Formangepasste diskrete Cosinus-Transformation für die Prädiktionsverbesserung im HEVC

Dipl.-Ing. (FH) Eugen Wige, Andreas Heindel, Prof. Dr.-Ing. André Kaup; {wige, kaup}@LNT.de
Lehrstuhl für Multimediakommunikation und Signalverarbeitung,
Universität Erlangen-Nürnberg, Cauer. 7, 91058 Erlangen

Kurzfassung

Um die Kompression in der Videocodierung zu verbessern, führen wir eine explizite Referenzbildentrauschung in die Codierschleife eines Videocodecs ein. Motiviert durch den Gedanken, dass die Leistung des Prädiktionsfehlers höher sein kann, falls Rauschen in dem zu codierenden Video vorhanden ist, wird die Bewegungskompensation durch die eingeführten Module verbessert. Es wird gezeigt wie man einen solchen Ansatz für die Codierung bei sehr kleinen Einstellungen des Quantisierungsparameters aber auch bei sehr groben Quantisierungseinstellungen verwenden kann. Die entwickelten Algorithmen wurden in der Referenzsoftware des aktuellen HEVC-Standards getestet. Die Simulationsergebnisse zeigen, dass mit der vorgeschlagenen Vorgehensweise maximale Bitratenersparnisse von bis zu 10 % für niedrige als auch hohe Quantisierungsparametereinstellungen erreicht werden können. Im Durchschnitt wurden Bitratenersparnisse von 7 % für hohe Qualität und 5 % für niedrige Qualität bei Codierung der ClassB-Sequenzen erreicht.

1. Einleitung

In der Videoverarbeitung spielt die Kompression eine wesentliche Rolle. Bei professionellen Anwendungen (z.B. medizinische Anwendungen) muss das Videomaterial möglichst ohne visuelle Verluste archiviert werden. Die unkomprimierte Speicherung der Daten würde enorme Speicherkapazitäten benötigen, weshalb eine Kompression unabdingbar ist. Dabei ist die Berücksichtigung des Rauschens in dem zu codierenden Video für eine effiziente Kompression wichtig. Es hat sich herausgestellt, dass die Prädiktion zur verlustlosen oder nahezu verlustlosen Codierung von Videos verbessert werden kann, wenn das Referenzbild rauschgefiltert wird [1][2]. Dazu wurde ein Schleifenfilter in den Codec eingebracht, das speziell für die Prädiktion nützlich ist. Dieses Filter agiert zusätzlich zu den in den Videocodierstandards vorhandenen Schleifenfiltern, die zur Verbesserung des Ausgangsbildes eingesetzt werden.

In dieser Arbeit untersuchen wir die formangepasste diskrete Cosinus-Transformation (engl.: Shape-Adaptive Discrete Cosine Transformation, SA-DCT [3]) zur Referenzbildentrauschung. Der Vorteil der SA-DCT gegenüber der blockweisen DCT zur Rauschuterdrückung liegt in der guten Anpassungsfähigkeit an die zu transformierende Umgebung. Somit ist sie vor allem an Kanten der blockweisen DCT überlegen. Besonders interessant ist sie für unsere Anwendung, weil sie auch im erweiterten MPEG-4 Standard zur Codierung von Videos verwendet wird [4]. Deshalb könnte ein MPEG-4 konformer Chip die SA-DCT auch in Hardware berechnen, was vor allem für Echtzeitanwendungen wichtig wäre. In diesem Beitrag stellen wir die Verwendung der SA-DCT zur Verbesserung der nahezu verlustlosen und der verlustlosen Videocodierung vor.

Darüber hinaus zeigen wir eine Möglichkeit auf, wie man die SA-DCT parametrisieren kann, um auch bei höheren Quantisierungsstufen einen Codiergewinn erzielen zu können.

2. Motivation der Referenzbildfilterung

In diesem Kapitel wird erklärt warum eine Referenzbildfilterung einen Vorteil in der Codierung mit sich bringen kann. Es wird ein bestimmtes Modell eines zu codierenden Videos angenommen und gezeigt, dass eine Referenzbildfilterung für eine effizientere Codierung vorteilhaft ist.

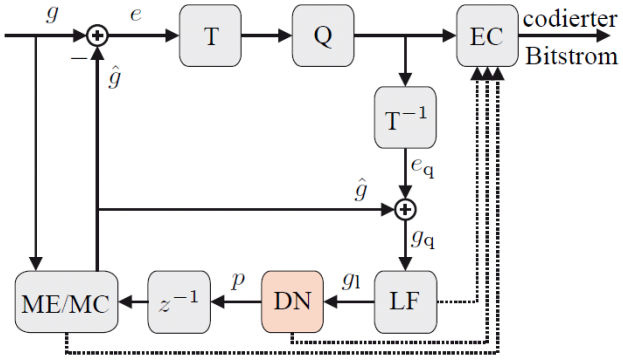

Bild 1 Illustration eines Videocoders; DN-Block ist in einem standardkonformen Coder nicht vorhanden.

2.1. Grundideen eines Hybridcoders

In Bild 1 ist eine vereinfachte Version eines Videocoders dargestellt. In einem Standard-Videocoder ist der DN-Block ausgelassen und die zeitliche Prädiktion geschieht direkt aus dem rekonstruierten Referenzbild g_l. Zur Bildung des Prädiktionsfehlersignals e wird vom zu codierenden Block aus dem aktuellen Bild g der bewegungskompensierte Block aus dem rekonstruierten Referenzbild $g_l$subtrahiert. Wenn die Leistung im Prädiktionsfehlersignal geringer ist als im aktuell zu codierenden Signal, ist es besser den Prädiktionsfehler weiter zu codieren um dadurch Datenrate einsparen zu können. Der Prädiktionsfehler wird weiter transformiert und quantisiert.

Die quantisierten Frequenzkoeffizienten werden entropiecodiert und zum Decoder übertragen. Die quantisierten

Transformationskoeffizienten werden im Coder wie im Decoder zurück transformiert und ergeben das verlustbehaftet rekonstruierte Prädiktionsfehlersignal e_q. Dieses wird auf den Prädiktor $\hat{g}$ addiert um das verlustbehaftet rekonstruierte Signal g_q zu erhalten. Da durch die Quantisierung Blockartefakte entstehen können, wird im LF-Block das Signal gefiltert. Das resultierende Signal g_l wird einerseits vom Decoder ausgegeben und andererseits in den Referenzbildspeicher zur Prädiktion der nächsten Bilder geschrieben.

Der wesentliche Vorteil eines Videocodecs zu einem reinen Bildcodec besteht vor allem in der Ausnutzung der zeitlichen Korrelation in einem Videosignal. Jedoch wird im nächsten Kapitel erklärt, dass diese Korrelation in einem standardkonformen Codec nicht optimal ausgenutzt wird.

2.2. Prädiktion aus verrauschten Daten

Im Folgenden nehmen wir an, dass es sich beim Coder um einen verlustfreien Standardcoder handelt. Wir fügen den Zeitindex t hinzu um die zeitliche Abfolge der Bilder zu kennzeichnen. Somit werden die Blöcke Q, LF und DN ausgelassen und das Signal $g_q[t]=g_l[t]=p[t]$ entspricht dem Originalsignal $g[t]$. Nun nehmen wir an, dass das Signal $g[t]$ aus einem Signalanteil $f[t]$ und einem Rauschanteil $n[t]$ besteht:

$$g[t]=f[t]+n[t] \qquad (1)$$

Für die Prädiktion von $g[t]$ wird das vorhergehende Bild $g[t-1]$ genommen, welches ebenso aus einem Rauschanteil $n[t-1]$ und einem Signalanteil $f[t-1]$ besteht und das bewegungskompensierte Signal $\hat{g}[t]$ gebildet. Das Signal $\hat{g}[t]$ besteht ebenso aus einem Rauschanteil $\tilde{n}[t-1]$ und einem Signalanteil $\tilde{f}[t-1]$. Die Bildung des Prädiktionsfehlers $e[t]$ resultiert in folgender Beziehung:

$$e[t]=f[t]-\tilde{f}[t-1]+n[t]-\tilde{n}[t-1] \qquad (2)$$

Man kann erkennen, dass der Prädiktionsfehler wieder aus einem Signalanteil $f_e[t]=f[t]-\tilde{f}[t-1]$ und einem Rauschanteil $n_e[t]=n[t]-\tilde{n}[t-1]$ besteht. Wenn die Korrelation zwischen aufeinanderfolgenden Bildern hoch ist, so kann die Leistung des Signalanteils $f_e[t]$ kleiner werden. Jedoch wird die Leistung des Rauschanteils $n_e[t]$ üblicherweise größer, da das Rauschen von aufeinanderfolgenden Bildern im Allgemeinen nicht korreliert ist.

2.3. Schleifenfilter für die Referenzbildfilterung

Im vorhergehenden Kapitel wurde gezeigt, dass das Prädiktionsfehlersignal eine höhere Rauschleistung haben kann, als das zu codierende Signal selbst. Um diese Rauschleistung zu minimieren, könnte man das Rauschen im prädizierten Signal $\hat{g}$ entfernen. Um jedoch die Bewegungssuche und dadurch die Bewegungskompensation durch das Rauschen nicht negativ zu beeinflussen, entfernen wir das Rauschen vor der Bewegungssuche/Bewegungskompensation. Dies wird in unserem Ansatz durch ein explizites Rauschfilter in der Schleife des

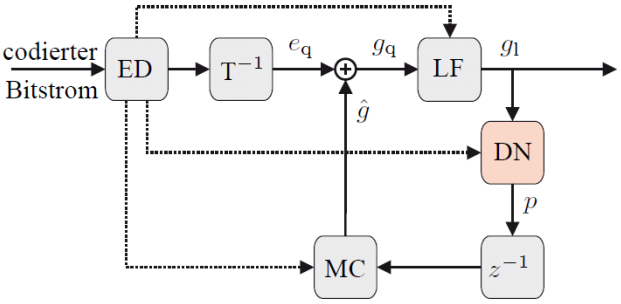

Bild 2 Illustration eines Videodecoders; DN-Block ist nicht vorhanden in einem Standard-konformen Decoder.

Codecs erreicht. Es ist mit dem DN-Block im Coder (Bild 1) und Decoder (Bild 2) illustriert.

Insbesondere das Diagramm des Decoders verdeutlicht die Wirkungsweise des eingeführten Schleifenfilters. Es wird angewendet nachdem das rekonstruierte Bild ausgegeben und bevor es in den Referenzbildspeicher geschrieben wird. Auf diese Weise ist es möglich einen solchen Ansatz auch für die verlustlose Codierung anzuwenden [1], da nur die Prädiktion beeinflusst wird. Für verlustbehaftete Codierung ist es ebenfalls sinnvoll, da das Rauschen für viele Werte des Quantisierungsparameters noch vorhanden ist [2]. In dem DN-Block wird die Rauschleistung geschätzt und das Rauschen wird durch ein dediziertes Rauschfilter entfernt.

In diesem Beitrag verwenden wir den Ansatz der Referenzbildfilterung ebenfalls für die Prädiktionsverbesserung bei sehr groben Quantisierungseinstellungen. Motiviert durch die Tatsache, dass die SA-DCT für die Unterdrückung der Blockartefakte eines JPEG-codierten Bildes eingesetzt werden konnte [5], haben wir sie für die Filterung des Referenzbildes bei hohen QPs eingesetzt.

3. Formangepasste diskrete Cosinus-Transformation mit Hilfe der ICI-Regel

In diesem Kapitel werden die Algorithmen zur Referenzbildfilterung erklärt. Zuerst wird die Anwendung der SA-DCT kurz erläutert. Dabei wird auf eine ausführliche Beschreibung der SA-DCT mit dem nötigen Formalismus in diesem Beitrag verzichtet. Für eine detailliertere Beschreibung der implementierten SA-DCT wird der Leser auf [5] verwiesen. Anschließend werden die Methoden erläutert, wie das modellierte Rauschen im Referenzbild für niedrige sowie für hohe Quantisierungsparameter geschätzt werden kann.

3.1. SA-DCT Überblick

Die SA-DCT hat gegenüber der blockweisen DCT den Vorteil, dass beliebig geformte Umgebungen transformiert werden können. Dadurch ist es möglich die zu transformierenden Regionen so zu wählen, dass sie im Frequenzbereich kompakt darstellbar sind, was eine Rauschfilterung im Frequenzbereich erheblich erleichtern kann. Die Filterung eines Bildes geschieht dabei in zwei

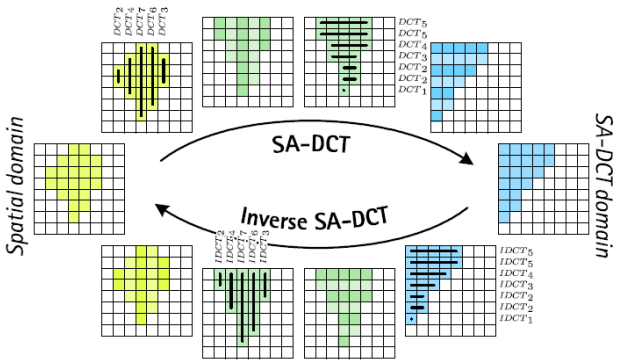

Bild 3 Illustration der SA-DCT aus [5]

Iterationen. Im ersten Schritt wird eine grobe Schätzung eines rauschfreien Bildes gemacht und im zweiten Schritt wird das Ergebnis aus dem ersten Schritt verfeinert. In beiden Iterationen wird für jede Pixelposition eine Filterung im SA-DCT-Bereich durchgeführt. Nachdem in der jeweiligen Iteration für alle Pixelpositionen die Filterung abgeschlossen ist, werden alle Schätzwerte miteinander abhängig von ihrer Zuverlässigkeit kombiniert.

Die Transformation in den SA-DCT-Bereich ist in Bild 3 illustriert. Auf der linken Seite erkennt man eine bestimmte Umgebung im Ortsbereich (Spatial domain), die in den Frequenzbereich abgebildet werden soll. Zuerst werden 1-D-DCTs unterschiedlicher Längen auf alle Spalten dieser Umgebung angewendet. Nachfolgend werden die transformierten Spalten an ihren DC-Koeffizienten ausgerichtet. Anschließend werden horizontale 1-D-DCTs unterschiedlicher Längen auf alle Zeilen angewendet und das Ergebnis erneut ausgerichtet. Im Frequenzbereich kann nun eine Filterung erfolgen. Bei der Rücktransformation in den Ortsbereich wird analog vorgegangen. Zuerst erfolgt die Rücktransformation der Zeilen und nach Ausrichtung der Koeffizienten erfolgt die spaltenweise Rücktransformation. Zum Schluss werden die Pixelwerte auf ihre ursprüngliche Position verschoben.

In der ersten Iteration erfolgt eine Schwellwertbildung der SA-DCT-Koeffizienten. Dabei werden die Koeffizienten, die eine bestimmte Schwelle unterschreiten, zu Null gerundet. Die Schwelle ist durch

$$T = \alpha \cdot \sigma_n \cdot \sqrt{2\ln(|N|)+1} \qquad \textbf{(3)}$$

gegeben [5], wobei $|N|$ die Anzahl der SA-DCT-Frequenzkoeffizienten angibt, σ_n die Standardabweichung des Rauschens ist und α ein konstanter Parameter. In der zweiten Iteration erfolgt eine klassische Wienerfilterung. Dabei wird das gefilterte Bild aus dem ersten Schritt verwendet, um das Leistungsdichtespektrum der Umgebungen des originalen unverrauschten Bildes anzunähern.

Die Effizienz des Filterergebnisses hängt stark von der Wahl des Gebietes für die SA-DCT ab. Um eine möglichst kompakte Repräsentation im Frequenzbereich zu erreichen, sollte eine homogene Region mit möglichst wenig Kanten benutzt werden. In diesem Beitrag verwenden wir zur Wahl des Gebietes die Methoden aus [5]. Im

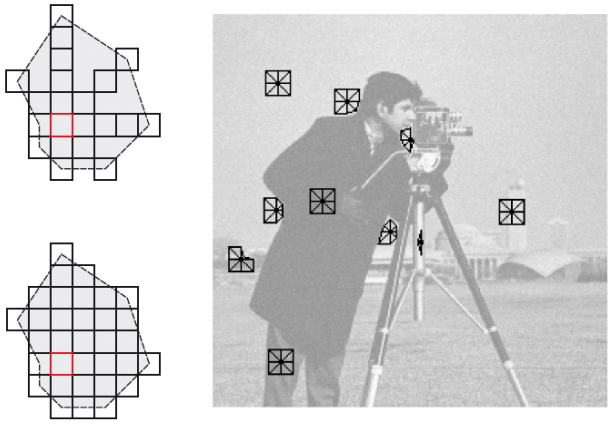

Bild 4 Illustration der Polygon-Bildung für die Auswahl der stationären Umgebung für die SA-DCT auf der linken Seite und Beispiele für gewählte Polygone auf der rechten Seite; weitere Beispiele und Illustrationen in [5][6]

ersten Schritt wird dazu die Ausweitung der Umgebung vom aktuellen Pixel aus in acht Richtungen überprüft, wie es im linken oberen Teil von Bild 4 illustriert ist. Die Ausweitung in eine bestimme Richtung erfolgt hierbei so lange, bis das neu hinzugefügte Pixel eine Schätzung des aktuellen zentralen Pixels deutlich negativ beeinflussen würde (oder eine maximale Weite erreicht wurde). Aus den geschätzten Maximallängen wird ein Polygon gebildet, welches alle Pixel enthält, die für die Transformation verwendet werden (links unten). Auf der rechten Seite von Bild 4 sind Beispiele für die Fensterwahl gezeigt. Man kann erkennen, dass sich das Fenster sehr gut an den Bildinhalt anpassen kann.

Für die Ermittlung der Maximallängen in jede Richtung wird die ICI-Regel (engl. Intersection of Confidence Intervals) mit gerichteten eindimensionalen LPA-Faltungskern (engl. Local Polynomial Approximation) angewendet. Ziel ist hierbei die Schätzung des aktuellen Pixels durch Faltung der eindimensionalen Umgebung in eine bestimmte Richtung mit einem LPA-Kern. Je länger der LPA-Kern und die Umgebung hierbei gewählt werden, desto geringer ist die Varianz des resultierenden Schätzwertes. Andererseits weicht der Schätzwert für längere Kern stärker vom wahren Wert des zu schätzenden Pixels ab. Um die optimale Kernlänge zu finden, wird für jede Kernlänge h_i ein Konfidenzintervall $\mathcal{D}_i$ berechnet:

$$\mathcal{D}_i = \left[\hat{y}_{h_i}(x) - \Gamma\sigma_{\hat{y}_{h_i}(x)},\ \hat{y}_{h_i}(x) + \Gamma\sigma_{\hat{y}_{h_i}(x)}\right] \qquad \textbf{(4)}$$

wobei $\hat{y}_{h_i}(x)$ der resultierende Schätzwert mit dem aktuellen Kern k_{h_i} ist, $\sigma_{\hat{y}_{h_i}(x)}$ die Standardabweichung des Rauschens des resultierenden Schätzwertes und $\Gamma > 0$ ein konstanter Modellparameter ist. Für den allgemeinen Fall ortsvarianten Rauschens lässt sich $\sigma_{\hat{y}_{h_i}(x)}$ gemäß [6] durch

$$\sigma_{\hat{y}_{h_i}} = \sqrt{\sigma_n^2 * k_{h_i}^2} \qquad \textbf{(5)}$$

berechnen, wobei hier sowohl die Rauschvarianz σ_n^2 als auch die Kernfunktion k_{h_i} als ortsabhängige Funktionen

zu betrachten sind. Als optimale Größe h^+ wird schließlich die größte Kernlänge gewählt, für die alle vorhergehenden Konfidenzintervalle eine nicht-leere Schnittmenge $\mathcal{I}_i$ besitzen. Dies ist in Bild 5 illustriert. Hier wurden vier Kernlängen in aufsteigender Reihenfolge verwendet, wobei sich schließlich $h3$ als optimale Länge ergibt.

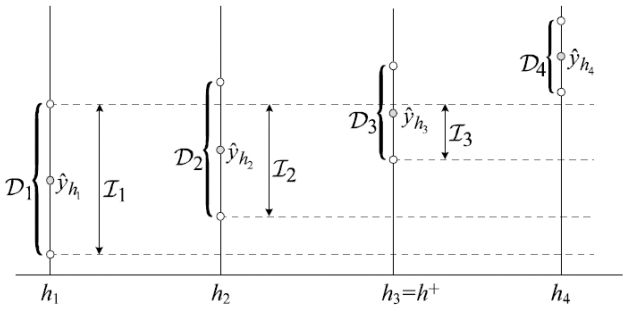

Bild 5 Illustration der ICI Regel aus [5]

3.2. Rauschmodellierung und Schätzung für Codierung mit niedrigen Quantisierungsparametern

Bei niedrigen Einstellungen des Quantisierungsparameters kann man davon ausgehen, dass das Eingangsrauschen im Referenzbild noch vorhanden ist [2] und deshalb auch mit einem expliziten Filter geschätzt werden kann.

In diesem Fall verwenden wir die Methode von [7], wo das Rauschen codierungsvariant geschätzt wird. Dabei geht man davon aus, dass für Intra-codierte Blöcke, Inter-codierte Blöcke sowie für Skip-codierte Blöcke die Rauschleistung unterschiedlich groß ist (MA, engl. Mode Adaptive). Die Schätzung der Standardabweichung des Rauschens wird durch die folgende Formel beschrieben:

$$\sigma_{n_q}[M] = \frac{1}{6|\mathbf{\Omega}_M|}\sqrt{\frac{\pi}{2}}\sum_{\mathbf{j}\in\mathbf{\Omega}_M}|g_l[\mathbf{j}] * \mathbf{h}| \qquad \textbf{(6)}$$

wo $\sigma_{n_q}[M]$ die geschätzte Standardabweichung des Rauschens für die Pixel ist, welche zu den jeweiligen Modi $M \in \{Intra, Inter, Skip\}$ zugehören. $g_l[\mathbf{j}]$ ist das rekonstruierte verrauschte Bild, $\mathbf{h}$ ist das Schätzfilter, Ω_M beinhaltet alle Koordinaten der Pixel, die zu der Klasse M gehören und $|\Omega_M|$ ist die Anzahl der Einträge in Ω_M. Die Filtermaske für $\mathbf{h}$ ist nachfolgend gegeben:

$$\mathbf{h} = \begin{array}{|r|r|r|}\hline 1 & -2 & 1 \\ \hline -2 & 4 & -2 \\ \hline 1 & -2 & 1 \\ \hline \end{array} \qquad \textbf{(7)}$$

Aus (6) kann man erkennen, dass für die Rauschschätzung eine einzige gemeinsame Filteroperation benötigt wird und abhängig von der Klasse M die gefilterten Pixel aufaddiert werden, um die Standardabweichung des Rauschens aller Pixel, die zu dieser Klasse M gehören, zu schätzen. Bei der Anwendung der SA-DCT für die Rauschreduktion des Referenzbildes wird die geschätzte Standardabweichung $\sigma_{n_q}[M]$ dem SA-DCT-Algorithmus zugeführt.

3.3. Rauschmodellierung und Schätzung für Codierung mit hohen Quantisierungsparametern

Bei hohen Quantisierungsparametern gehen wir davon aus, dass das Eingangsrauschen bereits durch die Quantisierung und die nachfolgenden Standard-Loop-Filter (illustriert mit Block LF in den Bildern 1 und 2) entfernt worden ist. In diesem Fall nehmen wir im Modell an, dass durch die Quantisierung Rauschen in das Referenzbild eingeführt wird, das ebenfalls die Prädiktion negativ beeinflussen könnte. Um die SA-DCT unverändert für die Rauschunterdrückung zu verwenden, modellieren wir das Rauschen mit einer Standardabweichung σ_q, die dem SA-DCT-Algorithmus als Parameter übergeben wird.

In dieser Anwendung wählen wir einen vorwärtsadaptiven Ansatz für die Schätzung der Standardabweichung des Rauschens. Hierbei nutzen wir die Kenntnis des Coders über das Originalbild vor der Codierung und berechnen die Wurzel des mittleren quadratischen Fehlers zwischen dem Originalbild g und dem rekonstruierten gefilterten Bild g_l. Somit läßt sich σ_q^z (z steht dabei für ein Zwischenergebnis) folgendermaßen im Coder berechnen:

$$\sigma_q^z = \sqrt{\frac{1}{|\mathbf{\Omega}|}\sum_{\mathbf{j}\in\mathbf{\Omega}}(g[\mathbf{j}] - g_l[\mathbf{j}])^2} \qquad \textbf{(8)}$$

Hierbei enthält Ω alle Koordinaten des Bildes g und $|\Omega|$ stellt die Anzahl der Pixel im Bild dar. Da der Decoder keine Kenntnis über das Originalbild hat, muss der Parameter σ_q^z zum Decoder übertragen werden. Dies ist für jedes Bild notwendig, weshalb σ_q^z in unserer Anwendung auf acht Bit folgendermaßen quantisiert wird:

$$\sigma_q = \min(\mathrm{round}(\sigma_q^z \cdot 10)/10, 25.5) \qquad \textbf{(9)}$$

wobei round die Rundung im mathematischen Sinne darstellt. Das quantisierte σ_q wird für die Ansteuerung der Filterung im Coder sowie im Decoder verwendet.

Als letzten Schritt führen wir noch eine Adaption im Coder ein. Hierbei entscheidet der Coder, ob das gefilterte Referenzbild gut oder schlecht ist. Dies geschieht durch einen expliziten Vergleich der mittleren quadratischen Fehler $MSE(g, g_l) = \sigma_q^z$ und $MSE(g, p) = \sigma_p^z$. Dabei lässt sich σ_q ebenfalls mit (7) berechnen, wenn p Anstelle von g_l eingesetzt wird. Falls $MSE(g, p) > MSE(g, g_l)$ ist, wird $\sigma_q = 0$ gesetzt und übertragen, woraufhin der Decoder weiß, dass das ungefilterte Bild g_l für die zeitliche Prädiktion benutzt werden soll.

4. Simulationsergebnisse

In diesem Kapitel zeigen wir einige Simulationsergebnisse für die Codierung mit Anwendung der Referenzbildfilterung im aktuellen HEVC Referenzcodec. Als Basis haben wir die Version HM-8.2 verwendet. Für die Codierungssimulationen verwenden wir ClassB und ClassD Sequenzen aus der Standardisierung [8]. Dabei haben

ClassB Sequenzen eine Auflösung von 1920x1080 Pixeln und die ClassD Sequenzen eine Auflösung von 416x240 Pixeln. Somit sind die ClassB-Sequenzen insbesondere repräsentativ für professionelle Anwendungen.

Es wurden jeweils 100 Bilder der Videosequenzen mit *lowdelay P* Simulationseinstellungen codiert [8]. Die Codierergebnisse werden gegen die Original-Referenzsoftware ohne Referenzbildfilterung verglichen.

4.1. Codierung bei hohen Qualitätseinstellungen

Für diese Simulationen wurde die Rauschmodellierung und Schätzung aus Kapitel 3.2 verwendet. Für die Filterung des Referenzbildes verwenden wir die SA-DCT (SA-DCT-MA). Zusätzlich werden die Codierergebnisse für den Algorithmus (AWF-MA), der bereits in [7] präsentiert wurde, gezeigt und verglichen. In Tabelle 1 sind die Simulationsergebnisse für dieses Szenario zusammengefasst. Es wird zwischen 3 Qualitätsstufen unterschieden: verlustlos, hohe Qualität ($QP \in \{12,...,27\}$) und mittlere Qualität ($QP \in \{22,...,37\}$). In der Tabelle sind die Durchschnittswerte der Bitratenersparnisse für die jeweiligen Qualitätsbereiche angegeben.

Tabelle 1 Durchschnittliche Bitratenersparnisse für unterschiedliche Qualitätsbereiche (LL: Verlustlos, HQ: Hohe Qualität, MQ: Mittlere Qualität)

Sequenz	Δ Bitrate in %					
	AWF-MA			SA-DCT-MA		
	LL	HQ	MQ	LL	HQ	MQ
BasketballDrive	-1.72	-3.46	0.12	-3.04	-10.23	-2.27
BQTerrace	0.03	-0.82	7.66	-1.56	-9.08	-5.21
Cactus	-1.92	-3.61	1.66	-2.61	-7.26	-3.54
Kimono1	-1.30	-1.32	-0.45	-2.25	-9.84	-3.32
ParkScene	-0.58	2.15	2.73	-1.18	1.74	-0.25
Durchschnitt	**-1.10**	**-1.41**	**2.34**	**-2.13**	**-6.93**	**-2.92**
BasketballPass	3.86	3.97	0.82	-0.06	-0.16	-0.09
BlowingBubbles	1.42	4.89	1.84	-0.20	-0.48	0.02
BQSquare	9.55	27.89	12.80	-0.50	-0.01	0.65
RaceHorses	-0.07	0.07	0.14	-0.03	-0.15	-0.03
Durchschnitt	**3.69**	**9.21**	**3.90**	**-0.20**	**-0.20**	**0.14**

Man kann erkennen, dass der vorgeschlagene Ansatz der Referenzbildfilterung nur für höhere Auflösungen des Videomaterials gut funktioniert. Vor allem bei Verwendung von AWF-MA kann man auch deutliche Bitratenerhöhungen bei Codierung von niedrig-aufgelösten Sequenzen erkennen. Dies liegt an der Tatsache, dass das Verhältnis zwischen Signal- und Rauschleistung bei solchen Auflösungen sehr hoch ist und durch die relativ einfache Rauschschätzung die Rauschvarianz zu hoch geschätzt wird. Dies führt zu einer Überfilterung des Referenzbildes, was sich, bei Verwendung eines einfachen Filterungsansatzes, in einer schlechteren Prädiktion wiederspiegelt. Grundsätzlich ist das Problem auch bei der Verwendung der SA-DCT vorhanden. Jedoch ist dort eine falsche Rauschvarianzschätzung nicht so tragisch, da die Signalmodellierungseigenschaften der SA-DCT sehr gut sind und eine zu starke Filterung noch nicht auftritt.

Wenn man die Ergebnisse für hohe Qualität und niedrige Qualität für die ClassB-Sequenzen vergleicht, erkennt man, dass die angewendete Referenzbildfilterung vor allem bei kleinen Quantisierereinstellungen (HQ) effizienter ist. Der Grund für dieses Verhalten ist die inhärente Rauschunterdrückung durch die stärker werdende Quantisierung [2], was die explizite Rauschfilterung des Referenzbildes weniger nötig macht. Das Verhalten für verschiedene Quantisierungseinstellungen ist am Beispiel der Cactus-Sequenz in Bild 6 illustriert.

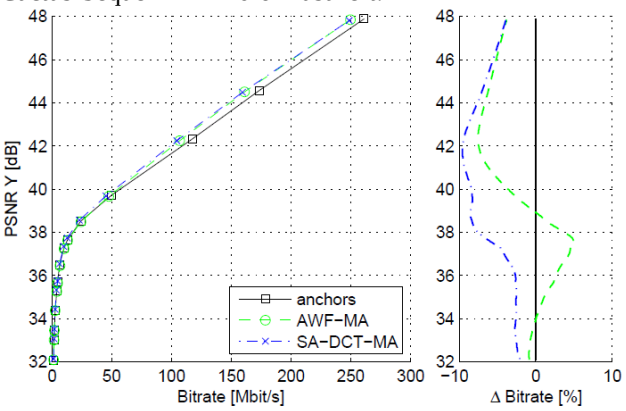

Bild 6 Ratenverzerrungskurve (links) und die relativen Bitrateneinsparnisse (rechts) für Codierung der Cactus-Sequenz

4.2. Codierung bei niedrigen Qualitätseinstellungen

Bei den nachfolgenden Simulationensergebnissen wurde die Rauschmodellierung und Schätzung aus Kapitel 3.3 verwendet. Für die Filterung des Referenzbildes verwenden wir die SA-DCT. Zusätzlich werden die Codierergebnisse bei Verwendung des AWF-Algorithmus zur Filterung, angegeben. In Tabelle 2 sind die Simulationsergebnisse für dieses Szenario zusammengefasst. Es wird zwischen zwei verschiedenen Qualitätsstufen unterschieden: mittlere Qualität ($QP \in \{22,...,37\}$) und niedrige Qualität ($QP \in \{37,...,51\}$). In der Tabelle sind die Durchschnittswerte der Bitratenersparnisse für die jeweiligen Qualitätsbereiche angegeben.

Tabelle 2 Durchschnittliche Bitratenersparnisse für unterschiedliche Qualitätsbereiche (MQ: Mittlere Qualität, LQ: Niedrige Qualität)

Sequenz	Δ Bitrate in %					
	AWF ideal		SA-DCT ideal		SA-DCT real	
	MQ	LQ	MQ	LQ	MQ	LQ
BasketballDrive	-0.13	-2.11	-3.57	-5.47	-3.57	-5.42
BQTerrace	0.00	-0.38	-2.75	-3.78	-2.74	-3.69
Cactus	0.00	-0.88	-2.70	-2.90	-2.70	-2.84
Kimono1	-2.65	-9.42	-8.24	-10.18	-8.23	-10.12
ParkScene	0.00	-2.24	-0.58	-4.64	-0.58	-4.57
Durchschnitt	**-0.56**	**-3.01**	**-3.57**	**-5.39**	**-3.56**	**-5.33**
BasketballPass	-0.09	-2.64	-1.69	-4.54	-1.61	-3.84
BlowingBubbles	0.00	-0.42	-0.10	-1.64	-0.04	-1.01
BQSquare	0.07	0.17	-0.27	-2.25	-0.21	-1.69
RaceHorses	-0.01	-0.69	-0.92	-4.34	-0.88	-3.94
Durchschnitt	**-0.01**	**-0.90**	**-0.75**	**-3.19**	**-0.69**	**-2.62**

In diesen Simulationen musste das coderseitig geschätzte σ_q zum Decoder übertragen werden. Vor allem bei höheren Quantisierungseinstellungen kann die Übertragung von σ_q für jedes Bild einen relativ großen Anteil an der Gesamtbitrate ausmachen (z.B. BQSquare). Deshalb zeigen wir hier die Ergebnisse für einen hypothetischen Fall, dass die Rauschparameter im Decoder selbst geschätzt werden könnten (AWF ideal und SA-DCT ideal). Darüber hinaus wird auch das Ergebnis für den realen Fall, in dem σ_q zum Decoder übertragen wird, gezeigt (SA-DCT real).

Den Codierergebnissen nach ergeben sich fast immer Bitratenersparnisse. Jedoch sind die Gewinne, wenn man die SA-DCT zur Referenzbildfilterung verwendet, deutlich höher als bei AWF. Die Begründung dafür ist ähnlich wie im Kapitel zuvor. Die SA-DCT hat sehr gute Signalmodellierungseigenschaften, weshalb das Ergebnis nach der Filterung deutlich besser ist als bei Verwendung von AWF. Darüber hinaus erkennt man bei Vergleich von „SA-DCT ideal“ und „SA-DCT real“, dass die Seiteninformation vor allem bei kleineren Auflösungen einen erheblichen Anteil an der Gesamtbitrate haben kann. Jedoch könnte dieser Anteil weiter minimiert werden, wenn eine bessere Entropiecodierung mit Kontextmodellierung verwendet werden würde.

Des Weiteren kann man erkennen, dass die Bitratenersparnisse für sehr starke Quantisierungseinstellungen deutlich höher sind als für mittlere Quantisierungseinstellungen. Für stärkere Quantisierung wird ein höherer Störfaktor in das Referenzbild eingeführt, der durch die Filterung verkleinert wird. In Bild 7 ist eine Illustration des Verhaltens der Bitratenersparnisse über verschiedene Quantisierungseinstellungen für die ParkScene-Sequenz gegeben. Im Bild erkennt man ebenfalls, dass die Bitratenersparnisse für ansteigende Quantisierungsparameter größer werden.

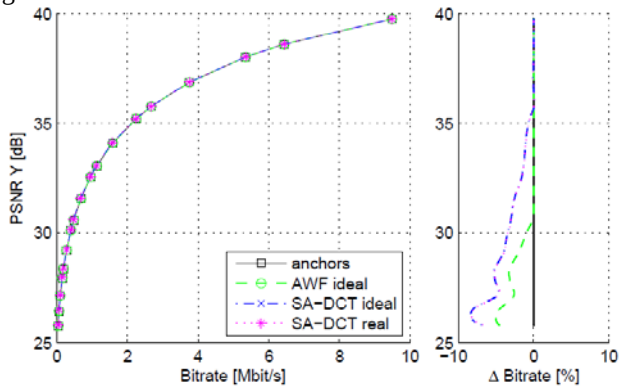

Bild 7 Ratenverzerrungskurve (links) und die relativen Bitrateneinsparnisse (rechts) für Codierung der ParkScene-Sequenz

5. Zusammenfassung

In diesem Beitrag haben wir gezeigt, wie ein explizites Filter zur Verarbeitung des Referenzbildes verwendet werden kann, um die Codiereffizienz eines Standard-Videocodecs für bestimmte Qualitätsbereiche zu verbessern. Es wurde motiviert, dass für hohe Qualitätsbereiche das Rauschen, das bei der Aufnahme entstanden ist, entfernt werden muss um die Prädiktionseffizienz zu verbessern. Für höhere Quantisierungsparametereinstellungen verschwindet zwar das Eingangsrauschen, jedoch entsteht ein anderes Fehlersignal, was ebenfalls die Prädiktion verschlechtert. Durch die Verwendung der jeweils für den Anwendungsbereich abgestimmten Referenzbildfilter-Algorithmen konnte die Codiereffizienz deutlich gesteigert werden. Mit Hilfe der SA-DCT konnten sowohl für hohe Qualitätseinstellungen als auch für niedrige Qualitätseinstellungen Bitrateneinsparungen von bis zu 10 % erzielt werden.

Den Ergebnissen zu Folge kann man vermuten, dass im mittleren Qualitätsbereich die Codierung mit HEVC derzeit optimal funktioniert. Lediglich für niedrige oder hohe Qualitäten ist Einsparungspotenzial gegeben. Für die Zukunft bleibt die Frage offen, wie man beide Ansätze miteinander kombinieren kann.

Literatur

[1] E. Wige, P. Amon, A. Hutter, and A. Kaup, “In-loop denoising of reference frames for lossless coding of noisy image sequences,” in Proc. of IEEE International Conference on Image Processing (ICIP), Sep. 2010, pp. 461–464.

[2] E. Wige, G. Yammine, P. Amon, A. Hutter, and A. Kaup, “Analysis of in-loop denoising in lossy transform coding,” in Proc. of Picture Coding Symposium (PCS), Dec. 2010, pp. 82–85.

[3] T. Sikora and B. Makai, “Shape-adaptive dct for generic coding of video,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 5, no. 1, pp. 59 –62, Feb. 1995.

[4] A. Kaup and S. Panis, “On the performance of the Shape-Adaptive DCT in object-based coding of motion compensated difference images,” in Proc. of Picture Coding Symposium, Berlin, Sep. 1997, pp. 653–657.

[5] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise Shape-Adaptive DCT for high-quality denoising and deblocking of grayscale and color images,” IEEE Trans. on Image Processing, vol. 16, no. 5, pp. 1395 –1411, May 2007.

[6] A. Foi, V. Katkovnik, and K. Egiazarian, “Signal-dependent noise removal in Pointwise Shape-Adaptive DCT domain with locally adaptive variance,” in Proc. of 15th European Signal Processing Conference (EUSIPCO), Poznan, Poland, Sep. 2007.

[7] E. Wige, G. Yammine, W. Schnurrer, and A. Kaup, “Mode adaptive reference frame denoising for high fidelity compression in HEVC,” in IEEE International Conference on Visual Communications and Image Processing (VCIP), San Diego, CA, USA, Nov. 2012, pp. 1–6.

[8] F. Bossen, Common test conditions and software reference configurations, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Std., doc. JCTVC-K1100, Shanghai, China, Oct. 2012.