



**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 28. November 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 . . . . .	4
2.3 Quantisierung . . . . .	4
<b>3 Redundanzreduktion</b>	<b>5</b>
3.1 Entropiecodierung . . . . .	5
3.2 Inter- und Intraprediction . . . . .	5
3.3 Motion Compensation . . . . .	5
<b>4 Ausblick</b>	<b>7</b>
<b>5 Zusammenfassung</b>	<b>9</b>
<b>Literaturverzeichnis</b>	<b>xv</b>





# **Abbildungsverzeichnis**



# 1 Einleitung

<Text>

## 1 *Einleitung*

## 2 Irrelevanzreduktion

\* Wie in Einleitung geschrieben: Irrelevanzreduktion basiert auf Psychovisuellen Effekten. Im wesentlichen wird ausgenutzt: \* Varianzen in der Helligkeit nimmt menschliches Auge besser wahr, als Varianzen im Farbton \* Niedrige Ortsfrequenzen nimmt menschliches Auge besser wahr als hohe (Was zu Hölle sind Ortsfrequenzen?) \* Quellen: Etwas Besseres als [Dan06] S.358 & S.359? -> Vllt [Akr14] S.13 \* pv. Effekte kann man sich so zu nutze machen, dass wir relevantere Informationen genauer Speichern, als weniger relevante \* Da Daten in RAW RGB im Normalfall vorliegen müssen wir die Daten erst vorbereiten, um sie dann nutzen zu können. \* Um an Helligkeit heran zu kommen -> RGB -> YUV (Oder YCrCb, wo ist der Unterschied?) \* Weiterverarbeitung via Subsampling \* Um an Ortsfrequenzen heran zu kommen -> DCT \* Weiterverarbeitung via Quantisierung \* Wobei allerdings nicht direkt weniger Speicherplatz verbraucht wird, sondern vielmehr die Daten besser komprimierbar für RLE gemacht werden, welches im nächsten Kapitel behandelt wird.

### 2.1 Chroma Subsampling

\* RGB -> YUV \* Formeln nach <https://www.fourcc.org/fccyvrgb.php> -> Gibt es vllt. eine bessere Quelle? Kann ich das nach CCIR 601 ableiten? \* 4:2:0 -> 50% Komprimierung!

## 2.2 Diskrete Kosinus Transformation

\* DCT ist eine spezielle Form der Fourier-Transformation \* Fourier-Transformation  
aproximiert eine Funktion mittels Sinus-Funktionen \* 4 Probleme [Sym] S.71: \* \*It  
assumes that the time domain signal is infinite in extent\* \* \*It assumes continous  
funtions in time\* \* Nicht ohne weiteres auf 2D anwendbar \* Generierte Koeffitienten  
sind 2D (Amplitude + Phase bzw. sinus + cosine) \* DCT funktioniert, solange nach  
dem Nyquist Theorem gesampled 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**





## 4 Ausblick

ÄÖÜäöüß

#### *4 Ausblick*

# 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.
- [ITU] ITU-T. H.261: Video codec for audiovisual services at  $p \times 64$  kbits.
- [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.
- [Sym] Peter Symes. *Digital Video Compression*. The McGraw-Hill Companies, Inc.