h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Hochschule Darmstadt
- Fachbereich Informatik -

Grundlagen der Videokompression

Seminararbeit im Kurs
Wissenschaftliches Arbeiten in der Inforamtik I

vorgelegt von
Justin Böhm und Matthias Greune

Referentin: <Name>

Ausgabedatum: <Datum>
Abgabedatum: <Datum>

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

<Name>
<Ort>, den 3. Dezember 2016

Abstrakt

Videos sind seit der Entwicklung des Fernsehers zum Massenmedium kaum noch aus dem alltäglichen Leben wegzudenken. Seit dem Aufstieg des Internets als zentrales Kommunikationsmedium haben sich allerdings die Anforderungen an geeignete Speichertechniken von Videos drastisch verändert. Die heutigen Abspielgeräte haben noch immer begrenzten Speicherplatz und sind häufig nur mit schmalbandigen Internetanbindungen ausgestattet. Die Auflösung der Videos ist hingegen stark gestiegen. Um diese Ansprüche zu adressieren wurden Kompressionsalgorithmen entwickelt, die eine effiziente Speicherung speziell für bewegte Bilder ermöglichen. Die resultierenden Probleme aus dieser Art der Speicherung, wie Bildartefakte, sind heutigen Nutzern wohlbekannt. Die eigentliche Funktionsweise von Videokompression bleibt aber oft unbemerkt.

Deshalb möchten wir in dieser wissenschaftlichen Arbeit eine Übersicht über die Grundlagen von Videokompressionsverfahren geben.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

<Text>

2 Irrelevanzreduktion

Die rohe Aufnahme eines Bildes bietet eine Fülle an Informationen. Mit Blick auf die Eigenschaften des menschlichen Sehsinns lässt sich hierbei allerdings feststellen, dass einige Informationen relevanter für das Erkennen eines Bildes sind, als andere. Die Irrelevanzreduktion beschäftigt sich mit der Trennung und Reduzierung von weniger wichtigen Informationen und bietet damit Methoden zur verlustbehafteten Datenkompression an.

Bei der Videokompression werden im wesentlichen zwei Eigenschaften zur Reduktion von Daten ausgenutzt. Zum einen nimmt das Auge Varianzen in der Helligkeit (Luminanz) stärker wahr, als Änderungen im Farbton (Chrominanz). Zum Anderen ist das Auge besser in der Lage niedrige Ortsfrequenzen zu erkennen, als hohe - erkennt also grobe Strukturen eher als feinere. Diese Eigenschaften können nun ausgenutzt werden, um einen guten Kompromiss aus akzeptabler Bildqualität und guter Datenreduktion zu finden [Akr14].

2.1 Chroma Subsampling

Das Chroma Subsampling nutzt den Umstand aus, dass Helligkeitsvarianzen besser wahrgenommen werden, als Farbvarianzen. Zumeist liegen die Bildinformationen im Ausgangsformat jedoch im RGB Farbmodell vor, wobei hier die Helligkeitswerte in jeden Kanal eingehen. Um nun aber die Chrominanz bei gleichbleibender Auflösung der Luminanz zu reduzieren wird eine getrennte Darstellung dieser Informationen benötigt. Hierfür wird im MPEG-1 Standard die $\mathrm{YC}_B\mathrm{C}_R$ Darstellung verwendet, wobei das Y für die Luminanz steht und in C_B und C_R die Farbwerte codiert werden. Die Umrechnung lässt sich mittels folgender Formeln realisieren [ITU95]:

$$
\begin{aligned}
Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\
U &= (B - Y) \cdot 0.493 \\
V &= (R - Y) \cdot 0.877
\end{aligned}
$$

Nun kann das eigentliche Subsampling stattfinden, welches bei MPEG-1 bei einer Auflösung von 4:2:0 realisiert wird. Die erste Zahl gibt hierbei die horizontale Abtastrate der Luminanz an. Die zweite Zahl steht für die horizontale Abtastrate der C_B und C_R Kanäle in Relation zum ersten Wert. Die dritte Zahl gibt die vertikale Samplingrate an, wobei diese entweder 2 oder 0 betragen kann, also entweder kein vertikales Subsampling, oder vertikales Subsampling von 2:1 stattfindet. Für den Fall von 4:2:0 Subsampling bedeutet dies, dass jeweils 2x2 Bildpunkte des C_B und C_R Kanals auf einen Bildpunkt in der Ergebnismenge abgebildet werden. Hiermit wird also die Auflösung des C_B und C_R Kanals halbiert, was zu einer Datenreduktion von 50% führt. [Poy]

Das Chroma Subsampling bietet somit eine gute Möglichkeit der Kompression, die allerdings nicht verlustfrei abläuft. Artefakte können, wie im Anhang 5 dargestellt, bei Verwendung dieser Methode vor allem bei scharfen, farbigen Kanten entstehen, wenn diese durch einen gesubsampleten Block verlaufen.

2.2 Diskrete Kosinus Transformation

Wie bereits oben beschrieben neigt der menschliche Sehsinn dazu niedrige Ortsfrequenzen eher zu erkennen, als höhere. Eine Ortsfrequenz ist definiert als „Anzahl bestimmter periodischer Erscheinungen bezogen auf einen räumlichen Abstand" [Atm]. Wir erkennen also gröbere Strukturen mit einer niedrigen Ortsfrequenz eher als feinere Strukturen mit einer höheren. Um diesen Umstand nun auszunutzen muss das Ausgangsbild von der räumlichen Ebene auf eine Frequenzebene transformiert werden, damit anschießend, in dem darauf folgenden Schritt der Quantisierung, die höheren Frequenzen reduziert werden können. Diese Transformation lässt sich mittels einer zweidimensionalen Diskreten Kosinus Transformation (DCT) bewerkstelligen.

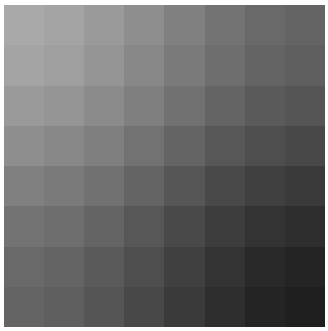
Die DCT ist eine Sonderform der Foriertransformation, bei der eine Funktion mittels Sinusschwingungen approximiert wird. Die Fourriertransformation hat allerdings unter anderem den Nachteil, dass für jeden abgetasteten Punkt ein Tupel aus Amplitude und Phase bzw. Sinus und Kosinus Koeffizienten gespeichert werden muss. Die DCT nutzt nun den Umstand aus, dass das betrachtete Intervall begrenzt ist. Durch eine vertikale Spiegelung dieses Intervalls lassen sich die Sinus Anteile herauskürzen, wobei am Ende lediglich Kosinus Anteile übrig bleiben, also nur ein Koeffizient pro abgetasteten Punkt gespeichert werden muss. Des Weiteren bewirkt die Spiegelung, dass Start- und Endpunkt

equivalent sind. Da die Fouriertransformation von einer unendlichen Folge ausgeht, muss der letzte Koeffizient den ggf. großen Unterschied zwischen Start- und Endpunkt ausgleichen. Sind diese Punkte aber equivalent, wird die Kraft des letzten Koeffizienten nicht verschwendet [Sym04]. Verarbeitet werden mit der zweidimensionalen DCT immer 8x8 Blöcke eines jeden Kanals mit der Formel:

$$F(u,v) = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

$$\text{wobei} \begin{cases} C_u = \frac{1}{\sqrt{2}} \text{ für } u = 0, \text{ ansonsten } C_u = 1 \\ C_v = \frac{1}{\sqrt{2}} \text{ für } v = 0, \text{ ansonsten } C_v = 1 \end{cases}$$

Die Abbildung 2.1 zeigt das Resultat einer angewandten DCT auf einen schwarz-weißen 8x8 Pixelblock (siehe Anhang **??**), welcher aus jeweils einer horizontalen und einer vertikalen Kosinus Schwingung besteht. Der so genannte DC Wert ist der erste Wert der Matrix und gibt die mittlere Helligkeit an. Alle anderen Komponenten beschreiben die relative Abweichung zu diesem Wert und werden gemeinhin als AC Werte betitelt, wobei diese zugleich die zum unteren rechten Rand hin höher werdenden Ortsfrequenzen repräsentieren. Wie bereits zu erkennen führt die DCT oftmals selbst schon durch Rundung auf ganzzahlige Ergebnisse zu einer Matrix mit vielen gleichen Werten, die sich für die Anwendung weiterer, verlustfreier, Kompressionsmethoden eignet.



800	200	0	0	0	0	0	0
200	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Abbildung 2.1: Mittels DCT gut komprimierbarer 8x8 Pixelblock
Links: Ausgangsbild, Rechts: Resultierende DCT-Matrix

2.3 Quantisierung

Im vorigen Schritt wurde durch Anwendung der Diskreten Kosinus Transformation eine Matrix mit den korrespondieren Ortsfrequenzen eines 8x8 Pixelblocks gewonnen. Um nun tatsächlich eine Reduktion der höheren Ortsfrequenzen zu erreichen wird die Methode der Quantisierung angewandt. Hierbei wird eine ganzzahlige Division eines jeden DCT Koeffizienten mit einem Quantisierungswert vorgenommen. Das abgerundete Ergebnis ist dann der quantisierte Wert. Durch diese Division und Rundung wird versucht die bisher noch hohen Koeffizienten zu verkleinern, sowie in den höheren Frequenzbereichen möglichst auf Ergebnisse gleich Null zu kommen.

Im Fall von MPEG-1 wird hierfür ein Uniform Scalar Quantizer verwedet, bei dem die Eingangswerte durch Division der Schrittgröße auf Bereiche gleicher Größe abgebildet werden, wobei eine stufenähnliche Charakteristik entsteht. [Sym04]. Um die errechneten Ortsfrequenzen in Relation zur Wahrnehmung des menschlichen Auges zu reduzieren wird hierfür eine Quantisierungsmatrix verwendet. Diese beinhaltet separate Werte für jeden DCT Koeffizienten. Die Schrittgröße setzt sich für AC-Werte zusammen aus dem korrespondierenden Quantisierungswert der Quantisierungsmatrix und einem Quantisierungsfaktor. Der Quantisierungsfaktor dient der Steuerung der Bildqualität und kann einen Wert zwischen 1 und 31 annehmen, wobei ein Quantisierungsfaktor von 1 für eine hohe Bildqualität sorgt, ein Faktor von 31 hingegen für eine stark reduzierte. Da das Auge sensibel gegenüber großräumigen Luminanzfehlern ist, wird der DC durch eine feste Schrittgröße von 8 dividiert. [11193] Eine Implementierung des vorgestellten Algorithmus ist in Listing 2.1 zu sehen.

```
def quantize(dct, quantizer, MQuant=1):
    result = np.empty_like(dct)
    for x, row in enumerate(dct):
        for y, coefficient in enumerate(row):
            if x == 0 and y == 0:
                result[x][y] = int(coefficient / 8)
            else:
                result[x][y] = int( 8 * coefficient / (MQuant *
                    quantizer[x][y] ))
    return result
```

Listing 2.1: Implementierung des Quantisierungsprozesses nach MPEG-1 Standard

3 Redundanzreduktion

<Text>

3.1 Entropiecodierung

3.2 Inter- und Intraprediction

3.3 Motion Compensation

4 Ausblick

ÄÖÜäöüß

5 Zusammenfassung

ÄÖÜäöüß

Appendices

Chroma Subsampling Artefakte

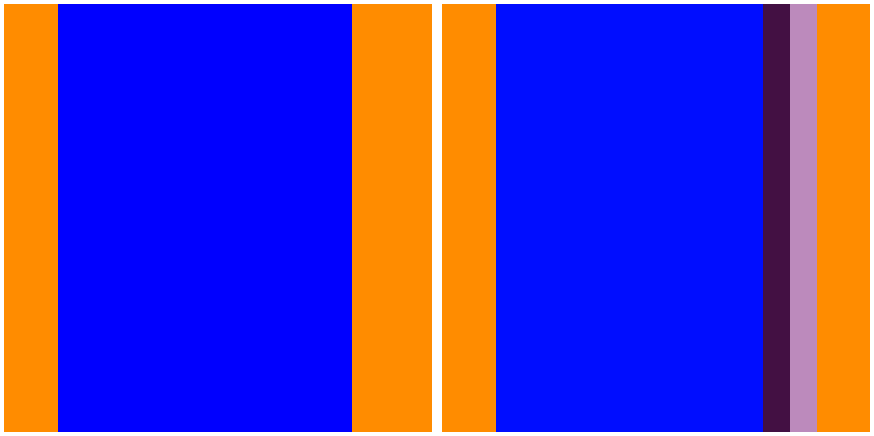


Abbildung .1: Artefakte durch Chroma Subsampling
Links: Original, Rechts: Subsampled. Die rechte Kante des blauen Farbblocks liegt in gesubsampleten 2x2 Blöcken, wodurch Artefakte entstehen. Die linke Kante liegt zwischen zwei 2x2 Blöcken, weshalb es zu keiner falschen Darstellung kommt.

Literatur

[11193] ISO/IEC 11172-2:1993. *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video*. Standard. International Organization for Standardization, 1993.

[Akr14] Shahriar Akramullah. *Digital Video Concepts, Methods, and Metrics*. Berkeley, CA: Apress, 2014.

[Atm] AtmWiki. *Ortsfrequenz*. URL: `http://www.otterstedt.de/wiki/index.php/Ortsfrequenz` (besucht am 02. 12. 2016).

[ITU95] ITU-T. *Recommendation ITU-R BT.601-5: Studio encoding Parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios*. 1995.

[Poy] Charles Poynton. *Chroma subsampling notation*. URL: `http://www.poynton.com/PDFs/Chroma_subsampling_notation.pdf` (besucht am 30. 11. 2016).

[Sym04] Peter Symes. *Digital Video Compression*. The McGraw-Hill Companies, Inc, 2004.