

Friedrich-Alexander-Universität Erlangen-Nürnberg

**Lehrstuhl für Multimediakommunikation und  
Signalverarbeitung**

Prof. Dr.-Ing. André Kaup

Forschungspraktikum

**Schätzung der Dekodierenergie von  
codierten Videos**

von Peng Wu

April 2016

Betreuer: Dipl.-Ing. Christian Herglotz

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
<b>2</b>	<b>Algorithmus</b>	<b>2</b>
<b>3</b>	<b>Implementierung</b>	<b>5</b>
<b>4</b>	<b>Ergebnis</b>	<b>7</b>
<b>5</b>	<b>Zusammenfassung</b>	<b>8</b>
<b>A</b>	<b>Anhang Kapitel</b>	<b>9</b>
	A.1 Aufgabenstellung . . . . .	9
	<b>Literaturverzeichnis</b>	<b>11</b>

# Kapitel 1

## Einführung

Die Technologien zur Codierung von Videos haben sich mit hoher Geschwindigkeit entwickelt. Zur gleichen Zeit benutzt man immer mehr tragbare Geräte wie Smartphones oder Tablet PCs statt traditioneller Desktop PCs. Diese Geräte können in Echtzeit die Bitströme der Videos direkt im Internet oder WLAN decodieren, sodass Benutzer die Videos jederzeit und überall sehen können. Aufgrund der Komplexität der Decoder verbraucht dieser Prozess viele Energie von der Batterie. In den letzten Jahren beschäftigt sich die Forschung zunehmend mit energieeffizienten Algorithmen, um die Betriebszeit von tragbaren Geräten zu erhöhen.

Aktuelle Forschung am Lehrstuhl hat gezeigt, dass sich die Decodierenergie für HEVC-codierte Videosequenzen anhand verschiedener Parameter sehr gut schätzen lässt. In meinem Forschungspraktikum untersuche ich, ob dies auch für den H.264 Codierstandard zutrifft.

# Kapitel 2

## Algorithmus

Um die Decodierenergie für H.264-codierte Videosequenzen durch die Parameter zu schätzen und bestimmen, fünf Energiemodelle mit verschiedenen Parameter werden in diesem Praktikum angewendet.

Das einfachste Energiemodell zur Schätzung des Energieverbrauchs basiert auf der Decodierzeit [HWK15]:

$$\hat{E}_{dec} = E_0 + P_{mean} \cdot t_{dec}, \quad (2.1)$$

wobei  $\hat{E}_{dec}$  die Schätzenergie,  $t_{dec}$  die Decodierzeit des Systems,  $P_{mean}$  der Mittelwert der Prozessleistung und  $E_0$  ein konstanter Ausgleich ist. Im Folgenden wird dieses Modell als “T ”(Modell basiert auf der Decodierzeit) bezeichnet. Dieser Algorithmus zeigt, dass die Decodierzeit des Systems näherungsweise linear zu der Decodierenergie ist.

Die Energiemodelle basiert auf High-Level Parameter, die durch die Funktionen mit den high-level Varianten beschrieben werden. Mit den Eigenschaften der Bitströmen können die Auflösung des Frames, die Framerate und der Quantisierungsparameter in folgendes Modell [LMF12] angewendet werden:

$$\hat{E} = P_{max} \cdot \left(\frac{S}{S_{max}}\right)^{c_s} \cdot \left(\frac{f}{f_{max}}\right)^{c_f} \cdot \left(\frac{q}{q_{min}}\right)^{c_q} \cdot t_{dec}, \quad (2.2)$$

wobei  $\hat{E}$  die Schätzenergie, S die Auflösung des Frames, f die Framerate, q der Quantisierungsparameter und  $t_{dec}$  die Decodierzeit des Systems ist.  $c_s, c_f, c_q$  sind die Parameter, die durch die Algorithmen Curve Fixing und Parameter Training mit verschiedenen Parameter der Bitströmen optimiert und bestimmt werden können. Im Folgenden wird dieses Modell als “ $H1_T$ ” (Modell basiert auf der high-level Eigenschaften mit der Decodierzeit) bezeichnet.

Ein weitere Modell wird mit der Bitrate und der Wiederholrate der Intraframe zur Schätzung der Prozessleistung beschrieben [RP13]:

$$\hat{E} = (c_1 \cdot \alpha \cdot b + c_2 \cdot \alpha + c_3 \cdot b + c_4) \cdot t_{dec}, \quad (2.3)$$

wobei  $\hat{E}$  die Schätzenergie, b die Bitrate,  $\alpha$  die Intra-Refresh-Rate und  $t_{dec}$  die Decodierzeit des Systems ist.  $c_1, c_2, c_3, c_4$  sind die Systemparameter, die durch die Algorithmen Curve Fixing und Parameter Training mit verschiedenen Parameter der Bitströmen optimiert und bestimmt werden können. Im Folgenden wird dieses Modell als “ $H2_T$ ” (Das zweite Modell basiert auf der high-level Eigenschaften mit der Decodierzeit) bezeichnet. Ein modifizierte Modell vom Modell  $H2_T$  [HSR<sup>+</sup>ns] wird mit der normalisierten Bitrate beschrieben, die mit den Durchschnittswert der Bits von jedem Pixel statt der Bitrate in der Funktion (2.3) gezeichnet sind:

$$\hat{E} = (c_1 \cdot \alpha \cdot b_{pixel} + c_2 \cdot \alpha + c_3 \cdot b_{pixel} + c_4) \cdot N \cdot S, \quad (2.4)$$

wobei  $\hat{E}$  die Schätzenergie, S die Auflösung des Frames, N die Anzahl der Frames,  $\alpha$  die Intra-Refresh-Rate und  $b_{pixel}$  die normalisierten Bitrate ist.  $c_1, c_2, c_3, c_4$  sind die Systemparameter, die durch die Algorithmen Curve Fixing und Parameter Training mit verschiedenen Parameter der Bitströmen optimiert und bestimmt werden können. Durch die Multiplikation mit der vollständigen Anzahl von den Pixel und die Auflösung des Frames kann man den Vorteil erhalten, dass dieses Modell von der Decodierzeit unabhängig ist. Im Folgenden wird dieses Modell als “H2” (Das zweite Modell basiert auf der high-level Eigenschaften ohne Decodierzeit) bezeichnet.

Das folgende Modell [HK15] ist durch die Varianten: die Bitrate, die Anzahl der Frames und die Auflösung des Frames gebaut:

$$\hat{E} = C + S \cdot N \cdot (\alpha + \beta \cdot b^\gamma), \quad (2.5)$$

wobei  $\hat{E}$  die Schätzenergie, S die Auflösung des Frames, N die Anzahl der Frames,  $\alpha$  die Intra-Refresh-Rate und b die Bitrate ist. C,  $\alpha$ ,  $\beta$ ,  $\gamma$  sind die Systemparameter, die durch die Algorithmen Curve Fixing und Parameter Training mit verschiedenen Parameter der Bitströmen optimiert und bestimmt werden können. Im Folgenden wird dieses Modell als “H3” (Das dritte Modell basiert auf der high-level Eigenschaften) bezeichnet.

Um die Schätzfähigkeiten der Modellen zu bestimmen und zeigen, der relative Schätzfehler [HSR<sup>+</sup>ns] ist als Kriterium angewendet:

$$\varepsilon = \frac{\hat{E}_{dec} - E_{dec}}{E_{dec}}, \quad (2.6)$$

wobei  $\hat{E}$  als die Schätzenergie und  $E_{dec}$  als die durch das bestimmten Decodiersystem gemessenen Decodierenergie bezeichnet ist. Mit dieser Methode kann die Schätzfehler unabhängig von die Auflösung und der Länge der Sequenzen sein.

# Kapitel 3

## Implementierung

Zuerst codiere ich in Matlab verschiedene Sequenzen in verschiedenen Qualitäten mit der JM Software, die die offizielle Referenz-Software für den H.264 Codec ist [HHI10]. Die Sequenzen von Class A mit 16 Frames und Class B,C,D mit 8 Frames werden mit sieben QP (10, 15, 20, 25, 30, 35, 40, 45) und sechs Konfigurationen (encoder, encoder baseline, encoder max performance, intra, lowdelay, and randomaccess) encodiert, die verschiedenen Encodierungsverfahren und verschiedenen Intra-Frame-Rate haben. Die Sequenzen und ihre Eigenschaften werden in Tabelle I gelistet.

Nach der Codierung teste ich die Parameter der Decodierung für die Modelle. Ich benutze FFmpeg [FFmte] für die Decodierung. Die Geschwindigkeit der Decodierung von FFmpeg ist schneller als von JM. Mit “time”-Funktion teste ich die Decodierzeit. Die Decodierenergie ist durch das Decodierungssystem “FFmpeg dual core on Pandaboard” [HSR<sup>+</sup>ns] gemessen. Das Decodierungssystem ist eine 32-bit-ARM Ebene am Pandaboard [pan15], die in den Smartphones und den Tablet PCs benutzt wird. Die Messungen der Decodierenergie werden mit dem System im FFmpeg multimedia framework [fra] mit den dualen Cores durchgeführt.

Insgesamt gibt es 720 Sequenzen. Ich speichere den Namen, die Framerate, die Decodierzeit, die Bytes, die Auflösung, die Anzahl der Frames, die Anzahl von I-Frames und die Quantisierungsparameter jeder Sequenz in einer Matrix.

Anschließend wende ich diese Parameter auf fünf Energiemodelle an, um die Schätzge-

---

nauigkeiten zu bestimmen und mit den gegebenen Werten des HEVC zu vergleichen.

Tabelle I: Die Sequenzen und ihre Eigenschaften

Name	Class	Auflösung	Framerate
PeopleOnStreet	A	2560×1600	30
Traffic	A	2560×1600	30
BasketballDrive	B	1920 × 1080	50
BQTerrace	B	1920 × 1080	60
Cactus	B	1920 × 1080	50
Kimono1	B	1920 × 1080	24
ParkScene	B	1920 × 1080	24
BasketballDrill	C	832 × 480	50
BQMall	C	832 × 480	60
PartyScene	C	832 × 480	50
RaceHorses	C	832 × 480	30
BasketballPass	D	416 × 240	50
BlowingBubbles	D	416 × 240	50
BQSquare	D	416 × 240	60
RaceHorses	D	416 × 240	30



# Kapitel 4

## Ergebnis

Die relativen Schätzfehler für das Decodierungssystem “FFmpeg dual core on Panda-board” von fünf Modellen werden in der Tabelle II dargestellt.

Tabelle II:

	T(sys)	T(dec)	$H1_T$	$H2_T$	H2	H3
Decodierungssystem	22.14%	3.45%	14.92%	18.60%	22.99%	21.27%

Der Wert von T(sys) wird auf dem lokalen Rechner, der auf dem CentOS Linux 7 System mit der 64-bit-Architektur und dem Intel-Core basiert, mit den Decodierzeiten aus FFMpeg mit der “time”-Funktion ausgerechnet. Der Wert von T(dec) wird mit den Decodierzeiten direkt bei der Messung des Decodierungssystems ausgerechnet.

Das Ergebnis von T(dec) ist besser als von T(sys), weil die Decodierzeiten bei der Messung des Systems den Decodierenergien gut passen. Damit schätzt dieses Modell genauer.

Die gegebene Werte des HEVC[HSR<sup>+</sup>ns] sind in Tabelle III gezeigt:

Tabelle III:

	T	$H1_T$	$H2_T$	H2	H3
Decodierungssystem	27.12%	23.89%	24.67%	21.03%	16.87%

Im Vergleich kann man sehen, dass die Schätzgenauigkeiten von obenstehenden Modellen für den H.264 Codierstandard besser als die für den HEVC Codierstandard sind.

# Kapitel 5

## Zusammenfassung

In diesem Forschungspraktikum habe ich die in verschiedenen Qualitäten codierten Sequenzen auf Decodierzeit und -energie getestet und die Parameter der Eigenschaften benutzt, um die Modelle für den H.264 Codierstandard zu untersuchen. Durch die Ergebnisse bestimme ich die Schätzgenauigkeiten, die zeigen, dass die Modelle für den H.264 Codierstandard anwendbar sind. Die Ergebnisse zeigt, dass die Schätzgenauigkeit vom Modell "T" mit den Decodierzeiten von der Messung am Pandaboard am besten ist. Die Ergebnisse von den Modellen " $H1_T$ " und " $H2_T$ " sind auch besser als die anderen Modelle, die nicht mit der Decodierzeit gerechnet werden. Die Schätzgenauigkeiten von den Modellen mit der Decodierzeit für den H.264 Codierstandard sind besser als den Modellen für den HEVC Codierstandard. Auf der Gegenseite sind die Schätzgenauigkeiten von den Modellen ohne Decodierzeit für den H.264 Codierstandard schlimmer als den Modellen für den HEVC Codierstandard.

# Anhang A

## Anhang Kapitel

### A.1 Aufgabenstellung

Lehrstuhl für Multimediakommunikation  
und Signalverarbeitung  
Prof. Dr.-Ing. A. Kaup



## **Forschungspraktikum**

für

**Frau Peng Wu**

### **Schätzung der Dekodierenergie von codierten Videos**

Aktuelle Forschung am Lehrstuhl hat gezeigt, dass sich die Decodierenergie für HEVC-codierte Videosequenzen anhand verschiedener Parameter sehr gut schätzen lässt. Nun soll untersucht werden, ob dies auch für andere Codierstandards zutrifft. Hierzu können verschiedene Standards (z.B H.264 oder VP9) untersucht werden.

Frau Wu soll nun für ausgewählte Codierstandards verschiedene Sequenzen in verschiedenen Qualitäten codieren und auf Decodierzeit und –energie testen. Anschließend soll sie diese Sequenzen auf verschiedene, einfache Energiemodelle anwenden, Schätzgenauigkeiten bestimmen und mit den gegebenen Werten des HEVC vergleichen.

Start: 1.1.2016

# Literaturverzeichnis

- [FFmte] FFmpeg: [Online]. <http://ffmpeg.org/>, Offizielle Webseite.
- [fra] FRAMWORK., FFmpeg multimedia: [Online]. Available: <http://ffmpeg.org/>
- [HHI10] HHI, [Online]. h.: H.264/AVC JM Reference Software, 2010
- [HK15] HERGLOTZ, C. ; KAUP, A.: Estimating The HEVC Decoding Energy Using High-Level Video Features. In: *European Signal Processing Conf. (EUSIP-CO)*. Nice, France, August 2015, S. 1591–1595
- [HSR<sup>+</sup>ns] HERGLOTZ, Christian ; SPRINGER, Dominic ; REICHENBACH, Marc ; STABERNACK, Benno ; KAUP, André: Modeling the Energy Consumption of the HEVC Decoding Process, submitted to IEEE transactions.
- [HWK15] HERGLOTZ, Christian ; WALENCIK, Elisabeth ; KAUP, Andre: Estimating the HEVC decoding energy using the decoder processing time. In: *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on IEEE*, 2015, S. 513–516
- [LMF12] LI, Xin ; MA, Zhan ; FERNANDES, Felix C.: Modeling power consumption for video decoding on mobile platform and its application to power-rate constrained streaming. In: *Visual Communications and Image Processing (VCIP), 2012 IEEE IEEE*, 2012, S. 1–6
- [pan15] PANDABOARD.ORG.: [Online]. Available: <http://pandaboard.org/>, 2015.

- [RP13] RAOUFI, Pariya ; PETERS, Jochen: Energy-efficient wireless video streaming with H. 264 coding. In: *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on IEEE*, 2013, S. 1–6