\frac{N}{B^2}$ innerhalb des Bildes mit der Auftretenswahrscheinlichkeit $P_B(S_2)$ multipliziert:

$$\begin{aligned} R_V &= \frac{1}{N}\left(P_B(S_2)\frac{N}{B^2}\log_2\frac{\frac{1}{2}\pi e T_t^4\sigma_a^2}{\Delta^2}\right) \\ &= \frac{P_B(S_2)}{B^2}\log_2\frac{\frac{1}{2}\pi e T_t^4\sigma_a^2}{\Delta^2}. \end{aligned} \tag{3.15}$$

Die Wahrscheinlichkeit $P_B(S_2)$ ist abhängig von der gewählten Blockgröße. Da der Aufbau der physikalischen Objekte jedoch unabhängig von der gewählten Blockgröße ist, soll der Zusammenhang von Blockgröße und Wahrscheinlichkeit untersucht werden. Bei dem Beispiel in Abbildung 3.4 mit der Blockgröße B ändert sich der Versatz zwischen zwei Blöcken (Zustand S_2) sechs mal und bleibt zwei mal unverändert (Zustand S_1). Damit ergibt sich die Wahrscheinlichkeit für Zustand S_2 zu

$$P_B(S_2) = \frac{7}{7+2} = \frac{7}{9}. \tag{3.16}$$

Die Versatzfunktion $d(j)$ in Abbildung 3.5 wurde aus dem Bildausschnitt in Abbildung 3.4 durch die zeilenweise Abtastung mit halbierter Blockgröße ermittelt. Durch die Abtastung mit halbierter Blockgröße verdoppelt sich auch die Anzahl der Zeilen. Jedoch ist eine Zeile mit gerader Zeilennummer identisch zur vorangegangenen Zeile, wodurch sich die relativen Häufigkeiten von Zustand S_1 und Zustand S_2 nicht ändern.

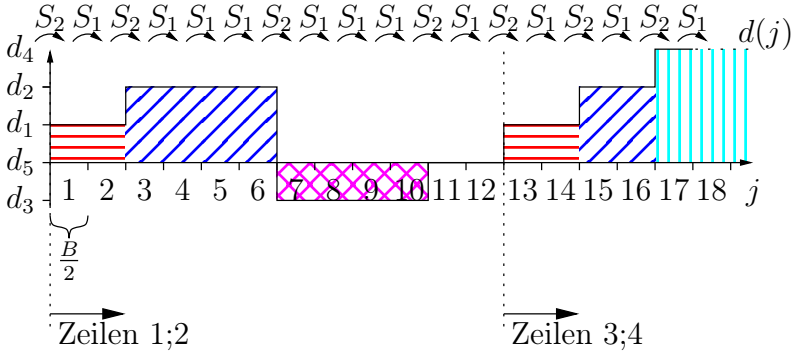


Abbildung 3.5: Verlauf des Versatzes durch nichtlineare Bewegung aus Abbildung 3.4 bei einer Blockgröße von $\frac{B}{2}$.

Es ist zu erkennen, dass die Anzahl der Blöcke im Zustand S_2 konstant bleibt. Jedoch verdoppelt sich die Gesamtzahl der Blöcke und die Wahrscheinlichkeit $P_B(S_2)$ ändert sich entsprechend zu

$$P_{\frac{B}{2}}(S_2) = \frac{7}{2 \cdot (7+2)} = \frac{7}{18} = \frac{P_B(S_2)}{2}. \tag{3.17}$$

Durch Verallgemeinerung von Gleichung 3.17 wird die Wahrscheinlichkeit

$$P_B(S_2) = P_{B_0}(S_2) \cdot \frac{B}{B_0}, \tag{3.18}$$

wobei B_0 ein ganzzahliges Vielfaches von B sein muss. Damit immer noch gewährleistet ist, dass ein Block nur ein Objekt enthält, darf die Blockgröße B_0 nicht zu groß gewählt werden.

Wird nun Gleichung 3.18 in 3.15 eingesetzt, erhält man die mittlere Rate zur Codierung des Versatzes pro Bildpunkt in Abhängigkeit der Blockgröße:

$$R_V = \frac{P_{B_0}}{B_0} \frac{1}{B} \log_2 \frac{\frac{1}{2}\pi e T_t^4 \sigma_a^2}{\Delta^2}. \tag{3.19}$$

Zur besseren Übersicht wurde $P_{B_0}(S_2)$ durch P_{B_0} ersetzt.

Demnach ist die Rate zur Codierung antiproportional zur Blockgröße B. Außerdem sinkt die Datenrate bei steigender Abtastfrequenz und der daraus folgenden kürzeren Abtastzeit $T_t = \frac{1}{f_t}$, da in diesem Fall die Auswirkungen der beschleunigten Bewegung abnehmen. $\frac{P_{B_0}}{B_0}$ ist ein Maß für die Anzahl der unabhängigen Objekte im Bild und sorgt dafür, dass die Rate bei geringer Objektzahl ebenfalls sinkt.

3.1.3 Modellierung der Prädiktionsfehlerdatenrate

Nachdem die benötigte Rate für den Versatz bestimmt wurde, soll im Folgenden die Datenrate, die zur Codierung des Prädiktionsfehlers aufgewendet werden muss, durch eine mathematische Beschreibung modelliert werden. Um eine verlustbehaftete Codierung zu ermöglichen, wird der Prädiktionsfehler mit der Stufenbreite Q gleichförmig quantisiert. In der Videocodierung wird der Prädiktionsfehler mit Hilfe einer DCT transformiert, um unter anderem örtliche Korrelationen auszunutzen. Bei kleinen Blöcken ist eine genau Prädiktion möglich, wodurch der Prädiktionsfehler weitestgehend unkorreliertem Rauschen entspricht. Daher hat eine Transformation lediglich geringen Einfluss auf die Codierung. Zur einfacheren Berechnung wird daher angenommen, dass das Residuum ohne Ausnutzung von örtlichen Korrelationen codiert wird. Somit entspricht die zu erwartende Datenrate R_R in etwa der Entropie $H(e')$ des Residuums:

$$R_R \approx H(e'). \tag{3.20}$$

In [15] wurde gezeigt, dass sich die diskrete Entropie des Residuums aus der differentiellen Entropie $h(e)$ des Prädiktionsfehlers durch

$$H(e') = h(e) - \log_2 Q \tag{3.21}$$

bestimmen lässt. Mit der Annahme, dass die Fehler der einzelnen Bildpunkte, wie in [48] beschrieben, laplaceverteilt sind, ergibt sich die Rate zur Codierung des Residuums zu

$$R_R \approx \log_2\left(\sqrt{2}e\sigma_e\right) - \log_2 Q. \tag{3.22}$$

Jedoch gilt Gleichung 3.21 nur für sehr feine Quantisierungen. Daher wird in [56] die Entropie bei sehr großen Quantisierungstufen Q, im Vergleich zur Varianz σ_e^2, durch eine lineare Funktion $f(\sigma_e^2) = m\sigma_e^2$ approximiert. Dabei wird die Steigung m so gewählt, dass die Funktion die Gleichung 3.22 mit gleicher Steigung schneidet. Wie in Abbildung 3.6 zu sehen, gilt dies für $\sigma_e^2 = \frac{Q^2}{2e}$ und $m = \frac{e}{Q^2 \ln 2}$. Durch diese

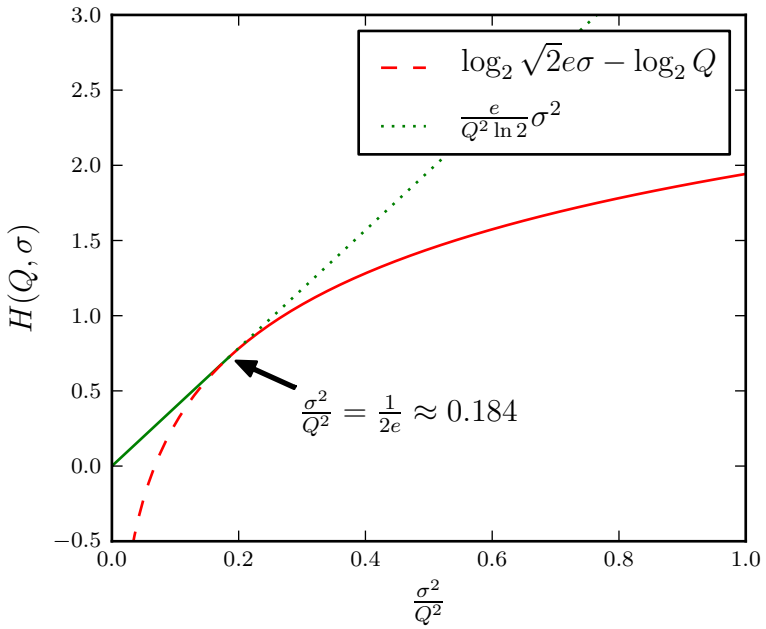


Abbildung 3.6: Approximierter Verlauf der Entropie einer laplaceschen Wahrscheinlichkeitsverteilung nach [56].

Aufteilung der Entropie in zwei Funktionen ergibt sich die Residuumrate zu

$$R_R \approx \begin{cases} \log_2 \sqrt{2}e\sigma_e - \log_2 Q, & \sigma_e^2 > \frac{Q^2}{2e} \\ \frac{e}{Q^2 \ln 2}\sigma_e^2, & \sigma_e^2 \leq \frac{Q^2}{2e}. \end{cases} \tag{3.23}$$

Bei der konventionellen unidirektionalen Prädiktion, wie sie auch in [56] zu Grunde gelegt wird, kann die Prädiktionsfehlervarianz σ_e^2 aus der Differenz des Originalbildes s und der aus dem Referenzbild s'_{-1} gewonnenen, bewegungskompensierten Prädiktion $u = \hat{s}(s'_{-1})$ des aktuellen Bildes berechnet werden:

$$\sigma_e^2 = E\left[(s-u)^2\right] \tag{3.24}$$

$$= E\left[(e_u)^2\right]. \tag{3.25}$$

In [57] wurde gezeigt, dass sich der Prädiktionsfehler e_u auf vier voneinander unabhängige Ursachen zurückführen lässt:

$$\begin{aligned} e_u = \; & e_\Delta^{(u)} && \text{begrenzte Genauigkeit der Bewegungsvektoren} \\ & +e_V^{(u)} && \text{unterschiedliche Bewegung innerhalb eines Blockes} \\ & +e_Q^{(u)} && \text{Verzerrung des Referenzbildes durch Quantisierung} \\ & +e_n^{(u)}. && \text{Verdeckung, Kamerarauschen, Beleuchtungsänderung} \end{aligned} \tag{3.26}$$

Werden diese Werte in Gleichung 3.25 eingefügt, erhält man, unter Berücksichtigung der statistischen Unabhängigkeit der Werte untereinander, die Prädiktionsfehlervarianz

$$\begin{aligned} \sigma_e^2 &= E\left[(e_u)^2\right] \\ &= E\left[(e_\Delta + e_V + e_Q + e_n)^2\right] \\ &= E\left[e_\Delta^2\right] + E\left[e_V^2\right] + E\left[e_Q^2\right] + E\left[e_n^2\right] \\ &\quad + \underbrace{E\left[2e_\Delta e_V\right]}_{0} + \underbrace{E\left[2e_\Delta e_Q\right]}_{0} + \ldots \\ &= \sigma_\Delta^2 + \sigma_V^2 + \sigma_Q^2 + \sigma_n^2. \end{aligned} \tag{3.27}$$

Die einzelnen Komponenten lassen sich, wie in [57] hergeleitet, folgendermaßen berechnen:

$$\sigma_\Delta^2 = \Delta^2 G, \tag{3.28}$$

$$\sigma_V^2 = 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right) GB, \tag{3.29}$$

$$\sigma_Q^2 = \frac{Q^2}{12}, \tag{3.30}$$

$$\sigma_n^2 = \mu. \tag{3.31}$$

Der Parameter c_A ist der Korrelationskoeffizient eines autoregressiven Zufallprozesses erster Ordnung, welcher zur Beschreibung des wahren Bewegungsvektorfeldes

genutzt wird. G stellt ein Maß für die Textur des Bildes dar und wird durch Mittelung des Quadrats der Gradienten aller N Bildpunkte berechnet:

$$G = \frac{1}{N} \sum_{n_x,n_y} \left((s(n_x,n_y) - s(n_x+1,n_y))^2 + (s(n_x,n_y) - s(n_x,n_y+1))^2 \right). \quad (3.32)$$

Im Gegensatz zur unidirektionalen Prädiktion entsteht die Prädiktion $\hat{s}$ des aktuell zu codierenden Bildes bei der Verwendung von DSME, wie in Abbildung 3.7 gezeigt, aus der bilinearen Interpolation zweier Referenzbilder s'_{-1} und s'_{+1}. Die Prädiktion

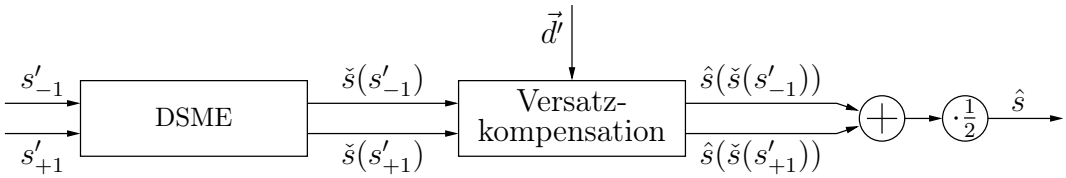


Abbildung 3.7: Schematische Darstellung der bewegungskompensierten und versatzkompensierten Prädiktionen am Decoder.

wird in diesem Fall durch Mittlung der beiden am Decoder bewegungskompensierten und versatzkompensierten Prädiktionen $u = \hat{s}(\check{s}(s'_{-1}))$ und $v = \hat{s}(\check{s}(s'_{+1}))$ erzeugt. Somit muss die Berechnung in Gleichung 3.24 erweitert werden:

$$\begin{aligned}
\sigma_e^2 &= E\left[\left(s - \frac{u+v}{2}\right)^2\right] \\
&= E\left[\left(\frac{s-u}{2} + \frac{s-v}{2}\right)^2\right] \\
&= \frac{1}{4} E\left[(e_u + e_v)^2\right] \\
&= \frac{1}{4}\left(E\left[e_u^2\right] + E\left[e_v^2\right] + 2E\left[e_u e_v\right]\right). \qquad (3.33)
\end{aligned}$$

Aufgrund der Stationarität gilt, dass die Erwartungswerte von e_u^2 und e_v^2 identisch sind. Die Kovarianz $E[e_u e_v]$ ergibt unter Berücksichtigung der statistischen Unabhängigkeit der einzelnen Parameter

$$\begin{aligned}
E[e_u e_v] &= E\left[\left(e_\Delta^{(u)} + e_V^{(u)} + e_Q^{(u)} + e_n^{(u)}\right)\left(e_\Delta^{(v)} + e_V^{(v)} + e_Q^{(v)} + e_n^{(v)}\right)\right] \\
&= E\left[e_\Delta^{(u)} e_\Delta^{(v)}\right] + E\left[e_V^{(u)} e_V^{(v)}\right] + E\left[e_Q^{(u)} e_Q^{(v)} + E\left[e_n^{(u)} e_n^{(v)}\right]\right] \\
&\quad + \underbrace{E\left[2e_\Delta^{(u)} e_V^{(v)}\right]}_{0} + \underbrace{E\left[2e_\Delta^{(u)} e_Q^{(v)}\right]}_{0} + \ldots \\
&= c_\Delta \sigma_\Delta^2 + c_V \sigma_V^2 + c_Q \sigma_Q^2 + c_n \sigma_n^2, \qquad (3.34)
\end{aligned}$$

wobei c_Δ, c_V, c_Q und c_n die Korrelationskoeffizienten der einzelnen Fehlerkomponenten zwischen den beiden Referenzbildern sind. Somit vereinfacht sich Gleichung 3.33 mit Hilfe von 3.27 und 3.34 zu

$$\begin{aligned}\sigma_e^2 = &\frac{1}{2}\left(\sigma_\Delta^2 + \sigma_V^2 + \sigma_Q^2 + \sigma_n^2\right)\\ &+\frac{1}{2}\left(c_\Delta\sigma_\Delta^2 + c_V\sigma_V^2 + c_Q\sigma_Q^2 + c_n\sigma_n^2\right).\end{aligned} \tag{3.35}$$

Sind die Referenzbilder s'_{-1} und s'_{+1} identisch, ergeben die Korrelationskoeffizienten $c_\Delta = c_V = c_Q = c_n = 1$ und man erhält die Varianz für den Fall unidirektionaler Prädiktion wie in Gleichung 3.27.

Im Fall von DSME lassen sich die Korrelationskoeffizienten nur mit sehr hohem Aufwand bestimmen. Jedoch kann mit Gleichung 3.35 gezeigt werden, dass die Varianz des Prädiktionsfehlers für die decoderseitige Schätzung immer kleiner oder gleich der Varianz des Prädiktionsfehlers der konventionellen Codierung ist, solange der Fehler durch nichtlineare Bewegung kompensiert wird. Daher wird zum Vergleich der Datenrate der konventionellen Prädiktion und DSME angenommen, dass die Varianz des Prädiktionsfehlers für beide Methoden identisch ist.

Somit ist die Datenrate zur Codierung des Residuums in Gleichung 3.23 unter Berücksichtigung der Gleichungen 3.28 bis 3.31 und Gleichung 3.35 bei feiner ($\sigma_e^2 > \frac{Q^2}{2e}$) und grober Quantisierung

$$R_R \approx \begin{cases} \check{R}_R = \log_2 \frac{\sqrt{2}e\sqrt{\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right)GB + \frac{Q^2}{12} + \mu}}{Q} & \text{(fein)} \\ \hat{R}_R = \frac{e}{Q^2 \ln 2}\left(\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right)GB + \frac{Q^2}{12} + \mu\right). & \text{(grob)} \end{cases} \tag{3.36}$$

Die kombinierte Datenrate zur Codierung eines Bildpunktes mit Hilfe decoderseitiger Bewegungsschätzung aus Gleichung 3.1 ergibt mit Gleichung 3.19 und Gleichung 3.36

$$\begin{aligned} R^{(\mathrm{DSME})} \approx &\frac{P_{B_0}}{B_0}\frac{1}{B}\log_2\frac{\frac{1}{2}\pi e T_t^4 \sigma_a^2}{\Delta^2}\\ &+\begin{cases} \log_2 \frac{\sqrt{2}e\sqrt{\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right)GB + \frac{Q^2}{12} + \mu}}{Q} & \text{(fein)} \\ \frac{e}{Q^2 \ln 2}\left(\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right)GB + \frac{Q^2}{12} + \mu\right). & \text{(grob)} \end{cases} \end{aligned} \tag{3.37}$$

Diese Rate wird in Abschnitt 3.2 mit der Datenrate der konventionellen Codierung mit unidirektionaler Prädiktion aus [57]

$$\begin{aligned} R^{(\mathrm{KONV})} \approx &\frac{1}{B^2}\log_2\frac{4e^2\sigma_V^2 \ln\left(\frac{1}{\tilde{c}_A}\right)B}{\Delta^2}\\ &+\begin{cases} \log_2 \frac{\sqrt{2}e\sqrt{\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right)GB + \frac{Q^2}{12} + \mu}}{Q} & \text{(fein)} \\ \frac{e}{Q^2 \ln 2}\left(\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right)GB + \frac{Q^2}{12} + \mu\right). & \text{(grob)} \end{cases} \end{aligned} \tag{3.38}$$

verglichen. Der neue Parameter $\tilde{c}_A$ ist der Korrelationskoeffizient des autoregressiven Prozesses zur Modellierung des gemessenen Vektorfeldes und kann sich stark vom Korrelationskoeffizienten c_A des wahren Bewegungsvektorfeldes unterscheiden.

Ein Vergleich mit bidirektionaler Prädiktion, wie sie aus B-Bildern bekannt ist, wird nicht durchgeführt. Da für jeden Block zwei Bewegungsvektoren benötigt werden, verdoppelt sich auch die Rate zur Codierung der Vektoren. Die Reduzierung der Rate zur Codierung des Residuums durch die verbesserte Prädiktion kann in dem vorgestellten Modell durch die vereinfachende Annahme $c_\Delta = c_Q = c_V = c_n = 1$ jedoch nicht berücksichtigt werden. Somit ist die modellierte Gesamtdatenrate der bidirektionalen Prädiktion immer höher als bei unidirektionaler Prädiktion und ein Vergleich mit DSME daher überflüssig.

3.1.4 Berechnung der optimalen Blockgröße

In den vorangegangenen Abschnitten wurde ein Modell für die Gesamtdatenrate bei Verwendung von DSME erstellt. Um diese Rate mit der aus [57] bekannten Gesamtdatenrate bei konventioneller, unidirektionaler Prädiktion (Gleichung 3.38) vergleichen zu können, muss die optimale Blockgröße ermittelt werden. Dazu wird Gleichung 3.37 nach der Blockgröße B abgeleitet und gleich Null gesetzt:

$$\frac{\partial \check{R}^{(\mathrm{DSME})}}{\partial B} = \frac{\partial R_V^{(\mathrm{DSME})}}{\partial B} + \frac{\partial \check{R}_R^{(\mathrm{DSME})}}{\partial B} \stackrel{!}{=} 0, \tag{3.39}$$

$$\frac{\partial \hat{R}^{(\mathrm{DSME})}}{\partial B} = \frac{\partial R_V^{(\mathrm{DSME})}}{\partial B} + \frac{\partial \hat{R}_R^{(\mathrm{DSME})}}{\partial B} \stackrel{!}{=} 0. \tag{3.40}$$

Aus Gleichung 3.39 folgt, dass

$$\check{B}_{\mathrm{opt}}^{(\mathrm{DSME})} = k_3 + \sqrt{(k_3)^2 + 2\frac{k_2 k_3}{k_1}} \tag{3.41}$$

mit

$$k_1 = 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right) G, \tag{3.42}$$

$$k_2 = \Delta^2 G + \frac{Q^2}{12} + \mu, \tag{3.43}$$

$$k_3 = \frac{P_{B_0}}{B_0} \ln \frac{\frac{1}{2}\pi e T_t^4 \sigma_a^2}{\Delta^2} \tag{3.44}$$

die optimale Blockgröße mit der geringsten Gesamtdatenrate bei feiner Quantisierung ist. Entsprechend ist die optimale Blockgröße bei grober Quantisierung

$$\hat{B}_{\mathrm{opt}}^{(\mathrm{DSME})} = \sqrt{\frac{Q^2}{e}\frac{k_3}{k_1}}. \tag{3.45}$$

In [57] wurde gezeigt, dass die optimale Blockgröße bei konventioneller Codierung und grober Quantisierung durch die Gleichung

$$0 = 1 - 2k_4 - 2\ln \hat{B}_{\text{opt}}^{(\text{KONV})} + \frac{e}{Q^2} k_1 \hat{B}_{\text{opt}}^{(\text{KONV})\,3} \tag{3.46}$$

mit

$$k_4 = \ln \frac{4e^2\sigma_V^2 \ln\left(\frac{1}{\tilde{c}_A}\right)}{\Delta^2} \tag{3.47}$$

beschrieben wird. Zur Berechnung der optimale Blockgröße bei feiner Quantisierung muss folgende Gleichung gelöst werden:

$$0 = 2\left(k_1 \check{B}_{\text{opt}}^{(\text{KONV})} + k_2\right)\left(1 - 2k_4 - 2\ln \check{B}_{\text{opt}}^{(\text{KONV})}\right) + k_1 \check{B}_{\text{opt}}^{(\text{KONV})\,3}. \tag{3.48}$$

Die detaillierte Herleitung der optimalen Blockgrößen ist in Abschnitt A.1 gezeigt.

3.2 Bewertung des Modells anhand experimenteller Untersuchungen

Das in Abschnitt 3.1 hergeleitete Modell soll durch experimentelle Versuche verifiziert werden. An den Coder sind spezielle Anforderungen zu stellen, um mit dem Modell vergleichbar zu sein. Der Aufbau des Versuchscoders wird im folgenden Abschnitt erläutert. Anschließend wird das Modell anhand der experimentellen Ergebnisse bewertet.

3.2.1 Versuchsaufbau des Coders

Zur Modellierung der Datenrate, welche zur Codierung des Prädiktionsfehlers und des Versatzes durch nichtlineare Bewegung aufgebracht werden muss, wurden Einschränkungen gemacht. So wird die Blockgröße zur Bewegungskompensation als konstant angenommen, anstatt sie, wie in aktuellen Codern, adaptiv zu wählen. Auch eine Transformation zur Ausnutzung von örtlichen Korrelationen innerhalb des Prädiktionsfehlers wird nicht berücksichtigt. Um dennoch eine Einschätzung des Modells zu ermöglichen, wurde ein vereinfachter Encoder entworfen. Wie in Abbildung 3.8 zu sehen, ist dieser Versuchsaufbau dem vereinfachten Blockdiagramm in Abbildung 2.2 sehr ähnlich. Jedoch wird der Prädiktionsfehler nicht transformiert, sondern direkt im Zeitbereich quantisiert. Außerdem ist die Blockgröße in der Bewegungsschätzung nicht mehr adaptiv veränderbar. Mit dem dargestellten Schalter S lässt sich zwischen der konventionellen Codierung in der unteren Position und der Codierung unter Zuhilfenahme decoderseitiger Bewegungsschätzung in der oberen

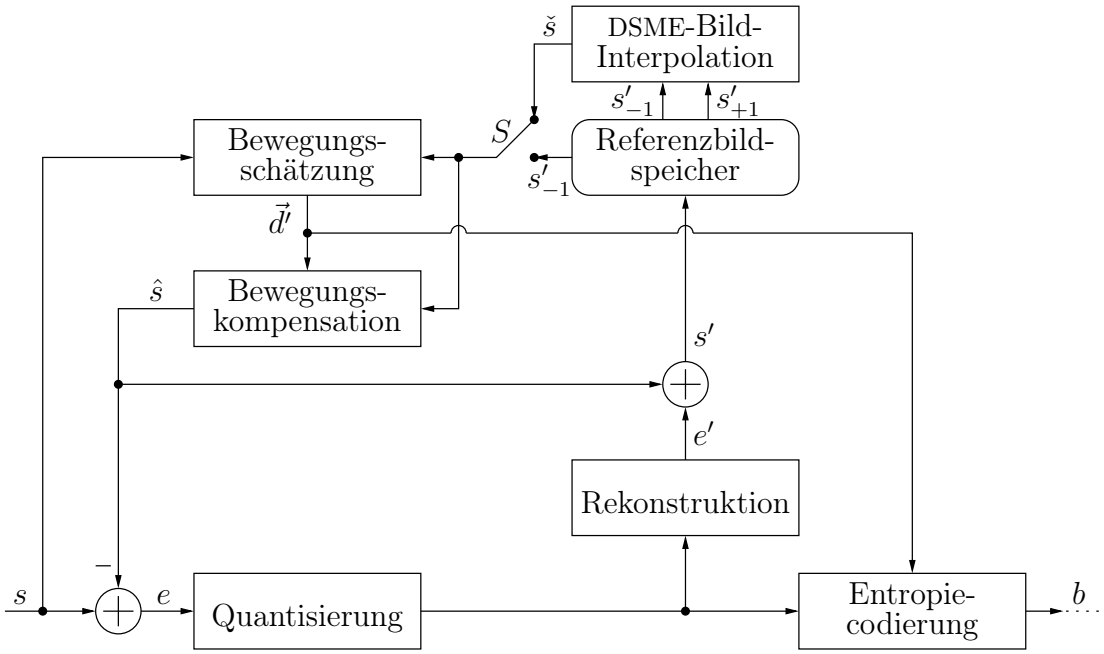


Abbildung 3.8: Versuchsaufbau des Encoders. Ist der Schalter S in der unteren Position, wird das Bild konventionell codiert, in der oberen Position wird decoderseitige Bewegungsschätzung verwendet.

Position wählen. Der gewählte Modus wird während der gesamten Sequenz nicht geändert.

Die DSME-Bild-Interpolation erhält aus dem Referenzbildspeicher ein zeitlich vorangegangenes Referenzbild und ein Referenzbild aus der Zukunft und erzeugt daraus eine Prädiktion des aktuell zu codierenden Bildes. Eine detaillierte Beschreibung der verwendeten DSME-Algorithmen ist in Abschnitt 4.2 zu finden.

3.2.2 Auswertung

Anhand der PeopleOnStreet-Sequenz wird das vorgestellte Modell detailliert bewertet. Eine kurze Beschreibung sowie das erste Bild dieser Sequenz findet sich in Abschnitt 5.1.1. Am Ende dieses Abschnitts sind die Ergebnisse weiterer Sequenzen gezeigt.

Zur Bewertung des Modells wird die Testsequenz mit Hilfe des Versuchscoders aus Abbildung 3.8 codiert. Dabei wird die Quantisierungsstufenbreite auf $Q = 10$ gesetzt, um eine Qualität von etwa 40 dB zu erreichen. Die verwendete Bewegungsschätzung arbeitet mit einer Genauigkeit von $\Delta = 0{,}25$ pel. Dabei wird die Sequenz sieben mal mit unterschiedlichen Blockgrößen für die bewegungskompensierende Prädiktion (1, 2, 4, 8, 16, 32, 64) codiert.

Die zur Modellierung der Residuum- und Bewegungsvektordatenrate eingeführten, sequenzabhängigen Parameter G, σ_V, $\tilde{c}_A$, σ_a und $\frac{P_{B_0}}{B_0}$ werden anhand der Testsequenz PeopleOnStreet gemessen. Der Texturparameter G kann leicht mit Gleichung 3.32 aus der Originalsequenz bestimmt werden. Mit Hilfe der, durch konventioneller Bewegungsschätzung, ermittelten Vektoren wird die Standardabweichung des Bewegungsvektorfeldes σ_V approximiert. $\tilde{c}_A$ kann, wie in [57] beschrieben, aus σ_V und der Varianz der Differenzvektoren benachbarter Bewegungsvektoren berechnet werden. Die Standardabweichung der beschleunigten Bewegung σ_a kann mit Hilfe von Gleichung 3.10 bestimmt werden. Dazu muss jedoch das mit Hilfe von decoderseitiger Bewegungsschätzung ermittelte Vektorfeld mit dem Vektorfeld der konventionellen Bewegungsschätzung verglichen werden, um den Versatz zu bestimmen. Die normierte Wahrscheinlichkeit $\frac{P_{B_0}}{B_0}$ wird bei einer Blockgröße von 16 pel ermittelt. Es wird demnach davon ausgegangen, dass sich jedes Objekt im Bild aus mehreren 16×16 Blöcken zusammensetzen lässt.

Zur Berechnung des Korrelationskoeffizienten c_A müsste die wahre Bewegung für jeden Bildpunkt bekannt sein. Jedoch konnte in [18] gezeigt werden, dass c_A für diverse Sequenzen relativ konstant ist. Daher wird im Folgenden der Korrelationskoeffizient, wie in [57] beschrieben, auf $c_A = 0{,}998$ festgelegt. Der Parameter μ repräsentiert verschiedene Faktoren wie Verdeckung, Kamerarauschen und Beleuchtungsänderungen zwischen den einzelnen Bildern. Somit ist es nicht möglich, diese Konstante durch einfache Messungen zu bestimmen. Aus diesem Grund wurde μ so bestimmt, dass für die sieben Blockgrößen der quadratische Fehler zwischen den gemessenen Raten des Residuums und den Werten des Modells minimal wird. Die ermittelten Werte aller Parameter sind in Tabelle 3.1 zusammengefasst.

Auf den ersten Blick erscheint die Standardabweichung der Beschleunigung a sehr hoch. Laut [72] gilt, dass 99,7 % aller Objekte eine Beschleunigung im Bereich von $\pm 3\sigma_a = \pm 8274$ haben. Rechnet man dieses Intervall mit Hilfe von Gleichung 3.6 in den Versatz um, erhält man

$$\begin{aligned} d_{99,7\%} &= \frac{1}{2} a_{99,7\%} T_t^2 \\ &= \pm \frac{1}{2} 3\sigma_a T_t^2 \\ &= \pm 4{,}51 \,\text{pel}. \end{aligned} \tag{3.49}$$

Somit besitzen 99,7 % der Objekte im Bild einen Versatz kleiner oder gleich 4,51 pel. Im Vergleich dazu liegen 99,7 % der Bewegungsvektoren bei der konventionellen Codierung im Intervall $d_{99,7\%} = \pm 3\sigma_V = \pm 10{,}84$ pel.

Die normierte Auftretenswahrscheinlichkeit sagt aus, dass zwei benachbarte Bildpunkte mit einer Wahrscheinlichkeit von $1 - 0{,}056 = 0{,}944$ zum selben physikalischen Objekt gehören.

Der durch Verdeckung, Kamerarauschen und Beleuchtungsänderungen verursachte Prädiktionsfehler e_n hat einen relativ großen Einfluss auf die Gesamtfehlervarianz

Tabelle 3.1: Anhand der Sequenz PeopleOnStreet ermittelte Parameter zur Modellierung der Datenraten für konventionelle und DSME Codierung. Lediglich μ wurde zur Anpassung an die realen Daten optimiert.

Parameter	**Wert**	**Bedeutung**
G	19,17	Stärke der Texturierung des Bildes
σ_V	3,614	Standardabweichung des Bewegungsvektorfeldes
$\tilde{c}_A$	0,932	Korrelationskoeffizient des AR-Prozesses zur Modellierung des geschätzten Vektorfeldes
σ_a	2758	Standardabweichung der Beschleunigung
$\frac{P_{B_0}}{B_0}$	0,056	Normierte Auftretenswahrscheinlichkeit von unterschiedlichen Objekten
μ	13,45	Durch Verdeckung, Kamerarauschen und Beleuchtungsänderungen verursachte Prädiktionsfehlervarianz
T_t	0,033	Abtastzeit ($T_t = \frac{1}{f_t}$)

aus Gleichung 3.27. Bei einer Blockgröße von 4 pel ist $\sigma_n^2 = \mu = 13{,}45$ nach der Fehlervarianz $\sigma_V^2 = 24{,}06$, verursacht durch die rein translatorische und blockbasierte Bewegungskompensation, der zweitgrößte Summand. Die Quantisierung mit der Stufenbreite $Q = 10$ verursacht eine Fehlervarianz von $\sigma_Q^2 = 8{,}33$. Der Einfluss durch die begrenzte Genauigkeit der Bewegungsvektoren von 0,25 pel ist mit $\sigma_\Delta^2 = 1{,}20$ vergleichsweise sehr gering.

In Abbildung 3.9 sind die analytisch berechneten Datenraten zur Codierung der Bewegungsvektoren für die konventionelle Codierung (KONV) und für die Codierung mit Hilfe von decoderseitiger Bewegungsschätzung (DSME) in Abhängigkeit der Blockgröße gezeigt. Es ist deutlich zu erkennen, dass die Rate für DSME geringer ist, da lediglich der durch beschleunigte Bewegung hervorgerufene Versatz kompensiert werden muss. Zusätzlich zu den berechneten Kurven sind die Messwerte der Versuchscoder in dem Diagramm eingetragen. Diese Werte werden sehr gut durch die beiden Modelle approximiert.

Analog zur Rate der Bewegungsvektoren ist in Abbildung 3.10 die Datenrate des Residuums dargestellt. Wie zuvor beschrieben, wird für beide Methoden dasselbe Modell zur Berechnung der Rate R_R genutzt. Für kleine Blöcke werden die realen Datenraten gut von der theoretischen Kurve approximiert. Für größere Blöcke wird das Modell jedoch ungenauer. Eine Ursache für diese Ungenauigkeit ist die Approximation des Bildsignals in [57] durch eine Taylorreihe erster Ordnung. Dadurch steigt die modellierte Rate für die Codierung des Residuums aus Gleichung 3.36 stetig mit

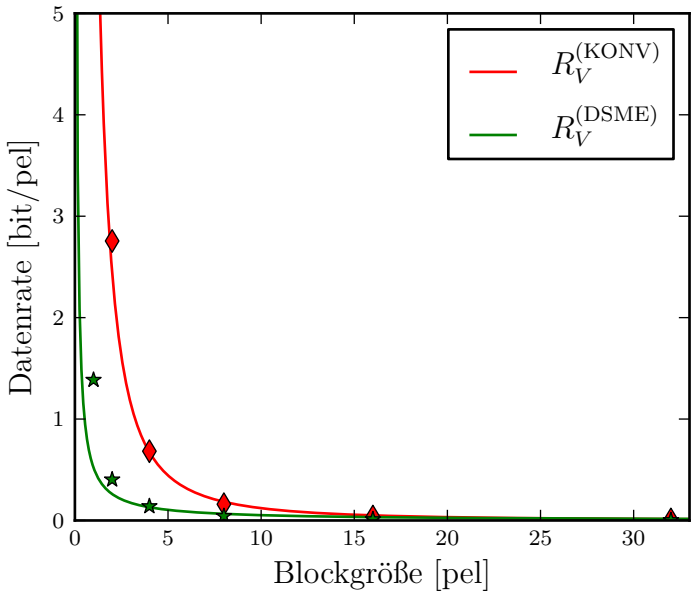


Abbildung 3.9: Rate zur Codierung der Bewegungsvektoren. Die Kurven sind mit Hilfe der Modelle berechnet worden, während die einzelnen Werte mit Hilfe des Versuchscoders gemessen worden sind.

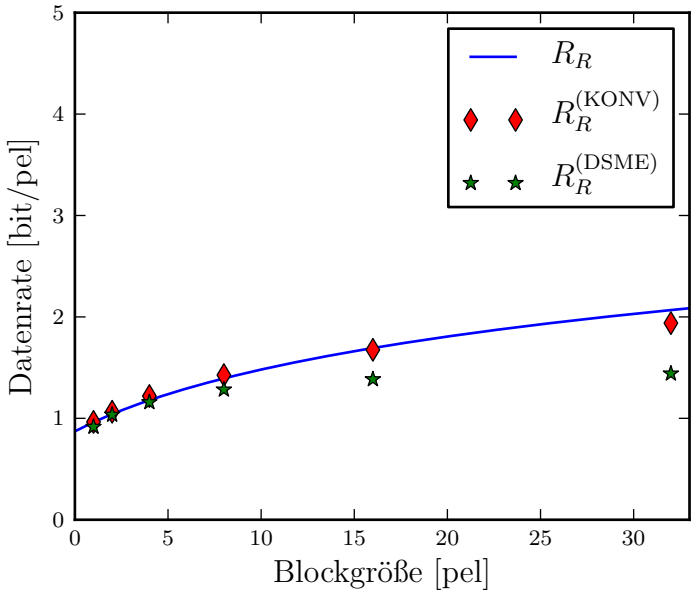


Abbildung 3.10: Rate zur Codierung des quantisierten Prädiktionsfehlers. Die Kurve ist mit Hilfe des Modells berechnet worden, während die einzelnen Werte mit Hilfe der Versuchscoder gemessen worden sind.

der Blockgröße an. Hingegen verschlechtert eine Erhöhung der Blockgröße die Prädiktionsgenauigkeit nicht weiter, sobald die Bewegungsschätzung die Bewegung bei großen Blöcken nicht mehr ermitteln kann. Dieser Fehler ist für die Analyse jedoch nicht von großer Bedeutung, da die interessanten Bereiche bei kleinen Blöcken liegen.

Abbildung 3.10 verifiziert außerdem die Beobachtung aus Gleichung 3.35, dass der Prädiktionsfehler für die decoderseitige Schätzung immer kleiner oder gleich des Prädiktionsfehlers der konventionellen Codierung ist, solange beschleunigte Bewegungen kompensiert werden. Für alle Blockgrößen liegen die DSME-Messwerte unterhalb der Werte der konventionellen Codierung.

Die Gesamtdatenrate, bei der Residual- und Bewegungsvektordatenrate addiert wurden, ist in Abbildung 3.11 gezeigt. Die höhere Ungenauigkeit bei großen Blöcken

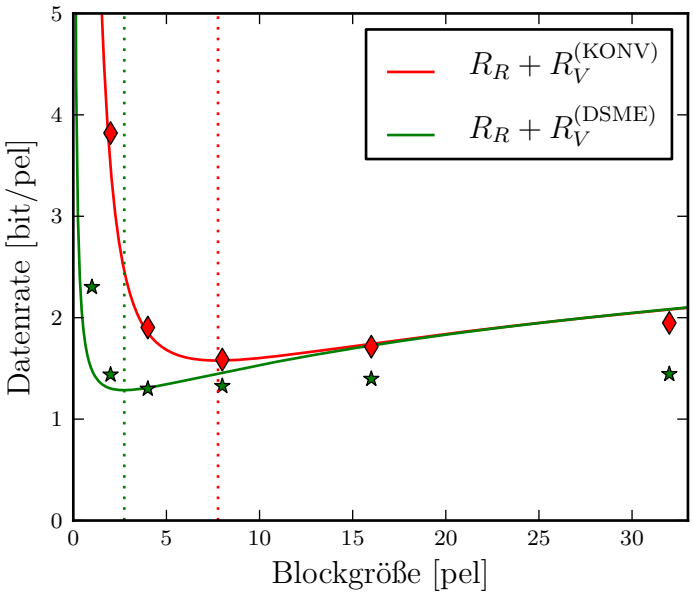


Abbildung 3.11: Gesamtdatenrate zur Codierung des quantisierten Prädiktionsfehlers und der Bewegungsvektoren. Die Kurven sind mit Hilfe der Modelle berechnet worden, während die einzelnen Werte mit Hilfe des Versuchscoders gemessen worden sind. Das Minimum der jeweiligen Kurve liegt bei einer Blockgröße von 7,77 pel beziehungsweise 2,74 pel.

– verglichen mit den gemessenen Werten – ist durch die zuvor beschriebene Einschränkung des Modells für die Residualdatenrate zu erklären. Es ist zu erkennen, dass die optimale Blockgröße für beide Techniken unterschiedlich ist. Der maximale Gewinn bei der Reduktion der Datenrate durch DSME lässt sich anhand der beiden Minima ermitteln. Bei der theoretisch optimalen Blockgröße von 7,77 pel benötigt die konventionelle Bewegungskompensation 1,58 bit pro Bildpunkt. DSME hingegen

erzielt eine Datenrate von 1,29 bit pro Bildpunkt bei einer Blockgröße von 2,74 pel. Die mittlere Datenrate kann somit um gut 18 % verringert werden.

Jedoch gilt dieser Wert nur bei Codierung mit fester Blockgröße. Durch eine adaptive Codierung, bei der zwischen DSME- und konventioneller Codierung gewechselt wird, kann die Datenrate weiter reduziert werden. In Abbildung 3.12 ist, neben den mittleren Raten aus Abbildung 3.11, das Intervall der Datenrate eingezeichnet, die bei 66 % aller Blöcke benötigt wird [72]. Es ist zu erkennen, dass die Daten-

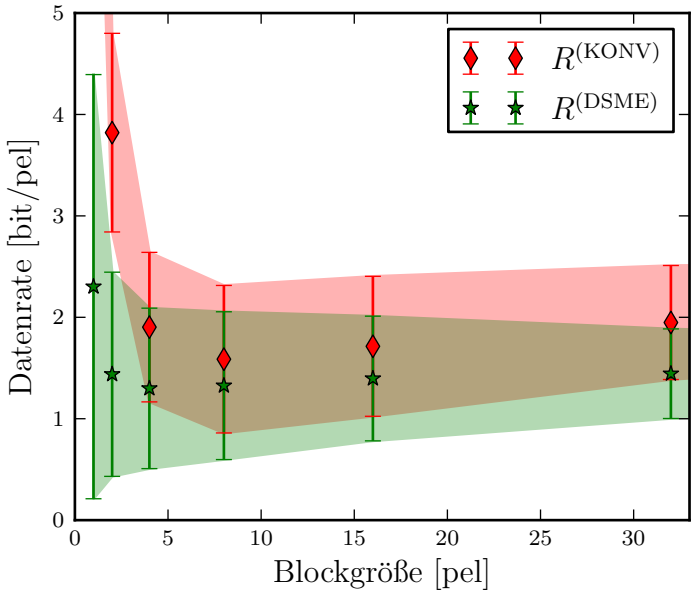


Abbildung 3.12: Gemessene Gesamtdatenrate bei Verwendung der konventionellen Bewegungsschätzung (KONV) und DSME. Der Bereich der Datenrate, in den 66 % der codierten Blöcke fallen, ist farblich hinterlegt.

rate für einzelne Blöcke stark um die mittlere Rate schwankt. Durch den großen Überlappungsbereich der Intervalle wird deutlich, dass die Datenrate eines konventionell codierten Blockes geringer sein kann als die entsprechende DSME-Rate. Eine ähnliche Beobachtung lässt sich beim Vergleich der Intervalle für unterschiedliche Blockgrößen machen. Der große Überlappungsbereich zeigt deutlich, dass für jeden Block individuell entschieden werden sollte, welcher Modus und auch welche Blockgröße zu nutzen sind. Daher wird in Kapitel 4 eine Architektur vorgestellt, in der die Blockgröße und auch der Codiermodus adaptiv gewählt werden.

Die Abbildungen 3.13 und 3.14 zeigen, dass das hergeleitete Modell auch die gemessenen Ergebnisse anderer Testsequenzen aus Abschnitt 5.1 sehr gut approximiert. Jedoch ist bei der BasketballDrive-Sequenz das zuvor beschriebene Problem mit der ungenaueren Modellierung der Residuumsrate aufgrund sehr hoher Varianz σ_V^2 der Bewegungsvektoren viel deutlicher ausgeprägt.

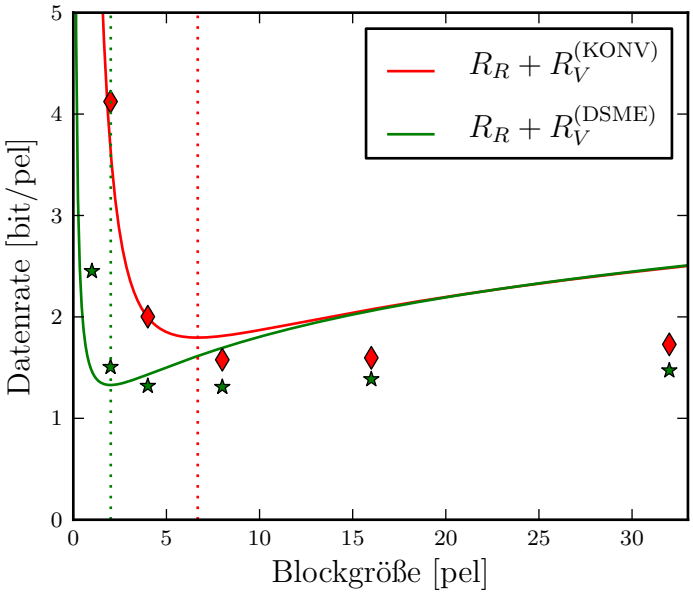


Abbildung 3.13: Gesamtdatenrate zur Codierung des quantisierten Prädiktionsfehlers und der Bewegungsvektoren für die BasketballDrive-Sequenz.

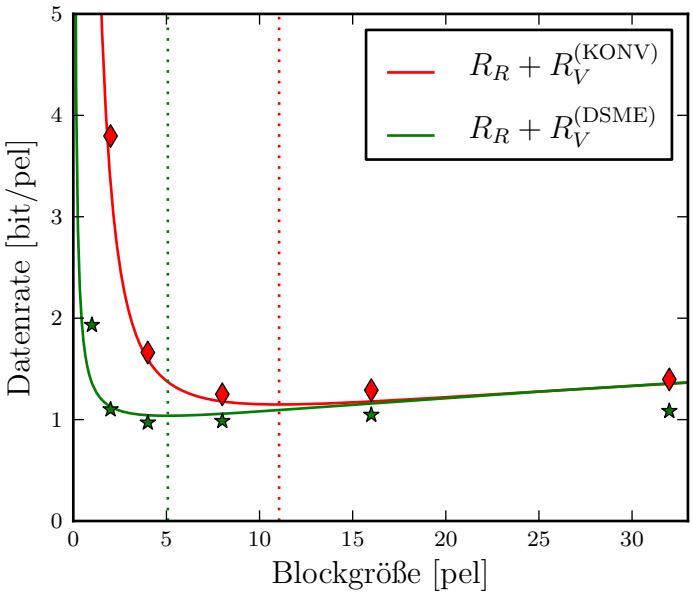


Abbildung 3.14: Gesamtdatenrate zur Codierung des quantisierten Prädiktionsfehlers und der Bewegungsvektoren für die Kimono-Sequenz.

4 Bewegungsschätzung am Decoder

Im vorangegangenen Kapitel wurde gezeigt, dass eine Schätzung der Bewegung am Decoder die benötigte Gesamtdatenrate zur Codierung einer Sequenz verringern kann. Bereits in aktuellen Videocodierstandards wird die Bewegung eines Blockes mit begrenztem Aufwand am Decoder geschätzt. Bei AVC wird von einem örtlich homogenen Bewegungsvektorfeld ausgegangen, sodass die Bewegung, wie in Abschnitt 2.2 beschrieben, mit Hilfe eines Medianfilters aus drei benachbarten Bewegungsvektoren prädiziert wird (MVP) und lediglich die Differenz zwischen prädiziertem und gemessenem Bewegungsvektor übertragen wird.

Eine Erweiterung, um auch zeitliche Korrelationen der Bewegungsvektoren auszunutzen, wurde in [44] vorgestellt und in [45] weiterentwickelt. Die Idee ist, den Bewegungsvektor nicht mit nur einem Prädiktor zu schätzen, sondern mehrere Kandidaten zu bestimmen und die Auswahl des geeignetsten Kandidaten zu signalisieren (*Motion Vector Competition*, MVC). Bei der Bestimmung der Kandidaten werden auch Bewegungsvektoren aus vorangegangenen Bildern verwendet, was insbesondere bei zeitlich homogenen Bewegungsvektorfeldern von Vorteil ist.

Jedoch nutzen beide Verfahren lediglich bereits vorhandene Bewegungsinformationen und sind somit häufig ungenau. Am Decoder stehen aber noch weitere Daten, wie zum Beispiel die bereits decodierten Bildpunkte, zur Verfügung, die zu einer besseren Schätzung der Bewegung genutzt werden können. So wurde vor der Standardisierung von H.261 untersucht, wie decodierte Bilder verwendet werden können, um die Bewegung ausschließlich am Decoder zu bestimmen. Eine Interpolation von Zwischenbildern wurde bereits 1985 in [52] vorgestellt und zur künstlichen Erhöhung der Bildwiederholungsfrequenz am Decoder eingesetzt. Diese Technik musste jedoch nicht in der Standardisierung berücksichtigt werden, da der Bitstrom unverändert bleibt.

Mit der Einführung der B-Bilder, für welche zusätzliche Referenzbilder aus der Zukunft zur Codierung bereitstehen, ist es auch am Encoder möglich geworden, die Bewegung zu interpolieren und diese Information während der Codierung zu nutzen. Diverse Forschungen [36, 40, 50] haben gezeigt, dass mit der Bewegungsschätzung am Decoder die Datenrate weiter reduziert werden kann. Daher wird dieses Gebiet auch bei den Standardisierungsaktivitäten innerhalb von JCT-VC berücksichtigt, und sogenannte *Tool Experiments* zur Evaluation werden durchgeführt [71].

Im folgenden Kapitel wird ein System vorgestellt, welches die Bewegung aus bereits decodierten Referenzbildern ermittelt und dabei die zeitliche Homogenität des Bewegungsvektorfeldes ausnutzt.

4.1 Architektur zur Bewegungsschätzung am Encoder und Decoder

Wie in Kapitel 2 beschrieben, wird in AVC eine Liste von bereits decodierten Bildern genutzt, um aus diesen Bildern das aktuell zu decodierende Bild zu prädizieren. Dazu wird zu jedem Block ein Bewegungsvektor übertragen und zusätzlich die Nummer des entsprechenden Referenzbildes innerhalb der Liste signalisiert. Wäre eines der Referenzbilder bereits ein bewegungskompensiertes Abbild des zu decodierenden Bildes, würde eine Übertragung von Bewegungsvektoren überflüssig werden. In einer Vorarbeit [40] wurde zum ersten Mal ein System vorgestellt (DSME), welches das aktuelle Bild mit Hilfe von bereits übertragenen Informationen schätzt und diese Prädiktion in die Referenzbildliste einfügt. Im Folgenden werden der Aufbau des Systems mit zusätzlichen Erweiterungen detailliert beschrieben und besondere Eigenschaften aufgezeigt.

4.1.1 Aufbau

Wie in [40] beschrieben, wird DSME in einen konventionellen Decoder eingebettet. Die nötigen Änderungen im Vergleich zum hybriden Decoder aus Abbildung 2.3 sind in Abbildung 4.1 grau hinterlegt.

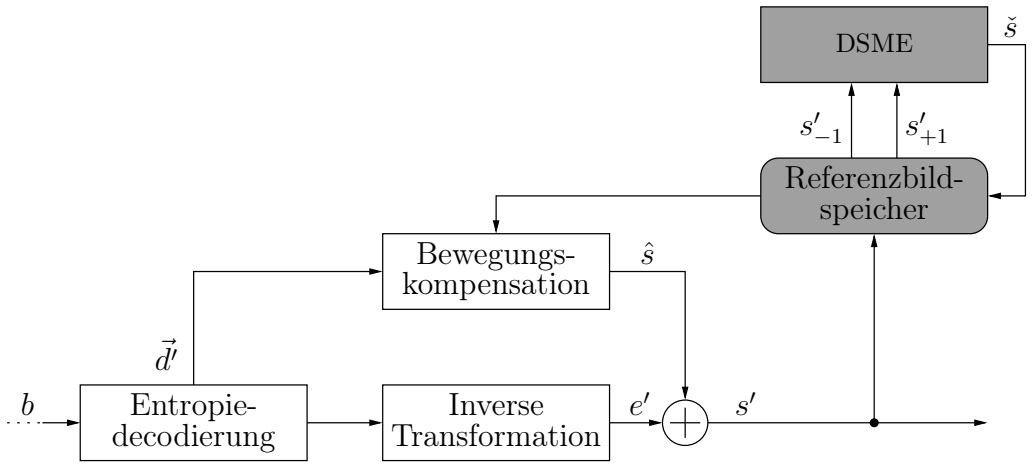


Abbildung 4.1: Vereinfachtes Blockdiagramm eines DSME-Decoders.

Der DSME-Block erhält aus dem Referenzbildspeicher ein zeitlich vorangegangenes Bild s'_{-1} und ein zukünftiges Bild s'_{+1}. Aus diesen beiden Bildern wird eine bewegungskompensierte Schätzung $\check{s}$ des zu decodierenden Bildes interpoliert. Da die Bewegungsschätzung innerhalb des DSME-Blocks zur Interpolation des zu decodierenden Bildes die Effizienz des vorgestellten Ansatzes maßgeblich beeinflusst, wird diese in Kapitel 4.2 detaillierter beschrieben.

Es ist auch möglich, lediglich vorangegangene Referenzbilder in dem DSME-Block zu verwenden. Ein möglicher Extrapolationsalgorithmus, welcher die Bewegung aus drei vorangegangenen Bildern schätzt, wird in [7] erläutert. Jedoch ist bei der Extrapolation grundsätzlich ein schlechteres Ergebnis als bei der Interpolation zu erwarten. Sie ist daher nur bei P-Bildern sinnvoll, wo eine Interpolation aufgrund des fehlenden Referenzbildes aus der Zukunft nicht möglich ist. Da in der Videocodierung P-Bilder für gewöhnlich weniger häufig auftreten, wurde dieser Ansatz nicht weiter verfolgt.

Nach der Berechnung des DSME-Bildes $\check{s}$ wird dieses dem Referenzbildspeicher zugeführt und in die Referenzbildliste eingetragen. Dabei werden alle folgenden Bilder, wie in Abbildung 4.2 gezeigt, entsprechend verschoben. Somit erhöht sich die

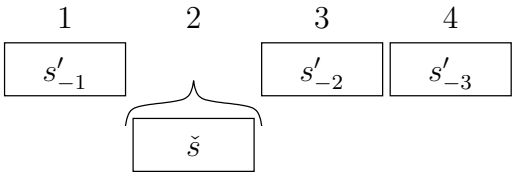


Abbildung 4.2: Die Referenzbildliste wird durch das Einfügen des DSME-Bildes, wie in diesem Beispiel für die Position 2, erweitert.

Gesamtzahl der Referenzbilder um eins. Die Auswahl von DSME wird bei diesem Aufbau indirekt über den Referenzbildindex signalisiert.

Um eine Drift zwischen Encoder und Decoder zu vermeiden, muss auch der Encoder aus Abbildung 2.2 entsprechend geändert werden. Abbildung 4.3 zeigt das erweiterte Blockschaltbild des Encoders.

Zum leichteren Verständnis werden die DSME-Algorithmen im Folgenden aus der Sicht des Encoders beschrieben, da in diesem alle Entscheidungen zur Codierung getroffen werden.

4.1.2 Merkmale

Die in Abbildung 4.3 vorgestellte Architektur hat den Vorteil, dass alle konventionellen Codiertools die Möglichkeit haben, das bereits bewegungskompensierte DSME-Bild zu nutzen. Somit ist es nicht notwendig, einen zusätzlichen Modus für DSME zu definieren. Die Signalisierung erfolgt sehr effizient indirekt über den Referenzbildindex. Jedoch ist zu beachten, dass der Referenzbildindex in AVC mit variabler Codewortlänge codiert wird. Wie in Abschnitt 2.2.1 beschrieben, werden kleine Indizes kürzeren Codewörtern zugeordnet, da davon ausgegangen wird, dass diese Bilder häufiger zur Prädiktion herangezogen werden und somit häufiger übertragen werden müssen. Die Kosten der Codierung steigen daher mit dem Wert des Referenzbildindex. Es muss also darauf geachtet werden, welcher Referenzbildindex dem DSME-Bild zugeordnet wird. Bei häufiger Verwendung des DSME-Bildes, sollte der Index

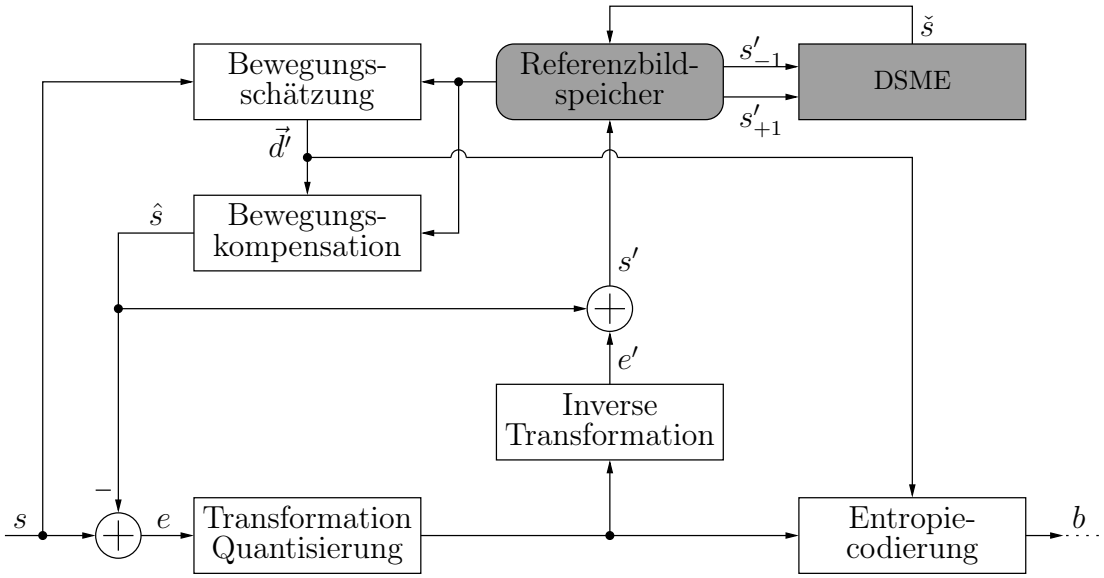


Abbildung 4.3: Vereinfachtes Blockdiagramm eines DSME-Encoders.

möglichst klein sein. In Abschnitt 5.4.3 wird dazu eine Auswertung des optimalen Referenzbildindex für das DSME-Bild durchgeführt.

Die Signalisierung von DSME mit Hilfe des Referenzbildindex ist jedoch nicht sehr flexibel. Sollte es einen AVC-Modus geben, welcher sehr häufig oder selten auf das DSME-Bild zurückgreift, kann die Signalisierung des DSME-Bildes für diesen Modus nicht der geänderten Statistik angepasst werden.

Diese Architektur bietet jedoch den Vorteil, nichtlineare Bewegung zu kompensieren. Wie in Abschnitt 3.1 beschrieben, sollten interpolierte Blöcke, die aufgrund beschleunigter Bewegung falsch positioniert wurden, mit Hilfe eines zusätzlichen Vektors verschoben werden, anstatt den entstandenen Fehler durch Übertragung des Residuums zu eliminieren. Innerhalb des Referenzbildspeichers wird nicht zwischen DSME-Bild und anderen Referenzbildern unterschieden. Somit wird, wie bei allen anderen Referenzbildern auch, beim DSME-Bild eine Bewegungsschätzung durchgeführt. Liegt lediglich lineare Bewegung vor, ist der durch die konventionelle Bewegungsschätzung ermittelte Bewegungsvektor Null, da das DSME-Bild bereits ein bewegungskompensiertes Abbild des zu codierenden Bildes ist. Somit kann dieser Vektor sehr effizient codiert werden. Ist die Annahme der zeitlich konstanten Bewegung nicht gegeben, kann der fehlerhaft kompensierte Block trotzdem durch die Übertragung eines entsprechenden Bewegungsvektors genutzt werden (Abbildung 3.2).

Der entscheidende Vorteil dieser Architektur ist die gekapselte Implementierung, wodurch sich die Bewegungsschätzung und -kompensation innerhalb des DSME-Blocks sehr einfach austauschen lässt. So kann jeder Algorithmus, welcher aus zwei oder mehreren Referenzbildern eine Interpolation erstellt, verwendet werden. Da-

durch ist die Blockgröße bei der Bewegungsschätzung innerhalb der DSME-Interpolation unabhängig von der Blockgröße in dem konventionellen Teil des Coders. Es wird möglich, ein dichtes Bewegungsvektorfeld zu berechnen, um auch an Objektgrenzen eine exakte Interpolation zu erreichen. Es ist ebenfalls denkbar, eine Segmentierung durchzuführen und die Bewegung für jedes Segment einzeln zu berechnen.

In [50] wurde ein erweiterter Skipmodus vorgestellt, welches parallel zu DSME entwickelt wurde, die zuvor genannten Freiheiten jedoch nicht besitzt. Der vorgestellte Algorithmus ersetzt die von AVC bekannte zeitliche und örtliche Prädiktion der Bewegung im Skipmodus durch eine bidirektionale Bewegungsschätzung. Das System wurde in [51] erweitert, sodass die Bewegung auch im sogenannten Direktmodus durch bidirektionale Bewegungsschätzung am Decoder geschätzt werden kann. Der Vorteil liegt darin, dass der Decoder die Bewegung nicht für das gesamte Bild schätzen muss, sondern lediglich für Blöcke, die im Skip- oder Direktmodus codiert wurden. Somit kann die geschätzte Bewegung jedoch nicht für andere Modi verwendet werden. Die Evaluation in Abschnitt 5.4.2 zeigt allerdings, dass auch die bewegungskompensierte Interprädiktion von der decoderseitigen Bewegungsschätzung profitiert. Außerdem ist der Algorithmus zur Bewegungsschätzung nicht so flexibel und einfach austauschbar wie bei der vorgestellten DSME-Architektur.

4.2 Bewegungskompensierende Interpolation

Eine exakte Bewegungsschätzung ist der entscheidende Faktor für die Effizienz der decoderseitigen Bewegungsschätzung. Wurde die Bewegung zwischen zwei Referenzbildern ermittelt, kann durch die Annahme der linearen Bewegung leicht auf die Bewegung des dazwischen liegenden Bildes geschlossen werden. Die ermittelten Vektoren können anschließend zur Erstellung einer bewegungskompensierten Prädiktion des zu codierenden Bildes genutzt werden. Ist die geschätzte Bewegung – und somit auch die Interpolation – ungenau, werden die konventionellen Algorithmen nur selten das DSME-Bild als Referenz nutzen, wodurch der Codiergewinn nicht gesteigert werden kann.

In AVC findet eine einfache, blockbasierte Bewegungsschätzung Verwendung. Ein Block im zu codierenden Bild wird in anderen Referenzbildern innerhalb eines definierten Suchbereichs gesucht. Als Optimierungskriterium wird dabei die mittlere absolute Differenz (*Mean Absolute Difference*, MAD) zwischen dem zu codierenden Block und dem Kandidaten berechnet, da davon ausgegangen wird, dass bei steigender MAD auch das zu übertragende Residuum ansteigt. Ist diese Differenz gering, wird der entsprechende Bewegungsvektor für die weitere Codierung verwendet. Jedoch ist es bei dieser konventionellen Bewegungsschätzung unerheblich, ob die ermittelte Bewegung der wahren Bewegung entspricht, oder ob lediglich ein Block mit ähnlicher Textur gefunden wurde. Solange der Prädiktionsfehler und somit das Residuum gering sind, wird ein Kompressionsgewinn erzielt.

Jedoch kann, wie in Abbildung 4.4 gezeigt, durch die Annahme der linearen Bewegung bei DSME, eine falsche Zuordnung zu großen Fehlern führen. In diesem Beispiel

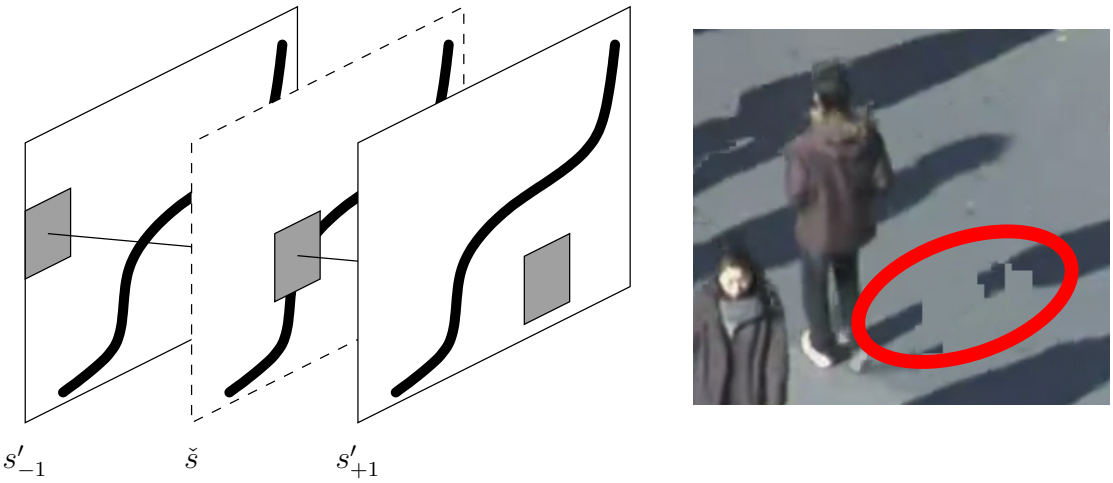


Abbildung 4.4: Bewegungskompensation aufgrund falscher Korrespondenzen (a) kann, wie für ein Detail der PeopleOnStreet Sequenz gezeigt (b), große Fehler verursachen.

hat die Bewegungssuche zwischen den Referenzbildern s'_{-1} und s'_{+1} einen Vektor ermittelt, welcher die MAD ohne Berücksichtigung des zu codierenden Bildes minimiert. Da dieser Vektor jedoch nicht der wahren Bewegung innerhalb der Sequenz entspricht, wird bei der Bewegungskompensation von $\check{s}$ falscher Inhalt interpoliert. In Abbildung 4.4b ist der dadurch entstandene Fehler deutlich zu erkennen.

Aus diesem Grund ist für DSME ein anderer Bewegungschätzalgorithmus notwendig, welcher die wahre Bewegung der Objekte verfolgen kann. Daher wird im Folgenden eine speziell angepasste, hierarchische Bewegungsschätzung vorgestellt. Anschließend werden weitere Alternativen mit unterschiedlichen Eigenschaften vorgestellt.

4.2.1 Hierarchische Schätzung der wahren Bewegung

Die Genauigkeit der Bewegungsschätzung am Decoder beeinflusst direkt die Effizienz der DSME-Architektur. Mit einem exakten Bewegungsvektorfeld ist eine gute Interpolation des DSME-Bildes möglich, welches anschließend häufiger bei den konventionellen Codiertools Verwendung findet. Jedoch muss bei der blockbasierten Bewegungsschätzung darauf geachtet werden, dass nicht ein falsches Minimum gefunden wird (Abbildung 4.4). Dieses Problem lässt sich mit einer hierarchischen Bewegungsschätzung verringern. Dabei wird als erstes die grobe Bewegung mit Hil-

fe von sehr großen Blöcken berechnet. Anschließend werden die Blöcke verkleinert und ausgehend von der bereits ermittelten Bewegung eine weitere Bewegungssuche durchgeführt.

Bereits in [5] wurde eine hierarchische Schätzung der Bewegung zwischen zwei Referenzbildern vorgestellt. Innerhalb von drei Stufen wurde die Blockgröße von 65×65 auf 13×13 verringert. Die Bewegung wird dabei, ähnlich wie beim Newtonschen Näherungsverfahren [47], iterativ bestimmt. In Abbildung 4.5 ist diese Methode an einem kontinuierlichen, eindimensionalen Beispiel gezeigt, bei dem lediglich die Bewegung in x-Richtung einer beliebigen Zeile y_0 betrachtet wird. Die Intensität

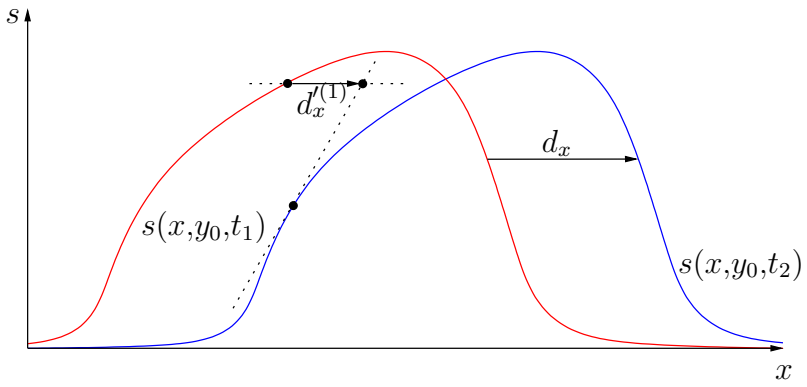


Abbildung 4.5: Eindimensionales Beispiel für die iterative Schätzung der Bewegung.

dieser Zeile y_0 zum Zeitpunkt t_1 wird durch die Funktion $s(x,y_0,t_1)$ beschrieben. Durch Bewegung in x-Richtung haben sich die Intensitätswerte zum Zeitpunkt t_2 um d_x verschoben ($s(x,y_0,t_2)$). Mit Hilfe der Steigung lässt sich in der ersten Iteration eine Bewegung $d_x'^{(1)}$ schätzen. Dieser Wert wird genutzt, um die Bewegung zu kompensieren. Anschließend wird mit Hilfe der bewegungskompensierten Funktion $s(x + d_x'^{(1)},y_0,t_1)$ eine weitere Bewegung $d_x'^{(2)}$ berechnet. Diese Schritte werden so lange durchgeführt, bis die akkumulierte Bewegung

$$d_x' = d_x'^{(1)} + d_x'^{(2)} + \dots \tag{4.1}$$

sich nur noch minimal ändert. Dieses Grundprinzip lässt sich auf den zweidimensionalen Fall erweitern, um einen Bewegungsvektor $\vec{d}'$ zu ermitteln. Bei großen Bewegungen kann es jedoch vorkommen, dass viele Iterationen nötig sind, um den Bewegungsvektor $\vec{d}'$ zu bestimmen. Außerdem ist die Berechnung von $\vec{d}'_{(i)}$ in jeder Iteration relativ aufwändig, sodass die gesamte Bewegungsschätzung sehr komplex ist.

Im Gegensatz dazu wurden sogenannte *Block-Matching*-Algorithmen, bei denen der korrespondierende Block mit Hilfe die MAD bestimmt wird, in den letzten Jahren durch Verwendung von Grafikprozessoren (*Graphics Processing Unit*, GPU) erheblich optimiert [21]. Daher wird auch für DSME, wie in Abbildung 4.6 gezeigt,

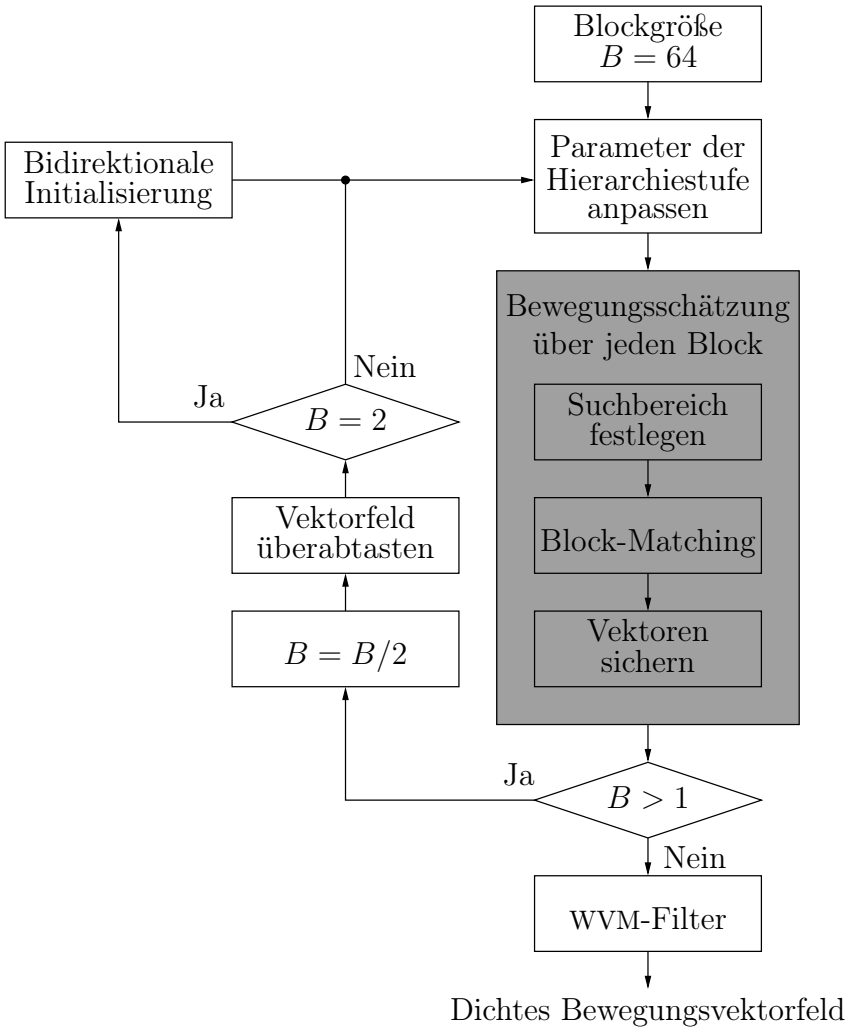


Abbildung 4.6: Blockdiagramm der hierarchischen Bewegungsschätzung.

ein hierarchisch aufgebauter Block-Matching-Algorithmus zur Bewegungsschätzung verwendet.

Die Bewegungsschätzung beginnt in der ersten Hierarchiestufe H_1 mit Blöcken mit 64×64 Bildpunkten und einem Suchfenster von $\pm128\,\text{pel} = 256\,\text{pel}$. Die Blockgröße wird für jede weitere Hierarchieebene H_i halbiert, bis ein dichtes Bewegungsvektorfeld zur Verfügung steht. Entsprechend wird das bisher ermittelte Vektorfeld ohne Interpolation überabgetastet, was bedeutet, dass von jedem Vektor drei weitere Kopien erzeugt werden. Auch der Suchbereich wird bei jeder Hierarchiestufe verringert, da die grobe Bewegung bereits in vorangegangenen Schritten bestimmt wurde. Zur Verringerung der Komplexität werden die Referenzbilder bei der ersten Bewegungsschätzung, wie in [5] vorgeschlagen, vor dem Block-Matching mit einem Tiefpass

gefiltert und anschließend unterabgetastet. Dadurch müssen weniger Positionen bei der Bewegungssuche geprüft werden. Die Genauigkeit der ermittelten Bewegungsvektoren verringert sich somit auf 4 pel. In allen weiteren Hierarchiestufen werden die ungefilterten Bilder zur Bewegungssuche verwendet.

Die Bewegung wird in jeder Hierarchiestufe mit einem konventionellen Block-Matching-Algorithmus geschätzt, welcher die MAD eines Suchblocks minimiert. Die Untersuchung in Abschnitt 5.2 zeigt, dass die Bewegungsschätzung mit der mittleren quadratischen Differenz (*Mean Squared Difference*, MSD) als Optimierungskriterium leichte Gewinne bringt. Jedoch sind diese Gewinne zu gering, um den erhöhten Aufwand bei der Berechnung zu rechtfertigen. Um unempfindlicher gegen Rauschen in den Referenzbildern zu sein, wird bei der Bewegungssuche für Blöcke kleiner als 16×16 Bildpunkte (H_4 bis H_7) ein Überlappungsbereich verwendet. Bei dieser – auch als *Overlapped Block Motion Estimation* (OBME) [19] bekannten – Technik werden zur Bewegungssuche, wie in Abbildung 4.7 gezeigt, größere Blöcke verwendet als zur späteren Bewegungskompensation.

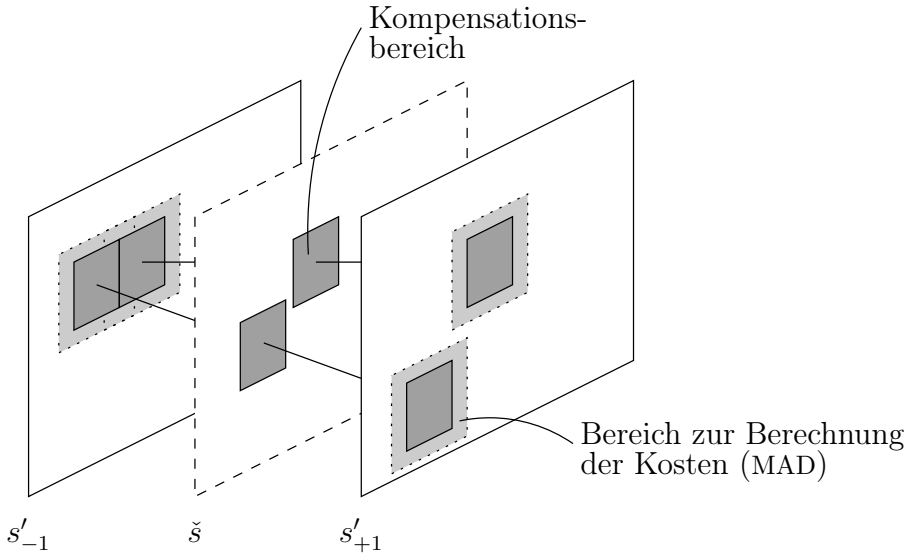


Abbildung 4.7: Bewegungsschätzung mit überlappenden Blöcken (OBME).

Durch die vorwärts gerichtete Bewegungsschätzung von Bild s'_{-1} zu s'_{+1} kann es bei der Interpolation des DSME-Bildes, wie in Abbildung 4.8 dargestellt, zu Lücken oder Überlappungen kommen. Um diese undefinierten Bereiche zu vermeiden, wird ab einer Blockgröße von 2×2 (H_6) eine bidirektionale Bewegungsschätzung angewendet. Dazu wird, wie in [3] vorgestellt, das Vektorfeld dem Blockraster des DSME-Bildes angepasst (bidirektionale Initialisierung). Für jeden Block im DSME-Bild wird der Bewegungsvektor gewählt, welcher diesen am dichtesten zur Mitte durchstößt. Bei der Bewegungsschätzung in den folgenden Hierarchiestufen wird nun der Bewegungsvektor, wie in Abbildung 4.9 gezeigt, an der Mitte des Blockes fixiert und die

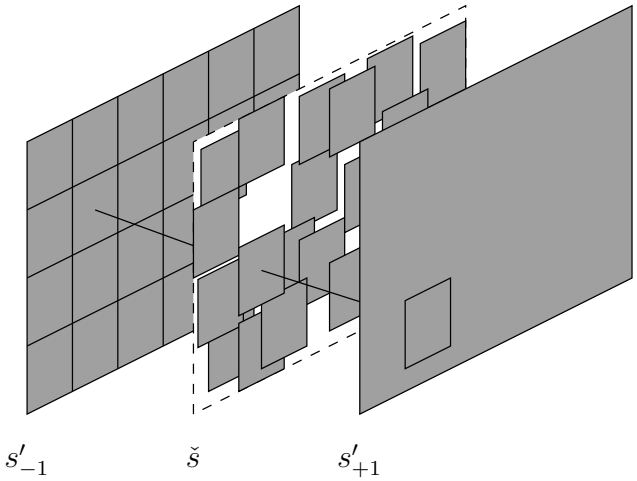


Abbildung 4.8: Aufgrund der vorwärtsgerichteten Bewegungsschätzung entstehen Lücken und überlappende Bereiche.

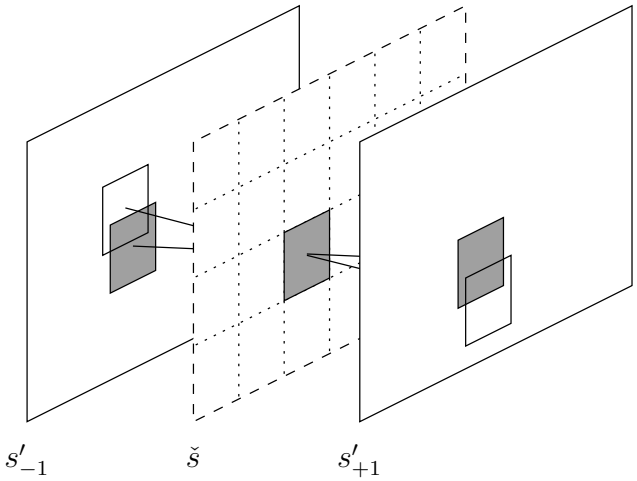
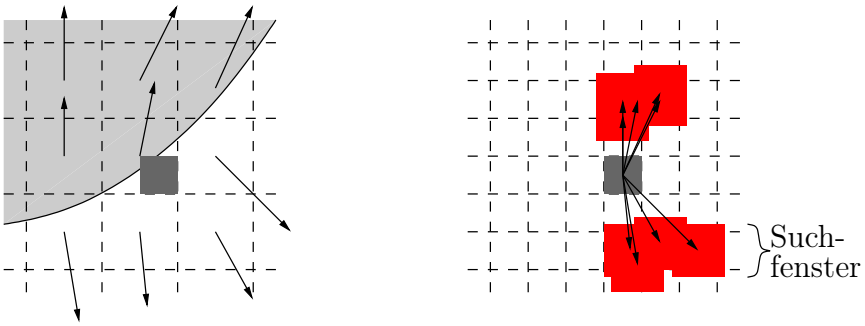


Abbildung 4.9: Ausrichtung am Blockraster des DSME-Bildes und anschließende bidirektionale Bewegungsschätzung zur Vermeidung von Lücken und Überlappungen im DSME-Bild.

Suchblöcke in den Referenzbildern gegeneinander verschoben. Die Einhaltung der Annahme von zeitlich konstanter Bewegung wird somit gewährleistet.

Wie bereits erwähnt, wird der Suchbereich in jeder Hierarchiestufe verringert, da eine grobe Schätzung der Bewegung bereits bekannt ist. Jedoch kann es vorkommen, dass ein Teil des Blocks, wie in Abbildung 4.10a skizziert, eine ganz andere Bewegung vollzieht, als die zuvor ermittelte Bewegung. Aus diesem Grund wird



(a) Hierarchiestufe H_{i-1} (b) Hierarchiestufe H_i

Abbildung 4.10: Der Suchbereich (rot) wird aus der vorangegangenen Hierarchiestufe H_{i-1} hergeleitet, um die Bewegung an Objektgrenzen besser verfolgen zu können.

hier der Suchbereich für die Hierarchiestufe H_i mit Hilfe von neun benachbarten Bewegungsvektoren aus der vorangegangenen Hierarchiestufe H_{i-1} berechnet. Der Suchbereich, in Abbildung 4.10b rot dargestellt, setzt sich dabei aus den Suchfenstern der neun Kandidatvektoren zusammen. Somit ist es möglich, Bewegung auch an Objektkanten korrekt zu schätzen, obwohl für den Block in der vorangegangene Hierarchiestufe eine andere Bewegung geschätzt wurde. Ein Spezialfall des dynamischen Suchbereichs ist das für diese Arbeit entwickelte Vektorlatching [42]. Ist die aktuelle Blockgröße deutlich kleiner als die im Bild vorhandenen Objekte, kann davon ausgegangen werden, dass die Bewegung der Objekte bereits zuverlässig geschätzt wurde. An Objektgrenzen muss lediglich entschieden werden, zu welchem Objekt der aktuelle Block gehört. Dies lässt sich sehr einfach realisieren, indem das Suchfenster zu Null gesetzt wird. Dem Block wird in diesem Fall direkt ein Vektor von den neun Kandidatvektoren aus Abbildung 4.10a zugeordnet. Zusätzlich zu der Verringerung des Rechenaufwandes vermindert dieses Verfahren deutlich fehlerhafte Bewegungsvektoren durch lokale Minima.

Im Vorfeld der DSME-Entwicklung wurde gezeigt, dass auch bei decoderseitiger Bewegungschätzung die Interpolation mit Hilfe von sub-pel-genauer Schätzung verbessert werden kann [39]. Daher wird die Bewegung ab einer Blockgröße von 8×8 mit $^1/_4$-pel-Genauigkeit geschätzt. Die $^1/_2$-pel-Positionen werden mit dem aus AVC bekannten Wienerfilter berechnet. Anstatt ein bilineares Filter zur Berechnung der

1/4-pel-Positionen zu verwenden, kommt ein angepasstes Wienerfilter [39] zum Einsatz. Die Koeffizienten der beiden Filter sind in Tabelle 4.1 angegeben.

Tabelle 4.1: Filterkoeffizienten zur Interpolation von 1/2-pel- und 1/4-pel-Positionen.

Position	Filterkoeffizienten
1/2-pel	$(1, -5, 20, 20, -5, 1)/32$
1/4-pel	$(5, -18, 114, 37, -11, 1)/128$

Um eventuelle Ausreißer im Bewegungsvektorfeld zu eliminieren, wird die Bewegung am Ende der hierarchischen Schätzung mit Hilfe eines gewichteten Vektor-Median-Filters (*Weighted Vector Median Filter*, WVMF) [2] geglättet. Dazu wird für jeden Block ein Vektor $\vec{d}_{\mathrm{WVM}}$ als Bewegungsvektor entsprechend

$$\sum_{i=1}^{M} w_i \left\| \vec{V}_{\mathrm{WVM}} - \vec{V}_i \right\|_2 \leq \sum_{i=1}^{M} w_i \left\| \vec{V}_j - \vec{V}_i \right\|_2 \qquad \forall j = 1 \ldots N \tag{4.2}$$

zugeordnet. Hierbei ist $\| \ldots \|_2$ die Euklidische Norm und M die Anzahl der benachbarten Bewegungsvektoren, die im Filter berücksichtigt werden sollen. Das Gewicht w_i ist antiproportional zum Prädiktionsfehler des aktuellen Blockes, wenn der entsprechende Vektor $\vec{V}_i$ zur Bewegungskompensation genutzt wird. Somit erhalten Vektoren, welche gut geeignet sind (geringer Prädiktionsfehler), eine große Gewichtung.

Das gefilterte Bewegungsvektorfeld wird nun zur Interpolation des DSME-Bildes verwendet. Wie in Abbildung 4.11 zu sehen, lässt sich mit diesem Algorithmus die wahre Bewegung deutlich besser schätzen.

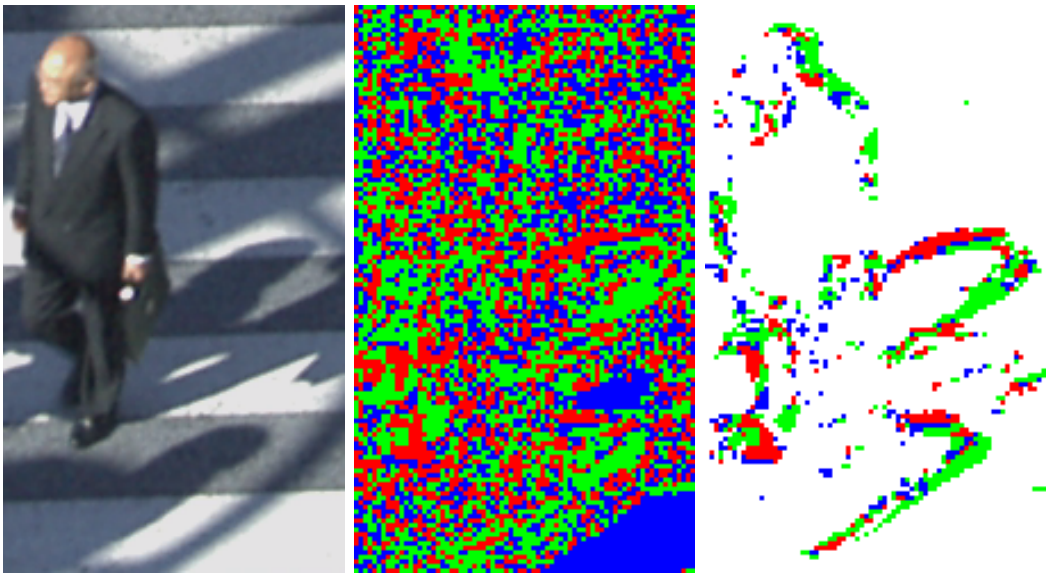
Abbildung 4.11: Verglichen mit Abbildung 4.4, lässt sich mittels der hierarchischen Bewegungsschätzung eine deutlich bessere Interpolation erzeugen.

4.2.2 Kennzeichnung von Verdeckung durch optionale Seiteninformation

Ein Problem bei der DSME-Bild-Interpolation stellen Verdeckung und freiwerdender Hintergrund dar, da das DSME-Bild immer bidirektional aus beiden Referenzbildern berechnet wird. Sollte ein Bereich in einem Referenzbild verdeckt worden sein, wird der falsche Inhalt dieses Referenzbildes trotzdem zur Interpolation verwendet, anstatt lediglich Bildinhalte des anderen Referenzbildes zu verwenden.

Zur Bestimmung von verdeckten Bereichen, könnten Bewegungstrajektorien über mehrere Bilder ermittelt werden. Dieser Ansatz würde jedoch die Komplexität am Decoder signifikant erhöhen. Am Encoder ist die Bestimmung dagegen deutlich einfacher, da zusätzlich das zu codierende Bild s zur Verfügung steht. Über zusätzliche Seiteninformation kann der Encoder dem Decoder mitteilen, welche Bereiche bidirektional interpoliert werden können und für welche lediglich ein Referenzbild verwendet werden soll.

Diese zusätzliche Seiteninformation ist exemplarisch für die PeopleOnStreet-Sequenz in Abbildung 4.12 visualisiert. Das erste Bild zeigt einen Ausschnitt aus dieser



(a) Originalausschnitt (b) Seiteninformation (c) Reduziert

Abbildung 4.12: Bewegungskompensation bei Verdeckung in der PeopleOnStreet-Sequenz: von vorangegangenem Referenzbild (rot), von nachfolgendem Referenzbild (grün), bidirektional von beiden Referenzbildern (blau).

Sequenz. Bei der bidirektionalen Interpolation dieses Bildes ergibt sich eine Varianz des Fehlers zwischen DSME- und Originalbild von $\sigma_e^2 = 296$. Mit Hilfe von

Gleichung 3.22 und unter Annahme feiner Quantisierung ($Q = 1$), ergibt sich die theoretische Datenrate zur Kompensation des Interpolationsfehlers zu 6,05 bit pro Bildpunkt. Da im Folgenden nur die Datenratenreduktion interessant ist, hat die gewählte Quantisierung Q keinen Einfluss.

In Abbildung 4.12b ist die am Encoder getroffene Wahl der optimalen Interpolationsmethode für jeden Bildpunkt gezeigt. Mit dieser Interpolation verringert sich die Fehlervarianz und es ergibt sich eine theoretische Rate von 5,88 bit pro Bildpunkt. Die Entropie dieser Seiteninformation beträgt 1,12 bit pro Bildpunkt. Trotz der zur rein bidirektionalen Interpolation um 0,17 bit pro Bildpunkt verringerten Datenrate des Interpolationsfehlers, ist die zusätzliche Rate der Seiteninformation zu hoch, um noch einen Gesamtgewinn erzielen zu können.

Es kann jedoch davon ausgegangen werden, dass keine Verdeckung vorliegt, wenn die entsprechenden Blöcke in den beiden Referenzbildern ähnlich sind und somit eine geringe MAD zueinander haben. Daher kann der Decoder entscheiden, ob ein Block bidirektional interpoliert werden kann, oder ob Zusatzinformationen vom Encoder benötigt werden. Liegt die MAD zwischen den bewegungskompensierten Blöcken der Referenzbilder unter einem Schwellwert, kann bidirektionale Interpolation ohne zusätzliche Signalisierung verwendet werden. Nur für Blöcke mit hohen Differenzen müssen die optimalen Interpolationsmethoden signalisiert werden. Abbildung 4.12c visualisiert diese reduzierte Seiteninformation. Für die weißen Bereiche hat der Decoder selbstständig entschieden, dass bidirektionale Interpolation verwendet werden soll. Lediglich die Information aus den farbigen Bereichen muss vom Encoder übertragen werden. Der mittlere Informationsgehalt pro Bildpunkt sinkt dadurch auf 0,12 bit. Der bei dieser Interpolation entstandene Fehler kann mit einer Rate von 5,91 bit pro Bildpunkt kompensiert werden. Dies ergibt gegenüber der rein bidirektionalen DSME-Interpolation eine Einsparung von 0,14 bit pro Bildpunkt, wodurch sich ein theoretischer Gewinn von 0,02 bit pro Bildpunkt erzielen lässt.

Mit dieser Methode lässt sich auch die Datenrate bei anderen Sequenzen verringern. So wird bei der BasketballDrive-Sequenz die Gesamtdatenrate um 0,15 bit pro Bildpunkt reduziert. Jedoch steht der theoretische Gewinn in keinem Verhältnis zu dem zusätzlichen Aufwand durch die Signalisierung der Seiteninformation. Außerdem werden bei den Standardisierungsgremien, wie dem JCT-VC, solche weitreichenden Änderungen in der Syntax des Bitstroms für gewöhnlich nur bei sehr hohen Kompressionsgewinnen akzeptiert. Daher wird diese Methode nicht weiter untersucht.

4.2.3 Alternative Algorithmen zur bewegungskompensierten Interpolation

Durch die flexible DSME-Architektur ist es nicht zwingend erforderlich, die Bewegung zwischen den Referenzbildern mit einem blockbasierten Verfahren zu schätzen. So

wurde in [49] ein Verfahren zur Interpolation des DSME-Bildes vorgestellt, welches ein verformbares Gitternetz zu Bewegungsschätzung verwendet (Abbildung 4.13). Dazu werden in einem Referenzbild Merkmalspunkte mit Hilfe eines Harris-Kan-

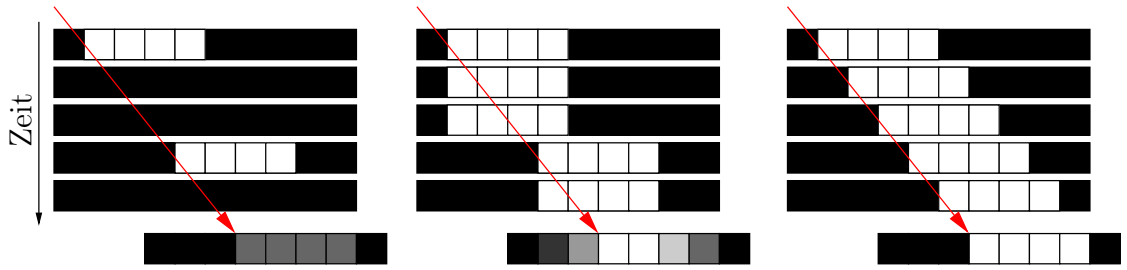
Abbildung 4.13: Aufteilung eines Ausschnitts der PeopleOnStreet-Sequenz mit Hilfe eines Gitternetzes.

tendetektors [20] ermittelt und diese durch eine erweiterte Kanade-Lucas-Tomasi-Merkmalsverfolgung [61] im zweiten Referenzbild gesucht. Um ein dichtes Bewegungsvektorfeld zu erhalten, werden die Merkmalspunkte mit Hilfe einer Delaunay-Triangulation zu Dreiecken zusammengefasst. Anschließend wird jeder Bildpunkt innerhalb eines Dreiecks durch eine affine Transformation auf das zu interpolierende DSME-Bild projiziert. Dieses Verfahren arbeitet aufgrund der Kompensation von affinen Verzerrungen besonders gut bei Sequenzen mit Zoom oder Bewegungen längs der Kameraachse. Bei Sequenzen mit überwiegend translatorischer Bewegung wird jedoch nicht die Effizienz des blockbasierten Verfahrens erreicht.

Ein weiteres Verfahren, welches in der DSME-Architektur zur Bewegungsschätzung genutzt werden kann, basiert auf der Bestimmung des Vektorfeldes – auch optischer Fluss genannt – mit Hilfe von Variationsansätzen. Diese Ansätze haben die Eigenschaft, verschiedene Annahmen über das Bewegungsvektorfeld mathematisch korrekt im Rahmen eines gemeinsamen Minimierungsproblems zu formulieren [9]. Jedoch sind diese Verfahren auf relativ geringe Bewegungen ausgelegt und werden bei großen Bildern durch die globale Minimierung sehr komplex.

Algorithmen zur Bewegungsschätzung ohne Verwendung des Originalbildes sind außerdem in aktuellen LCD- und Plasmabildschirmen zu finden. Bei diesen sogenannten *Hold-Type*-Displays wird der Zustand eines Bildpunktes während der Darstellung eines Bildes nicht geändert (Erhaltungsdarstellung). Im Gegensatz dazu, wird bei konventionellen Röhrenmonitoren (*Impulse-Type*-Display) ein Bildpunkt nur kurz durch den Elektronenstrahl angeregt und erlischt dann wieder.

In [4] wird gezeigt, dass die Probleme durch die Erhaltungsdarstellung auf die Eigenschaften des menschlichen Sehapparates zurückzuführen sind. Während der Verfolgung eines bewegten Objekts auf dem Bildschirm (*Smooth Pursuit Eye Tracking*) integriert das Auge die Helligkeit über eine Zeitspanne auf. Eine schematische Darstellung dieses Effekts ist für die beiden Bildschirmtypen in Abbildung 4.14 gezeigt. Zu sehen ist eine Bildschirmzeile zu fünf verschiedenen Zeitpunkten und die durch das Auge wahrgenommene Zeile.



a) Impulse-Type b) Hold-Type c) Hold-Type mit Zwischenbildern

Abbildung 4.14: Veranschaulichung der integrierenden Eigenschaften des Auges für verschiedene Darstellungsarten. Der Pfeil beschreibt die verfolgende Bewegung des Auges. Die letzte Zeile zeigt das vom Betrachter wahrgenommene Bild.

Beim Impulse-Type-Display wird das bewegte Objekt kurz dargestellt (Abbildung 4.14a). Der rote Pfeil beschreibt die Verfolgung des Objekts durch das Auge. Bei der Integration der Helligkeit über die Zeit ergibt sich die unten abgebildete Zeile. Die Verringerung der Helligkeit wird dabei durch hellere Ansteuerung der einzelnen Bildpunkte kompensiert. Im Gegensatz dazu werden bei Hold-Type-Displays (Abbildung 4.14b) die Objektkanten unscharf, da der Bildinhalt erhalten bleibt. Diese Bewegungsunschärfe (*Motion Blur*) kann sich sehr störend auf die subjektiv wahrgenommene Qualität auswirken.

Wie in Abbildung 4.14c dargestellt, kann durch Einfügen von bewegungskompensierten Zwischenbildern die Bewegungsunschärfe verringert werden. Daher wird in vielen Bildschirmen die sogenannte *Frame Rate Up Conversion* (FRUC) [13] verwendet, um die Bewegungsunschärfe bei Hold-Type-Displays zu verringern. Ähnlich wie in [52], wird dabei die Bildwiederholfrequenz erhöht, indem Zwischenbilder berechnet werden.

Ein sehr effektives Verfahren zur Zwischenbildberechnung wird in [75] vorgestellt. Bei dem sogenannten STAR-Verfahren (*Spatio-Temporal Autoregressive*) wird jeder zu interpolierende Bildpunkt aus einer Linearkombination von zeitlich und örtlich benachbarten Bildpunkten berechnet (Abbildung 4.15). Untersuchungen in [75] haben gezeigt, dass mit diesem Verfahren eine bessere Interpolation als mit Hilfe von

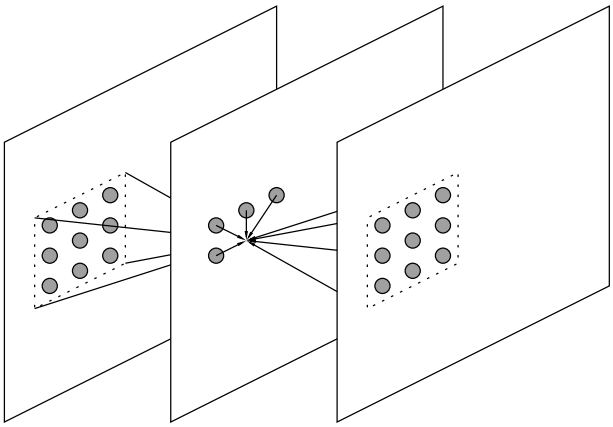


Abbildung 4.15: Interpolation eines Bildpunktes mit Hilfe von STAR durch eine Linearkombination zeitlich und örtlich benachbarter Bildpunkte.

3D Recursive Search (3DRS) [16] und *Adaptive Overlapped Block Motion Compensation* (AOBMC) [12] erreicht werden kann.

Jedoch sind FRUC-Algorithmen häufig für hohe Bildwiederholfrequenzen entwickelt worden, da beispielsweise ein Signal von 50 Hz auf 100 Hz hochgerechnet werden soll. Somit ergibt sich, wie in Abbildung 4.16a skizziert, ein zeitlicher Abstand von $T_t = \frac{1}{f_t} = 20\,\text{ms}$, in welchem nur geringere Bewegungen auftreten. Im Gegensatz dazu, kann der zeitliche Abstand zwischen zwei Referenzbildern bei der Videocodierung aufgrund von hierarchischen B-Bildern auf die achtfache Zeit ansteigen (Abbildung 4.16b). Einzelne Blöcke können in dieser Zeit eine deutlich größere Bewegung vollziehen, wodurch eine robustere Bewegungsschätzung benötigt wird.

Trotz der Nachteile von FRUC-Algorithmen für DSME kann die Verwendung Vorteile bringen. Nutzt der DSME-Decoder den FRUC-Algorithmus, welcher bereits in LCD- oder Plasmabildschirmen implementiert ist, kann die zusätzliche Komplexität durch DSME deutlich verringert werden.

4.3 Kombination mit Decoder-side Motion Vector Derivation

Im Gegensatz zum vorgestellten DSME-Ansatz, in welchem konstante Bewegung zwischen den Referenzbildern angenommen wird, schätzt DMVD [36] die Bewegung unter der Annahme, dass örtlich benachbarte und bereits decodierte Bildbereiche, die gleiche Bewegung vollzogen haben. Aus diesem Grund werden hier beide Techniken kombiniert, um örtliche sowie zeitliche Abhängigkeiten auszunutzen [43]. Im folgenden Abschnitt wird das Grundprinzip von DMVD kurz erläutert, um anschließend auf die Verknüpfung beider Techniken eingehen zu können.

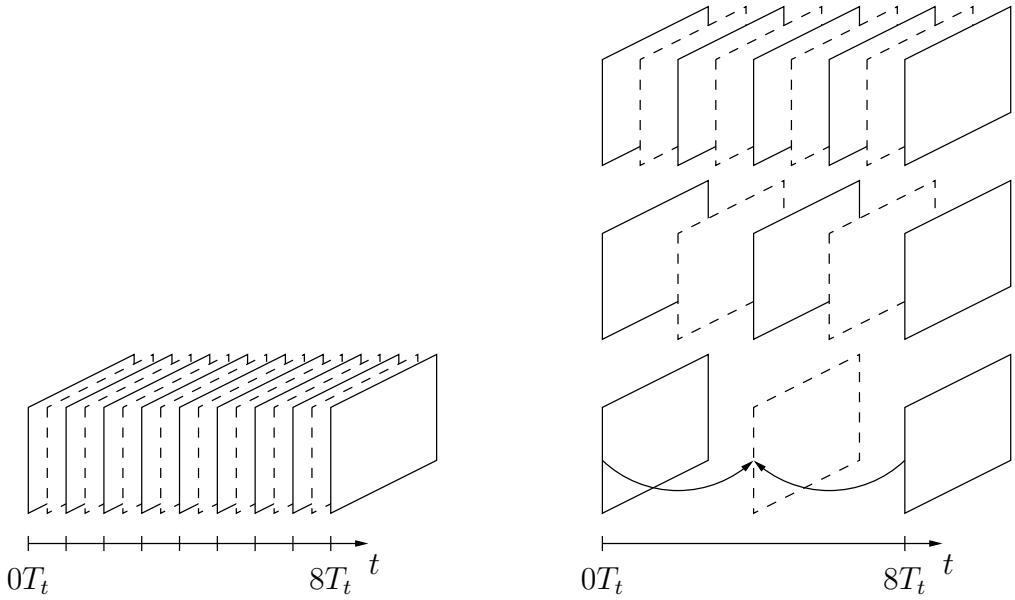


Abbildung 4.16: Zeitlicher Abstand von Referenzbildern bei a) Frame Rate Up Conversion für Holdtype-Bildschirme und bei b) Videocodierung mit hierarchischen B-Bildern.

4.3.1 Funktionsweise von Decoder-side Motion Vector Derivation

In aktuellen Standards werden die einzelnen Makroblöcke innerhalb eines Bildes zeilenweise codiert. Daher liegen in den meisten Fällen bereits decodierte Bildpunkte oberhalb und links des zu codierenden Blockes vor. Unter der Annahme eines örtlich homogenen Bewegungsvektorfeldes kann demnach davon ausgegangen werden, dass diese Randbereiche derselben Bewegung unterliegen wie der zu codierende Block.

Bei DMVD wird diese Annahme verwendet, um den Bewegungsvektor zu schätzen, anstatt diesen explizit zu übertragen. Dazu wird, wie in Abbildung 4.17 gezeigt, eine L-förmige Schablone (*Template*) von bereits decodierten Bildpunkten verwendet, um die Bewegung relativ zu den Referenzbildern innerhalb eines Suchbereichs zu bestimmen. Als Kostenfunktion wird dazu der Betrag der Differenz zwischen einem Bildpunkt im Template und einen Bildpunkt an der aktuellen Position im Referenzbild berechnet und über alle Werte aufsummiert (Summe der absoluten Differenzen, SAD).

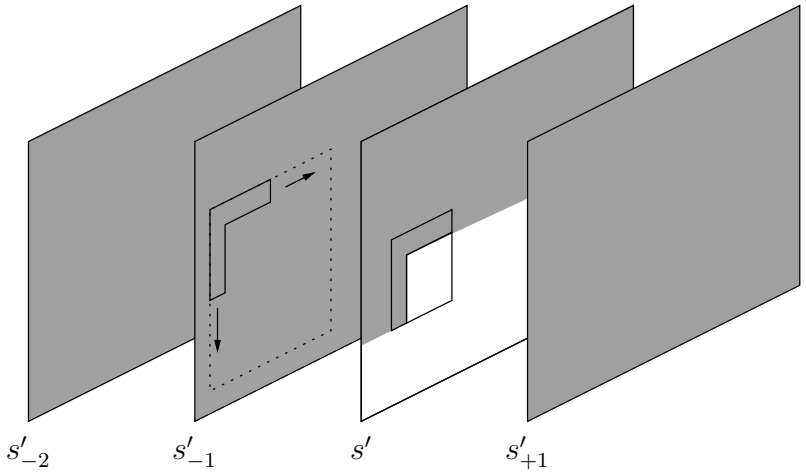


Abbildung 4.17: DMVD Template Matching mit L-förmiger Schablone innerhalb eines begrenzten Suchbereichs.

Der so geschätzte Bewegungsvektor wird am Decoder zur bewegungskompensierten Prädiktion verwendet, ohne dass ein Vektor explizit übertragen werden muss. In [37] wurde gezeigt, dass eine Prädiktion mit mehreren Hypothesen den Codiergewinn weiter steigern kann. Anstatt einen Bewegungsvektor zu schätzen und den entsprechenden Block als Prädiktion zu verwenden, werden mehrere Bewegungsvektoren mit den geringsten Kosten (SAD) ermittelt. Die entsprechenden Blöcke werden zu einem Block zusammengefasst und verbessern so die Prädiktion.

Um die Komplexität durch DMVD zu verringern, wird in aktuellen Implementierungen eine kandidatenbasierte Suche durchgeführt [11]. Anstatt einen sehr großen

Suchbereich abzudecken, werden dabei neun Kandidatvektoren ermittelt und in der Nachbarschaft dieser Vektoren die Bewegungsschätzung durchgeführt.

Am Encoder wird mit Hilfe einer Lagrange-Optimierung [65] für jeden Block entschieden, ob der Vektor am Decoder geschätzt oder im Bitstrom codiert werden soll, da ein falsch geschätzter Bewegungsvektor die Prädiktion und damit den Codiergewinn stark beeinträchtigen kann.

4.3.2 Umsetzung

Wie bereits beschrieben, arbeitet der vorgestellte DSME-Ansatz bildweise. Vor der Codierung des aktuellen Bildes wird das DSME-Bild berechnet und in die Referenzbildliste eingefügt. Anschließend können alle blockbasierten Codieralgorithmen ohne Modifikationen verwendet werden. Wie in Abbildung 4.18 gezeigt, ermöglicht dieses gekapselte Design, den vorgestellten Coder mit den DMVD-Algorithmen zu erweitern. Die Signalisierung von DMVD wurde in der Abbildung zur besseren Übersicht weggelassen. In Abbildung 4.19 ist der entsprechende Encoder gezeigt. Es wird deutlich, dass DSME und DMVD am Encoder sowie Decoder vollkommen isoliert voneinander arbeiten und somit ohne Anpassungen eingesetzt werden können.

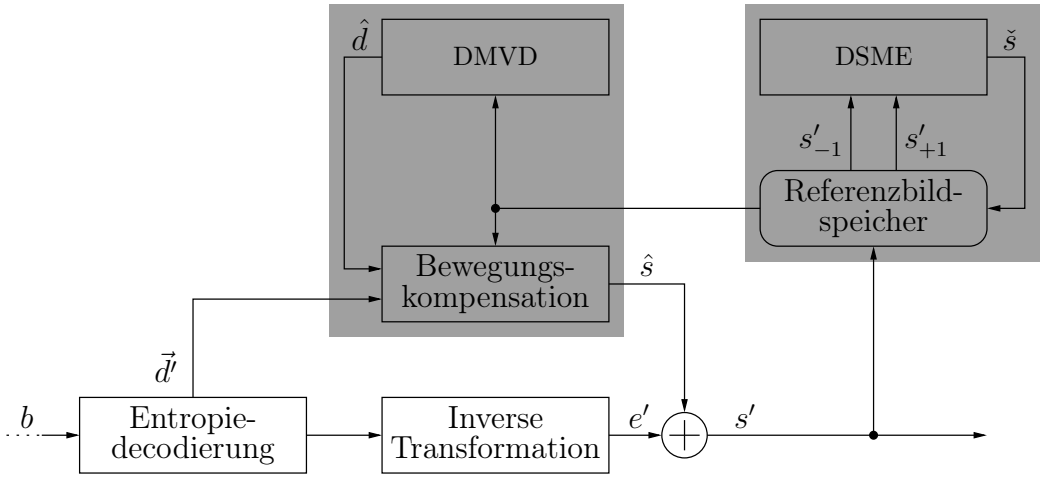


Abbildung 4.18: Vereinfachtes Blockdiagramm eines Decoders mit der Kombination von DSME mit DMVD.

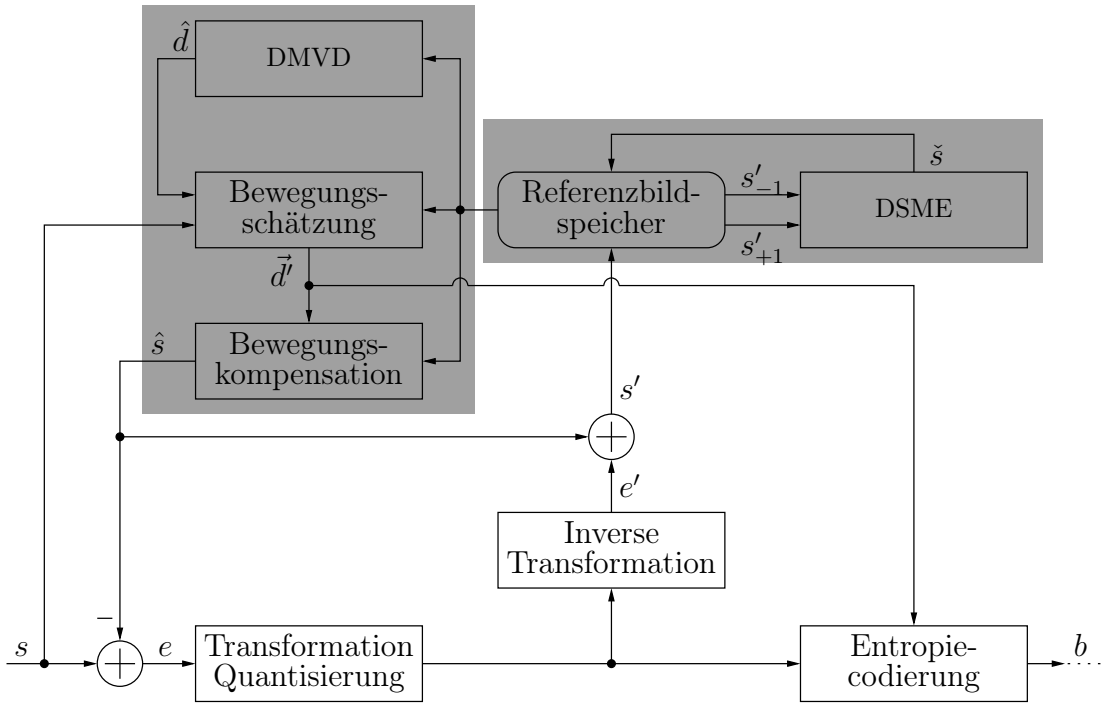


Abbildung 4.19: Vereinfachtes Blockdiagramm eines Encoders mit der Kombination von DSME mit DMVD.

5 Experimentelle Ergebnisse

In diesem Kapitel wird die Effizienz der vorgestellten Videocoderarchitektur mit decoderseitiger Bewegungsschätzung anhand experimenteller Ergebnisse evaluiert. Als Erstes werden in Abschnitt 5.1 die verwendeten Testsequenzen vorgestellt und das objektive Bewertungsmaß erläutert. Ein Vergleich der entwickelten Technik mit Algorithmen zur Zwischenbildberechnung aus der Literatur wird in Abschnitt 5.2 vorgenommen. Außerdem werden die einzelnen Gewinne von Teilen der DSME-Architektur untersucht. Anschließend werden in Abschnitt 5.3 die Kompression des DSME-Coders mit dem internationalen Standard ISO/IEC 14496-10 / ITU-T H.264 (AVC) verglichen. Eine detaillierte Gegenüberstellung von DSME und dem HEVC-Testmodell, welches dem aktuellen Stand der Forschung entspricht, findet in Abschnitt 5.4 statt. Dabei wird neben der Kompressionseffizienz auch die subjektiv wahrgenommene Qualität bewertet und der resultierende Bitstrom genauer untersucht.

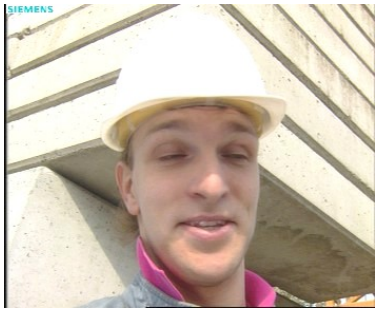
5.1 Evaluationskriterien

In diesem Abschnitt werden die verwendeten Testsequenzen vorgestellt und das Bewertungsmaß zur objektiven Evaluation der Ergebnisse definiert.

5.1.1 Testsequenzen

Zur Evaluation von DSME werden zwei verschiedene Sets von Testsequenzen verwendet. Das erste Testset besteht aus sieben Sequenzen in verschiedenen Auflösungen, die bereits seit vielen Jahren bei der Bewertung von Videocodieralgorithmen Verwendung finden. Das erste Bild jeder dieser Sequenzen ist in Abbildung 5.1 gezeigt.

Das zweite Set besteht aus Sequenzen, die bei JCT-VC zur Evaluierung während der HEVC-Standardisierung verwendet werden [8]. Da besonders hochaufgelöste Sequenzen von der decoderseitigen Bewegungsschätzung profitieren können, wurden alle Sequenzen der Klasse A (2560×1600 Bildpunkte) und der Klasse B (1080p) in das zweite Testset aufgenommen. Abbildung 5.2 und Abbildung 5.3 zeigen einzelne Bilder dieser Sequenzen. Alle Sequenzen wurden progressiv aufgenommen, da das Zeilensprungverfahren [73] aller Voraussicht nach zukünftig nicht mehr berücksichtigt wird. Da diese Sequenzen relativ neu sind und noch keine weite Verbreitung außerhalb der HEVC-Standardisierung haben, werden sie im Folgenden genauer beschrieben.



(a) Foreman, QCIF

(b) Mobile, QCIF/CIF/4CIF

(c) Bus, CIF

(d) City, CIF/4CIF/720p

(e) Flower, CIF/4CIF

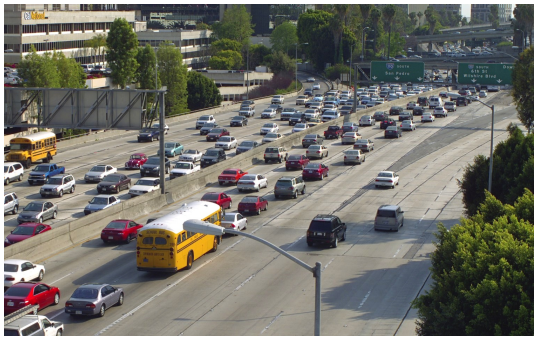
(f) Tempete, CIF

(g) Sheriff, 720p

(h) Spincalendar, 720p

Abbildung 5.1: Set 1: Erstes Bild der verwendeten Testsequenzen. Neben den Sequenznamen sind die verfügbaren Auflösungen angegeben.

(a) PeopleOnStreet



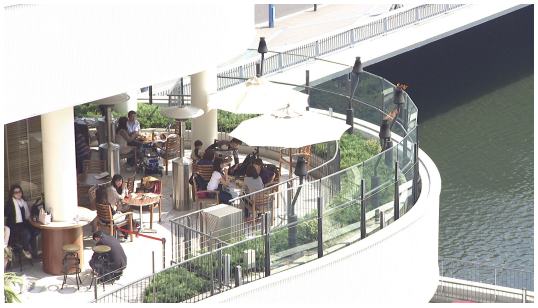
(b) Traffic

(c) NebutaFestival

(d) SteamLocomotiveTrain

Abbildung 5.2: Set 2: Erstes Bild der Testsequenzen der Klasse A aus [8]. Alle Sequenzen haben eine Auflösung von 2560×1600 Bildpunkten.

(a) BasketballDrive



(b) BQTerrace

(c) Cactus

(d) Kimono

(e) ParkScene

Abbildung 5.3: Set 2: Erstes Bild der Testsequenzen der Klasse B aus [8]. Alle Sequenzen haben eine Auflösung von 1920×1080 Bildpunkten (1080p).

Die Sequenzen der Klasse A aus [8] haben eine Auflösung von 2560×1600 Bildpunkten. In PeopleOnStreet ist ein großer Fußgängerüberweg zu sehen, auf dem viele Personen in unterschiedliche Richtungen gehen. Besonders die sich bewegenden Schlagschatten der Personen durch die tiefstehende Sonne, sind eine Herausforderung bei der Codierung. Diese Sequenz zeigt lediglich einen Ausschnitt und wurde ursprünglich mit einer Auflösung von 3840×2160 Bildpunkten durch eine statische Kamera aufgenommen. Das Ausgangsmaterial der Traffic-Sequenz hat die Auflösung von 4096×2048 Bildpunkten und wurde ebenfalls mit einer statischen Kamera erstellt. Zu sehen ist eine vielbefahrene Autobahn. Die NebutaFestival-Sequenz zeigt eine bunte Gestalt, die bewegt wird. Hier hatte das Ausgangsmaterial ein Auflösung von 7680×4320 Bildpunkten. Die Besonderheit dieser Sequenz ist der erweiterte Dynamikbereich von 10 bit statt der üblichen 8 bit pro Bildpunkt. Jedoch hat diese Sequenz einen sehr hohen Rauschanteil. SteamLocomotiveTrain wurde mit der gleichen Kamera wie NebutaFestival aufgenommen und besitzt daher ebenfalls den erweiterten Dynamikbereich. Sie zeigt eine Dampflokomotive und beinhaltet sehr große Bewegungen von über 150 pel zwischen den einzelnen Bildern [62]. Außerdem stellt der aufsteigende Dampf aufgrund der Transparenz und Deformation eine weitere Herausforderung bei der Codierung dar.

Die in [8] verwendeten Sequenzen der Klasse B haben alle eine HDTV-Auflösung von 1080p (1920×1080 Bildpunkte). Die BasketballDrive-Sequenz zeigt ein Basketballspiel und weist, aufgrund der schnellen Kamerabewegung, starke Bewegungsunschärfe auf. Ein sehr langsamer Kameraschwenk ist in BQTerrace zu sehen. Jedoch können die schimmernde Wasseroberfläche und das Flimmern der Luft durch Fackeln bei der Codierung Probleme bereiten. Die Sequenz Cactus zeigt eine künstliche Szene rotierender und schwingender Objekte. Durch den experimentellen Aufbau ist diese Sequenz gut ausgeleuchtet und besitzt eine hohe Schärfe. Kimono ist die einzige der Testsequenzen, die einen Szenenschnitt enthält. In der ersten Hälfte geht eine Person an Bäumen entlang, welche durch die Tiefe unscharf dargestellt werden. Nach dem Schnitt geht die Person auf ein Haus zu. Eventuelle Codierartefakte machen sich besonders auf dem gleichmäßig strukturierten Dach des Hauses bemerkbar. In ParkScene sind mehrere Fahrradfahrer zu sehen, welche sich schnell durch das Bild bewegen. Auch hier kommt es zu Bewegungsunschärfe.

5.1.2 Bewertungsmaß

Zur objektiven Qualitätsbewertung der decoderseitigen Bewegungsschätzung wird das Spitzensignal-Rausch-Verhältnis (*Peak Signal to Noise Ratio*, PSNR) der Luminanzkomponente berechnet. Bei einer Sequenz mit 8 bit pro Abtastwert ist der größtmögliche Wert $2^8 - 1 = 255$. Für diesen Fall ist der PSNR für ein Bild der

Sequenz definiert als

$$PSNR_{n_t}^{(8\,\mathrm{bit})} = 10\log_{10}\left(\frac{255^2}{MSD_{n_t}}\right), \tag{5.1}$$

wobei die mittlere quadratische Differenz *MSD* definiert ist als

$$MSD_{n_t} = \frac{1}{MN}\sum_{n_x=0}^{M-1}\sum_{n_y=0}^{N-1}\left\|s(n_x,n_y,n_t) - s'(n_x,n_y,n_t)\right\|^2. \tag{5.2}$$

Das PSNR einer gesamten Sequenz wird aus dem Mittelwert der einzelnen PSNRs berechnet:

$$PSNR = \frac{1}{N_t}\sum_{n_t=0}^{N-t-1} PSNR_{n_t}. \tag{5.3}$$

Soll das Spitzensignal-Rausch-Verhältnis einer Sequenz mit 10 bit pro Bildpunkt berechnet werden, muss der Spitzenwert angepasst werden. Innerhalb JCT-VC hat man sich darauf geeinigt, nicht $2^{10}-1 = 1023$ zu verwenden, sondern sich an die ITU-T-Empfehlung [30] zu halten. Diese besagt, dass der Spitzenwert für 10 bit durch hinzufügen von zwei niederwertigen Bits an den 8 bit-Spitzenwert [64] zu bestimmen ist, was einer Multiplikation mit 2^2 entspricht:

$$\begin{aligned} PSNR_{n_t}^{(10\,\mathrm{bit})} &= 10\log_{10}\left(\frac{(255\cdot 2^2)^2}{MSD_{n_t}}\right) \\ &= 10\log_{10}\left(\frac{1020^2}{MSD_{n_t}}\right). \end{aligned} \tag{5.4}$$

Das PSNR wird in Abschnitt 5.2 verwendet, um das DSME-Bild mit einer FRUC-Interpolation zu vergleichen. Um die gesamte DSME-Architektur mit den Referenzverfahren in Abschnitt 5.3 und 5.4 vergleichen zu können, wird die sogenannte operative *Rate-Distortion* (RD) Kurve bestimmt, bei der die Verzerrung (PSNR) der Datenrate gegenübergestellt wird. Dazu wird eine Sequenz in verschiedenen Arbeitspunkten codiert und die einzelnen Rate-PSNR-Kombinationen als Kurve dargestellt. Die Einstellung der verschiedenen Arbeitspunkte erfolgt mit Hilfe des Quantisierungsparameters, welcher ausschlaggebend für die Qualität der codierten Sequenz ist. Mit Hilfe der RD-Kurve kann die Erhöhung des PSNR – also die Verbesserung der Qualität – bei gleicher Datenrate bestimmt werden. Auf die gleiche Weise ist es möglich, die Reduktion der Datenrate bei gleicher Qualität zu ermitteln.

Zur Berechnung der mittleren Reduktion der Datenrate über den gesamten Qualitätsbereich werden, wie in [6] vorgeschlagen, die RD-Kurven durch eine kubische Funktion approximiert und anschließend die Differenz gebildet. Dieses sogenannte Bjøntegaard-Delta (BD) wird auch innerhalb der Standardisierungstätigkeiten des JCT-VC zur Evaluierung der eingereichten Beiträge verwendet.

5.2 Bewertung der bewegungskompensierten Interpolation

Wie bereits in Kapitel 4 erläutert, wird die Effizienz des DSME-Ansatzes wesentlich durch die bewegungskompensierte Interpolation am Decoder beeinflusst. Daher wird die Effizienz des vorgestellten Algorithmus evaluiert und mit einer Interpolationsmethode, welche dem Stand der Technik entspricht, verglichen. Für diesen Vergleich wurde die in Abschnitt 4.2.3 kurz vorgestellte STAR-Methode [75] gewählt, da mit dieser bessere Ergebnisse erzielt werden können als mit 3DRS [16] und AOBMC [12].

Zur Evaluation wurde die 13 Testsequenzen des ersten Testsets aus Abschnitt 5.1.1 genutzt, welche unterschiedliche Charakteristiken, Bildwiederholraten und Auflösungen besitzen. Jedes zweite Bild wird aus der jeweiligen Sequenz entfernt und mit Hilfe von STAR und der hier vorgestellten hierarchischen Bewegungsschätzung interpoliert. Anschließend werden die interpolierten Bilder mit den entfernten Originalbildern verglichen. Die Ergebnisse sind in Tabelle 5.1 aufgeführt. Bis auf drei Ausnahmen erzielt die DSME-Interpolation bessere oder sehr ähnliche Ergebnisse.

Tabelle 5.1: Mittlerer PSNR für die STAR-Interpolation [75] und die vorgestellte hierarchische Interpolation für verschiedene Testsequenzen. Unterschiede von über 0,25 dB sind hervorgehoben. Zusätzlich sind die Ergebnisse der hierarchischen Interpolation unter Verwendung des mittleren quadratischen Fehlers (MSD) als Optimierungskriterium angegeben.

Auflösung	**Sequenz**	**STAR**	**hier. Int. (MAD)**	**hier. Int. (MSD)**
QCIF	Foreman	**39,67 dB**	37,60 dB	37,67 dB
	Mobile	36,19 dB	**36,76 dB**	36,74 dB
CIF	Bus	27,27 dB	**29,54 dB**	29,51 dB
	City	34,83 dB	**35,46 dB**	35,49 dB
	Flower	**33,45 dB**	32,59 dB	33,11 dB
	Mobile	29,49 dB	**31,87 dB**	31,89 dB
	Tempete	30,86 dB	30,96 dB	30,94 dB
4CIF	City	**30,13 dB**	28,91 dB	29,14 dB
	Flower	28,70 dB	**33,56 dB**	33,58 dB
	Mobile	26,75 dB	**31,96 dB**	31,97 dB
720p	City	31,66 dB	31,57 dB	31,73 dB
	Sheriff	38,08 dB	37,96 dB	37,95 dB
	Spincalendar	29,51 dB	**35,20 dB**	35,32 dB
Mittelwert		32,05 dB	**33,38 dB**	33,46 dB

Ebenso interessant wie der Vergleich mit einem in der Literatur beschriebenen Verfahren, ist die Auswertung, welche Methoden hauptsächlich für die guten Ergebnisse verantwortlich sind. Dazu wurden zwei weitere Tests mit der hierarchischen Interpolation durchgeführt, bei denen die minimale Blockgröße auf 4×4 Bildpunkte angehoben beziehungsweise das Latchingverfahren deaktiviert wurde. Die PSNR-Differenzen zwischen der unveränderten Interpolation und diesen beiden Tests sind in Tabelle 5.2 angegeben.

Tabelle 5.2: PSNR-Differenz der hierarchischen Interpolation durch Erhöhung der minimalen Blockgröße von 1 pel auf 4 pel und durch Deaktivierung des Latchingverfahrens.

Auflösung	**Sequenz**	**4×4 Blöcke**	**kein Latching**
QCIF	Foreman	-0,26 dB	-1,30 dB
	Mobile	-0,15 dB	-5,23 dB
CIF	Bus	0,01 dB	-2,40 dB
	City	-0,19 dB	-1,05 dB
	Flower	-0,55 dB	-1,52 dB
	Mobile	-0,17 dB	-3,38 dB
	Tempete	-0,03 dB	-0,83 dB
4CIF	City	-0,15 dB	-0,53 dB
	Flower	-0,15 dB	-1,43 dB
	Mobile	-0,01 dB	-1,68 dB
720p	City	0,12 dB	-0,64 dB
	Sheriff	-0,03 dB	-1,53 dB
	Spincalendar	0,12 dB	-4,80 dB
Mittelwert		-0,11 dB	-2,02 dB

Wird kein dichtes Bewegungsvektorfeld berechnet und stattdessen die minimale Blockgröße auf 4×4 Bildpunkte eingeschränkt, verringert sich die erzielte Interpolationsqualität nur leicht. Drei Sequenzen bilden Ausnahmen, für welche die Qualität durch die eingeschränkte Blockgröße leicht ansteigt. Die Erhöhung der Qualität im Mittel um 0,11 dB durch Interpolation mit einem dichten Bewegungsvektorfeld ist zwar sehr gering, jedoch ist auch der zusätzliche Aufwand nicht sehr hoch. Aufgrund des Latchings müssen für jeden Block lediglich neun Kandidaten überprüft und keine volle Bewegungsschätzung durchgeführt werden.

Die Verbesserung der Qualität durch das vorgestellte Vektorlatching sind deutlich höher. Durch die geringe Blockgröße und die somit sehr wenigen Texturinformationen, schätzt eine volle Bewegungssuche sehr häufig falsche Bewegungsvektoren.

Somit erhöht dieses Verfahren die Interpolationsqualität und verringert zusätzlich den Berechnungsaufwand bei der Bewegungsschätzung.

5.3 Vergleich mit dem AVC-Standard

Um einen objektiven Vergleich von DSME mit einem aktuellen Videocodierstandard zu bekommen, wurde die Technik in die AVC-Referenzsoftware JM 16.2 [34] implementiert. Alle Codierparameter sind entsprechend dem gemeinsamen *Call for Proposals* [28] der ISO/IEC JTC1/SC29/WG11 (MPEG) und der ITU-T SG16 Q.6 (VCEG) gesetzt worden. Im wesentlichen beinhaltet dies die Codierung im sogenannten High Profile mit hierarchischen B-Bildern und einem I-Bild pro Sekunde, für einen verzögerungsarmen Direktzugriff. Die Anzahl der Referenzbilder ist auf zwei Bilder pro Referenzbildliste beschränkt. Beim Hinzufügen des DSME-Bildes erhöht sich die Zahl somit auf drei Bilder. Der Quantisierungsparameter wurde für jede Sequenz angepasst, um vergleichbare Datenraten zu erzielen.

Beispielhaft sind zwei RD-Kurven in Abbildung 5.4 und 5.5 gezeigt. Da DSME nur bei B-Bildern angewendet wird und andere Bilder nicht weiter beeinflusst werden, ist lediglich die Datenrate zur Codierung dieser B-Bilder gezeigt. Es ist zu erkennen, dass DSME insbesondere bei geringeren Dateraten hohe Gewinne erzielt. Bei höheren Datenraten verringert sich der Gewinn gegenüber JM. Dies ist darauf zurückzuführen, dass sich der relative Anteil der Rate der Bewegungsvektoren – wie in Abbildung 3.1 gezeigt – verringert, und somit weniger Einfluss auf die Gesamtdatenrate hat.

Die durchschnittliche Qualitätssteigerung und Datenratenreduzierung des kompletten zweiten Testsets ist in Tabelle 5.3 gezeigt. Lediglich die Sequenzen mit dem erweiterten Wertebereich von 10 bit pro Bildpunkt wurden ausgelassen, da JM diese Bittiefe nicht unterstützt.

DSME arbeitet besonders effizient bei der PeopleOnStreet-Sequenz. Die einzelnen Personen bewegen sich gleichmäßig, sodass die Bewegung sehr präzise geschätzt werden kann. Dagegen ist die Schätzung der Bewegung der Wasseroberfläche in der Sequenz BQTerrace sehr kompliziert, wodurch geringere Gewinne erzielt werden.

Der verwendete AVC-Standard wurde bereits vor mehreren Jahren entwickelt. In der Zwischenzeit wurden effizientere Algorithmen zur Codierung von Videosequenzen erforscht. Diese neuen Verfahren können sich positiv oder auch negativ auf die Effizienz von DSME auswirken. Daher wird im folgenden Abschnitt DSME mit dem HEVC-Testmodell (HM) verglichen, welches dem aktuellen Stand der Forschung entspricht. Dort wird neben der Kompressionseffizienz auch die subjektive Qualität beurteilt.

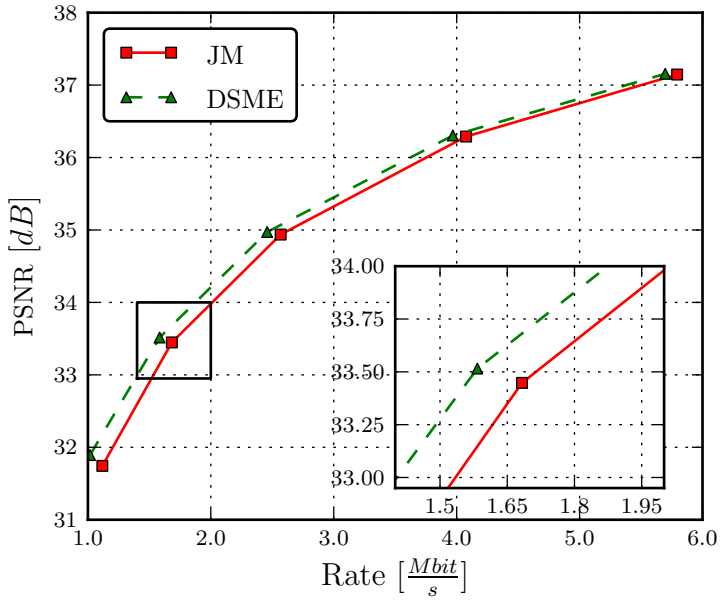


Abbildung 5.4: Gemessener PSNR der B-Bilder in Abhängigkeit der Datenrate für die Sequenz BasketballDrive bei Codierung mit JM und DSME.

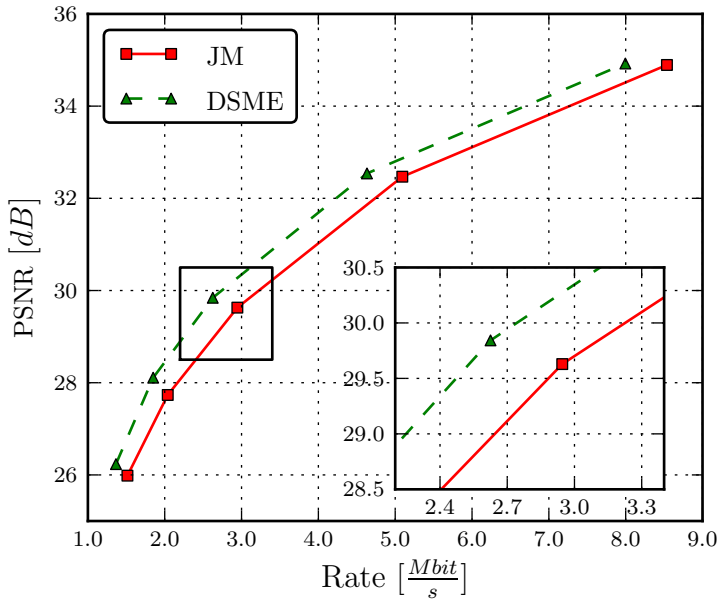


Abbildung 5.5: Gemessener PSNR der B-Bilder in Abhängigkeit der Datenrate für die Sequenz PeopleOnStreet bei Codierung mit JM und DSME.

Tabelle 5.3: Bjøntegaard-Delta des PSNR und der Rate zwischen JM und DSME für verschiedene Testsequenzen.

	B-Bilder		Gesamt	
Sequenz	**BD-PSNR**	**BD-Rate**	**BD-PSNR**	**BD-Rate**
BasketballDrive	0,24 dB	-6,95 %	0,15 dB	-4,29 %
BQTerrace	0,15 dB	-8,94 %	0,07 dB	-3,76 %
Cactus	0,33 dB	-10,71 %	0,17 dB	-5,12 %
Kimono	0,25 dB	-7,33 %	0,16 dB	-4,20 %
ParkScene	0,41 dB	-12,24 %	0,21 dB	-5,51 %
PeopleOnStreet	0,78 dB	-13,98 %	0,51 dB	-9,25 %
Traffic	0,56 dB	-16,76 %	0,24 dB	-5,96 %
Mittelwert	0,39 dB	-10,99 %	0,22 dB	-5,44 %

5.4 Vergleich mit dem HEVC-Testmodell

In diesem Abschnitt soll DSME mit dem aktuellen Stand der Forschung verglichen werden. Dazu wird als Referenz das HEVC-Testmodell (HM) verwendet, welches von JCT-VC zur Entwicklung eines Nachfolgers des AVC-Standards verwendet wird. Um einen direkten Vergleich zu erhalten, wurde DSME in HM 2.0 implementiert.

Da DSME nur bei B-Bildern verwendet wird und auf hohe Codiereffizienz abzielt, wurden die Codierparameter wie beim sogenannten *Random Access, High Efficiency* Test aus [8] gewählt. Wie schon bei dem Vergleich mit AVC in Abschnitt 5.3 werden auch hier hierarchische B-Bilder verwendet und auch die Anzahl der Referenzbilder ist zwei. Im Gegensatz zu der Konfiguration in [28] werden die Quantisierungsparameter jedoch für alle Testsequenzen unabhängig von der Datenrate, wie in [8] vorgeschlagen, auf die Werte 22, 27, 32 und 37 gesetzt.

Wie in Abbildung 5.6 zu erkennen, ist die subjektive Qualität für die Referenz und DSME nahezu identisch. Auch bei genauer Betrachtung ist keine Verschlechterung zu erkennen, obwohl die mit Hilfe von DSME codierte Sequenz 7 % weniger Datenrate benötigt. Auch für die weiteren Testsequenzen ist kein subjektiver Unterschied festzustellen. Ein objektiver Vergleich der Kompressionsgewinne wird im folgenden Abschnitt durchgeführt.

5.4.1 Kompressionseffizienz von DSME

Es ist zu erwarten, dass der Kompressionsgewinn von DSME in Verbindung mit HM geringer ausfallen wird als die Gewinne gegenüber AVC aus Abschnitt 5.3, da DSME durch manche neuen Tools beeinflusst wird und sich somit die Gewinne von HM und

(a) HM-codiert bei 8,77 $^{\text{Mbit}}/_{\text{s}}$

(b) DSME-codiert bei 8,18 $^{\text{Mbit}}/_{\text{s}}$

Abbildung 5.6: Subjektiver Vergleich eines Ausschnitts der decodierten PeopleOnStreet-Sequenzen bei unterschiedlichen Datenraten.

DSME gegenüber AVC nicht voll aufaddieren.

In den Abbildungen 5.7 und 5.8 sind die operativen RD-Kurven für die Sequenzen BasketballDrive und PeopleOnStreet gezeigt. Wie erwartet, ist der Kompressionsgewinn zwischen HM und DSME aufgrund neuer Techniken gesunken. Jedoch ist insbesondere für PeopleOnStreet weiterhin eine deutliche Reduktion der Datenrate zu erkennen.

Zusätzlich zu dem DSME-Algorithmus aus Abschnitt 4.1 wurde außerdem getestet, wie sich DSME ohne die Kompensation beschleunigter Bewegung verhält (DSME_0). Dazu wurde die konventionelle Bewegungsschätzung am Encoder bei Verwendung des DSME-Bildes als Referenz deaktiviert. Wie bereits in Abschnitt 3.1.1 angesprochen, führt DSME_0 zu einer leichten Erhöhung der Datenrate (Abbildungen 5.7 und 5.8).

Die mittlere Reduktion der Datenraten für alle Testsequenzen sind in Tabelle 5.4 aufgelistet. Im Gegensatz zu den in [38] vorgestellten Ergebnissen, wurde hier die

Tabelle 5.4: Bjøntegaard-Delta der Gesamtdatenrate zwischen HM und DSME sowie DSME ohne Versatzkompensation (DSME_0) für verschiedene Testsequenzen.

	B-Bilder		**Gesamt**	
Sequenz	**DSME**	**DSME_0**	**DSME**	**DSME_0**
BasketballDrive	-5,19 %	-4,29 %	-2,72 %	-2,25 %
BQTerrace	-7,25 %	-7,05 %	-1,26 %	-0,98 %
Cactus	-11,56 %	-11,41 %	-4,52 %	-4,47 %
Kimono	-8,13 %	-6,78 %	-3,81 %	-3,22 %
ParkScene	-11,56 %	-11,32 %	-3,26 %	-3,18 %
PeopleOnStreet	-11,30 %	-10,63 %	-6,64 %	-6,18 %
Traffic	-25,95 %	-25,93 %	-5,98 %	-5,95 %
NebutaFestival	-2,00 %	-1,85 %	-0,29 %	-0,23 %
SteamLocomotiveTrain	-1,85 %	-1,65 %	-0,76 %	-0,71 %
Mittelwert	-9,42 %	-8,99 %	-3,25 %	-3,02 %

Bewegungsschätzung zur Interpolation des DSME-Bildes mit 1/4-pel-Genauigkeit bestimmt. Wie bei der DSME-Implementation in JM, sind auch hier die größten Gewinne bei den Sequenzen PeopleOnStreet und Traffic zu erreichen. Die Datenratenreduktion der beiden zusätzlichen Sequenzen sind jedoch sehr gering. Bei NebutaFestival liegt dies an der rein globalen Bewegung, welche effizient mit dem Skipmodus kompensiert werden kann. Der Rauch und die Bewegungsunschärfe während der Großaufnahme des Zuges bei der SteamLocomotiveTrain-Sequenz erlauben lediglich eine ungenaue Schätzung der Bewegung, die zu der geringen Reduktion der Datenrate

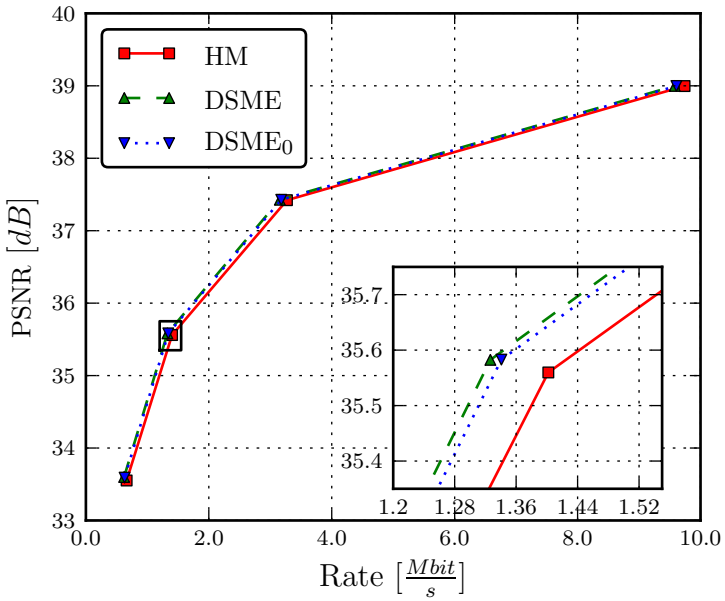


Abbildung 5.7: Gemessener PSNR der B-Bilder in Abhängigkeit der Datenrate für die Sequenz BasketballDrive bei Codierung mit HM, DSME und DSME ohne Versatzkompensation ($DSME_0$).

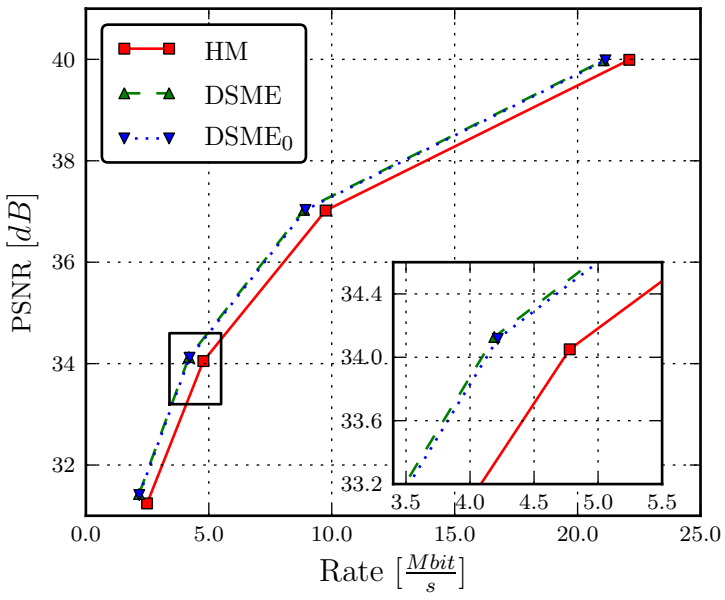


Abbildung 5.8: Gemessener PSNR der B-Bilder in Abhängigkeit der Datenrate für die Sequenz PeopleOnStreet bei Codierung mit HM, DSME und DSME ohne Versatzkompensation ($DSME_0$).

führt.

Insbesondere für die Sequenzen BasketballDrive, BQTerrace, Kimono und PeopleOnStreet ist die Reduktion der Datenrate durch die zusätzliche Versatzkompensation (DSME) gegenüber nicht kompensiertem Versatz (DSME_0) in Tabelle 5.4 gut abzulesen. Während bei der BasketballDrive durch schnelle Bewegungen zu kompensierende Beschleunigung zu erwarten ist, sind bei den anderen Sequenzen schwache Texturen der Hauptgrund. So bereiten bei der BQTerrace-Sequenz die Wasseroberfläche und bei der Kimono-Sequenz die unscharfen Bäume im Hintergrund Probleme. Als problematisch bei der Bewegungsschätzung erweisen sich auch Schlagschatten wie sie in der Sequenz PeopleOnStreet vorkommen. Es kommt am Decoder zu ungenauer Bewegungsschätzung, welche mit Hilfe der zusätzlich übertragenen Bewegungsvektoren kompensiert wird.

Durch das hinzugefügte DSME-Bild besitzt der DSME-Decoder ein Bild mehr in der Referenzbildliste als das Referenzverfahren. Untersuchungen mit dem HM-Referenzcoder und einer – von [8] abweichenden – Konfiguration mit drei anstatt zwei Referenzbildern haben jedoch gezeigt, dass der Gewinn durch die erweiterte Referenzbildliste minimal ist. So wird die Datenrate zwischen HM mit drei und zwei Referenzbildern für die BasketballDrive-Sequenz im Mittel lediglich um 0,15 % gesenkt. Für die Sequenz PeopleOnStreet liegt die Reduktion mit 0,05 % nochmals deutlich niedriger. Demgegenüber stehen die Gewinne durch das zusätzliche DSME-Bild von 2,72 % beziehungsweise 6,64 %. Es wird deutlich, dass nicht das zusätzliche Referenzbild für die hohe Effizienz des DSME-Coders verantwortlich ist, sondern die decoderseitige Bewegungsschätzung.

5.4.2 Analyse des codierten Bitstroms

Zum besseren Verständnis der Gewinne von DSME, soll in diesem Abschnitt das codierte Signal analysiert werden. Die folgenden Auswertungen beziehen sich immer auf die B-Bilder. I- und P-Bilder werden nicht berücksichtigt, da diese nicht von DSME beeinflusst werden. Als Quantisierungsparameter wurde 32 gewählt, da die resultierende Codierung in etwa Rundfunkqualität entspricht. Eine detailliertere Analyse wird anhand der Sequenzen BasketballDrive und PeopleOnStreet durchgeführt, da diese sehr unterschiedliche Charakteristiken aufweisen.

In Abbildung 5.9 und 5.10 sind die verschiedenen Prädiktionsmodi für diese Sequenzen visualisiert. Die Intraprädiktion wird bei beiden Sequenzen nur sehr selten verwendet. Für die BasketballDrive-Sequenz ist zu erkennen, dass DSME besonders häufig in den bewegten Bereichen verwendet wird. Insgesamt wird etwa ein Viertel des Bildes mit Hilfe des DSME-Bildes prädiziert. Eine genaue Auflistung der prozentualen Nutzung des DSME-Bildes für alle Testsequenzen ist in Tabelle 5.5 gegeben. Bei PeopleOnStreet liegt die Verwendung des DSME-Bildes zur Prädiktion im Mittel bei 59 %. Dieser Trend ist auch in Abbildung 5.10 zu sehen, wo DSME einen großen Teil des Bildes abdeckt.

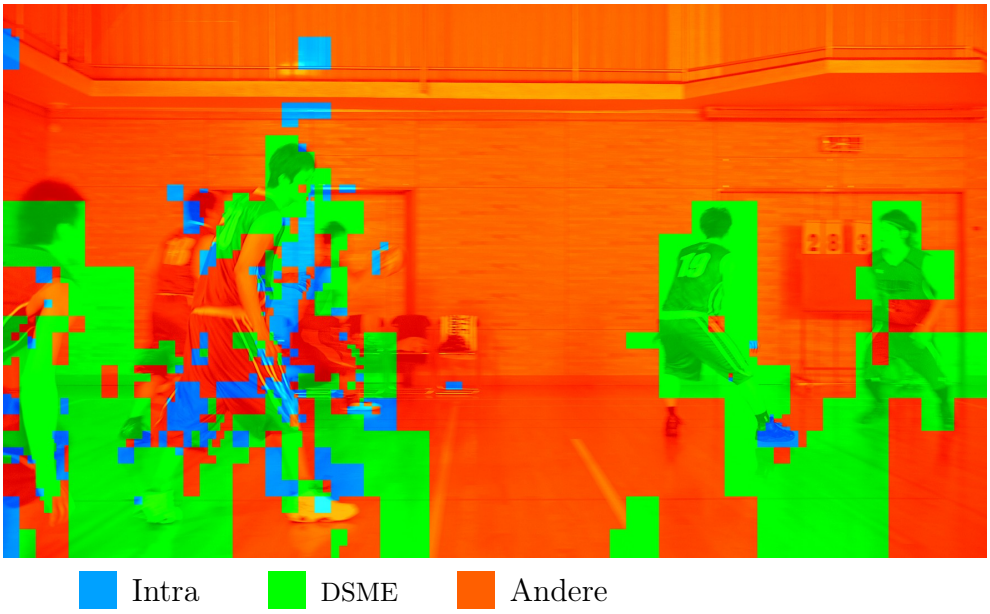


Abbildung 5.9: Visualisierung der verwendeten Prädiktionsmethoden (Intra- und Interprädiktion mit DSME-Bild oder anderen Referenzbildern) für ein Bild der BasketballDrive-Sequenz.

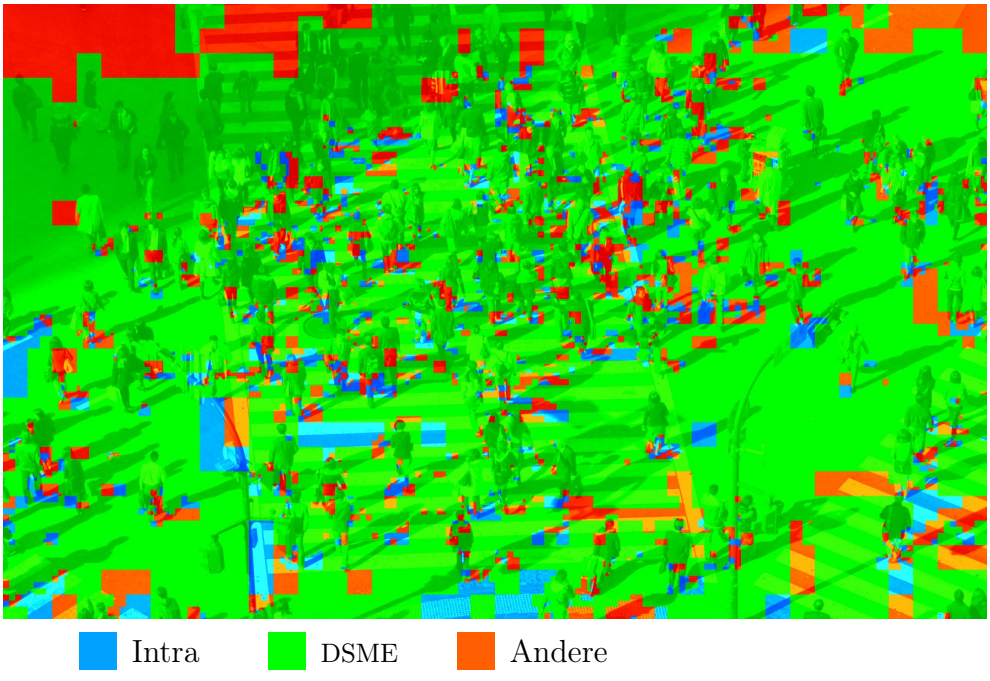


Abbildung 5.10: Visualisierung der verwendeten Prädiktionensmethoden (Intra- und Interprädiktion mit DSME-Bild oder anderen Referenzbildern) für ein Bild der PeopleOnStreet-Sequenz.

Tabelle 5.5: Prozentsatz der Blöcke, die zur Prädiktion das DSME-Bild verwenden.

Sequenz	**DSME Nutzung**
BasketballDrive	23,3 %
BQTerrace	17,0 %
Cactus	39,5 %
Kimono	48,7 %
ParkScene	51,9 %
PeopleOnStreet	58,7 %
Traffic	51,7 %
NebutaFestival	11,1 %
SteamLocomotiveTrain	17,4 %

Die Häufigkeit für die Verwendung des DSME-Bildes hängt jedoch auch von dem zeitlichen Abstand der zur Interpolation verwendeten Bilder ab. Ist der Abstand groß, ist eine genaue Schätzung schwieriger, da größere Bewegungen auftreten können und weniger Korrelationen zwischen den Bildern zu erwarten sind. Somit hat das DSME-Bild eine geringere Qualität und wird seltener als Referenz verwendet. Dagegen ist bei geringem Abstand eine sehr genaue Interpolation möglich. Wie in Abbildung 4.16b abgebildet, ergeben sich für die hier verwendete 3-stufige, hierarchische Codierung der B-Bilder auch drei unterschiedliche Abstände der zur DSME-Bild-Interpolation verwendeten Referenzbilder ($2T_t$, $4T_t$, $8T_t$). Die relative Häufigkeit der Interprädiktion mit Hilfe des DSME-Bildes, der Interprädiktion mit anderen Referenzbildern und der Intraprädiktion ist in Abbildung 5.11 für alle neun Testsequenzen gezeigt.

Es ist zu erkennen, dass die Häufigkeit für DSME mit steigendem Abstand sinkt. Lediglich bei der ParkScene-Sequenz steigt die Häufigkeit minimal an. Wie erwartet, wird die Intraprädiktion bei steigendem Abstand häufiger verwendet, da die Korrelationen zwischen den Bildern abnimmt und somit eine Interprädiktion – egal ob mit DSME-Bild oder anderen Referenzbildern – ineffizienter wird.

In den Abbildungen 5.9 und 5.10 entsteht jedoch der Eindruck, dass die Anzahl kleiner Blöcke bei PeopleOnStreet deutlich höher ist als bei BasketballDrive. Dies lässt sich anhand der Abbildungen 5.12 und 5.13 bestätigen, welche die Häufigkeiten der einzelnen Blockgrößen für den unmodifizierten HM-Encoder und den Encoder mit DSME zeigen. Während bei BasketballDrive 70 % des Bildes die maximale Blockgröße besitzen, sind die unterschiedlichen Blockgrößen bei PeopleOnStreet deutlich breiter gestreut.

Es ist außerdem zu erkennen, dass bei Verwendung von DSME feine Strukturen

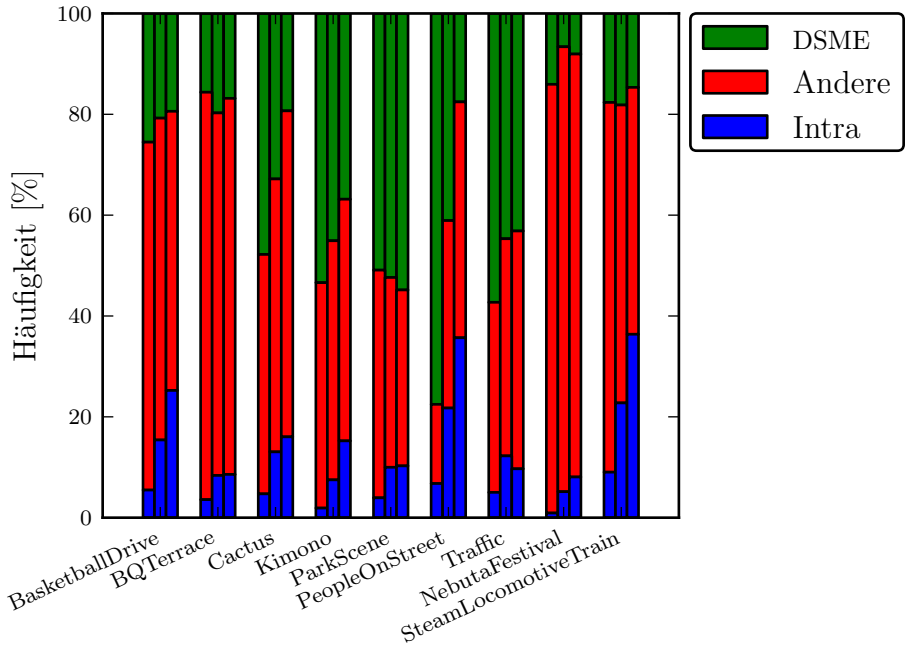


Abbildung 5.11: Relative Häufigkeit der unterschiedlichen Prädiktionsmodi getrennt nach den drei Hierarchiestufen der B-Bild-Codierung für alle neun Testsequenzen. Die jeweils linke Säule repräsentiert dabei die letzte Hierarchiestufe mit dem geringsten zeitlichen Abstand.

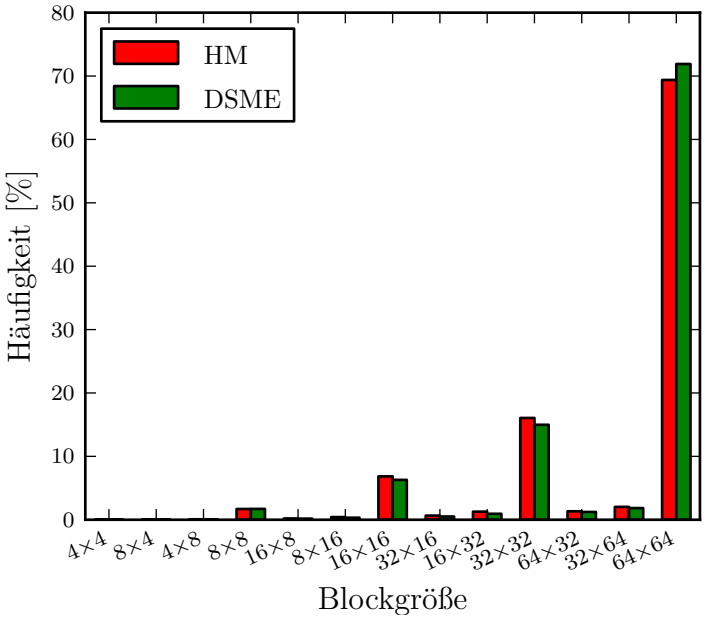


Abbildung 5.12: Relative Häufigkeit der verwendeten Blockgrößen bei Codierung der BasketballDrive-Sequenz mit Hilfe von HM und DSME.

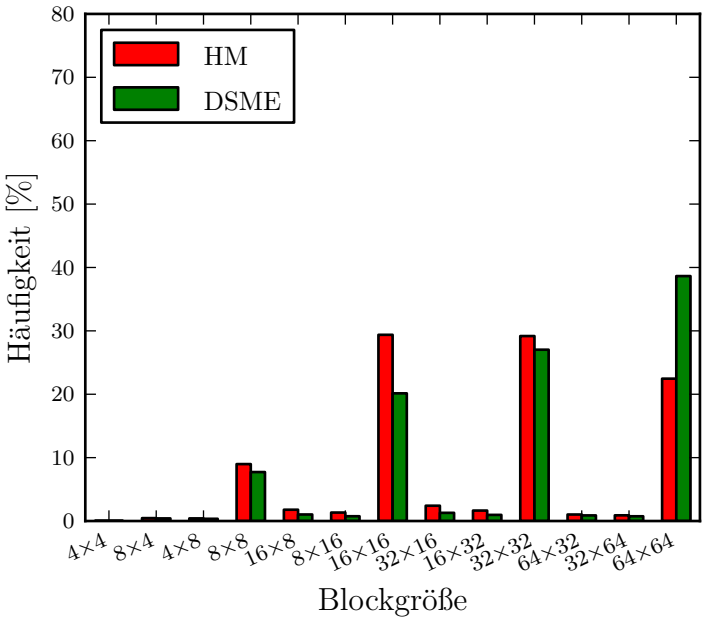


Abbildung 5.13: Relative Häufigkeit der verwendeten Blockgrößen bei Codierung der PeopleOnStreet-Sequenz mit Hilfe von HM und DSME.

gut kompensiert und somit größere Blöcke bei der Prädiktion genutzt werden können. Während bei der Codierung der PeopleOnStreet-Sequenz mit Hilfe von HM etwa 30 % mit 16×16-Blöcken prädiziert werden, sinkt der Anteil bei DSME um 1/3. Dementsprechend steigt der Anteil der maximalen Blockgröße von etwa 20 % auf knapp 40 %. Als Konsequenz sinkt die Anzahl der zu codierenden Bewegungsvektoren, wodurch die hohen Gewinne bei dieser Sequenz zu erklären sind.

Ein direkter Vergleich der verwendeten Blockgrößen mit den theoretischen Ergebnissen aus Abschnitt 3.2.2 ist jedoch nicht möglich. Das analytische Modell berücksichtigt nicht variable Blockgrößen, wie sie in HEVC verwendet werden. Daher tendiert das analytische Modell zu kleineren Blöcken um auch an Objektkanten genau prädizieren zu können. Jedoch lässt sich bereits aus der normierten Auftretenswahrscheinlichkeit unterschiedlicher Objekte von lediglich $\frac{P_{B_0}}{B_0} = 5{,}6\%$ ablesen, dass es viele große Objekte gibt, welche durch große Blöcke prädiziert werden können.

Die Reduktion der Datenrate ist jedoch nicht nur auf die Verringerung der benötigten Rate zur Codierung der Bewegungsvektoren zurückzuführen. DSME ermöglicht durch die Schätzung eines dichten Bewegungsvektorfeldes auch eine genauere Prädiktion an Objektkanten. Diese Schätzung verringert wiederum das zu codierende Residuum. In Tabelle 5.6 ist die prozentuale Reduktion der Datenrate für die Bewegungsvektoren und die restlichen Informationen gezeigt. Die Gesamtdatenrate setzt

Tabelle 5.6: Reduktion der Datenrate durch DSME gegenüber HM für verschiedene Sequenzen bei einem Quantisierungsparameter von 32.

Sequenz	**Bewegungs-vektoren**	**Rest**	**Gesamt**
BasketballDrive	7,9 %	4,8 %	5,4 %
BQTerrace	10,0 %	9,2 %	9,3 %
Cactus	20,6 %	10,5 %	12,2 %
Kimono	9,5 %	7,7 %	8,0 %
ParkScene	11,4 %	14,5 %	13,9 %
PeopleOnStreet	10,9 %	12,9 %	12,3 %
Traffic	34,8 %	29,9 %	31,0 %
NebutaFestival	3,5 %	-0,2 %	0,0 %
SteamLocomotiveTrain	1,8 %	1,8 %	1,8 %

sich aus beiden Teilen zusammen, wird jedoch aufgrund des geringeren Anteils der Bewegungsvektoren weniger stark durch diese beeinflusst. Es ist deutlich zu erkennen, dass nicht nur die Rate zur Codierung der Bewegungsvektoren verringert wird, sondern auch die Residuumsinformation.

5.4.3 Optimaler Referenzbildindex des DSME-Bildes

Durch die in Abschnitt 2.2.1 beschriebene Codierung des Referenzbildindex mit unterschiedlicher Codewortlänge, hat die Position des eingefügten DSME-Bildes innerhalb der Referenzbildliste einen Einfluss auf den Kompressionsgewinn. Wird das DSME-Bild sehr häufig als Referenz verwendet, sollte es vorne in die Liste eingefügt werden, um ein kürzeres Codewort zu erhalten. Ist hingegen die Interpolation des DSME-Bildes ungenauer, werden andere Referenzbilder häufiger verwendet. In diesem Fall kann das DSME-Bild ein längeres Codewort erhalten und somit am Ende der Referenzbildliste eingefügt werden.

Für die hier genutzte Konfiguration mit zwei Referenzbildern ergeben sich somit drei Positionen für das DSME-Bild: Vorne in der Liste (Index=0), zwischen den beiden bestehenden Referenzbildern (Index=1) oder hinten angehängt (Index=2). Die bereits vorhandenen Referenzbilder 1 und 2 werden entsprechend Abbildung 4.2 verschoben. Zur Bestimmung der optimalen Position des DSME-Bildes innerhalb der Referenzbildliste, sind in Abbildung 5.14 die mittleren Datenratenreduktionen aller neun Testsequenzen für die drei möglichen Positionen gezeigt.

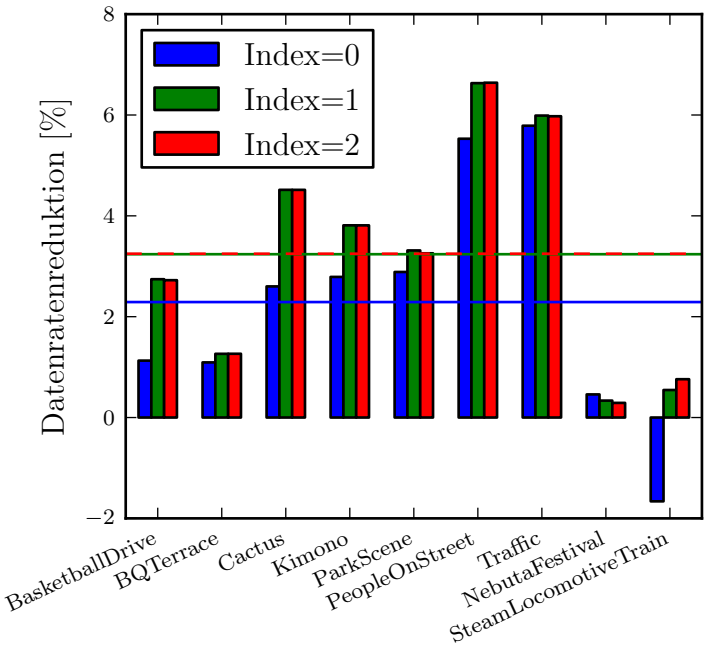


Abbildung 5.14: Reduktion der Datenrate durch Hinzufügen des DSME-Bildes an verschiedenen Stellen in der Referenzbildliste. Die mittlere Reduktion für die verschiendenen Postionen ist als horizontale Linie in der entsprechenden Farbe eingezeichnet.

Der geringste Gewinn von 2,29 % wird erzielt, wenn das DSME-Bild vorne eingefügt wird. Daraus lässt sich schließen, dass das Referenzbild 1 häufiger zur Prädiktion genutzt wird, als das DSME-Bild. Bei der SteamLocomotiveTrain-Sequenz zeigt sich, dass das DSME-Bild so selten genutzt wird, dass die Datenrate durch die unterschiedlichen Codewortlängen sogar ansteigt. Wird das DSME-Bild in der Mitte oder am Ende der Liste eingefügt, unterscheiden sich die mittleren Datenratenreduktionen von 3,24 % beziehungsweise 3,25 % nur minimal. Es kann demnach davon ausgegangen werden, dass das Referenzbild 2 und das DSME-Bild in etwa gleich häufig zur Prädiktion genutzt werden.

5.4.4 Kompressionseffizienz der Kombination von DSME und DMVD

Für DMVD wurde die kandidatenbasierte Implementation aus [11] genutzt. Um eine möglichst hohe Datenratenreduktion mit Hilfe von DMVD zu erreichen, wurde die in [10] vorgestellte Nachschätzung der Bewegungsvektoren genutzt. Die operativen RD-Kurven für die BasketballDrive- und PeopleOnStreet-Sequenz sind in den Abbildungen 5.15 und 5.16 dargestellt.

Für die BasketballDrive-Sequenz liegt DMVD leicht unterhalb von DSME. Es ist gut zu erkennen, dass die Kombination beider Techniken die Codiereffizienz weiter ansteigen lässt. Auch für die Sequenz PeopleOnStreet hat DMVD einen geringeren Gewinn als DSME. Jedoch ist die Codiereffizienz für den kombinierten Ansatz nur unwesentlich größer als für DSME. Wie in Abbildung 5.10 gezeigt, wird ein Großteil der Blöcke mit Hilfe des DSME-Bildes prädiziert, wodurch DMVD weniger Spielraum für die Reduktion der Datenrate bleibt. Die BD-Raten für alle Testsequenzen sind in Tabelle 5.7 gezeigt.

Wie zu erwarten, wird durch die Kombination von DSME und DMVD die Codiereffizienz erhöht. Jedoch addieren sich die Einzelgewinne bei der Kombination von DSME und DMVD nicht voll auf. Wird zur Prädiktion das DSME-Bild verwendet, kann DMVD keinen weiteren Gewinn erzielen. Bei einem Vergleich der Ergebnisse aus Tabelle 5.7 mit der Nutzungshäufigkeit von DSME aus Tabelle 5.5 zeigt sich, dass sich die Einzelgewinne für Sequenzen mit weniger als 25 % DSME-Nutzung – wie zum Beispiel BasketballDrive, NebutaFestival oder SteamLocomotiveTrain – durch den kombinierten Ansatz nahezu aufaddieren.

Nur die Ergebnisse für die Sequenz BQTerrace zeigen ein anderes Verhalten. Da der Einzelgewinn von DMVD für diese Sequenz bereits sehr gering ist, ist eine weitere Steigerung bei der Kombination nicht zu erwarten. Der Gewinn für DMVD durch die Verringerung der Rate für die Codierung der Bewegungsvektoren reicht gerade aus, um den zusätzlichen Signalisierungsaufwand dieser Methode zu kompensieren. Jedoch ist der Signalisierungsaufwand von DMVD so hoch, dass die Reduktion der Datenrate bei dem kombinierten Ansatz geringer ausfällt als für DMVD.

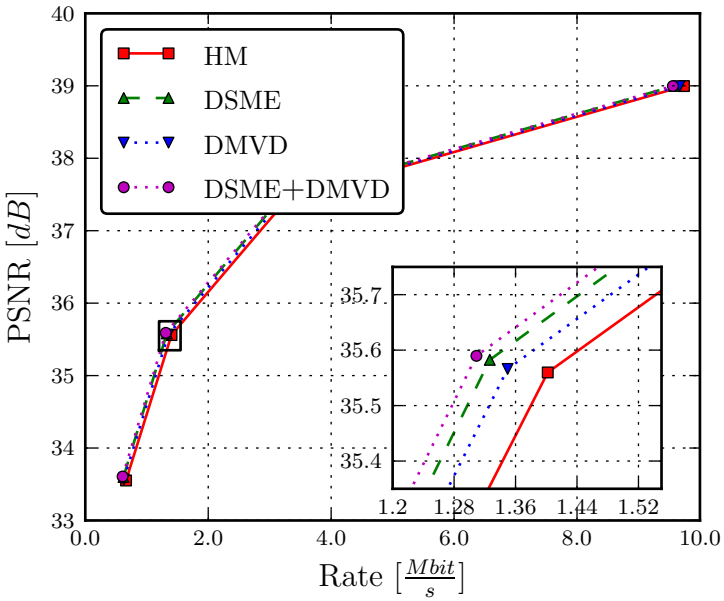


Abbildung 5.15: Gemessener PSNR der B-Bilder in Abhängigkeit der Datenrate für die Sequenz BasketballDrive bei Codierung mit HM, DSME, DMVD und der Kombination beider Techniken (DSME+DMVD).

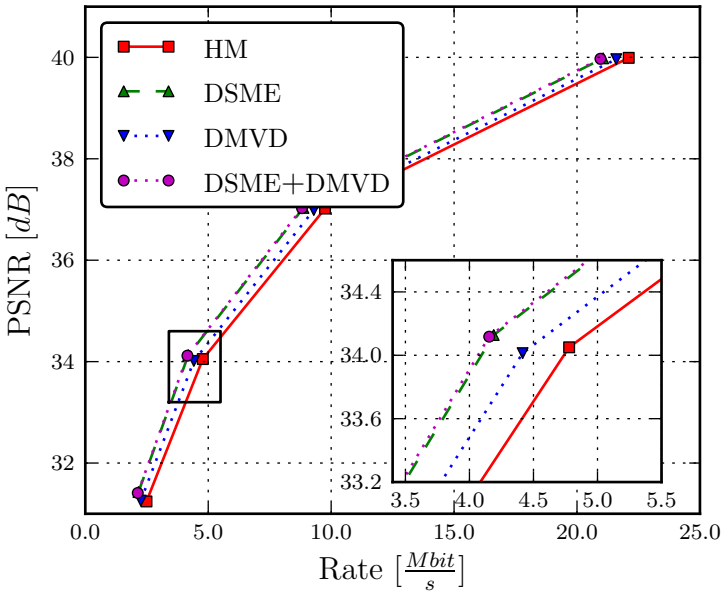


Abbildung 5.16: Gemessener PSNR der B-Bilder in Abhängigkeit der Datenrate für die Sequenz PeopleOnStreet bei Codierung mit HM, DSME, DMVD und der Kombination beider Techniken (DSME+DMVD).

Tabelle 5.7: Bjøntegaard-Delta der Gesamtdatenrate zwischen HM und DSME, DMVD sowie der Kombination beider Techniken (DSME+DMVD) für verschiedene Testsequenzen.

Sequenz	**DSME**	**DMVD**	**DSME+DMVD**
BasketballDrive	-2,72 %	-1,72 %	-3,84 %
BQTerrace	-1,26 %	-0,06 %	-0,85 %
Cactus	-4,52 %	-2,62 %	-4,92 %
Kimono	-3,81 %	-2,05 %	-4,22 %
ParkScene	-3,26 %	-1,69 %	-3,70 %
PeopleOnStreet	-6,64 %	-3,01 %	-7,02 %
Traffic	-5,98 %	-1,82 %	-6,22 %
NebutaFestival	-0,29 %	-0,18 %	-0,41 %
SteamLocomotiveTrain	-0,76 %	-1,45 %	-2,10 %
Mittelwert	-3,23 %	-1,62 %	-3,70 %

5.5 Zusammenfassung der Ergebnisse

Die Ergebnisse der experimentellen Untersuchungen in diesem Kapitel lassen sich mit folgenden Aussagen zusammenfassen:

- DSME verringert die Datenrate bei allen verwendeten Testsequenzen.
- Die größten Einsparungen der Datenraten gegenüber AVC und HEVC von etwa 7,6 % beziehungsweise 6,3 % lassen sich bei den 8 bit-Sequenzen mit einer Auflösung von 2560×1600 Bildpunkten erzielen.
- Die mittlere Reduktion der Datenraten aller untersuchten Testsequenzen liegt bei 5,4 % für AVC und 3,3 % für HEVC.
- Je geringer die Qualität der codierten Sequenz ist und somit auch die benötigte Datenrate, desto höher ist der Codiergewinn durch DSME.
- Die optimale Positionierung des DSME-Bildes innerhalb der Referenzbildliste ist unabhängig von der gewählten Sequenz.
- Bis auf eine Ausnahme, lässt sich durch Kombination von DSME mit DMVD die Codiereffizienz bei allen Testsequenzen weiter erhöhen.

6 Zusammenfassung

Bei aktuellen Videocodierungstandards besteht bis zu 25 % der zur Codierung aufgewendeten Datenrate aus Bewegungsinformationen, welche zur Erstellung einer bewegungskompensierten Prädiktion benötigt werden. Wäre der Decoder in der Lage, die Bewegung zuverlässig zu schätzen, könnten diese Daten bei der Codierung entfallen oder zumindest verringert werden. Bereits im aktuellen AVC-Standard werden die Bewegungsvektoren mit einem einfachen Medianfilter aus den Vektoren benachbarter Blöcke geschätzt, wodurch lediglich der Schätzfehler übertragen werden muss. Dieser Ansatz wurde in *Motion Vector Competition* (MVC) erweitert, um auch bereits decodierte Bewegungsvektoren aus anderen Bildern zur Schätzung der Bewegung zu nutzen. Jedoch nutzt keine dieser Methoden decodiertes Bildmaterial für eine genauere Schätzung der Bewegung.

In der vorliegenden Arbeit wurden die Möglichkeiten bei der Bewegungsschätzung am Decoder, basierend auf bereits decodiertem Bildmaterial, untersucht. Dabei wurden zwei besonders erfolgversprechende Ansätze identifiziert, die auf unterschiedlichen Annahmen bezüglich des Bewegungsvektorfeldes beruhen: Bei *Decoder-Side Motion Vector Derivation* (DMVD) wird davon ausgegangen, dass das Bewegungsvektorfeld örtlich homogen ist, benachbarte Bildpunkte also eine sehr ähnliche Bewegung haben. Demgegenüber ist es denkbar, die Bewegung zwischen Referenzbildern zu bestimmen, indem davon ausgegangen wird, dass keine Beschleunigung vorliegt. Bei diesem Ansatz wird somit ein zeitlich homogenes Bewegungsvektorfeld vorausgesetzt.

Um den theoretischen Nutzen einer Bewegungsschätzung am Decoder unter Annahme zeitlich homogener Bewegung abschätzen zu können, wurde in dieser Arbeit ein aus der Literatur bekanntes Modell zur Berechnung der Datenrate bei prädiktiver Codierung um die decoderseitige Bewegungsschätzung erweitert. Es wird davon ausgegangen, dass die Bewegung aus zwei bereits decodierten Bildern, von denen jeweils eines zeitlich vor und eines nach dem zu decodierenden Bild liegen, bestimmt wird. Ist die Bewegung zwischen diesen Bildern jedoch nicht konstant, kommt es zu Fehlern bei der Schätzung. Daher wird unter anderem der Einfluss beschleunigter Bewegung in das Modell einbezogen und mit lediglich zwei sequenzabhängigen Parametern modelliert. Der eine Parameter beschreibt die Varianz der Beschleunigung zwischen den Bildern und der zweite Parameter die Verteilung unterschiedlicher Beschleunigungen innerhalb des Bildes. Da das aus der Literatur entnommene Modell keine Transformation zur Codierung des Prädiktionsfehlers berücksichtigt und die Blockgröße als konstant angenommen wird, anstatt sie adaptiv dem Bildinhalt

anzupassen, wurde ein experimenteller Coder mit den gleichen Einschränkungen entworfen. Obwohl in der Modellannahme lediglich zwei neue Parameter eingeführt wurden, konnte mit Hilfe des Coders gezeigt werden, dass das Modell den praktischen Ergebnissen der neun HEVC-Testequenzen sehr gut entspricht.

Mit Hilfe des erweiterten Modells konnte gezeigt werden, dass eine Bewegungsschätzung am Decoder die Gesamtdatenrate senkt. Im Falle beschleunigter Bewegung muss diese durch zusätzliche Seiteninformation kompensiert werden. Daher wurde im Rahmen dieser Arbeit eine Coderarchitektur entwickelt, bei der die Bewegung aus bereits decodierten Bildern ermittelt und daraus anschließend eine bewegungskompensierte Schätzung des zu decodierenden Bildes berechnet wird. Dieses künstlich erzeugte Bild kann zusätzlich zu anderen Referenzbildern am Decoder zur Prädiktion einzelner Blöcke genutzt werden, ohne zusätzliche Bewegungsvektoren zu benötigen. Der Encoder entscheidet blockweise, ob dieses künstlich erzeugte Bild oder andere bereits decodierte Bilder als Referenz genutzt werden sollen. Dieser, als *Decoder-side Motion Estimation* (DSME) bezeichnete Ansatz, findet bei B-Bildern Verwendung, da für diese Referenzbilder aus der Vergangenheit und Zukunft vorhanden sind. Unter der Annahme konstanter Bewegung zwischen diesen Referenzbildern ist es mit Hilfe bewegungskompensierter Interpolation möglich, das aktuelle Bild zu schätzen. Sollte dennoch beschleunigte Bewegung vorhanden sein, kann diese leicht durch zusätzliche Übertragung von Bewegungsinformationen kompensiert werden, ohne dass dazu der zugrunde liegende Coder modifiziert werden muss.

Die Herausforderung bei der Interpolation des sogenannten DSME-Bildes liegt in der Bestimmung der Bewegung zwischen den Referenzbildern, da die wahre Bewegung benötigt wird. Eine falsche Korrespondenz zwischen den Referenzbildern kann bei der Interpolation zu starken Artefakten führen, wodurch der Kompressionsgewinn gegenüber konventioneller Codierung verschwindet. Somit sind einfache Blockmatching-Algorithmen zur Bewegungsschätzung, wie sie meist in Videoencodern genutzt werden, ungeeignet. Sie minimieren lediglich ein Fehlermaß wie die Summe der Differenzbeträge (SAD), ohne die Plausibilität der Bewegung zu berücksichtigen. Insbesondere bei sich wiederholenden Strukturen kann es so zu falschen Zuordnungen kommen.

Als Teil dieser Arbeit ist eine angepasste hierarchische Bewegungsschätzung entwickelt worden, die es erlaubt, die wahre Bewegung der Objekte zu schätzen. Dieser Algorithmus macht es möglich, ein dichtes Bewegungsvektorfeld zu bestimmen, bei dem jeder Bildpunkt einen eigenen Vektor besitzt. Dadurch lässt sich insbesondere an Objektgrenzen die Bewegung gut kompensieren. Im Gegensatz zu Algorithmen für die Zwischenbildberechnung in LCD-TVs – auch bekannt als *Frame Rate Up Conversion* (FRUC) – ist bei DSME die subjektive Qualität nicht von vorrangiger Bedeutung. Es ist ausreichend, wenn die Mehrzahl der Blöcke genau prädiziert werden. Einzelne fehlerhaft prädizierte Blöcke innerhalb des DSME-Bildes sind unproblematisch, da der Encoder für die betroffenen Blöcke andere Referenzbilder wählen kann. Aus diesem Grund ergeben sich andere Anforderungen an die Bewegungsschätzung.

Das innerhalb der hierarchischen Bewegungsschätzung verwendete Blockmatching verliert bei kleinen Blöcken an Robustheit. Es stehen nur noch sehr wenige Bildpunkte zur Verfügung, wodurch die Minimierung des Fehlermaßes stark durch Bildrauschen beeinflusst wird. Daher ist für kleine Blöcke das sogenannte Vektorlatching eingeführt worden, bei dem Blöcke sich nur noch der Bewegung angrenzender Objekte anpassen, aber keine unabhängige Bewegung mehr vollziehen können. Somit kann weiterhin die Bewegung an Objektgrenzen sehr genau bestimmt werden, ohne zu anfällig für Bildrauschen zu werden.

In dieser Arbeit wurde außerdem untersucht, wie sich zeitlich und örtlich homogene Bewegungen in einem Coder berücksichtigen lassen, um die Kompression weiter zu steigern. Dazu wurde DSME in einen DMVD-fähigen Coder implementiert und mit den beiden getrennten Implementierungen verglichen.

Experimentelle Untersuchungen anhand des AVC-Referenzcoders JM zeigen, dass DSME die benötigte Datenrate bei gleicher Qualität im Mittel um 5,4 % reduziert. Dies entspricht einer objektiven Qualitätssteigerung von 0,22 dB bei konstanter Datenrate. Da der AVC-Standard bereits mehrere Jahre alt ist, wird DSME außerdem mit dem HEVC-Testmodell (HM) verglichen, welches den aktuellen Stand der Forschung bei der Entwicklung eines neuen Videocodierstandards darstellt. Trotz neuer und effizienterer Algorithmen erreicht DSME weiterhin eine mittlere Reduktion der benötigten Datenraten von 3,3 %. Sowohl DSME als auch DMVD haben zum Ziel, die Datenrate für die Bewegungsvektoren zu verringern. Dennoch wird aufgrund der unterschiedlichen Annahmen über die Bewegung bei der Kombination beider Techniken ein Kompressionsgewinn von 3,7 % gegenüber der Referenz erzielt. Unter Berücksichtigung des Einzelgewinns bei DMVD von 1,6 % wird jedoch deutlich, dass sich die Gewinne bei der Kombination nicht aufaddieren. Die einzelnen Gewinne durch die unterschiedlichen Annahmen sind somit nicht vollständig orthogonal zueinander, da die Signalisierung der Bewegung für Blöcke, bei denen zeitliche als auch örtliche Homogenität der Bewegung vorliegt, nur einmal eingespart werden kann.

Dem Kompressionsgewinn durch DSME steht die gestiegene Komplexität am Decoder gegenüber. Jedoch kann aufgrund der flexiblen Architektur die verwendete decoderseitige Bewegungsschätzung leicht ausgetauscht werden. Somit lässt sich leicht die Komplexität durch simplere Algorithmen auf Kosten der Codiereffizienz reduzieren. Außerdem ist eine Kombination des Decoders mit einem aktuellen LCD-TV in nur einem Gerät denkbar. Zur Interpolation des DSME-Bildes könnte dann dieselbe Hardware genutzt werden, welche auch die – für LCD-TVs notwendige – Zwischenbildberechnung durchführt. Ein Verschmelzung von Decoder und Anzeigegerät ist somit zu bevorzugen, was bereits in aktuellen Fernsehern zu beobachten ist. Neben der reinen Darstellung eines decodierten Videosignals übernehmen viele Fernseher immer mehr Aufgaben, wie das Anzeigen von Onlineinhalten und die Decodierung von Internetvideos.

A Anhang

A.1 Herleitung der optimalen Blockgröße

Zur Berechnung der optimalen Blockgröße in Abschnitt 3.1.4, wird das Modell nach der Blockgröße B abgeleitet und mit Null gleichgesetzt:

$$\frac{\partial R}{\partial B} = \frac{\partial R_V}{\partial B} + \frac{\partial R_R}{\partial B} \stackrel{!}{=} 0. \tag{A.1}$$

Dabei hängt die Rate für die Bewegungsvektoren R_V von dem verwendetem Modell (konventionell oder DSME) ab. Die Rate R_R des Residuums ist bei beiden Modellen identisch, es muss jedoch zwischen feiner ($\check{R}_R$) und grober Quantisierung ($\hat{R}_R$) unterschieden werden. Als Erstes wird die partielle Ableitung der einzelnen Komponenten berechnet:

- Ableitung der Rate für das Residuum bei feiner Quantisierung aus Gleichung 3.36

$$\begin{aligned}
\frac{\partial \check{R}_R}{\partial B} &= \frac{\partial}{\partial B} \log_2 \frac{\sqrt{2}e\sqrt{\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right) GB + \frac{Q^2}{12} + \mu}}{Q} \\
&= \frac{\partial}{\partial B} \log_2 \frac{\sqrt{2}e}{Q} \\
&\quad + \frac{\partial}{\partial B} \frac{1}{2} \log_2 \left(\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right) GB + \frac{Q^2}{12} + \mu \right) \\
&= \frac{1}{2} 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right) G \frac{1}{\ln 2} \frac{1}{\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right) GB + \frac{Q^2}{12} + \mu} \\
&= \frac{1}{2\ln 2} k_1 \left(k_1 B + k_2\right)^{-1}
\end{aligned} \tag{A.2}$$

  mit

$$k_1 = 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right) G, \tag{A.3}$$

$$k_2 = \Delta^2 G + \frac{Q^2}{12} + \mu. \tag{A.4}$$

- Ableitung der Rate für das Residuum bei feiner Quantisierung aus Gleichung 3.36

$$\begin{aligned}\frac{\partial \hat{R}_R}{\partial B} &= \frac{\partial}{\partial B}\frac{e}{Q^2 \ln 2}\left(\Delta^2 G + 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right) GB + \frac{Q^2}{12} + \mu\right) \\ &= \frac{e}{Q^2 \ln 2} 6\sigma_V^2 \ln\left(\frac{1}{c_A}\right) G \\ &= \frac{e}{Q^2 \ln 2} k_1. \end{aligned} \tag{A.5}$$

- Ableitung der Rate für die Bewegungsvektoren bei DSME-Codierung aus Gleichung 3.19

$$\begin{aligned}\frac{\partial R_V^{(\mathrm{DSME})}}{\partial B} &= \frac{\partial}{\partial B}\frac{P_{B_0}}{B_0}\frac{1}{B}\log_2 \frac{\frac{1}{2}\pi e T_t^4 \sigma_a^2}{\Delta^2} \\ &= -\frac{P_{B_0}}{B_0}\frac{1}{B^2}\log_2 \frac{\frac{1}{2}\pi e T_t^4 \sigma_a^2}{\Delta^2} \\ &= -\frac{1}{\ln 2} k_3 \frac{1}{B^2} \end{aligned} \tag{A.6}$$

mit

$$k_3 = \frac{P_{B_0}}{B_0} \ln \frac{\frac{1}{2}\pi e T_t^4 \sigma_a^2}{\Delta^2}. \tag{A.7}$$

- Ableitung der Rate für die Bewegungsvektoren bei konventioneller Codierung aus [57]

$$\begin{aligned}\frac{\partial R_V^{(\mathrm{KONV})}}{\partial B} &= \frac{\partial}{\partial B}\frac{1}{B^2}\log_2 \frac{4e^2\sigma_V^2 \ln\left(\frac{1}{\tilde{c}_A}\right) B}{\Delta^2} \\ &= -2\frac{1}{B^3}\log_2 \frac{4e^2\sigma_V^2 \ln\left(\frac{1}{\tilde{c}_A}\right) B}{\Delta^2} \\ &\quad + \frac{1}{B^2}\frac{1}{\ln 2}\frac{\Delta^2}{4e^2\sigma_V^2 \ln\left(\frac{1}{\tilde{c}_A}\right) B}\frac{4e^2\sigma_V^2 \ln\left(\frac{1}{\tilde{c}_A}\right)}{\Delta^2} \\ &= \frac{1}{B^3}\frac{1}{\ln 2}\left(1 - 2\ln \frac{4e^2\sigma_V^2 \ln\left(\frac{1}{\tilde{c}_A}\right) B}{\Delta^2}\right) \\ &= \frac{1}{B^3}\frac{1}{\ln 2}\left(1 - 2k_4 - 2\ln B\right) \end{aligned} \tag{A.8}$$

mit

$$k_4 = \ln \frac{4e^2\sigma_V^2 \ln\left(\frac{1}{\tilde{c}_A}\right)}{\Delta^2}. \tag{A.9}$$

Mit diesen Ableitungen können nun die optimalen Blockgrößen für die unterschiedlichen Konfigurationen hergeleitet werden:

- Berechnung der optimalen Blockgröße $B_{\text{opt}} = \check{B}_{\text{opt}}^{(\text{DSME})}$ für DSME-Codierung bei feiner Quantisierung aus Gleichung A.1 mit den Gleichungen A.2 und A.6

$$\begin{aligned}
0 &\stackrel{!}{=} \frac{\partial \check{R}^{(\text{DSME})}}{\partial B} \\
0 &= -\frac{1}{\ln 2} k_3 \frac{1}{B_{\text{opt}}^2} + \frac{1}{2\ln 2} k_1 \left(k_1 B_{\text{opt}} + k_2\right)^{-1} \\
0 &= -k_3 \left(k_1 B_{\text{opt}} + k_2\right) + \frac{1}{2} k_1 B_{\text{opt}}^2 \\
0 &= B_{\text{opt}}^2 - 2k_3 B_{\text{opt}} - 2\frac{k_2 k_3}{k_1} \\
\Rightarrow B_{\text{opt}} &= k_3 + \sqrt{(k_3)^2 + 2\frac{k_2 k_3}{k_1}}.
\end{aligned} \tag{A.10}$$

  Die zweite Lösung der quadratischen Gleichung wird nicht berücksichtigt, da diese eine negative Blockgröße ergibt und somit keine praktische Bedeutung hat.

- Berechnung der optimalen Blockgröße $B_{\text{opt}} = \hat{B}_{\text{opt}}^{(\text{DSME})}$ für DSME-Codierung bei grober Quantisierung aus Gleichung A.1 mit den Gleichungen A.5 und A.6

$$\begin{aligned}
0 &\stackrel{!}{=} \frac{\partial \hat{R}^{(\text{DSME})}}{\partial B} \\
0 &= -\frac{1}{\ln 2} k_3 \frac{1}{B_{\text{opt}}^2} + \frac{e}{Q^2 \ln 2} k_1 \\
0 &= B_{\text{opt}}^2 - \frac{Q^2}{e}\frac{k_3}{k_1} \\
\Rightarrow B_{\text{opt}} &= \sqrt{\frac{Q^2}{e}\frac{k_3}{k_1}}.
\end{aligned} \tag{A.11}$$

  Auch hier ist die zweite Lösung der Gleichung negativ und wird nicht berücksichtigt.

- Berechnung der optimalen Blockgröße $B_{\text{opt}} = \check{B}_{\text{opt}}^{(\text{KONV})}$ für konventionelle Codierung bei feiner Quantisierung aus Gleichung A.1 mit den Gleichungen A.2 und A.8

$$\begin{aligned} 0 &\stackrel{!}{=} \frac{\partial \check{R}^{(\text{KONV})}}{\partial B} \\ 0 &= \frac{1}{B_{\text{opt}}^3} \frac{1}{\ln 2} \left(1 - 2k_4 - 2 \ln B_{\text{opt}}\right) + \frac{1}{2 \ln 2} k_1 \left(k_1 B_{\text{opt}} + k_2\right)^{-1} \\ 0 &= 2 \left(k_1 B_{\text{opt}} + k_2\right) \left(1 - 2k_4 - 2 \ln B_{\text{opt}}\right) + k_1 B_{\text{opt}}^3. \end{aligned} \tag{A.12}$$

  Zur Bestimmung der optimalen Blockgröße aus dieser Gleichung können numerische Verfahren verwendet werden.

- Berechnung der optimalen Blockgröße $B_{\text{opt}} = \hat{B}_{\text{opt}}^{(\text{KONV})}$ für konventionelle Codierung bei grober Quantisierung aus Gleichung A.1 mit den Gleichungen A.5 und A.8

$$\begin{aligned} 0 &\stackrel{!}{=} \frac{\partial \hat{R}^{(\text{KONV})}}{\partial B} \\ 0 &= \frac{1}{B_{\text{opt}}^3} \frac{1}{\ln 2} \left(1 - 2k_4 - 2 \ln B_{\text{opt}}\right) + \frac{e}{Q^2 \ln 2} k_1 \\ 0 &= 1 - 2k_4 - 2 \ln B_{\text{opt}} + \frac{e}{Q^2} k_1 B_{\text{opt}}^3. \end{aligned} \tag{A.13}$$

  Auch hier können numerische Verfahren zur Bestimmung der optimalen Blockgröße verwendet werden.

A.2 Parameter der hierarchischen Bewegungsschätzung

Hierarchielevel	**Blockgröße**	**Suchfenster**	**Genauigkeit**	**Bidirektional**	**Anzahl Nachbarn**	**Überlappende Blockgröße**
1	64 pel	256 pel	2 pel		0	0 pel
2	32 pel	16 pel	1 pel		8	0 pel
3	16 pel	8 pel	1 pel		8	0 pel
4	8 pel	4 pel	0.5 pel		8	12 pel
5	4 pel	0 pel	0.25 pel		8	8 pel
6	2 pel	0 pel	0.25 pel	✓	8	4 pel
7	1 pel	0 pel	0.25 pel	✓	8	3 pel

Literaturverzeichnis

[1] N. Ahmed, T. Natarajan und K.R. Rao. Discrete Cosine Transform. *IEEE Transactions on Computers*, C-23(1):90–93, Januar 1974.

[2] Luciano Alparone, Mauro Barni, Franco Bartolini und Vito Cappellini. Adaptive weighted vector-median filters for motion fields smoothing. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, S. 2267–2270, Georgia, USA, May 1996.

[3] João Ascenso, Catarina Brites und Fernando Pereira. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. In *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Smolenice, Slovak Republic, Juli 2005.

[4] Michael Becker und Ulrike Kuhlmann. Rasante Zeiten: Techniken zur besseren Bewegtbilddarstellung auf Flachbildschirmen. *c't Magazin*, 2005(9):126–129, September 2005.

[5] Matthias Bierling und Robert Thoma. Motion compensating field interpolation using a hierarchically structured displacement estimator. *Signal Processing*, 11 (4):387–404, Dezember 1986.

[6] Gisle Bjøntegaard. Calculation of average PSNR differences between RD curves. In *ITU-T SG16/Q6 Output Document VCEG-M33*, Austin, Texas, April 2001.

[7] Stefan Borchert. *Distributed Video Coding (DVC): Motion estimation and DCT quantization in low complexity video compression.* TU Delft Mediamatica, Delft, Netherland, Juni 2010.

[8] Frank Bossen. Common test conditions and software reference configurations. In *JCT-VC Output Document JCTVC-D600*, Daegu, Korea, Januar 2011.

[9] Andrés Bruhn. *Variational optic flow computation – Accurate modelling and efficient numerics.* Department of Mathematics and Computer Science, Saarland University, Saarbrücken, Germany, Juli 2006.

[10] Yi-Jen Chiu, Lidong Xu, Wenhao Zhang und Hong Jiang. CE1: Report of self derivation of motion estimation in TMuC 0.9. In *JCT-VC Document JCTVC-D167*, Daegu, Korea, Januar 2011.

[11] Yi-Jen Chiu, Lidong Xu, Wenhao Zhang, Hong Jiang, Mingyuan Yang, Sixin Lin und Haoping Yu. CE1 Subtest1: A joint proposal of candidate-based decoder-side motion vector derivation. In *JCT-VC Output Document JCTVC-D448*, Daegu, Korea, Januar 2011.

[12] Byeong-Doo Choi, Jong-Woo Han, Chang-Su Kim und Sung-Jea Ko. Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(4):407 –416, April 2007.

[13] Byung-Tae Choi, Sung-Hee Lee und Sung-Jea Ko. New frame rate up-conversion using bi-directional motion estimation. *IEEE Transactions on Consumer Electronics*, 46(3):603–609, August 2000.

[14] Cisco Systems, Inc. Visual networking index: Forecast and methodology, Juni 2010. `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html`.

[15] Thomas M. Cover und Joy A. Thomas. *Elements of Information Theory*, Kapitel 9.3. John Wiley & Sons Inc., New York, NY, 1991.

[16] Gerard de Haan, Paul W.A.C. Biezen, Henk Huijgen und Olukayode A. Ojo. True-motion estimation with 3-D recursive search block matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(5):368–379, oct 1993.

[17] Bernd Girod. Motion-compensating prediction with fractional-pel accuracy. *IEEE Transactions on Communications*, 41(4):604–612, April 1993.

[18] P. Guillotel und C. Chevance. Comparison of motion vector coding techniques. In *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, S. 1594–1604, Chicago, IL, USA, Februar 1994.

[19] Taehyeun Ha, Seongjoo Lee und Jaeseok Kim. Motion compensated frame interpolation by new block-based motion estimation algorithm. *IEEE Transactions on Consumer Electronics*, 50(2):752–759, Mai 2004.

[20] Chris Harris und Mike Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, S. 147–151, 1988.

[21] Chi-Wang Ho, Oscar C. Au, S.-H. Gary Chan, Shu-Kei Yip und Hoi-Ming Wong. Motion estimation for H.264/AVC using programmable graphics hardware. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, S. 2049–2052, Toronto, Canada, Juli 2006.

[22] Shunsuke Ihara. *Information theory for continuous systems*, Kapitel 1.8. World Scientific, Singapore, 1993.

[23] ISO/IEC. *ISO/IEC 11172-2 (MPEG-1 Part 2): Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video.* August 1993.

[24] ISO/IEC. *ISO/IEC 14496:2000-2: Information technology - Coding of audio-visual objects - Part 2: Visual.* Dezember 2000.

[25] ISO/IEC und ITU-T. *Recommendation ITU-T H.264 and ISO/IEC 14496-10 (MPEG-4 Part 10): Advanced Video Coding (AVC) - 3rd Edition.* Geneva, Switzerland, Juli 2004.

[26] ISO/IEC und ITU-T. *Recommendation ITU-T H.263 and ISO/IEC 13818-2 (MPEG-2 Part 2): Information technology - Generic coding of moving pictures and associated audio information: Video.* März 1995.

[27] ISO/IEC JTC1/SC29/WG11 MPEG. Vision, applications and requirements for High-Performance Video Coding (HVC). In *ISO/IEC JTC1/SC29/WG11 MPEG Output Document N11096*, Kyoto, January 2010.

[28] ISO/IEC JTC1/SC29/WG11 MPEG. Joint Call for Proposals on video compression technology. In *ISO/IEC JTC1/SC29/WG11 MPEG Output Document N11113*, Kyoto, January 2010.

[29] ITU-R. *Recommendation ITU-R BT.601-5: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios.* International Telecommunication Union, Geneva, Switzerland, 1995.

[30] ITU-R. *Recommendation ITU-R BT.709-3: Parameter values for the HDTV standards for production and international exchange.* International Telecommunication Union, Geneva, Switzerland, 1998.

[31] ITU-T. *Recommendation ITU-T H.261: Video codec for audiovisual services at p x 64 kbit/s.* Geneva, Switzerland, November 1988.

[32] ITU-T. *Recommendation ITU-T H.263: Video codec for low bit rate communication.* Geneva, Switzerland, 1998.

[33] JCT-VC. HEVC reference software HM. `https://hevc.hhi.fraunhofer.de/`.

[34] JVT. Joint Model (JM) H.264 / MPEG-4 AVC software. `http://iphome.hhi.de/suehring/tml/`.

[35] Steffen Kamp und Mathias Wien. Decoder-side motion vector derivation for hybrid video inter coding. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, S. 1277–1280, Singapore, Juli 2010.

[36] Steffen Kamp, Michale Evertz und Mathias Wien. Decoder side motion vector derivation for inter frame video coding. In *Proceedings of the IEEE International Conference on Image Processing*, S. 1120–1123, San Diego, CA, USA, Oktober 2008.

[37] Steffen Kamp, Johannes Ballé und Mathias Wien. Multihypothesis prediction using decoder side motion vector derivation in inter frame video coding. In *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, San José, CA, USA, Januar 2009.

[38] Sven Klomp und Jörn Ostermann. Evaluation of decoder-side motion estimation within HM 2.0. In *JCT-VC Output Document JCTVC-E055*, Geneva, Switzerland, März 2011.

[39] Sven Klomp, Yuri Vatis und Jörn Ostermann. Side information interpolation with sub-pel motion compensation for Wyner-Ziv decoder. In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, S. 178–182, Setúbal, Portugal, August 2006.

[40] Sven Klomp, Marco Munderloh, Yuri Vatis und Jörn Ostermann. Decoder-side block motion estimation for H.264 / MPEG-4 AVC based video coding. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, S. 1641–1644, Taipei, Taiwan, Mai 2009.

[41] Sven Klomp, Marco Munderloh und Jörn Ostermann. Block size dependent error model for motion compensation. In *Proceedings of the IEEE International Conference on Image Processing*, S. 969–972, Hong Kong, September 2010.

[42] Sven Klomp, Marco Munderloh und Jörn Ostermann. Decoder-side hierarchical motion estimation for dense vector fields. In *Proceedings of the Picture Coding Symposium*, S. 362–366, Nagoya, Japan, Dezember 2010.

[43] Sven Klomp, Marco Munderloh und Jörn Ostermann. Decoder-side motion estimation assuming temporally or spatially constant motion. *ISRN Signal Processing*, April 2011.

[44] Guillaume Laroche, Joel Jung und Beatrice Pesquet-Popescu. A spatio-temporal competing scheme for the rate-distortion optimized selection and coding of motion vectors. In *Proceedings of the European Signal Processing Conference*, Florence, Italy, September 2006.

[45] Guillaume Laroche, Joel Jung und Beatrice Pesquet-Popescu. RD optimized coding for motion vector predictor selection. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9):1247–1257, September 2008.

[46] Ken McCann, Benjamin Bross und Shun ichi Sekiguchi Shun-ichi Sekiguchi. High Efficiency Video Coding (HEVC) Test Model 2 (HM 2) Encoder Description. In *JCT-VC Output Document JCTVC-D502*, Daegu, Korea, Januar 2011.

[47] Gerhard Merziger und Thomas Wirth. *Repetitorium der Höheren Mathematik*, Kapitel 12.7. Binomi Verlag, Springe, Germany, 1999.

[48] Fabrice Moscheni, Frédéric Dufaux und H. Nicolas. Entropy criterion for optimal bit allocation between motion and prediction error information. In *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, S. 235–242, Boston, MA, November 1993.

[49] Marco Munderloh, Sven Klomp und Jörn Ostermann. Mesh-based decoder-side motion estimation. In *Proceedings of the IEEE International Conference on Image Processing*, S. 2049–2052, Hong Kong, September 2010.

[50] Tomokazu Murakami und Shohei Saito. Advanced B skip mode with decoder-side motion estimation. In *ITU-T SG16/Q6 Output Document VCEG-AK12*, Yokohama, Japan, April 2009.

[51] Tomokazu Murakami, Shohei Saito, Yuto Komatsu, Katsuyuki Nakamura und Toru Yokoyama. Enhancement of H.264/AVC for higher coding efficiency using motion estimation between reference frames. *IEEE Transactions on Consumer Electronics*, 56(2):925–929, Mai 2010.

[52] Hans-Georg Musmann, Peter Pirsch und Hans-Joachim Grallert. Advances in picture coding. *Proceedings of the IEEE*, 73(4):523–548, April 1985.

[53] Matthias Narroschke. *Adaptive Prädiktionsfehlercodierung für die Hybridcodierung von Videosignalen.* Nummer 786 in Fortschritt-Berichte VDI. VDI Verlag GmbH, Düsseldorf, Germany, 2008.

[54] Society of Motion Picture und Television Engineers. *SMPTE 2036-1: Ultra High Definition Television - Image parameter values for program production.* Dezember 2009.

[55] Fernando Pereira, Luis Torres, Christine Guillemot, Touradj Ebrahimi, Riccardo Leonardi und Sven Klomp. Distributed Video Coding: Selecting the most promising application scenarios. *Signal Processing: Image Communication*, 23 (5):339–352, Juni 2008.

[56] Jordi Ribas-Corbera und David L. Neuhoff. On the optimal motion vector accuracy for block-based motion-compensated video coders. In *Proceedings of the SPIE Conference on Digital Video Compression*, S. 302–314, San José, CA, USA, Januar 1996.

[57] Jordi Ribas-Corbera und David L. Neuhoff. On the optimal block size for block-based, motion-compensated video coders. In *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, S. 1132–1143, San José, CA, USA, Januar 1997.

[58] Jordi Ribas-Corbera und David L. Neuhoff. Optimizing block size in motion-compensated video coding. *Journal of Electronic Imaging*, 7(1):155–165, Januar 1998.

[59] Iain E. G. Richardson. *H.264 and MPEG-4 video compression*, Kapitel 6.4.5.3. John Wiley & Sons Ltd., West Sussex, England, 2003.

[60] John G. Robson. Spatial and temporal contrast-sensitivity functions of the visual system. *Journal of the Optical Society of America*, 56(8):1141–1142, August 1966.

[61] Jianbo Shi und Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, S. 593–600, Seattle, Washington, USA, Juni 1994.

[62] Yoshiaki Shishikui, Yasutaka Matsuo, Atsuro Ichigaya, Kazuhisa Iguchi und Shinichi Sakaida. Characteristics of Super Hi-Vision test sequences. In *JCT-VC Document JCTVC-C032*, Guangzhou, China, Oktober 2010.

[63] David Slepian und Jack K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, Juli 1973.

[64] Gary J. Sullivan und Jens-Rainer Ohm. Meeting report of the fourth meeting of the Joint Collaborative Team on Video Coding (JCT-VC). In *JCT-VC Output Document JCTVC-D500*, Daegu, Korea, Januar 2011.

[65] Gary J. Sullivan und Thomas Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 15(11):74–90, November 1998.

[66] Yuri Vatis. *Non-symmetric adaptive interpolation filter for motion compensated prediction*. Nummer 802 in Fortschritt-Berichte VDI. VDI Verlag GmbH, Düsseldorf, Germany, 2009.

[67] Thomas Wedi. Adaptive interpolation filters and high-resolution displacements for video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(4):484–491, April 2006.

[68] Oliver Werner. Drift analysis and drift reduction for multiresolution hybrid video coding. *Signal Processing: Image Communication*, 8(5):387–409, Juli 1996.

[69] Thomas Wiegand, Heiko Schwarz, Anthony Joch, Faouzi Kossentini und Gary J. Sullivan. Rate-constrained coder control and comparison of video coding standards. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):688–703, Juli 2003.

[70] Thomas Wiegand, Jens-Rainer Ohm, Gary J. Sullivan, Woo-Jin Han, Rajan Joshi, Thiow Keng Tan und Kemal Ugur. Special section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) standardization. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(12):1661–1666, Dezember 2010.

[71] Mathias Wien und Yi-Jen Chiu. Tool Experiment 1: Decoder-side motion vector derivation. In *JCT-VC Output Document JCTVC-A301*, Dresden, Germany, April 2010.

[72] Wikipedia. Standardabweichung — Wikipedia, Die freie Enzyklopädie, 2011. `http://de.wikipedia.org/w/index.php?title=Standardabweichung&oldid=93578652`. [Online; Stand 18. September 2011].

[73] Wikipedia. Zeilensprungverfahren — Wikipedia, Die freie Enzyklopädie, 2011. `http://de.wikipedia.org/w/index.php?title=Zeilensprungverfahren&oldid=90837648`. [Online; Stand 3. August 2011].

[74] Aaron D. Wyner und Jacob Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, Januar 1976.

[75] Yongbing Zhang, Debin Zhao, Xiangyang Ji, Ronggang Wang und Wen Gao. A spatio-temporal auto regressive model for frame rate upconversion. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(9):1289–1301, September 2009.

Lebenslauf von Sven Klomp

08.11.1979	geboren in Nordhorn, Deutschland

Beruf

seit 09/2011	Softwarearchitekt bei der *Robert Bosch Car Multimedia GmbH*, Hildesheim
07-10/2009	Praktikum bei den *Mitsubishi Electric Research Laboratories*, Cambridge, Massachusetts, USA
2004 - 2011	Wissenschaftlicher Mitarbeiter am *Institut für Informationsverarbeitung* an der *Leibniz Universität Hannover*

Studium

1999 - 2004	Studium der Elektrotechnik an der *Leibniz Universität Hannover*, Studienrichtung Nachrichtentechnik Abschluss mit Diplom (Dipl.-Ing.)

Schulbildung

1996 - 1999	Besuch des Fachgymnasium für Technik in Nordhorn Abschluss mit allgemeiner Hochschulreife
1992 - 1996	Besuch der Realschule in Uelsen
1990 - 1992	Besuch der Orientierungsstufe in Uelsen
1986 - 1990	Besuch der Grundschule in Uelsen