

BACHELORARBEIT

Vom klassischen Broadcast zum modernen Streaming: Eine Beurteilung der aktuellen Videocodecs

vorgelegt am 31. August 2023
Dennis Köckritz

Erstprüferin: Prof. Dr. Marco Grimm
Zweitprüfer: Nathalie Mai

**HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG**

Department Medientechnik
Finkenau 35
20081 Hamburg

Zusammenfassung

Die vorliegende Bachelorarbeit "Vom klassischen Broadcast zum modernen Streaming: Eine Beurteilung der aktuellen Videocodecs" widmet sich der Untersuchung und Bewertung verschiedener Videocodecs im Kontext der Evolution von Rundfunk zu Streaming-Diensten. Die Arbeit analysiert die Leistung und Qualität der Codecs H.264, H.265, AV1, VP9, MPEG-2 und MPEG-4, um Erkenntnisse darüber zu gewinnen, wie diese Codecs die Anforderungen moderner Videodistributionstechnologien erfüllen. Die Ergebnisse zeigen bei Codecs wie H.265, MPEG-4 oder AV1 deutliche Verbesserungen gegenüber ihren Vorgängern. Sowohl in der Komprimierungseffizienz als auch in der Videoqualität, die mittels des PSNR-Wertes gemessen wird, werden deutliche Fortschritte aufgezeigt. Die Bachelorarbeit liefert damit Einblicke in die Auswahl und Nutzung von Videocodecs im Zeitalter des modernen Streamings.

Abstract

This bachelor thesis titled "From Classical Broadcast to Modern Streaming: An Assessment of Current Video Codecs" is dedicated to the investigation and evaluation of various video codecs in the context of the evolution from broadcasting to streaming services. The paper analyzes the performance and quality of the codecs H.264, H.265, AV1, VP9, MPEG-2, and MPEG-4 to gain insights into how these codecs meet the requirements of modern video distribution technologies. The findings reveal significant improvements in codecs such as H.265, MPEG-4, or AV1 compared to their predecessors. Substantial progress is shown both in terms of compression efficiency and video quality, measured using the PSNR metric. As such, the bachelor's thesis provides insights into the selection and utilization of video codecs in the era of modern streaming.

Inhaltsverzeichnis

Abkürzungsverzeichnis	VI
Abbildungsverzeichnis	VI
Tabellenverzeichnis.....	VII
1 Einleitung.....	8
1.1 Hintergrund und Motivation.....	8
1.2 Zielsetzung der Arbeit	9
1.3 Aufbau der Arbeit	9
2 Grundlagen.....	10
2.1 Klassischer Broadcast: Entwicklung und Merkmale.....	10
2.2 Modemes Streaming: Bedeutung und Herausforderungen	11
2.3 Videocodecs: Funktionen und Klassifizierung.....	12
3 Entwicklung der Videocodecs.....	12
3.1 Frühe Videocodecs: Überblick und Eigenschaften	12
3.1.1 H.261	13
3.1.2 MPEG-1.....	13
3.1.3 MPEG-2.....	13
3.1.4 MPEG-4.....	14
3.1.5 H.263.....	14
3.1.6 H.264.....	15
3.1.7 Eigenschaften früher Videocodecs.....	15
3.2 Fortschritte in der Videocodierungstechnologie.....	16
3.2.1 Fortschritte in der Komprimierungseffizienz	16
3.2.2 Unterstützung höherer Auflösungen und Farbtiefe.....	16
3.2.3 Effiziente Bitratenkontrolle.....	16
3.3 Aktuelle Videocodecs: Vorstellung und Vergleich.....	17
3.3.1 H.265 (HEVC).....	17
3.3.2 VP9.....	17
3.3.3 AV1.....	18

4	Bewertungskriterien der Videocodecs.....	19
4.1	Videoqualität: Subjektive und Objektive Bewertungsmethoden.....	19
4.1.1	Peak Signal to Noise Ratio (PSNR).....	19
4.1.2	Visuelle Suche nach Blockartefakten.....	20
4.2	Effizienz der Datenkomprimierung.....	21
5	Experiment und Experteninterview.....	21
5.1	Einleitung und Zielsetzung des Experiments.....	21
5.2	Versuchsaufbau.....	22
5.2.1	Produktion von Videos mit der Blackmagic Ursa Broadcast G2.....	22
5.2.2	Erstellung einer Animation in Adobe After Effects.....	24
5.3	Vorstellung der Software FFmpeg.....	28
5.3.1	Funktionen von FFmpeg.....	29
5.3.2	Nutzung von FFmpeg im Experiment.....	29
5.4	Durchführung von Kodierung und Dekodierung.....	29
5.4.1	MPEG-2.....	30
5.4.2	MPEG-4.....	31
5.4.3	H.264.....	32
5.4.4	H.265.....	32
5.4.5	VP9.....	33
5.4.6	AV1.....	34
5.4.7	Ermittlung des PSNR-Wertes.....	36
5.5	Durchführung eines Experteninterviews.....	36
6	Ergebnisse und Diskussion.....	38
6.1	Evaluierung der Videocodecs anhand festgelegter Kriterien.....	38
6.2	Vergleich einzelner Codecs.....	39
6.2.1	Vergleich von H.264 und H.265 (HEVC).....	39
6.2.2	Vergleich von MPEG-2 und MPEG-4.....	43
6.2.3	Vergleich von AV1 und VP9.....	47
7	Zusammenfassung und Schlussfolgerungen.....	51
7.1	Zusammenfassung der wichtigsten Ergebnisse.....	51

7.2	Analyse der Stärken und Schwächen der einzelnen Codecs	53
7.3	Erfüllung der Zielsetzung der Arbeit.....	54
	Literaturverzeichnis.....	56
	Anhang.....	59
	Eigenständigkeitserklärung.....	60

Abkürzungsverzeichnis

ABR.....	<i>Adaptive Bitrate Streaming</i>
AOMedia	<i>Alliance for Open Media</i>
AV1.....	<i>AOMedia Video 1</i>
AVC.....	<i>Advanced Video Coding</i>
CBR.....	<i>Constant Bit Rate</i>
CD.....	<i>Compact-Disc</i>
Codec.....	<i>Compression/Decompression</i>
DCT.....	<i>diskrete Kosinus-Transformation</i>
DTV.....	<i>Digital Television</i>
DVB	<i>Digital Video Broadcasting</i>
HDTV.....	<i>High-Definition-Television</i>
HEVC.....	<i>High Efficiency Video Coding</i>
IEC.....	<i>International Electrotechnical Commission</i>
ISO.....	<i>International Organization for Standardization</i>
ITU-T.....	<i>International Telecommunication Union - Telecommunication Standardization Sector</i>
JVT.....	<i>Joint Video Team</i>
MPEG-1.....	<i>Moving Picture Experts Group - Part 1</i>
PAL	<i>Phase-Alternating-Line</i>
PSNR.....	<i>peak signal-to-noise ratio</i>
VBR.....	<i>Variable Bit Rate</i>
VCD	<i>Video-Compact-Disc</i>
VCEG.....	<i>Video Coding Experts Group</i>
VCR.....	<i>digitale Videokassettenrekorder</i>
VMAF.....	<i>Video Multi-Method Assessment Fusion</i>

Abbildungsverzeichnis

Abbildung 1: Entwicklung der Videokodierstandards (SCHMIDT 2021).....	13
Abbildung 2: Versuchsaufbau.....	22
Abbildung 3: Betrachtung der Rohdaten mithilfe der Software MediaArea.....	23
Abbildung 4: Exporteinstellungen in Premiere Pro	24
Abbildung 5: Effekt „CC Snowfall“ in After Effects.....	25
Abbildung 6: „CC Snowfall“-Parameter in After Effects.....	26
Abbildung 7: Erstellung des Konfettis in After Effects.....	27
Abbildung 8: Anpassung des Maskenpfads in After Effects	27
Abbildung 9: Anwendung des Effekts "CC Pixel Molly" in After Effects.....	28
Abbildung 10: Parameter des MPEG-2 Videos nach erfolgter Kodierung.....	31
Abbildung 11: Parameter des H.265 Videos nach erfolgter Kodierung.....	33
Abbildung 12: FFmpeg nach erfolgreicher VP9-Kodierung.....	34
Abbildung 13: FFmpeg nach erfolgreicher AV1-Kodierung.....	35

Abbildung 14: Ergebnis der PSNR-Werte.....	36
Abbildung 15: H.264 vs. H.265 (animiert).....	40
Abbildung 16: H.264 vs. H.265 ohne Bewegungen.....	40
Abbildung 17: H.264 vs. H.265 mit Bewegungen.....	41
Abbildung 18:Vergrößerung von H.264.....	41
Abbildung 19:Vergrößerung von H.265.....	42
Abbildung 20: H.264 vs. H.265 (min-Werte).....	42
Abbildung 21: MPEG-2 vs. MPEG-4 (animiert).....	44
Abbildung 22: MPEG-2 vs. MPEG-4 ohne Bewegungen.....	44
Abbildung 23: MPEG-2 vs. MPEG-4 mit Bewegungen.....	45
Abbildung 24: Vergrößerung von MPEG-2.....	45
Abbildung 25: Vergrößerung von MPEG-4.....	46
Abbildung 26: MPEG-2 vs. MPEG-4 (min-Werte).....	46
Abbildung 27: AV1 vs. VP9 (animiert).....	48
Abbildung 28: AV1 vs. VP9 ohne Bewegungen.....	48
Abbildung 29: AV1 vs. VP9 mit Bewegungen.....	49
Abbildung 30: Vergrößerung von AV1.....	49
Abbildung 31: Vergrößerung von VP9.....	50
Abbildung 32: AV1 vs. VP9 (min-Werte).....	50
Abbildung 33: Vergleich aller Codecs (animiert).....	52
Abbildung 34: Vergleich aller Codecs (min-Werte).....	53

Tabellenverzeichnis

Tabelle 1: Vergleich der Datengröße und der Länge der Codierung (H.264 vs. H.265).....	43
Tabelle 2: Vergleich der Datengröße und der Länge der Codierung (MPEG-2 vs. MPEG-4).....	47
Tabelle 3: Vergleich der Datengröße und der Länge der Codierung (AV1 vs. VP9).....	51
Tabelle 4: Vergleich aller Codecs (sortiert nach abfallender Bitrate).....	52

1 Einleitung

1.1 Hintergrund und Motivation

In den letzten Jahrzehnten hat sich die Art und Weise, wie audiovisuelle Inhalte konsumiert werden, grundlegend verändert. Während der klassische Broadcast als dominierende Übertragungsmethode diente, hat das moderne Streaming zunehmend an Bedeutung gewonnen. Die Fortschritte in der Computertechnologie, Kompressionstechnologie und der Ausbau von schnellen Netzwerken haben es ermöglicht, Echtzeit-Multimediaservices über das Internet bereitzustellen. Echtzeit-Multimedia, wie der Begriff bereits suggeriert, unterliegt zeitlichen Einschränkungen. Beispielsweise müssen Audio- und Videodaten kontinuierlich abgespielt werden. Falls die Daten nicht rechtzeitig eintreffen, wird der Wiedergabeprozess unterbrochen, was für das menschliche Gehör und die Augen unangenehm ist (vgl. Da-peng Wu et al. 2001, S. 282).

Die Verschiebung vom Broadcast zum Streaming Verschiebung hat zu einer Vielzahl von technologischen Innovationen geführt, insbesondere im Bereich der Videocodecs, die eine entscheidende Rolle bei der Komprimierung und Übertragung von Videoinhalten spielen.

Video „Compression/Decompression“, abgekürzt Codec, ist ein Algorithmus oder eine Sammlung von Algorithmen, die dazu dienen, die Menge an Daten in einem Videostream zu reduzieren, ohne dabei eine unakzeptable Verschlechterung der visuellen Qualität zu verursachen. Die Klassifizierung von Videocodecs basiert auf verschiedenen Aspekten ihrer Funktionalität, Effizienz und Anwendungsbereiche (vgl. Shahriar Akramullah, S.3).

Im Rahmen dieser Arbeit wird die Entwicklungsgeschichte der Videocodecs analysiert und die Funktionsweise und Merkmale aktueller Videocodecs eingehend betrachtet. Dabei liegt ein besonderer Fokus auf der Beurteilung ihrer Leistung in Bezug auf Videoqualität, die mithilfe von subjektiven und objektiven Bewertungsmethoden ermittelt wird, sowie der Effizienz der Datenkomprimierung unter Berücksichtigung von Datengröße und Dauer des Kodierungsvorgangs. Hieraus lässt sich folgende Forschungsfrage formulieren:

Wie unterscheiden sich die aktuellen Videocodecs hinsichtlich ihrer Komprimierungseffizienz, Videoqualität und Eignung für verschiedene Streaming-Szenarien?

Außerdem soll der Übergang vom Broadcast zum Streaming näher beleuchtet werden. Hierfür wird auf die Geschichte der einzelnen Codecs eingegangen, um technische Meilensteine der Kodierverfahren hervorzuheben. Hieraus lässt sich eine zweite eine Forschungsfrage ableiten:

Wie haben die Entwicklungen im Bereich der Videocodecs den Wandel vom klassischen Broadcast zum modernen Streaming beeinflusst?

Um einen zusätzlichen Blick auf die aktuellen Geschehnisse der Streaming-Branche zu erlangen, wird ein Experteninterview mit dem Geschäftsführer der Pontius Media GmbH durchgeführt.

1.2 Zielsetzung der Arbeit

Das Ziel dieser Arbeit ist es, einen umfassenden Blick auf den Wandel vom klassischen Broadcast zum modernen Streaming zu werfen und dabei insbesondere die bekanntesten Videocodecs zu untersuchen. Die Wahl des richtigen Videocodecs ist von entscheidender Bedeutung, um eine optimale Videoqualität bei effizienter Datenkomprimierung zu gewährleisten.

Die Umsetzung und Auswertung von praktischen Experimenten sowie relevanten Forschungsarbeiten sollen außerdem zu fundierten Erkenntnissen beitragen und es ermöglichen, die oben gestellten Forschungsfragen zu beantworten. Hierfür ist ein strukturiertes Vorgehen vorgesehen, welches im Folgenden erläutert wird.

1.3 Aufbau der Arbeit

Um die eben genannten Ziele zu erreichen, wird die Arbeit in folgende Kapitel unterteilt:

2. *Grundlagen*: Der erste Schritt besteht darin, die grundlegenden Konzepte des klassischen Broadcast und des modernen Streamings zu erläutern. Hier werden historische Entwicklungen und technologische Hintergründe erörtert, um so einen umfassenden Überblick zu verschaffen.

3. *Entwicklung der Videocodecs*: In diesem Abschnitt erfolgt eine Auseinandersetzung mit der Entwicklung der einzelnen Videocodecs von den frühen Standards bis zu den aktuellen Technologien. Es werden die technischen Fortschritte, Innovationen und Meilensteine auf dem Gebiet der Videokompression beleuchtet.

4. *Bewertungskriterien der Videocodecs*: Die Analyse und Bewertung der aktuellen Videocodecs erfordern klare Kriterien. Daher werden in diesem Abschnitt die relevanten Bewertungskriterien definiert, die bei der Beurteilung der Codecs im nachfolgenden Experiment eine Rolle spielen werden. Hierzu zählen Aspekte wie Komprimierungseffizienz sowie subjektive und objektive Bewertungsmethoden.

5. *Experiment und Experteninterview*: Um empirische Erkenntnisse zu gewinnen, wird ein Experiment durchgeführt, in dem ausgewählte Videocodecs anhand der definierten Kriterien verglichen werden. Hierfür wird geeignetes Videomaterial produziert. Zusätzlich wird ein Interview mit einem Experten aus der Video- und Streaming-Branche geführt, um praktische Perspektiven und Einblicke zu erhalten.

6. *Ergebnisse und Diskussion*: In diesem Abschnitt werden die Ergebnisse des Experiments präsentiert. Die Leistung der verschiedenen Videocodecs in Bezug auf die festgelegten Bewertungskriterien wird analysiert und dargestellt.

7. *Zusammenfassung und Schlussfolgerungen*: Die wichtigsten Ergebnisse werden zusammengefasst und interpretiert. Die Forschungsfragen werden beantwortet und es werden mögliche Gründe für die Unterschiede in den Leistungen der Videocodecs erörtert und Zusammenhänge zu den theoretischen Grundlagen hergestellt.

2 Grundlagen

2.1 Klassischer Broadcast: Entwicklung und Merkmale

Der klassische Broadcast hat im Laufe der Zeit eine bedeutende Entwicklung durchlaufen. Ursprünglich auf Rundfunknetze wie Radio und Fernsehen beschränkt, basierte er auf einer linearen Übertragung, bei der Inhalte von einem zentralen Sender an ein breites Publikum gesendet wurden. Dieses Modell war durch festgelegte Sendezeiten und Kanäle gekennzeichnet, wodurch die Zuschauer ihre Programme zu bestimmten Zeiten und auf vordefinierten Kanälen konsumieren mussten (vgl. Bhattacharya/Mallick 2010, S.1).

Broadcast ist eine äußerst effiziente Kommunikationsform für populäre Inhalte, da sie oft in der Lage ist, diese effizient an alle Empfänger gleichzeitig zu übermitteln. Ein wichtiger Aspekt der Broadcast-Kommunikation besteht darin, dass das System so konzipiert sein muss, dass es jedem beabsichtigten Empfänger das benötigte Signal liefert. Dies ist von besonderer Bedeutung, da unterschiedliche Empfänger auch unterschiedliche Kanalcharakteristiken erfahren können. Als Folge ist das System oft für den ungünstigsten Kanalzustand ausgelegt. Ein Beispiel hierfür ist die digitale Fernsehübertragung, bei der Quellencodierung und Kanalcodierung darauf abzielen, eine ausreichende Empfangsqualität für Empfänger am Rand des erforderlichen Empfangsbereichs sicherzustellen. Dabei wird jedoch zugunsten der Qualität bei Empfängern in Gebieten mit höherer Empfangsqualität, wie etwa im Stadtzentrum, Kompromisse gemacht. Außerdem ist aufgrund der großen Anzahl von Empfängern ein Rückkanal vom Empfänger zum Sender in der Regel nicht realisierbar, was die Anpassungsfähigkeit des Systems begrenzt (vgl. John G. Apostolopoulos et al. 2002, S.2).

Im Laufe der Jahre wurden verschiedene technologische Fortschritte eingeführt, um den klassischen Broadcast zu verbessern. Dazu gehört die Einführung des Farbfernsehens, die Entwicklung fortschrittlicher Übertragungsstandards wie Digital Video Broadcasting (DVB) und die Digitalisierung der Inhalte. Digital-Television (DTV) beschreibt die standardisierte Übermittlung von digitalen Inhalten an Haushalte, wodurch das bisherige analoge Phase-Alternating-Line-Fernsehsystem oder auch PAL-Fernsehsystem ersetzt wird. Im Zuge der Entwicklung des digitalen Fernsehens spielen insbesondere die Aspekte der Quellencodierung, Kanalcodierung und Übertragungswege eine entscheidende Rolle. Die Quellencodierung befasst sich mit der Definition der zu übertragenden Signalform, während die Kanalcodierung Verfahren zur Fehlerkorrektur und -schutz umfasst (vgl. Schmidt 2021, S.312).

Diese und weitere Entwicklungen ermöglichten eine verbesserte Signalqualität, eine größere Anzahl von Kanälen und eine erweiterte Darstellungsmöglichkeit von Inhalten. Gleichzeitig führte die Verbreitung des Internets und die fortschreitende Digitalisierung zu einer weiteren Transformation des Broadcast, indem sie neue Möglichkeiten für die Übertragung und den Konsum von audiovisuellen Inhalten schuf (vgl. Bhattacharya/Mallick 2010, S.4).

Der klassische Broadcast bleibt in einigen Bereichen weiterhin relevant, doch der Übergang zum modernen Streaming hat die Art und Weise, wie Inhalte übertragen und konsumiert werden, erheblich verändert. Das moderne Streaming basiert auf flexibleren, interaktiven und personalisierten Ansätzen, die es den Nutzern ermöglichen, Inhalte nach ihren eigenen Vorlieben und Bedürfnissen zu konsumieren (vgl. Schmidt 2021, S.745).

Die Entwicklung des klassischen Broadcast ist ein Spiegelbild der technologischen Fortschritte und der sich verändernden Bedürfnisse der Zuschauer. Sie hat den Weg für neue Formen der Medienübertragung geebnet und den Medienlandschaften neue Möglichkeiten eröffnet. Das so ermöglichte, heutige Streaming soll im nachfolgenden Kapitel hinsichtlich seiner Bedeutung aber auch der Herausforderungen näher betrachtet werden.

2.2 Modernes Streaming: Bedeutung und Herausforderungen

„Seit seiner Einführung Anfang der 1990er Jahre hat das Konzept des Streaming-Medien eine dramatische Entwicklung und Transformation von einer neuartigen Technologie zu einer der gängigsten Methoden, wie Menschen das Internet heute nutzen, durchlaufen.“ (Gregory J. Conklin et al. 2001, S.1). Das moderne Streaming ist heute zu einer der vorherrschenden Arten geworden, wie Menschen Medieninhalte konsumieren. Es ermöglicht nahezu Echtzeitübertragung von Audio- und Videodateien über das Internet, wodurch Nutzer Inhalte direkt auf ihren Endgeräten abspielen können, ohne sie vorher vollständig herunterladen zu müssen. Dieser Wandel hat das Medienverhalten und die Zuschauererwartungen stark beeinflusst. Streaming bietet nicht nur die Möglichkeit, Inhalte in hoher Qualität zu genießen, sondern ermöglicht auch personalisierte Erfahrungen und Interaktionen in Echtzeit. Mit der wachsenden Beliebtheit des Streamings sind jedoch auch verschiedene technische, qualitätsbezogene und infrastrukturelle Herausforderungen aufgetreten. So kommen mit hochauflösenden Inhalten wie 4K oder 8K gleichzeitig auch Anforderungen an eine hohe Bandbreite. Ebenfalls kann die Vielzahl von vorhandenen Codecs und deren unterschiedlichen Eigenschaften die Auswahl des richtigen Codecs für erschweren. Die Unterstützung verschiedener Codecs auf verschiedenen Geräten und Plattformen kann zusätzlich zu Kompatibilitätsproblemen führen. Die Sicherstellung einer konsistenten Videoqualität über verschiedene Geräte, Bildschirmgrößen und Netzwerkbedingungen hinweg ist ebenfalls eine Herausforderung. Schwankungen in der Bandbreite und Verbindungsqualität können zu Artefakten, Pufferproblemen und Qualitätsverlust führen (vgl. Sammoud et al. 2017, S.1-3).

Für das Bewältigen dieser Herausforderungen wurden in den vergangenen Jahrzehnten eine Reihe Codecs entwickelt, die in ihren Funktionen und Klassifizierungen unterschiedlich sein können.

2.3 Videocodecs: Funktionen und Klassifizierung

Videocodecs können anhand ihrer Funktionsweise in zwei Hauptkategorien unterteilt werden: Verlustfreie und verlustbehaftete Codecs. Verlustfreie Codecs zielen darauf ab, Videodaten ohne jeglichen Informationsverlust zu komprimieren. Diese werden jedoch selten in der Praxis verwendet, da sie in der Regel geringere Komprimierungsraten bieten und nicht in der Lage sind, signifikante Reduzierungen der Dateigröße zu erreichen. Verlustbehaftete Codecs hingegen dominieren die Medienindustrie, da sie durch die Entfernung von weniger wahrnehmbaren Informationen eine erhebliche Kompression erzielen (vgl. Shahriar Akramullah, S.3).

Die Klassifizierung von Videocodecs kann auch anhand ihrer Kodierungsstrategien erfolgen, wie beispielsweise der Verwendung von Blocktransformationen, Vorhersage- und Entropiekodierungstechniken. Hierbei werden lineare Transformationen, wie die diskrete Kosinus-Transformation (DCT), zur Darstellung von Bildblöcken verwendet, während Vorhersage- und Entropiekodierungsmethoden zur Reduzierung redundanter Informationen und zur effizienten Darstellung von Videodaten beitragen (vgl. Shahriar Akramullah, S. 39, 40).

Weiterhin werden Videocodecs aufgrund ihrer Fähigkeit zur Unterstützung verschiedener Auflösungen und Farbformate kategorisiert. Hierbei unterscheiden sich Codecs hinsichtlich ihrer Skalierbarkeit und Flexibilität, um den Anforderungen moderner Anwendungen gerecht zu werden.

Zusammenfassend gesehen sind Videocodecs essenzielle Komponenten für die moderne Medienverarbeitung und bieten eine breite Palette von Funktionalitäten und Anwendungsbereichen. Ihre kontinuierliche Weiterentwicklung trägt dazu bei, die Herausforderungen der heutigen digitalen Medienlandschaft zu bewältigen und qualitativ hochwertige Videodienste für eine breite Palette von Nutzern zu gänglich zu machen.

3 Entwicklung der Videocodecs

Um die Entwicklung der Videocodecs besser nachvollziehen zu können, werden in diesem Kapitel frühere und aktuelle Codecs vorgestellt. Außerdem beleuchtet das Kapitel wichtige Eigenschaften und Funktionen, die im Laufe der Zeit entwickelt wurden.

3.1 Frühe Videocodecs: Überblick und Eigenschaften

Dieser Abschnitt widmet sich frühen Videocodecs, die in den Anfängen der Videotechnologie entwickelt wurden. Diese Videocodecs legten den Grundstein für die spätere Entwicklung und dienten als Wegbereiter für modernere Kompressionsverfahren.

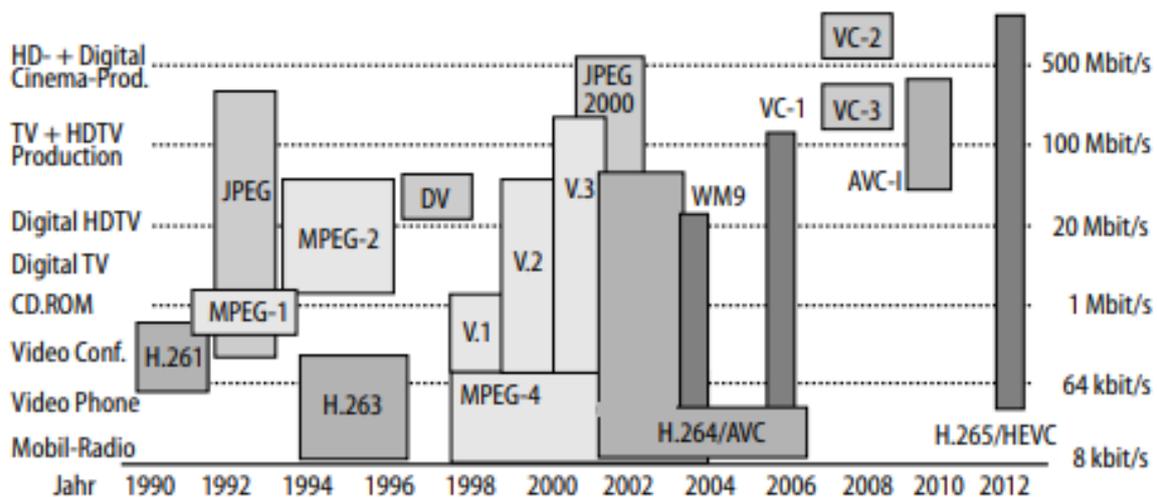


Abbildung 1: Entwicklung der Videokodierstandards (SCHMIDT 2021)

Wie aus Abbildung 1 zu entnehmen, stieg seit Beginn der Entwicklung von Videocodern die benötigte Bandbreite enorm. Während frühere Technologien nur wenige Megabyte benötigten, sind heutige Coders auf die Übertragung von bis zu 500 Megabyte pro Sekunde ausgelegt. Dies wurde ermöglicht durch eine lange Entwicklungsgeschichte, die mit dem Videocodern H.261 ihren Anfang nahm.

3.1.1 H.261

Einer der frühesten Videocodern war H.261, wurde bereits in den späten 1980er Jahren vom ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) standardisiert. H.261 war einer der ersten verlustbehafteten Videocodern, der eine effiziente Kompression von Videodaten ermöglichte. Er basierte auf dem sogenannten blockbasierten Bewegungsausgleich, bei dem ein Bild in Blöcke unterteilt und die Bewegung dieser Blöcke zwischen aufeinanderfolgenden Frames analysiert und codiert wurde (vgl. Schmidt 2021, S.253).

3.1.2 MPEG-1

MPEG-1 (Moving Picture Experts Group - Part 1) wurde Anfang der 1990er Jahre entwickelt und war einer der ersten Videocodern, der in kommerziellen Produkten wie CDs und VCDs (Video-CDs) verwendet wurde. Dieser Codern ermöglichte die Komprimierung von Videodaten bei relativ geringen Datenraten von bis zu 1,5 Mbit/s und erlaubte die Verbreitung von Videoinhalten über das damals aufkommende Internet (vgl. Schmidt 2021, S.221).

3.1.3 MPEG-2

Der Nachfolger von MPEG-1 wurde als Norm für die generische Kodierung von bewegten Bildern und dazugehörigem Audio festgelegt. Die Spezifikation dieses Standards erfolgte durch einen gemeinsamen technischen Ausschuss der International Organization for Standardization (ISO), der International

Electrotechnical Commission (IEC) sowie der ITU-T und wurde im Jahr 1993 sowohl als internationaler ISO/IEC-Standard 13818 als auch als ITU-T Empfehlung H.262 ratifiziert.

Mit dem Ziel, bestehende Herausforderungen von MPEG-1 zu bewältigen, lag der Fokus der Standardisierungsarbeit von MPEG-2 auch auf Aspekten wie die Erweiterung der Audio-Kompressionskanäle und der Farbabtastung.

MPEG-2 wurde für verschiedenste Anwendungen bei einer Bitrate von 2 Megabytes pro Sekunde (Mbps) oder höher entwickelt, wobei die Qualität von guter NTSC-Qualität bis hin zu High-Definition-Television (HDTV) reicht. Neben der Verwendung als Format für digitale Fernsehsignale in terrestrischen, Kabel- und Satelliten-TV-Systemen findet es auch Anwendung in digitalen Videokassettenrekordern (VCR), digitalen Videodiscs (DVD) und Ähnlichem. Als generischer Standard, der ein breites Anwendungsspektrum von 2 Mbps bis 40 Mbps abdeckt, zielt MPEG-2 auf ein Kompressionsverhältnis zwischen 30 und 40 ab (vgl. Shahriar Akramullah, S. 62).

3.1.4 MPEG-4

Der MPEG-4-Standard stellt einen Meilenstein in der Videocodierung dar, der 1998 von ISO und ITU entwickelt wurde. Die Entwicklung soll den Nutzern ein neues Maß an Interaktionen mit visuellen Inhalten bieten. Anwendungsbereiche reichen von digitalem Fernsehen über Streaming-Videos bis hin zu mobilen Multimedia-Anwendungen und Spielen. Der natürliche MPEG-4-Video-Standard besteht aus einer Sammlung von Tools, die diese Anwendungsbereiche unterstützen. Der Standard bietet Werkzeuge für die Formcodierung, Bewegungsschätzung und -kompensation, Texturcodierung, Fehlerresilienz, Sprite-Codierung und Skalierbarkeit (vgl. Ebrahimi/Horne 2000, S.1).

3.1.5 H.263

H.263 war eine Weiterentwicklung von H.261 und wurde Mitte der 1990er Jahre eingeführt. Er optimierte die Komprimierung von Videodaten für geringe Datenraten und wurde zum beliebten Codec für Videokonferenzen und Videoübertragungen über das Internet. H.263, standardisiert durch die ITU-T, zielt auf die Videocodierung mit niedrigen Bitraten ab, ohne jedoch eine Begrenzung der Videobitrate festzulegen; solche Begrenzungen werden von den Endgeräten oder dem Netzwerk vorgegeben. Das Ziel von H.263 besteht darin, eine deutlich bessere Bildqualität als sein Vorgänger H.261 zu bieten. Konzeptuell betrachtet ist H.263 netzwerkunabhängig und kann für eine Vielzahl von Anwendungen verwendet werden, wobei seine Hauptanwendungen visuelle Telekommunikation und Multimedia in niedrigbitratigen Netzwerken wie PSTN, ISDN und drahtlosen Netzwerken sind. Zu den wichtigen Überlegungen für H.263 gehören geringer Overhead, geringe Komplexität mit niedrigen Kosten, Interoperabilität mit bestehenden Videokommunikationsstandards (z.B. H.261, H.320), Robustheit gegenüber Kanalfehlern und Qualitätssicherungsparameter (QoS). Basierend auf diesen Überlegungen wurde

ein effizienter Algorithmus entwickelt, der den Herstellern die Flexibilität bietet, Kompromisse zwischen Bildqualität und Komplexität zu treffen. Im Vergleich zu H.261 bietet es bei weniger als der Hälfte der Bitrate dieselbe subjektive Bildqualität (vgl. Shahriar Akramullah, S. 67).

3.1.6 H.264

Der H.264-Standard, auch bekannt als MPEG-4 Part 10 oder Advanced Video Coding (AVC), wurde entwickelt und verabschiedet durch die langjährige Zusammenarbeit des Joint Video Teams (JVT), der ITU-T, der Video Coding Experts Group (VCEG) und der ISO/IEC MPEG. Die Entwicklung begann bereits in den späten 1990er Jahren. Der Standard wurde im Jahr 2003 verabschiedet. H.264 wurde für den Einsatz in Hochauflösungssystemen wie Blu-ray, HDTV und HD DVD entwickelt und bietet im Vergleich zu MPEG-2 und MPEG-4 eine bessere Qualität bei geringerem Speicherplatzbedarf (vgl. Richardson, S.81-83).

Im Gegensatz zu MPEG-4 konzentrierte man sich bei der Entwicklung von H.264/AVC auf die Steigerung der Codiereffizienz unter Vermeidung der Komplexität von MPEG-4. H.264 beruht wie MPEG-2 auf dem bewährten Konzept der hybriden Kodierung, das heißt, dass bewegungskompensiert prädierte Bildblöcke einer Transformation unterzogen werden, wobei mit einer geschlossenen Schleife dafür gesorgt wird, dass im Kodierer die Rekonstruktion im Decodierer vorweggenommen wird (vgl. Schmidt 2021, S.254).

3.1.7 Eigenschaften früher Videocodecs

Zuvor wurden die einzelnen Videocodecs der 1990er und frühen 2000er Jahre begutachtet. Zusammenfassend lässt sich sagen, dass frühe Videocodecs sich durch moderate Kompressionsraten und vergleichsweise niedrige Videoqualität auszeichneten. Sie waren vor allem darauf ausgerichtet, die damals verfügbaren Übertragungs- und Speicherkapazitäten zu nutzen. Die Videokomprimierungstechniken dieser Codecs waren im Vergleich zu modernen Codecs weniger effizient, führten oft zu sichtbaren Kompressionsartefakten und waren auf bestimmte Anwendungen wie Videoübertragung und Speicherung beschränkt.

Wie bereits eingangs erwähnt, legten trotz ihrer begrenzten Leistungsfähigkeit frühe Videocodecs den Grundstein für spätere Entwicklungen und waren wichtige Meilensteine in der Geschichte der Videokomprimierung. Die Erforschung ihrer Eigenschaften ermöglicht ein besseres Verständnis der technologischen Fortschritte und Innovationen, die zur Verbesserung der Videokomprimierung und -übertragung im Laufe der Zeit geführt haben. Diese sollen im Folgenden betrachtet werden.

3.2 Fortschritte in der Videocodierungstechnologie

Die Videocodierungstechnologie hat in den letzten Jahren bedeutende Fortschritte gemacht und spielt eine entscheidende Rolle bei der Bereitstellung hochwertiger Videoinhalte über verschiedene Plattformen und Netzwerke.

In diesem Kapitel werden einige wichtige Fortschritte in der Videocodierungstechnologie aufgezeigt, die dazu beigetragen haben, die Videoqualität zu verbessern, die Datenkomprimierung zu optimieren und die Leistungsfähigkeit moderner Videocodecs zu steigern.

3.2.1 Fortschritte in der Komprimierungseffizienz

Ein bedeutender Fortschritt in der Videocodierungstechnologie ist die Steigerung der Komprimierungseffizienz. Moderne Videocodecs wie H.265, auch bekannt als High Efficiency Video Coding (HEVC), und AOMedia Video 1 (AV1) bieten eine erheblich verbesserte Komprimierungseffizienz im Vergleich zu ihren Vorgängern wie H.264 oder dem VP9-Format. Durch die Integration fortschrittlicher Kompressionsalgorithmen und Kodierungstechniken sind diese Codecs in der Lage, eine höhere Videoqualität bei deutlich niedrigeren Datenraten zu erreichen. Dies ermöglicht eine effizientere Nutzung der verfügbaren Bandbreite und eine optimierte Übertragung von Videos über das Internet (vgl. Schmidt 2021, S. 258-259).

3.2.2 Unterstützung höherer Auflösungen und Farbtiefe

Ein weiterer Fortschritt in der Videocodierungstechnologie ist die Unterstützung höherer Auflösungen und Farbtiefen. Frühere Videocodecs waren hauptsächlich auf Standardauflösungen wie SD (Standard Definition) oder HD (High Definition) beschränkt. Moderne Codecs hingegen ermöglichen die Kodierung und Übertragung von Videos in ultrahoher Auflösung (UHD) oder sogar 8K. Darüber hinaus unterstützen einige Codecs eine größere Farbtiefe, wie zum Beispiel 10-Bit oder 12-Bit-Farbabbildung, was zu einer präziseren und realistischeren Farbdarstellung führt (vgl. Bross et al. 2013, S.1).

3.2.3 Effiziente Bitratenkontrolle

Des Weiteren beschreibt die effiziente Bitratenkontrolle eine weitere wichtige Weiterentwicklung. Moderne Videocodecs bieten fortschrittliche Bitratenkontrollmechanismen wie konstante Bitraten (CBR), variable Bitraten (VBR) und adaptive Bitraten (ABR).

Die variable Bitratenkodierung ermöglicht eine dynamische Zuweisung von Bits, wobei mehr Bits den komplexeren Videoabschnitten und weniger Bits den weniger komplexen Abschnitten zugewiesen werden. Der Benutzer gibt eine subjektive Qualitätsstufe vor, und der Encoder passt die Bitzuweisung entsprechend an, um das gewünschte Qualitätsniveau zu erreichen. Dadurch wird ein konsistenteres Wahrnehmungserlebnis erzielt. Allerdings muss das komprimierte Video immer noch in die verfügbare Kanalbandbreite passen, was eine Begrenzung der maximalen Bitrate erfordert. Daher ermöglicht die VBR-

Kodierungsmethode normalerweise die Festlegung eines Bitratenbereichs mit maximalen und/oder minimalen zulässigen Bitraten. Insbesondere für Speicheranwendungen erweist sich VBR im Vergleich zu CBR als geeigneter. Zusätzlich zu CBR und VBR kann auch die Kodierung mit durchschnittlicher Bitrate verwendet werden, um sicherzustellen, dass der Ausgabe-Videostream eine vorhersehbare langfristige Durchschnittsbitrate aufweist (vgl. Shahriar Akramullah, S.155). Diese in Kapitel 3.2 vorgestellten Techniken finden Anwendungen in aktuellen Codecs auf die nun genauer eingegangen wird.

3.3 Aktuelle Videocodecs: Vorstellung und Vergleich

Die bemerkenswerte Zunahme des Videoverbrauchs über IP-Netzwerke hat eine entscheidende Notwendigkeit für die Entwicklung fortschrittlicher Videokompressionstechnologien hervorgebracht. Videodaten machten bereits im Mai 2015 mehr als 75% des gesamten Internetverkehrs aus (vgl. Martin Rerabek et al. 2015, S1.).

Diese neuen Entwicklungen sollen erheblich effizienter sein als bisherige Videocodecs und haben so zur Konzeption von bedeutenden Codecs geführt:

3.3.1 H.265 (HEVC)

Ein sehr bekannter unter ihnen ist der High Efficiency Video Coding. Auch bekannt als H.265-Standard (ISO/IEC 23008-2), wurde er im Jahr 2013 von den Organisationen ITU-T Video Coding Experts Group (VCEG) und ISO/IEC Moving Picture Experts Group als jüngster gemeinsamer Videocodierungsstandard verabschiedet. Dieser Standard knüpft an den früheren H.264-Standard an. Das Ziel von HEVC besteht darin, den steigenden Ansprüchen an Videos mit immer höheren Auflösungen, wie HD (1920 × 1080 Pixel), Ultra-HD (4k × 2k Pixel) und darüber hinaus, gerecht zu werden (vgl. Sullivan, Gary J. et al. 2012, S. 1649). Dabei werden vor allem zwei zentrale Herausforderungen angesprochen: die erhöhte Videoauflösung die gestiegene Nutzung von parallel verarbeitenden Architekturen. Infolgedessen ist das HEVC-Verfahren darauf ausgelegt, eine doppelt so hohe Kompressionseffizienz wie H.264 zu erreichen (vgl. Shahriar Akramullah, S. 84).

3.3.2 VP9

Ein weiterer Codec ist die VP9-Videokompression, eine von Google entwickelte lizenzgebührenfreie Alternative zu HEVC. VP9 legt einen starken Fokus auf die Unterstützung des videobasierten Konsums über das Internet.

Die Tatsache, dass VP9 in einem ähnlichen Zeitraum wie HEVC veröffentlicht und als überlegene Alternative formuliert wurde, weckte Interesse in der Forschungs- und Fachgemeinschaft. Dies führte zu mehreren Untersuchungen, in denen diese beiden Codecs sowie H.264 miteinander verglichen wurden. Dabei wurden hauptsächlich objektive Metriken zur Messung der Kodierungseffizienz verwendet. Die

Ergebnisse dieser Studien waren jedoch widersprüchlich und hingen stark von den jeweiligen Untersuchungsansätzen ab (vgl. Martin Rerabek et al. 2015, S.1).

In einer der Studien¹ wurde argumentiert, dass VP9 sowohl gegenüber H.264 als auch HEVC unterlegen sei. Es wurde gezeigt, dass HEVC im Vergleich zu VP9 durchschnittliche Bitratensparnisse von 43,3% aufweise. Eine andere Untersuchung² kam jedoch zu einem gegenteiligen Schluss. Hier wurde festgestellt, dass VP9 eine vergleichbare Kompressionseffizienz wie HEVC aufweise und deutlich effizienter sei als H.264.

Die Unterschiede in diesen Schlussfolgerungen lassen sich vor allem auf unterschiedliche Annahmen bezüglich der Anwendungsszenarien in den Studien sowie auf unterschiedliche Kodierungseinstellungen zurückführen. Um die Vergleichbarkeit zu erhöhen, haben die Autoren von der Studie³ diese erweitert und ein Szenario mit geringer Verzögerung⁴ untersucht, das für Echtzeitanwendungen relevant ist. Dabei wurde festgestellt, dass HEVC im Vergleich zu VP9 durchschnittliche Bitratensparnisse von 32,5% aufweist, wenn Messungen wie „peak signal-to-noise ratio“ (PSNR) verwendet werden (vgl. Martin Rerabek et al. 2015, S.1; Chen et al. 2018).

3.3.3 AV1

Das AOMedia Video 1 Codec ist ein moderner, offener Videokompressionsstandard, der von der Alliance for Open Media (AOMedia) entwickelt wurde. Die Allianz besteht aus einer Zusammenarbeit von führenden Unternehmen in der Technologiebranche, die das Ziel verfolgen, einen hoch effizienten und lizenzgebührenfreien Videocodec zu schaffen.

Die Entwicklung von AV1 konzentriert sich darauf, unter anderem eine konsistente, hochwertige Echtzeit-Videowiedergabe zu ermöglichen, Skalierbarkeit für moderne Geräte bei verschiedenen Bandbreiten sicherzustellen, eine beherrschbare Rechenlast zu gewährleisten, Hardware-Optimierungen vorzunehmen und Flexibilität für kommerzielle und nichtkommerzielle Inhalte zu bieten (vgl. Chen Yue et al. 2020, S.1-2). Der Codec wurde zunächst mit VP9-Tools und Verbesserungen initialisiert, und anschließend wurden neue Codierungstools vorgeschlagen, getestet, diskutiert und in den Arbeitsgruppen von AOMedia für Codec, Hardware und Tests weiterentwickelt (vgl. Chen et al. 2018, S. 41).

AV1 übertrifft VP9 um 15,4 % bzw. 19,5 % bei der Bitrate-Einsparung für UHD- und HD-Videos. Allerdings benötigt der AV1-Encoder wesentlich mehr Zeit für die Videokodierung. Insbesondere ist

¹ D. Grois, D. Marpe, A. Mulyoff, B. Itzhaky, and O. Hadar, “Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders,” in PCS, 2013, pp. 394–397

² D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu, and R. Bultje, “The latest open-source video codec VP9-an overview and preliminary results.” in PCS, 2013, pp. 390–393.

³ D. Grois, D. Marpe, A. Mulyoff, B. Itzhaky, and O. Hadar, “Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders,” in PCS, 2013, pp. 394–397

⁴ D. Grois, D. Marpe, T. Nguyen, and O. Hadar, “Comparative assessment of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders for low-delay video applications,” in Proc. SPIE, vol. 9217, 2014, pp. 92 170Q–92 170Q–10

die Laufzeit des AV1-Encoders im Vergleich zu VP9 etwa 117,5-mal langsamer. Dies bedeutet, dass die recht erheblichen Bitrate-Einsparungen von AV1 im Vergleich zu VP9 einen enormen Anstieg der Rechenkomplexität erfordern (vgl. Minopoulos et al. 2020, S. 112)

Nachdem nun einige Codecs vorgestellt wurden, folgt der Bedarf an geeigneten Bewertungskriterien, um sie besser miteinander vergleichen zu können.

4 Bewertungskriterien der Videocodecs

In diesem Kapitel werden die relevanten Bewertungskriterien im Rahmen dieser Arbeit näher erläutert. Hierzu gehören subjektive und objektive Bewertungsmethoden sowie die Effizienz der Datenkomprimierung unter Berücksichtigung von Datengröße und Dauer des Kodierungsvorgangs. Diese Kriterien spielen eine zentrale Rolle beim Vergleich und der Bewertung verschiedener Videocodecs für verschiedene Anwendungsszenarien (vgl. Minopoulos et al. 2020, S.112).

4.1 Videoqualität: Subjektive und Objektive Bewertungsmethoden

Die Bewertung der Videoqualität ist ein zentraler Bestandteil für die Beurteilung von Videocodecs. Da in der heutigen Medienwelt immer mehr Videocodecs entwickelt werden, ist es notwendig, die Leistung und Qualität dieser Codecs vergleichbar zu machen. Im Gegensatz zu einer subjektiven Bewertungsmethode, bei der die wahrgenommene Qualität eines Videos von menschlichen Beobachtern bewertet wird, ermöglichen objektive Methoden eine quantitative Analyse und Messung der Qualität des Videos. Hierfür wurden in der Vergangenheit einige Techniken mithilfe von Logarithmen entwickelt.

Es liegt nahe, als ersten Ansatz die Differenz der Datenworte zwischen dem durch Anwendung eines Codes erzeugten Bild und dem ursprünglichen Referenzbild zu ermitteln (vgl. Schmidt 2021, S.266). Weiter beschreibt Schmidt, dass der gesamte Codec, einschließlich der Decodierung, auf das veränderte Bild angewendet werden soll. Auf diese Weise kann der Fehler quantifiziert werden, der umso größer wird, je größer die Differenz zwischen den Bildern ist. Eine erhebliche Differenz aufgrund von Bildfehlern lässt sich mit den Signalfehlern in der analogen Welt vergleichen, wo diese Fehler als Signalrauschen auftreten. Ein hohes Verhältnis zwischen dem Nutzsignal und dem Rauschsignal, das als Signal-Rausch-Verhältnis (SNR) bezeichnet wird, deutet darauf hin, dass das Signal nur geringfügig gestört ist. In der analogen Nachrichtentechnik wird das SNR logarithmisch aus dem Verhältnis zwischen der Nutzleistung (PSignal) und der Rauschleistung (PNoise) gebildet.

4.1.1 Peak Signal to Noise Ratio (PSNR)

Das identische Verfahren kann auf die Fehler angewendet werden, die durch die Reduktion der Daten entstehen. Als Ergebnis wird das Peak Signal to Noise Ratio ermittelt, welches wie folgt berechnet werden kann:

$$PSNR = 10 * \log_{10}(peakval^2/MSE)$$

Dabei ist peakval der maximale Wert im Bild. Wenn es sich um 8-Bit-Daten handelt, beträgt peakval beispielsweise 255. Diese Gleichung drückt den absoluten Fehler in Dezibel (dB) aus.

Das PSNR berechnet das Verhältnis zwischen der maximal möglichen Signalstärke und dem störenden Rauschen, das die Qualität der Darstellung beeinflusst. Es ist eine gängige Methode zur Bewertung der Qualität von verlustbehafteten Bildkompressionsverfahren (vgl. Schmidt 2021, S. 266-267). Dabei wird das Originalbild als Signal betrachtet und der Fehler, der durch die Kompression oder Verzerrung entsteht, als Rauschen bezeichnet.

Das PSNR liefert eine grobe Einschätzung der menschlichen Wahrnehmung der Rekonstruktionsqualität im Vergleich zu den Kompressionsverfahren. Er wird üblicherweise auf einer logarithmischen Dezibelskala berechnet, da Signale einen großen Dynamikbereich haben. Dieser Bereich hängt von der Qualität der Bilder ab und kann variieren

Bei der Kompression von Bildern und Videos liegt der PSNR-Wert normalerweise zwischen 30 und 50 dB für 8-Bit-Daten und zwischen 60 und 80 dB für 16-Bit-Daten. Bei drahtloser Übertragung wird ein akzeptabler Qualitätsverlust in einem Bereich von etwa 20-25 dB akzeptiert. Ab einem PSNR von 35 dB oder höher kann eine akzeptable Bildqualität erwartet werden. Dennoch können Bilder mit demselben PSNR-Wert subjektiv sehr unterschiedlich beeinträchtigt wirken. Der PSNR ermöglicht hauptsächlich den Vergleich von Codecs, wenn dieselbe Bildvorlage verwendet wird. Dies liegt unter anderem daran, dass Fehler weniger auffällig sind, wenn das Bild viele unregelmäßige Strukturen aufweist und ein Verdeckungseffekt eintritt (vgl. Schmidt 2021, S. 268; Sara et al. 2019, S.11). Da das im Rahmen dieser Arbeit durchgeführte Experiment für alle Codecs mit dem gleichen Bildmaterial verwendet wird, eignet sich die PSNR-Methode gut zur Beurteilung der Codecs im Rahmen des Experiments.

4.1.2 Visuelle Suche nach Blockartefakten

Subjektive Bewertungsmethoden werden bei der Analyse von Videoqualität und der Beurteilung der Videos im Rahmen dieser Arbeit ebenfalls eine wichtige Rolle spielen. Eine spezifische Herangehensweise innerhalb dieser Methode besteht darin, die Auswirkungen von Blockartefakten auf die Videoqualität zu untersuchen, während die Bitrate schrittweise reduziert wird. Blockartefakte treten in komprimierten Videos auf, die mit verlustbehafteten Videocodecs erstellt wurden. Sie manifestieren sich als unerwünschte sichtbare Quadratmuster oder Kacheln, die die Bildqualität beeinträchtigen. Bei niedrigeren Bitraten können diese Artefakte verstärkt auftreten, da der Codec mehr Daten komprimieren muss und dabei feinere Details reduziert werden können.

Dieser Ansatz zielt darauf ab, das Verhalten von Videocodecs mit zunehmend niedrigeren Bitraten hinsichtlich der visuellen Qualität zu erfassen.

4.2 Effizienz der Datenkomprimierung

Die Effizienz der Datenkomprimierung ist ein kritischer Faktor für die Qualität und die Größe des komprimierten Videos. Eine effiziente Komprimierung ermöglicht es, die Datenmenge zu reduzieren, ohne dabei die visuelle Qualität des Videos signifikant zu beeinträchtigen. Eine hohe Komprimierungseffizienz ist besonders wichtig für Anwendungen, bei denen die Bandbreite begrenzt ist, wie zum Beispiel beim Streaming. Effiziente Komprimierung sorgt dafür, dass die Übertragung des Videos reibungslos und mit minimalen Wartezeiten erfolgt, ohne die Videoqualität zu beeinträchtigen (vgl. Shahriar Akramullah, S.52-54). Die Datengröße und die Dauer des Kodiervorgangs werden im folgenden Experiment als wesentliche Faktoren für die Beurteilung der Effizienz verwendet.

5 Experiment und Experteninterview

5.1 Einleitung und Zielsetzung des Experiments

Um nun die bereits vorgestellten Codecs sowohl subjektiv als auch objektiv und messbar bewerten zu können, bedarf es eines Experiments, welches darauf abzielt, Videos mit möglichst hoher Frequenz sowie starken Bewegungen zu produzieren. Diese werden dann anschließend in ihrer Datenbitrate begrenzt und anschließend kodiert. Die dabei gewonnenen empirischen Daten lassen sich dann miteinander vergleichen. Als objektive Bewertungsmethode wird anschließend für jedes kodierte Video der PSNR-Wert ermittelt, indem es mit dem Ursprungsmaterial, dem im Folgenden genannten „Ground-Zero-Video“, verglichen wird. So können erwartete Erkenntnisse und Eigenschaften der jeweiligen Codecs überprüft werden. Für die Beantwortung der eingangs erwähnten Forschungsfragen ist das Experiment ebenfalls notwendig.

Das Experiment wird für eine genauere Untersuchung mit zwei unterschiedlichen Ground-Zero-Videos durchgeführt. Hierfür dient zum einen das gewonnene Rohmaterial, welches im folgenden beschriebenen Versuchsaufbau produziert wird. Zum anderen wird eine Video-Animation mithilfe des Programms Adobe After Effects erstellt. Die Animation, die im Folgenden als „Ground-Zero-animiert“ bezeichnet wird, dient hierbei mit hohen Frequenzen als „Stresstest“ für den Kodierungsvorgang, während das Ground-Zero-Video als möglichst praxisnahes Beispiel für heutige Streaming-Angebote dienen soll.

Für die Produktion der Videos wird eine Blackmagic Ursa Broadcast G2 verwendet. Dieses Modell wird in aktuellen Mediaproduktionen und Live-Streaming Veranstaltungen genutzt, um hochwertiges und professionelles Material herzustellen. Ein weiterer Vorteil ist die Vielfalt an Einstellungen, die getroffen werden können, um das Material nahezu unkomprimiert weiterzuverarbeiten.

Zusätzlich werden mit der Befragung eines Experten aus der Film- und Streaming-Branche Informationen über den aktuellen Stand in der Medienwelt gewonnen.

5.2 Versuchsaufbau

5.2.1 Produktion von Videos mit der Blackmagic Ursa Broadcast G2

Um möglichst viele, voneinander unabhängig bewegende Texturen mit einer hohen Frequenz zu erhalten, wurde ein Versuch aufgebaut, bei dem kontrastreiches Kreppband in lange Streifen geschnitten wurde. Anschließend folgte die Fixierung der Bänder in abwechselnder Reihenfolge am oberen Ende (vgl. Abbildung 2). Für schnelle Bewegungen der einzelnen Bänder sorgten zwei Ventilatoren, die auf die Bänder gerichtet waren.



Abbildung 2: Versuchsaufbau

Wie bereits erwähnt, diente die Blackmagic Ursa Broadcast G2 als Kamera für dieses Experiment. Sie war ausgestattet mit einem HA18X7.6BERD-S5 Objektiv der Marke FUJINON. Die Kamera ist eine leistungsstarke, speziell für klassische und Online-Broadcaster designte Kamera und findet Anwendung in zahlreichen professionellen Medienproduktionen (vgl. Blackmagic Design 2023).

Das Videomaterial wurde mit einer Bildgröße von 1920 x 1080 bei 30 Bildern pro Sekunde sowie dem Codec „Apple ProRes 422 HQ“ aufgezeichnet.

„Apple ProRes 422 HQ ist eine Version von Apple ProRes 422 mit höherer Datenrate, die eine visuelle Qualität auf dem gleichen hohen Niveau wie Apple ProRes 4444 bietet, jedoch für 4:2:2-Bildquellen. Apple ProRes 422 HQ ist in der Postproduktion von Videos sehr gebräuchlich und bietet die visuell verlustfreie Erhaltung hochqualitativer, professioneller HD-Videos, die von einem Single-Link-HD-SDI-Signal übertragen werden können. Dieser Codec unterstützt 4:2:2-Videoquellen mit einer 10-Bit-Farbtiefe in voller Breite und bleibt gleichzeitig über zahlreiche Decodierungs- und Codierungsprozesse hinweg visuell verlustfrei. Die Zieldatenrate beträgt ungefähr 220 Mbit/s bei 1920 x 1080.“ (Apple Support 2023).

Um diese Werte zu überprüfen, wurden die Rohdaten mithilfe der Open-Source-Software MediaArea angezeigt (vgl. Abbildung 3).

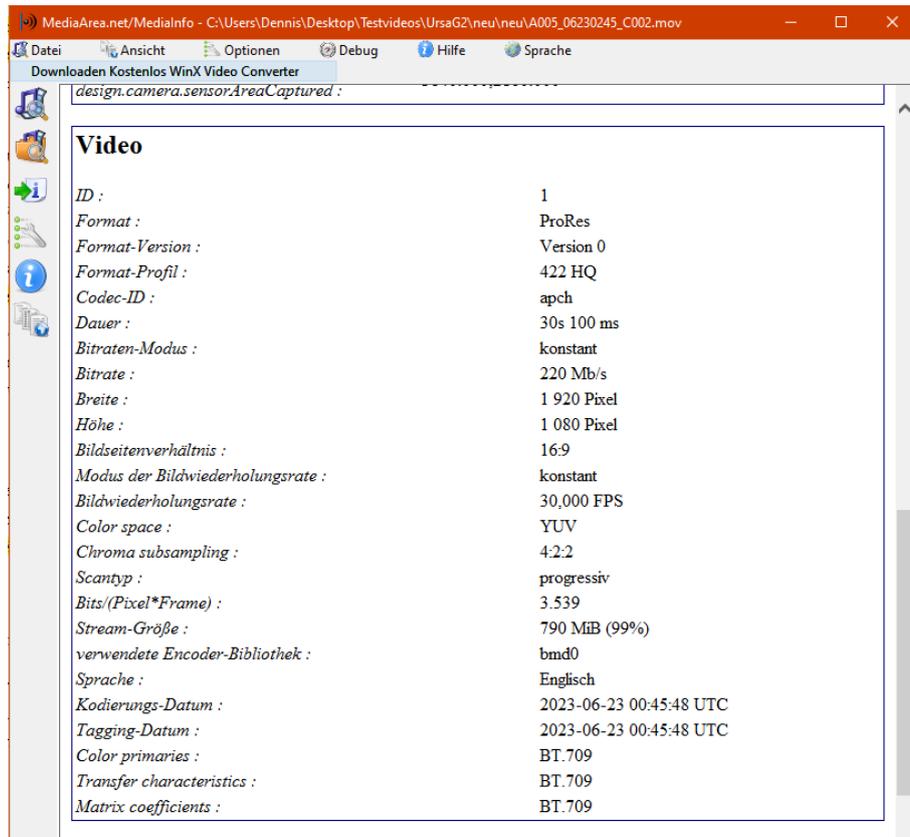


Abbildung 3: Betrachtung der Rohdaten mithilfe der Software MediaArea

Anschließend folgte der Schnitt des Videomaterials mithilfe des Programms Adobe Premiere Pro. Hierbei wurde darauf geachtet, dass jeweils die Anfangs- und Endszene keine großen Bewegungen aufwiesen. So ist der Übergang, an dem als erstes Blockartefakte auftreten können, besser sichtbar und auch mithilfe des PSNR-Wertes lässt sich anhand der Extremwerte „min“ und „max“ ein deutlicherer Unterschied messen. Das geschnittene Ground-Zero-Video wurde anschließend mit den in Abbildung 4 gezeigten Eigenschaften exportiert.

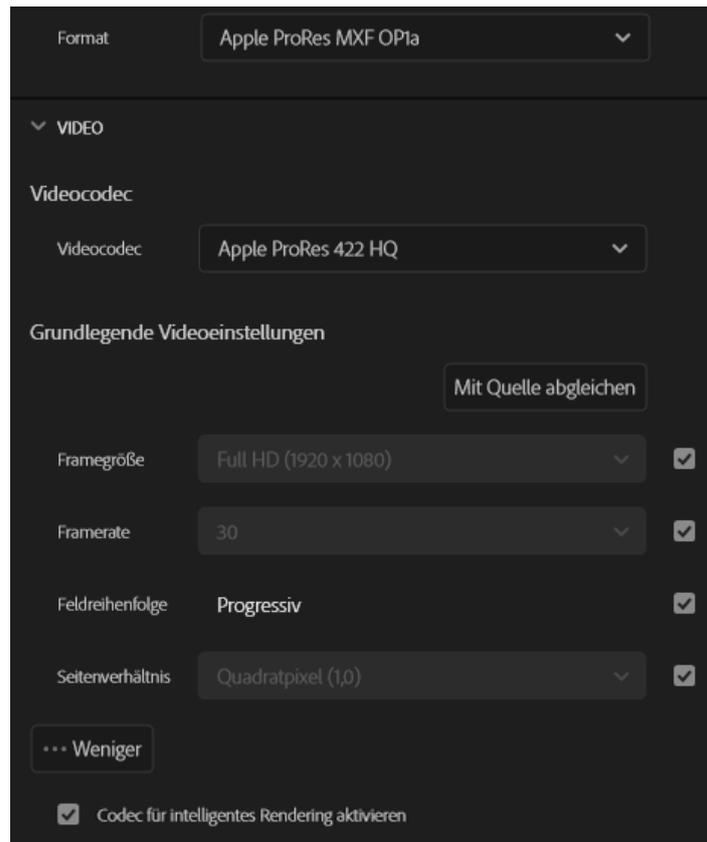


Abbildung 4: Exporteinstellungen in Premiere Pro

Hierbei galt besondere Aufmerksamkeit darauf, die Eigenschaften des Rohmaterials unverändert zu belassen. Zudem war die Funktion "intelligentes Rendering" aktiviert. (vgl. Abbildung 4).

„Beim Exportieren kann das intelligente Rendering für bestimmte Formate verwendet werden, um eine bessere Ausgabequalität zu erzielen, indem eine erneute Komprimierung, sofern möglich, vermieden wird. Intelligentes Rendering funktioniert nur, wenn Quellcodec, Größe, Framerate und Bitrate mit den Exporteinstellungen übereinstimmen“ (Adobe 2023). Das fertige Video diente nun als Grundlage für die im Experiment vorgenommenen Kodierungen. Außerdem wurde es durch eine Animation erweitert, was im folgenden Abschnitt beschrieben wird.

5.2.2 Erstellung einer Animation in Adobe After Effects

Mithilfe einer Animation ist es wesentlich einfacher, bestimmte Merkmale oder Szenarien gezielt zu erstellen, um die Auswirkungen der Codecs auf diese Aspekte zu untersuchen. Beispielsweise können hochfrequente Bewegungen, Farbverläufe oder komplexe Texturen in einer Animation einfacher produziert und gesteuert werden als in natura. Um die Bewegungsdichte einzelner Bildbereiche nun maximal zu erhöhen, erhielt das in Abschnitt 5.2.1 produzierte Ground-Zero-Video zusätzlich einen künstlich erzeugten Schneesturm sowie Konfetti, welches das Bild permanent an nahezu jeder Stelle in Bewegung bleiben lässt.

Um nun eine solche Animation zu erzeugen, eignet sich das Programm Adobe After Effects sehr gut, da es mit wenigen einfachen Schritten ein akzeptables Ergebnis liefert. Für den Schnee-Effekt diente der „CC Snowfall“ Effekt, der in der Effekt-Bibliothek unter dem Reiter „Simulation“ zu finden ist (vgl. Abbildung 5).

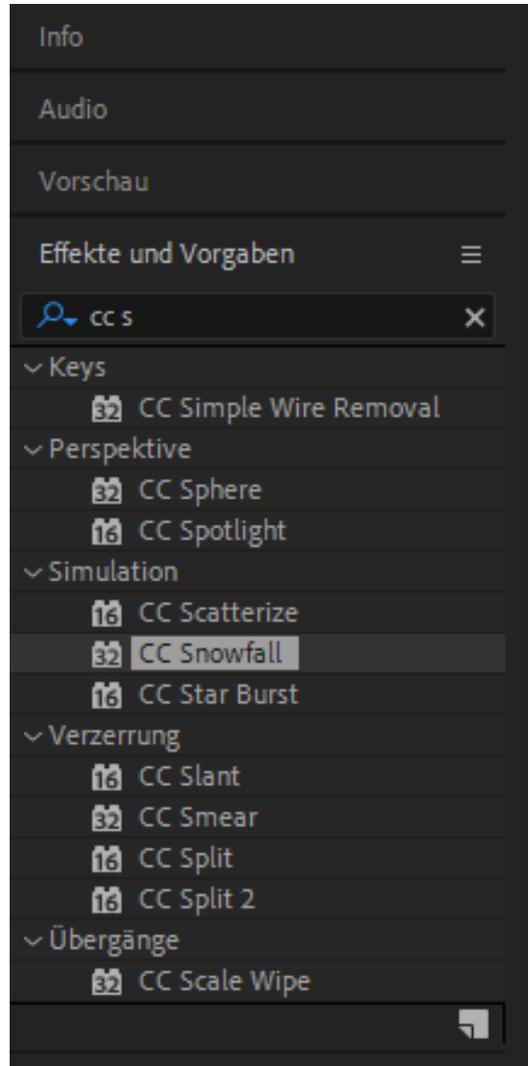


Abbildung 5: Effekt „CC Snowfall“ in After Effects

Dieser Effekt lässt sich mit einigen Einstellungen sehr gut auf das gewünschte Ergebnis anpassen. Hierfür wurde die Anzahl der einzelnen Schneeflocken um den Faktor 4 erhöht und die Größe der Schneeflocken

um zwei Einheiten verringert. Dies geschah mit den Einstellungen „Flakes“ und „Size“ (vgl. Abbildung 6). Das Ergebnis wurde mit dem im folgenden Abschnitt entstehenden Konfetti kombiniert.

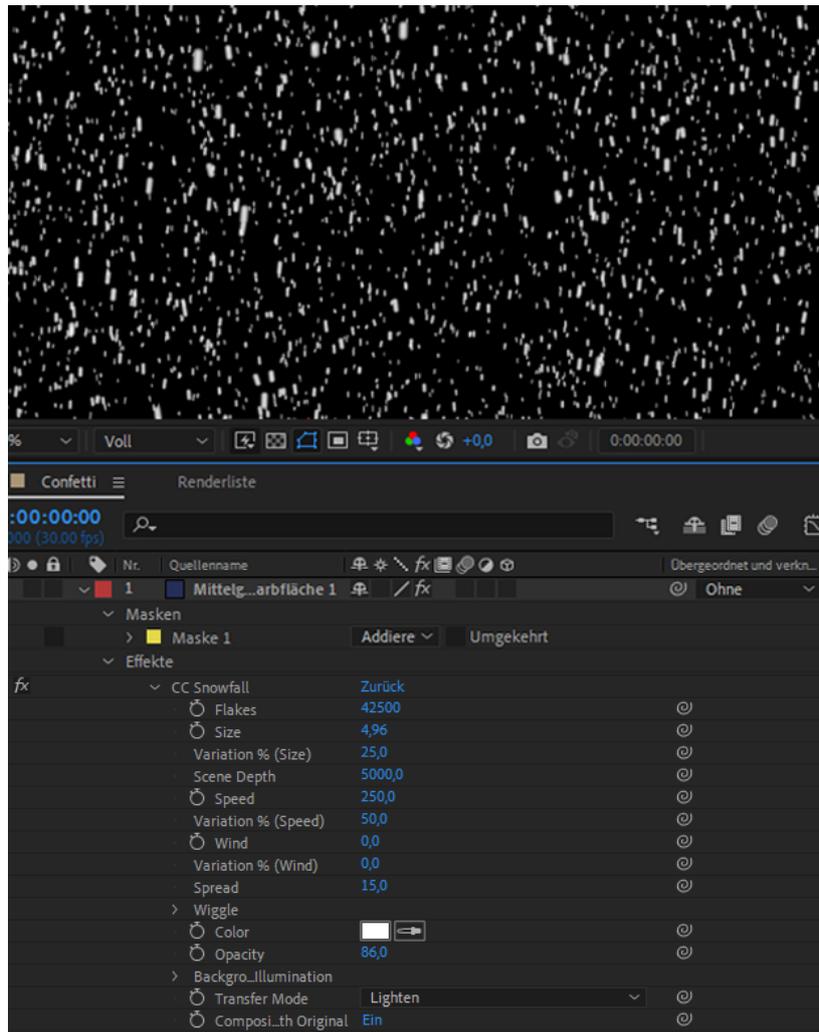


Abbildung 6: „CC Snowfall“-Parameter in After Effects

Der Konfetti-Effekt entstand durch eine Reihe von Schritten ebenfalls mithilfe von After Effects. Zu Beginn wurde dafür eine neue Farbfläche angelegt, auf der dann eine neue Maske, wie in Abbildung 7 zu erkennen, erzeugt wurde.

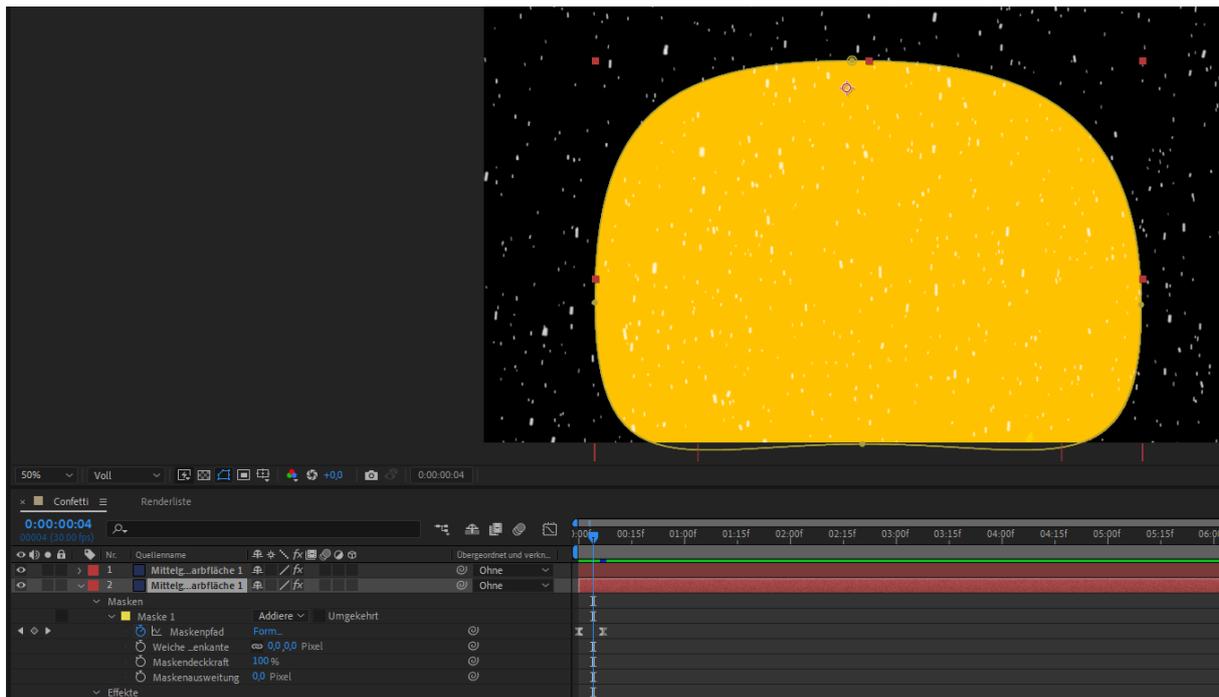


Abbildung 7: Erstellung des Konfettis in After Effects

Diese Maske wurde anschließend mithilfe von Keyframes entgegen der Zeitachse verzerrt, sodass die Form der Maske aus Abbildung 7 die Zielposition darstellte. Um dies zu erreichen, setzte man am Anfang des Videos einen Keyframe unter der Einstellung „Maskenpfad“ und verschob die Kanten des Kreises (vgl. Abbildung 8). Diese animierte Bewegung der Maske sorgt so für den „Explosionseffekt“ des einzelnen Konfettis, bevor es an den unteren Bildschirmrand fällt.

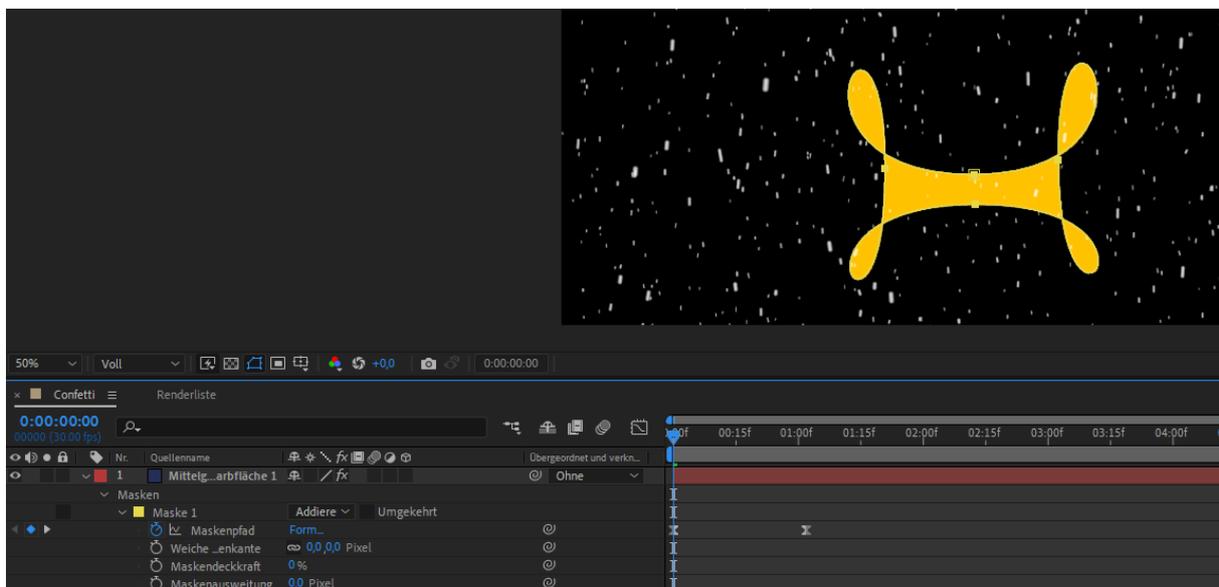


Abbildung 8: Anpassung des Maskenpfads in After Effects

Um abschließend einen Konfetti-Effekt zu erzeugen, war es notwendig, auf diese Maske der Simulationseffekt „CC Pixel Molly“ anzuwenden. Dieser bewirkte das Aufteilen der Maske in einzelne Partikel,

die dann mit mehreren Parametern wie „Gravity“ oder „Force“ weiter angepasst werden konnten (vgl. Abbildung 9).

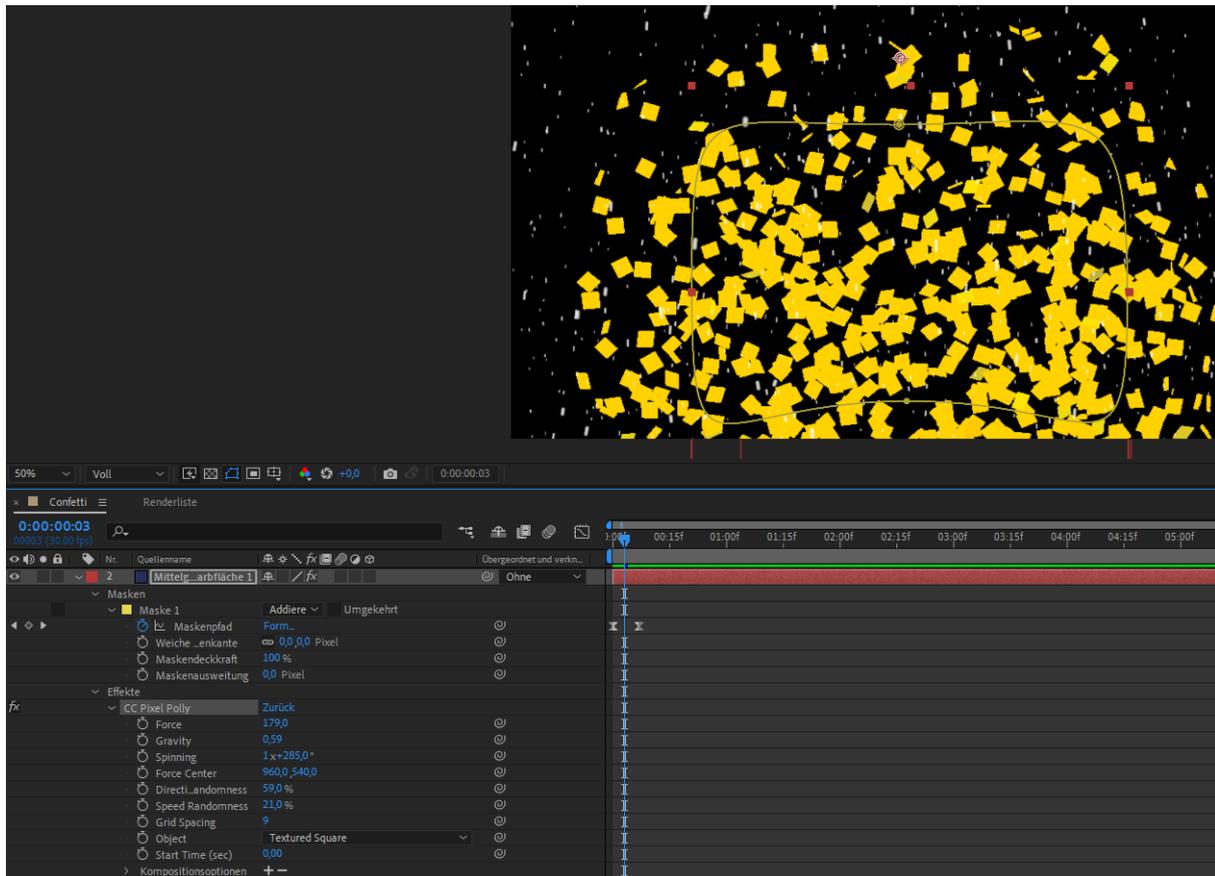


Abbildung 9: Anwendung des Effekts "CC Pixel Molly" in After Effects

Nachdem sowohl die Farbe des Konfettis als auch die Länge der Animation etwas angepasst wurde, konnte sie unter den Rendereinstellungen mit dem Ausgabemodul „Verlustfrei“, und der Auswahl des RGB- und Alphakanals für die Transparenz des Hintergrundes, exportiert werden. Um die Farben noch etwas bunter zu gestalten, wurde die fertige Animation mithilfe des Überblendmodus „Differenz“ in Premiere Pro über das in Kapitel 5.2.1 erstellte Video gelegt. Dies wurde dann mit den gleichen Einstellungen erneut exportiert und bildete das zweite Testvideo.

5.3 Vorstellung der Software FFmpeg

Im Folgenden wird die Software FFmpeg vorgestellt, die einen wichtigen Grundstein für das Bearbeiten der Videos im weiteren Verlauf des Experiments darstellte.

FFmpeg ist eine leistungsstarke Open-Source-Software, die für die Verarbeitung von Multimedia-Daten, insbesondere von Videos und Audios, entwickelt wurde. Die Software ist plattformunabhängig und unterstützt eine Vielzahl von Betriebssystemen, darunter Windows, macOS, Linux und weitere. FFmpeg ist bekannt für seine Flexibilität, Geschwindigkeit und Vielseitigkeit bei der Bearbeitung, Umwandlung

und Kodierung von Mediendateien. Außerdem unterstützt es gängige Methoden zur Bewertung der Videoqualität wie zum Beispiel Video Multi-Method Assessment Fusion (VMAF) oder PSNR (vgl. FFmpeg 2023).

5.3.1 Funktionen von FFmpeg

FFmpeg bietet eine große Bandbreite an Funktionen und Werkzeugen, die es zu einem äußerst nützlichen Werkzeug machen (vgl. ffmpeg Documentation 2023). Zwei der wichtigsten Funktionen von FFmpeg sind die Video- und Audiokodierung sowie die Konvertierung von Dateiformaten. FFmpeg unterstützt dabei verschiedene Videocodecs wie H.264, H.265, VP9, MPEG-4 Part 2 und viele andere. Außerdem können Mediendateien zwischen verschiedenen Formaten umgewandelt werden, etwa von AVI zu MP4, von MKV zu MOV und umgekehrt. Es unterstützt eine breite Palette von Containerformaten und Codecs. Damit erfüllt FFmpeg als Software die Bedingungen für die Durchführung des Experiments. Die Funktionen sind wichtige Hilfen für die Bearbeitung und Neukodierung der Ground-Zero-Videos.

5.3.2 Nutzung von FFmpeg im Experiment

Im Rahmen des Experiments wurde FFmpeg als das Hauptwerkzeug für die Weiterverarbeitung der erfassten, hochfrequentigen Videos verwendet. Die Software konnte eingesetzt werden, um die Rohdaten in verschiedene Codecs und Formate zu konvertieren, die Datenbitrate anzupassen, die Audiospur zu trennen und um die Leistungsfähigkeit der Codecs in Bezug auf Laufzeit und Effizienz zu testen. Die dafür genutzten Befehlszeilen sind sowohl in der offiziellen Dokumentation des Entwicklers (vgl. ffmpeg Documentation 2023), als auch im FFmpeg-Wikipedia (vgl. trac.FFmpeg 2023) sehr gut beschrieben.

5.4 Durchführung von Kodierung und Dekodierung

Der folgende Abschnitt behandelt die Bearbeitung der im Kapitel 5.2 sowie Kapitel 5.3 produzierten Videos mithilfe der Software FFmpeg. Dabei wurde darauf geachtet, dass die verglichenen Codecs mit gleichen Qualitätseinstellungen wie etwa CBR und VBR, sowie im selben Modi kodiert wurden, um einen besseren Vergleich darzustellen.

Nachdem zunächst benötigte Umgebungsvariablen gesetzt und das Programm in Windows PowerShell geladen wurde, konnte mit den Kodierungsprozessen begonnen werden. Es sei noch zu erwähnen, dass im Rahmen dieses Experimentes mehrere Codecs verglichen werden sollten. Daher war eine übersichtliche Ordnerstruktur sinnvoll. Das System, mit dem die Bearbeitung der Videos durchgeführt wurde, besaß einen AMD Ryzen 5 5600X Prozessor sowie eine Nvidia Geforce RTX 3070 Grafikkarte.

Da bei diesem Experiment die Tonspuren vernachlässigt werden konnten, bestand der erste Schritt zunächst darin, bei beiden Ausgangsvideos die Tonspur von der Videospur zu trennen. Um die Audiospuren eines Videos zu löschen, wird der jeweilige Audio-Codec explizit deaktiviert. Dies geschieht mit dem „-an“ (Audio-None) Parameter (vgl. Codeblock 1).

Codeblock 1: Deaktivierung der Audiospur
Quelle: Eigene Darstellung

```
1 ffmpeg -i ground_zero.mxf -c:v copy -an ground_zero.mxf
```

Dabei gab der Parameter „-i“ die Eingangsquelle an. Mithilfe des Parameters „-c:v copy“ wurde der Video-Stream direkt in die Ausgabedatei kopiert, ohne eine erneute Kodierung vorzunehmen. Dadurch blieb die Videoqualität unverändert. Nachdem dieser Schritt für beide Ground-Zero-Videos durchgeführt wurde, konnte die Konvertierung in andere Codecs erfolgen.

5.4.1 MPEG-2

Für eine Konvertierung der Videos können in der Befehlszeile von FFmpeg verschiedene Parameter angegeben werden.

Um eine MXF-Datei in eine MPEG-2 Datei zu konvertieren, wurde mit dem Parameter „-c:v mpeg2video“ der MPEG-2-Video-Codec für die Videoausgabe ausgewählt (vgl. Codeblock 2).

Codeblock 2: Kodierung zu MPEG-2
Quelle: Eigene Darstellung

```
1 ffmpeg -i ground_zero.mxf -c:v mpeg2video -q:v 15 Video_9.mpg
```

Der Parameter „-q:v“ gab den Wertebereich für diesen Parameter vor. Dieser liegt zwischen 1-31, wobei ein höherer Wert eine geringere Bitrate und kleinere Dateigrößen erzielt. Hierbei handelt es sich um einen variablen Bitratenmodus. Dieser Schritt wurde nun mit verschiedenen Bitraten wiederholt. Anschließend wurden die Rohdaten mithilfe der Open-Source-Software „MediaArea.net“ ausgelesen und in eine Excel Tabelle übertragen (vgl. Abbildung 10).

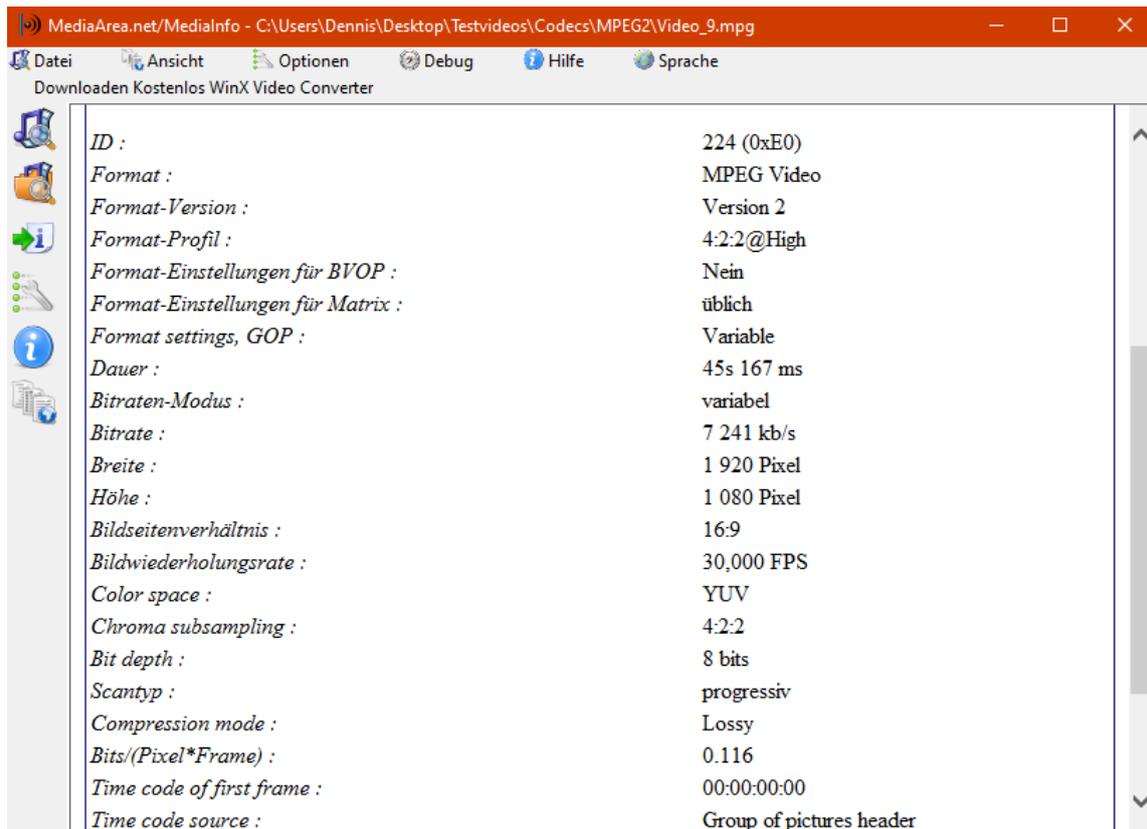


Abbildung 10: Parameter des MPEG-2 Videos nach erfolgter Kodierung

5.4.2 MPEG-4

Für den MPEG-4-Codec wurde der native Encoder „mpeg4“ genutzt. Dies hat den Vorteil, dass keine externe Bibliothek benutzt werden musste (Encode/MPEG-4 2023). Die Bitrate wurde hier mit dem Parameter „-q:v“ festgelegt. Der Wertebereich für diesen Parameter liegt ebenfalls zwischen 1-31, wobei ein höherer Wert eine geringere Bitrate und kleinere Dateigrößen erzielt. Hierbei handelte es sich ebenfalls um einen variablen Bitratenmodus. Als Container diente das AVI-Format. (vgl. Codeblock 3).

Codeblock 3: Kodierung zu MPEG-4
Quelle: Eigene Darstellung

```
1 ffmpeg -i ground_zero.mxf -c:v mpeg4 -q:v 10 Video_1.avi -benchmark
```

Mithilfe des Parameters „-benchmark“ konnte die benötigte Zeit der Kodierung ausgegeben werden, die für die Auswertung erforderlich war.

5.4.3 H.264

Für den Vorgang der H.264-Konvertierung wurde die in Codeblock 4 dargestellte Codezeile genutzt:

Codeblock 4: Kodierung zu H.264
Quelle: Eigene Darstellung

```
1 ffmpeg -i ground_zero.mxf -c:v libx264 -video_track_timescale 30  
2 -crf 30 Video_1.mp4
```

Der Constant Rate Factor (CRF) ermöglicht es dem Encoder, eine spezifische Ausgabequalität für die gesamte Datei anzustreben, wenn die Dateigröße weniger relevant ist. Dies führt zu maximaler Kompressionseffizienz in einem Durchlauf. Durch Anpassen des sogenannten Quantisierers für jedes Frame erhält er die benötigte Bitrate, um das gewünschte Qualitätsniveau aufrechtzuerhalten. Ebenfalls ist es möglich, eine Vorlage aus „ultrafast“, „superfast“, „veryfast“, „faster“, „fast“, „medium“, „slow“, „slower“, „veryslow“ sowie „placebo“ zu wählen. Diese beeinflussen sowohl die Kompressionseffizienz als auch die Codierungsgeschwindigkeit.

Für einen besseren Vergleich des Materials nach dem Kodiervorgang wurde hier für jede Kodierung die Voreinstellung „medium“ benutzt. Somit konnte der Parameter weggelassen werden. Mit der Benennung der Ausgabedatei wurde außerdem der Container MP4 vorgegeben (vgl. *Encode/H.264* 2023). Der Parameter „-video_track_timescale“ gab dem Container die Zeitskala vor. Da diese für einen Vergleich mit dem Ground-Zero-Video zur korrekten Synchronisation übereinstimmen mussten, wurden sie hier manuell auf 30 Frames pro Sekunde gesetzt.

5.4.4 H.265

Im Folgenden wird erläutert, wie die Videos mithilfe des H.265-Codecs konvertiert werden konnten. Um diese dann anschließend gut mit dem Vorgänger, dem H.264 Codec, vergleichen zu können, wurde auch hier der CRF-Faktor genutzt (vgl. Codeblock 5).

Codeblock 5: Codierung zu H.265
Quelle: Eigene Darstellung

```
1 ffmpeg -i ground_zero.mxf -c:v libx265 -crf 50 -video_track_timescale 30  
2 Video_1.mp4
```

Auch hier diente der Parameter „-c:v libx265“ als Befehl für das Programm, den H.265 Codec zu verwenden.

Mithilfe der MediaArea konnten die Daten des neu kodierten Videos zur Kontrolle ausgelesen werden (vgl. Abbildung 11).

Allgemein	
<i>Vollständiger Name :</i>	C:\Users\Dennis\Desktop\Testvideos\Codecs\h.265\animiert\Video_5.mp4
<i>Format :</i>	MPEG-4
<i>Format-Profil :</i>	Base Media
<i>Codec-ID :</i>	isom (isom/iso2/mp41)
<i>Dateigröße :</i>	36,6 MiB
<i>Dauer :</i>	45s 67 ms
<i>Gesamte Bitrate :</i>	6 804 kb/s
<i>Bildwiederholungsrate :</i>	30,000 FPS
<i>Kodierendes Programm :</i>	Lavf60.4.100

Video	
<i>ID :</i>	1
<i>Format :</i>	HEVC
<i>Format/Info :</i>	High Efficiency Video Coding
<i>Format-Profil :</i>	Format Range@L4@Main
<i>Codec-ID :</i>	hev1
<i>Codec-ID/Info :</i>	High Efficiency Video Coding
<i>Dauer :</i>	45s 67 ms
<i>Bitrate :</i>	6 800 kb/s
<i>Breite :</i>	1 920 Pixel
<i>Höhe :</i>	1 080 Pixel
<i>Bildseitenverhältnis :</i>	16:9
<i>Modus der Bildwiederholungsrate :</i>	konstant
<i>Bildwiederholungsrate :</i>	30,000 FPS
<i>Color space :</i>	YUV
<i>Chroma subsampling :</i>	4:2:2
<i>Bit depth :</i>	10 bits

Abbildung 11: Parameter des H.265 Videos nach erfolgter Kodierung

5.4.5 VP9

Für die Umwandlung der Videos zu einem VP9-Codec wurde folgender Code genutzt:

Codeblock 6: Kodierung zu VP9
Quelle: Eigene Darstellung

```
1 ffmpeg -i ground_zero.mxf -c:v libvpx-vp9 -quality realtime -cpu-used 4 -
2 row-mt 1 -crf 30 -b:v 0 Video_1.webm
```

Hier wählte erneut der Parameter „-c:v libvpx-vp9“ den VP9-Video-Codec für die Videoausgabe aus (vgl. Codeblock 6). Auch hier gab es eine Möglichkeit, das Video für bessere Vergleichbarkeit ähnlich wie bei dem H.265 mit einem CRF-Wert zu konvertieren. Um diesen Modus zu aktivieren, wurde eine Kombination aus dem „-crf“-Parameter und dem „-b:v“-Parameter verwendet. Für einen schnelleren Kodierprozess wurden außerdem Parameter wie „-cpu-used“ und „row-mt“ verwendet. Letzterer sorgt für eine Unterstützung von Multithreading (vgl. Encode/VP9 – FFmpeg 2023). Die Ausgabe erfolgte mit dem Container webm. Nach dem Kodierungsprozess ließen sich alle wichtigen Informationen aus der Kommandozeile ablesen (vgl. Abbildung 12).

```

Input #0, mxf, from 'ground_zero_animiert.mxf':
Metadata:
  operational_pattern_ul: 060e2b34.04010101.0d010201.01010100
  uid : 51327e71-3112-11ee-bd97-00d861c10601
  generation_uid : 51327e72-3112-11ee-848c-00d861c10601
  company_name : Adobe Inc.
  product_name : Premiere Pro
  product_version_num: 2.0.0.0.1
  product_version : 23.2.0
  application_platform: win32
  product_uid : 0c3919fe-46e8-11e5-a151-feff819cdc9f
  toolkit_version_num: 5.3.6.0.1
  modification_date: 2023-08-02T08:55:23.648000Z
  material_package_umid: 0x060A2B340101010501010D11130000008A2BEA01580106A5726B00D861C10601
  timecode : 00:03:51:14
Duration: 00:00:45.07, start: 0.000000, bitrate: 214756 kb/s
Stream #0:0: Video: prores (HQ) (apch / 0x68637061), yuv422p10le(bt709, progressive), 1920x1080, SAR 1:1 DAR 16:9, 30 tbr, 30 tbn
Metadata:
  file_package_umid: 0x060A2B34010105010105121308D40A852BEA01580106A50B9100D861C10601
  file_package_name: Source Package
  track_name : Track 1
Stream mapping:
  Stream #0:0 -> #0:0 (prores (native) -> vp9 (libvpx-vp9))
Press [q] to stop, [?] for help
[libvpx-vp9 @ 000002817b22fe00] v1.13.0-97-g5fae248f2
Output #0, webm, to 'Video_8.webm':
Metadata:
  operational_pattern_ul: 060e2b34.04010101.0d010201.01010100
  uid : 51327e71-3112-11ee-bd97-00d861c10601
  generation_uid : 51327e72-3112-11ee-848c-00d861c10601
  timecode : 00:03:51:14
  material_package_umid: 0x060A2B340101010501010D11130000008A2BEA01580106A5726B00D861C10601
  product_version_num: 2.0.0.0.1
  modification_date: 2023-08-02T08:55:23.648000Z
  application_platform: win32
  product_uid : 0c3919fe-46e8-11e5-a151-feff819cdc9f
  toolkit_version_num: 5.3.6.0.1
  encoder : Lavf60.4.100
Stream #0:0: Video: vp9, yuv422p10le(tv, bt709, progressive), 1920x1080 [SAR 1:1 DAR 16:9], q=2-31, 30 fps, 1k tbn
Metadata:
  file_package_umid: 0x060A2B34010105010105121308D40A852BEA01580106A50B9100D861C10601
  file_package_name: Source Package
  track_name : Track 1
  encoder : Lavc60.6.100 libvpx-vp9
Side data:
  cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: N/A
frame= 1352 fps=1.6 q=30.0 Lsize= 202475kB time=00:00:45.03 bitrate=36832.4kbits/s speed=0.0546x
video:202462kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.006141%

```

Abbildung 12: FFmpeg nach erfolgreicher VP9-Kodierung

5.4.6 AV1

Die Konvertierung der Videos zu einem AV1-Codec war ebenfalls mit FFmpeg möglich. Hierfür gibt es neben einem Constant-Quality-Modus (CQ) und einem Constrained-Quality-Modus (QC) ebenfalls den CRF- Modus.

“Ähnlich wie CRF in x264 und x265 versucht diese Methode der Datenrate-Steuerung sicherzustellen, dass jeder Frame die Anzahl der Bits erhält, die er verdient, um ein bestimmtes (wahrnehmbares) Qualitätsniveau zu erreichen.“ (Encode/AV1 – FFmpeg 2023).

Codeblock 7: Kodierung zu AV1

Quelle: Eigene Darstellung

```

1 ffmpeg -i ground_zero.mxf -c:v libsvtav1 -video_track_timescale 30
2 -svtav1-params fast-decode=1 -preset 4 -crf 25 Video_7.mp4 -benchmark

```

Außerdem wird hier der “libsvtav1“-Encoder genutzt (vgl. Codeblock 7). SVT-AV1 (libsvtav1) ist ein Encoder, der ursprünglich von Intel in Zusammenarbeit mit Netflix entwickelt wurde. Im Jahr 2020 wurde SVT-AV1 von AOMedia übernommen und dient als Grundlage für die zukünftige Entwicklung von AV1 sowie für zukünftige Codec-Bemühungen. Der Encoder unterstützt eine breite Palette von Geschwindigkeits-Effizienz-Abwägungen und skaliert recht gut auf vielen CPU-Kernen (vgl. Encode/AV1 – FFmpeg 2023). Ähnlich wie bei VP9 war es hier möglich, mit zusätzlichen Parametern wie

„fast-decode=1“ eine bessere Auslastung der CPU zu erreichen, um den Kodiervorgang zu beschleunigen.

```
Input #0, mxf, from 'ground_zero_animiert.mxf':
Metadata:
  operational_pattern_ul: 060e2b34.04010101.0d010201.01010100
  uid : 51327e71-3112-11ee-bd97-00d861c10601
  generation_uid : 51327e72-3112-11ee-848c-00d861c10601
  company_name : Adobe Inc.
  product_name : Premiere Pro
  product_version_num: 2.0.0.1
  product_version : 23.2.0
  application_platform: win32
  product_uid : 0c3919fe-46e8-11e5-a151-feff819cdc9f
  toolkit_version_num: 5.3.6.0.1
  modification_date: 2023-08-02T08:55:23.648000Z
  material_package_umid: 0x060A2B3401010501010D11130000008A2BEA01580106A5726B00D861C10601
  timecode : 00:03:51:14
Duration: 00:00:45.07, start: 0.000000, bitrate: 214756 kb/s
Stream #0:0: Video: prores (HQ) (apch / 0x68637061), yuv422p10le(bt709, progressive), 1920x1080, SAR 1:1 DAR 16:9, 30 tbr, 30
Metadata:
  file_package_umid: 0x060A2B34010105010105121308D40A852BEA01580106A50B9100D861C10601
  file_package_name: Source Package
  track_name : Track 1
File 'Video_1.mp4' already exists. Overwrite? [y/N] y
Stream mapping:
  Stream #0:0 -> #0:0 (prores (native) -> av1 (libsvtav1))
Press [q] to stop, [?] for help
Svt[info]: -----
Svt[info]: SVT [version]: SVT-AV1 Encoder Lib v1.4.1-98-ge06b3f2f
Svt[info]: SVT [build] : GCC 12.2.0 64 bit
Svt[info]: LIB Build date: Mar 5 2023 12:37:19
Svt[info]: -----
Svt[info]: Number of logical cores available: 12
Svt[info]: Number of PPCS 135
Svt[info]: [asm level on system : up to avx2]
Svt[info]: [asm level selected : up to avx2]
Svt[info]: -----
Svt[info]: SVT [config]: main profile tier (auto) level (auto)
Svt[info]: SVT [config]: width / height / fps numerator / fps denominator : 1920 / 1080 / 30 / 1
Svt[info]: SVT [config]: bit-depth / color format : 10 / YUV420
Svt[info]: SVT [config]: preset / tune / pred struct : 10 / PSNR / random access
Svt[info]: SVT [config]: gop size / mini-gop size / key-frame type : 161 / 32 / key frame
Svt[info]: SVT [config]: BRC mode / rate factor : CRF / 35
Svt[info]: -----
Output #0, mp4, to 'Video_1.mp4':
Metadata:
  operational_pattern_ul: 060e2b34.04010101.0d010201.01010100
  uid : 51327e71-3112-11ee-bd97-00d861c10601
  generation_uid : 51327e72-3112-11ee-848c-00d861c10601
  timecode : 00:03:51:14
  material_package_umid: 0x060A2B3401010501010D11130000008A2BEA01580106A5726B00D861C10601
  product_version_num: 2.0.0.1
  modification_date: 2023-08-02T08:55:23.648000Z
  application_platform: win32
  product_uid : 0c3919fe-46e8-11e5-a151-feff819cdc9f
  toolkit_version_num: 5.3.6.0.1
  encoder : Lavf60.4.100
Stream #0:0: Video: av1 (av01 / 0x31307661), yuv420p10le(tv, bt709, progressive), 1920x1080 [SAR 1:1 DAR 16:9], q=2-31, 30 fps
Metadata:
  file_package_umid: 0x060A2B34010105010105121308D40A852BEA01580106A50B9100D861C10601
  file_package_name: Source Package
  track_name : Track 1
  encoder : Lavc60.6.100 libsvtav1
frame= 1352 fps= 40 q=35.0 lsize= 139344kB time=00:00:45.03 bitrate=25348.1kbits/s speed=1.33x
video:139337kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.005336%
bench: utime=339.094s stime=15.141s rtime=33.761s
bench: maxrss=4041320kB
PS C:\Users\Dennis\Desktop\Testvideos\Codecs\AV1\animiert>
```

Abbildung 13: FFmpeg nach erfolgreicher AV1-Kodierung

In der Abbildung 13 ist nun die Kommandozeile nach erfolgter AV1-Kodierung zu sehen. In den letzten fünf Zeilen lassen sich auch die benötigte Kodierzeit, sowie die Größe und erreichte Bitrate ablesen.

5.4.7 Ermittlung des PSNR-Wertes

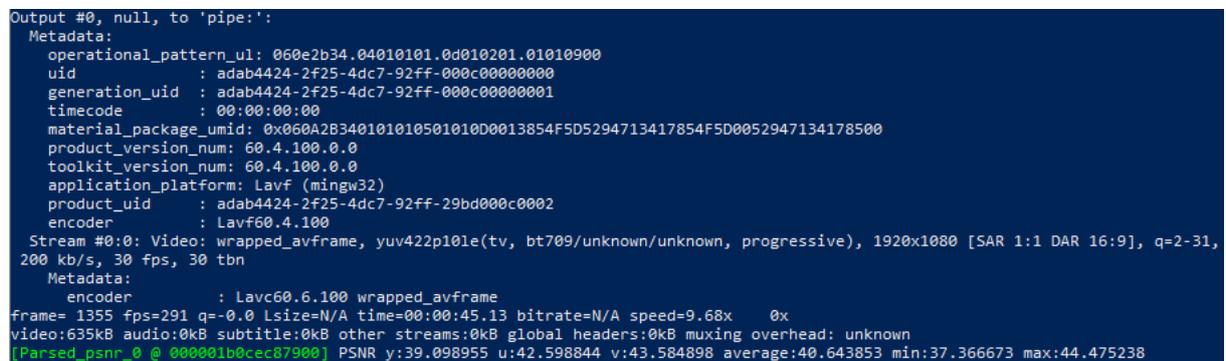
Um den PSNR-Wert zweier Videos mit FFmpeg zu berechnen, wurde der PSNR-Filter verwendet. Der Befehl sieht anhand eines konkreten Beispiels wie folgt aus:

Codeblock 8: Errechnung des PSNR-Wertes
Quelle: Eigene Darstellung

```
1 ffmpeg -i ground_zero.mxf -i Video_5.mp4 -lavfi psnr -f null -
```

Hierbei wurde mit dem ersten „-i“-Parameter das Originalvideo angegeben. Mit dem zweiten „-i“-Parameter wurde das komprimierte Video angegeben, während „-lavfi psnr“ den Filter auswählte.

Der letzte Parameter „-f null -“ sorgte für eine direkte Ausgabe der Werte auf der Konsole (vgl. Abbildung 14).



```
Output #0, null, to 'pipe':
  Metadata:
    operational_pattern_ul: 060e2b34.04010101.0d010201.01010900
    uid : adab4424-2f25-4dc7-92ff-000c00000000
    generation_uid : adab4424-2f25-4dc7-92ff-000c00000001
    timecode : 00:00:00:00
    material_package_umid: 0x060A2B340101010501010D0013854F5D5294713417854F5D00052947134178500
    product_version_num: 60.4.100.0.0
    toolkit_version_num: 60.4.100.0.0
    application_platform: Lavf (mingw32)
    product_uid : adab4424-2f25-4dc7-92ff-29bd000c0002
    encoder : Lavf60.4.100
  Stream #0:0: Video: wrapped_avframe, yuv422p10le(tv, bt709/unknown/unknown, progressive), 1920x1080 [SAR 1:1 DAR 16:9], q=2-31, 200 kb/s, 30 fps, 30 tbn
  Metadata:
    encoder : Lavc60.6.100 wrapped_avframe
frame= 1355 fps=291 q=-0.0 lsize=N/A time=00:00:45.13 bitrate=N/A speed=9.68x 0x
video:635kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
[Parsed_psnr_0 @ 000001b0cec87900] PSNR y:39.098955 u:42.598844 v:43.584898 average:40.643853 min:37.366673 max:44.475238
```

Abbildung 14: Ergebnis der PSNR-Werte

Im oben gezeigten Beispiel betrug der Durchschnittswert 40,64 dB. Außerdem wurden Minimal-, sowie Maximalwerte angezeigt. Dieser Schritt wiederholte sich für alle kodierten Videos. Dabei waren lediglich die Bezeichnungen der jeweiligen Videos und gegebenenfalls der Pfad korrekt anzupassen. Die Ergebnisse wurden für weitere Auswertungen in eine Excel-Datei übertragen und gespeichert.

Für eine genauere Analyse war es ebenfalls möglich, mit dem im Codeblock 9 gezeigten Befehl eine Logdatei zu erhalten, in der die PSNR-Werte für jedes einzelne Frame gespeichert wurden.

Codeblock 9: Anlegen einer Logdatei mit PSNR-Werten pro Frame
Quelle: Eigene Darstellung

```
1 ffmpeg.exe -i ground_zero.mxf -i Video_5.mp4 -lavfi
2 psnr=stats_file=psnr_logfile.txt -f null -
```

5.5 Durchführung eines Experteninterviews

Bevor die aus dem Experiment gewonnenen nun ausgewertet wurden, war es ebenfalls sinnvoll die aktuellen Geschehnisse aus der Medienwelt zu betrachten. Hierfür wurde im Rahmen dieser Arbeit ein Experteninterview durchgeführt. Simon Pontius, Geschäftsführer und IT-Consultant der Pontius Media GmbH, ist als spezialisierter Dienstleister für Audio-, Video-, Licht- und Servertechnik verantwortlich

für den professionellen Ablauf von Live-Streaming und Veranstaltungskonzepten aller Art. Es folgt eine wortgenaue Wiedergabe der Fragen und Antworten, die per E-Mail beantwortet wurden.

Die erste Frage lautete: *Welche Unterschiede gibt es bei dem Livestreaming im Gegensatz zu einer Aufzeichnung hinsichtlich der Datengröße, der Kodierung und der Bitrate?*

Simon Pontius antwortete darauf folgendes:

„Es gibt mehrere Unterschiede zwischen Livestreaming und der Aufzeichnung von Inhalten in Bezug auf Datengröße, Codierung und Bitrate.

1. *Datengröße, Kodierung und Bitrate:*

Livestreaming: Bei einem Livestream müssen die Daten in Echtzeit übertragen werden. Die Datengröße ist daher geringer, da nur die aktuellen Inhalte übertragen werden. Diese kann im schlechtesten Fall bei 2000 Kbit bei FHD-Auflösung mit 50FPS liegen. In den meisten Fällen wird mit 6000 Kbit gesendet. Bei der Wahl der Datenrate muss man beachten, dass diejenigen die sich den Stream ansehen diese Rate beim Download verbrauchen.

Aufzeichnung: Bei der Aufzeichnung können die Daten vor der Veröffentlichung komprimiert und optimiert werden, um eine bessere Qualität zu gewährleisten. Dies kann zu größeren Dateigrößen führen, da eine höhere Qualität beibehalten werden kann. Die Datenrate liegt zwischen 20.000 Kbit und 150.000 Kbit. Je nachdem ob es sich um eine reine Speicherung handelt oder einen hohe Rate für die spätere Bearbeitung benötigt wird.

2. *Verzögerung:*

Livestreaming: Livestreams haben normalerweise eine gewisse Verzögerung aufgrund der Kodierung, Übertragung und Decodierung der Daten in Echtzeit. Die Verzögerung kann je nach Plattform und Konfiguration variieren. Meist liegt diese zwischen 5 und 30 Sekunden.“

Die zweite Frage wurde wie folgt formuliert: *Welche Videostreaming-Protokolle verwenden Sie und warum?*

„RTMP und RTMPS nutze ich für alle gängigen Streaming Plattformen (YouTube, Twitch, Facebook). Die meisten Plattformen geben RTMP oder RTMPS vor und bieten keine alternative Übertragungsart an.

NDI wird bei uns ausschließlich im Intranet/eigenem Netzwerk angewendet. NDI wird von einigen neueren PTZ-Kameras bereits mitgeliefert und lässt sich leicht im Produktionsworkflow integrieren.“

Auf die dritte Frage „Welche Anforderungen an die Hardware gibt es?“ schrieb Simon Pontius folgendes:

„Es gibt Hardware und Software Encoder. Die Anforderungen variiert je nach gewünschten Sendeparametern. Bei der Wahl der Parameter muss man sich auf jeden Fall an die Übertragungsparameter der jeweiligen Empfängerplattform halten. Eine hohe Sendequalität ist manchmal durch die Plattform selbst blockiert. Zum Beispiel erlaubt Facebook nur 720p mit 30 FPS. Sendet man mehr Daten,

reduziert der Facebook Decoder den Stream. Das führt zu einem Qualitätsverlust, aber auch zu einer reduzierten Anforderung an Hardware. Bei einem 4K Live-Stream muss man bspw. mit deutlich höheren Werten arbeiten als bei einem 1080p oder 720p Stream. Die entsprechende Hard- oder Software muss damit umgehen können und wird meist durch die Auflösung und Framerate limitiert.“

Eine weitere Frage lautete: *Wie sehen Sie die Zukunft der Videocodierung hinsichtlich neuer Formate und besseren Kompressionsverfahren?*

„Es werden von unterschiedlichsten Herstellern neue Formate und Kompressionsverfahren entwickelt, da sich die Anforderungen mit immer besseren Endgeräten erhöhen. Ich würde daher erwarten, dass in den nächsten 5-10 Jahren 1-2 revolutionäre Kompressionsverfahren entwickelt werden, die die aktuell gängigen Verfahren ergänzen oder ersetzen.“

Ebenfalls gefragt wurde folgendes: *Ist die Nachfrage der Streaming-Aufträge in den letzten Jahren gestiegen?*

„Während der Corona Phase ab April 2020 bis Mitte 2022 war die Nachfrage nach Streaming Dienstleistern und Plattformen sehr hoch. Ende 2022 hat die Nachfrage nachgelassen und 2023 hat sich die Nachfrage auf einem niedrigen Niveau fortgesetzt.“

Die letzte Frage *„Wie hat sich die Entwicklung von Videocodecs und Streaming-Technologien in den letzten Jahren auf die Effizienz und Qualität von Videoübertragungen ausgewirkt?“* wurde von Simon Pontius wie folgt beantwortet:

„Einige Codecs existieren schon seit mehreren Jahren. Erst durch vereinfachte oder weggefallene Lizensierungen erlebten einige ältere Codecs einen Nachfrageboom. In den letzten Jahren hat es erhebliche Fortschritte in der Entwicklung von Videocodecs und Streaming-Technologien gegeben, die sich doch sehr auf die Effizienz und Qualität von Videoübertragungen ausgewirkt haben.“

6 Ergebnisse und Diskussion

6.1 Evaluierung der Videocodecs anhand festgelegter Kriterien

Ein zentraler Punkt des Vergleichs zwischen den Codecs ist die Effizienz ihrer Kodierung. In diesem Kapitel wurde auf die Qualität des komprimierten Videos und die visuellen Artefakte unter Berücksichtigung ihrer Bitrate eingegangen. Außerdem wurden die benötigte Zeit der Kodierung sowie die Größe der neu kodierten Dateien in Relation gesetzt. Dabei kamen sowohl die objektive Qualitätsmetrik PSNR als auch eine subjektive, visuelle Bewertung zum Einsatz. Das Vorgehen eines Vergleichs zweier Codecs lautete wie folgt:

1. Zunächst wurden die gesammelten PSNR-Werte im Zusammenhang mit der Bitrate anhand einer Grafik verglichen. Hierzu diente das in Kapitel 5.2.2 animierte Video, da dieses Video permanente Bewegungen enthielt und dadurch einen guten Extremtest der PSNR-Errechnung darstellte.
2. Als Nächstes folgte die visuelle Suche nach der Stelle eines Videos, welches bei immer niedriger werdender Bitrate als erstes Block-Artefakte aufwies. War diese Stelle gefunden, wurde der zugehörige, minimale PSNR-Wert sowie Dateigröße beider Videos miteinander verglichen. Hierfür eignete es sich, dass in Kapitel 5.2.1 produzierte Video zu verwenden, da durch die erst später einsetzende Bewegung im Video Block-Artefakte leichter zu erkennen waren. Für die möglichst genaue Ermittlung des zugehörigen PSNR-Wertes wurde der jeweilige Minimalwert genutzt.
3. Die Datengröße sowie die verwendete Rechenleistung wurde in Form von benötigter Kodierungszeit miteinander verglichen. Hierfür wurden drei Videos mit einem jeweils nahezu identischen PSNR-Durchschnittswert genutzt. So konnte sichergestellt werden, dass die verglichenen Videos ein etwa gleiches Maß an Qualität aufwiesen. Das Ursprungsmaterial ist dabei die neu kodierte Variante des Ground-Zero-animiert-Videos.

Durch die Analyse der Leistung dieser Metriken wurden die Effizienz und die Qualität der beiden Standards verglichen.

6.2 Vergleich einzelner Codecs

Um auch den Fortschritt der Videokompressionstechnologien besser verdeutlichen zu können, folgt nun ein Vergleich der Codecs mit seinem jeweiligen Nachfolger.

6.2.1 Vergleich von H.264 und H.265 (HEVC)

Der Vergleich der Codecs H.264 und H.265 ist hierbei von besonderem Interesse, da er Einblicke in die technologische Entwicklung des weit verbreiteten H.264-Codecs bietet.

Die Auswertung der aus dem Experiment gewonnenen Daten des animierten Videos ergab nun ein Ergebnis, welches als Punkt-(XY)-Diagramm dargestellt werden kann (vgl. Abbildung 15).

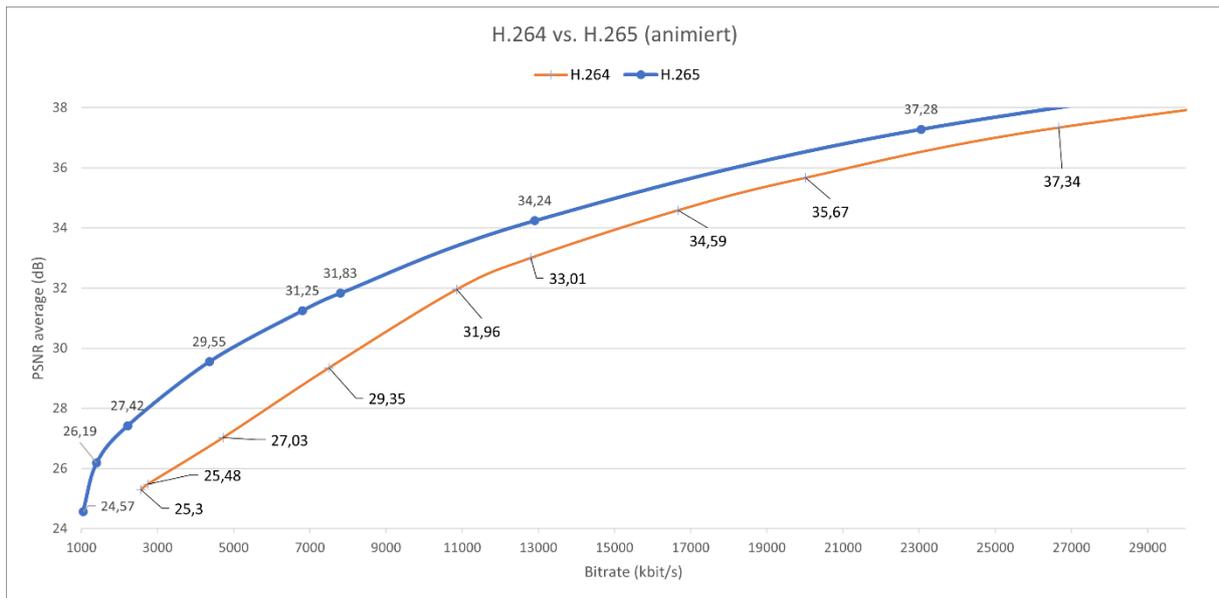


Abbildung 15: H.264 vs. H.265 (animiert)

Vergleicht man die Codecs nun visuell bei etwa gleicher Bitrate wird ebenfalls ein Unterschied deutlich. Im Folgenden wurden nun identische Frames zweier Videos gegenübergestellt.

Dabei enthält die linke Bildhälfte ein Video, welches mit dem H.264 Codec und einer Bitrate von 2196 kbit/s codiert wurde. In der rechten Bildhälfte ist ein H.265-codiertes Video mit einer Bitrate von 2143 kbit/s zu sehen.



Abbildung 16: H.264 vs. H.265 ohne Bewegungen

Der erste Frame, der genauer untersucht wurde, zeigt eine nahezu identische Qualität bei beiden Codecs (vgl. Abbildung 16). Die Szenerie hat an diesem Punkt des Videos so gut wie keine Bewegungen enthalten.

Verglich man aber nun einen Frame, der sehr schnelle Bewegungen aufwies, fiel das Ergebnis unterschiedlicher aus. Auf der linken Seite sind bei dem H.264-Codec bereits erste Blockartefakte zu sehen. Auf der anderen Seite gelingt es dem H.265-Codec auch bei der niedrigen Bitrate noch ein akzeptables Bild zu liefern und selbst die raue Oberfläche der Kreppbänder ist noch an einigen Stellen zu erkennen (vgl. Abbildung 17, 18, 19).

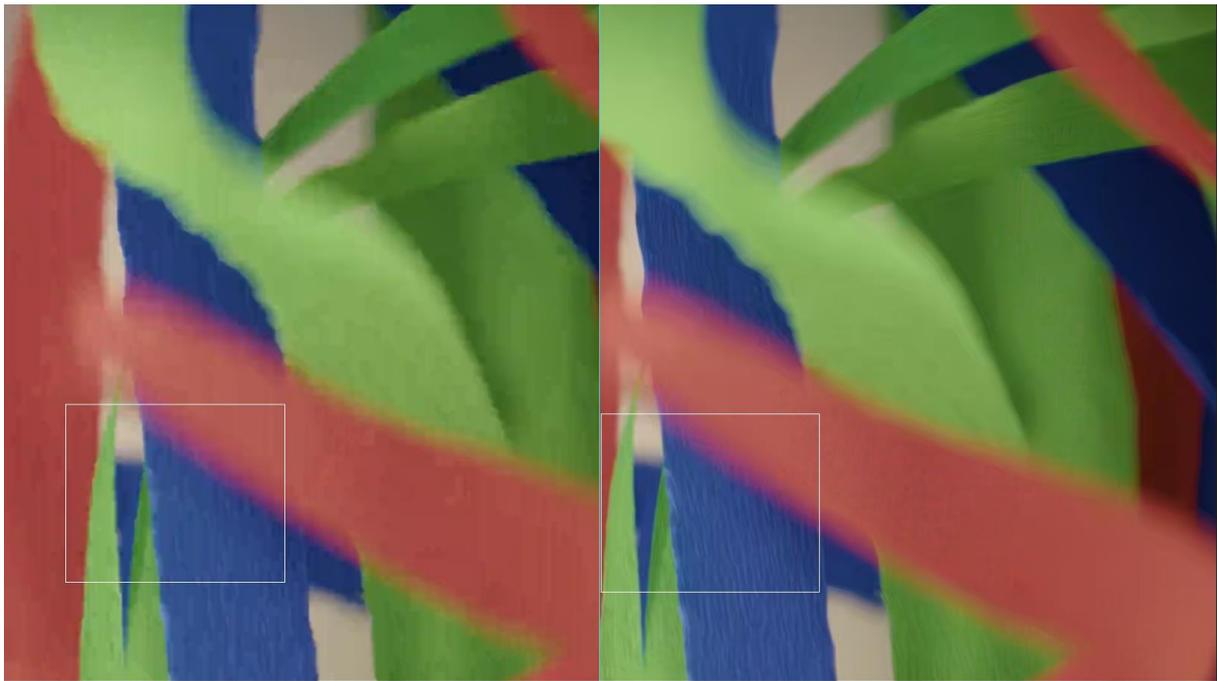


Abbildung 17: H.264 vs. H.265 mit Bewegungen

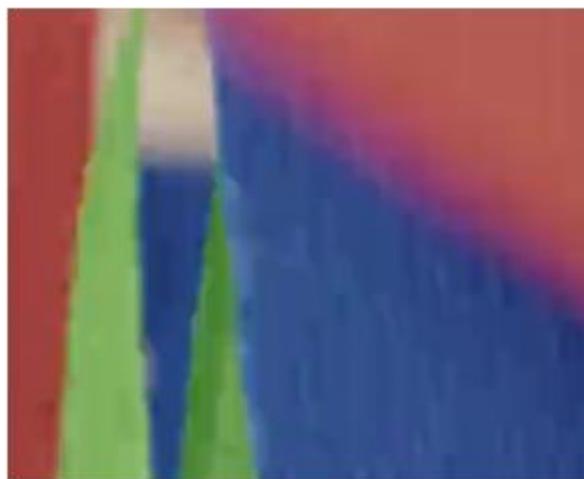


Abbildung 18: Vergrößerung von H.264

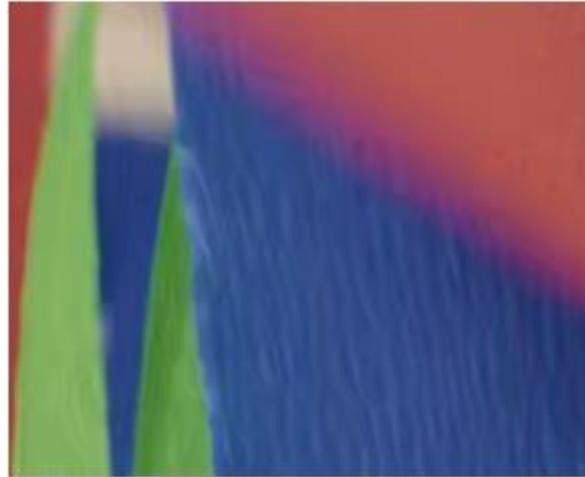


Abbildung 19: Vergrößerung von H.265

Der zugehörige PSNR-Minimalwert beträgt bei dem H.264-Video 39,27 dB und bei dem H.265-Video 41,05 dB. Ein etwa vergleichbarer PSNR-Wert des H.265 kodierten Videos von 39,39 dB wurde bei einer Bitrate von 1367 kBit/s erreicht (vgl. Abbildung 18).

Beim Betrachten der PSNR-Minimalwerte beider Codecs fiel auf, dass H.264 in der Bewertung deutlich früher abfällt. Der H.265 Codec schaffte es hingegen bei niedrigen Bitraten von weniger als 2000 kbit/s nur moderaten Qualitätsverlust zu erleiden, erlittete dann jedoch ab einer Bitrate von unter 500 kbit/s einen rapideren Abfall als sein Vorgänger H.264.

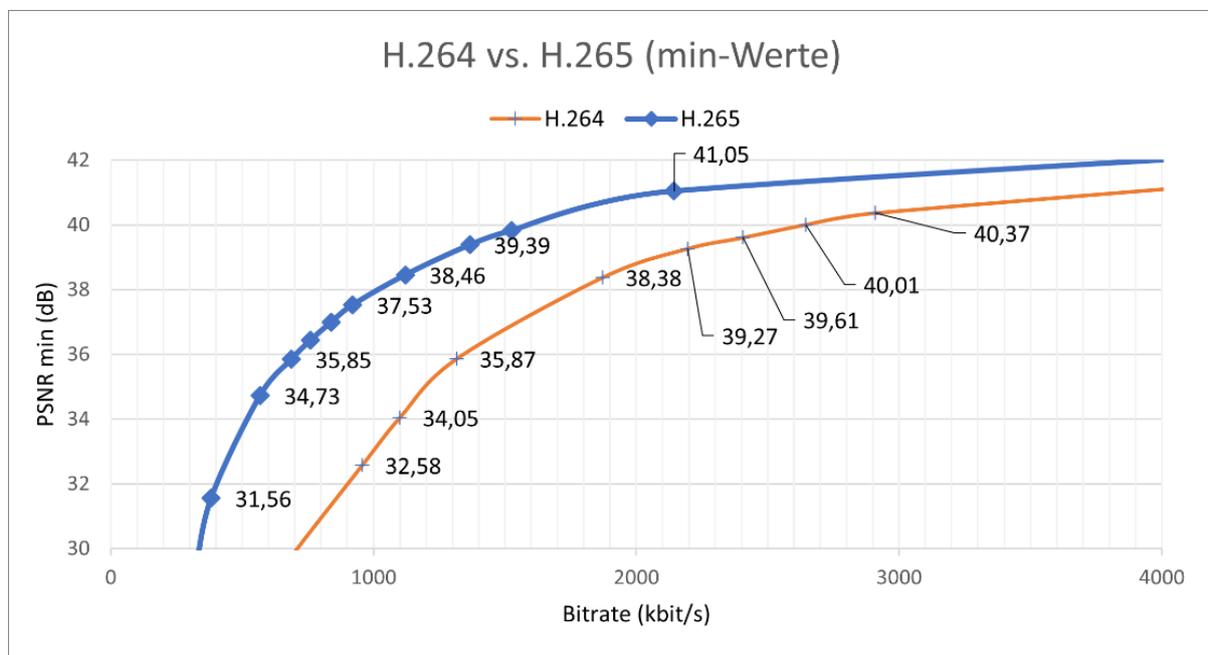


Abbildung 20: H.264 vs. H.265 (min-Werte)

In Tabelle 1 sind nun jeweils 3 Videos der beiden Codecs mit etwa gleicher Bildqualität anhand des PSNR-Wertes zu sehen. Als Datengrundlage diente das animierte Video. Schaut man sich nun die drei

Faktoren Länge der Kodierung, Größe des Exportvideos sowie die zugehörige Bitrate an, wird ein Muster deutlich. Der H.264-Codec erzielte klar schlechtere Ergebnisse. Bei etwa gleicher Bildqualität benötigte H.265 allerdings fast die dreifache Zeit für die Kodierung. Außerdem benötigte er dafür eine deutlich niedrigere Bitrate und hatte somit eine Größensparnis zwischen 30-50%.

Tabelle 1: Vergleich der Datengröße und der Länge der Codierung (H.264 vs. H.265)

<i>Codec</i>	<i>PSNR-Wert (dB)</i>	<i>Länge (s)</i>	<i>Größe (Mb)</i>	<i>Bitrate (kbit/s)</i>
H.265	40,29	176,48	215	40155
H.264	40,31	55,2	247	45007
H.265	29,55	87,49	23,4	4361
H.264	29,35	30,6	41,2	7503
H.265	34,24	120,98	69,2	12909
H.264	34,59	41,7	91,6	16672

6.2.2 Vergleich von MPEG-2 und MPEG-4

Der Vergleich der Videocodecs MPEG-2 und MPEG-4 spielt ebenfalls eine bedeutende Rolle in der digitalen Medienverarbeitung und -übertragung. Beide Standards sind Teil der MPEG-Familie und haben verschiedene Eigenschaften, die in unterschiedlichen Anwendungen von Bedeutung sind.

Auch hier wurden zunächst die durchschnittlichen PSNR-Werte des animierten Videos als Punkt-XY-Diagramm ausgewertet (vgl. Abbildung 21).

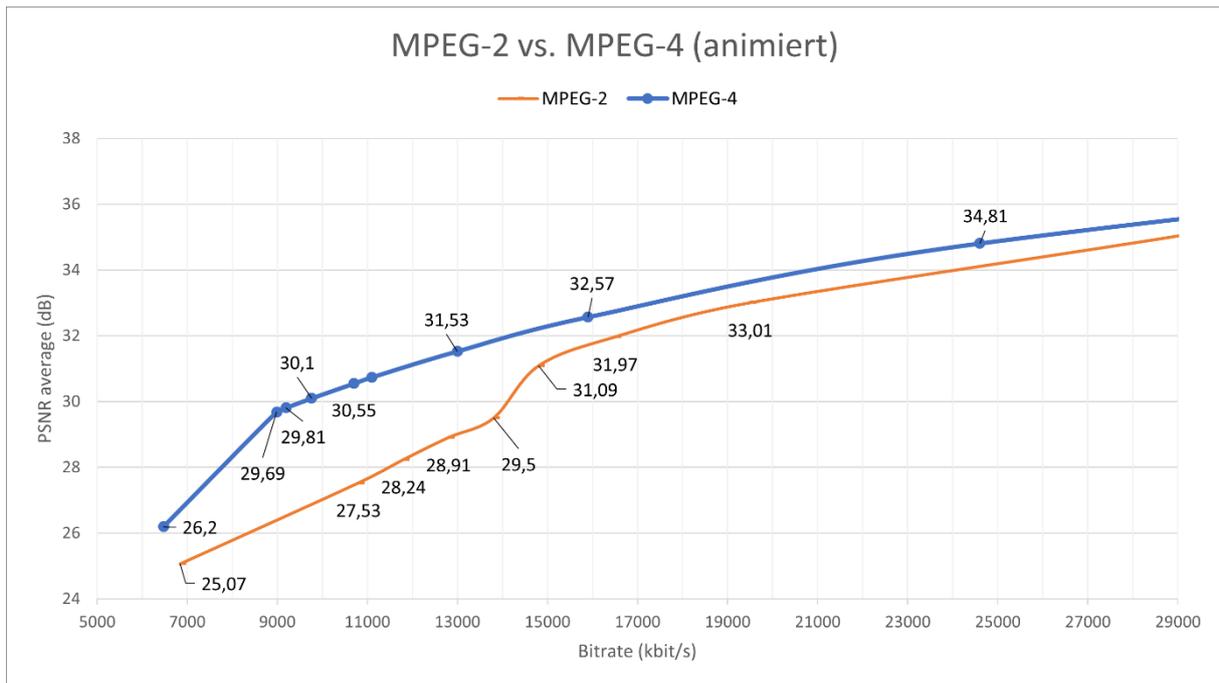


Abbildung 21: MPEG-2 vs. MPEG-4 (animiert)

Aus dieser Grafik erkennt man deutlich einen rapiden Abfall des MPEG-2-Codecs unter einer Bitrate von etwa 15000 kbit/s. Einen etwa ähnlichen Abfall zeigt sich hingegen bei dem MPEG-4-Codec erst bei einer Bitrate von unter 9000 kbit/s.

Auch bei diesem Vergleich wurde nun nach dem Video gesucht, welches bei etwa gleicher Bitrate als erstes Blockartefakte aufwies. Für diesen Vergleich wurden die kodierten Versionen des Ground-Zero-Videos nebeneinandergelegt und einzelne Frames verglichen.

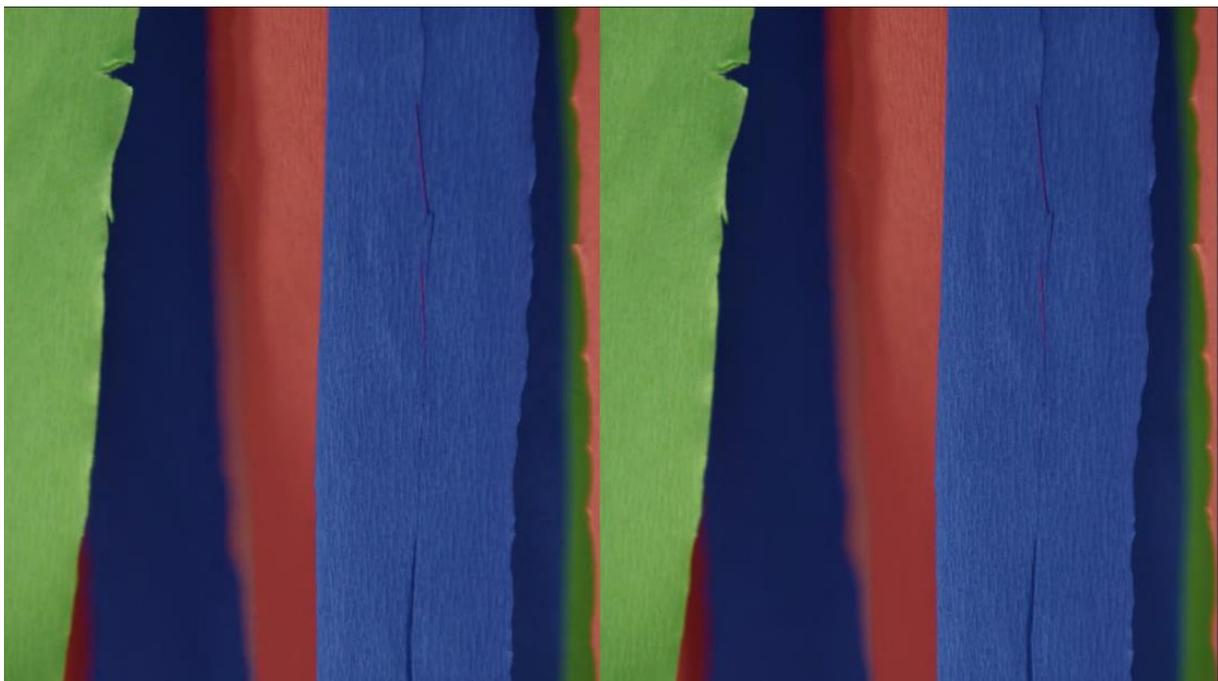


Abbildung 22: MPEG-2 vs. MPEG-4 ohne Bewegungen

In der linken linken Bildhälfte befindet sich ein Video, welches mit dem MPEG-2-Codec und einer Bitrate von 11,8 Mb/s kodiert wurde. In der rechten Bildhälfte ist ein MPEG4-kodiertes Video mit einer Bitrate von 10,6 Mb/s zu sehen (vgl. Abbildung 22). Auch hier gab es zum Zeitpunkt des ersten Frames kaum Bewegung im Video. Sobald nun im Video die Bewegungen begannen, fiel auch die Bildqualität bei dem MPEG-2-Codec deutlich ab und es waren Blockartefakte sichtbar (vgl. Abbildung 23).

Auch der MPEG-4 Codec schaffte es bei dieser Bitrate nicht mehr, die genauen Strukturen des Bildmaterials scharf darzustellen. Trotzdem war noch keine Bildung von Blockartefakten zu erkennen. Dadurch kam es zu einem späteren Abfall der PSNR-Wertung (vgl. Abbildung 25, 26).

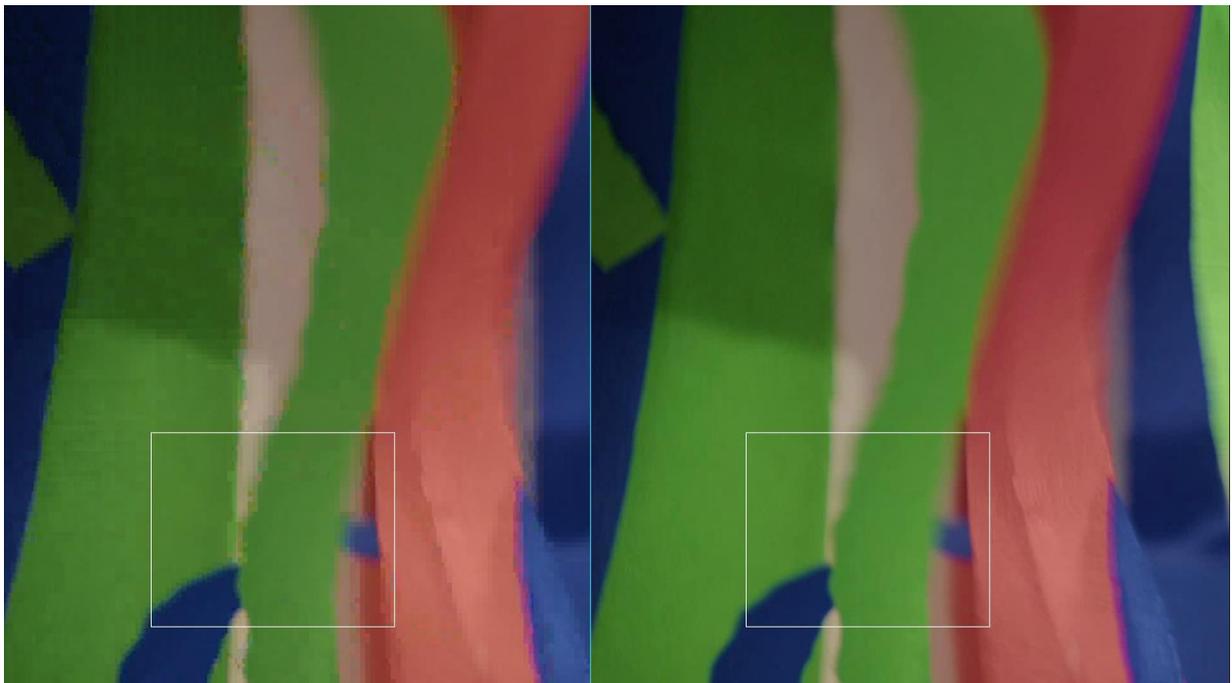


Abbildung 23: MPEG-2 vs. MPEG-4 mit Bewegungen

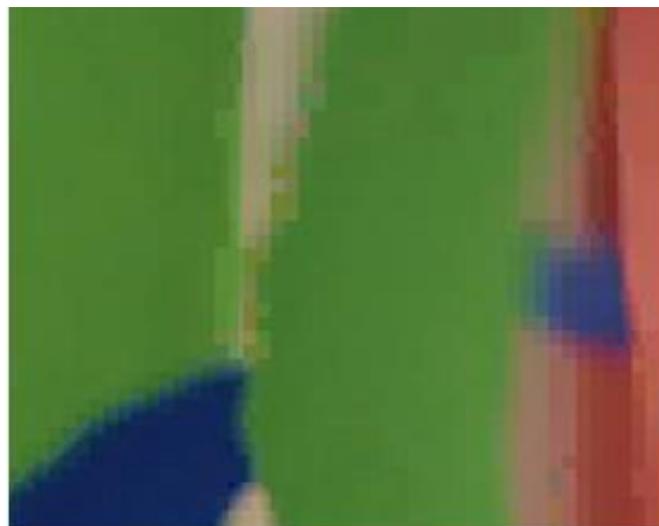


Abbildung 24: Vergrößerung von MPEG-2



Abbildung 25: Vergrößerung von MPEG-4

Der zugehörige PSNR-Minimalwert beträgt bei dem MPEG-2-Video 38,23 dB und bei dem MPEG-4-Video 42,73 dB. Ein etwa vergleichbarer PSNR-Wert des MPEG-4 kodierten Videos von 38,53 dB wurde erst bei einer Verringerung seiner Bitrate auf 5349 kbit/s erreicht (vgl. Abbildung 26).

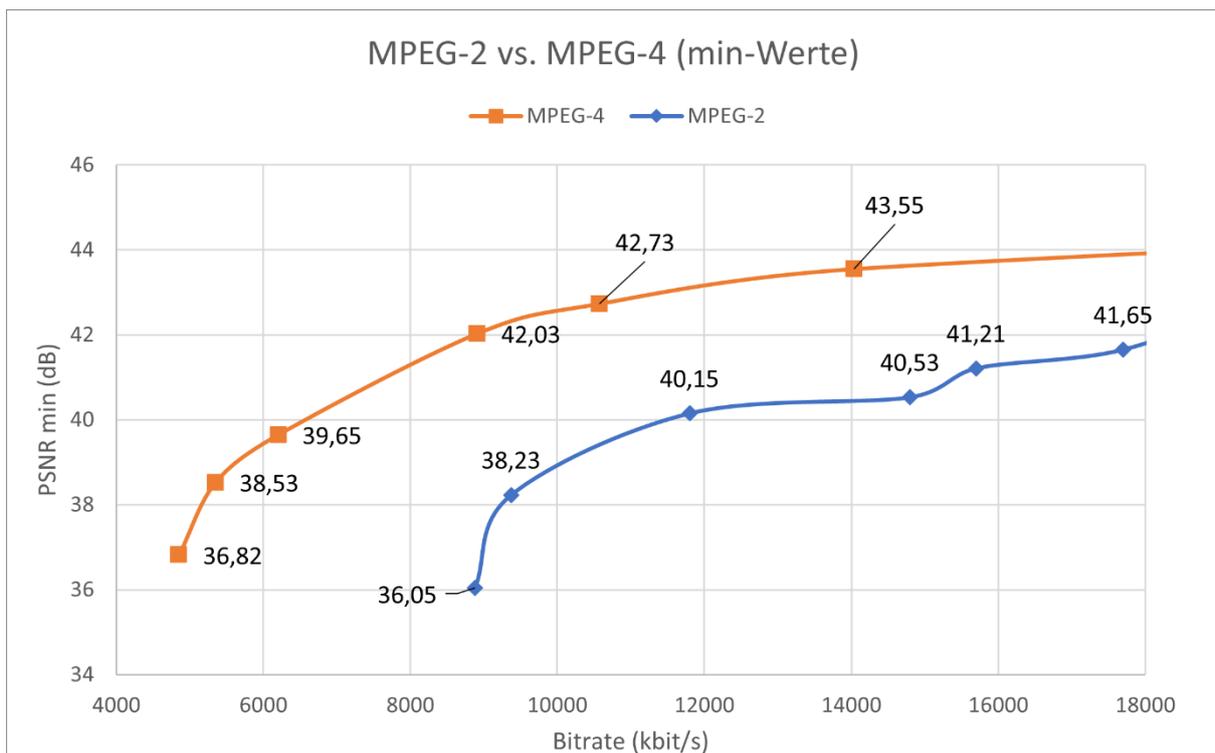


Abbildung 26: MPEG-2 vs. MPEG-4 (min-Werte)

Bei Betrachtung der Minimalwerte beider Codecs erkennt man eine deutliche Verschiebung. MPEG-4 konnte durch effizientere Nutzung der vorhandenen Bitrate ein besseres Bild erzeugen und erreichte den Punkt des sichtbaren Qualitätsverlustes erst bei einer Bitrate von etwa 5500 kbit/s.

In der Tabelle 2 sind nun erneut 3 Videos der beiden Codecs mit etwa gleicher Bildqualität anhand des PSNR-Wertes zu sehen. Wie auch im ersten Vergleich diente das animierte Video als Grundlage. Hierbei fiel auf, dass sowohl MPEG-2 als auch MPEG-4 nur sehr wenig Zeit für den Kodierungsvorgang benötigten. Die daraus resultierenden, verhältnismäßig großen Dateigrößen waren bei MPEG-4 im Durchschnitt etwa 25% kleiner. Auch die benötigte Bitrate für etwa den gleichen PSNR-Wert bei MPEG-4 sank um etwa 25%.

Tabelle 2: Vergleich der Datengröße und der Länge der Codierung (MPEG-2 vs. MPEG-4)

<i>Codec</i>	<i>PSNR-Wert (dB)</i>	<i>Länge (s)</i>	<i>Größe (Mb)</i>	<i>Bitrate (kbit/s)</i>
MPEG-4	29,69	4,98	49,5	8988
MPEG-2	29,50	5,89	80,4	13800
MPEG-4	31,53	5,67	71,4	13000
MPEG-2	31,97	6,02	92,8	16500
MPEG-4	34,81	6,07	135	24600
MPEG-2	35,15	6,45	165,5	29500

6.2.3 Vergleich von AV1 und VP9

Es folgt nun der Vergleich der Codecs VP9 und AV1. Beide Codecs repräsentieren moderne Technologien, die in der Lage sind, selbst hochauflösende Videos mit vergleichbar geringer Datenmenge zu konvertieren. Da es sich bei dem AV1-Codec um den Nachfolger des VP9-Codecs handelt, ist auch dieser Vergleich eine gute Bewertungsgrundlage für dessen Weiterentwicklung.

Beim Betrachten der Durchschnittswerte der PSNR-Bewertung war ein etwa gleicher Qualitätsabfall bei beiden Codecs erkennbar, der jedoch bei dem AV1-Codec erst bei wesentlich kleineren Bitraten erzielt wurde. Hier wird eine effizientere Kodierung des neueren AV1-Codecs deutlich (vgl. Abbildung 27).

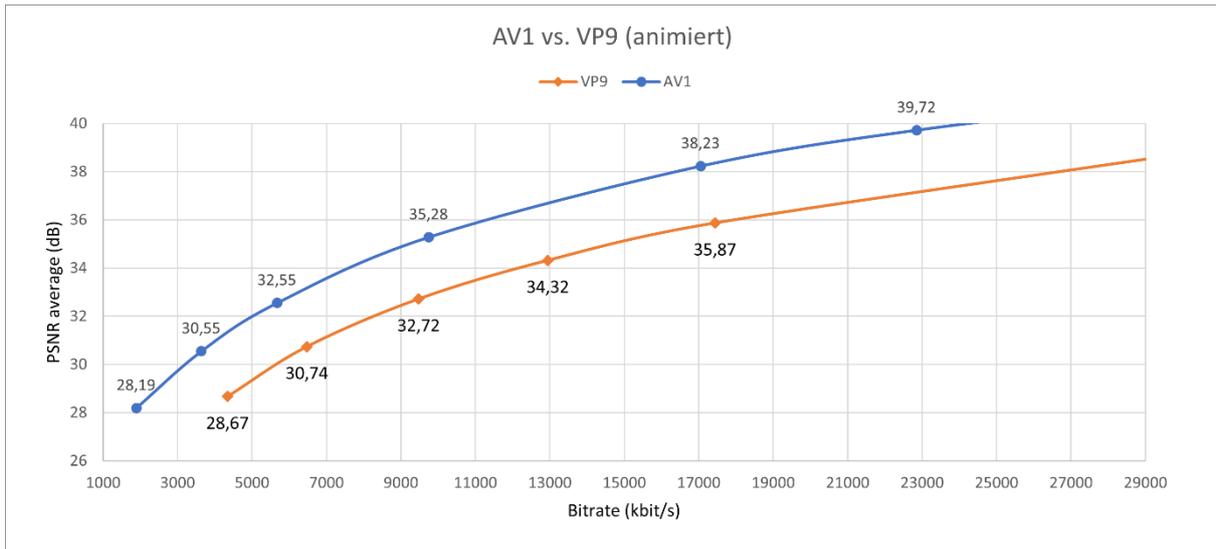


Abbildung 27: AV1 vs. VP9 (animiert)

Wie bei vorherigen Vergleichen wurden wieder beide Codecs mit etwa gleicher Bitrate gegenüber gestellt. Als Datengrundlage diente das Ground-Zero-Video. In der linken Bildhälfte befindet sich ein AV1-Video mit einer Bitrate von 1012 kbit/s. Rechts zu sehen ist ein VP9-Video mit einer Bitrate von 1006 kbit/s. Die Qualität ist in der ruhigen Szene nahezu identisch (vgl. Abbildung 28).



Abbildung 28: AV1 vs. VP9 ohne Bewegungen

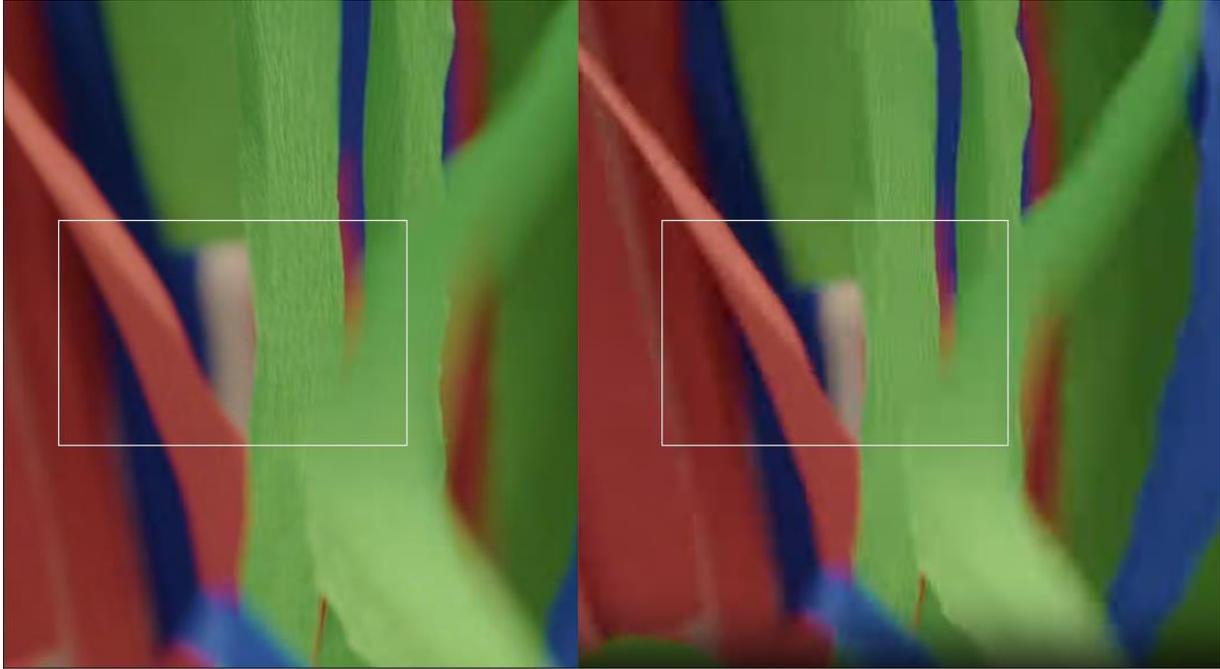


Abbildung 29: AV1 vs. VP9 mit Bewegungen

Sieht man sich nun eine Stelle im Video mit höheren Frequenzen an, erkennt man die Bildung von Blockartefakten im Bereich des roten Bandes bei dem VP9-Codec, wohingegen bei dem AV1-Codec noch genaue Strukturen des grünen Bandes zu erkennen sind (vgl. Abbildung 29, 30, 31).



Abbildung 30: Vergrößerung von AV1



Abbildung 31: Vergrößerung von VP9

Der zugehörige PSNR-Minimalwert der Videos lag für das AV1-Video bei 41,04 dB und für das VP9-Video bei 39,12 dB (vgl. Abbildung 32).

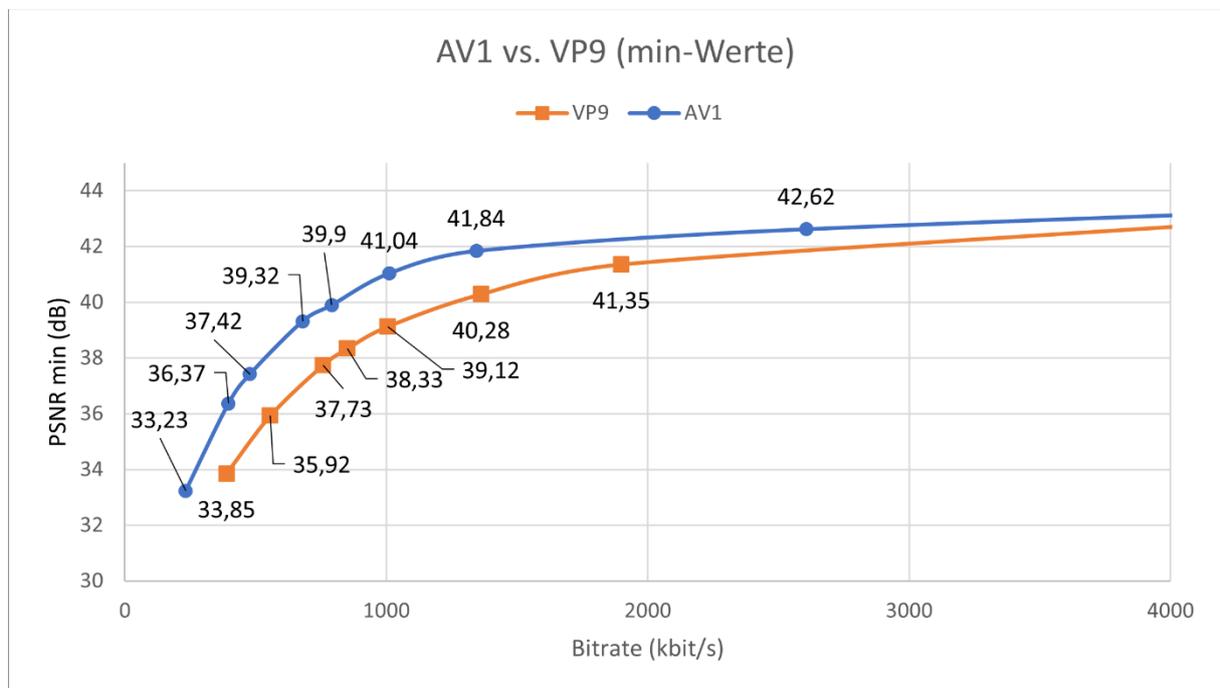


Abbildung 32: AV1 vs. VP9 (min-Werte)

Ähnlich wie bei dem Vergleich der Durchschnittswerte aus Abbildung 27 erkennt man auch bei den PSNR-Minimalwerten einen etwa ähnlichen Abfall der Bildqualität, der bei dem AV1-Codec erst bei niedrigeren Bitraten begann.

Bei Betrachtung der Daten aus Tabelle 3 fällt auf, dass sowohl der AV1-Codec deutlich mehr Zeit für den Kodierungsvorgang benötigten als alle bisher getesteten Codecs. VP9 ist mit der benötigten Kodierzeit in etwa so schnell wie der H.265-Codec.

Tabelle 3: Vergleich der Datengröße und der Länge der Codierung (AV1 vs. VP9)

<i>Codec</i>	<i>PSNR-Wert (dB)</i>	<i>Länge (s)</i>	<i>Größe (Mb)</i>	<i>Bitrate (kbit/s)</i>
VP9	39,13	157,97	174,5	31764
AV1	39,72	1132	125,6	22853
VP9	35,87	138,94	95,8	17435
AV1	35,28	1147,5	53,6	9749
VP9	28,67	108,26	24,1	4340
AV1	28,19	547,7	10,4	1892

Diese langen Kodierzeiten haben relativ zu ihrer Qualität kleine Dateien zufolge. Ebenso fällt auf, dass die Effizienz des AV1-Codecs mit sinkender Bildqualität größer ist. So benötigte er für einen durchschnittlichen PSNR-Wert von etwa 39 dB eine um etwa 28% verringerte Datenbitrate gegenüber VP9, wohingegen für einen PSNR-Wert von etwa 28 dB eine um etwa 56% verringerte Datenbitrate benötigt wird (vgl. Tabelle 3).

7 Zusammenfassung und Schlussfolgerungen

Die aus dem Experiment gewonnenen Daten und deren Analyse lieferten einen spannenden Einblick und zeigten, wie sich die einzelnen Codecs in Laufzeit der Kodierung, Datengröße und vor allem der Qualität unterscheiden. Dies wird auch bei Betrachtung der wichtigsten Ergebnisse deutlich.

7.1 Zusammenfassung der wichtigsten Ergebnisse

Verglich man die durchschnittlichen PSNR-Werte aller Codecs miteinander, war eine deutliche Steigerung in der Qualität bei immer kleiner werdender Bitrate zu erkennen. Dabei hat AV1 wie erwartet die besten Ergebnisse erzielt. Bei allen Codecs begann eine sichtbare Blockartefakt-Bildung und der damit verbundene Qualitätsabfall etwa bei einem PSNR-Minimalwert zwischen 38 dB und 39 dB.

Die aus Kapitel 3.3.1 sowie Kapitel 3.3.2 erwarteten Ergebnisse der Codecs H.265, H.265 und VP9 sind ungefähr eingetreten. So hatte H.265 gegenüber dem VP9-Codec eine Bitratensparnis von 32,6%. Außerdem zeigte sich, dass die etwa doppelte Kompressionseffizienz gegenüber H.264 ebenso bewahrt ist, gemessen an der Datengröße (vgl. Tabelle 4).

Auch die in Kapitel 3.3.3 erwähnten Ergebnisse des AV1-Codecs zeichnen sich hier ab. Der AV1-Codec benötigte für die gleiche Bildqualität deutlich mehr Zeit. Die Bitratensparnis gegenüber VP9 lag sogar deutlich über den erwarteten Ergebnissen.

Tabelle 4: Vergleich aller Codecs (sortiert nach abfallender Bitrate)

Codec	PSNR-Wert (dB)	Länge (s)	Größe (Mb)	Bitrate (kbit/s)
MPEG-2	30,13	5,89	83	16500
MPEG-4	30,1	5,15	53,7	9763
H.264	29,35	30,6	41,2	7503
VP9	30,74	110,58	35,6	6473
H.265	29,55	87,49	23,4	4361
AV1	30,55	832,7	20	3640

Auch beim Betrachten der durchschnittlichen PSNR-Werte aller Codecs bei steigender Bitrate ist ein deutliches Bild erkennbar. AV1 lieferte als neuester Codec über alle Bitraten hinweg das beste Bild. Dahinter erzielten VP9 und H.265 zusammen ein in etwa gleiches Qualitätsmaß (vgl. Abbildung 33).

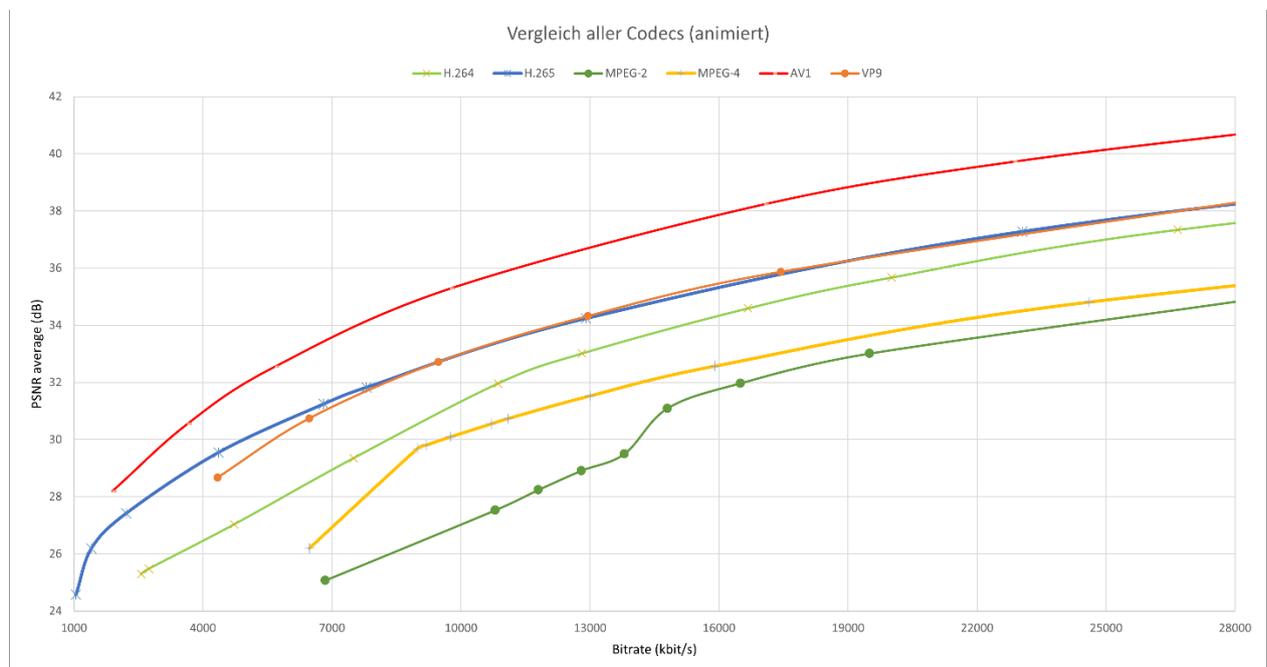


Abbildung 33: Vergleich aller Codecs (animiert)

Deutlich schlechtere Werte erzielten hingegen die Codecs MPEG-4 und MPEG-2, wobei MPEG-4 eine Bitratensparnis von etwa 25% gegenüber seinem Vorgänger erzielt.

In Abbildung 34 sind nun die Minimalwerte aller Codecs zu sehen:

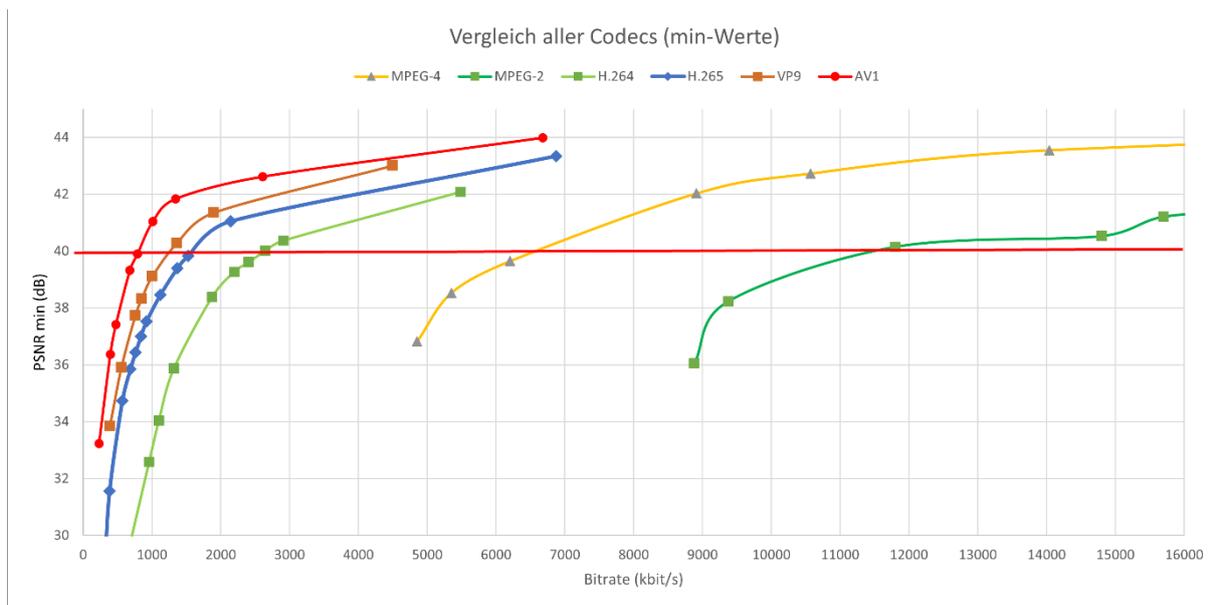


Abbildung 34: Vergleich aller Codecs (min-Werte)

Die rote Linie beschreibt dabei den Punkt der sichtbaren Bildung von Blockartefakten. Hierbei fällt auf, dass neuere Codecs wie AV1, H.265, VP9 schon vor Erreichen der roten Linie einen deutlichen Abfall der Minimalwerte aufweisen. Daraus lässt sich schließen, dass durch den Einsatz von modernen Deblocking-Filtern eine sichtbare Bildung von Blockartefakten erst später erfolgte.

Ebenfalls erkennbar ist eine um etwa Faktor 6 erhöhte Bitrate des MPEG-4-Codecs gegenüber VP9. Bei dem MPEG-2-Codec steigt dieser Wert etwa auf den Faktor 11. Im folgenden werden die einzelnen Codecs nun genauer analysiert.

7.2 Analyse der Stärken und Schwächen der einzelnen Codecs

MPEG-2 ist ein etablierter Codec mit langjähriger Anwendung im Rundfunk und damit der älteste Codec in diesem Experiment. Er findet durch hohe Kompatibilität mit älteren Wiedergabegeräten auch heute noch Einsatz. Allerdings wurde durch einen Vergleich zu neueren Codecs wie H.265 oder AV1 seine begrenzte Kompressionsfähigkeit offenbart. Dadurch ist es nicht möglich, mit den Anforderungen an die effiziente Nutzung von Bandbreite und Speicherplatz in modernen Streaming-Umgebungen Schritt zu halten.

Sein Nachfolger, der MPEG-4-Codec, hebt sich zwar durch erweiterte Funktionen zur Unterstützung interaktiver Multimedia-Anwendungen gegenüber seinem Vorgänger hervor. Jedoch zeigt die Analyse, dass seine Komprimierungseffizienz im Vergleich zu Codecs wie H.265 oder VP9 ebenfalls deutlich beschränkt ist. Dies führt zu einem Kompromiss zwischen Qualität und Dateigröße, der in Zeiten von HD- und UHD-Inhalten immer bedeutsamer wird.

H.264 bietet eine solide Balance zwischen Kompressionsleistung und Qualität. So erzielte der Codec im durchgeführten Experiment akzeptable Ergebnisse unter Berücksichtigung der Bitrate. Allerdings zeigte sich auch bei diesem Codec die geringere Effizienz im Vergleich zu neueren Codecs wie H.265 und AV1. Dennoch bleibt seine weitreichende Unterstützung in der Industrie ein wichtiger Faktor.

Sein Nachfolger H.265 stellt einen signifikanten Fortschritt im Vergleich zu H.264 dar. Die Analyse unterstreicht die bemerkenswerte Kompressionsleistung und die verbesserte Videoqualität von H.265. Nichtsdestotrotz sind Lizenzkosten und eventuelle Inkompatibilitäten in der breiten Akzeptanz zu berücksichtigen.

VP9, ein offener und lizenzkostenfreier Codec, erweist sich als eine vielversprechende Option für Streaming-Dienste. Seine Kompressionsfähigkeit und Videoqualität waren im Allgemeinen zufriedenstellend. Jedoch sind eventuelle Inkompatibilitäten mit einigen Wiedergabegeräten und die begrenzte Verbreitung im Vergleich zu etablierten Codecs wie H.264 und H.265 zu berücksichtigen.

Als letztes bietet der AV1-Codec eine ausgezeichnete Kompressionsfähigkeit und Qualität. Es wird jedoch deutlich, dass die Decodierung von AV1-Videos eine sehr viel höhere Rechenleistung erfordert, was möglicherweise zu Einschränkungen bei älteren Geräten führen kann. Die rasche Weiterentwicklung von AV1 und die Aussicht auf verbesserte Hardware-Unterstützung könnten jedoch diese Hürden zukünftig mindern.

7.3 Erfüllung der Zielsetzung der Arbeit

Die vorliegende Bachelorarbeit widmete sich der Analyse und Bewertung von verschiedenen Videocodecs im Kontext der Evolution von klassischem Broadcast zu modernen Streaming-Diensten. Die Untersuchung zielte darauf ab, Erkenntnisse über die Leistung, die Qualität und die Effizienz der Codecs zu gewinnen.

Die beiden Forschungsfragen, die die Grundlage dieser Untersuchung bildeten, wurden umfassend analysiert, um die vielschichtige Beziehung zwischen Videocodecs und den sich verändernden Anforderungen der heutigen Mediennutzung zu verstehen.

Die erste Forschungsfrage "Wie unterscheiden sich die aktuellen Videocodecs hinsichtlich ihrer Komprimierungseffizienz, Videoqualität und Eignung für verschiedene Streaming-Szenarien?" führte zu einer detaillierten Untersuchung der ausgewählten Codecs. Die Analyse ergab, dass jeder Codec eine spezifische Kombination von Stärken und Schwächen aufweist. Während Codecs wie H.265 und AV1 mit beeindruckender Kompressionseffizienz und hoher Videoqualität auftrumpfen, dafür jedoch hohe Kosten in Form von Rechenleistung verursachen, bleiben ältere Codecs wie MPEG-2 und H.264 in bestimmten Anwendungsbereichen relevant, da sie eine hohe Kompatibilität gewährleisten.

Die zweite Forschungsfrage "Wie haben die Entwicklungen im Bereich der Videocodecs den Wandel vom klassischen Broadcast zum modernen Streaming beeinflusst?" verlangte eine historische Perspektive auf die Evolution der Videocodecs. Die Untersuchung zeigte die Fortschritte in diesem Bereich, der den Wandel von linearem Broadcast zu flexiblen Streaming-Plattformen ermöglicht hat. Die rasche Entwicklung neuer Codecs, wie AV1, und die kontinuierlichen Verbesserungen etablierter Codecs, wie H.265, haben die Streaming-Landschaft maßgeblich geprägt. Dadurch werden nicht nur höhere Videoqualität und effizientere Datenübertragung ermöglicht, sondern auch die Ansprüche an Bandbreite und Speicherplatz erfüllt. Auch das Experteninterview lässt auf eine positive Erwartungshaltung schließen. So berichtete Simon Pontius, dass neue Kompressionsverfahren innerhalb der nächsten Jahre erwartet werden können, die erneut neue Maßstäbe in Effizienz und Technik bieten könnten.

Zusammenfassend zeigt diese Bachelorarbeit, dass die Wahl des richtigen Videocodecs eine entscheidende Rolle für die Qualität und Effizienz moderner Videodistribution spielt. Die stetigen Entwicklungen in diesem Bereich reflektieren den wachsenden Bedarf nach höherer Qualität, flexibleren Verbreitungsmethoden und optimierter Nutzung von Ressourcen. Als Nutznießer dieses technologischen Fortschritts stehen sowohl Inhaltsanbieter als auch Konsumenten im Mittelpunkt eines fortlaufenden Wandels, der die Art und Weise, wie Medieninhalte konsumiert und verteilt werden, grundlegend beeinflusst.

Literaturverzeichnis

- (2023). Encode/AV1 – FFmpeg. Online verfügbar unter <https://trac.ffmpeg.org/wiki/Encode/AV1> (abgerufen am 20.08.2023).
- (2023). Encode/H.264. Online verfügbar unter <https://trac.ffmpeg.org/wiki/Encode/H.264> (abgerufen am 18.08.2023).
- (2023). Encode/VP9 – FFmpeg. Online verfügbar unter <https://trac.ffmpeg.org/wiki/Encode/VP9> (abgerufen am 18.08.2023).
- (2023). ffmpeg Documentation. Online verfügbar unter <https://ffmpeg.org/ffmpeg-all.html> (abgerufen am 28.08.2023).
- Adobe (2023). Intelligentes Rendering in Premiere Pro. Online verfügbar unter <https://helpx.adobe.com/de/premiere-pro/using/smart-rendering.html> (abgerufen am 17.08.2023).
- Apple Support (2023). Informationen zu Apple ProRes. Online verfügbar unter <https://support.apple.com/de-de/HT202410> (abgerufen am 17.08.2023).
- Bhattacharya/Mallick (Hg.) (2010). Stellar Moments in the History of Broadcasting, 2010 Second IEEE Region 8 Conference on the History of Telecommunications (HISTELCON), Madrid, Spain, 03.11.2010 - 05.11.2010. IEEE.
- Blackmagic Design (2023). Blackmagic URSA Broadcast | Blackmagic Design. Online verfügbar unter <https://www.blackmagicdesign.com/de/products/blackmagicursabroadcast> (abgerufen am 17.08.2023).
- Bross, Benjamin/George, Valeri/Alvarez-Mesa, Mauricio/Mayer, Tobias/Chi, Chi Ching/Brandenburg, Jens/Schierl, Thomas/Marpe, Detlev/Juurlink, Ben (2013). HEVC performance and complexity for 4K video. In: 2013 IEEE Third International Conference on Consumer Electronics ζ Berlin (ICCE-Berlin), 2013 IEEE Third International Conference on Consumer Electronics ζ Berlin (ICCE-Berlin), Berlin, Germany, 09.09.2013 - 11.09.2013. IEEE, 44–47.
- Chen Yue *, Debargha Murherjee*, Jingning Han*, Adrian Grange*, Yaowu Xu*, Zoe Liu*, Sarah Parker*, Cheng Chen*, Hui Su*, Urvang Joshi*, Ching-Han Chiang*, Yunqing Wang*, Paul Wilkins*, Jim Bankoski*, Luc Trudeau†, Nathan Egge†, Jean-Marc Valin†, Thomas Davies‡, Steinar Midtskogen‡, Andrey Norkin§ and Peter de Rivaz. (2020). An Overview of Coding Tools in AV1: the First Video Codec from the Alliance for Open Media. APSIPA Transactions on Signal and Information Processing 9 (1). <https://doi.org/10.1017/ATSIP.2020.2>.

- Chen, Yue/Murherjee, Debargha/Han, Jingning/Grange, Adrian/Xu, Yaowu/Liu, Zoe/Parker, Sarah/Chen, Cheng/Su, Hui/Joshi, Urvang/Chiang, Ching-Han/Wang, Yunqing/Wilkins, Paul/Bankoski, Jim/Trudeau, Luc/Egge, Nathan/Valin, Jean-Marc/Davies, Thomas/Midtskogen, Steinar/Norkin, Andrey/Rivaz, Peter de (2018). An Overview of Core Coding Tools in the AV1 Video Codec. In: Yue Chen et al. (Hg.). Picture Coding Symposium (PCS), 2018 Picture Coding Symposium (PCS), San Francisco, CA, 24.06.2018 - 27.06.2018. IEEE, 41–45.
- Dapeng Wu et al. (2001). Streaming video over the Internet: approaches and directions. *IEEE Transactions on Circuits and Systems for Video Technology* 11 (3), 282–300.
<https://doi.org/10.1109/76.911156>.
- Ebrahimi/ Horne (2000). MPEG-4 natural video coding An overview.
- Encode/MPEG-4 (2023). Encode/MPEG-4 – FFmpeg. Online verfügbar unter <https://trac.ffmpeg.org/wiki/Encode/MPEG-4> (abgerufen am 17.08.2023).
- FFMPEG (2023). About FFmpeg. Online verfügbar unter <https://www.ffmpeg.org/about.html> (abgerufen am 17.08.2023).
- ffmpeg Documentation (2023). ffmpeg Documentation. Online verfügbar unter <http://ffmpeg.org/ffmpeg.html> (abgerufen am 17.08.2023).
- Georgios Minopoulos et al. (2020). Comparison of Video Codecs Performance for Real-Time Transmission. 26-29 June 2020, Nagoya, Japan. Piscataway, NJ, IEEE.
- Gregory J. Conklin et al. (2001). Video coding for streaming media delivery on the internet - *Circuits and Systems for Video Technology*, IEEE.
- John G. Apostolopoulos et al. (2002). *Video Streaming: Concepts, Algorithms, and Systems*.
- Martin Rerabek et al. (2015). Quality Evaluation of HEVC and VP9 Video Compression in Real-Time Applications.
- Richardson, Iain E. *The H.264 Advanced Video Compression Standard, Second Edition*.
- Sammoud et al. (2017). Real-Time Streaming Challenges in Internet of Video Things. ISCAS 2017 - IEEE International Symposium of Circuits & Systems : Baltimore, MD, USA, May 28-31, 2017 : 2017 conference proceedings. Piscataway, NJ, IEEE.
- Sara, Umme/Akter, Morium/Uddin, Mohammad Shorif (2019). Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *Journal of Computer and Communications* 07 (03), 8–18. <https://doi.org/10.4236/jcc.2019.73002>.
- SCHMIDT, ULRICH (2021). *Professionelle Videotechnik. Grundlagen, filmtechnik, fernsehtechnik, und... studioteknik in sd, hd, di, 3d.* [S.l.], MORGAN KAUFMANN.
- Shahriar Akramullah. *Digital Video Concepts, Methods, and Metrics*.

Sullivan, Gary J. et al. (2012). Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Transactions on Circuits and Systems for Video Technology 22 (12), 1649–1668. <https://doi.org/10.1109/TCSVT.2012.2221191>.

trac.FFmpeg (2023). FFmpeg. Online verfügbar unter <https://trac.ffmpeg.org/wiki> (abgerufen am 17.08.2023).

Anhang

USB-Stick:

Auf dem beigefügten USB-Stick befinden sich folgende Anhänge:

- Exceldatei mit ausgewerteten Daten und Grafiken
- Ordnerstruktur mit kodierten Videos der Formate MPEG-2, MPEG-4, H.264, H.265, VP9 und AV1 sowie das jeweilige Ausgangsvideo „Ground-Zero“ und „Ground-Zero-animiert“

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit mit dem Titel:

selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Datum

Unterschrift