Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Bachelorthesis

Aktuelle Entwicklung von wahrnehmungsbasierter Videokompression

von Kevin Fröhlich
(9019336)

31. August 2016

Erstprüfer: Prof. Dr. Karl Jonas
Zeitprüfer: M. Sc. Michael Rademacher

Inhaltsverzeichnis

Abbildungsverzeichnis

Eidesstattliche Erklärung

Ich versichere an Eides statt, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Datum, Ort, Unterschrift

1 Einleitung

Videos sind im Alltag in großem Umfang vertreten und sind nicht nur durch das Fernsehen, sondern auch durch Plattformen wie YouTube oder sozialen Medien seit den vergangenen Jahren vor allem im Internet weit verbreitet. Videos entwickeln sich in ihrer Qualität stetig weiter, sowohl in ihrer Auflösung als auch in ihrer Framerate (Anzahl der Bilder pro Sekunde) [Xu Liang Wang 2015, S. 284].

In einer Analyse von Cisco Systems Inc. aus dem Jahr 2015 wird die Tendenz der Qualität und Auflösung von Videos sowie des Speicherplatzbedarfs für Videos als stark ansteigend vorausgesagt. Videos im Internet werden im Jahr 2019 mit über 100 Exabyte Daten pro Monat ca. 90 Prozent des Gesamtdatenaufkommens ausmachen [Cisco 2015].

In Zukunft werden also neue und bessere Kompressionsverfahren für Videos eine große Rolle spielen. Eine der neueren Methoden, an denen aktuell und in den letzten Jahren vermehrt gearbeitet und geforscht wurde, ist die Berücksichtigung der menschlichen Wahrnehmung bei der Kompression von Videos [Lee Ebrahimi 2012, Chen Lin Ngan 2010, Chen Li 2015]. Denn der Mensch nimmt die Qualität von Videos auf Grund seiner verschiedenen Seheigenschaften und der Art der Verarbeitung der Bilddaten im Gehirn z.B. räumlich nicht gleichmäßig wahr, was in aktuellen Videokompressionsverfahren jedoch angenommen wird. Ein Beispiel für eine solche wahrnehmungsbasierte Kompression ist, dass der Mensch beim Anschauen eines Videos bewegende Objekte fokussiert. Der Hintergrund ist also in diesem Moment nicht so wichtig wie das bewegende Objekt und daher kann der Hintergrund stärker komprimiert werden als das bewegende Objekt [Lee Ebrahimi 2012], ohne dass es dem Zuschauer besonders negativ auffällt. Bei einem üblichen Verfahren würden sowohl der Hintergrund als auch das bewegende Objekt mit gleicher Qualität kodiert. Es bietet sich also das Potential, Videos mit solchen Methoden weiter zu komprimieren, als es beispielsweise bei üblichen Verfahren wie HEVC der Fall ist.

Einige Vorschläge zu wahrnehmungsbasierten Videokompressionsverfahren, die auf unterschiedlichen Eigenschaften der menschlichen Wahrnehmung basieren, wurden bereits entwickelt und publiziert. Es wird daher zunächst ein Überblick geschaffen, wie die aktuell üblichen Videokompressionsverfahren arbeiten, welche Eigenschaften die menschliche Wahrnehmung ausmachen und welche dann für die Videokompression wie ausgenutzt werden können. Des Weiteren wird beschrieben wie Kompressionsverfahren bzw. deren Resultate miteinander verglichen und bewertet werden können und es werden einige neue wahrnehmungsbasierte Verfahren vorgestellt, die solche menschlichen Eigenschaften ausnutzen. Anschließend wird eines der vorgestellten Verfahren, dessen Programmcode von den Autoren eines Ansatzes zur Verfügung gestellt wurde und somit vorliegt, an drei zuvor ausgewählten bzw. selbst erstellten Videosequenzen angewendet. Die Ergebnisse werden bewertet und mit einem üblichen HEVC-Verfahren verglichen.

Abschließend wird ein Fazit gezogen und bewertet ob und in wie weit solche wahrnehmungsbasierten Verfahren eine Verbesserung der Videokompression bewirken können.

2 Stand der Forschung

Im folgenden Kapitel wird zunächst erklärt, warum es sinnvoll ist Videokompressionsverfahren zu verwenden, wie die grundsätzlichen Vorgehensweisen bei dieser sind und wie aktuelle Verfahren diese umsetzen. Im Anschluss daran werden verschiedene Eigenschaften der menschlichen Wahrnehmung vorgestellt, die in wahrnehmungsbasierter Kompression umgesetzt werden können.

2.1 Zweck von Videokompression

Durch die immer weiter ansteigende Qualität und Auflösung von Videos wird folglich auch deutlich mehr Speicherplatz für ein Video benötigt. Ein Bild mit einer Auflösung von 1920 x 1080 Pixeln (Full-HD) und einer Farbtiefe von 24 Bit benötigt unkomprimiert bereits einen Speicherplatz von circa 5,93 Megabyte. Ein Video mit einer solchen Auflösung und 25 Einzelbildern pro Sekunde benötigt damit schon circa 148,32 Megabyte Speicherplatz pro Sekunde. So kommen pro Stunde Videomaterial allein für das Videobild über 521 Gigabyte Daten zusammen, hinzu kommen noch die Audiodaten. Inzwischen werden beispielsweise auf der Plattform YouTube schon Videos mit 60 Einzelbildern pro Sekunde [Heise 2014] angeboten, wodurch ein entsprechend noch höherer Speicherbedarf entsteht.

Ebenso ist es aber auch ein Problem die vielen Daten der Videos über das Netzwerk zu verschicken bzw. von einer Webseite herunterzuladen. Denn dabei entsteht viel Datenverkehr und je nach Größe des Videos dauert es eine entsprechende Zeit, um das Video herunterzuladen. Insbesondere bei mobilen Endgeräten wie Smartphones, bei denen je nach Verbindung nicht zwingend immer eine hohe Datenrate gegeben ist, ist die Größe eines Videos mit hoher Auflösung mitunter eine große Herausforderung.

Um den genannten Problematiken entgegenzuwirken werden Videokompressionsverfahren verwendet. Sie beinhalten verschiedene Algorithmen, die aus mathematischen Umformungen der Videodaten bestehen. Dazu zählen auch Umformungen anhand des Videoinhaltes (z.B. auch die Berücksichtigung von Einzelbildern vor und nach dem aktuell zu bearbeitenden Einzelbild), wodurch die Datenmenge der Videos reduziert werden kann. Hier gibt es viele verschiedene Verfahren, die in den nachfolgenden Kapiteln noch näher beschrieben werden.

Das Ziel und der Zweck dieser Verfahren und von Videokompression im Allgemeinen besteht darin, Videodateien möglichst klein zu halten, während die optisch wahrgenommene Qualität gleich bleibt und keine Daten verloren gehen. Hier unterscheidet man in verlustfreie Kompressionsverfahren und verlustbehaftete Kompressionsverfahren [Henning 2007, S.38], bei denen die Daten häufig auf Kosten der Videoqualität in akzeptablem Maß reduziert werden. Qualitätsverluste sollen dabei jedoch für den Zuschauer so wenig wie möglich sichtbar sein. Je nach dem wie stark die Kompression sein soll, ist es allerdings kaum möglich hierfür nur mit nicht verlustbehafteten Verfahren zu arbeiten.

Da sich die Auflösungen und die Qualität für Videos voraussichtlich auch noch weiter vergrößern und verbessern werden, was z.B. Studien von [Cisco 2015] voraussagen, ist es auch in der nahen Zukunft notwendig weitere, neuere und bessere Videokompressionsverfahren zu entwickeln. Inzwischen gehören Videos in Ultra-HD mit einer Auflösung von 3840 x 2160 Pixeln zum Alltag auf Plattformen wie YouTube und erste Fernsehsendungen werden inzwischen zumindest für die Veröffentlichung im Internet in Ultra-HD produziert [Heise 2016].

Die Videokompression ist also sowohl aktuell als auch weiterhin in der nahen Zukunft ein wichtiges Thema, wenn es darum geht, Videos in hoher Qualität und Auflösung so zu kodieren, dass der Speicherplatzbedarf akzeptabel bleibt.

2.2 Übliche Kompressionsverfahren

Es gibt verschiedene Algorithmen und Verfahren, die es ermöglichen Bilder und Videos zu komprimieren. Die meisten Verfahren bestehen aus drei großen Arbeitsschritten. Zum einen versucht man die räumliche Redundanz in einem einzelnen Bild (Frame) zu entfernen, zum anderen sollen zeitliche Redundanzen über mehrere Bilder hinweg entfernt werden. Außerdem werden die Daten anschließend so kodiert, dass sie möglichst wenig Speicherplatz einnehmen, aber dennoch wieder dekodiert werden können und so das originale Video abbilden.

2.2.1 Räumliche Redundanz

Bilder sind in den meisten Fällen so aufgebaut, dass viele Bereiche zusammenhängen und voneinander abhängig sind. Diese Bereiche sind optisch ähnlich und daher redundant. Ein Beispiel für eine solche sogenannte räumliche Redundanz wäre ein Bild, auf dem eine Landschaft abgebildet ist. Auf einem solchen Bild gäbe es größere Wiesenflächen und entsprechend eine größere Anzahl an grünen Pixeln neben- oder übereinander, wodurch Bildbereiche entstehen, die von den Pixeln her sehr ähnlich sind.

Bei Videokompressionsverfahren wird deshalb die Annahme getroffen, dass Strukturen einer Bildregion ähnlich wie die Strukturen in der nahen Umgebung dieser Bildregion sind. Diese Strukturen können dann mit Hilfe von benachbarten Bereichen im Bild vorhergesagt werden [Wien 2015, S. 40]. Um diese räumlichen Redundanzen in Bildern zu entfernen, wird eine sogenannte Intrapicture-Prediction (Vorhersage innerhalb eines Bildes) verwendet. Ein solches Verfahren wird beispielsweise bei den Videostandards H.264/AVC und H.265/HEVC verwendet.

Frames die nur Intrapicture-Prediction enthalten, werden als I-Frames bezeichnet. Sie sind unabhängig von anderen Frames und werden üblicherweise mit dem Kompressionsverfahren JPEG kodiert. Sie enthalten also die vollständigen Bilddaten und für die Dekodierung dieser werden keine anderen Frames als Referenzen benötigt. I-Frames sind dadurch zwar unabhängig von anderen Frames, haben aber die geringste Kompressionsrate und den größten Speicherplatzbedarf, da hier die meisten Daten noch vorhanden sind. Neben I-Frames existieren noch P-Frames und seit MPEG-2 B-Frames. Diese finden in der Entfernung der zeitlichen Redundanz Anwendung [Henning 2007, S.201ff].

2.2.2 Zeitliche Redundanz

Bei Videos kommt neben räumlichen Redundanzen einzelner Bilder noch eine deutlich größere Redundanz durch andere Frames zu Stande. Ein signifikanter Teil einer Videosequenz besteht aus Objekten, die sich bewegen - entweder durch Bewegungen der Kamera oder durch eigenständige Bewegungen. Es gibt also nur einen geringen Unterschied zwischen aufeinanderfolgenden Frames [Wien 2015, S. 41].

Im vorherigen Frame wird daher nach einer Unterteilung des Bildes in Blöcke der Block gesucht, der am besten mit dem aktuellen Block im zu kodierenden Bild übereinstimmt. Es wird also berechnet, wie genau sich welcher Block von Frame zu Frame verschoben bzw. bewegt hat und entsprechende Bewegungsvektoren gebildet. Diese werden dann gespeichert und die Differenz der Blöcke zusammen mit dem Vektor an den Decoder übergeben, der dann das Bild auf Grundlage des vorherigen Frames, der bereits zuvor dekodiert wurde, wiederherstellen kann [Henning 2007, S.198]. Abbildung 1 zeigt diese Vorgehensweise (linkes Bild).

Man unterscheidet bei den vorhergesagten Frames in P-Frames (Predicted Frames) und B-Frames (Bidirectional Frames). Abbildung 1 zeigt die verschiedenen Frametypen in ihrer

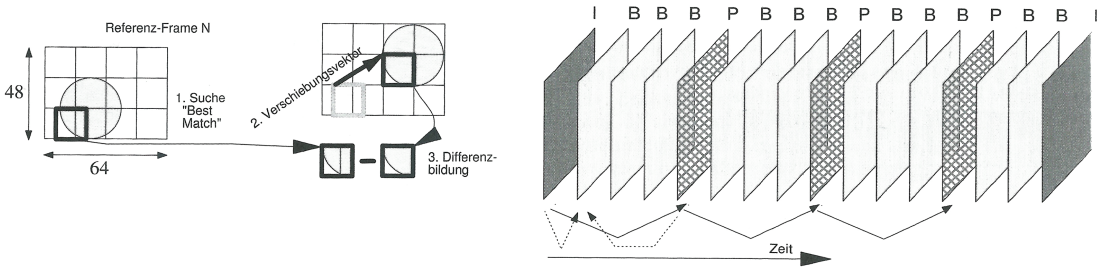

Abbildung 1: Interpicture-Prediction und verschiedene Frametypen
(in Anlehnung an [Henning 2007, S.202 und S.198])

zeitlichen Anordnung (rechtes Bild). P-Frames beinhalten die Vorhersage aus dem letzten I-Frame in Form von Bewegungsvektoren und sind entsprechend von diesem abhängig. B-Frames beinhalten sowohl die Vorhersage aus dem letzten I- bzw. P-Frame als auch dem nächsten I- bzw. P-Frame, weshalb diese zuvor im Decoder bekannt sein müssen, damit diese Frames dekodiert werden können. Die Kompressionsrate von P-Frames ist deutlich höher als die von I-Frames, da nicht alle Bilddaten vollständig kodiert werden müssen, sondern lediglich die Unterschiede zum vorhergehenden Frame. B-Frames haben eine noch mal deutlich höhere Kompressionsrate, da hier die Unterschiede sowohl zum vorherigen als auch nächsten I- oder P-Frame enthalten sind [Henning 2007, S.202ff].

2.2.3 Kodierung

Die resultierenden Daten aus den vorhergehenden Operationen der räumlichen und zeitlichen Redundanzentfernung sollen in der Regel mit einer oder auch mehreren Kodierungen weiter reduziert werden. Dafür werden typischerweise Transformationen wie zum Beispiel die diskrete Kosinustransformation (DCT) angewendet, wodurch die Bilddaten in den Frequenzraum umgeformt werden. Diese Transformation ist verlustfrei. Die resultierenden Koeffizienten aus dieser Transformation werden dann durch die Werte in einer festgelegten Quantisierungsmatrix dividiert und gerundet (Quantisierung), wodurch Daten verloren gehen, aber auch entsprechend Bits eingespart werden können [Henning 2007, S.119f]. Je höher die Werte in der Quantisierungsmatrix sind, desto mehr Daten gehen verloren und entsprechend höher ist die Kompressionsrate.

Diese umgeformten Daten können dann wiederum weiter kodiert werden. Dies kann durch die Anwendung von Entropiekodierung, arithmetischer Kodierung oder ähnlichen Verfahren passieren, wodurch die Daten weiter komprimiert werden können. Entropiekodierung und arithmetische Kodierung befinden sich dabei rein auf der Datenebene unabhängig vom Inhalt des Videos. Sie fassen lediglich Bits zusammen, z.B. durch die Verwendung von Wörterbüchern bei der Entropiekodierung, wobei eine Zeichenfolge ein kürzeres Zeichen als „Übersetzung“ erhält. Ein Beispiel einer solchen Kodierung wäre die Huffman-Kodierung, bei der ein Zeichen je nach Häufigkeit ihres Vorkommens in den zu kodierenden Daten einen kürzeren Code erhält. Die am häufigsten Vorkommenden Zeichen erhalten dabei den kürzesten möglichen (eindeutigen) Code, während Zeichen die nur selten vorkommen, einen längeren erhalten. Durch die häufigen kürzeren Codes und die selteneren längeren Codes kommt so eine Dateneinsparung zu Stande.

Der gesamte Bitstream wird dann gespeichert und die Kodierung im Decoder wieder rückgängig gemacht. Die verlorenen Daten durch die verlustbehaftete Kompression können nicht wiederhergestellt werden, weshalb die Qualität des Videos schlechter wird.

2.3 Aktuelle Videoformate

Die beiden folgenden Unterkapitel liefern einen Überblick über die aktuell üblicherweise verwendeten Verfahren zur Videokompression.

2.3.1 H.264 und H.265

In Zusammenarbeit von ITU-T VCEG und ISO/IEC MPEG wurde 2003 der Videostandard H.264 bzw. Advanced Video Coding (AVC) veröffentlicht. Er löste MPEG-2 bzw. H.262 ab. Gegenüber den vorausgehenden Standards wurden viele Veränderungen und Verbesserungen vorgenommen [Wiegand et al. 2003].

Um die Videokodierung im Bereich der Telekommunikation und Netzwerke zu verbessern wurde ein sogenannter Network Abstraction Layer (NAL) eingeführt, um die Videodaten, die in einem sogenannten Video Coding Layer (VCL) enthalten sind, auf die Transportebene für die Übermittlung in Netzwerken abzubilden. Demnach werden die Daten in Pakete (NAL Units) unterteilt und übertragen [Wiegand et al. 2003].

Jede Videosequenz besteht aus einer Sequenz von kodierten Bildern, die wiederum unterteilt werden in Makroblöcke (MB) mit einer festen Größe von 16 x 16 Pixeln. Es gibt außerdem eine Trennung von Helligkeitswerten (Y) und Farbwerten (Cb und Cr), wobei ein Farbsampling von 4:2:0 stattfindet, d.h. es werden alle Helligkeitswerte, aber nur jeder zweite Farbwert verwendet. Mehrere Makroblöcke werden in Slices zusammengefasst, die eigenständig dekodiert werden können [Wiegand et al. 2003].

Zudem wird sowohl Intra-Frame Prediction als auch Inter-Frame Prediction verwendet. Bei der Intra-Frame Prediction wird zwischen drei verschiedenen Modi unterschieden. Im Modus Intra_4x4 werden 4 x 4 Pixel große Blöcke mit Hilfe der benachbarten Blöcke vorhergesagt, wobei neun verschiedene Richtungen berücksichtigt werden [ITU-T 2013, S. 101]. Alternativ können im Modus Intra_16x16 16 x 16 Pixel große Blöcke verwendet werden [ITU-T 2013, S. 105]. Mit dem Modus I_PCM wird eine Vorhersage übersprungen [ITU-T 2013, S. 110]. Die Inter-Frame Prediction zieht für die Vorhersage weitere Frames in Betracht. Bei H.264 werden die Vorhersagen mit einer Genauigkeit von einem viertel Pixel getroffen, was eine Verbesserung im Gegensatz zu den vorhergehenden Standards bewirkt [Wiegand et al. 2003].

Anschließend werden die Daten mit einer Integer-Transformation, die vergleichbare Eigenschaften mit einer diskreten Kosinus-Transformation besitzt, transformiert. Diese hat den Vorteil, dass sie hauptsächlich mit ganzzahligen Additionen und Bitverschiebungen auskommt [Richardson 2003, S. 187]. Die Transformation wird gefolgt von einer Quantisierung mit 52 möglichen Parameterwerten und einer Entropiekodierung. Bei dieser können zwei verschiedene Verfahren verwendet werden: Context-Adaptive Variable-Length Coding (CAVLC) und Context-Adaptive Binary Arithmetic Coding (CABAC). Zusätzlich wird ein Deblocking-Filter verwendet, der die Entstehung von Artefakten auf Grund der Blockstrukturen verhindern soll [Wiegand et al. 2003].

Insgesamt sind drei Profile in H.264 definiert (Baseline, Main und Extended Profile), die verschiedene Funktionalitäten enthalten oder eben nicht enthalten und für unterschiedliche Zwecke verwendet werden können [Wiegand et al. 2003].

H.265 bzw. High Efficiency Video Coding (HEVC) ist der aktuelle Standard, der von der ITU-T VCEG und der ISO/IEC MPEG (zusammen als JCT-VC) entwickelt und im Jahr 2013 veröffentlicht wurde [ITU-T 2015]. Es wurde nicht nur eine textuelle Beschreibung veröffentlicht, in der beschrieben ist wie dieser Standard umgesetzt wird, sondern es wurde auch eine Implementierung angefertigt, die als Referenzsoftware zur Verfügung steht. Als Nachfolger vom Standard H.264 wurde versucht, die Videokompression noch weiter - vor

allem für hochauflösende Videos - zu verbessern, wofür einige Änderungen im Gegensatz zu H.264 vorgenommen wurden.

Ein großer Unterschied zu H.264 ist die Unterteilung der Frames eines Videos. Während in H.264 jedes Bild in gleichgroße Makroblöcke aufgeteilt wurde, gibt es auch in H.265 eine solche Bildaufteilung in Blöcke. Hier werden zunächst Unterteilungen in 64 x 64 Pixel (je nach Encoder auch 32 x 32 Pixel oder 16 x 16 Pixel) große Coding Tree Units (CTUs) vorgenommen. Eine solche Coding Tree Unit besteht aus einem Coding Tree Block (CTB) für die Helligkeit (Y) und den dazugehörigen Blöcke für die Farbigkeit (Cr und Cb). Hier findet - wie aus der Bildkompression bekannt - eine 4:2:0 Unterabtastung (englisch Subsampling) statt. Ein solcher Helligkeitsblock und die dazugehörigen Farbblöcke bilden eine sogenannte Coding Unit (CU) mit zunächst gleicher Größe wie die CTU. Die Neuerung in H.265 ist, dass diese Blöcke nun in einer Baumstruktur mit jeweils vier Knoten (einem sogenannten Quadtree) je nach Inhalt des Bildes weiter in Prediction Units (PUs) und Transform Units (TUs) mit einer Größe zwischen 4 x 4 und 32 x 32 Pixeln unterteilt werden können [Sullivan et al. 2012]. Jedes Bild wird also so unterteilt, dass es möglichst an den Inhalt angepasst ist. Mit dieser Block-Unterteilung wird nun weiter gearbeitet.

In H.265 wird sowohl Intrapicture-Prediction, also die Vorhersage von Bildbereichen auf Grund von benachbarten Bildbereichen, als auch Interpicture-Prediction, also die Vorhersage von Bildbereichen auf Grund vorheriger Frames, verwendet. Bei der Intrapicture-Prediction gibt es drei verschiedene Methoden: Intra_DC, Intra_Planar und Intra_Angular. Die ersten beiden Methoden verwenden Durchschnittswerte von Referenzblöcken, während bei Intra_Angular 33 verschiedene Richtungen abdeckt werden, wodurch eine bessere Vorhersage gemacht werden kann. Wie bei H.264 gibt es je nach Richtung der Vorhersage eine Glättung der Referenz-Blöcke, um Unterbrechungen oder Artefakte an Blockgrenzen zu vermeiden [Sullivan et al. 2012]. Intrapicture-Prediction wird angewendet, wenn keine anderen Bilder für eine Interpicture-Prediction zur Verfügung stehen - zum Beispiel beim ersten Frame einer neuen Videosequenz - oder Interpicture-Prediction zu rechenaufwendig oder wenig effizient wäre [Wien 2015, S. 40].

Bei der Interpicture-Prediction werden in Referenzbildern Bildbereiche identifiziert, die sich verschoben haben und Bewegungsvektoren berechnet. Diese können eine Genauigkeit von einem achtel Pixel haben und geben an in welche Richtung eine Bewegung erfolgt ist. Beim ersten Bild einer Videosequenz wird die Intrapicture-Prediction verwendet, bei den übrigen Bildern die Interpicture-Prediction. Die Differenz zwischen vorhergesagtem Bild und tatsächlichem Bild wird anschließend weiter verarbeitet [Sullivan et al. 2012].

Zu dieser weiteren Verarbeitung zählen Transformation, Quantisierung und Entropiekodierung. Für die Transformation wird die Approximierung einer Diskreten Cosinustransformation (DCT) verwendet und für die Quantisierung ein Uniform Reconstruction Quantization Schema (URQ) mit einem Quantisierungsparameter zwischen 0 und 51, die die Schrittgröße der Quantisierung festlegen. Diese beiden Verarbeitungsschritte waren bereits in H.264 enthalten. Bei der Entropiekodierung wird Context-Adaptive Binary Arithmetic Coding (CABAC) verwendet [Sullivan et al. 2012]. Das Ziel ist es dabei Zeichenfolgen, die mehrfach vorkommen, so zu kodieren, dass sie insgesamt weniger Speicherplatz benötigen.

H.265 bietet verschiedene Profile an, mit denen Videos kodiert werden können. Dort sind maximale Samplerate, maximale Bildgröße, maximale Bitrate, minimales Kompressionsverhältnis und Kapazitäten für die Zwischenspeicher für die De-/Kodierung definiert. Unterschieden wird hier zwischen einer normalen Stufe („Main Tier“) für die meisten Anwendungen und einer hohen Stufe („High Tier“) für anspruchsvolle Anwendungen [Sullivan et al. 2012].

Insgesamt kann H.265 im Gegensatz zu seinem Vorgänger H.264 circa doppelt so viele Bits einsparen [Sullivan et al. 2012] und ist damit deutlich effizienter in der Videokompression.

2.3.2 VP9 (WebM)

Die Enkoder H.264 und auch H.265 sind allerdings nicht lizenzfrei. Aus diesem Grund startete Google 2010 das sogenannte WebM-Projekt, um ein offenes und lizenzfreies Medienformat speziell für die Verwendung im Web zu entwickeln. Dieses ist ein offenes Dateiformat, das sowohl Video- als auch Audioformate definiert. Der aktuell darin verwendete Videocodec ist VP9, der Nachfolger von VP8 [Mukherjee et al. 2013].

Einen Überblick über das VP9-Verfahren liefert [Mukherjee et al. 2013]. Demnach besitzt VP9 viele Ähnlichkeiten zu den Codecs H.264 und H.265 in seiner Vorgehensweise zur Videokodierung. So gibt es auch hier - wie bereits im Vorgänger VP8 - eine grundsätzliche Blockstruktur bei der Verarbeitung von Frames. Bei VP9 können diese Blöcke - ähnlich wie bei H.265 - eine Blockgröße von 64 x 64 Pixeln haben und können dann rekursiv auf bis zu 4 x 4 Pixel verkleinert werden.

Die grundsätzliche Vorgehensweise ist ebenfalls ähnlich zu der von H.265. Nach der Unterteilung der Einzelbilder in Blöcke wird Intra- oder Inter-Prediction durchgeführt und Bewegungen abgeschätzt bzw. vorhergesagt, gefolgt von einer anschließenden Umformung und Kodierung.

Bei der Intra-Prediction existieren zehn verschiedene Modi für Blöcke zwischen 4 x 4 und 32 x 32 Pixel Größe, die auch verschiedene festgelegte Winkel beinhalten. Bei der Inter-Prediction gibt es vier verschiedene Modi für Blöcke zwischen 4 x 4 und 64 x 64 Pixel Größe. Die Bewegungsvektoren können eine Genauigkeit von maximal einem achtel Pixel erreichen, alternativ kann die Genauigkeit jedoch auch auf einen viertel Pixel umgestellt werden.

Bei der anschließenden Verarbeitung der Daten sind drei verschiedene Transformationen möglich. Mit Inter-Prediction kodierte Blöcke verwenden eine Diskrete Kosinustransformation (DCT) wie bei H.265, bei Intra-kodierten Blöcken ist zusätzlich die Anwendung einer Asymmetrischen Diskreten Sinustransformation (ADST) möglich. Bei der gewählten Option der verlustfreien Kodierung (festgelegt durch den niedrigsten möglichen Quantisierungsparameter) wird eine Walsh-Hadamard Transformation (WHT) durchgeführt. Im Anschluss an die Transformationen und Quantisierung werden die Daten durch die Anwendung einer arithmetischen Kodierung kodiert.

Um Artefakten durch die Verwendung einer Blockstruktur entgegenzuwirken wird bei VP9 (wie auch schon bei H.265) ein Loop-Filter angewendet, der drei verschiedene Filter enthält, die je nach Transformationsgröße angewendet werden.

Des Weiteren bietet VP9 ein Segmentierungs-Framework an, das es erlaubt das Bild durch Angabe einer Segmentierungs-Map in bis zu acht Segmente zu unterteilen. Dadurch soll es ermöglicht werden, für die Segmente veränderte Quantisierungsparameter, die Stärke des Loop-Filters oder einen Block-Skip-Mode zu spezifizieren. Das Ziel ist es, verschiedene Tools anwendungsspezifisch anzuwenden, um die Kodierungsqualität zu verbessern.

Zudem unterstützt VP9 noch weitere Funktionen speziell für die Übertragung via Internet, was der Haupteinsatzzweck des Codecs sein sollte. Es kann daher ein spezielles Flag, das die Fehlertoleranz erhöhen soll, verwendet werden, das dafür sorgt, dass die Dekodierung von Frames auch bei Fehlern oder verloren gegangenen Frames fortgesetzt werden kann. Dies geschieht, indem die Werte bei der Entropiekodierung an jedem Frameanfang zurückgesetzt werden, wodurch allerdings ca. 4-5% Performanzverlust eintritt.

Außerdem ist eine parallele Entropiedekodierung von Frames durch lediglich Parsen der Frame-Header und durch sogenannte Tiles (unabhängig kodierten Untereinheiten eines Frames) möglich, wodurch eine höhere Geschwindigkeit bei der Dekodierung mit Threads erreicht werden soll.

Beim Vergleich mit H.264 und H.265 in [Sharabayko Markov 2016] zeigte VP9 Kompressionsergebnisse im Bereich zwischen H.264 und H.265. Es ist also nicht ganz so effizient wie

H.265, jedoch besser als H.264 und bietet damit eine Alternative zu den nicht lizenzfreien Verfahren H.264 und H.265.

Google begann zwar 2013 mit dem geplanten Nachfolger von VP9 (VP10), dieser ist allerdings nicht als solcher veröffentlicht worden, sondern große Teile des Codecs sind in einen neuen Codec AV1 übergegangen, an dem zur Zeit noch gearbeitet wird. Eine Fertigstellung ist für 2017 geplant. Aus diesem Grund ist VP9 der aktuelle und letzte Codec der VP-Reihe [Ozer 2016].

2.3.3 Weitere

Weitere bekannte Verfahren sind beispielsweise DivX oder VC-1. DivX (Digital Video Express) ist ein inzwischen als Open-Source weiterentwickeltes Verfahren von DivXNetworks, das viele Elemente des MPEG-4-Verfahrens enthält, da es vom Standard H.261 abgeleitet ist. Beispielsweise enthält DivX eine sogenannte globale Bewegungskompensation (GMC), die zur Datenreduktion bei Kamerabewegungen bei statischen Bildern verwendet wird. Des Weiteren ist hier ein Umschalten der Genauigkeit von einem halben Pixel auf einen viertel Pixel möglich, wodurch Vorhersagen genauer werden [Henning 2007, S. 212f].

VC-1 ist der Codec, der im Windows Media Video (WMV) Format von Microsoft verwendet wird. Auch dort ist die Hauptfunktionalität zur Datenkompression die Unterteilung des Bildes in Blöcke und die anschließende Anwendung einer Bewegungsabschätzung wie in MPEG. Ebenso wird ein Deblocking-Filter verwendet, um Artefakte, die auf Grund der Unterteilung in Blöcke entstanden sind, zu entfernen. Außerdem sind wie in den zuvor vorgestellten Kompressionsverfahren verschiedene Profile und Level verfügbar, die je nach Anforderung ausgewählt werden können und steuern welche Eigenschaften das Resultatvideo haben soll [Loomis Wasson 2007].

2.4 Menschliche Wahrnehmung

Bei verschiedenen Videokompressionsverfahren wird versucht bei einzelnen Bildern die räumliche Redundanz zu entfernen, also die Redundanz von Pixeln oder ganzen Bereichen innerhalb eines Bildes. Bei Videosequenzen wird versucht die zeitliche Redundanz zu entfernen, also die Redundanz zwischen aufeinanderfolgenden Bildern. Die Entfernung dieser Redundanzen hat das Ziel die großen Datenmengen von Videos deutlich zu reduzieren. Ein weiterer wichtiger Punkt, der ebenfalls Datenmengen reduzieren könnte, ist der Einbezug der menschlichen Wahrnehmung in die Videokompression und das Ausnutzen der menschlichen Wahrnehmung für eben diese.

Um Eigenschaften der menschlichen Wahrnehmung ausnutzen zu können, ist es wichtig zu wissen, wie diese funktioniert und welche Eigenschaften überhaupt erst vorhanden sind, die man in der Videokompression verwenden und ausnutzen könnte, um vom Menschen möglichst unbemerkt Datenmengen einzusparen.

Der Mensch kann sehen, indem Licht, das von Objekten in einer Szene reflektiert wird, in sein Auge, insbesondere auf eine Rezeptorfläche, die sogenannte Netzhaut, trifft. Sie besteht aus ca. 75-150 Millionen Stäbchen, die für die Helligkeitswahrnehmung zuständig sind, und ca. 6-7 Millionen Zapfen, die für die Farbwahrnehmung zuständig sind und nur geringfügig für die Helligkeitswahrnehmung [Erhardt 2008, S.11]. Durch die deutlich höhere Anzahl an Stäbchen im Gegensatz zu den für die Farbwahrnehmung benötigten Zapfen im menschlichen Auge ist der Mensch also empfindlicher bei Helligkeitsunterschieden als bei Farbunterschieden. Zapfen sind zudem in der Lage feinere Details und schnelle Veränderungen wahrzunehmen [Chen Lin Ngan 2010].

In der Mitte der Netzhaut befindet sich die Sehgrube (englisch Fovea), die die Sehschärfe bringt, da sich in ihrem Bereich fast ausschließlich Zapfen befinden [Erhardt 2008, S.11], wie die folgende Abbildung 2 verdeutlicht.

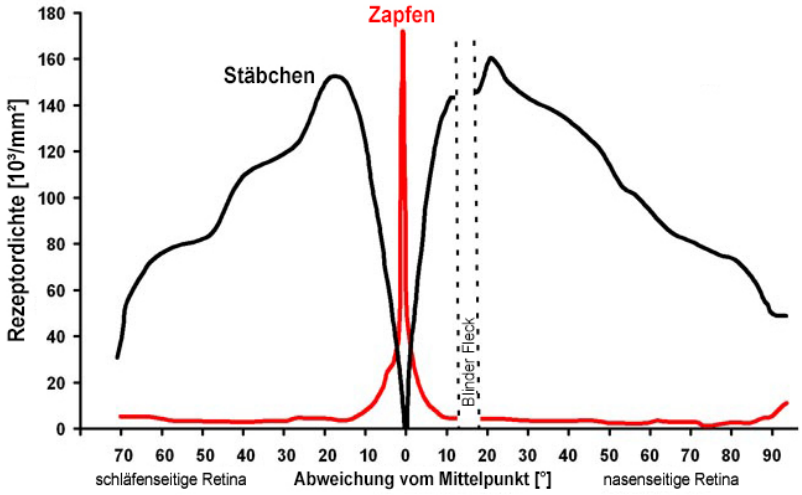

Abbildung 2: Verteilung der Stäbchen und Zäpfen im menschlichen Auge (in Anlehnung an [Kalloniatis Luu 2007])

Die Abbildung zeigt die Verteilung der Stäbchen und Zapfen im menschlichen Auge im Bezug zum Mittelpunkt des Auges. Es wird deutlich, dass die Rezeptordichte der Stäbchen, deren Aufgabe die Helligkeitswahrnehmung ist, vom äußersten Auge in Richtung der Mitte immer weiter ansteigt, sowohl auf der linken als auch auf der rechten Seite. Im Bereich zwischen ungefähr 20° sinkt die Dichte auf beiden Seiten deutlich ab. Im Gegensatz dazu ist die Rezeptordichte der Zapfen in den äußeren Bereichen sehr gering und steigt ab ca. 10° zur Mitte hin deutlich an auf über 170.000 Zapfen pro Quadratmillimeter. Im Bereich bei etwa 15° nasenseitig befindet sich der sogenannte Blinde Fleck, an dem der Sehnerv in den Augapfel eintritt [Erhardt 2008, S.11], wo weder Zapfen noch Stäbchen vorhanden sind. Der Mensch kann an dieser Stelle kein Bild wahrnehmen.

Diese Verteilung der Zapfen und Stäbchen bedeutet nun folgendes: Auf Grund der hohen Dichte der Stäbchen in den äußeren Bereichen des Auges nimmt der Mensch Helligkeitsunterschiede deutlich wahr, allerdings wegen der geringen Dichte der Zapfen nur wenige Farbunterschiede. In dem kleinen Bereich in der Mitte des Auges nimmt er dagegen Farbunterschiede deutlich wahr und nur geringe Helligkeitsunterschiede.

Durch die Stäbchen und Zapfen entstehen Spannungsimpulse, die über den Sehnerv an das Sehzentrum im Gehirn weitergeleitet werden, wodurch der Mensch das Bild wahrnehmen kann, dass er gerade mit seinen Augen sieht [Erhardt 2008, S.12].

Zu diesen physikalisch bedingten Eigenschaften kommen weitere Eigenschaften hinzu. Bei der Wahrnehmung von Oberflächen gibt es beispielsweise Unterschiede. Der Mensch ist nicht so empfindlich bei der Wahrnehmung von sich wiederholenden Mustern, beispielsweise bei statischen Texturen wie Holz oder dynamischen Texturen wie fließendem Wasser [Zhang Bull 2011]. Das hängt damit zusammen, dass bei gleichzeitigem Auftreten vieler verschiedener Reize, wie es bei Texturen beispielsweise der Fall ist, ein Reiz einen anderen nicht so starken Reiz reduziert oder eliminiert, also überlagert. Dies wird auch als visuelle Maskierung (englisch visual Masking) bezeichnet - ein Fehler oder kleinere Veränderungen in einer texturierten Region werden dadurch schlechter erkannt als ein einer glatten nicht-texturierten Region [Lee Ebrahimi 2012].

Der Mensch ist zudem weniger empfindlich gegenüber Fehlern bzw. Abweichungen in hoch-

frequenten Bereichen als in niedrigfrequenten. Dies kann z.B. ebenfalls in Texturen der Fall sein. Sind die Kontraste eines Signals unterhalb eines bestimmten Schwellwertes, ist der Mensch mit seiner visuellen Wahrnehmung nicht in der Lage diese wahrzunehmen. Dieser Schwellwert wird als differentielle Wahrnehmbarkeitsschwelle (JND, englisch Just-Noticeable Difference) bezeichnet. Erst bei Überschreiten dieser Schwelle ist der Mensch fähig Unterschiede wahrzunehmen [Lee Ebrahimi 2012].

Dem Menschen werden außerdem permanent sehr große Massen von visuellen Informationen durch seine Augen zugeführt. Die Masse der Informationen ist so groß, dass der Mensch es nicht schafft sie vollständig zu verarbeiten. Der Mensch, im speziellen sein Gehirn, muss hier also priorisieren. Das passiert durch die sogenannte selektive Aufmerksamkeit. Das Gehirn bestimmt also, welche Informationen bzw. Daten von größtem Interesse sind und verarbeitet diese weiter [Fintrop Rome Christensen 2009]. Einen ähnlichen Effekt, den sogenannten Cocktailparty-Effekt, gibt es z.B. auch bei den akustischen Daten, die durch das Ohr ans Gehirn zugeführt werden. Während sich beispielsweise in einem Raum viele Personen miteinander unterhalten, kann sich der Mensch dennoch auf die Stimme eines einzelnen anderen, mit dem er sich unterhält, konzentrieren [Fintrop Rome Christensen 2009]. Auch hier wird zwischen wichtigen Informationen und nicht so wichtigen Informationen im Gehirn unterschieden. Das Sehen funktioniert ebenfalls nach einem vergleichbaren Prinzip.

Bei dieser selektiven Aufmerksamkeit wird im Gehirn immer nur ein kleiner Bereich einer Szene im Detail in einem Moment analysiert. Das ist in der Regel die Region, die von den Augen fixiert wird. Die übrigen Bereiche der Szene werden dabei größtenteils ignoriert, kleinere Änderungen in diesen nicht fokussierten Bereichen bleiben vom Menschen also weitgehend unbemerkt. Der Mensch ist jedoch ebenso in der Lage automatisch interessante Bildbereiche, sogenannte Regions-Of-Interest (ROI) in der Umgebung wahrzunehmen und seinen Fokus und seine Aufmerksamkeit dorthin zu ändern [Fintrop Rome Christensen 2009]. Menschliche Gesichter sind dabei einer der wichtigsten Quellen in Bildern, die menschliche Aufmerksamkeit auf sich ziehen [Lee Ebrahimi 2012].

Nach [Fintrop Rome Christensen 2009] gibt es zwei verschiedene Kategorien von Faktoren, die die Aufmerksamkeit steuern und auf sich ziehen. Zum einen gibt es die sogenannten „Bottom-Up“-Faktoren, die ausschließlich aus der visuellen Szene heraus auftreten. Hier gibt es Bereiche, die besonders im Gegensatz zu anderen herausstechen, wie z.B. eine Person mit einem auffälligen roten Pullover in mitten einer grünen Wiese. Diese Bereiche werden als auffällig (englisch salient) bezeichnet. Bottom-Up Effekte sind nur sehr schwer zu unterdrücken, eine hoch auffällige Region in einem Bild lenkt in der Regel die Aufmerksamkeit des Zuschauers auf sich.

Zum anderen gibt es die sogenannten „Top-Down“-Faktoren, die Aufmerksamkeit durch geistige Faktoren des Zuschauers wie Wissen oder Erwartungen auf sich ziehen. Beispielsweise ziehen gelbe Gegenstände auf einem Schreibtisch mehr Aufmerksamkeit auf sich, wenn der Zuschauer nach einem gelben Gegenstand sucht [Fintrop Rome Christensen 2009].

Außerdem beeinflussen sich visuelle und auditive Wahrnehmung gegenseitig. Aufmerksamkeit erlangen also auch Bildbereiche, die durch Audiosignale hervorstechen, sofern es möglich ist diese dem Bildbereich zuzuordnen [Lee Ebrahimi 2012].

Analog zur menschlichen Wahrnehmung gibt es auch bei der Videokodierung die Problematik, dass sehr viele Bilddaten vorhanden sind. Diese könnten zwar im Gegensatz zu menschlichen Wahrnehmung von der Menge her verarbeitet werden, diese Verarbeitung kostet jedoch sowohl Zeit und einen hohen Rechenaufwand bei der Kodierung und Dekodierung eines Videos, als auch Speicherplatz beim späteren Speichern und Übertragen eines Videos. Es bietet sich also an auch hier zu priorisieren, was in den üblichen Standards wie HEVC ja bereits in Form des Farb-Unterabtastung passiert.

2.5 Wahrnehmungsbasierte Kompression

Aktuelle Standards der Videokompression versuchen mit verschiedenen Verfahren räumliche und zeitliche Redundanzen aus Videos zu entfernen, um die Datenmengen möglichst klein zu halten. Die menschliche Wahrnehmung bietet jedoch ebenfalls viele Möglichkeiten Videodaten zu komprimieren.

Viele Eigenschaften der menschlichen Wahrnehmung aus Kapitel 2.4 eignen sich, um die Kodierungseffizienz von Videokompressionsverfahren ohne für den Mensch signifikant wahrnehmbaren Qualitätsverlust zu verbessern [Lee Ebrahimi 2012]. Die Zielgruppe von Videos sind eben Menschen und daher können diese Eigenschaften für die Videokompression ausgenutzt werden. Man möchte sich also die Eigenschaften der menschlichen Wahrnehmung zusätzlich zu den üblichen Verfahren der Videokompression zu Nutzen machen und die Videokompression dadurch verbessern.

Bereits in üblichen Kompressionsverfahren werden solche Eigenschaften vereinzelt ausgenutzt. In H.265 und seinen Vorgängern, aber auch anderen gängigen Verfahren, werden Farbkomponenten und Helligkeitskomponenten voneinander getrennt und bei den Farbwerten eine Unterabtastung vorgenommen, d.h. die Farbwerte haben eine geringere Auflösung als die Helligkeitswerte. Diese kann eben wegen der schlechteren Farbwahrnehmung des Menschen durchgeführt werden, ohne dass dieser einen signifikanten Unterschied merkt. Er kann die Farbwerte genauso gut wahrnehmen, obwohl diese beispielsweise nur noch halb so oft vorhanden sind [Wien 2015, S.35].

Generell können Ansätze der wahrnehmungsbasierten Videokompression in verschiedene Kategorien unterteilt werden. In [Chen Lin Ngan 2010] erfolgt eine solche Kategorisierung in Sehmodell-basierte Ansätze, signalgesteuerte Ansätze und gemischten Ansätzen. Sehmodell-basierte Ansätze sind Ansätze, die auf den Eigenschaften der menschlichen optischen Wahrnehmung, wie die in Kapitel 2.4 beschriebene selektive Aufmerksamkeit, basieren. Bei diesen Verfahren werden bestimmte Regionen eines Bildes bzw. einer Sequenz von Bildern je nach ihrem Inhalt mit besserer Qualität kodiert als andere Regionen. Da der Mensch ohnehin nur auf bestimmte Bereiche eines Bildes achtet, können die übrigen Bereiche mit entsprechend geringerer Qualität kodiert werden. Zu signalgesteuerten Ansätzen zählen Ansätze, die auf der Analyse visueller Merkmale der Signalverarbeitung basieren. Hier wird ausgenutzt, dass der Mensch beispielsweise Texturen nur wenig detailliert wahrnehmen kann und Blöcke mit diesen z.B. in niedriger Qualität kodiert werden als andere Blöcke mit glatten Bereichen oder aber gar nicht kodiert werden, sondern erst im Decoder synthetisiert und wiederhergestellt werden. Unter gemischten Ansätzen werden Vorgehensweisen zusammengefasst, die Verfahren aus beiden der vorherigen Kategorien beinhalten. Durch Kombinationen verschiedener Verfahren können Vorteile gegenüber einzelnen Verfahren entstehen. Möglich ist beispielsweise eine Kombination der differentielle Wahrnehmbarkeitsschwelle (JND) mit weiteren Verfahren wie Gesichtserkennungen.

Vor allem bei den Sehmodell-basierten Ansätzen, die mit der menschlichen Aufmerksamkeit arbeiten, wird davon ausgegangen, dass unabhängig vom jeweiligen Zuschauer immer die gleichen Bereiche in einem Bild die Aufmerksamkeit auf sich ziehen und immer die gleichen Bereiche vom Menschen als wichtig betrachtet werden [Lee Ebrahimi 2012]. Umso wichtiger ist es zu verstehen, wie genau die selektive Aufmerksamkeit des Menschen funktioniert.

Neben diesen drei Kategorien gibt es in [Lee Ebrahimi 2012] noch zwei weitere Kategorien, wobei eine von ihnen die manuelle Auswahl von für den Zuschauer wichtigen Bildregionen umfasst. Diese Auswahl kann z.B. durch Eye-Tracking erfolgen, aber auch durch eine manuelle Auswahl durch den Zuschauer per Mausklick oder per Touchpad. Diese Kategorie könnte bei der obigen Kategorisierung jedoch auch bei Sehmodell-basierten Ansätzen eingeordnet werden. Die andere Kategorie beinhaltet sogenannte multimodale Ansätze. Multimodale Ansätze

behandeln sowohl visuelle als auch akustische Signale in ihrem Zusammenhang. Es werden zum Beispiel nicht nur Regionen, die visuelle Aufmerksamkeit auf sich ziehen berücksichtigt, sondern auch solche, von denen akustische Signale ausgehen. Diese werden dann als Regions-Of-Interest behandelt und dadurch in besserer Qualität kodiert als andere. Abbildung 3 stellt die verschiedenen Kategorien aus [Chen Lin Ngan 2010] und [Lee Ebrahimi 2012] und ihre Ansätze in einer Übersicht grafisch dar.

Sehmodellbasiert	Signalgesteuert	Multimodal	Gemischt
- Hintergrund/Vordergrund - Gesichtserkennung - Bewegende Objekte - Saliency-Map	- Masking-Effekte - Textursynthese - Differentielle Wahrnehmbark. (JND)	- Akustische & Visuelle Wahrnehmung - Sound-Emitting Region	- Texturwahrnehmung + JND - Gesichtserkennung + JND

Abbildung 3: Kategorien wahrnehmungsbasierter Ansätze
nach [Chen Lin Ngan 2010] und [Lee Ebrahimi 2012]

Im Allgemeinen gibt es zwei Möglichkeiten wahrnehmungsbasierte Kompressionsverfahren umzusetzen. Zum einen ist es möglich die Einzelbilder einer Videosequenz noch vor der eigentlichen Kodierung mit üblichen Kompressionsverfahren, wie zum Beispiel H.265, in einem sogenannten Pre-Processing vorzuverarbeiten. Diese ist die einfachste Möglichkeit der Implementierung und bietet den Vorteil, dass für die eigentliche Kompression konventionelle Kompressionsverfahren ohne Veränderung verwendet werden können. Möglich ist hier beispielsweise die Verwendung von Filtern, um unwichtige Regionen eines Bildes weichzuzeichnen (sogenannte Blurring-Filter). Solche Blurring-Filter können zusätzlich den positiven Nebeneffekt haben, dass weniger Artefakte durch die Verarbeitung des Bildes in Blöcken auftreten. [Lee Ebrahimi 2012].

Zum anderen können neue Algorithmen und Verfahren entweder in komplett neuen Encodern implementiert werden oder bereits vorhandene verändert werden. Diese Art der Umsetzung ist entsprechend aufwendiger. Meistens bauen wahrnehmungsbasierte Videokompressionsverfahren auf traditionelle Verfahren auf und es erfolgt eine Veränderung dieser durch Anpassungen von bestimmten Parametern, wie beispielsweise eine Veränderung des Quantisierungsparameters für bestimmte Bildbereiche, wobei ein großer Quantisierungsparameter eine schlechtere Qualität und ein niedriger eine bessere Qualität für die Bildbereiche zur Folge hat oder eine Festlegung einer maximalen Bitzahl für bestimmte Bildbereiche [Lee Ebrahimi 2012]. Die Erweiterung von bestehenden Verfahren hat den Vorteil, dass die üblichen Algorithmen wie die Bewegungsabschätzung oder die Entropiekodierung nicht neu implementiert werden müssen.

Obwohl sich viele verschiedene Wissenschaften - unter anderem die Biologie, die Psychologie und die Neurowissenschaften - mit der menschlichen Wahrnehmung befassen, sind nach wie vor noch nicht alle Funktionalitäten und Mechanismen vollständig erforscht und verstanden [Lee Ebrahimi 2012, Xu et al. 2014, Chen Li 2015]. Es ist also auch in der Zukunft möglich, dass weitere Eigenschaften der menschlichen Wahrnehmung entdeckt werden, die es ermöglichen das menschliche visuelle System anders oder auch besser für die Videokompression auszunutzen.

3 Kriterien für den Vergleich von Videokompression

Um vergleichen oder einschätzen zu können wie gut oder wie schlecht ein Videokompressionsverfahren - auch im Vergleich zu anderen Verfahren - ist, können verschiedene Kriterien in Betracht gezogen werden, die in den folgenden Unterkapiteln näher beschrieben werden. Außerdem werden Bewertungsmethoden vorgestellt, die eine Bewertung der entstandenen Videoqualität ermöglichen.

3.1 Dateigröße

Ein erstes Kriterium, um zu beurteilen wie gut bzw. wie stark die Kompression in einem Videokompressionsverfahren ist, ist die Dateigröße der resultierenden Videodatei. Je kleiner die Dateigröße, desto besser ist die Kompression des Verfahrens. Denn durch die Dateigröße wird nicht nur viel Speicherplatz eingenommen, sondern bei der Übertragung der Videodaten, beispielsweise via Internet, müssen diese Daten versendet werden. Je kleiner das Video komprimiert werden kann, desto weniger Daten müssen verschickt werden. Je nach Verbindung - vor allem bei mobilen Endgeräten - kann dies einen großen Zeitunterschied beim Laden eines Videos bedeuten.

Übliche Verfahren wie H.265/HEVC sind in der Lage Videos in ihrer Dateigröße deutlich zu reduzieren. Damit ist es beispielsweise möglich verlustbehaftet mit einem Quantisierungsparameter von 32 ein Video in Full-HD-Auflösung von 600 Megabyte Dateigröße auf ca. 1 Megabyte so zu reduzieren, dass der Inhalt noch gut erkennbar ist, wodurch aber eine deutliche Bitersparnis entsteht. Es gilt allerdings zu beachten, dass bestimmte Daten in jedem Video schon auf Grund des Video- bzw. Dateiformats vorhanden sind, die auch nicht komprimiert werden. Dazu gehört zum Beispiel ein sogenannter Header direkt am Anfang der Videodatei, in dem unter anderem festgelegt ist, um was für einen Dateityp es sich handelt (sogenannter Magic-String). Diese Daten machen in der Regel aber keine signifikante Datenmenge aus.

Ein Vergleich zweier Videos über eine geringere Dateigröße macht allerdings nur dann Sinn, wenn die Qualität (z.B. Auflösung, keine starken Verzerrungen, Verfälschungen im Gegensatz zum Originalvideo, etc.) der beiden Videos gleich oder vergleichbar ist. Ein Videokompressionsverfahren ist also dann besser, wenn das resultierende Video in einer vergleichbar guten Qualität - oder einer durch den Menschen gleich wahrgenommenen Qualität - weniger Speicherplatz benötigt.

3.2 Performanz

Ein weiteres Kriterium zur Beurteilung eines Videokompressionsverfahrens ist die Performanz des Verfahrens während der Kodierung und Dekodierung. Je nach Rechenaufwand um ein Video mit einem bestimmten Verfahren zu kodieren, wird die Zeit länger, die dafür benötigt wird. Eine bessere Performanz bzw. ein niedrigerer Rechenaufwand und eine daraus resultierende geringere Zeit, die für das Kodieren benötigt wird, ist also besser. Für das als Beispiel gewählte HD-Video aus Kapitel 3.1 wird beispielsweise eine Rechenzeit von 8448 Sekunden (bei einem 64-Bit Betriebssystem mit 8 Gigabyte RAM und Intel Core i5 CPU) für die Kodierung in H.265/HEVC mit der Referenzsoftware HM9.0 benötigt, was ungefähr 2,5 Stunden entspricht und damit weit entfernt ist von einer wünschenswerten Echtzeitkodierung.

Ein Vergleich der Performanz von Videokompressionsverfahren ist nur dann aussagekräf-

tig, wenn für alle Vergleiche die selben Grundbedingungen gelten. Oft nimmt man jedoch eine schlechtere Performanz bzw. höhere Rechenzeit für eine bessere Kompression oder eine bessere Qualität und geringere Dateigröße in Kauf. Einige Performanzeinbußen sind allerdings so gut wie unvermeidbar, beispielsweise die Anwendung von Quantisierung und diskreter Kosinustransformation und anschließende Umkehrung dieser für die Prediction bei MPEG, auf dessen resultierendem Bild die Vorhersage getätigt wird. Die Performanz hat zudem keine Auswirkungen auf die aus dem Verfahren resultierende Videoqualität an sich oder die resultierende Dateigröße, sondern sagt lediglich etwas darüber aus, wie gut bzw. wie effizient die im Kompressionsverfahren verwendeten Algorithmen sind und wie schnell dieses arbeitet.

3.3 Videoqualität und Optik

Das wichtigste Kriterium, mit dem verschiedene Videokompressionsverfahren verglichen und bewertet werden können, ist die Qualität und die Optik des Resultats. Eine geringere Dateigröße macht ein Kompressionsverfahren nur dann effizient oder gut, wenn die Qualität des Videos nach der Datenreduktion vergleichbar gut bleibt.

Generell gibt es zwei verschiedene Möglichkeiten die Qualität von Videos zu bewerten. Zum einen die subjektive Qualitätsbewertung und zum anderen die objektive Qualitätsbewertung. Bei der subjektiven Qualitätsbewertung werden einer größeren Zahl von Probanden Videos gezeigt (nach [Wu Rao 2005, S.125ff] liefern zwischen 16 und 24 Probanden ein statistisch stichhaltiges Ergebnis), die sie im Hinblick auf ihre optisch wahrgenommene Qualität im Vergleich zu Referenzvideos bewerten sollen. Als Probanden können entweder Experten im Gebiet der Videoverarbeitung oder auch Nicht-Experten befragt werden. Dabei kann es von Vorteil sein vor allem auch Nicht-Experten miteinzubeziehen, da diese Artefakte bemerken können, die Experten nicht zwingend sehen, da sie auf andere Dinge achten [Wu Rao 2005, S.125ff]. Die subjektive Qualitätsbewertung ist der präziseste und zuverlässigste Weg die tatsächliche Bildqualität zu bestimmen, ist allerdings auch deutlich zeitaufwendiger und teurer als andere Verfahren [Lee Ebrahimi 2012].

Bei der objektiven Qualitätsbewertung werden physikalische Eigenschaften eines Videos bewertet bzw. gemessen. Die bekanntesten Methoden zur objektiven Bewertung sind der mittlere quadratische Fehler (MSE, englisch Mean Squared Error) und das Spitzen-Signal-Rausch-Verhältnis (PSNR, englisch Peak Signal to Noise Ratio) [Wu Rao 2005, S.159].

Der mittlere quadratische Fehler MSE ist definiert als der durchschnittliche quadrierte Unterschied zwischen den Grauwerten zweier Bilder oder Bildsequenzen:

$$MSE = \frac{1}{TMN} \sum_{t} \sum_{m} \sum_{n} [x_0(m,n,t) - x_r(m,n,t)]^2,$$

wobei das zu bewertende Video eine Größe von MxN Pixeln hat und die zu betrachtende Sequenz T Frames beinhaltet. x_0 und x_r bezeichnen das ursprüngliche Bild bzw. die ursprüngliche Sequenz (0) und das bzw. die aus dem angewendeten Verfahren resultierende (r) [Wu Rao 2005, S.159]. Der mittlere quadratische Fehler ist entsprechend gering, wenn die Videoqualität hoch ist bzw. um genauer zu sein, wenn das Resultatvideo mehr dem ursprünglichen Video entspricht, die entstandenen Fehler bei der Kodierung also gering sind.

Das Spitzen-Signal-Rausch-Verhältnis $PSNR$ gibt an wie klar das Eingangssignal im Resultatvideo noch ankommt bzw. wie stark das resultierende Bild oder die resultierende Sequenz dem Original ähnelt:

$$PSNR = 10log_{10}\frac{I^2}{MSE},$$

wobei I der maximale Wert ist, den ein Pixel haben kann (bei 8 Bit 255) [Wu Rao 2005, S.159]. Das Spitzen-Signal-Rausch-Verhältnis ist bei hoher Videoqualität bzw. großer Ähnlichkeit zum Originalvideo ebenfalls hoch. Da die Effizienz der Videokompressionsverfahren auch auf den Inhalt der Videos ankommt - bei vielen Szenenwechseln mit viel Bewegung ist z.B. nicht so viel Bewegungsvorhersage möglich wie bei langen Szenen mit wenig Bewegung -, sollten für die genauere Bewertung der Videoqualität außerdem mehrere verschiedene Videosequenzen verwendet werden [Wu Rao 2005, S.127], die ein Gesamtergebnis bestimmen. Zudem sollte der Vergleich von zwei Verfahren mit den selben Videos stattfinden, da ein Vergleich von PSNR-Werten nur Sinn macht, wenn als Eingabe das selbe Video verwendet wurde.

3.4 Bewertung von wahrnehmungsbasierter Kompression

Generell sind die üblichen Methoden zur objektiven Bewertung von Videokompressionsverfahren wie mittlerer quadratischer Fehler (MSE) oder auch Spitzen-Signal-Rausch-Verhältnis (PSNR) nur bedingt geeignet, um die Videoqualität verschiedener Verfahren zu bewerten und zu vergleichen. Die folgende Abbildung verdeutlicht dies.

Abbildung 4: Unterschiedliche Wahrnehmung von Rauschen [Wu Rao 2005, S.160]

Auf dem linken und rechten Bild in Abbildung 4 wurde der jeweils gleiche Anteil an Rauschen hinzugefügt. Auf dem linken Bild wurde das Rauschen im unteren Bereich des Bildes hinzugefügt, auf dem rechten Bild im oberen Bereich des Bildes. Da beide Bilder vor dem Hinzufügen des Rauschens identisch waren und beiden der gleiche Anteil an Rauschen hinzugefügt wurde, haben beide Bilder den selben Wert beim Spitzen-Signal-Rausch-Verhältnis (PSNR). Beide Bilder müssten also objektiv betrachtet die gleiche Videoqualität besitzen. Der Mensch mit seinen visuellen Eigenschaften nimmt die unterschiedliche Positionierung des Rauschens in den beiden Bildern jedoch unterschiedlich wahr. Im linken Bild wird das Rauschen auf Grund des Masking-Effektes für das menschliche Auge kaum sichtbar, denn hier sind ohnehin hohe Frequenzen durch Felsen und Wasser im Bild vorhanden, wodurch das Rauschen überlagert wird und so gut wie gar nicht mit dem bloßen Auge auffällt. Im rechten Bild ist im oberen Bereich vor allem der glatte Himmel mit nur wenigen hohen und vor allem niedrigen Frequenzen und gleichmäßigen Farben. Hier wird das Rauschen kaum durch den Masking-Effekt überlagert und für den Menschen deutlich sichtbar. Obwohl beide Bilder den gleichen PSNR-Wert besitzen, ist die durch den Menschen wahrgenommene Qualität im rechten Bild deutlich schlechter als im linken Bild.

Pixelbasierte Metriken zur Bewertung von Bildern und Videos wie MSE oder PSNR sind zwar relativ einfach zu berechnen, sind aber nicht zwingend zuverlässige Bewertungsmaßstäbe für die vom Menschen wahrgenommene Qualität, da sie - wie Abbildung 4 gezeigt hat - nicht miteinbeziehen wie das menschliche Sehen funktioniert [Wu Rao 2005, S.160].

Für die Bewertung von wahrnehmungsbasierten Kompressionsverfahren ist die Verwendung von solchen Bewertungsverfahren in ihrer üblichen Form also nur eingeschränkt sinnvoll, da in wahrnehmungsbasierten Kompressionsverfahren besonders auf die menschliche Wahrnehmung Bezug genommen wird und diese in besonderem Maße ausgenutzt wird. Hier machen Verfahren zur objektiven Bewertung Sinn, die ebenfalls bestimmte Teile eines Bildes oder Videos unterschiedlich gewichten, nämlich so wie der Mensch die Teile wahrnimmt. Bewertungen durch PSNR machen beispielsweise nur dann Sinn, wenn man sie für den Vergleich von bestimmten Bildbereichen verwendet. Bei Verfahren, die einen Hintergrund in schlechterer Qualität kodieren als den Vordergrund eines Videos, würde es Sinn machen, die PSNR-Werte des Vordergrundes einzeln zu betrachten und die PSNR-Werte des Hintergrundes. Schlechtere PSNR-Werte im Vordergrund würden die wahrgenommene Bildqualität tatsächlich verschlechtern, während dies bei schlechten PSNR-Werten im Hintergrund nur geringfügig der Fall wäre. Bei dem Beispielbild in Abbildung 4 wäre es also z.B. sinnvoll den Bildbereich mit dem Himmel einzeln miteinander zu vergleichen, denn dann würde man feststellen, dass der PSNR-Wert im rechten Bild schlechter ist als im linken. Allerdings ist diese manuelle Unterteilung und Bewertung deutlich zeitaufwendiger und arbeitsintensiver als ein automatisches Verfahren.

Alternativ kann eine subjektive Bewertung vorgenommen werden, die diese menschlichen Aspekte ohnehin für die Bewertung verwendet, da sie durch Menschen durchgeführt wird. Diese ist allerdings sehr zeitaufwendig sowohl in Vorbereitung, Ausführung als auch in der Auswertung und benötigt zusätzlich eine größere Anzahl an unterschiedlichen Testpersonen [Wu Rao 2005, S.155].

Die Bewertungsmetriken können generell in drei verschiedene Kategorien eingeteilt werden: Full-Reference (FR) Metriken vergleichen Frame für Frame zweier Videosequenzen (ursprüngliches Video als Referenz und das aus dem verwendeten Verfahren resultierende Video) miteinander, No-Reference (NR) Metriken bewerten das resultierende Video unabhängig vom ursprünglichen Referenzvideo und Reduced-Reference (RR) Metriken beziehen nur ausgewählte Eigenschaften des Referenzvideos im Vergleich zum resultierenden Video in die Bewertung ein [Wu Rao 2005, S.158].

Häufig wurden die üblichen Bewertungsmethoden wie MSE und PSNR im Hinblick auf die menschliche Wahrnehmung angepasst. Auf Grundlage des PSNR gewichtet das Verfahren bei FPSNR (Foveal-Peak-Signal-To-Noise-Ratio) beispielsweise in Anlehnung an die menschliche Sehgrube (englisch Fovea), die für die Sehschärfe in der Mitte des Bildes verantwortlich ist, Veränderungen oder Fehler mit sinkender Gewichtung je mehr außen sie im Bild auftreten. Bei PSPNR (Peak-Signal-To-Perceptible-Noise-Ratio) werden nur Fehler oder Veränderungen gewertet, die oberhalb einer für den Menschen wahrnehmbaren Schwelle (JND) liegen [Lee Ebrahimi 2012]. Die Auswahl solcher veränderten Metriken kommt auch auf das zu bewertende Verfahren an.

Es existieren ebenfalls Metriken zur Bewertung zweier Videos, die nicht nur einzelne Pixel vergleichen, sondern ganze Strukturen. Der sogenannte „Structural Similarity Index“ (SSIM) vergleicht die drei verschiedenen Komponenten Helligkeit, Kontrast und Struktur, um zu bewerten wie gleich sich beide Videos sind. In natürlichen Videos sind viele Strukturen vorhanden, die sich durch Abhängigkeiten der Signale zeigen, insbesondere wenn diese räumlich im Bild aufeinander folgen. Zum Beispiel ist ein völlig unscharfes Bild in dieser Metrik qualitativ schlechter als ein Bild mit einzelnen Bildfehlern (z.B. Salt-And-Pepper-Noise), obwohl beide Bilder den gleichen MSE-Wert haben können. Ein höherer Wert beim

SSIM zeigt eine höhere strukturelle Gleichheit an, wobei die Werte zwischen 0 und 1 liegen [Wu Rao 2005, S.225ff]. Auch bei SSIM gibt es bei wahrnehmungsbasierten Verfahren je nach Verfahren unzuverlässige Bewertungen, z.B. dann wenn ein Hintergrund unschärfer sein darf, da der Mensch auf ihn nicht besonders achtet. Dieser würde mit SSIM aber schlechter bewertet.

Geeignet für die objektive Bewertung bei wahrnehmungsbasierter Videokompression ist somit die Verwendung von üblichen Verfahren wie PSNR oder SSIM, allerdings sind diese bei der Verwendung als Metrik für gesamte Bilder kritisch zu bewerten, je nachdem wie das Vorgehen des bewerteten Verfahrens ist. Gegebenenfalls macht es also Sinn manuellen Aufwand zu investieren und einzelne Bildbereiche einzeln zu bewerten und miteinander zu vergleichen und nicht das Gesamtbild. Zudem muss abgewogen werden, ob das eingesetzte Mess- und Bewertungsverfahren für das zu testende wahrnehmungsbasierte Verfahren sinnvoll ist und ob es die Verbesserungen bzw. Verschlechterungen, die dadurch entstehen, aufzeigen und bewerten kann. Alternativ wäre eine relativ zuverlässige Möglichkeit der Bewertung von Videoqualität die Durchführung einer subjektiven Qualitätsbewertung mit einer ausreichend großen Anzahl an unterschiedlichen Probanden, da hier die menschlichen Seheigenschaften automatisch in die Bewertung mit einfließen.

Für die objektive Qualitätsbewertung sind in der Zukunft jedoch noch bessere Verfahren notwendig, vor allem um wahrnehmungsbasierte Kompressionsverfahren zu verifizieren [Lee Ebrahimi 2012], um diese manuelle Bewertung einzelner Bildbereiche zu umgehen.

4 Verfahren wahrnehmungsbasierter Kompression

Das Ausnutzen einer oder mehrerer Eigenschaften der menschlichen visuellen Wahrnehmung aus Kapitel 2.4 kann durch unterschiedliche Verfahren umgesetzt werden. In den folgenden Unterkapiteln werden verschiedene Ansätze vorgestellt.

4.1 Ausnutzen von Detail- und Texturenwahrnehmung

Ein möglicher Ansatz für die Umsetzung von wahrnehmungsbasierter Videokompression ist das Ausnutzen der menschlichen Detail- und Texturenwahrnehmung. Wie in Kapitel 2.4 beschrieben, ist der Mensch auf Grund der Masking-Effekte nicht besonders gut in der Lage Texturen und insbesondere Details in diesen wahrzunehmen. Dies wird in Verfahren wie „A Parametric Framework for Video Compression Using Region-Based Texture Models“ [Zhang Bull 2011] ausgenutzt. Es ist eine Erweiterung des üblichen H.264-Verfahrens.

Das Ziel der Videokompression ist es nicht mit dem Resultatvideo möglichst nah an jeden einzelnen Pixel des Originalvideos heranzukommen und möglichst gleiche Frames zu den Originalframes zu liefern, sondern viel mehr eine gute subjektive Qualität zu erreichen. Die Idee des Verfahrens ist es daher, die Kodierung der Texturen zu vermeiden und diese später durch den Decoder erzeugen (im Folgenden auch synthetisieren genannt) zu lassen.

Um diese Idee umzusetzen werden folgende Schritte im Verfahren von [Zhang Bull 2011] durchgeführt: Zunächst wird festgestellt wo Bildbereiche sind, die aus Texturen bestehen. Es wird analysiert um was für Texturen es sich handelt und schließlich wird auf Grundlage einer Qualitätsbewertung entschieden, ob dieser Bereich bei der Kodierung übersprungen wird und die Textur im Decoder synthetisiert wird.

Der erste Schritt ist das Finden von texturierten Bildbereichen, was durch ein Segmentierungsverfahren für jeden Frame umgesetzt wird. Da Texturen häufig hochfrequente Bildbereiche sind, also viele aber regelmäßige Farb- und Helligkeitsänderungen enthalten sein können, ist eine Segmentierung mit Hilfe von lediglich gleichen Farbwerten nicht immer hilfreich. Daher wird ein Verfahren verwendet, das ein Bild auf Grund von seinen Pixelwerten mit der Wasserscheidentransformation, aber auch mit Hilfe von Clustering segmentiert.

Alle segmentierten Bildbereiche werden nun auf ihre Inhalte analysiert und je nach Art der Textur klassifiziert. Zunächst einmal wird unterschieden, ob es sich bei dem segmentierten Bereich um einen texturierten Bereich oder einen nicht-texturierten Bereich handelt. Typischerweise enthalten texturierte Bereiche mehr hochfrequente Bildanteile als nicht-texturierte, weshalb eine Aufteilung durch statistische Eigenschaften erfolgen kann.

Texturierte Bereiche werden weiter unterteilt in statische Texturen und dynamische Texturen. Statische Texturen sind sich nicht bewegende bzw. einfach bewegende Texturen, beispielsweise Holz oder Wände, wobei Bewegungen typischerweise nur durch Kamerabewegungen ausgelöst werden. Dynamische Texturen sind komplexere Texturen, die durch Bewegung über einen Zeitraum entstehen, z.B. fließendes Wasser oder sich bewegende Blätter an Bäumen. Um diese Klassifizierung durchzuführen, wird die texturierte Region in 8 x 8 Pixel große Blöcke aufgeteilt und eine Bewegungsanalyse mit den benachbarten Frames durchgeführt. Sind die daraus resultierenden Bewegungsvektoren regelmäßig und damit also „normal“ - die Bewegung kommt dann durch Kamerabewegungen und nicht durch Bewegungen der Textur selbst -, dann wird diese Textur als statisch, ansonsten als dynamisch klassifiziert.

Als Beispiel für eine solche Segmentierung und Klassifizierung dient eine Videosequenz, in der ein Boot auf einem Fluss fährt, dargestellt in Abbildung 5. Das linke Bild zeigt einen Frame dieses Videos nach der Segmentierung, wobei die weißen Linien anzeigen, in welche Bereiche das Bild aufgeteilt wurde. Das rechte Bild zeigt die Klassifizierung der segmentierten

Bildbereiche, wobei schwarz nicht-texturierte Bereiche symbolisiert, rot steht für statische Texturen und blau für dynamische. Das Wasser des Flusses wurde also als dynamisch erkannt, da es sich selbst in einer Fließbewegung bewegt. Das Gras und die Steine am Ufer wurden als statisch erkannt, da deren Bewegung lediglich durch Bewegung der Kamera zu Stande kam und das Boot als nicht-texturiert.

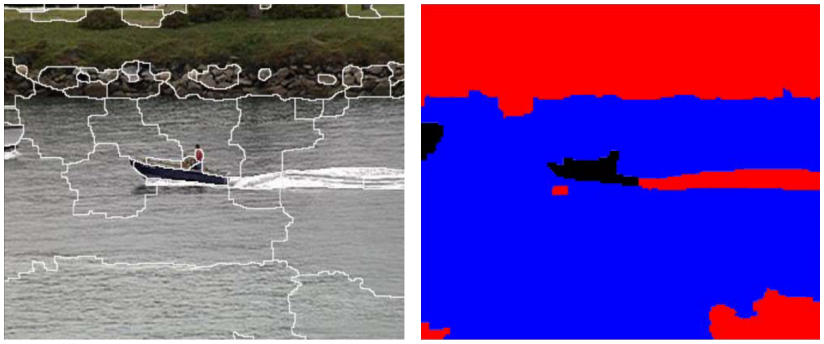

Abbildung 5: Segmentierung und Klassifizierung einer Szene [Zhang Bull 2011]

Alle nicht-texturierten Bereiche werden mit dem zu Grunde liegenden Verfahren - in diesem Fall H.264 - wie üblich kodiert. Bei statischen und dynamischen Texturen wird überprüft, ob eine Synthese im Decoder sinnvoll ist und dann wird der texturierte Bereich in Form seiner Blöcke beim Kodieren übersprungen (als SKIP markiert). Dort können also Bits deutlich eingespart werden, denn diese kommen erst im Decoder hinzu und müssen weder übertragen noch permanent gespeichert werden. Sinnvoll ist die Synthese, wenn die Qualität der Blöcke besser oder gleich gut im Vergleich zum üblichen H.264-Verfahren wäre und nicht mehr als die Bits benötigt würden, die das übliche Verfahren benötigt hätte.

Für die Beurteilung, ob die Qualität besser oder gleich gut ist, wird ein eigenes Messverfahren (AVM) verwendet. Dieses soll vor allem für den Menschen sichtbare Artefakte wie Unschärfe, Gleichheit oder Kanten erkennen, die alle durch die Synthese entstehen können, denn durch diese wird die später wahrgenommene Qualität schlechter. Das Verfahren soll allerdings nur angewendet werden, wenn die Qualität besser wird oder gleich bleibt. Für jeden Frame wird eine sogenannte „AVM Index Map“ erstellt. Wenn der durchschnittliche Wert für eine Bildregion in dieser AVM Index Map nach der Synthese der Texturen besser ist als die des Standardverfahrens, also einen bestimmten Schwellwert nicht unterschreitet, wird die Synthese verwendet.

Die Synthese der Texturen, die später im Decoder abläuft, aber im Encoder auch zur Überprüfung, ob sich eine Synthese lohnen würde, läuft für statische und dynamische Texturen unterschiedlich ab. Generell werden Texturen nicht in jedem Frame synthetisiert, sondern lediglich in sogenannten B-Frames (Frames, die mit benachbarten Frames vorhergesagt werden). Alle anderen Frames werden nicht verändert, sondern mit dem üblichen H.264-Verfahren kodiert und im Folgenden als Key-Frames bezeichnet.

Bei statischen Texturen wird eine Art Interpolation („bi-directional texture warping“) angewendet. Zunächst werden die Bewegungsinformationen mit den zwei nächsten Key-Frames, die zuvor der Klassifizierung der Textur dienten, wiederverwendet, um ein perspektivisches Bewegungsmodell, das auf acht Parametern basiert, zu bilden. Dieses Bewegungsmodell wird verwendet, um die Bewegung jedes Blocks, die zwischen den beiden als Referenz verwendeten Key-Frames stattfindet, zu beschreiben. Für die Synthese bzw. Rekonstruktion der Textur werden diese Parameter dann verwendet um die Textur aus den beiden Referenz-Key-Frames (von 5 Frames sind 2 Frames Key-Frames) so zu verzerren, dass eine ähnliche Textur entsteht wie im Originalframe. Da durch das Verzerren Unschärfe (englisch Blurring) entstehen kann,

wird anschließend ein Gauß'scher Deblurring Filter angewendet. Der Decoder erhält die üblichen Bilddaten und zusätzlich die Informationen in Form der Parameter, um die fehlenden (übersprungenen) Texturen mit „Texture Warping“ wieder zu erzeugen.

Für dynamische Texturen wird eine in „Dynamic Textures“ [Doretto et al. 2003] vorgestellte Synthesemethode verwendet, die in der Lage ist nach der Analyse von Texturen in einigen Testbildern fortlaufend dynamische Texturen zu generieren. Es wird in [Zhang Bull 2011] daher angenommen, dass dynamische Texturen für einen kurzen Zeitraum (von 5 Frames) gleichmäßig sind und für diesen Zeitraum wird dann mit dieser Synthesemethode eine dynamische Textur generiert. Nach diesen fünf Frames werden diese dann wiederum als Testbilder für die Methode verwendet, um erneut dynamische Texturen für folgende Frames zu generieren. Damit die dynamischen Texturen nicht zu sehr verfälscht werden, werden lediglich der 2. und 4. Frame der Gruppe von insgesamt 5 Frames mit dieser Methode synthetisiert. Die übrigen werden wie bei statischen Texturen verzerrt und dienen wiederum als Eingabe für die Methode.

Das Ergebnis des Verfahrens zeigt Abbildung 6. Das linke Bild ist ein Ausschnitt aus dem Originalvideo aus Abbildung 5, das rechte Bild die synthetisierte Textur. Es ist deutlich zu erkennen, dass beide Texturen sich sehr ähnlich sind und Unterschiede mit dem bloßen Auge nur gering wahrzunehmen sind. Wichtige Artefakte aus dem Originalbild wie z.B. der hellere Streifen mittig am oberen Bildrand sind auch in der synthetisierten Textur vorhanden. Ein Differenzbild der beiden oberen Bilder zeigt, dass Unterschiede vorhanden sind, allerdings sind diese mit dem menschlichen Auge im späteren Video auf Grund der angesprochenen schlechten Texturwahrnehmung und der Masking-Effekte kaum zu erkennen.

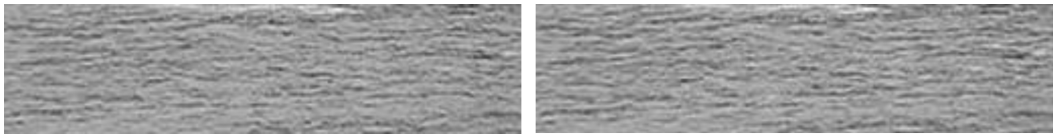

Abbildung 6: Ergebnis der Textursynthese [Zhang Bull 2011]
(links: Originaltextur, rechts: synthetisierte Textur)

[Zhang Bull 2011] haben das Verfahren mit der H.264/AVC-Referenzsoftware durch eine objektive und eine subjektive Qualitätsbewertung mit 22 Testpersonen verglichen. Es konnte bei gleich wahrgenommener Qualität eine Biteinsparung von maximal bis zu 60%, bei einigen anderen Videos zwischen 30% und 50% erreicht werden. Die Effizienz des Verfahrens ist stark davon abhängig wie viele Anteile eines Videos aus Texturen bestehen. Allerdings ist das Verfahren auch deutlich komplexer und rechenaufwendiger als das übliche H.264-Verfahren (ca. 3-4 mal so komplex), da viele Berechnungen, wie die Qualitätsbewertung oder die Berechnung der Parameter für die Synthese, für jeden einzelnen Frame durchgeführt werden müssen. Dadurch kommt ein deutlich höherer Zeitaufwand beim Kodieren von Videos zu Stande. Zudem wird auch ein veränderter Decoder benötigt, der diese Synthese umsetzt.

4.2 Ausnutzen der differentiellen Wahrnehmbarkeitsschwelle (JND)

Eine ähnliche Eigenschaft wie die schlechtere Wahrnehmung von Details in Texturen in Kapitel 4.1 ist die differentielle Wahrnehmbarkeitsschwelle (JND). Der Mensch kann beispielsweise Kontraste unterhalb eines bestimmten Schwellwertes nicht mehr wahrnehmen und Werte unter dieser Schwelle unterscheiden. Dies wird zum Beispiel in dem Verfahren „An HEVC-Compliant Perceptual Video Coding Scheme Based on JND Models for Variable Block-Sized Transform Kernels“ [Kim Bae Kim 2015] ausgenutzt.

Das Verfahren basiert auf der HEVC-Referenzsoftware HM11.0 und erweitert diese so, dass ein HEVC-konformes Ergebnis entsteht. Die Werte, die nach der Bildvorhersage (Intra- oder Inter-Prediction) auf Grundlage der Transformationsblöcken (TU) weiter transformiert werden sollen, werden durch Unterdrückung der Transformationskoeffizienten zunächst verändert (je höher der JND-Wert ist desto geringer wird der Transformationskoeffizient), bevor sie anschließend transformiert und quantisiert werden. In HEVC gibt es einen sogenannten „Transform Skip Mode“ (TSM), der jedoch nur bei 4 x 4 Pixel großen Blöcken möglich ist und bewirken soll, dass für den Block in diesem Modus keine Transformation durchgeführt wird. Dies sorgt bei bestimmten Bildinhalten wie zum Beispiel computergenerierten Bildern oder gemischten Bildern aus Grafiken und Kamerabild für eine bessere Kodierungseffizienz [Kim Bae Kim 2015].

Bei diesem Modus werden die Koeffizienten des Transformationsblocks also nur skaliert und anschließend quantisiert, die Transformation (mit der diskreten Kosinustransformation) entfällt. Alle TUs, die diesen Modus nicht verwenden, werden wie üblich transformiert und quantisiert. Die Veränderung durch das Verfahren findet bei der Transformation statt, weshalb in [Kim Bae Kim 2015] eine Fallunterscheidung für Blöcke, die diesen Modus (TSM) verwenden, und Blöcke, die diesen Modus nicht verwenden (non-TSM), durchgeführt wird.

Für die Nicht-TSM-Blöcke werden verschiedene JND-Eigenschaften ausgenutzt und zusammengefügt. Die erste Eigenschaft ist die Kontrastempfindlichkeit. Der Mensch nimmt unterschiedliche räumliche Frequenzen unterschiedlich wahr, denn Kontraste im mittleren Frequenzbereich können besser wahrgenommen werden als im niedrigen oder hohen. Als Grundlage für das Verfahren dient die Funktion $H_{CSF}(\omega_{i,j}, \varphi_{i,j})$ (Basic Contrast Sensitivity Function (CSF)) mit den Eingabewerten der räumlichen Frequenz $\omega_{i,j}$ und dem ausgerichteten Winkel des Transformationskoeffizienten $\varphi_{i,j}$. Ihr Ergebniswert ist der JND-Wert, der später für die Veränderung der Transformationskoeffizienten verwendet wird. Die Ergebnisse der Funktion werden zusätzlich abgeändert durch drei weitere Funktionen.

Eine dieser Funktionen beschreibt das „Luminance Masking“ (LM), also die Maskierung von Helligkeitswerten. Hier wird ausgenutzt, dass der Mensch vor allem in mittleren Helligkeitsbereichen zwischen Werten besser differenzieren kann als in dunklen oder hellen. Um diese Eigenschaft auszunutzen, wird eine Funktion $MF_{LM}(\mu_p)$ formuliert, die auf die durchschnittlichen Helligkeitswerte μ_p der TU-Blöcke angewendet wird. Die Formel besteht im Wesentlichen aus einer Fallentscheidung, ob der Helligkeitswert kleiner als ein Parameter ist, zwischen zwei Parametern liegt oder größer als ein anderer Parameter ist, woraus der neue Wert resultiert. Die Parameter wurden in Experimenten zuvor so festgelegt, dass durch diese Werte kein wahrgenommener Unterschied entsteht. Für die durchschnittlichen Helligkeitswerte im Eingangsbild zwischen 0 und 82 sinkt der Ergebniswert von 3 auf 1 kontinuierlich ab, im Bereich zwischen 82 und 95 bleibt er konstant bei 1 und für alle Werte größer als 95 steigt er wieder von 1 auf 3 an. Die Werte im Bereich zwischen 82 und 95 werden also durch diesen JND-Wert nicht beeinflusst, während die Werte im dunklen und hellen Bereich etwas vergrößert werden.

Des Weiteren ist der Mensch empfindlicher gegenüber Intensitätsänderungen bei langsameren Bewegungen als bei schnelleren Bewegungen. Hier wird eine Funktion $MF_{TM}(\omega_{i,j}, mv)$ für das sogenannte „Temporal Masking“ (TM) verwendet. Als Eingabeparameter für diese Funktion werden die räumliche Frequenz $\omega_{i,j}$ und der Bewegungsvektor mv benötigt. Je nach Frequenz, also je nach Geschwindigkeit der Änderungen, wird der Resultatwert berechnet. Durch die Funktion erhalten Blöcke mit niedrigen räumlichen und zeitlichen Frequenzen den Faktor 1 (also keine Beeinflussung), ansonsten werden die Werte je nach Frequenzen höher.

Die letzte verwendete Funktion ist die Funktion $MF_{CM}(\omega_{i,j}, \tau)$ für die sogenannte Kontrastmaskierung. τ steht hier für die Kantendichte des Blocks, die durch die Anwendung eines Sobel-Kanten-Operators bestimmt wird. Wie bereits im Verfahren in Kapitel 4.1 ausgenutzt,

sieht der Mensch Verschlechterungen in glatten Bildbereichen besser als in texturierten bzw. hochfrequenten Bereichen. Es wird in der Funktion daher unterschieden ob die räumliche Frequenz einen bestimmten Wert überschreitet oder nicht. Je nachdem wird dann ein JND-Wert für den Block berechnet, der in die Gesamtformel einfließt. Bei einer räumlichen Frequenz von 4,17 Zyklen pro Minute resultiert der höchste JND-Wert, je weiter die Frequenz von diesem entfernt ist, desto kleiner wird er.

Die oben erwähnten Funktionen für die einzelnen Eigenschaften werden nun durch Multiplikation miteinander und mit einer weiteren Funktion („Summation Effect Function“) je nach Blockgröße zu einer gemeinsamen Formel zusammengesetzt. Sie ergeben so die Formel JND_{nonTSM}, die für jeden Transformationsblock, der keinen Skip-Modus verwendet, berechnet wird. Die Grundfunktion ist also die Funktion für die Kontrastempfindlichkeit, die durch die weiteren Funktionen verändert wird. Die Gesamtformel wird dann in das übliche HEVC-Verfahren in den Transformationsprozess integriert und berechnet für jeden Transformationskoeffizienten einen JND-Wert, der diesen verändert/unterdrückt.

Für die TSM-Blöcke wird ein bereits existierendes Verfahren auf Pixelebene verwendet. Hier wird lediglich eine Funktion $JND_{TSM}(\mu_p)$ für die Helligkeitsmaskierung (LM) verwendet. Diese sorgt dafür, dass bei Blöcken mit durchschnittlich niedrigeren Werten (kleiner als 127) einen Ergebniswert zwischen 20 und 3 erhalten, Blöcke mit höheren Werten (größer als 127) erhalten einen Wert zwischen 3 und 6. Je weiter die Werte von 127 entfernt sind, desto höher ist der Ergebniswert, wodurch die äußeren Bereiche (hell und insbesondere dunkel) insgesamt für einen höheren Wert sorgen.

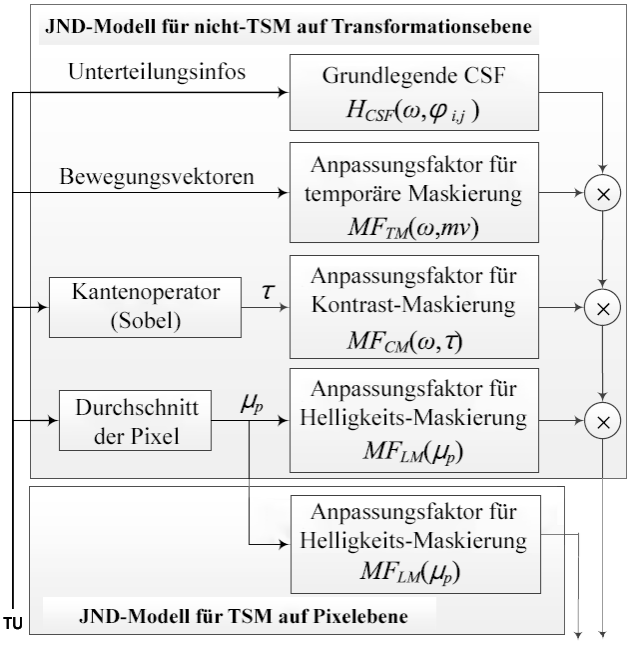

Abbildung 7: Aufbau des JND-Verfahrens (in Anlehnung an [Kim Bae Kim 2015])

Abbildung 7 zeigt den Aufbau des Verfahrens und die Verschiedenen Funktionen in einer grafischen Übersicht. Als Dateneingabe dienen die TUs, die nach TSM und nicht-TSM unterschieden werden. Bei nicht-TSM-Blöcken dient die Funktion für die Kontrastempfindlichkeit als Grundfunktion, die durch die drei Funktionen darunter verändert wird, bei TSM-Blöcken wird nur die Helligkeitsmaskierung angewendet. Insgesamt gilt: Je höher der endgültige JND-Wert durch die einzelnen Faktoren wird, desto geringer wird der Transformationskoeffizient bzw. desto mehr wird er unterdrückt, wodurch sich die Werte des Blocks später bei Transformation und Quantisierung aneinander angleichen.

Das Verfahren wurde durch [Kim Bae Kim 2015] sowohl objektiv in Bitratenreduktion und Kodierungszeit als auch subjektiv mit 15 Testpersonen bewertet. Bei ungefähr gleich bleibender bzw. gleich wahrgenommener Videoqualität konnte das Verfahren demnach eine maximale Bitratenreduktion von 49,10% liefern, wobei diese je nach Inhalt der Testvideos schwankte. Durchschnittlich konnte eine Bitratenreduktion von ca. 16,10% im Vergleich zum üblichen HEVC-Verfahren (im Form der Referenzsoftware HM11.0.) erreicht werden. Allerdings ist das veränderte Verfahren durch die erhöhten Berechnungen für jeden TU-Block insgesamt ca. 11,25% langsamer in der Kodierungszeit.

4.3 Ausnutzen selektiver Aufmerksamkeit

Auch die selektive Aufmerksamkeit des Menschen kann in wahrnehmungsbasierten Videokompressionsverfahren ausgenutzt werden. Beispiele für solche Verfahrens sind „A New Saliency Based Video Coding Method with HEVC“ [Zhengrong et al. 2015] und „Saliency Based Perceptual HEVC“ [Li et al. 2014]. Beide sind keine vollständig eigenständigen Verfahren, sondern bauen auf dem aktuellen Standard H.265/HEVC in Form seiner Referenzsoftware HM12.0 bzw. HM11.0 auf und erweitern diese. Die üblichen Vorgehensweisen von HEVC wie die Unterteilung aller Einzelbilder in Blöcke oder auch die Bewegungsvorhersagen sind also nach wie vor vorhanden.

Bei solchen Ansätzen möchte man ausnutzen, dass der Mensch zu viele visuelle Informationen bekommt und deshalb priorisieren muss, auf welche Objekte oder Bereiche eines Bildes er genauer achten möchte und auf welche nicht. Bereiche eines Bildes, auf die der Mensch besonders achtet, werden als „auffällig“ oder „ins Auge springend“ (englisch salient) bezeichnet.

Die Grundidee bei diesen Verfahren ist, dass Bereiche, auf die der Mensch ohnehin auf Grund seiner Wahrnehmung nur sehr wenig oder gar nicht achtet, mit einer schlechteren Videoqualität kodiert werden können. In Bereichen, auf die der Mensch besonders achtet (auffällige Bereiche) sollte die Videoqualität möglichst hoch sein. So wird das Bild bzw. die Videosequenz insgesamt vom Zuschauer als qualitativ hochwertiger und besser wahrgenommen. Nach [Zhengrong et al. 2015] hängt die Qualität des gesamten Videobildes also von den auffälligen Bereichen im Video ab. Würden diese in einer höheren Qualität kodiert, hätte das eine Verbesserung der Qualität für das ganze Video zur Folge, während in den nicht auffälligen Bereichen Videodaten in Form von Bits eingespart werden könnten.

Den Schlüssel zur Umsetzung dieser gewünschten ungleich verteilten Qualität liefert die Quantisierung im üblichen HEVC-Algorithmus. Diese ist verlustbehaftet und mit Hilfe des Quantisierungsparameters (QP) kann festgelegt werden wie stark die Quantisierung durchgeführt werden soll, woraus resultiert wie stark der Qualitätsverlust im Video später ist. In H.265/HEVC werden üblicherweise alle Blöcke eines Bildes mit dem gleichen Quantisierungsparameter quantisiert, was jedoch mit den Prinzipien der menschlichen Wahrnehmung nicht übereinstimmt [Zhengrong et al. 2015].

Das Verfahren kann in drei Schritte aufgeteilt werden. Zunächst einmal wird eine sogenannte „Saliency-Map“ erstellt, die feststellt und zeigt wo im Bild Bereiche sind, die besonders herausstechen und auf die der Zuschauer besonders achtet. Eine solche Saliency-Map besteht aus Werten im Bereich zwischen 0 und 255 für jeden Pixel des ursprünglichen Bildes und kann daher relativ einfach als Graustufenbild visualisiert werden. Für jedes Einzelbild eines Videos wird eine Saliency-Map erstellt. Je höher der Wert eines Pixels in der Saliency-Map ist, desto wichtiger ist auch der Bereich, den er abbildet und desto höher ist auch die Aufmerksamkeit, die der Zuschauer diesem Bereich schenkt.

Es wirken drei verschiedene Faktoren darauf ein, welche Bereiche für den Menschen wich-

tig erscheinen. Einer dieser Punkte ist Bewegung, denn durch sie wird die menschliche Aufmerksamkeit auf ein Objekt gezogen. Zunächst wird also auf Grund der Bewegung in einer Videosequenz eine Saliency-Map generiert. Dafür wird ein Algorithmus zur Erkennung von sogenannter Motion Saliency (durch Bewegung ausgelöste Aufmerksamkeit) von [Ren Chia Rajan 2012] verwendet.

Um auffällige Objekte oder Bildbereiche zu erkennen, die durch Bewegung auffällig werden, reicht es demnach nicht aus, eine einfache Differenz zwischen einzelnen Frames zu bilden, da sich nicht nur die einzelnen Objekte bewegen. Man kann nicht davon ausgehen, dass die Kamera der Szene statisch ist, sondern muss davon ausgehen, dass diese sich ebenfalls bewegt und mit ihr auch der Hintergrund bzw. die gesamte Szene. Die Ursache für Bewegungen des Hintergrundes ist allerdings in den meisten Fällen die Bewegung der Kamera, weshalb man bei der Erkennung der auffälligen Bereiche in [Ren Chia Rajan 2012] diese Bewegungen als 2D-Transformationen wie Skalierung, Rotation und Translation modelliert. Die transformierten Einzelbilder werden dann aneinander angepasst und die Objekte, die sich in eine andere Richtung oder mehr bewegen als der Rest in der Szene sind auffällig, also Bereiche, denen der Mensch besondere Aufmerksamkeit schenkt.

Zusätzlich werden in [Ren Chia Rajan 2012] auch räumliche Auffälligkeiten bei der Erstellung der Saliency-Map zusätzlich berücksichtigt, z.B. durch seltenere Farbwerte, die mit Hilfe von Histogrammen bestimmt werden können. Größere Farbunterschiede und seltene Farben bekommen einen höheren Wert in der Saliency-Map zugewiesen.

Durch eine Kombination der Saliency-Map mit auffälligen Bewegungen und der Saliency-Map mit auffälligen Farben wird eine erste Saliency-Map, die auffällige Bewegungen beinhaltet, für das Verfahren von [Zhengrong et al. 2015] erstellt.

Des Weiteren bekommen in [Zhengrong et al. 2015] Objekte Aufmerksamkeit, die sich von ihrer Umgebung durch andere Farben unterscheiden. Daher wird mit Hilfe von Farbinformationen eine weitere Saliency-Map erstellt. Je höher der Farbkontrast eines Pixels zu anderen Pixeln, desto höher ist sein Wert in der Saliency-Map, Pixel mit gleicher Farbe erhalten den selben Wert in der Saliency-Map. Neben Farben wird Aufmerksamkeit auch durch verschiedene Texturen erregt. Daher werden auch diese in einer eigenen Saliency-Map zusammengetragen. Insgesamt entstehen so also drei unterschiedliche Saliency-Maps für ein Einzelbild bzw. eine Videosequenz.

Als nächstes werden die zuvor erstellten Saliency-Maps normalisiert und zu einer gesamten Saliency-Map zusammengestellt. Dafür werden die drei Saliency-Maps mit unterschiedlichen Gewichtungen zu einer finalen Saliency-Map addiert. Da der Mensch am empfindlichsten gegenüber der Bewegung ist und diese die meiste Aufmerksamkeit auf sich zieht, wird die entsprechende Saliency-Map für die Bewegung mit 40% am stärksten gewichtet, die anderen beiden erhalten eine Gewichtung von 30%.

Der letzte Schritt ist die Nachverarbeitung. Unter Berücksichtigung des Aspekts, dass der Mensch beim Schauen eines Videos grundsätzlich in den mittleren Bereich eines Videos schaut, werden die Werte der äußersten Bildbereiche der Saliency-Map auf 0 gesetzt. Diese Vorgehensweise spart weitere Bits ein. Ein Beispiel für eine als Graustufenbild visualisierte Saliency-Map aus [Zhengrong et al. 2015] zeigt Abbildung 8.

Das linke Bild in Abbildung 8 zeigt eine Basketball-Szene, in der ein Basketballspieler im unteren mittleren Bildbereich einen Basketball auf einen Korb in der linken oberen Ecke des Bildes wirft. Außerdem laufen in der Mitte und im rechten oberen Bereich des Bildes zwei weitere Basketballspieler. Das rechte Bild zeigt die aus diesem Frame resultierende Saliency-Map. Sowohl auf Grund der Bewegung der Spieler und des Balls, aber auch wegen der hohen farblichen und kontrastbezogenen Unterschiede zwischen Spielern bzw. Ball und dem relativ einfarbigen Boden der Sporthalle, erhalten die Spieler und der Ball höhere Werte in der Saliency-Map. Die hell gekennzeichneten Bereiche sind also die Bereiche, auf die der

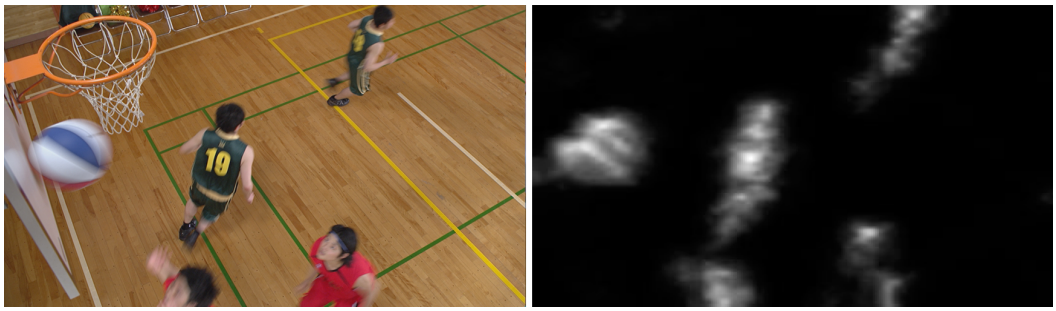

Abbildung 8: Resultierende Saliency-Map aus einer Szene [Zhengrong et al. 2015]

Zuschauer besonders achtet und diese sind daher möglichst mit hoher Qualität zu kodieren, während ein Qualitätsverlust in den übrigen Bereichen - vor allem am Boden - nicht besonders auffallen wird.

Die erstellte Saliency-Map kann nun verwendet werden, um die Informationen über die menschliche selektive Aufmerksamkeit in ihr im aktuellen Frame bzw. in der aktuellen Videosequenz für die Videokompression umzusetzen. Dies geschieht mit dem vorher angesprochenen Quantisierungsparameter, der nicht wie üblich für jeden Block gleich ist, sondern je nach Wert in der Saliency-Map angepasst wird. Bei der Quantisierung steht ein niedriger Wert für einen hohen Qualitäts- bzw. Detailgrad im späteren Video, ein hoher Wert für einen hohen Qualitätsverlust. Möglich sind Werte in einem Wertebereich von 0 bis 51. Blöcke mit hohen Werten in der Saliency-Map bekommen also einen geringen Wert als Quantisierungsparameter und Blöcke mit niedrigen Werten in der Saliency-Map einen hohen Wert als Quantisierungsparameter. In den in der Saliency-Map nicht so hoch gewichteten Bereichen des Videos werden also Bits eingespart, während in den hoch gewichteten Bits zusätzlich zur Verfügung gestellt werden [Zhengrong et al. 2015].

Auch der zweite Ansatz von [Li et al. 2014] geht ähnlich vor und erstellt zunächst eine Saliency-Map, die dann für die Quantisierung verwendet wird.

Der Algorithmus für die Anpassung der Quantisierung geht dabei wie folgt vor. Es werden für jedes Einzelbild einer Videosequenz alle vom klassischen HEVC-Algorithmus aufgeteilten Blöcke (Coding Units) durchlaufen. Für jeden Block wird nun ein Durchschnittswert aus den entsprechenden Pixeln in der Saliency-Map gebildet und ein Durchschnittswert der Pixel in der Saliency-Map für das gesamte Bild gebildet.

Es folgt die Berechnung des neuen Quantisierungswertes für die aktuelle Coding Unit nach folgender Formel aus [Li et al. 2014]:

$$QP(i,j) = \begin{cases} QP_0 + m, & y(i,j) \leqslant \alpha \\ Int[QP_0 + m + \frac{m}{\alpha - \beta} \times (y(i,j) - \alpha)], & \alpha < y(i,j) \leqslant \beta \\ QP_0, & y(i,j) > \beta \end{cases}$$

Demnach wird unterschieden ob der Wert für den aktuellen Block $y(i,j)$ kleiner oder gleich eines Schwellwertes α, größer als ein Schwellwert β ist oder ob er zwischen diesen beiden Werten liegt. Die Schwellwerte α und β setzen sich aus vorher festgelegten Werten und dem Durchschnittswert des gesamten Bildes zusammen. Ist der aktuelle Wert kleiner als α, wird auf den ursprünglichen Quantisierungsparameter ein festgelegter maximaler Modifikationswert m[1] addiert, denn ein kleinerer Wert in der Saliency-Map als der durchschnittliche Wert bedeutet, der aktuelle Block ist unwichtiger als der Durchschnitt. Hier wird der Quantisierungsparameter erhöht, die Qualität also schlechter.

[1] Die Bedeutung von m weicht hier von der Bedeutung in [Li et al. 2014] ab, da diese nach Rücksprache mit dem Autor fehlerhaft war, der ursprüngliche Parameter n wird nicht benötigt

Ist der aktuelle Wert größer als β, bedeutet dies, dass der aktuelle Block wichtiger ist als der Durchschnitt der Blöcke. In diesem Fall bleibt der Quantisierungsparameter für den Block wie er initial war. Die beste Qualität bzw. der niedrigste Quantisierungsparameter ist der, der initial für das gesamte Video festgelegt wurde. Dieser bekommt nun gewissermaßen die Bedeutung minimal möglicher Quantisierungsparameter bzw. größtmögliche Qualität.

Befindet sich der Wert des aktuellen Blocks zwischen beiden Schwellwerten, wird je nach genauem Wert aus der Saliency-Map eine größere oder kleinere Anzahl auf den Quantisierungsparameter addiert. Dieser liegt dann zwischen maximal möglichem Quantisierungsparameter (Summe von initialem QP und maximalem Modifikations-Wert) und initialem Quantisierungsparameter.

Abschließend wird der Quantisierungsparameter für den aktuellen Block, sofern er größer als 51 ist, auf 51 gesetzt, was der maximal mögliche Wert der Quantisierungsparameter ist und analog dazu auf 0, falls der Quantisierungsparameter kleiner als 0 hätte sein sollen.

Das Endresultat des Verfahrens aus [Zhengrong et al. 2015] ist in Abbildung 9 dargestellt. Das linke Bild ist ein Einzelbild aus der ausgewählten Basketballszene kodiert mit der HEVC-Referenzsoftware und das rechte Bild wurde mit dem Saliency-Map-Verfahren kodiert. Es sind beim Blick auf die Basketballspieler und den Basketball optisch keine Unterschiede feststellbar, allerdings erkennt man z.B. beim Boden im rechten Bild einen leichten Farbunterschied oberhalb des Basketballs, da hier minimal hellere Streifen der Textur verloren gegangen sind.

Abbildung 9: Resultat des Saliency-Map-Verfahrens [Zhengrong et al. 2015]
(links: Originalvideo, rechts: nach Saliency-Map-Verfahren)

Zum Bewerten des Verfahrens wurde in [Li et al. 2014] eine subjektive Qualitätsbewertung mit verschiedenen Videos in unterschiedlichen Auflösungen und mit unterschiedlichen initialen Quantisierungsparametern im Vergleich zur Referenzsoftware ohne diese Erweiterungen verwendet. Demnach ist es möglich mit dem Verfahren eine Reduktion der Bits von bis zu 12% zu erreichen, wobei die von den Testpersonen wahrgenommene Qualität im Vergleich zum Originalvideo gleich ist.

Generell führte das Verfahren bei keinem der Testvideos zu einer schlechter wahrgenommenen Videoqualität. Damit ist das Ziel des Verfahrens, die Größe des Videos zu reduzieren, aber die wahrgenommene Videoqualität zu erhalten, erreicht. Auf Grund der automatischen selektiven Aufmerksamkeit des Menschen ist die Anwendung dieses Verfahrens auch in einer vermutlich sehr großen Zahl von Videos möglich und sinnvoll.

4.4 Ausnutzen von Aufmerksamkeit durch Gesichtserkennung

Eine weitere Vorgehensweise, die sich die selektive Aufmerksamkeit zu Nutzen macht, ist die besondere Behandlung von Gesichtern in Videosequenzen. Gesichter lenken, wie in Kapitel 2.4 beschrieben, die Aufmerksamkeit des Menschen besonders stark auf sich. Verschiede-

ne Verfahren versuchen daher Gesichter in Videosequenzen zu erkennen und besser als die übrigen Bildbereiche zu kodieren.

Ein Beispiel für ein solches Verfahren ist „Region-of-Interest Based Conversational HEVC Coding with Hierarchical Perception Model of Face“ [Xu et al. 2014]. Das Verfahren baut auf der HEVC-Referenzsoftware HM9.0 auf und ist kein eigenständiges Verfahren. Daher bestehen auch hier weiterhin zusätzlich die Vorteile der konventionellen HEVC-Kodierung.

Die Idee bei dem Verfahren von [Xu et al. 2014] ist ähnlich wie bei dem Verfahren zum Ausnutzen der selektiven Aufmerksamkeit aus Kapitel 4.3. Die Bereiche, auf die der Zuschauer besonders achtet, sind menschliche Gesichter, sofern welche im Bild vorhanden sind. Diese sollen in besserer Qualität kodiert werden und die Bildbereiche, auf die nur wenig oder gar nicht geachtet wird, sollen in schlechterer Qualität kodiert werden. So sollen sowohl Bits als auch Zeit beim Kodieren eingespart werden. Es bietet sich an, ein solches Verfahren vor allem bei dialogorientierten Videos wie Videokonferenzen, Interviews oder Nachrichten zu verwenden, da dort Gesichter vermehrt enthalten sind und gut extrahiert werden können.

Grundsätzlich besteht das Gesamtverfahren aus drei Teilen: der Gesichtserkennung, einer Unterteilung und Gewichtung der Szene und der Anpassung des Quantisierungsparameters. Zunächst muss erkannt werden, wo im zu kodierenden Video Gesichter vorhanden sind. Dafür wird das Gesichtserkennungsverfahren aus „Face Alignment through Subspace Constrained Mean-Shifts“ [Saragih Lucey Cohn 2009] verwendet. Mit diesem Verfahren können sowohl die Umrisse der Gesichter, als auch die Gesichtszüge als eine Reihe von Punkten erkannt werden. Diese werden miteinander zu Umrissen verbunden und dienen als Grundlage für die im nächsten Schritt folgende Unterteilung der Szene.

Die Szene wird, wie in Abbildung 10 dargestellt, hierarchisch unterteilt, wobei jeder Knoten eine eigene Gewichtung erhält. Je wichtiger ein Knoten ist, desto höher ist seine Gewichtung.

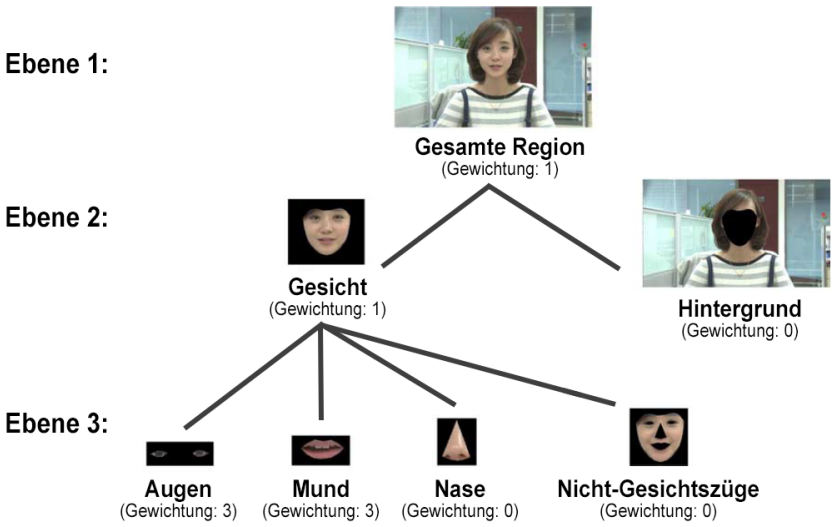

Abbildung 10: Hierarchische Unterteilung der Szene (in Anlehnung an [Xu et al. 2014])

Den ersten Knoten bildet die gesamte Szene, die wiederum unterteilt wird in Gesicht und Hintergrund, wobei der Hintergrund in diesem Fall alles ist, was nicht Gesicht ist. Da das Gesicht wichtiger ist als der Hintergrund und der Mensch auf dieses stärker achtet, bekommt das Gesicht eine Gewichtung von 1, der Hintergrund eine Gewichtung von 0. Das Gesicht wird erneut unterteilt in Augen, Mund, Nase und Nicht-Gesichtszüge. Da der Mensch besonders auf die Augen und den Mund achtet, werden diese jeweils mit 3 gewichtet, Nase und Nicht-Gesichtszüge mit 0. Die Addition der Gewichtungen von unten nach oben ergeben die pixelweise Gewichtung, die in einer sogenannten Weight-Map, ähnlich der Saliency-Map

aus Kapitel 4.3, zusammengefasst werden. Um diese Weight-Map noch zu verbessern und harte und damit möglicherweise sichtbare Kanten an den Rändern der Augen, dem Mund und dem gesamten Gesicht zu vermeiden, wird zusätzlich anschließend ein Gaußfilter angewendet, der die Kanten abflacht.

Die erstellte Weight-Map wird für die Anpassung zweier Bereiche des konventionellen HEVC verwendet: Die Blockunterteilung und die Quantisierung. In HEVC wird das Bild in Blöcke zwischen 32 und 4 Pixeln unterteilt (genauer in Kapitel 2.3.1) und zwar je nach Inhalt des Bildes. Ob eine weitere Unterteilung sinnvoll ist, wird durch den Aufwand für die Bestimmung der Datenrate bei gegebenem Qualitätsverlust (englisch rate-distortion cost) bestimmt. Ist diese für den aktuellen Block höher als die Summe der vier nach einer Unterteilung erhaltenen Blöcke, wird der Block bis zu drei mal unterteilt. Durch diese Berechnungen wird allerdings die Rechenzeit bei der Blockaufteilung deutlich höher.

Optional kann in HEVC aber auch eine maximale Unterteilungstiefe angegeben werden, die nicht überschritten wird. Detaillierte Blockunterteilungen sind in den Bildbereichen, die ohnehin durch den Menschen nicht viel Aufmerksamkeit bekommen, nicht notwendig, weshalb im Verfahren von [Xu et al. 2014] diese maximale Tiefe für den Hintergrund niedriger gesetzt wird, um die Rechenzeit zu reduzieren. Je niedriger der Durchschnittswert eines Blocks in der Weight-Map ist, desto niedriger ist auch die maximal mögliche Tiefe für diesen Block. Welchen Wert die maximale Tiefe erhält, wird mit zwei Schwellwerten geregelt. Sinnvoll eingestellt haben die Bereiche der Augen und des Mundes also die detaillierteste Unterteilung, gefolgt vom übrigen Gesicht und die geringste Unterteilung gibt es beim Hintergrund.

Auch bei der Quantisierung wird die Weight-Map als Grundlage verwendet. Auch hier werden wie bei dem Verfahren in Kapitel 4.3 und anders als beim konventionellen HEVC-Verfahren für jeden Block unterschiedliche Quantisierungsparameter verwendet. Die Weight-Map eines Frames des Beispielvideos zeigt die linke Grafik in Abbildung 11, die Quantisierungsparameter sind in der rechten Grafik dargestellt. Dabei ist zu beachten, dass in der rechten Grafik jeder Block 64 x 64 Pixel groß ist und bei weiterer Unterteilung der durchschnittliche Quantisierungsparameter dargestellt ist.

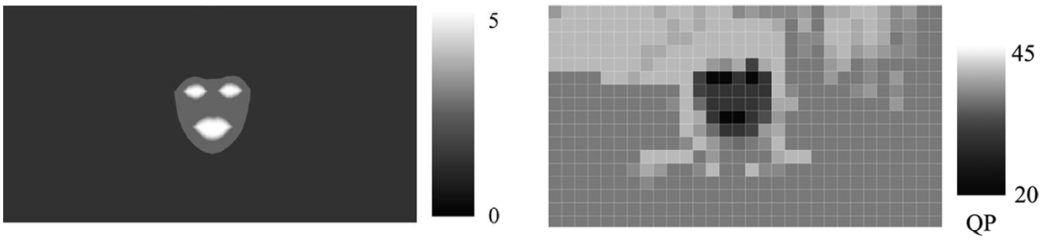

Abbildung 11: Weight-Map und Quantisierungsparameter [Xu et al. 2014]

Das Gesicht bekommt demnach die geringsten Quantisierungsparameter, insbesondere Augen und Mund, wodurch hier die meisten Bits verwendet werden, die Blöcke des Hintergrunds erhalten hohe Werte und erhalten dadurch nur wenige Bits und damit eine geringere Qualität.

Das gesamte Verfahren wurde bei mehreren Gesichter enthaltenden Videosequenzen angewendet und sowohl objektiv mit PSNR-Werten (für bestimmte Bildbereiche getrennt bewertet), als auch subjektiv mit zwölf Testpersonen bewertet. Durch die veränderte Unterteilung der Blöcke kann mit dem Verfahren eine zeitliche Einsparung der Rechenzeit von bis zu 23,8% bei CIF-Auflösungen und bis zu 62,8% bei HD-Auflösungen erreicht werden. Das Verfahren übertrifft das klassische HEVC-Verfahren bei der Bewertung von Gesichtern, denn hier waren die durchschnittlichen PSNR-Werte deutlich verbessert, im Hintergrund dafür eher schlechter. Auch das Resultat der subjektiven Bewertung besagt, dass die von den Testpersonen wahrgenommene Qualität der Videos mit dem Gesichtserkennungsverfahren besser

ist als das klassische HEVC-Verfahren. Ein Beispiel für ein Endergebnis des Verfahrens zeigt Abbildung 12.

Abbildung 12: Resultat des Gesichtserkennungsverfahrens [Xu et al. 2014]
(links: Übliches HEVC-Verfahren, rechts: Gesichtserkennungsverfahren)

Die linken drei Bilder zeigen ein Gesicht aus einem Beispielvideo, das mit dem üblichen HEVC-Verfahren kodiert wurde. Vor allem im Mund, aber auch an den Augen erkennt man hellere Verfärbungen, die durch die Kodierung aufgetreten sind. Die rechten drei Bilder zeigen die selbe Szene kodiert mit dem Gesichtserkennungsverfahren. Hier fällt vor allem auf, dass die Pixel im Mund deutlich weniger Verfärbungen aufweisen. Die übrige Haut im Gesicht sowie die Haare sind von der Qualität her ungefähr gleich, hier fallen keine großen Unterschiede auf. Der Hintergrund der Szene ist auf diesen beiden Bildern nicht vorhanden, daher ist hier keine Aussage möglich, wie sich dieser nach der Anwendung des Verfahrens verändert hat.

Insgesamt konnte durch das Verfahren also die Kodierungszeit und die wahrgenommene Qualität der Videos verbessert werden. Der Einsatzbereich des Verfahrens ist allerdings auf Videos, die Gesichter enthalten, beschränkt bzw. nur dort sinnvoll. Diese Videos sind allerdings in großer Zahl z.B. bei Nachrichtensendungen oder Videotelefonaten vertreten und könnten mit dem Verfahren somit verbessert werden.

5 Wahl von Beispielvideos

Für die Anwendung eines Verfahrens der wahrnehmungsbasierten Videokompression werden zunächst Beispielvideos ausgewählt, die als Referenz für den Vergleich mit einem klassischen Kompressionsverfahren dienen sollen. In den folgenden Unterkapiteln wird zunächst beschrieben, welche Kriterien bei der Auswahl von Videos beachtet werden sollten. Außerdem werden die ausgewählten Videos vorgestellt und erklärt, warum sie für die Anwendung des Verfahrens geeignet sind.

5.1 Kriterien zur Wahl

Für die Auswahl von Beispielvideos sind verschiedene Kriterien zu beachten. Zum Beispiel sollten die Videos unverarbeitet in einem Rohformat vorliegen, damit es zu keiner Verfälschung der Ergebnisse durch die Vorverarbeitung durch andere Videocodecs kommt. In verschiedenen Videodatenbanken sind diverse Videos im Rohformat zu finden, größtenteils „Standard“-Videos, die bereits für verschiedene Videocodecs verwendet wurden und sich für den Vergleich von Videokompressionsverfahren eignen. Je nach Inhalt und je nachdem was man testen oder vergleichen möchte, gibt es geeignete Videos in unterschiedlichen Auflösungen, unterschiedlichen Frameraten und mit verschiedenen Inhalten, wie zum Beispiel die Videosequenz „Akiyo“, die eine sprechende Nachrichtensprecherin vor blauen Bildschirmen im Hintergrund zeigt, oder die Videosequenz „Football“, die ein Football-Spiel zeigt.

Da der Fokus der Kodierung auf dem Videobild und nicht auf dem Ton des Videos liegt, kann dieser vernachlässigt werden. In den meisten Beispielvideos ist ohnehin keine Tonspur vorhanden. Es können Videos mit unterschiedlichen Auflösungen verwendet werden. Da H.265/HEVC für die bessere Kompression von Videos in HD-Auflösung entwickelt wurde, spricht nichts dagegen, entgegen der meisten klassischen Videos in QCIF- (176 x 144 Pixel) oder CIF-Auflösung (352 x 288 Pixel), auch höhere Auflösungen wie HD-Videos zu verwenden.

Als Verfahren der wahrnehmungsbasierten Videokompression liegt das in Kapitel 4.4 vorgestellte Verfahren der Gesichtserkennung vor, das verwendet werden kann um die Beispielvideos zu kodieren. Auch das vorliegende Verfahren fließt in die Auswahl der Beispielvideos mit ein. Um sich auch vollständig von seiner Arbeitsweise zu überzeugen, ist es sinnvoll Videos auszuwählen, in denen auch Gesichter vorhanden sind, die durch das Verfahren erkannt werden können, damit überhaupt erst ein Unterschied zum üblichen HEVC-Verfahren festgestellt werden kann.

Aus diesen Gründen hat sich die Suche nach geeigneten Beispielvideos auf Videos, in denen Menschen zu sehen sind, konzentriert, im Speziellen auf Videos mit erkennbarem Gesicht. Solche Inhalte sind sehr weit verbreitet und finden sich beispielsweise häufig bei Videos von Nachrichtensendungen oder Interviews, bei denen der Mensch im Vordergrund steht. Interessant sind aber ebenso Videos, in denen sich nicht nur Gesichter statisch vor einem ruhigen Hintergrund befinden, sondern auch Videos, in denen sich die Personen bewegen oder auch mehrere Personen im Bild sind. Die meisten Testvideos, die häufig verwendet werden, haben allerdings häufig nur eine geringe Auflösung. Das Ziel der Suche war es hingegen auch Videos in HD- oder Full-HD-Auflösung auszuwählen, um das vorliegende Verfahren zu testen. Videos in einer solchen Auflösung mit Gesichtern waren allerdings kaum zu finden, weshalb schließlich zwei eigene Videos in dieser Auflösung aufgenommen wurden.

Es wurden insgesamt für die Anwendung des wahrnehmungsbasierten Verfahrens drei unterschiedliche Beispielvideosequenzen ausgewählt, die im folgenden Kapitel näher beschrieben werden.

5.2 Eigenschaften der Originalvideos

Alle drei ausgewählten Videosequenzen liegen in einem unkomprimierten Format vor. Es wurde lediglich zuvor bereits ein Farb-Subsampling von 4:2:0 vorgenommen, die Videos sind also jeweils in Helligkeit (Y), die Abweichung von blau (U) und die Abweichung von rot (V) aufgeteilt, wobei die beiden Farbabweichungen nur die Hälfte der Auflösung des Helligkeitsbildes haben. Es handelt sich außerdem um die reinen Bilddaten, es sind keine Audiospuren enthalten.

Ursprünglich war geplant auch ein Video mit einer Auflösung von 1280 x 720 Pixeln zu verwenden, was aber auf Grund eines unbekannten Fehlers im vorliegenden wahrnehmungsbasierten Verfahren, der auch nach Rücksprache mit dem Autor nicht gelöst werden konnte, jedoch zunächst nicht möglich war. Stattdessen wurden zwei eigene Videos mit einer Auflösung von 1920 x 1080 Pixeln im Rohformat mit der Industriekamera „mvBlueCOUGAR-XD“ von MATRIX VISION aufgenommen und in das erforderliche YUV-Format umgewandelt, die dann als Beispielvideos in HD-Auflösung verwendet werden konnten.

5.2.1 Beispielvideo „Foreman“

Die erste ausgewählte Video ist die klassische Videosequenz „Foreman“[1], die häufig zur Demonstration von Videokompressionsverfahren verwendet wird und daher relativ bekannt ist. Das Video liegt unkomprimiert mit einer für CIF (Common Intermediate Format) üblichen Auflösung von 352 x 288 Pixeln und einer Framerate von 29,97 Frames pro Sekunde vor. Es enthält 300 Frames, was ungefähr einer Länge von 10 Sekunden entspricht und eine Dateigröße von ca. 43,5 Megabyte.

Die Videosequenz beginnt mit einer Nahaufnahme des Kopfes eines Bauarbeiters (Bauführers, englisch Foreman) vor einer Betonwand, der einen Helm trägt und etwas in die Kamera spricht. Dabei bewegt er sein Gesicht ein wenig und sein Mund und seine Gesichtszüge bewegen sich ohnehin auf Grund des Sprechens. Zusätzlich ist die Kameraführung etwas verwackelt. Nach circa 6 Sekunden Sprechen des Bauarbeiters schwenkt die Kamera auf einen sich in einer Baustelle befindenden Rohbau um. Mit dessen Bild und einer nach wie vor leicht wackelnden Kameraführung endet das Video nach ca. 10 Sekunden.

Ein Screenshot aus dem Video ist in Abbildung 13 dargestellt. Das linke Bild zeigt den Bauarbeiter während er spricht, das rechte Bild die Endszene des Rohbaus.

Abbildung 13: Frame 115 und 300 des Videos „Foreman“

Die Videosequenz eignet sich in sofern für die Anwendung des Verfahrens der Gesichtserkennung, als dort eine längere Zeit lang ein Gesicht - das des Bauarbeiters - zu sehen ist.

[1] von https://media.xiph.org/video/derf/

Es ist keine rein statische Szene, sondern sowohl die Kameraführung als auch der Kopf des Bauarbeiters bewegen sich. Außerdem bewegen sich seine Gesichtszüge und später schwenkt die Kamera um, sodass gar kein Gesicht mehr zu sehen ist.

Es wäre hier interessant zu sehen, ob das Gesicht bei der relativ niedrigen Auflösung von 352 x 288 Pixeln richtig erkannt wird und ob eine bessere Qualität des Gesichts - insbesondere Augen und Mund - im Gegensatz zum Hintergrund im Resultat wahrnehmbar ist.

5.2.2 Beispielvideo „Sprecher“

Das zweite ausgewählte Video ist eines der beiden selbst aufgenommenen Videos und trägt den Titel „Sprecher“. Es hat eine Full-HD-Auflösung von 1920 x 1080 Pixeln und eine Framerate von 15 Frames pro Sekunde. Es beinhaltet 60 Frames, was einer Länge von circa 4 Sekunden gleichkommt. Das Video hat eine Dateigröße von ungefähr 177,98 Megabyte.

Das Video zeigt den Oberkörper eines Sprechers vor einer weißen Tafel und weißen Wandfläche. Während des gesamten Videos spricht dieser und schaut dabei die meiste Zeit in die Kamera, wobei das Gesicht zwischenzeitlich auch etwas nach links schaut. Dabei bewegt er sich insgesamt nur wenig, die Kamera an sich ist statisch und bewegt sich nicht. Abbildung 14 zeigt zwei Frames des Videos.

Abbildung 14: Frame 20 und 43 des Videos „Sprecher“

Das Video hat eine deutlich höhere Auflösung als das erste Video und dadurch kann vermutlich eine deutlich höhere Kompression allein schon durch Anwendung des konventionellen HEVC erzielt werden. Zusätzlich ist relativ wenig Bewegung zu sehen und das Gesicht ist relativ nah an der Kamera, wodurch das Gesicht voraussichtlich gut erkannt werden kann. Die Erwartungen an das wahrnehmungsbasierte Verfahren sind hoch, da hier gute Bedingungen von Seiten des Videos durch die Größe des Gesichts im Bild und die ruhigen Bewegungen existieren.

5.2.3 Beispielvideo „Vortrag“

Die dritte ausgewählte Videosequenz „Vortrag“ hat ebenfalls eine Full-HD-Auflösung von 1920 x 1080 Pixeln und eine Framerate von 25 Frames pro Sekunde. Es handelt sich ebenfalls um ein selbst aufgenommenes Video. Bei 207 Frames hat es damit eine Länge von etwas mehr als 8 Sekunden. Die Dateigröße des Videos beträgt damit etwa 617 Megabyte. Die Abbildung 15 zeigt zwei Einzelbilder des Videos.

In diesem Video ist eine Person zu sehen, die vor der gleichen Szene wie im Video „Sprecher“ steht. Dieses mal spricht sie allerdings stehend und bewegt sich im Verlauf des Videos weiter nach links, während die Kamera mitschwenkt. Nach etwa 3 Sekunden läuft eine weitere Person durch das Bild und verdeckt kurz den Vortragenden. Anschließend wird die Tafel im Hintergrund in verschiedene Richtungen bewegt, bis die Kamera so weit nach links

Abbildung 15: Frame 23 und 87 des Videos „Vortrag“

geschwenkt hat, dass diese im Hintergrund nicht mehr zu sehen ist, sondern eine weitere weiße Tafel und ein Fenster. Der Vortragende spricht bis zum Ende des Videos und geht mit der Kamera weiter nach links. Durch die Helligkeit des Fensters wird das Gesicht des Vortragenden gegen Ende des Videos etwas dunkler.

Auch dieses Video ist geeignet für die Anwendung des Gesichtserkennungsverfahrens. Interessant wäre hier zu sehen, wie es sich verhält, wenn die Person durch das Bild läuft oder wenn die Tafel unregelmäßig während des Kameraschwenkens bewegt wird und wie viele Bits hier insgesamt eingespart werden können, wenn das Gesicht nur einen geringfügigen Teil des gesamten Videos ausmacht. Außerdem kann mit diesem Video geprüft werden, wie gut die Gesichtserkennung das Gesicht noch erkennt, wenn von Seiten der Belichtung keine optimalen Bedingungen herrschen.

6 Anwendung wahrnehmungsbasierter Kompression

Als Grundlage für die Anwendung eines Verfahrens der wahrnehmungsbasierten Videokompression dienen die oben genannten Videosequenzen. Es wird sowohl das konventionelle H.265/HEVC-Verfahren in seiner Form als Referenzsoftware HM9.0 verwendet, als auch das Verfahren der Gesichtserkennung von [Xu et al. 2014], das auf eben dieser Referenzsoftware-Version aufbaut.

6.1 Durchführung der Kompression

Vor der Kodierung der Videos müssen die beiden Encoder zunächst erstellt (gebaut) werden, was bei dem wahrnehmungsbasierten Verfahren das zusätzliche Einbinden von Bibliotheken erfordert. Außerdem werden Konfigurationsdateien erstellt. Diese beiden Schritte werden in den beiden folgenden Unterkapiteln beschrieben.

6.1.1 Referenzsoftware HM9.0

Der Code der HM-Referenzsoftware ist beim Fraunhofer Heinrich Hertz Institut veröffentlicht[1] und kann dort aus einem Software-Repository heruntergeladen werden. Der Encoder für das klassische HEVC-Verfahren kann dann entweder mittels make-Datei in Linux oder unter Windows mit verschiedenen Entwicklungsumgebungen wie Visual Studio, Eclipse oder Xcode, je nach Wahl des Anwenders, erstellt werden. Die Programmiersprache des Programmcodes ist C++.

In diesem Fall wird Visual Studio 2010 mit seiner Konsole und dem Kompiler zum Kompilieren des Encoders verwendet, da dieses auch von den Autoren für das Gesichtserkennungsverfahren verwendet wurde, so mögliche auftretende Probleme mit dem Gesichtserkennungsverfahren vermieden werden sollen und die beiden Vorgehensweisen vergleichbar bleiben sollen. Über eine Kommandozeile von Visual Studio kann dieser in der Konfiguration „Release“ gebaut werden. Dadurch wird sowohl der Encoder (TAppEncoder.exe) als auch der Decoder (TAppDecoder.exe) erstellt, denen über die übliche Kommandozeile Parameter, wie z.B. eine Konfigurationsdatei, übergeben werden können.

Der Encoder bekommt seine Einstellungen über verschiedene Parameter in der Kommandozeile vorgegeben, abhängig vom zu kodierenden Video. Dafür wird für jedes der ausgewählten drei Beispielvideos eine Konfigurationsdatei als Textdatei erstellt. Dort sind beispielsweise Angaben über die Höhe und Breite des zu kodierenden Videos in Pixel oder die Framerate gemacht. Die Konfigurationen für die drei Beispielvideos „Foreman“, „Sprecher“ und „Vortrag“ finden sich im Anhang unter II.1. Zudem befinden sich die Befehle zum Kodieren der Videos in Anhang II.2. Für verschiedene Einstellungen des Kodierungsvorgangs wird die Konfigurationsdatei *encoder_lowdelay_P_main.cfg*, die standardmäßig in der Referenzsoftware enthalten ist, verwendet. Dort sind zum Beispiel Einstellungen wie der gewöhnliche Quantisierungsparameter festgelegt. Dieser ist darin auf 32 gesetzt.

Mit den gemachten Einstellungen aus Anhang II.1 werden die Beispielvideos nun mit dem üblichen HEVC-Verfahren kodiert, indem die Videodatei im Rohformat (als .yuv-Datei) und die Konfigurationsdateien als Parameter an den Encoder übergeben werden. Beim Kodieren wird neben dem Resultat als .hevc-Datei auch eine weitere Datei, die das während des Kodiervorgangs bereits wieder dekodierte Resultat im yuv-Format enthält, erstellt. Diese kann dann mit dem Video, das als Eingabe für den Encoder diente oder auch mit dem Ergebnis

[1] https://hevc.hhi.fraunhofer.de/

des wahrnehmungsbasierten Verfahrens, verglichen werden. Das erspart es, das resultierende komprimierte Video mit dem Decoder wieder mit Zeit-, Rechen- und Arbeitsaufwand zu dekodieren und spart so einen Arbeitsschritt ein.

Die Referenzsoftware ist entwickelt worden, um Entwicklern eine einfache Möglichkeit zu bieten verschiedene Tools zu testen oder den HEVC-Codec um eigene Implementationen zu erweitern und mögliche Experimente zu diesen durchzuführen. Diese Softwareimplementierung ist daher nicht besonders effizient und hat auch nicht das Ziel effizient zu sein - was sich auch in der Kodierungszeit bemerkbar macht -, sondern eher einfach erweiterbar zu sein [Bossen Flynn Sühring 2012]. Zusätzlich kostet die gleichzeitige Dekodierung des Videos darüber hinaus Rechenzeit. Des Weiteren ist die verwendete Version (HM9.0) nicht mehr aktuell und auch daher sind dort nicht alle aktuell gemachten Verbesserungen enthalten. Sie wird jedoch verwendet um Vergleichbarkeit zu erhalten, da das Gesichtserkennungsverfahren auf dieser aufbaut.

Nachdem die Kodierung der Videos abgeschlossen ist, erhält man eine Übersicht über das kodierte Video. Dort sind Informationen über den PSNR-Wert des Videos, aufgeteilt in Y-, U- und V- Komponenten, sowie die Bitrate, die Anzahl der geschriebenen Bytes und die Gesamtdauer für die Kodierung in Sekunden aufgelistet.

6.1.2 Gesichtserkennungsverfahren

Der vollständige Programmcode des Gesichtserkennungsverfahren von [Xu et al. 2014] wurde freundlicherweise von den Autoren zur Verfügung gestellt. Da dieses auf der Referenzsoftware HM9.0 aufbaut bzw. die Referenzsoftware für die Zwecke des Verfahrens verändert wurde, ist die Vorgehensweise bei der Kodierung von Videos ähnlich wie in Kapitel 6.1.1. Auch hier können entweder make-Dateien unter Linux oder Visual Studio und weitere zum Bauen des Encoders verwendet werden. Da die Vorgehensweise bei Visual Studio aus Kapitel 6.1.1 unkompliziert funktioniert hat, wird bei dem Gesichtserkennungsverfahren ebenfalls Visual Studio verwendet.

Für die Gesichtserkennung wird zusätzlich OpenCV benötigt. OpenCV ist eine Open-Source Software-Bibliothek für Computer Vision mit über 2500 optimierten Algorithmen unter anderem für C++ [OpenCV 2016], worin der Code des Verfahrens und der Code der Referenzsoftware geschrieben ist. Es muss also vor dem Bauen des Encoders OpenCV installiert werden und die Pfade zu den benötigten Bibliotheksdateien eingebunden werden. OpenCV wurde in der neusten Version (OpenCV 2.4.13) installiert und in den Einstellungen in Visual Studio die Pfade zu OpenCV als Include-Verzeichnisse angegeben. Zusätzlich werden die benötigten Bibliotheken für das Gesichtserkennungsverfahren angegeben. Folgende Bibliotheken werden hier benötigt:

- *opencv_core2413d.lib*
- *opencv_highgui2413d.lib*
- *opencv_imgproc2413d.lib*
- *opencv_objdetect2413d.lib*

Ist OpenCV richtig eingebunden, können Encoder und Decoder analog zur Referenzsoftware über die Konsole von Visual Studio gebaut werden.

Der Programmcode des Gesichtserkennungsverfahrens ist allerdings nicht veröffentlicht worden und daher ist das Gesamtverfahren von den Autoren nicht optimiert worden, weshalb das eigentliche Gesichtserkennungsverfahren einzeln vorab durchgeführt wird und die Ergebnisse dieses Verfahrens erst im Anschluss im Encoder verwendet werden. Die Kodierung eines Videos mit diesem Verfahren besteht also dann aus zwei Teilen: Der Gesichtserkennung, die

alle Ergebnispunkte des Verfahrens liefert und der eigentlichen Kodierung, die diese Punkte dann als Eingabe verwendet. Es entstehen dadurch aber keine großen zeitlichen oder performanzrelevanten Nachteile, da das Gesichtserkennungsverfahren bereits in wenigen Sekunden vollständig durchgeführt ist und dann die eigentliche Kodierung beginnen kann.

Die Konfigurationsdateien, die dem Encoder übergeben werden, werden für eine bessere Vergleichbarkeit der Resultatvideos aus Kapitel 6.1.1 übernommen. Lediglich die Dateinamen für die Ausgabedateien werden darin verändert, um diese von den mit dem üblichen HEVC-Verfahren kodierten Videodateien unterscheiden zu können. Für die Einstellungen des Encoders wird ebenfalls die Konfiguration aus *encoder_lowdelay_P_main* verwendet.

Analog zum Kodieren eines Videos mit der Referenzsoftware werden auch beim Gesichtserkennungsverfahren die Konfigurationsdateien per Kommandozeile an den Encoder übergeben und die Videos kodiert. Hier entstehen ebenfalls sowohl das fertige Resultat als auch eine bereits dekodierte Version im Rohformat für den Vergleich mit anderen Videos.

6.2 Auswertung der Kompression

Nachdem alle drei Beispielvideos sowohl mit dem üblichen HEVC-Verfahren, im speziellen als Implementierung in der HM9.0-Referenzsoftware, als auch mit dem veränderten Gesichtserkennungsverfahren aus Kapitel 4.4 mit den gleichen Einstellungen kodiert wurden, können die entstandenen Videos miteinander verglichen werden. Die aus der Kodierung resultierenden Rohdateien (die bei der Kodierung gleichzeitig wieder dekodierten HEVC-Videos) können mit einem YUV-Viewer - in diesem Fall wird der freie YUV-Viewer „viEWYUV“ verwendet - visuell dargestellt werden.

6.2.1 Ergebnis bei „Foreman“

Das Beispielvideo „Foreman“ wurde mit der Referenzsoftware HM9.0 kodiert. Das resultierende in HEVC kodierte Video hat eine Dateigröße von ca. 1,3 MB, was im Vergleich zum unkomprimierten Video mit ungefähr 43,5 MB eine Ersparnis von 42,2 MB und damit ca. 97% bedeutet. Die Einsparung konnte allerdings nur erfolgen, da das Verfahren verlustbehaftet ist und Videodaten bei der Kodierung unwiderruflich verloren gegangen sind. Das Resultatvideo des Gesichtserkennungsverfahrens hat eine Dateigröße von ca. 1,26 MB und ist damit 7.473 Byte kleiner als das übliche Verfahren. Das veränderte Verfahren konnte also bei diesem Video zwar mehr Daten sparen als das übliche, aber die gesparten Daten sind mit knapp 7 Kilobyte nur sehr gering. Die Bitrate liegt mit 1025,19 kB pro Sekunde beim üblichen Verfahren und 1019,41 kB pro Sekunde beim Gesichtserkennungsverfahren ebenfalls ungefähr im gleichen Bereich, sie ist beim letzteren Verfahren geringfügig kleiner.

Nach dem Kodiervorgang im üblichen HM-Verfahren als auch beim Gesichtserkennungsverfahren, das auf diesem aufbaut, erhält man eine Übersicht über verschiedene Werte des kodierten Videos und eine kurze Statistik über den Kodiervorgang. Auf Grund der gleichen Einstellungen, die beide Verfahren für den Decoder übergeben bekommen haben, sind auch die beiden Anzahlen der als I-Frame, P-Frame oder B-Frame kodierten Frames identisch. Allerdings unterscheiden sich beide Verfahren deutlich in ihrer Kodierungszeit. Das „normale“ Verfahren hatte für die insgesamt 300 Frames des Videos eine Kodierungszeit von ca. 1324 Sekunden, was in etwa 22 Minuten und einem durchschnittlichen Durchsatz von ca. 0,2265 Frames pro Sekunde entspricht. Das Gesichtserkennungsverfahren brauchte für die Kodierung des 300 Frames langen Videos eine Zeit von nur etwa 505 Sekunden, was ungefähr 8 Minuten und einem durchschnittlichen Durchsatz von 0,5937 Frames pro Sekunde entspricht. Die Kodierungszeit mit dem veränderten Verfahren ist also deutlich geringer (ca.

61,85%) als bei dem üblichen Verfahren, was damit zusammenhängt, dass bei dem Gesichtserkennungsverfahren zusätzlich zu der Veränderung der Quantisierungsparameter auch die Blockunterteilung verändert wurde. Diese wurde wie in Kapitel 4.4 beschrieben so verändert, dass nur in den relevanten Bereichen, wie Gesichtern (insbesondere Augen und Mund), eine Unterteilung in die maximale Tiefe erfolgt und in unwichtigen Teilen, wie dem Hintergrund, die Tiefe begrenzt wird. Dadurch wird offensichtlich ein großer Rechenaufwand für die Berechnung der Unterteilung der Blöcke eingespart, was das Verfahren insgesamt deutlich schneller macht. Andere Teile der Referenzsoftware wurden nicht verändert, die einen solchen zeitlichen Unterschied hätten hervorrufen können.

Auch die einzelnen Frames des Videos können miteinander verglichen werden. Vermutlich auf Grund der relativ geringen Auflösung des CIF-Videos sind Unterschiede in den beiden Videos nur sehr gering. Diese fallen nur bei genauem Hinsehen auf. Ein geringer, aber zu erkennender Unterschied ist beispielsweise ein Teil des Hintergrundes in Frame 59 (abgebildet in Abbildung 16).

Dort steht der Bauarbeiter vor einem Betonbau, an dessen Unterseite sich etwas Schmutz befindet (rechts neben dem Bauarbeiter). Das Gesicht ist in diesem Frame und auch im restlichen Video in beiden Videos fast gleich, es gibt nur sehr geringe Unterschiede, die mit dem bloßen Auge kaum bis gar nicht und in einem Differenzbild nur sehr gering zu erkennen sind. Auch dies liegt vermutlich an der geringen Videoauflösung mit 352 x 288 Pixeln. Was jedoch auffällt, ist der angesprochene Schmutz am Betonbau im Hintergrund. Da das Gesicht im Gesichtserkennungsverfahren als wichtig (Gewichtung in der Weight-Map mit 2 oder mehr) und als Vordergrund erkannt wird, ist alles andere als das Gesicht der Hintergrund. Dieser wird als nicht so wichtig (maximal Wertung 1 in der Gewichtung) bewertet. Das wirkt sich sowohl auf den Quantisierungsparameter für diesen Bereich, als auch auf die maximal mögliche Unterteilung der Blöcke in diesem Bereich aus.

Abbildung 16: Vergleich des Frame 59 in „Foreman“
(links/rechts oben: Übliches HEVC, Mitte/rechts unten: Gesichtserkennungsverfahren)

Das linke Bild in Abbildung 16 zeigt Frame 59 nach der Kodierung mit dem üblichen HEVC-Verfahren, das Bild rechts neben diesem zeigt den Frame nach der Kodierung mit dem Gesichtserkennungsverfahren. Die beiden Bildausschnitte ganz rechts zeigen vergrößert den erwähnten Schmutz am Betonbau im Hintergrund nach der Kodierung mit dem üblichen HEVC-Verfahren (oben) und dem veränderten Verfahren (unten).

Bei genauerer Betrachtung ist zu erkennen, dass der Schmutz im unteren Bild insgesamt etwas verwaschener und unschärfer wirkt, hier also nicht so viele Details erhalten geblieben sind wie im oberen mit dem üblichen Verfahren kodierten Frame. Dies ist sowohl auf die veränderte Blockunterteilung als auch den veränderten Quantisierungsparameter zurückzuführen. Das, worauf der menschliche Zuschauer in diesem Fall jedoch achtet, ist nicht der Schmutz an dem Betonbau im Hintergrund, sondern das Gesicht des Bauarbeiters. Und dieses ist im rechten Bild nicht verwaschen oder unschärfer als im linken Bild. Allerdings gibt

es auch nicht so gravierende Verbesserungen im Gegensatz zum üblichen HEVC-Verfahren, wie es beispielsweise in Abbildungen aus [Xu et al. 2014] der Fall war. Die in der Dateigröße eingesparten Bits sind also größtenteils vermutlich aus dem Hintergrund eingespart worden.

Nachdem die Kamera vom Bauarbeiter auf den Rohbau umgeschwenkt ist, gibt es kein Gesicht mehr im Bild, das erkannt werden könnte. Im Bild ist also nur Hintergrund zu sehen, der mit weniger Unterteilungstiefe in Blöcke strukturiert und schlechter quantisiert wird. Abbildung 17 zeigt Frame 184 des Videos, links mit dem üblichen und rechts mit dem veränderten Verfahren kodiert.

Abbildung 17: Vergleich des Frame 284 in „Foreman“
(links: Übliches HEVC, rechts: Gesichtserkennungsverfahren)

Dort ist zu sehen, dass die Bäume im Hintergrund - vor allem ganz rechts - etwas verschwommener sind. Blätter sind im linken Bild noch ein bisschen besser zu erkennen, im rechten Bild gehen alle Blätter fast in eine leicht verwaschene grüne Fläche über. Auch hier sind die Unterschiede der beiden Verfahren nicht enorm, aber doch zu erkennen.

Während der Kodierung der Videos wird für jeden Frame ein PSNR-Wert der Helligkeitswerte (Y) gebildet und ein durchschnittlicher PSNR-Wert für alle Frames in der Übersicht nach Abschluss der Kodierung angezeigt. Das übliche Verfahren brachte demnach einen durchschnittlichen PSNR-Wert von 41,46 dB, das Gesichtserkennungsverfahren im Durchschnitt einen PSNR-Wert von 40,26 dB. Es ist hier also etwas schlechter, was vermutlich damit zusammenhängt, dass die Bildflächen des Hintergrunds größer sind als die des Gesichts. Im letzten Teil des Videos ist gar kein Gesicht mehr vorhanden und dort ist das Rauschen im Bild etwas größer. Außerdem wird ein SSIM-Wert für die Bewertung verwendet. Der durchschnittliche SSIM-Wert wurde mit dem „MSU Video Quality Measurement Tool“ berechnet und ist bei beiden Videos fast identisch, das übliche Verfahren ist lediglich um 0,0008 Punkte besser als das Gesichtserkennungsverfahren.

Vergleicht man nun nur bestimmte Bildbereiche miteinander, kann verhindert werden, dass der gesamte PSNR-Wert (z.B. durch eine schlechtere Kodierung des Hintergrundes) verschlechtert wird, obwohl das Verfahren möglicherweise besser gearbeitet hat. Daher wurden bei verschiedenen Frames stichprobenartig PSNR- und SSIM-Wert sowohl für das gesamte Bild, also auch für das Gesicht einzeln und den Hintergrund einzeln berechnet und miteinander verglichen. Die genauen Werte von PSNR und SSIM sind in Anhang II.3 aufgeführt.

Festzustellen ist, dass die Resultate der beiden Verfahren sich auch hier sehr ähneln. In einigen Frames ist der PSNR-Wert des Gesamtbildes beim üblichen Verfahren besser (z.B. bei den Frames 24, 59 oder 284), wobei hier insgesamt keine eindeutigen Unterschiede festzustellen sind, denn vereinzelt sind auch Frames des Gesichtserkennungsverfahrens (z.B. die Frames 125 und 149) bei PSNR besser. Ähnliches gilt für den SSIM-Wert bei den genannten Frames. Beim Vergleich der Bildausschnitte, die nur das Gesicht enthalten, haben beide Verfahren sowohl deutlich bessere PSNR-, als auch deutlich bessere SSIM-Werte, wobei sich auch

hier entgegen der Erwartung kein deutlich besseres Ergebnis für das Gesichtserkennungsverfahren äußert. Einige Frames sind zwar in beiden Werten beim veränderten Verfahren besser, allerdings hat auch das übliche Verfahren einige Frames mit besseren Werten. Im Vergleich der Hintergrundbereiche ist das übliche Verfahren in den meisten Frames besser. Ein durchgehend besseres Verfahren gibt es bei diesem Video also nicht.

Das Ergebnis der Kodierung des Videos „Foreman“ fasst Tabelle 1 zusammen. Aus Platzgründen wird in den folgenden Tabellen (Tabellen 1, 2 und 3) (1) für das übliche HEVC-Verfahren und (2) für das Gesichtserkennungsverfahren verwendet.

	Dateigröße	**Kodierungszeit**	**∅ PSNR**	**∅ SSIM**	**Bitrate**
(1)	1.325.673 B	1324,419 Sek.	41,4648 dB	0,99626	1025,187 kB/s
(2)	1.318.200 B	505,330 Sek.	40,2596 dB	0,99546	1019,408 kB/s
Diff.	-7.473 B	-819,089 Sek.	-1,2052 dB	-0,0008	-5,779 kB/s

Tabelle 1: Vergleich des Videos „Foreman“

Zusammenfassend sind die beiden Videos sich ziemlich ähnlich, bei dem mit dem Gesichtserkennungsverfahren kodierten Video ist ein leichter Qualitätsverlust im Bereich des Hintergrundes festzustellen, worauf der Mensch aber nur bei bewusstem Hinsehen achtet, da er in der Regel das Gesicht fokussiert. Dafür hat dieses Verfahren einen deutlichen zeitlichen Vorteil gegenüber des üblichen HEVC-Verfahrens bei der Kodierung auf Grund der veränderten Blockunterteilung. Von der Dateigröße, PSNR- und SSIM-Werten her sind beide Videos ebenfalls vergleichbar gut, wobei das übliche Verfahren insgesamt nur minimal besser ist. Möglicherweise hängt dieses nicht eindeutige Ergebnis auch mit der geringen Auflösung des Videos zusammen. Das Verfahren hat bei dieser Auflösung also keine deutlich verbesserte optische Qualität oder andere Vorteile erzeugen können.

6.2.2 Ergebnis bei „Sprecher“

Auch das Video „Sprecher“ wurde sowohl mit dem üblichen HEVC der Referenzsoftware als auch dem veränderten Gesichtserkennungsverfahren kodiert. Die Resultatvideos haben eine Dateigröße von 508.771 Byte beim üblichen HEVC-Verfahren, was in etwa 496,85 kB entspricht und 485.026 Byte, was ca. 473,66 kB entspricht. Im Vergleich zum unkomprimierten Video bedeutet dies eine Einsparung von 177,5 Megabyte (ca. 99,73%). Das Gesichtserkennungsverfahren ist hier verglichen mit dem üblichen HEVC-Verfahren der Referenzsoftware noch mal um ca. 23,2 kB besser (ca. 99,74%), wobei dieser Unterschied im Hinblick auf die gesamte Datenmenge nur relativ gering ist. Das Gesichtserkennungsverfahren ist also von der Dateigröße des resultierenden Videos ungefähr gleich gut wie das übliche Verfahren.

Auch bei diesem Beispielvideo haben beide Verfahren die gleichen Einstellungen übergeben bekommen, weshalb hier, wie bereits im Video „Foreman“ zuvor, die Anzahlen der I-, P- und B-Frames in beiden Resultatvideos identisch sind. Auch bei diesem Video unterscheiden sich beide Verfahren deutlich in ihrer Kodierungszeit. Während das übliche Verfahren für die Kodierung ungefähr 2181 Sekunden benötigte - was ca. 36 Minuten und einem durchschnittlichen Datendurchsatz von 0,0275 Frames pro Sekunde entspricht -, betrug die Kodierungszeit beim Gesichtserkennungsverfahren etwa 1175 Sekunden (ca. 20 Minuten) und einem Datendurchsatz von 0,051 Frames pro Sekunde. Die Bitraten lagen bei ca. 1017,5 kB pro Sekunde beim üblichen Verfahren und bei etwa 970,1 kB pro Sekunde beim veränderten Verfahren. Die deutlich längeren Zeiten bei diesem Video ist auf die deutlich höhere Auflösung im Vergleich zum Video „Foreman“ zurückzuführen. Das veränderte Verfahren ist hier ebenfalls mehr als doppelt so schnell wie das übliche Verfahren (etwa 53,86%), allerdings ist die Zeitersparnis

nicht so deutlich höher wie im vorherigen Video.

Beim optischen Vergleich einzelner Frames des üblichen HEVC mit dem Gesichtserkennungsverfahren fällt im Gegensatz zum „Foreman“-Video in niedriger Auflösung auf, dass das Gesicht erkannt wurde und mehr Details im kodierten Video erhalten bleiben als beim üblichen Verfahren. Ein Beispiel für diese Details zeigt Frame 9 des Videos (Abbildung 18).

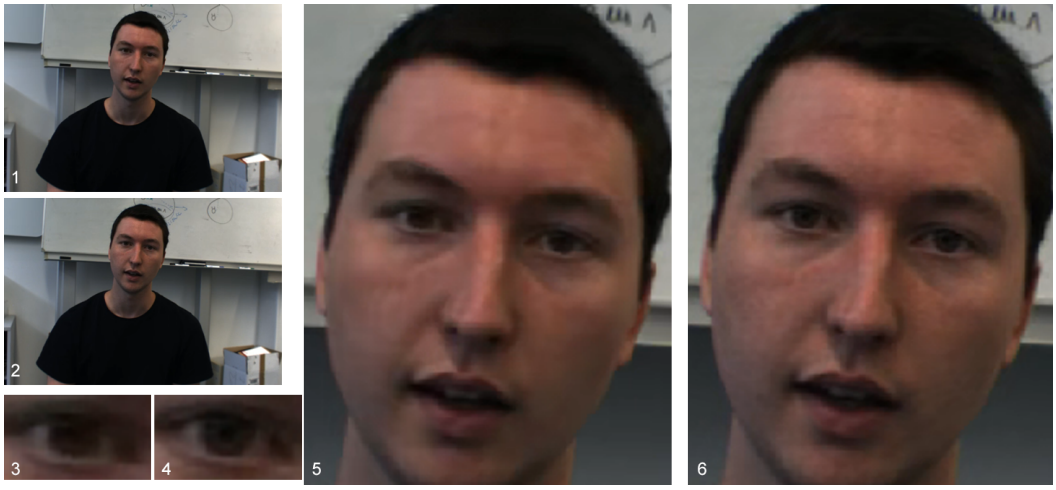

Abbildung 18: Vergleich des Frame 9 in „Sprecher“
(1/3/5: Übliches HEVC, 2/4/6: Gesichtserkennungsverfahren)

Die beiden Bildausschnitte links oben zeigen den gesamten Frame kodiert mit dem üblichen Verfahren (oben) und dem veränderten Verfahren darunter. Die beiden Bilder auf der rechten Seite zeigen das Gesicht vergrößert (links übliches, rechts verändertes Verfahren). In der Ecke links unten ist jeweils das linke Auge der beiden Frames vergrößert dargestellt. Es ist deutlich zu erkennen, dass der Bereich des Gesichts mit dem Gesichtserkennungsverfahren optisch deutlich verbessert wurde. Das linke Bild des Gesichts ist etwas verwaschener und hat bei der Kodierung weniger Strukturen erhalten können, während im rechten Bild des Gesichts viele Details noch vorhanden sind. Besonders fällt dies in den Bereichen der Augen und des Mundes auf, die die höchste Wertung in der Weight-Map des Verfahrens erhalten haben. Die Augen im Frame des veränderten Verfahrens sind noch so detailliert, dass man die Pupille beispielsweise noch deutlich erkennen kann. Im üblichen Verfahren geht diese in den übrigen dunklen Farben des Auges unter, man kann sie nur im stark vergrößerten Bild erahnen. Im unteren Bereich des Mundes erkennt man im mit dem Gesichtserkennungsverfahren kodierten Bild noch einen kleinen Teil der Zähne, während diese im üblichen Verfahren mit den Farben von Lippe und Zunge fast verschmolzen und kaum noch zu erkennen sind. Allerdings ist in diesem Video vermutlich auch auf Grund der geringen Schärfe, die bereits durch die Kamera beim Hintergrund hinzugefügt wurde, kein deutlicher optischer Qualitätsverlust, sondern lediglich minimal stärkere Unschärfe beim Hintergrund zu erkennen.

Ähnlich verhält es sich beispielsweise bei Frame 47, der in Abbildung 19 dargestellt ist. Das Gesicht insgesamt wirkt im rechten Bild, das mit dem Gesichtserkennungsverfahren kodiert wurde, deutlich schärfer und detaillierter. Im linken Bild ist der Mund, insbesondere die obere Lippe deutlich verpixelter und die Zähne sind unscharf. Im rechten Bild ist dieser Bereich deutlich schärfer und die Zähne sind besser voneinander differenzierbar. Auch bei diesem Frame sind im Hintergrund so gut wie keine optischen Unterschiede feststellbar.

Die in der Übersicht nach dem Kodieren ausgegebenen PSNR-Werte der beiden Videos sind wie folgt: Das übliche Verfahren hat einen durchschnittlichen PSNR-Wert von etwa 43,96 dB, das Gesichtserkennungsverfahren liegt bei durchschnittlich ca. 44,57 dB, es ist hier

Abbildung 19: Vergleich des Frame 47 in „Sprecher“
(links: Übliches HEVC, rechts: Gesichtserkennungsverfahren)

also insgesamt etwas besser als das übliche HEVC. Der durchschnittliche SSIM-Wert wurde wie bereits im Beispielvideo zuvor wieder durch das „MSU Video Quality Measurement Tool“ berechnet und beträgt beim üblichen Verfahren ungefähr 0,99452, ist also ziemlich nah an den Strukturen des Originalvideos (theoretischer Wert von 1,0), beim veränderten Verfahren ca. 0,99484. Hier ist also das veränderte Verfahren minimal besser, allerdings liegt der Unterschied hier lediglich im Zehntausendstel-Bereich.

Wie bereits im Video „Foreman“ angewendet, werden auch hier die PSNR- und SSIM-Werte genauer betrachtet, indem einzelne Bildbereiche einzeln mit den beiden Metriken bewertet werden. Auch hier wird eine Trennung zwischen dem Bildbereich, in dem sich das Gesicht befindet und dem Hintergrund durchgeführt. Demnach kann festgestellt werden, dass das Gesichtserkennungsverfahren anders als noch bei „Foreman“ bei allen Frames bessere Werte sowohl in PSNR als auch SSIM erreichen konnte. In den bewerteten Frames sind die Werte beim Gesichtserkennungsverfahren im gesamten Bild lediglich minimal höher als beim üblichen Verfahren. Ähnliches gilt für den Hintergrund alleine. Hier sind die Werte je nach Frame fast gleich und unterscheiden sich z.B. nur um Hundertstel-Stellen bei SSIM. Bei der alleinigen Betrachtung des Gesichts weichen die Werte schon eher von einander ab. Hier übertrifft das veränderte Verfahren das übliche deutlich, was sich auch optisch beispielsweise durch die Schärfe im Bild bemerkbar macht. Die genauen Ergebnisse der PSNR- und SSIM-Berechnungen sind im Anhang unter II.4 aufgeführt.

Das Ergebnis der Kodierung des Videos „Sprecher“ wird in der folgenden Tabelle 2 zusammengefasst.

	Dateigröße	**Kodierungszeit**	**∅ PSNR**	**∅ SSIM**	**Bitrate**
(1)	508.771 B	2180,903 Sek.	43,9550 dB	0,99452	1017,542 kB/s
(2)	485.026 B	1174,736 Sek.	44,5701 dB	0,99484	970,052 kB/s
Diff.	-23.745 B	-1006,167 Sek.	+0,6151 dB	+0,00032	-47,49 kB/s

Tabelle 2: Vergleich des Videos „Sprecher“

Zusammenfassend für das Video „Sprecher“ lässt sich sagen, dass die Videoauflösung große Auswirkungen auf die Effizienz des Verfahrens hat. Anders als beim Video „Foreman“ konnten deutlichere Verbesserungen in den Bildbereichen des Gesichts, insbesondere bei Augen und Mund, festgestellt werden. Bei einzelner Betrachtung von Gesicht und Hintergrund lässt sich im Gesicht nicht nur optisch, sondern auch in PSNR- und SSIM-Werten eine deutliche Verbesserung gegenüber des üblichen Verfahrens feststellen, im Hintergrund sind minimale Verschlechterungen zu erkennen, die aber gering sind und kaum auffallen, weshalb das ver-

änderte Verfahren insgesamt auch im gesamten Bild bessere Werte erzielen konnte als das übliche. Zudem ist das Gesichtserkennungsverfahren auch bei der Full-HD-Auflösung dieses Videos mehr als doppelt so schnell in der Kodierungszeit.

6.2.3 Ergebnis bei „Vortrag“

Das Video „Sprecher“ wurde ebenfalls mit beiden Verfahren kodiert. Das mit dem üblichen Verfahren kodierte Video hat eine Dateigröße von ca. 1,02 MB, das mit dem Gesichtserkennungsverfahren kodierte Video ist etwa 1,0 MB groß. Das veränderte Verfahren war also auch in diesem Fall wieder in der Lage das Resultatvideo zu verkleinern, allerdings wieder nur minimal im Kilobyte-Bereich. Im Vergleich zum unkomprimierten Eingangsvideo konnten beide Verfahren wieder deutliche Reduzierungen der Datenmenge von mehr als 615 MB (etwa 99,8%) erreichen, was im Vergleich zu den vorherigen Videos der höchste Wert ist.

Für die Kodierung benötigte das übliche Verfahren eine Zeit von ungefähr 8448 Sekunden, was in etwa 140 Minuten entspricht und damit deutlich entfernt ist von einer Echtzeitkodierung. Der durchschnittliche Datendurchsatz liegt demnach bei ca. 0,025 Frames pro Sekunde. Allerdings kommt es bei der Referenzsoftware nicht auf eine möglichst große Effektivität, sondern viel mehr auf eine klare und einfach zu verändernde Umsetzung des HEVC-Verfahrens ohne Performanz-Optimierungen an. Das veränderte Gesichtserkennungsverfahren konnte das Video in einer Zeit von etwa 4776 Sekunden kodieren, was ungefähr 80 Minuten und damit etwas mehr als die Hälfte der Kodierungszeit des üblichen Verfahrens entspricht. Der durchschnittliche Datendurchsatz liegt somit bei ca. 0,043 Frames pro Sekunde. Die Bitrate liegt bei ungefähr 1040,8 kB pro Sekunde, während sie beim veränderten Verfahren bei ca. 1017,4 kB pro Sekunde liegt. Wie bereits in den Videokodierungen zuvor ist also auch hier das veränderte Verfahren deutlich schneller, was auf die veränderte Blockunterteilung zurückzuführen ist.

Rein optisch erkennt man bei diesem Video nicht so deutliche Verbesserungen durch das Gesichtserkennungsverfahren wie beim vorherigen Video „Sprecher“, in dem für das Verfahren sehr gute Bedingungen durch geringe Bewegungen und ein relativ großer Bildanteil mit Gesicht gegeben waren. In „Vortrag“ ist der Anteil an Gesicht in dem Video deutlich geringer und insgesamt ist deutlich mehr Bewegung, sowohl durch die Kamera, als auch das Schieben der Tafel im Hintergrund und die Person, die durch das Bild läuft und nicht zuletzt durch die Bewegungen des Vortragenden selbst vorhanden.

In Frame 89 des Videos (ein Ausschnitt daraus ist dargestellt in Abbildung 20) ist der Vergleich des Gesichts in beiden kodierten Videos zu sehen. Im linken Bild wurde das Gesicht mit dem üblichen Verfahren kodiert und hier wirkt die von der Kamera aus linke Gesichtshälfte etwas verpixelter. Im rechten Bild wirkt das Gesicht etwas schärfer.

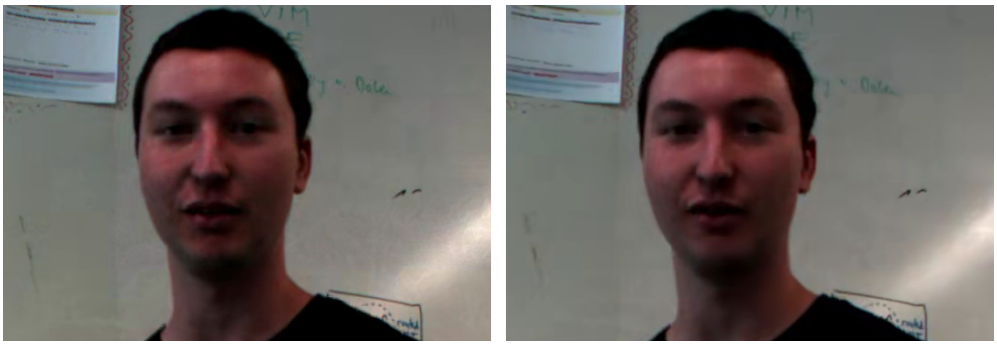

Abbildung 20: Vergleich des Frame 89 in „Vortrag“
(links: Übliches HEVC, rechts: Gesichtserkennungsverfahren)

Im Hintergrund fällt im rechten Bild allerdings der Tafelanschrieb rechts und der schwarze Strich links neben dem Kopf auf. Diese werden richtigerweise nicht als Gesicht erkannt und zählen daher zum Hintergrund für das Gesichtserkennungsverfahren. Dadurch erhalten sie eine geringere Wertung und sind dadurch im rechten Bild unschärfer als im linken Bild, da hier größere Blöcke und höhere Quantisierungsparameter verwendet wurden.

Abbildung 21 zeigt einen weiteren Bildausschnitt aus Frame 134 des Videos kodiert mit dem üblichen Verfahren links und dem veränderten Verfahren rechts. Durch die Helligkeit des Fensters links neben dem Vortragenden ist das Gesicht relativ dunkel. Im linken Bild (Referenzsoftware) wirkt das Gesicht wieder minimal verpixelter als im rechten Bild. Auch in diesem Frame fällt der etwas unscharfe Tafelanschrieb im mit dem Gesichtserkennungsverfahren kodierten Video auf.

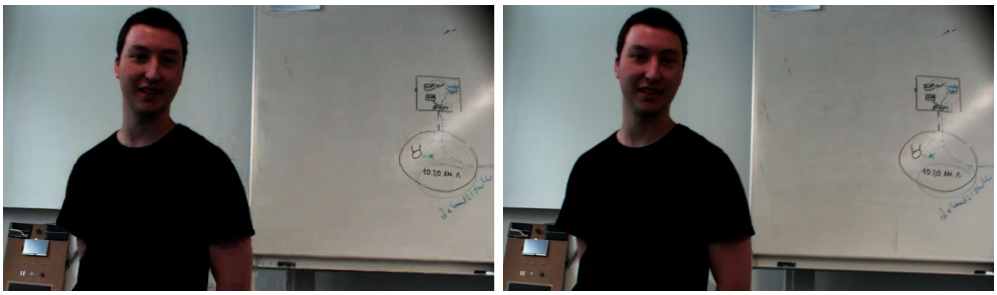

Abbildung 21: Vergleich des Frame 134 in „Vortrag“
(links: Übliches HEVC, rechts: Gesichtserkennungsverfahren)

Insgesamt hat das Video einen durchschnittlichen PSNR-Wert von 39,3 dB beim üblichen HEVC-Verfahren und einen besseren Wert von 42,18 dB beim veränderten Gesichtserkennungsverfahren. Der durchschnittliche SSIM-Wert des üblichen Verfahrens unterscheidet sich nur sehr geringfügig von dem des veränderten Verfahrens. Er liegt bei dem mit der Referenzsoftware kodierten Video bei 0,99256 und beim Gesichtserkennungsverfahren bei 0,99419, er ist dort also minimal besser.

Vergleicht man nun einzelne Bildbereiche mit PSNR und SSIM für sich, fällt auf, dass das Gesichtserkennungsverfahren bei diesem Video bei den Gesamtbildern bei allen Frames sowohl in PSNR als auch in SSIM besser ist, allerdings nicht in jedem Frame deutlich bessere Werte für Gesichtsbereiche erzielt werden konnten. Teilweise gibt es Frames, bei denen der PSNR- oder SSIM-Wert des Gesichts allein schlechter ist als beim üblichen HEVC. Möglicherweise wurde auf Grund der stärkeren Bewegungen oder durch die erschwerten Bedingungen durch die Unterbelichtung des Gesichts in manchen Frames das Gesicht durch das Verfahren nicht durchgehend gut erkannt. Bei den Hintergrundbereichen ist das veränderte Verfahren jedoch in allen verglichenen Frames in beiden Metriken besser. Alle genauen PSNR- und SSIM-Werte finden sich im Anhang unter II.5.

Einen Überblick über das Ergebnis der Kodierung des Videos „Vortrag“ gibt die folgende Tabelle 3:

	Dateigröße	**Kodierungszeit**	**∅ PSNR**	**∅ SSIM**	**Bitrate**
(1)	1.077.254 B	8447,832 Sek.	39,3043 dB	0,99256	1040,825 kB/s
(2)	1.052.966 B	4775,971 Sek.	42,1827 dB	0,99419	1017,358 kB/s
Diff.	-24.288 B	-3671,861 Sek.	+2,8784 dB	+0,00163	-23,467 kB/s

Tabelle 3: Vergleich des Videos „Vortrag“

Insgesamt ist das Resultat des Gesichtserkennungsverfahrens bei diesem Video nach den Metriken PSNR und SSIM wie auch im Video „Sprecher“ zuvor besser als das übliche HEVC-

Verfahren. Allerdings fallen hier vermutlich auf Grund der Unterbelichtung des Gesichts in manchen Frames und der stärkeren Bewegungen optisch keine so deutlichen Qualitätsverbesserungen im Gesicht auf. Der Hintergrund wirkt etwas unschärfer in verschiedenen Frames, was jedoch kein Problem darstellt, sondern durch das Verfahren durchaus gewollt ist, da der Mensch auf das Gesicht bzw. den Vortragenden insgesamt achtet. Zudem existieren einzelne Frames, in denen die PSNR- und SSIM-Werte auch im Bereich des Gesichts etwas schlechter sind als beim üblichen Verfahren. Bei diesem Video konnte das veränderte Verfahren zusammenfassend also nicht so überzeugen wie im Video zuvor, ist jedoch etwas besser als das übliche HEVC-Verfahren.

6.3 Fazit

Als Fazit der Anwendung des Gesichtserkennungsverfahrens aus Kapitel 4.4 kann festgestellt werden, dass das Kodieren von Videos mit dem veränderten Verfahren vergleichbar einfach ist wie mit der Referenzsoftware HM9.0, da das Verfahren auf dieser aufbaut und diese erweitert. Zusätzliche Bibliotheken aus OpenCV mussten jedoch für die Funktionalität des veränderten Verfahrens eingebunden werden. Durch die Verwendung der Referenzsoftware als Grundlage für das Gesichtserkennungsverfahren werden allerdings sowohl Vorteile, wie die einfache Steuerung der Einstellungen mit Konfigurationsdateien und Parametern oder die statistische Übersicht nach dem Kodieren, als auch die Nachteile, wie die nicht optimierte Kodierungseffizienz und daher zeitaufwendige Kodierung, übernommen. Insgesamt entstand dadurch jedoch der Vorteil, dass beide Verfahren gut miteinander verglichen werden konnten. Durch die kompetente Hilfe der Autoren konnten kleinere Fehler im Code behoben werden, die Kommunikation mit den Autoren diesbezüglich verlief sehr freundlich und gut.

Das Gesichtserkennungsverfahren konnte bei einer niedrigen Auflösung von 352 x 288 Pixeln keine deutlichen Verbesserungen der Videoqualität bewirken. Lediglich im Hintergrund konnten minimale Verschlechterungen der Qualität festgestellt werden, die allerdings nur bei genauem Hinsehen überhaupt auffallen. Bei Videos mit einer deutlich höheren Auflösung von 1920 x 1080 Pixeln konnten dagegen teilweise deutliche Unterschiede festgestellt werden, wobei es hier auch auf den Inhalt und die Qualität der Videos in Form von Bewegungen, Über-/Unterbelichtung und Größe des Gesichts im Video ankam. Bei guten Bedingungen konnte das wahrnehmungsbasierte Verfahren das Gesicht mit deutlich besserer Qualität in Form von Schärfe und Details kodieren, während dieses im üblichen HEVC-Verfahren unschärfer war. Auch hier wurden Bits im Hintergrund gespart, weshalb dieser beim veränderten Verfahren etwas unschärfer wirkte, was jedoch beabsichtigt war und bei der Qualitätswahrnehmung nicht besonders negativ auffällt.

Die Kodierungszeit war bei dem veränderten Verfahren auf Grund der Anpassung der maximalen Block-Unterteilungstiefe für alle Videos höchstens halb so lang wie die des üblichen HEVC-Verfahrens, wobei dieser zeitliche Vorteil bei der Beurteilung nicht zu hoch gewichtet werden sollte, da die Referenzsoftware ohnehin nicht zeitlich optimiert ist, sondern für Entwickler eher einfach zu verändern sein soll. Die Dateigröße war ungefähr gleich, beim veränderten Verfahren geringfügig kleiner.

Insgesamt bietet das Gesichtserkennungsverfahren eine gute Möglichkeit eine Eigenschaft der menschlichen Wahrnehmung auszunutzen, nämliche diese, dass der Mensch auf Gesichter besonders aufmerksam achtet. Möglicherweise gibt es jedoch noch zusätzliches Potential der Optimierung, sowohl durch die Verwendung einer aktuelleren Version der Referenzsoftware als auch durch Verbesserungen bei der Gesichtserkennung bei Videos mit nicht optimalen Bedingungen, z.B. Über- oder Unterbelichtung oder etwas verwackelten Gesichtern.

7 Zusammenfassung

Übliche Videokompressionsverfahren versuchen räumliche und zeitliche Redundanzen aus Videos zu entfernen und führen eine anschließende Kodierung der Daten durch um Videos zu komprimieren. Dabei sind die aktuell üblichen Verfahren H.264/AVC, H.265/HEVC und VP9. Die menschliche visuelle Wahrnehmung bietet ebenfalls verschiedene Eigenschaften, die für die Videokompression ausgenutzt werden können. Diese sind sowohl physikalisch bedingt, wie z.B. die geringere Farbwahrnehmung im Gegensatz zur Helligkeitswahrnehmung - die durch die Farbunterabtastung bereits bei üblichen Kompressionsverfahren ausgenutzt wird -, als auch durch die Verarbeitung der Daten im Gehirn bedingt, wie z.B. die selektive Aufmerksamkeit. Um diese Eigenschaften für die Videokompression auszunutzen, gibt es wahrnehmungsbasierte Kompressionsverfahren unterschiedlicher Kategorien.

Die Bewertung solcher Verfahren und der daraus resultierenden Videoqualität ist schwierig, da aktuell übliche Bewertungsmetriken wie PSNR oder SSIM stets das Gesamtbild bewerten und es bei solchen wahrnehmungsbasierten Verfahren üblich ist, dass eben nicht alle Bildbereiche qualitativ gleich gut sein müssen. Daher sind entweder subjektive Bewertungen mit einer größeren Anzahl an Probanden notwendig oder ein deutlich höherer manueller Aufwand für die Bewertungen und den Vergleich einzelner Bildbereiche miteinander. Zudem ist die menschliche Wahrnehmung noch nicht vollständig verstanden und wird weiterhin erforscht.

Es wurden Beispiele für wahrnehmungsbasierte Verfahren vorgestellt, die unterschiedliche Eigenschaften der menschlichen Wahrnehmung ausnutzen. Eines dieser Verfahren, das Gesichter erkennt und besser kodiert als die übrigen Bildbereiche, wurde von den Autoren zur Verfügung gestellt und konnte für die Kodierung von Videos verwendet werden. Dafür wurden drei Beispielvideos ausgewählt bzw. erstellt, die sich für die Anwendung des Verfahrens eigneten. Sie wurden sowohl mit der HEVC-Referenzsoftware HM9.0, als auch mit dem veränderten Verfahren, das auf dieser Software aufbaut, kodiert und miteinander verglichen.

Die Dateigröße konnte das Verfahren nur geringfügig verringern, dafür konnten teilweise deutliche Verbesserungen bei der Videoqualität erzielt werden. Bei niedriger Videoauflösung konnte das Gesichtserkennungsverfahren kaum Verbesserungen bei der Videoqualität erreichen, allerdings gelang es ihm bei Videos in Full-HD-Auflösung bei guten Bedingungen, wie einem ruhigen Bild und guter Belichtung deutliche qualitative Unterschiede zu bewirken. Das Gesicht eines Sprechers konnte in guter subjektiv wahrnehmbarer und objektiv mit PSNR und SSIM gemessener Qualität kodiert werden, während der Hintergrund etwas schlechter kodiert wurde. Beim gleichen Video wirkte beim üblichen HEVC-Verfahren das Gesicht unschärfer und weniger detailliert. Bei schlechteren Bedingungen, wie Unter- oder Überbelichtung oder relativ kleinem Gesicht mit viel Bewegung, konnte das Verfahren zwar nach wie vor Verbesserungen gegenüber des üblichen HEVC-Verfahrens erzielen, allerdings waren diese deutlich geringer als bei guten Bedingungen.

Insgesamt bieten wahrnehmungsbasierte Verfahren auch für die Zukunft gute Möglichkeiten Kompressionsverfahren zu verbessern und neben der aktuell üblichen Entfernung von räumlichen und zeitlichen Redundanzen auch Eigenschaften der menschlichen Wahrnehmung mit einzubeziehen, wodurch die resultierende wahrnehmbare Qualität für den Menschen erhöht werden kann. Das größte Potential hat dabei wohl das Saliency-Map-Verfahren, da in fast allen Videos auf bestimmte Bildbereiche geachtet wird, wodurch der Hintergrund unwichtiger wird. Das Verfahren bietet die Möglichkeit, diese Bereiche besser und den Hintergrund schlechter zu kodieren, wodurch insgesamt die wahrgenommene Qualität des Gesamtvideos steigt. Das vorgestellte Gesichtserkennungsverfahren bietet ebenfalls Potential zur Verbesserung von Kompressionsverfahren, der Einsatzbereich ist hier allerdings auf Videos mit menschlichen Gesichtern begrenzt, die nicht in allen Arten von Videos vorhanden sind.

I Literaturverzeichnis

Bossen F., Flynn D. und Sühring K.: *Software Manual HM9.0.* JCT-VC, 2012

Chen Z. und Li Y.: *Recent advances in perceptual H.265/HEVC video coding.* Aus IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), S.564-567, 2015

Chen Z., Lin W. und Ngan K. N.: *Perceptual Video Coding: Challenges and Approaches.* Aus IEEE International Conference on Multimedia and Expo (ICME), S.784-789, 2010

Cisco: *The Zettabyte Era: Trends and Analysis.* Cisco Systems, Inc., 2015

Doretto G., Chiuso A., Wu Y. N. und Soatto S.: *Dynamic Textures.* Aus International Journal of Computer Vision 51(2), S.91-109, 2003

Erhardt A.: *Einführung in die Digitale Bildverarbeitung, 1. Auflage.* Verlag Vieweg+Teubner, 2008

Fintrop S., Rome E. und Christensen H. I.: *Computational visual attention systems and their cognitive foundations: A survey.* Aus ACM Transactions on Applied Perception (TAP) (Volume 7, Issue 1, Article 6), S.6:1-6:39, 2009

Heise: *Heise Newsticker vom 01.11.2014, Youtube erlaubt Upload von HD-Videos mit 60 fps.* http://www.heise.de/newsticker/meldung/Youtube-erlaubt-Upload-von-HD-Videos-mit-60-fps-2440807.html, 2014. – letzter Zugriff 25.07.2016

Heise: *Heise Newsticker vom 05.02.2016, ZDF: Erste Ultra-HD-Produktion kommt im Mai ins Netz.* http://www.heise.de/newsticker/meldung/ZDF-Erste-Ultra-HD-Produktion-kommt-im-Mai-ins-Netz-3095301.html, 2016. – letzter Zugriff 21.07.2016

Henning P. A.: *Taschenbuch Multimedia, 4. Auflage.* Hanser Verlag, 2007

ITU-T: *ITU-T Rec. H.264 (05/2013) Advanced video coding for generic audiovisual services.* ITU-T, 2013

ITU-T: *ITU-T Rec. H.265 v3 (04/2015) High efficiency video coding.* ITU-T, 2015

Kalloniatis M. und Luu C.: *Light and Dark Adaptation.* http://webvision.med.utah.edu/book/part-viii-gabac-receptors/light-and-dark-adaptation/, 2007. – letzter Zugriff 14.07.2016

Kim J., Bae S. H. und Kim M.: *An HEVC-Compliant Perceptual Video Coding Scheme Based on JND Models for Variable Block-Sized Transform Kernels.* Aus IEEE Transactions on Circuits and Systems for Video Technology (Volume 25, Issue 11), S.1786-1800, 2015

Lee J. S. und Ebrahimi T.: *Perceptual Video Compression: A Survey.* Aus IEEE Journal of Selected Topics in Signal Processing Volume 6 Issue 6, S.684-697, 2012

Li Y., Liao W., Huang J., He D. und Chen Z.: *Saliency based perceptual HEVC.* Aus IEEE International Conference on Multimedia and Expo Workshops (ICMEW), S.1-5, 2014

Loomis J. und Wasson M.: *Microsoft Corporation: VC-1 Technical Overview.* https://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx, 2007. – letzter Zugriff 18.07.2016

Mukherjee D., Bankoski J., Grange A., Han J., Koleszar J., Wilkins P., Xu Y. und Bultje R.: *The latest open-source video codec VP9 - An overview and preliminary results.* Aus Picture Coding Symposium (PCS), S.390-393, 2013

OpenCV: *Über OpenCV.* http://opencv.org/about.html, 2016. – letzter Zugriff 16.07.2016

Ozer J.: *StreamingMedia: What Is VP9? / Alliance for Open Media.* http://www.streamingmedia.com/Articles/Editorial/-111334.aspx, 2016. – letzter Zugriff 15.08.2016

Ren Z., Chia L. T. und Rajan D.: *Video Saliency Detection with Robust Temporal Alignment and Local-Global Spatial Contrast.* Aus 2nd ACM International Conference on Multimedia Retrieval (ICMR), Artikel 47, 2012

Richardson I. E. G.: *H.264 and MPEG-4 video compression: video coding for next-generation multimedia.* Wiley, 2003

Saragih J. M., Lucey S. und Cohn J. F.: *Face Alignment through Subspace Constrained Mean-Shifts.* Aus IEEE 12th International Conference on Computer Vision, S.1034-1041, 2009

Sharabayko M. P. und Markov N. G.: *Contemporary video compression standards: H.265/HEVC, VP9, VP10, Daala.* Aus International Siberian Conference on Control and Communications (SIBCON), S.1-4, 2016

Sullivan G. J., Ohm J. R., Han W. J. und Wiegand T.: *Overview of the High Efficiency Video Coding (HEVC) Standard.* Aus IEEE Transactions on Circuits and Systems for Video Technology (Volume 22, Issue 12), S.1649-1668, 2012

Wiegand T., Sullivan G. J., Bjøntegaard G. und Luthra A.: *Overview of the H.264/AVC video coding standard.* Aus IEEE Transactions on Circuits and Systems for Video Technology (Volume 13, Issue 7), 2003

Wien M.: *High Efficiency Video Coding Coding Tools and Specification.* Springer Verlag Berlin Heidelberg, 2015

Wu H. R. und Rao K. R.: *Digital Video Image Quality and Perceptual Coding.* CRC Press, Inc., 2005

Xu M., Deng X., Li S. und Wang Z.: *Region-of-Interest Based Conversational HEVC Coding with Hierarchical Perception Model of Face.* Aus IEEE Journal of Selected Topics in Signal Processing (Volume 8, Issue 3), S.475-489, 2014

Xu M., Liang Y. und Wang Z.: *State-of-the-art Video Coding Approaches: A Survey.* Aus IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (IC-CI*CC), S.284-290, 2015

Zhang F. und Bull D. R.: *A Parametric Framework for Video Compression Using Region-Based Texture Models.* Aus IEEE Journal of Selected Topics in Signal Processing (Volume 5, Issue 7), S.1378-1392, 2011

Zhengrong X., Mei Y., Shuqing F. und Shengyang X.: *A New Saliency Based Video Coding Method with HEVC.* Aus International Conference on Intelligent Systems Research and Mechatronics Engineering (ISRME), S.672-679, 2015

II Anhang

II.1 Konfigurationsdateien der Beispielvideos

- „Foreman“:

```
BitstreamFile                 : Foreman_352x288_2997.hevc
ReconFile                     : Foreman_352x288_2997_out.yuv

FrameRate                     : 29.97
FrameSkip                     : 0
SourceWidth                   : 352
SourceHeight                  : 288
FramesToBeEncoded             : 300
```

- „Sprecher“:

```
BitstreamFile                 : Sprecher_1920x1080_15.hevc
ReconFile                     : Sprecher_1920x1080_15_out.yuv

FrameRate                     : 15
FrameSkip                     : 0
SourceWidth                   : 1920
SourceHeight                  : 1080
FramesToBeEncoded             : 60
```

- „Vortrag“:

```
BitstreamFile                 : Vortrag_1920x1080_25.hevc
ReconFile                     : Vortrag_1920x1080_25_out.yuv

FrameRate                     : 25
FrameSkip                     : 0
SourceWidth                   : 1920
SourceHeight                  : 1080
FramesToBeEncoded             : 207
```

II.2 Befehle zum Kodieren der Beispielvideos

- „Foreman“:

  TAppEncoder.exe -c Foreman.cfg -c encoder_lowdelay_P_main.cfg -i Foreman_352x288_2997.yuv

- „Sprecher“:

  TAppEncoder.exe -c Sprecher.cfg -c encoder_lowdelay_P_main.cfg -i Sprecher_1920x1080_15.yuv

- „Vortrag“:

  TAppEncoder.exe -c Vortrag.cfg -c encoder_lowdelay_P_main.cfg -i Vortrag_1920x1080_25.yuv

II.3 PSNR- und SSIM-Werte einzelner Frames aus „Foreman“

Für die Berechnung von PSNR und SSIM für einzelne Bildausschnitte (auch bei den folgenden beiden Videosequenzen) wurde das Bildverarbeitungsprogramm ImageJ mit den Plugins „Calculate SSIM Index“ von Gabriel Prieto Renieblas[1] und „SNR“ der Biomedical Image Group[2] verwendet.

In Frame 284 ist kein Gesicht mehr vorhanden, weshalb hier keine einzelne Betrachtung von Gesicht und Hintergrund vorgenommen werden konnte.

Frame 24	**Übliches Verf.**		**Gesichtserk.**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	41,83593094 dB	0,96996494444	39,56523365 dB	0,95544260498
Gesicht	49,57585655 dB	0,98635545397	47,87184069 dB	0,97949810292
Hintergrund	43,25515979 dB	0,96576654351	41,82916077 dB	0,95200017964

Frame 59	**Übliches Verf.**		**Gesichtserk.**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	41,23787832 dB	0,96768514042	38,75841642 dB	0,95242767692
Gesicht	48,47450926 dB	0,98366732100	46,34545543 dB	0,97230079218
Hintergrund	42,57807122 dB	0,96461242957	41,96428714 dB	0,95528893807

Frame 125	**Übliches Verf.**		**Gesichtserk.**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	41,34084018 dB	0,96827716015	43,50835136 dB	0,97724088673
Gesicht	52,72883504 dB	0,98204970109	54,22989764 dB	0,98819009142
Hintergrund	44,02423720 dB	0,96764983396	46,32538931 dB	0,97555925025

Frame 149	**Übliches Verf.**		**Gesichtserk.**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	41,45571581 dB	0,96659530918	41,56102873 dB	0,96524709253
Gesicht	49,08783323 dB	0,98235513201	49,44697702 dB	0,98282749287
Hintergrund	42,69462405 dB	0,96050915547	42,78647756 dB	0,95835552017

Frame 284	**Übliches Verf.**		**Gesichtserk.**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	39,15939097 dB	0,96863307078	37,75199133 dB	0,96195347963
Gesicht	-	-	-	-
Hintergrund	-	-	-	-

[1] https://imagej.nih.gov/ij/plugins/ssim-index.html
[2] http://bigwww.epfl.ch/sage/soft/snr/

II.4 PSNR- und SSIM-Werte einzelner Frames aus „Sprecher“

Frame 9	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	44,28182501 dB	0,97923384285	45,06808414 dB	0,98197927167
Gesicht	36,31037033 dB	0,95084547292	38,94733774 dB	0,96461253107
Hintergrund	44,31849810 dB	0,97439910276	44,85682217 dB	0,97672614158

Frame 21	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	44,32412181 dB	0,97971943897	45,26681783 dB	0,98395258277
Gesicht	36,83568137 dB	0,94965944830	39,51145955 dB	0,96612806309
Hintergrund	44,47224328 dB	0,97455187085	45,21041627 dB	0,97867345803

Frame 38	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	44,45218962 dB	0,98054317032	44,84470141 dB	0,98220585590
Gesicht	37,42127723 dB	0,95734672250	38,96789334 dB	0,96435069733
Hintergrund	44,65170647 dB	0,97530663634	44,90681398 dB	0,97626818885

Frame 47	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	44,33885636 dB	0,98077766645	44,76966890 dB	0,98190353823
Gesicht	35,49992931 dB	0,94768997073	37,21904527 dB	0,95747689826
Hintergrund	44,53638866 dB	0,97634902910	44,75652303 dB	0,97685385951

Frame 59	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	44,43126202 dB	0,98090308245	44,75307163 dB	0,98196096423
Gesicht	36,79188245 dB	0,95192755551	38,39066436 dB	0,96089663020
Hintergrund	44,77043744 dB	0,97615510121	44,95123076 dB	0,97658945428

II.5 PSNR- und SSIM-Werte einzelner Frames aus „Vortrag“

Frame 66	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	42,84917428 dB	0,97348232191	44,56199849 dB	0,97884580307
Gesicht	45,03857868 dB	0,97207234772	44,51970978 dB	0,96673796447
Hintergrund	41,76307705 dB	0,96640334072	43,41762268 dB	0,97256895363

Frame 89	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	41,18343734 dB	0,96305563987	43,34029146 dB	0,97568135005
Gesicht	41,72653813 dB	0,95331117961	41,94468557 dB	0,95758266217
Hintergrund	40,73515730 dB	0,96068607926	42,85994562 dB	0,97368643837

Frame 134	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	38,88086725 dB	0,95843624980	43,78393740 dB	0,97792996345
Gesicht	36,87818193 dB	0,94917773736	38,19342113 dB	0,95824376895
Hintergrund	37,40490924 dB	0,95482643083	42,81041402 dB	0,97593677994

Frame 166	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	36,89428852 dB	0,95986831479	41,01588382 dB	0,98025028898
Gesicht	42,98952209 dB	0,96580820375	43,16220876 dB	0,96841001973
Hintergrund	36,43761204 dB	0,95536291548	40,61014316 dB	0,97507692328

Frame 198	Übliches Verf.		Gesichtserk.	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Gesamt	38,11282444 dB	0,96972187250	40,42080701 dB	0,98294373396
Gesicht	41,00589070 dB	0,97547234025	40,13751185 dB	0,97196740065
Hintergrund	37,37726866 dB	0,96805588770	39,96546477 dB	0,98210455481