

Friedrich-Alexander-Universität Erlangen-Nürnberg

**Lehrstuhl für Multimediakommunikation und
Signalverarbeitung**

Prof. Dr.-Ing. André Kaup

Bachelorarbeit

**Analyse verschiedener Verfahren zur
Frameratenerhöhung von komprimierten
Videodaten unterschiedlicher zeitlicher
Auflösung**

von Eduard Reger

April 2016

Betreuer: Michel Bätz

Erklärung

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

Ort, Datum

Unterschrift

Inhaltsverzeichnis

Kurzfassung	III
Abstract	V
Abkürzungsverzeichnis	VII
1 Einleitung	1
2 Stand der Technik	3
2.1 Central Weighted Median	4
2.2 Verfahren von Sunwoo et al.	4
2.2.1 PMVS	5
2.2.2 PAMC	6
2.2.3 IPHI	6
2.3 HEVC	9
3 Versuchsaufbau	11
4 Ergebnisse	15
5 Fazit	21
A Tabellen	23
Abbildungsverzeichnis	27

Tabellenverzeichnis	29
Literaturverzeichnis	31

Kurzfassung

In dieser Arbeit werden verschiedene Algorithmen zur Frame Rate Up-Conversion (FRUC) untersucht. Sie werden auf Videosequenzen unterschiedlicher Auflösung angewandt und bei verschiedenen zeitlichen Auflösungen verglichen. Es werden zwei verschiedene Methoden verglichen, die Framerate auf einen Faktor von vier bzw. acht zu erhöhen. Hierbei werden die mittleren PSNR-Werte der jeweiligen Algorithmen sowie der PSNR-Verlauf der interpolierten Frames zwischen den Referenzframes verglichen.

Abstract

This thesis evaluates different frame rate up-conversion (FRUC) algorithms. These algorithms are applied to video sequences of differing spatial resolutions and compared for different temporal resolutions. Two methods of increasing the framerate to a factor of four or eight are compared. For this, the average PSNRs for the respective algorithms as well as the PSNR gradient between the non-interpolated reference frames are compared.

Abkürzungsverzeichnis

BMA	Block-Matching-Algorithmus
CWM	Central Weighted Median
FRUC	Frame Rate Up-Conversion
IPHI	Intrapredicted Hole Interpolation
JCT-VC	Joint Collaborative Team on Video Coding
MC-FRUC	Motion-compensated FRUC
MCI	Motion-Compensated Interpolation
ME	Motion Estimation
MV	Motion Vector
PAMC	Partial Average-based Motion Compensation
PMVS	Prediction-based Motion Vector Smoothing
QP	Quantization Parameter
SAD	Summe der absoluten Differenzen

Kapitel 1

Einleitung

Die Übertragung von Videodaten ist aus der heutigen Welt nicht mehr wegzudenken. Im Jahr 2015 machten in Nordamerika die Videostreaming-Dienste Netflix, YouTube, Amazon Video und Hulu bereits über 60 Prozent des Spitzen-Download-Internetverkehrs aus [1]. Hinzu kommt, dass Inhalte immer häufiger in HD- oder sogar 4K-Auflösung angeboten werden, sowohl online als auch offline. Da jedoch sowohl Speicher als auch Bitrate physikalisch begrenzt sind, ist es zumeist erforderlich, Videodaten vor der Übertragung zu komprimieren. Dies erfolgt mit Standards wie H.264/MPEG-AVC, seinem Nachfolger H.265/MPEG-HEVC, sowie dem von Google entwickelten VP9 [2].

Außerdem hat je nach Anwendungsgebiet das zu betrachtende Videomaterial, vor allem bei schnellen Bewegungen, eine unzureichend hohe zeitliche Auflösung und muss folglich einer Frameratenerhöhung (FRUC) unterzogen werden. Solche Fälle sind beispielsweise gegeben bei Videos, die zur Speicherbegrenzung häufig nur über eine geringe Framerate verfügen [3], beispielsweise Überwachungsaufnahmen, sowie bei Videos, auf denen Algorithmen zur Bewegungsverfolgung von Objekten oder Personen angewandt werden sollen. Außerdem kann mittels FRUC die Bewegungsunschärfe bei LCDs reduziert werden [4, 5]. Weiterhin kann FRUC bei der Konversion zwischen verschiedenen Videoformaten vonnöten sein [5, 6].

Diese Arbeit befasst sich mit verschiedenen FRUC-Algorithmen, angewandt auf diverse Videosequenzen, die alle jeweils mit Faktor zwei, vier und acht unterabgetastet sind.

Hierfür werden sowohl unkomprimierte als auch mit HEVC codierte Videos verwendet. Diese Arbeit gliedert sich wie folgt: Zunächst wird der Stand der Technik erläutert, einschließlich FRUC, Bewegungsschätzung, Bewegungskompensierung und HEVC. Anschließend wird der Versuchsaufbau präsentiert und schließlich die Ergebnisse gefolgt vom Fazit.

Kapitel 2

Stand der Technik

FRUC kann prinzipiell auf zwei Weisen realisiert werden: mittels relativ einfacher Verfahren wie Framewiederholung bzw. Averaging oder mit Motion-Compensated FRUC (MC-FRUC)[5]. Bei Framewiederholung wird jeder Frame so oft wiederholt, dass die erwünschte Framerate erreicht wird; beim Averaging bildet man einen Mittelwert zwischen dem vorherigen und nachfolgenden Frame. Sofern der interpolierte Frame nicht in der Mitte zwischen diesen zwei Frames ist, wird ein gewichteter Mittelwert gebildet, mit einem Gewichtungsfaktor α [6] gemäß der Formel

$$F_{avg} = (1 - \alpha)F_n + \alpha F_{n+1}$$

Dahingegen berücksichtigt MC-FRUC die Bewegung von Objekten und besteht aus zwei Stufen: Motion Estimation (ME) und Motion-Compensated Interpolation (MCI). Bei ME berechnet man Bewegungsvektoren (MVs), die die Bewegung von Objekten repräsentieren und nutzt diese anschließend bei MCI um den neuen Frame zu erstellen [5].

Ein häufig verwendeter Algorithmus zur Realisierung von ME ist der Block-Matching-Algorithmus (BMA). Hierbei wird das Bild in mehrere Blöcke unterteilt und anschließend deren Bewegungsframes ermittelt indem man innerhalb einer gewissen Suchweite im Nachbarframe einen möglichst ähnlichen Block findet [5, 6]. Als Fehlermaß zur Er-

mittlung des ähnlichsten Blocks wird üblicherweise die Summe der absoluten Differenzen (SAD) herangezogen [6]. BMA kann sowohl unilateral als auch bilateral umgesetzt werden. Bei unilateraler ME verlaufen die Bewegungsvektoren vom vorherigen zum nächsten (Vorwärts-ME) bzw. vom nächsten zum vorherigen Frame (Rückwärts-ME); bei bilateraler ME verlaufen jeweils zwei Vektoren, ausgehend vom zu ermittelnden Frame, in entgegengesetzter Richtung zum vorherigen und nächsten Frame [5].

2.1 Central Weighted Median

Bei der Interpolation mittels Central Wighted Median (CWM) ermittelt man den neuen Frame, indem man den Wert jedes Pixels sowie seiner horizontalen und evtl. vertikalen Nachbarn im vorherigen und folgenden Frame ermittelt und den Median bildet, wobei der zentrale Pixel jeweils stärker gewichtet wird als seine Nachbarn [6, 7]. Wenn man beispielsweise den zentralen Pixel mit Faktor sieben gewichtet und jeweils die drei linken und rechten Nachbarn betrachtet, wie in Abbildung 2.1 dargestellt, so werden 26 Werte zur Medianbildung verwendet [6, 7].



Abbildung 2.1: Bei CWM ist der mittlere Pixel stärker gewichtet als die anderen

2.2 Verfahren von Sunwoo et al.

Das in [5] vorgestellte FRUC-Verfahren von Sunwoo et al. besteht aus drei Algorithmen: Prediction-based Motion Vector Smoothing (PMVS), Partial Average-based Motion Compensation (PAMC) und Intrapredicted Hole Interpolation (IPHI).

Zunächst wird mittels Vorwärts- und Rückwärts-ME je ein Bewegungsvektorfeld generiert. Diese werden mittels PMVS geglättet um Ausreißer unter den Vektoren zu eliminieren. Daraufhin werden die Frames mittels PAMC einer MC unterzogen und zu

einem Frame zusammengefügt. Dabei entstandene werden mit IPHI gefüllt, welches auf der Intraprediktion von H.264/AVC basiert.

2.2.1 PMVS

PMVS glättet die Bewegungsvektoren folgendermaßen: Zunächst wird der Wert des prädizierten Vektors eines Blocks ermittelt, indem der Mittelwert aus den Nachbarn oben, links, und oben links gebildet wird (Abbildung 2.2). Anschließend werden aus den acht Nachbarn des prädizierten Vektors die drei mit der geringsten absoluten Differenz ermittelt. Als letztes ermittelt man den Mittelwert aus diesen drei Vektoren und dem prädizierten Vektor um den endgültigen Bewegungsvektor zu erhalten (Abbildung 2.3). [5]

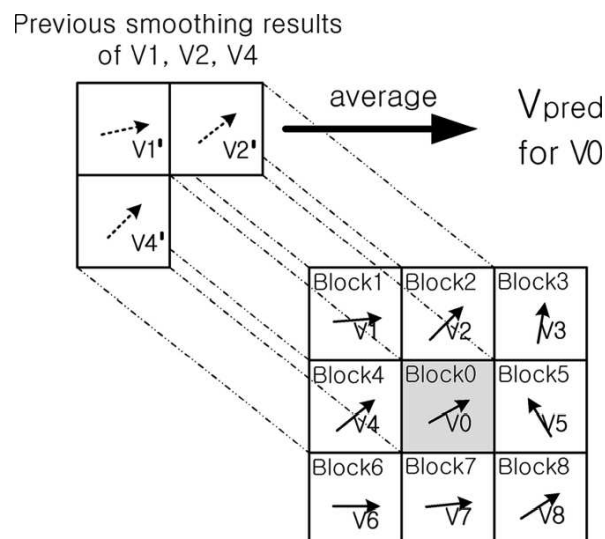


Abbildung 2.2: Prädizierter Bewegungsvektor [5]

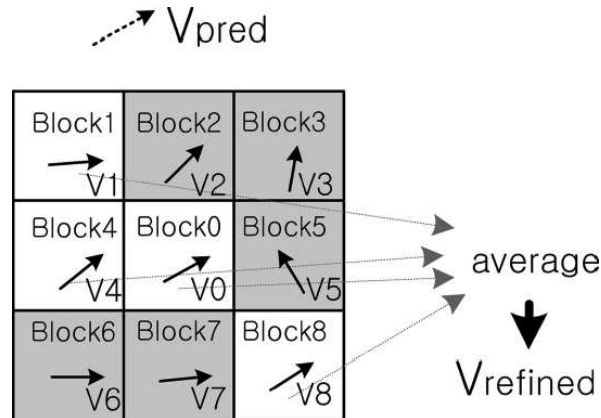


Abbildung 2.3: Geglätteter Vektor [5]

2.2.2 PAMC

Unter Verwendung der mittels PMVS ermittelten MVs und der Formel

$$F(x, y) = \frac{1}{2} \left(F_n \left(x - \frac{v_x}{2}, y - \frac{v_y}{2} \right) + F_{n-1} \left(x + \frac{v_x}{2}, y + \frac{v_y}{2} \right) \right), \quad (2.1)$$

berechnet man den vorwärts-interpolierten Frame, wobei (x, y) die Position der Pixel und (v_x, v_y) die MVs darstellen. F_n und F_{n-1} sind der aktuelle bzw. der Vergleichsframe. Die Rückwärtsinterpolation erfolgt analog.

Bei eventuell auftretenden Überlappungen werden die SAD-Werte der überlappenden Blöcke verglichen. Ist wie in Abbildung 2.4 der Wert für Block A am geringsten, wird der Durchschnitt für die überlappten Pixel in Region 1 gebildet, aber die Pixel von A im zentralen Bereich (weiß) beibehalten [5].

2.2.3 IPHI

IPHI betrachtet vier Richtungsmodi innerhalb von 4×4 -Pixel-Blöcken unter Verwendung der Nachbarpixel u_0 - u_7 , l_0 - l_3 und M gemäß Abbildung 2.5. Hierbei werden die Prädizierten Pixel für die verschiedenen Richtungsmodi anhand der Gleichungen in Tabelle 2.1 berechnet. Als nächstes wird der SAD-Wert zwischen jenen Pixeln, die keine Löcher sind, und den prädizierten Pixeln der verschiedenen Modi berechnet. Der Modus mit dem geringsten SAD-Wert wird als Blockorientierung ausgewählt. Löcher werden

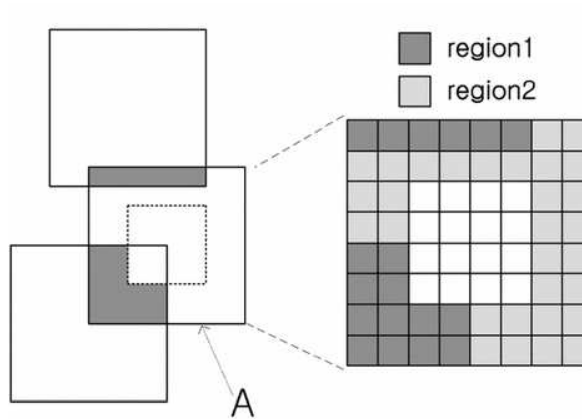


Abbildung 2.4: Vorgehensweise bei Überlappungen [5]

anschließend interpoliert, indem man aus ihren acht Nachbarpixeln einen gewichteten Mittelwert bildet wobei, die Nachbarn, die gemäß dem ermittelten Richtungsmodus orientiert sind, vierfach gewichtet werden. Abbildung 2.6 zeigt dies am Beispiel von horizontaler Interpolation. In diesem Fall würde man die Pixel D und E vierfach gewichten. [5]

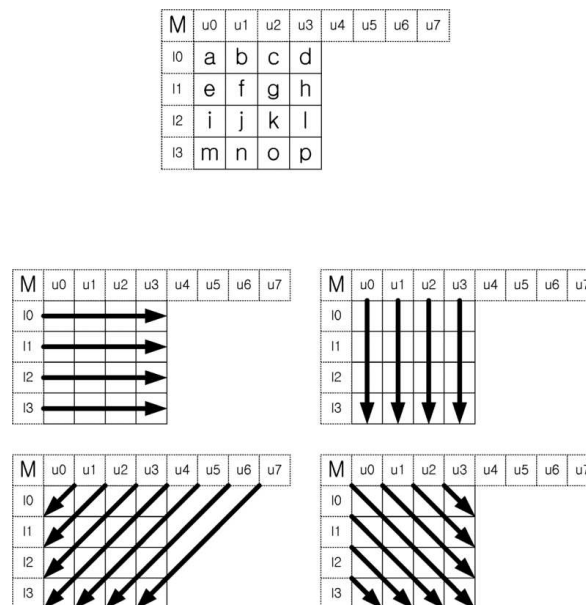


Abbildung 2.5: Richtungsmodi von IPHI [5]

Tabelle 2.1: Prädiktionsgleichungen [5]

Richtungsmodus	Prädiktionsgleichung
Vertikal	$A = E = I = M = U0$ $B = I = J = N = U1$ $C = G = K = O = U2$ $D = H = L = P = U3$
Horizontal	$A = B = C = D = L0$ $E = F = G = H = L1$ $I = J = K = L = L2$ $M = N = O = P = L3$
Unten-links	$A = (U0 + U1 + U1 + U2 + 2) \gg 2$ $B = E = (U1 + U2 + U2 + U3 + 2) \gg 2$ $C = F = I = (U2 + U3 + U3 + U4 + 2) \gg 2$ $D = G = J = M = (U3 + U4 + U4 + U5 + 2) \gg 2$ $H = K = N = (U4 + U5 + U5 + U6 + 2) \gg 2$ $L = O = (U5 + U6 + U6 + U7 + 2) \gg 2$ $P = (U6 + U7 + U7 + U7 + 2) \gg 2$
Unten-rechts	$D = (U1 + U2 + U2 + U3 + 2) \gg 2$ $C = H = (U0 + U1 + U1 + U2 + 2) \gg 2$ $B = G = L = (M + U0 + U0 + 1 + 2) \gg 2$ $A = J = K = P = (U0 + M + M + L0 + 2) \gg 2$ $E = J = O = (M + L0 + L0 + L1 + 2) \gg 2$ $I = N = (L0 + L1 + L1 + L2 + 2) \gg 2$ $M = (L1 + L2 + L2 + L3 + 2) \gg 2$

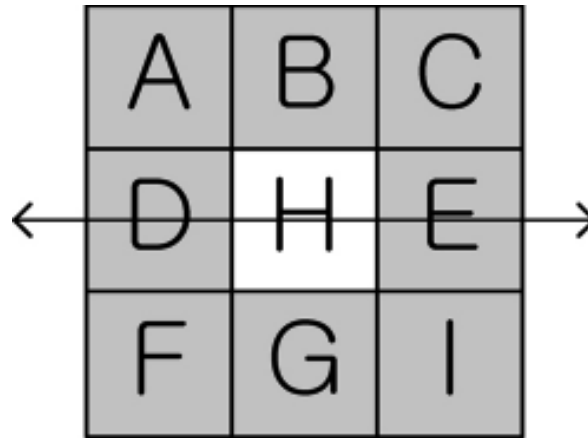


Abbildung 2.6: Horizontale Lochinterpolation [5]

2.3 HEVC

Der internationale Videocodingstandard HEVC wurde vom Joint Collaborative Team on Video Coding (JCT-VC) als Nachfolger von H.264/AVC entwickelt [8, 9, 10] und erreicht bei gleichbleibender subjektiver Bildqualität eine Bitratensparnis von ca. 50% [8, 11, 12].

HEVC verwendet I-, P- und B-Frames [8, 9]. I-frames (Intra Frames) sind Frames, die unabhängig von anderen Frames codiert sind. Sie dienen als Startpunkte falls der übertragene Bitstream fehlerhaft ist oder dienen dem Vor- und Zurückspulen sowie Szenenwechseln [8].

Ein P-Frame (Predictive Frame) ist unter Ausnutzung temporaler Redundanz in Abhängigkeit vorheriger Frames codiert [8, 9].

Ein B-Frame (Bi-predictive Frame) nimmt sowohl auf vorherige als auch auf nachfolgende Frames Bezug [8, 9].

Wenn es sich bei allen Frames um I-Frames handelt, nennt man diese Codingstruktur Intra-only; wenn nur der erste Frame ein I-Frame ist, handelt es sich um Low Delay und wenn I-Frames in regelmäßigen Abständen vorhanden sind, dann ist von Random Access die Rede [9].

Das Repräsentieren einer großen Menge von Werten mit einer begrenzten Anzahl an Bits nennt man Quantisierung [9, 13]. In HEVC ist der Grad der Quantisierung definiert über den Quantisierungsparameter (QP) [9, 10]. Der QP kann Werte zwischen 0 und 51 einnehmen [10], wobei höhere Werte eine höhere Quantisierungsschrittgröße, und somit einen höheren Qualitätsverlust, bedeuten [9, 10]. Eine Erhöhung des QP um sechs verdoppelt die Schrittgröße [10].

Kapitel 3

Versuchsaufbau

Es wurden neun Videos verschiedener Auflösung verwendet, siehe Tabelle 3.1. Von diesen Videos wurden Sequenzen, die aus den ersten 41 Frames bestehen, unkomprimiert jeweils mit Faktor zwei, vier und acht unterabgetastet und die fehlenden Frames durch interpolierte Frames ersetzt. Abbildung 3.1 zeigt dies am Beispiel von Faktor vier.

Für Faktor vier und acht erfolgte die Interpolation auf zwei verschiedene Weisen:

- Variation des Gewichtungsfaktors α entsprechend der Position des Frames zwischen den Referenzframes
- Rekursives Interpolieren zwischen zwei bereits bekannten Frames bei konstantem $\alpha = 0.5$. Abbildung 3.2 zeigt die Reihenfolge, in der die Interpolation erfolgt.

Die verwendeten FRUC-Algorithmen sind:

- CWM mit einer Gewichtung von sieben für den zentralen Pixel und eins für die jeweils nächsten drei Pixel links und rechts
- Bewegungskompensiertes CWM (MC-CWM)
- Bewegungskompensiertes Averaging (MC-AVG)
- Das Verfahren von Sunwoo [5]

Darüberhinaus wurden die selben unterabgetasteten Sequenzen mittels HEVC komprimiert und anschließend wie die unkomprimierten Sequenzen einer Bildratenerhöhung unterzogen. Abbildung 3.3 zeigt dies für Faktor vier. Es wurde ein Random Access-Codierung verwendet und QP-Werte von 20 und 30.

Für ME wurde ein Block-Matching-Algorithmus verwendet, mit Blockgröße 8×8 und Suchweite 32. Bei der SAD-Berechnung wurde die Blockgröße zusätzlich noch um eine Spanne von vier erweitert um einen exakteren Wert zu erhalten.

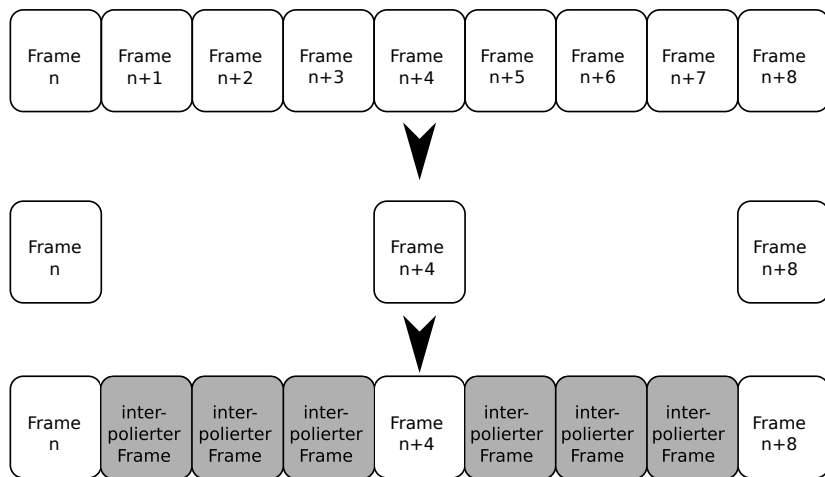


Abbildung 3.1: Interpolation von mit Faktor vier unterabgetastetem Video

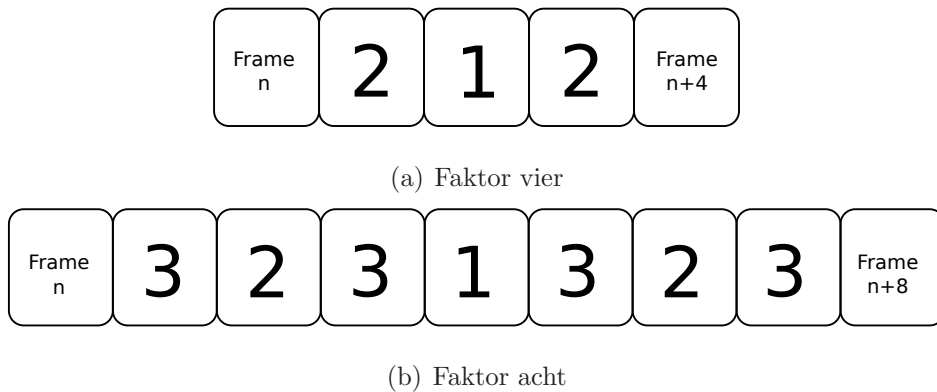


Abbildung 3.2: Reihenfolge bei rekursiver Interpolation

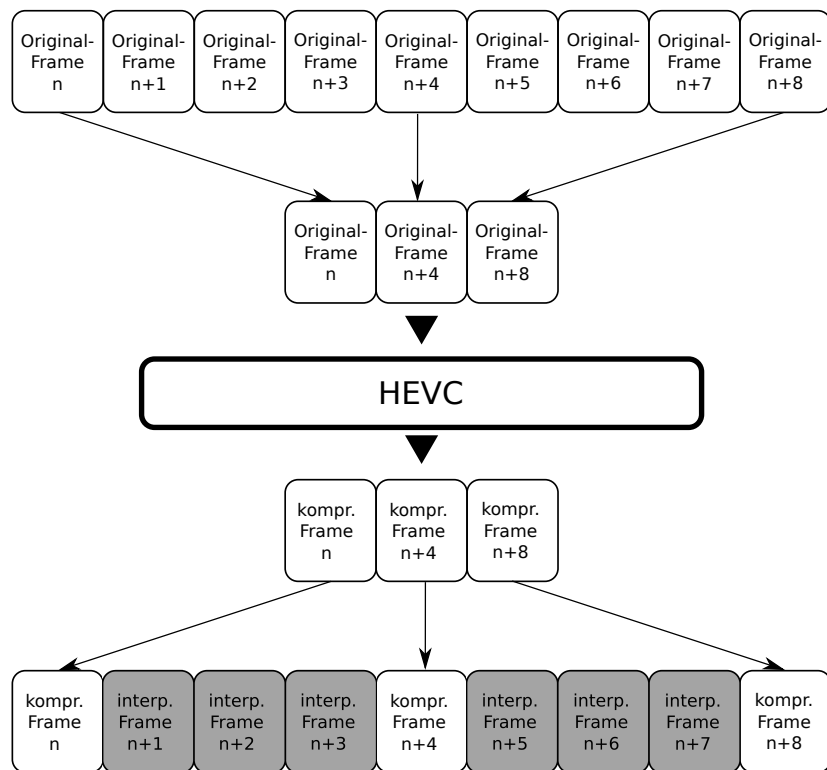


Abbildung 3.3: Kompression und anschließendes FRUC

Tabelle 3.1: Verwendete Sequenzen

Sequenz	Auflösung	Framerate
BasketballPass	416×240	50
BlowingBubbles	416×240	50
BQSquare	416×240	60
RaceHorses(1)	416×240	30
BQMall	832×480	60
PartyScene	832×480	50
RaceHorses(2)	832×480	30
BQTerrace	1920×1080	60
Cactus	1920×1080	50

Kapitel 4

Ergebnisse

Tabelle 4.1 zeigt den Vergleich zwischen den verschiedenen Algorithmen für die unkomprimierten und komprimierten Sequenzen, wobei im unkomprimierten Fall nur die PSNR-Werte der interpolierten Frames in den Durchschnitt einfließen, um unendlich hohe Werte zu vermeiden. Sunwoo schneidet in allen Fällen am besten ab und CWM am schlechtesten, mit einer Differenz von 1.04 dB für unkomprimierte und 0.15 bzw. 0.08 dB zum zweitbesten Algorithmus (MC-AVG) für die komprimierten Sequenzen.

Tabelle 4.2 zeigt die durchschnittlichen PSNR-Werte von unkomprimierten mit Faktor vier unterabgetasteten Sequenzen für den Fall der α -Variation und den rekursiven Fall. Tabellen A.1 und A.2 zeigen die entsprechenden Werte für QP = 20 bzw 30. Allen drei Tabellen ist zu entnehmen, dass die rekursive Variante um bis zu 0.22 dB (unkomprimiert) bzw. 0.14 dB (QP=20) und 0.11 dB (QP=30) besser abschneidet, außer für MC-CWM, wo die α -Variante um 0.17 dB bzw. 0.12 dB und 0.09 dB überlegen ist.

Analog dazu enthalten die Tabellen 4.3 sowie A.3 und A.4 die entsprechenden Werte für Faktor acht bei nicht komprimierten und mit einem Quantisierungsparameter von 20 bzw. 30 codierten Sequenzen. Auch hier ist die rekursive Variante außer bei MC-CWM leicht überlegen.

Tabelle 4.1: PSNR-Vergleich von unkomprimierten sowie mit QP = 20 bzw. 30 komprimierten Sequenzen

Sequenz	Unkomprimiert				QP=20				QP=30			
	Sunwoo	CWM	MC-CWM	MC-AVG	Sunwoo	CWM	MC-CWM	MC-AVG	Sunwoo	CWM	MC-CWM	MC-AVG
BasketballPass	38.1522	31.1671	35.4703	38.0166	39.9396	36.9480	38.8808	39.8556	34.8381	32.9305	34.3098	34.7861
BlowingBubbles	30.5693	27.9187	28.9985	30.2148	34.9468	33.8248	34.3056	34.8048	30.7254	30.0308	30.3957	30.6391
BQSquare	36.7229	32.5370	29.1974	34.3556	37.7137	36.1002	34.5877	36.7514	32.1541	31.4396	30.4272	31.6860
RaceHorses(1)	28.1442	21.5001	28.2048	28.5915	34.4211	31.2368	34.4494	34.6344	30.0450	27.1449	30.0780	30.2164
BQMall	32.2937	29.2703	31.4764	32.3578	36.4367	35.0516	36.0620	36.4600	32.9709	31.8734	32.7271	32.9831
PartyScene	36.1070	27.6346	27.4021	29.0427	34.9258	34.0691	33.5571	34.6550	30.1373	29.5361	29.2612	29.9384
RaceHorses(2)	26.8255	20.2154	27.0047	27.1053	33.7032	30.4705	33.7690	33.8216	29.7880	26.7169	29.8653	29.8917
BQTerrace	31.4573	29.0106	30.1870	31.1976	35.9311	34.5642	35.2021	35.7106	32.7928	31.4943	32.1785	32.5951
Cactus	32.5605	28.3714	32.0870	32.5887	36.0189	34.0060	35.7569	36.0002	33.4999	31.8678	33.3289	33.4845
Mittelwert	32.5370	27.5139	30.0031	31.4967	36.0041	34.0301	35.1745	35.8548	31.8835	30.3371	31.3969	31.8023

In Tabelle 4.4 sind die mittleren PSNR-Werte für die interpolierten Frames $n+1$, $n+2$ und $n+3$ bei Unterabtastung von Faktor vier angegeben, wobei n (und $n+4$) der Position der Originalframes entspricht. Man sieht, dass Frames $n+1$ und $n+3$ in jedem Fall einen höheren durchschnittlichen PSNR-Wert haben als der mittlere Frame, was wahrscheinlich daran liegt, dass sie nicht nur näher an einem Referenzframe liegen, sondern dieser auch stärker in die Gewichtung eingeht, so dass eine höhere Ähnlichkeit zum ursprünglich an dieser Stelle vorhandenen Frame erzielt wird.

Tabellen A.5 und A.6 zeigen analog zu oben die mittleren PSNR-Werte der Frames $n+1$ bis $n+7$ für Faktor 8. In Bild 4.1 zeigt sind die Werte für Sunwoo und MC-CWM graphisch dargestellt. Man erkennt, dass bei MC-CWM im rekursiven Fall der mittlere Frame einen (um bis zu 0.26 dB) höheren PSNR-Wert hat als seine Nachbarn. Abgesehen davon nehmen die Frames in der Mitte den geringsten Wert ein, während sie nach außen hin eine bessere PSNR-Performance erreichen.

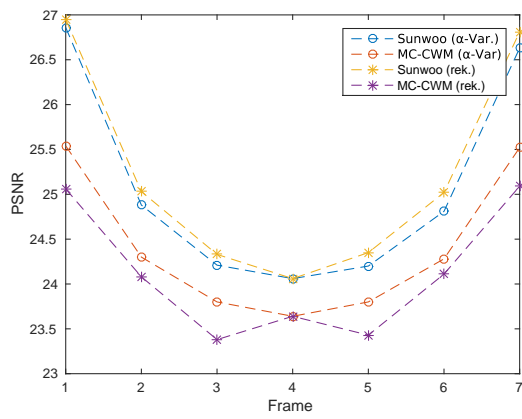
Alle betrachteten Fälle haben gemeinsam, dass Sunwoo am besten abschneidet, während MC-AVG an zweiter Stelle liegt. CWM erreicht in allen betrachteten Fällen den geringsten PSNR-Wert.

Tabelle 4.2: PSNR-Werte von mit Faktor 4 unterabgetasteten unkomprimierten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind

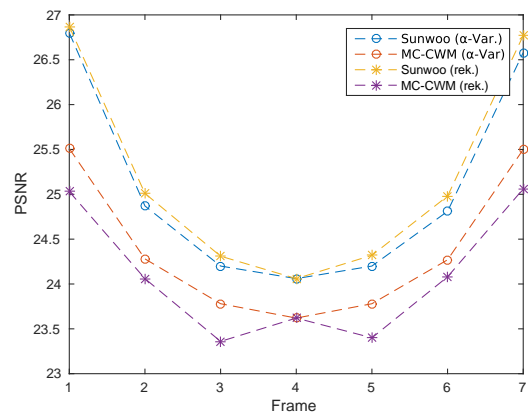
Sequenz	Sunwoo		CWM		MC-CWM		MC-AVG	
	α	rekursiv	α	rekursiv	α	rekursiv	α	rekursiv
BasketballPass	34.0092	33.9647	28.9074	28.9301	32.6630	32.3992	33.5479	33.9306
BlowingBubbles	27.8590	27.9034	25.7273	25.7390	26.9651	26.7429	27.5265	27.6386
BQSquare	33.2416	33.0007	27.6880	27.8918	27.6653	26.8938	30.7997	30.6365
RaceHorses (1)	23.7385	23.9158	19.0281	19.3412	24.1743	24.3097	23.9954	24.3628
BQMall	29.3061	29.4188	25.4905	25.9759	28.9221	29.0203	29.1104	29.5475
PartyScene	26.7038	26.8809	24.7791	24.9649	25.3310	25.0428	26.1916	26.4254
RaceHorses (2)	20.6975	21.1657	18.2069	18.4872	20.9253	21.3654	20.6198	21.1896
BQTerrace	30.1719	30.1314	26.2992	26.0007	29.2814	28.6563	30.0188	29.8354
Cactus	29.6120	29.7483	26.1804	26.3342	29.4721	29.4295	29.5889	29.7888
Mittelwert	28.3711	28.4589	24.7008	24.8517	27.2666	27.0956	27.9332	28.1506

Tabelle 4.3: PSNR-Werte von mit Faktor 8 unterabgetasteten unkomprimierten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind

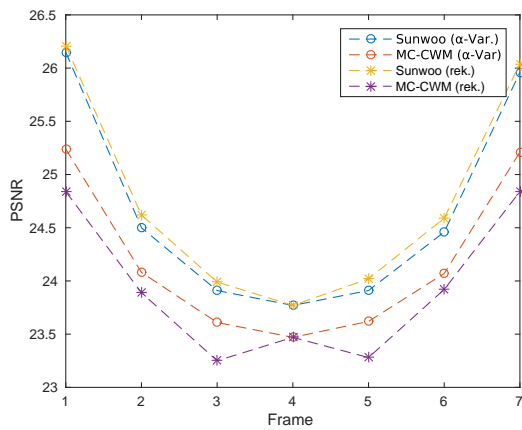
Sequenz	Sunwoo		CWM		MC-CWM		MC-AVG	
	α	rekursiv	α	rekursiv	α	rekursiv	α	rekursiv
BasketballPass	29.1437	29.4277	26.8752	26.9962	28.6130	28.6584	28.5774	29.1836
BlowingBubbles	25.2656	25.3851	24.0600	23.9345	25.0938	24.9591	25.1206	25.3947
BQSquare	30.6829	30.1233	24.3123	24.0812	26.7786	25.1480	29.1035	28.2734
RaceHorses (1)	18.1758	18.6456	16.8287	17.2134	18.2274	18.4507	17.9721	18.4078
BQMall	24.6855	25.1098	22.1351	22.9847	24.6114	24.9444	24.4048	25.1252
PartyScene	24.2247	24.5487	22.4326	22.5592	23.5147	23.1598	23.8801	24.1670
RaceHorses (2)	16.8320	17.2122	16.3518	16.6947	16.9117	17.1035	16.7428	17.0581
BQTerrace	30.1509	29.6132	24.3390	23.8135	29.2715	27.8897	29.9426	29.2789
Cactus	26.6691	26.9461	24.4035	24.7216	26.6733	26.7153	26.5356	26.9304
Mittelwert	25.0923	25.2235	22.4153	22.5554	24.4106	24.1143	24.6977	24.8688



(a) unkomprimiert



(b) QP=20



(c) QP=30

Abbildung 4.1: Vergleich der PSNR-Werte interpolierter Frames

Tabelle 4.4: PSNR-Vergleich für interpolierte Frames zwischen Referenzframes n und $n+4$ (Faktor 4)

Frame		unkomprimiert			QP=20			QP=30		
		n+1	n+2	n+3	n+1	n+2	n+3	n+1	n+2	n+3
Sunwoo	α -Var.	28.70	27.70	28.71	28.58	27.61	28.59	27.74	26.99	27.75
	rekursiv	28.84	27.71	28.84	28.68	27.61	28.70	27.83	26.99	27.83
CWM	α -Var.	24.87	24.34	24.90	24.85	24.32	24.88	24.64	24.15	24.67
	rekursiv	25.10	24.34	25.12	25.08	24.32	25.09	24.86	24.15	24.87
MC-CWM	α -Var.	27.39	26.93	27.48	27.33	26.87	27.42	26.90	26.48	26.96
	rekursiv	27.15	26.93	27.20	27.10	26.87	27.16	26.72	26.48	26.76
MC-AVG	α -Var.	28.16	27.46	28.18	28.05	27.36	28.07	27.39	26.81	27.38
	rekursiv	28.51	27.46	28.48	28.35	27.36	28.34	27.63	26.81	27.60

Kapitel 5

Fazit

Es wurde gezeigt, dass Sunwoos Methode von den getesteten Algorithmen hinsichtlich PSNR-Performance am besten anschnidet. Ferner wurde gezeigt, dass rekursive erfolgreiche Interpolation von mit Faktor vier und acht unterabgetasteten Sequenzen für die Algorithmen Sunwoo, MC-AVG und CWM besser abschnidet als bei direkter Interpolation durch Variation des Gewichtungsfaktors α . Weiterhin konnte konstatiert werden, dass in beiden Fällen bei Faktor vier und acht in den betrachteten Fällen der PSNR-Wert in der Mitte am geringsten ist und nach außen hin zunimmt, außer bei MC-CWM im rekursiven Fall für Faktor acht.

Anhang A

Tabellen

Tabelle A.1: PSNR-Werte von mit Faktor 4 unterabgetasteten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind (QP=20)

Sequenz	Sunwoo		CWM		MC-CWM		MC-AVG	
	α	rekursiv	α	rekursiv	α	rekursiv	α	rekursiv
BasketballPass	35.9886	35.9717	32.5039	32.5182	35.1219	34.9522	35.6773	35.9233
BlowingBubbles	30.8395	30.8678	29.4186	29.4223	30.2562	30.1079	30.6117	30.6730
BQSquare	34.6370	34.4506	30.9109	31.0512	30.8802	30.3396	32.9957	32.8863
RaceHorses (1)	28.2450	28.3562	24.7951	25.0271	28.5526	28.6581	28.4349	28.6873
BQMall	32.2787	32.3623	29.5246	29.8791	31.9850	32.0584	32.1074	32.4126
PartyScene	30.0148	30.1282	28.6637	28.7950	29.0400	28.8374	29.6383	29.7958
RaceHorses (2)	25.9986	26.3408	24.1730	24.3805	26.1648	26.4880	25.9404	26.3522
BQTerrace	33.0362	32.9516	30.0038	29.7810	32.2635	31.7614	32.8276	32.6313
Cactus	32.2695	32.3448	29.7679	29.8784	32.1290	32.0925	32.2270	32.3696
Mittelwert	31.4787	31.5304	28.8624	28.9703	30.7104	30.5884	31.1622	31.3035

Tabelle A.2: PSNR-Werte von mit Faktor 4 unterabgetasteten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind (QP=30)

Sequenz	Sunwoo		CWM		MC-CWM		MC-AVG	
	α	rekursiv	α	rekursiv	α	rekursiv	α	rekursiv
BasketballPass	32.9102	32.8982	30.2148	30.2273	32.3205	32.1976	32.6715	32.8073
BlowingBubbles	28.2693	28.3025	27.3300	27.3331	27.9604	27.8604	28.1428	28.2001
BQSquare	30.8150	30.7039	28.4252	28.5377	28.3954	27.9667	29.8209	29.7644
RaceHorses (1)	25.9601	26.0515	22.5888	22.8251	26.1864	26.2978	26.0743	26.3286
BQMall	30.2381	30.3027	27.7788	28.1153	30.0436	30.1032	30.1062	30.3399
PartyScene	27.4732	27.5447	26.3745	26.4898	26.6976	26.5355	27.1714	27.2976
RaceHorses (2)	23.9272	24.2470	22.1137	22.3274	24.0775	24.4035	23.8568	24.2564
BQTerrace	31.1454	30.9977	28.3208	28.1064	30.4953	29.9819	30.9765	30.7163
Cactus	30.7637	30.8203	28.5579	28.6690	30.6572	30.6364	30.7205	30.8319
Mittelwert	29.0558	29.0965	26.8560	26.9590	28.5371	28.4426	28.8379	28.9492

Tabelle A.3: PSNR-Werte von mit Faktor 8 unterabgetasteten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind (QP=20)

Sequenz	Sunwoo		CWM		MC-CWM		MC-AVG	
	α	rekursiv	α	rekursiv	α	rekursiv	α	rekursiv
BasketballPass	30.9530	31.1732	29.0951	29.1980	30.5367	30.5713	30.5030	30.9827
BlowingBubbles	27.3012	27.4198	26.3058	26.1966	27.1491	27.0415	27.1700	27.3766
BQSquare	31.7896	31.3825	26.5761	26.3805	28.6264	27.2731	30.4859	29.8100
RaceHorses (1)	21.4606	21.8202	20.2928	20.6252	21.4616	21.6539	21.2560	21.6232
BQMall	27.0152	27.3808	24.8015	25.5307	26.9134	27.1906	26.7361	27.3419
PartyScene	26.3516	26.6245	24.8794	24.9869	25.7819	25.4861	26.0768	26.3298
RaceHorses (2)	20.3109	20.6295	19.8883	20.1835	20.3615	20.5182	20.2224	20.4865
BQTerrace	31.8161	31.2353	26.6581	26.2035	30.9605	29.6682	31.5599	30.8177
Cactus	28.5743	28.7879	26.6257	26.9008	28.5323	28.5662	28.4342	28.7705
Mittelwert	27.2858	27.3837	25.0136	25.1340	26.7026	26.4410	26.9383	27.0599

Tabelle A.4: PSNR-Werte von mit Faktor 8 unterabgetasteten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind (QP=30)

Sequenz	Sunwoo		CWM		MC-CWM		MC-AVG	
	α	rekursiv	α	rekursiv	α	rekursiv	α	rekursiv
BasketballPass	29.3891	29.5418	27.7542	27.8524	29.1310	29.1786	29.0992	29.4645
BlowingBubbles	25.8729	25.9418	25.1172	25.0266	25.7942	25.7329	25.7899	25.9606
BQSquare	29.0744	28.7279	25.2624	25.0916	26.9790	25.8549	28.3021	27.8055
RaceHorses (1)	20.1983	20.5675	19.0348	19.3827	20.1972	20.3855	20.0041	20.3666
BQMall	25.8995	26.2221	23.8110	24.5285	25.8471	26.1005	25.6616	26.1932
PartyScene	24.9171	25.1277	23.6327	23.7484	24.4322	24.1823	24.6674	24.8711
RaceHorses (2)	19.0876	19.3871	18.6942	19.0042	19.1544	19.3091	19.0096	19.2691
BQTerrace	30.5546	29.9370	25.7156	25.2789	29.8115	28.5792	30.3336	29.5548
Cactus	27.7421	27.9187	25.9013	26.1814	27.7073	27.7457	27.6116	27.8958
Mittelwert	25.8595	25.9302	23.8804	24.0105	25.4504	25.2299	25.6088	25.7090

Tabelle A.5: PSNR-Vergleich für interpolierte Frames zwischen Referenzframes n und $n+7$ (Faktor 8, unkomprimiert)

Frame		n+1	n+2	n+3	n+4	n+5	n+6	n+7
Sunwoo	α -Var.	26.86	24.88	24.21	24.06	24.20	24.81	26.63
	rekursiv	26.95	25.03	24.33	24.06	24.35	25.02	26.81
CWM	α -Var.	23.64	22.28	21.76	21.65	21.75	22.27	23.56
	rekursiv	23.81	22.46	21.88	21.65	21.88	22.44	23.76
MC-CWM	α -Var.	25.54	24.30	23.80	23.64	23.80	24.28	25.52
	rekursiv	25.06	24.08	23.38	23.64	23.43	24.11	25.09
MC-AVG	α -Var.	26.00	24.58	24.01	23.82	24.00	24.54	25.93
	rekursiv	26.57	24.68	23.92	23.83	23.95	24.68	26.46

Tabelle A.6: PSNR-Vergleich für interpolierte Frames zwischen Referenzframes n und $n+7$ (Faktor 8, komprimiert)

QP		20							30						
Frame		n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+1	n+2	n+3	n+4	n+5	n+6	n+7
Sunwoo	α -Var.	26.80	24.87	24.20	24.06	24.20	24.81	26.58	26.15	24.50	23.91	23.77	23.91	24.46	25.96
	rekursiv	26.87	25.01	24.31	24.06	24.32	24.98	26.77	26.21	24.62	23.99	23.77	24.02	24.59	26.04
CWM	α -Var.	23.62	22.28	21.76	21.65	21.76	22.27	23.55	23.48	22.24	21.74	21.64	21.74	22.22	23.38
	rekursiv	23.80	22.46	21.88	21.65	21.88	22.44	23.76	23.70	22.42	21.86	21.64	21.85	22.39	23.64
MC-CWM	α -Var.	25.51	24.28	23.78	23.62	23.78	24.27	25.50	25.24	24.08	23.61	23.47	23.62	24.07	25.21
	rekursiv	25.03	24.06	23.36	23.62	23.40	24.08	25.06	24.84	23.89	23.25	23.47	23.28	23.92	24.84
MC-AVG	α -Var.	25.96	24.55	23.98	23.80	23.98	24.51	25.89	25.54	24.27	23.76	23.60	23.76	24.23	25.44
	rekursiv	26.47	24.64	23.87	23.80	23.90	24.62	26.37	25.96	24.36	23.67	23.60	23.70	24.33	25.80

Abbildungsverzeichnis

2.1	Bei CWM ist der mittlere Pixel stärker gewichtet als die anderen	4
2.2	Prädizierter Bewegungsvektor [5]	5
2.3	Geglätteter Vektor [5]	6
2.4	Vorgehensweise bei Überlappungen [5]	7
2.5	Richtungsmodi von IPHI [5]	7
2.6	Horizontale Lochinterpolation [5]	9
3.1	Interpolation von mit Faktor vier unterabgetastetem Video	12
3.2	Reihenfolge bei rekursiver Interpolation	12
3.3	Kompression und anschließendes FRUC	13
4.1	Vergleich der PSNR-Werte interpolierter Frames	18

Tabellenverzeichnis

2.1	Prädiktionsgleichungen [5]	8
3.1	Verwendete Sequenzen	13
4.1	PSNR-Vergleich von unkomprimierten sowie mit QP = 20 bzw. 30 komprimierten Sequenzen	16
4.2	PSNR-Werte von mit Faktor 4 unterabgetasteten unkomprimierten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind	17
4.3	PSNR-Werte von mit Faktor 8 unterabgetasteten unkomprimierten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind	17
4.4	PSNR-Vergleich für interpolierte Frames zwischen Referenzframes n und n+4 (Faktor 4)	19
A.1	PSNR-Werte von mit Faktor 4 unterabgetasteten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind (QP=20)	24
A.2	PSNR-Werte von mit Faktor 4 unterabgetasteten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind (QP=30)	24
A.3	PSNR-Werte von mit Faktor 8 unterabgetasteten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind (QP=20)	25
A.4	PSNR-Werte von mit Faktor 8 unterabgetasteten Sequenzen, die mit α -Variation bzw. rekursiv interpoliert worden sind (QP=30)	25
A.5	PSNR-Vergleich für interpolierte Frames zwischen Referenzframes n und n+7 (Faktor 8, unkomprimiert)	26

A.6 PSNR-Vergleich für interpolierte Frames zwischen Referenzframes n und n+7 (Faktor 8, komprimiert)	26
--	----

Literaturverzeichnis

- [1] Netflix, Inc., “Q4 15 Letter to shareholders,” Januar 2016.
- [2] D. Grois, D. Marpe, A. Mulyoff, B. Itzhaky, and O. Hadar, “Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders,” in *Proc. Picture Coding Symposium (PCS)*, Dezember 2013, pp. 394–397.
- [3] I. Hwang, H. S. Jung, and M. H. Sunwoo, “Novel frame rate up-conversion algorithm based on prediction and recursive search,” in *Signal Processing Systems (SiPS), 2015 IEEE Workshop on*, Oktober 2015, pp. 1–4.
- [4] Y. Guo, L. Chen, Z. Gao, and X. Zhang, “Frame Rate Up-Conversion Using Linear Quadratic Motion Estimation and Trilateral Filtering Motion Smoothing,” *Journal of Display Technology*, vol. 12, no. 1, pp. 89–98, Januar 2016.
- [5] U. S. Kim and M. H. Sunwoo, “New frame rate up-conversion algorithms with low computational complexity,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 384–393, März 2014.
- [6] G. de Haan, *Digital Video Post Processing*. Den Haag: Eigenverlag, 2006.
- [7] S. Dikbas, “A low-complexity approach for motion-compensated video frame rate up-conversion,” Ph.D. dissertation, Georgia Institute of Technology, 2011.
- [8] Elemental Technologies, Inc., “HEVC demystified - a primer on the H.265 video codec,” 2013.

- [9] A. Scaccialepre, “Design and Implementation of a Fast HEVC Random Access Video Encoder,” Master’s thesis, KTH Royal Institute of Technology, Stockholm, März 2014.
- [10] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dezember 2012.
- [11] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, “Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC),” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, Dezember 2012.
- [12] B. Min and R. C. C. Cheung, “A Fast CU Size Decision Algorithm for the HEVC Intra Encoder,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 892–896, Mai 2015.
- [13] K. Sayood, *Introduction to Data Compression*, 3rd ed. Morgan Kaufmann, 2006.