

Videocodecs

Seminar: Multimediale Datenverarbeitung (SS-23)

Ebubekir Ates (Matr-Nr.: 1616832)

19.06.2023

Inhaltsverzeichnis

1	Einleitung	3
2	Grundlagen der Videocodecs	3
2.1	Videokompression	3
2.2	Frames	4
2.3	Block-Matching Algorithmen	4
2.4	Intra-Frame-Kompression	4
2.5	Inter-Frame-Kompression	4
2.6	Bildqualität und Dateigröße	4
2.7	Bitrate	5
2.8	Anforderungen an Videokompression	5
2.9	Container	5
2.10	Dateiendungen von Videos	5
2.11	Arbeitsweise von Videocodecs	5
3	Grundlagen des H.264 Videocodecs	5
3.1	Frameerkennung und Makroblöcke	6
3.2	Intracodierung	6
3.3	Interkodierung mit Bewegungskompensation	8
3.4	Direkte Kosinustransformation	9
3.4.1	Komprimierung der Informationen	9
3.4.2	Entfernung von Redundanzen	9
3.4.3	Betonung wichtiger visueller Informationen	9
3.4.4	Trennung von Signal und Störungen	10
3.4.5	DCT-Formel	10
3.5	Quantisierung	10
3.6	Entropiekodierung	11
4	Weitere Videocodecs	11
4.1	HEVC(High Efficiency Video Coding) H.265	11

4.2	VP9 (Google VP9)	11
5	Exkurs: Zeilensprungverfahren	12
6	Schluss	12

1 Einleitung

Die rasante Entwicklung digitaler Technologien hat dazu geführt, dass die Übertragung und die Wiedergabe von Videos exponentiell gestiegen sind. Videodateien sind weit verbreitet und erfordern eine hohe Übertragungsgeschwindigkeit, Speicherkapazität und Bildqualität, wenn man Filme und Serien streamt, auf Smartphones aufzeichnet oder Videokonferenzen führt. Videocodecs sind also essentiell, diese Dateien zu bearbeiten und zu komprimieren. Das Ziel dieser Arbeit besteht darin, einen umfassenden Überblick über die Welt der Videocodecs zu vermitteln. Behandelt werden die Grundlagen von Videocodecs, verschiedene Arten von Videocodecs werden untersucht und ihre Einsatzgebiete diskutiert. Außerdem werden Technologien und Standards wie den H.264- und H.265-Codec genauer untersucht um ihre Komprimierungsmethoden besser verstehen zu können.

Die Zukunft verspricht die neuesten Entwicklungen und Trends im Bereich der Videocodecs. Welche Verbesserungen bietet der H.265 Videocodec zum H.264? Ein kurzer Exkurs in die Vergangenheit zeigt uns einen veralteten Videocodec, das Zeilensprungverfahren.

Diese Seminararbeit bietet einen umfassenden Einblick in die Funktionsweise von Videocodecs und ihre Bedeutung für die Komprimierung, Übertragung und Wiedergabe von Videos. Sie zeigt die Standards und Technologien, die hinter Videodateien stecken, und gibt uns einen Einblick in die Zukunft der Videocodecs.

2 Grundlagen der Videocodecs

Ein Video ist eine Folge von Bildern, die als Frames bezeichnet werden, wobei jedes Frame Informationen über die Farben und Positionen eines Pixels enthält. Videocodecs sind Verfahren und Algorithmen zur Codierung und Decodierung von Videos.[5] Bei der Videokompression werden Frames analysiert und entsprechend den Kompressionsalgorithmen codiert, um die Datenmenge zu reduzieren. Es gibt verschiedene Arten von Frames, darunter Keyframes (I-Frames), Predictive Frames (P-Frames) und Bi-directional Predictive Frames (B-Bilder).

2.1 Videokompression

Um die Dateigröße von Videos zu verringern und gleichzeitig eine akzeptable visuelle Qualität zu gewährleisten, ist die Videokompression ein wichtiger Prozess. Die Komprimierung basiert auf der Speicherung der für die Wiedergabe relevanten Informationen und der Reduzierung unnötiger Informationen. Durch die Verringerung der Datenmenge wird nicht nur Speicherplatz gespart, sondern es wird auch einfacher, Videos über beispielsweise Netzwerke zu übertragen. Bei der Arbeit mit Videocodecs gibt es einige wichtige Konzepte zu beachten. folglich sind einige der Schlüsselkonzepte.

2.2 Frames

Ein Video besteht aus einer Folge von Einzelbildern, die als Frames bezeichnet werden. Jeder Frame enthält Informationen über Farben und Pixelpositionen und wird nacheinander abgespielt, um eine Bewegung zu erzeugen.

2.3 Block-Matching Algorithmen

[10] Ein Block in einem Frame fehlt, der beste Weg ihn wiederzufinden ist durch sogenannte Block-Matching Algorithmen. Eine Suche im aktuellen Frame wird ausgeführt, das Ziel ist es einen übereinstimmenden Block zu finden, dessen Begrenzung am nächsten zum fehlenden Block liegt. Mit den Informationen des übereinstimmenden Blocks kann vorhergesagt werden, wie der fehlende Block aussehen könnte und den Wiederherstellungsprozess abgeschlossen werden kann. Wenn der Zielblock mit einem übereinstimmenden Block wiederhergestellt werden kann, wird eine bessere Vorhersage erhalten und damit ein besseres Ergebnis der Intra-Frame-Kompression.

2.4 Intra-Frame-Kompression

Die Daten werden bei der Intra-Frame-Kompression innerhalb eines einzelnen Frames komprimiert. Dies ermöglicht die Verringerung der Menge an Daten, die in einem einzigen Bild enthalten sind. Um räumliche Redundanzen innerhalb eines Frames zu entfernen, können verschiedene Kompressionsverfahren verwendet werden. Dazu können Transformationsmethoden wie die diskrete Kosinustransformation (DCT) verwendet werden, die das Bild von der räumlichen Domäne in die Frequenzdomäne umwandelt. Die Frequenzkoeffizienten werden dann quantisiert und codiert, um die Datenmenge zu reduzieren.

2.5 Inter-Frame-Kompression

Inter-Frame-Kompression komprimiert Daten mithilfe verschiedener Frames. Da viele Frames in einem Video ähnlich sind, können nur die Unterschiede zwischen den Frames gespeichert werden, was die Datenmenge reduziert. Da sich Objekte in einem Video häufig nur geringfügig bewegen, kommt die zeitliche Redundanz zum Tragen. Die Inter-Frame-Kompression basiert auf der Verwendung der vorherigen Frames, um die Positionen und Farben der Pixel in einem Frame vorherzusagen. Blockmatching Algorithmen wie Diamond Search ermöglichen die Vorhersage. Dann wird die Differenz zwischen dem tatsächlichen Frame und der Vorhersage komprimiert und gespeichert.

2.6 Bildqualität und Dateigröße

Die Videokompression reduziert sowohl die Dateigröße als auch die Bildqualität. Eine höhere Kompressionsrate führt zu einer geringeren Bildqualität und einer kleineren Dateigröße. Das richtige Gleichgewicht zwischen Dateigröße und Bildqualität hängt von den Anforderungen des Anwendungsfalls ab.

2.7 Bitrate

Die Bitrate bezeichnet die Menge an Daten, die pro Zeiteinheit übertragen werden. Eine höhere Bitrate führt zu einer höheren Qualität und einer größeren Dateigröße, während eine niedrigere Bitrate zu einer geringeren Qualität und einer kleineren Dateigröße führt.

2.8 Anforderungen an Videokompression

Die Anforderungen an die Videokompression variieren je nach Anwendung. Die Reduzierung der Datenmenge ist eine der Hauptanforderungen, um Speicherplatz zu sparen und die Übertragungseffizienz zu verbessern. Die Aufrechterhaltung einer akzeptablen visuellen Qualität, um sicherzustellen, dass das komprimierte Video für die Betrachter angenehm anzusehen ist, ist eine weitere wichtige Anforderung. Um eine Vielzahl von Anwendungen zu unterstützen, sollten Kompressionsalgorithmen in der Lage sein, mit verschiedenen Videoauflösungen umzugehen.

2.9 Container

Container fassen mehrere Medienströme, das Video, die Audio und Untertitel in eine Datei zusammen. Sie dienen als Behälter für die verschiedenen Elemente.

2.10 Dateieindungen von Videos

Bestimmte Dateieindungen, die auf das verwendete Containerformat hinweisen, werden häufig verwendet, um Videodateien zu identifizieren. Ein Containerformat ist eine Art Behälter, der Video-, Audio- und Metadaten in einer einzigen Datei zusammenfasst. Beispiele für bekannte Dateiformate sind MP4, AVI, MKV und MOV. Es gibt verschiedene Videocodecs, die unterstützt werden, und jedes Containerformat hat seine eigenen Merkmale. Die Struktur der Daten im Video wird durch diese Dateieindungen beschrieben, und es wird angezeigt, welche Codecs zur Wiedergabe verwendet werden können.

2.11 Arbeitsweise von Videocodecs

Videocodecs funktionieren in mehreren Schritten. Zunächst teilt man das Video in verschiedene Frames auf, anschließend werden verschiedene Komprimierungsmethoden verwendet, um die Datenmenge zu verringern. Dies kann die Verwendung von Entropiekodierung, Bewegungskompensation, die Nutzung der Ähnlichkeiten zwischen Frames und die Entfernung überflüssiger Informationen umfassen. Die komprimierten Daten werden schließlich in einem geeigneten Dateiformat gespeichert.

3 Grundlagen des H.264 Videocodecs

H.264 wird hauptsächlich verwendet, um Videosignale mit hoher Qualität bei einer geringen Datenrate zu übertragen oder zu speichern. H.264-Videos benötigen im Vergleich zu unkomprimierten Videos deutlich weniger Speicherplatz und können über Netzwerke

wie das Internet effizient übertragen werden. Die im H.264-Codec verwendeten verschiedenen Kompressionsverfahren ermöglichen diese Leistung.

Die Blockbasierte Bewegungskompensation teilt das Video in kleine Teile auf und versucht, die Bewegungsinformationen zwischen den aufeinanderfolgenden Frames für jeden Block zu schätzen. Die Vorhersagefehler werden nur kodiert, anstatt jedes Frame vollständig zu kodieren. Sie repräsentieren die Unterschiede zwischen dem aktuellen Block und seinem vorherigen Gegenstück aus einem anderen Frame. Dies ermöglicht eine erhebliche Verringerung der Datenmenge.

Darüber hinaus verwendet H.264 eine Vielzahl von Techniken, um die Kompressionseffizienz zu verbessern. Die variable Blockgröße ist eine dieser Methoden. H.264 ermöglicht die Verwendung unterschiedlicher Blockgrößen, um feine Details und große Bewegungen besser zu kodieren. Dies führt zu einer weiteren Verringerung der Anzahl der Daten.

[6] Die Verwendung von Transformationscodierung ist ein weiterer wesentlicher Bestandteil von H.264. Um Überfüllungen zu reduzieren, wird die Pixelblockdarstellung in den Frequenzbereich übertragen. Die Blockdaten werden durch den diskreten Kosinustransformationsalgorithmus (DCT) transformiert. Anschließend wird die Quantifizierung der transformierten Koeffizienten durchgeführt, um eine noch größere Datenreduktion zu erreichen.

3.1 Frameerkennung und Makroblöcke

Ein Video wird nun als Input geliefert, dieses wird in die jeweilige Frameart unterteilt, das heißt entweder I-Frame, P-Frame oder B-Frame. Komplexe Algorithmen und Parameter im Kodierungsprozess bestimmen, welcher Frame-Typ verwendet wird. Der Encoder analysiert die Bildsequenz, um Bewegungsmuster zu identifizieren, und wählt dann den geeignetsten Frame-Typ aus, um eine optimale Kompression und Bildqualität zu erreichen. Die Wahl des Frame-Typs kann je nach Anforderungen an Datenrate, Bildqualität, Bandbreite oder Speicherplatz von Anwendung zu Anwendung variieren. [1] Diese Frames werden dann nochmals intern in sogenannte Makroblöcke unterteilt, diese können von 16 x 16 Pixel bis zu 4 x 4 Pixel Größen unterteilt werden.

3.2 Intracodierung

I-Frames werden unabhängig von anderen Frames codiert, bedeutet also, dass sie nicht in die Kategorie der Bewegungskompensation fallen. Im Zuge dessen werden die 4x4 und 16x16 Makroblöcke behandelt.

Es gibt verschiedene Vorhersagemodi für jeden 4x4-Block, die als Intra-Modi bekannt sind, die verwendet werden können, um die Pixeldaten zu kodieren.

Die 9 Modi setzen sich wie folgt zusammen [10]:

1. DC-Modus (DC): In diesem Modus wird der DC-Wert (Gleichstromkomponente) der oberen und linken Pixel verwendet, um den aktuellen 4x4-Block vorherzusagen. Der DC-Modus funktioniert gut, wenn die Pixeldaten wenig variieren.

2. Vertikaler Modus (Vertical): In diesem Modus wird die Vorhersage des aktuellen Blocks durch den Wert des linken Pixels in der aktuellen 4x4-Makroblockgruppe erstellt. Wenn das Bild eine starke vertikale Struktur hat, ist dieser Modus ideal.
3. Horizontaler Modus (Horizontal): In diesem Modus wird die Vorhersage für den aktuellen Block durch den Wert des oberen Pixels in der aktuellen 4x4-Blockgruppe erstellt. Wenn das Bild eine starke horizontale Struktur hat, ist dieser Modus ideal.
4. Diagonaler Modus (Down Left / DL): In diesem Modus wird die Vorhersage durch die Werte der diagonalen Nachbarn des aktuellen Blocks erstellt. Wenn es eine schräge Struktur im Bild gibt, ist dieser Modus geeignet.
5. Diagonaler Modus (Down Right / DR): Dieser Modus verwendet die Werte der diagonalen Nachbarn auf der anderen Seite des Blocks, er ähnelt dem DL-Modus.
6. Vertikaler rechtsgerichteter Modus (Vertical Right / VR): In diesem Modus wird der Wert des linken oberen Pixels in der 4x4-Blockgruppe verwendet, um die Vorhersage für den aktuellen Block zu machen. Wenn es eine vertikale Struktur mit einem leichten Knick nach rechts gibt, ist dieser Modus besonders geeignet.
7. Horizontaler abwärtsgerichteter Modus (Horizontal Down / HD): In diesem Modus wird der Wert des linken oberen Pixels in der 4x4-Blockgruppe verwendet, um die Vorhersage für den aktuellen Block zu machen. Wenn es eine horizontale Struktur mit einem leichten Knick nach unten gibt, ist dieser Modus besonders geeignet.
8. Vertikaler linksgerichteter Modus (Vertical Left / VL): In diesem Modus wird der Wert des rechten oberen Pixels in der 4x4-Blockgruppe verwendet, um den aktuellen Block vorherzusagen. Wenn es eine vertikale Struktur mit einem leichten Knick nach links gibt, ist dieser Modus besonders geeignet.
9. Horizontaler aufwärtsgerichteter Modus (Horizontal Up / HU): In diesem Modus wird die Vorhersage des aktuellen Blocks durch den Wert des linken unteren Pixels in der 4x4-Blockgruppe erstellt. Wenn es eine horizontale Struktur mit einem leichten Knick nach oben gibt, ist dieser Modus besonders geeignet.

Man kann verschieden große Makroblöcke bearbeiten. Zum einen wie schon gezeigt 4 x 4 Blöcke zum anderen aber auch ganze 16 x 16 Blöcke. In Abbildung 1 ein Beispiel, wie ein ganzer 16 x 16 Block visuell codiert werden würde. Der Vertikal Modus wird im Zuge dessen genauer betrachtet. Man erkennt, dass für die Kodierung alle Pixelinformationen für eine Spalte von dem Pixelwert der ersten Zeile genommen werden. Die Pixelwerte für die Spalte 2 beispielsweise wird für alle Pixel durch die erste Zeile bestimmt. Dadurch entsteht diese gleichmäßige vertikale Struktur, die wir auf Abbildung 1 sehen.

Wichtig hierbei ist auch noch zu erwähnen ist, dass die Entscheidung welche Modi genutzt wird von vielen Faktoren wie der genauen Implementierung des Algorithmus, sowie Aspekten wie des Sum of Absolute Differences(SAD) abhängen. Der SAD berechnet die absoluten Unterschiede zwischen den Pixelwerten des aktuellen Blocks und den

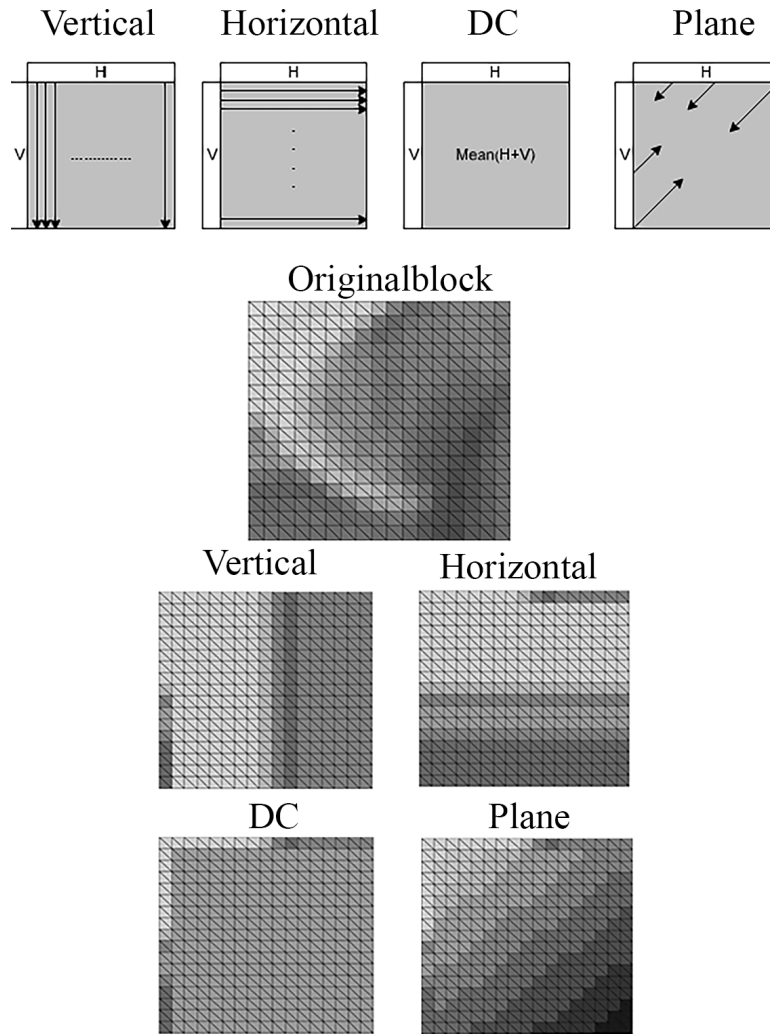


Abbildung 1: Visuelle Darstellung einer 16x16 Intracodierung Bilder von [6] und[7]

Pixelwerten des Referenzblocks. Dieser Wert definiert wie unterschiedlich die Pixelwerte der Blöcke sind.

3.3 Interkodierung mit Bewegungskompensation

Interkodierung nutzt Referenzframes um mit Block Matching Algorithmen, wie zum Beispiel dem Diamond Search vorzugehen. Falls das Ganze erfolgreich ist kann der Block mit einem Bewegungsvektor, also einem Vektor der auf die Position des Referenzframes, der durch einen Block Matching Algorithmus gefunden wurde, zeigen. Meistens findet der Algorithmus ein Block der als Referenzblock genutzt werden kann, jedoch wird dabei auch noch ein Fehler berechnet, nämlich der SAD. Zusammenfassend lässt sich sagen, dass der Encoder, wenn er einen übereinstimmenden Block in einem Referenzframe fin-

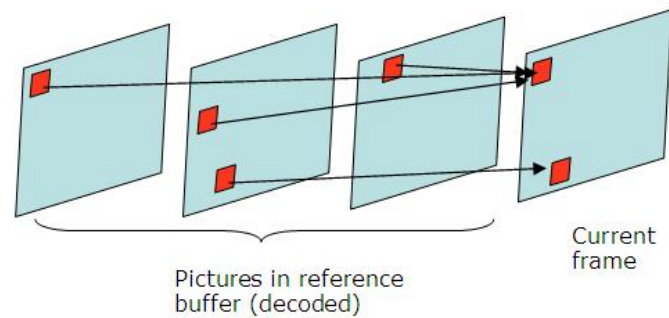


Abbildung 2: Bild von [9]

det, einen Bewegungsvektor erhält, der auf den übereinstimmenden Block zeigt. Dieser enthält außerdem den Vorhersagefehler. Wichtig hierbei ist, falls das gefundene Referenzframe auch ein Referenzframe ist, wird der SAD akkumuliert. Aufgrund dessen müssen immer wieder I-Frames eingesetzt werden, damit man den SAD minimal halten kann. Falls das Referenzframe gefunden wird, wird wie oben beschrieben vorgegangen, falls dies nicht der Fall ist, wird der Block roh kodiert, also statt ein Referenzframe zu nutzen wird einfach wie bei der Intra-Kodierung vorgegangen, dies erhöht die letztendliche Dateigröße. Abbildung 2 zeigt grob wie die Interkodierung funktioniert.

3.4 Direkte Kosinustransformation

Die Blöcke wurden in ihre Frame Art und weiteren Faktoren codiert. Genommen wird im Folgenden ein 4×4 Block in Matrixschreibweise, dieser ist noch mit den Werten der Pixel befüllt. Diese Pixelwerte wollen wir nun in den Frequenzbereich übersetzen, die hat folgende Gründe.

3.4.1 Komprimierung der Informationen

Die Umwandlung in den Frequenzbereich ermöglicht die Daten des Videos auf eine kompaktere Darstellung zu reduzieren.

3.4.2 Entfernung von Redundanzen

In Bereichen mit geringen Änderungen oder Wiederholungen können viele Informationen redundant sein. Redundante Informationen können durch die Transformation in den Frequenzbereich entfernt und effektiv komprimiert werden.

3.4.3 Betonung wichtiger visueller Informationen

Die DCT-Transformation ermöglicht die Priorisierung der visuellen Informationen eines Videos basierend auf ihrer Bedeutung. Höhere Koeffizienten können wichtige visuelle

Merkmale oder Strukturen darstellen, während niedrigere Koeffizienten weniger wichtige Informationen darstellen, da die DCT-Koeffizienten die Beiträge der verschiedenen Frequenzkomponenten des Bildes darstellen.

3.4.4 Trennung von Signal und Störungen

Die DCT-Transformation ermöglicht es, Signalanteile von Störungen wie Rauschen zu unterscheiden. Durch die Transformation in den Frequenzbereich können störende Einflüsse besser identifiziert und gegebenenfalls reduziert oder eliminiert werden.

3.4.5 DCT-Formel

[3] Folglich ein Beispiel wie eine ursprüngliche Pixel-Matrix nach der Transformation mit DCT aussehen könnte:

Vor der Transformation: (1)

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Nach der Transformation: (2)

$$\begin{bmatrix} -8 & 4.51 & 0 & -0.29 \\ 0.36 & -0.21 & 0 & 0.06 \\ 0 & -0.07 & 0 & -0.01 \\ -0.04 & 0.03 & 0 & 0 \end{bmatrix}$$

[8] Die erste Zahl in der neuen Matrix (-8) repräsentiert die Gleichanteilkomponente. Dieser Wert zeigt die durchschnittliche Helligkeit oder Helligkeitsverschiebung des Blocks an. Falls der Wert negativ ist deutet das auf eine Helligkeitsverschiebung nach unten an und falls der Wert positiv ist deutet das auf eine Helligkeitsverschiebung nach oben an. Die restlichen Werte sind die Wechselanteilkomponente, größere Werte weisen auf stärkere Schwankungen und kleine auf entsprechend schwächere Schwankungen. Diese Werte werden, entsprechend abgerundet, um weitere Komprimierung zu ermöglichen.

3.5 Quantisierung

Abhängig von den Einstellungen der Bitrate wird für jeden Makroblock ein Quantisierungsparameter QP festgelegt. Dieser geht von 0 bis 51. Die Quantisierung ist dazu da um nochmals jeden Wert weiter abzurunden, um weitere Komprimierung zu ermöglichen. Es gibt viele verschiedene Abwandlungen dazu, daher wird in dieser Seminararbeit keine spezifische Formel dafür angegeben.

3.6 Entropiekodierung

Nach der Quantisierung folgt noch abschließend die Entropiekodierung. Diese wandelt einen aus einzelnen Zeichen bestehenden Text in eine Bitfolge um. Die Huffman-Kodierung, sowie die arithmetische Kodierung sind hierbei typische Vertreter.

4 Weitere Videocodecs

Der H.264 ist einer der großen Standards in der aktuellen Videokodierung, jedoch gibt es auch noch einige weitere, die verschiedene Vorteile bieten und sich in ihrer Funktionsweise unterscheiden.

4.1 HEVC(High Efficiency Video Coding) H.265

H.265 nutzt einige Techniken, um eine Verbesserung zum H.264 zu bewerkstelligen. Dazu gehören zum Beispiel:

- Verwendung von fortschrittlicheren Kodierungstechniken wie größeren Blockgrößen
- erweiterter Bewegungsschätzung
- bessere Vorhersagealgorithmen

Diese führen zu einer besseren Kompressionsrate. Größere Blockgrößen bieten den Vorteil größere Bereiche zu analysieren. Dadurch können zusammenhängende Bereiche besser erfasst und besser kodiert werden. Abgesehen davon bietet H.265 aber auch eine höhere Flexibilität für Videoauflösungen, Bildraten und Farbräume. Nicht zu vergessen ist auch, die Unterstützung für erweiterte Bildformate wie High Dynamic Range(HDR). HDR verbessert die Darstellung von Kontrast und Farben in Videos. Dies führt dazu, dass die Frames lebendiger und realistischer dargestellt werden können.

4.2 VP9 (Google VP9)

[2] VP9 ist eine Videokodierung, die von Google entwickelt wurde und als Open-Source-Projekt verfügbar ist. Es ist der Nachfolger von VP8 und wurde entwickelt, um eine bessere Komprimierungseffizienz als sein Vorgänger zu bieten.

Der speziell für die Komprimierung von Webvideos entwickelte Codec findet häufig Anwendung in Bereichen wie Videostreaming und Online-Videos. Einer seiner stärksten Fähigkeiten ist es, hochwertige Videostreams bei niedrigeren Bitraten zu liefern.

YouTube, Netflix und verschiedene Webbrowser unterstützen VP9..

Wichtig zu erwähnen ist noch, dass VP9 eine höhere Rechenleistung erfordert, was im Vergleich zum H.264 ein Nachteil erschafft.

Unterschiede zum H.264 Videocodec liegen in unter anderem in Blockgrößen, die beim VP9 bis zu 64 x 64 Pixel groß sein können. Aber auch die Intraframe-Kodierung unterscheidet sich. VP9 nutzt vertikale und horizontale Modelle, während H.264 mehrere Modi, wie beispielsweise DC-Vorhersage und Planar-Vorhersage.

5 Exkurs: Zeilensprungverfahren

Abschließend wird noch das Zeilensprungverfahren behandelt, da dieses primär entwickelt wurde, um Anforderungen von Röhrenfernsehern gerecht zu werden. Beim Zeilensprungverfahren, auch Interlacing genannt, wird das Bild in zwei Halbbilder aufgeteilt. Dabei wird das Bild in die geraden und ungeraden Zeilen aufgespalten.[4] Haben wir also 1080 Zeilen gehören die Zeilen 1,3,5,7.. zum Ersten Bild und 2,4,6,8,.. zum Zweiten Bild. Dadurch konnte die Bildwiederholfrequenz erhöht werden, ohne dass eine höhere Bandbreite erforderlich war. Bei der Wiedergabe wird zuerst das erste Halbbild gezeigt und dann das zweite. Der schnelle Wechsel zwischen den Bildern erzeugt den Eindruck eines fließenden Videos. Wie schon angesprochen hatte das zur Zeit von Röhrenfernsehern den Vorteil, dass weniger Bandbreite erforderlich war, um das Video darzustellen. Früher war dies ein limitierender Faktor, jedoch nicht mehr heute, wodurch das Interlacing an Bedeutung verloren hat. Der klassische Treppeneffekt oder das Flimmern, den man von alten Fernsehern gewohnt ist kommt genau von dem Zeilensprungverfahren.

6 Schluss

In dieser Seminararbeit wurden die Grundlagen der Videocodecs behandelt, Begriffsbestimmungen getätigt und der H.264 Videocodec wurde genauer erklärt. Beim H.264 Videocodec wurden die Frameerkennung und die Makroblöcke analysiert, die Intracodierung im tiefsten Detail erklärt, sowie die Interkodierung mit Bewegungskompensation angeschnitten. Die direkte Kosinustransformation (DCT) wurde als eine Technik zur Umwandlung von Bildinformationen in den Frequenzbereich erläutert, abschließend wurde die Quantisierung und Entropiekodierung dargestellt.

Des Weiteren wurde ein kurzer Einblick auf weitere Videocodecs wie den verbesserten H.265 und den von Google entwickelten Vp9 Videocodec gegeben. Die Studie des Zeilensprungverfahrens rundet die Seminararbeit ab, indem es einen Exkurs zu einer älteren Technologie bietet, die für die Darstellung von Videos auf Röhrenfernsehern verwendet wurde. Zusammenfassend besitzen Videocodecs einen essentiellen Stellenwert bei der effizienten Kompression und Übertragung von Videoinhalten. Der H.264 Videocodec hat sich als einer der am wichtigsten Videocodecs etabliert, während der HEVC (H.265) Codec und der VP9 Codec als vielversprechende Weiterentwicklungen gelten.

Literatur

- [1] A. Ahmad, N. Khan, S. Masud, and M.A. Maud. Selection of variable block sizes in h.264. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–173, 2004.
- [2] Google Developers. VP9 Video Codec. <https://developers.google.com/media/vp9?hl=de>. .
- [3] MathWorks. Discrete Cosine Transform. <https://www.mathworks.com/help/images/discrete-cosine-transform.html>. .

- [4] Mediencommunity. Zeilensprungverfahren (Interlacing/Interlaced Scanning). <https://mediencommunity.de/content/zeilensprungverfahren-interlacinginterlaced-scanning/>. .
- [5] OTTVerse. What is Video Transcoding? <https://ottverse.com/what-is-video-transcoding/>. .
- [6] Iain E. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. Wiley, 2003.
- [7] VCODEx. H.264/AVC Inter Prediction. <https://www.vcodex.com/h264avc-inter-prediction/>. .
- [8] Wikipedia-Autoren. Diskrete Kosinustransformation — Wikipedia, Die freie Enzyklopädie. https://de.wikipedia.org/wiki/Diskrete_Kosinustransformation. .
- [9] Wikipedia contributors. Inter frame — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Inter_frame. .
- [10] Jiheng Yang, Baocai Yin, Yanfeng Sun, and Nan Zhang. A block-matching based intra frame prediction for h.264/avc. In *2006 IEEE International Conference on Multimedia and Expo*, pages 705–708, 2006.