



**Hochschule Darmstadt**  
- Fachbereich Informatik -

# **Grundlagen der Videokompression**

Seminararbeit im Kurs  
**Wissenschaftliches Arbeiten in der Informatik 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 1. Dezember 2016

## *Erklärung*

# 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.

## *Abstrakt*

# Inhaltsverzeichnis

<b>Erklärung</b>	<b>iii</b>
<b>Abstrakt</b>	<b>v</b>
<b>Abbildungsverzeichnis</b>	<b>ix</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Irrelevanzreduktion</b>	<b>3</b>
2.1 Chroma Subsampling . . . . .	3
2.2 Diskrete Kosinus Transformation . . . . .	5
2.3 Quantisierung . . . . .	5
<b>3 Redundanzreduktion</b>	<b>7</b>
3.1 Entropiecodierung . . . . .	7
3.2 Inter- und Intraprediction . . . . .	7
3.3 Motion Compensation . . . . .	7
<b>4 Ausblick</b>	<b>9</b>
<b>5 Zusammenfassung</b>	<b>11</b>
<b>Literaturverzeichnis</b>	<b>xv</b>





# Abbildungsverzeichnis

2.1	Artefakte durch Chroma Subsampling . . . . .	4
-----	--	---



# 1 Einleitung

<Text>

## 1 *Einleitung*

## 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 stärker wahr, als Änderungen im Farbton. 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 benötigen wir eine getrennte Darstellung dieser Informationen. Hierfür wird im MPEG-1 Standard die  $YC_B C_R$  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:

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

## 2 Irrelevanzreduktion

$$U = (B - Y) \cdot 0.493$$

$$V = (R - Y) \cdot 0.877 \text{ [ITUb]}$$

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 des Luma-Wertes 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 enorme Möglichkeit der Kompression, die allerdings nicht verlustfrei abläuft. Artefakte können bei Verwendung dieser Methode vor allem bei scharfen, farbigen Kanten entstehen, wenn diese durch einen gesampleten Block verlaufen. In Abbildung 2.1 ist dieser Sachverhalt dargestellt.

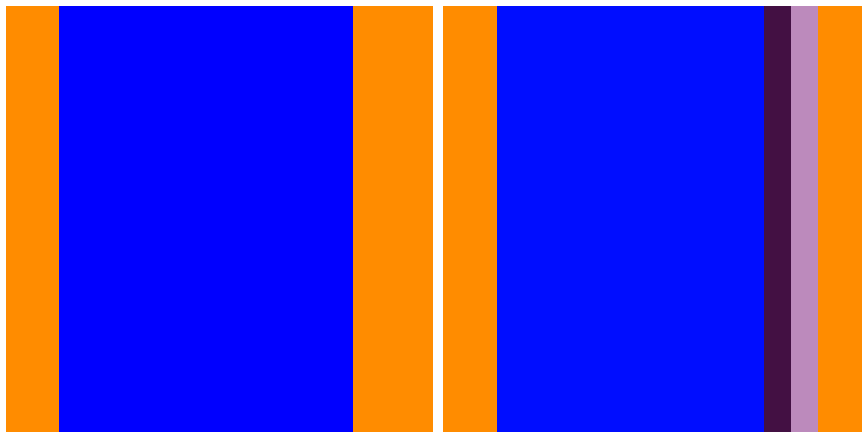


Abbildung 2.1: Artefakte durch Chroma Subsampling

*Links: Original, Rechts: Subsampled. Die rechte Kante des blauen Farbblocks liegt in gesubsampten 2x2 Blöcken, wodurch Artefakte entstehen. Die linke Kante liegt zwischen zwei 2x2 Blöcken, weshalb es zu keiner falschen Darstellung kommt.*

## 2.2 Diskrete Kosinus Transformation

\* DCT ist eine spezielle Form der Fourier-Transformation \* Fourier-Transformation  
 aproxiert eine Funktion mittels Sinus-Funktionen \* 4 Probleme [Sym] S.71: \* \*It  
 assumes that the time domain signal is infinite in extent\* \* \*It assumes continuous  
 functions in time\* \* Nicht ohne weiteres auf 2D anwendbar \* Generierte Koeffizienten  
 sind 2D (Amplitude + Phase bzw. sinus + cosine) \* DCT funktioniert, solange nach  
 dem Nyquist Theorem gesampelt wurde (warum?) \* Nutzt außerdem noch einen  
 Effekt aus, an den ich mich gerade nicht mehr erinnere -> bandwidth-limited data \*  
 DCT erlaubt uns Ortsfrequenzen zu extrahieren (warum? wodurch?)

\* Formel:

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

\* Quelle [Sym] S.75 \* Implementierung: Siehe src/dct.py

\* Es wird eine zweidimensionale DCT verwendet.

\* Wann funktioniert sie nicht so gut?

## 2.3 Quantisierung





# **3 Redundanzreduktion**

<Text>

## **3.1 Entropiecodierung**

## **3.2 Inter- und Intraprediction**

## **3.3 Motion Compensation**

### *3 Redundanzreduktion*

## 4 Ausblick

ÄÖÜäöüß



# 5 Zusammenfassung

ÄÖÜäöüß



# Literaturverzeichnis

- [Akr14] Shahriar Akramullah. *Digital Video Concepts, Methods, and Metrics*. Apress, 2014.
- [Dan06] Wilfried Dankmeier. *Grundkurs Codierung Verschlüsselung, Kompression, Fehlerbeseitigung*. Wiesbaden, 3., überarb. und erw. Aufl. edition, 2006.
- [ITUa] ITU-T. H.261: Video codec for audiovisual services at  $p \times 64$  kbits.
- [ITUb] 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.
- [Moo93] Andrew Moore. The theory of CCITT recommendation h. 261, video codec for audiovisual services at  $p \times 64$  kbit/s and review of such a codec. *a*, 16:37, 1993.
- [Poy] Charles Poynton. Chroma subsampling notation.
- [Sym] Peter Symes. *Digital Video Compression*. The McGraw-Hill Companies, Inc.