Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Multimediakommunikation und Signalverarbeitung

Praktikumszwischenbericht

Herleitung und Auswertung eines Zeitmodells aus dem Energiemodell in der HEVC Videocodierung

von Roman Guer

Betreuer: Christian Herglotz, Dipl.-Ing.

Inhaltsverzeichnis

1 Einleitung

Im Zeitalter von tragbaren Endgeräten wie Smartphones oder Tablets, ist der Anspruch an der Videocodierung gestiegen. Eine der wichtigsten Herausforderungen ist die Reduktion des Energieverbrauchs, da die Lebenszeit der Batterie begrenzt ist.
In den letzten Jahren wurde an einem Videostandard gearbeitet, der sich HEVC (High Efficiency Video Coding) nennt. Dieser soll als Nachfolger des etablierten H.264/AVC Standards sein und erzielt eine deutlich höhere Kompression.
Aufgrund des Bedeutungsgewinns von HEVC wurde in [2] und [3] ein Energiemodell vorgeschlagen, damit bei der Entwicklung von Decodern darauf zurückgegriffen werden kann, um den Energieverbrauch insbesondere bei mobilen Endgeräten zu optimieren.
In [1] wurde ein linearer Zusammenhang zwischen Prozesszeit und Energieverbrauch festgestellt. Ziel dieses Praktikums ist, die Herleitung eines Zeitmodells mit Hilfe des Energiemodells aus [3]. Anschließend soll das Zeitmodell mit gemessenen Werten verglichen werden und wenn möglich verbessert werden.

2 Zeitmodell

2.1 Herleitung aus Energie

In [1] wurde ein statistischer Zusammenhang zwischen Energie und Zeit bewiesen. Mit Hilfe der linearen Regression, wurden Korrelationskoeffizienten für verschiedene Implementierung des HEVC Standards berechnet.
Im Praktikum wurde nur die Open-Source Implementierung libde265 näher behandelt. Bei dieser Variante, beträgt der Rangkorrelationskoeffizient nach SPEARMAN mit minimalen Unterschied für Einzel- oder Doppelkernprozessoren $r_{s_k} \sim 0.9996$ [1].
Die entsprechende Regressionsgerade unter Verwendung des Rangkorrelationskoeffizienten für Einzelkernprozessoren lautet [1]:

$$E_i = 0.513 \cdot t_i + 0.0534 \tag{2.1}$$

Der Index i bezieht sich dabei auf das i-te Video aus der Messreihe von N Videos.
Stellt man die Gleichung aus 2.1 nach der Zeit um, erhält man folgende Gleichung:

$$T_i = 1.9493 \cdot e_i - 0.1041 \tag{2.2}$$

2.2 Berechnung der geschätzten Zeit aus der Energie

In [3] wurden M aussagekräftige Features ermittelt, die die Energie eines HEVC-Videos bestimmen. Gegeben waren die spezifischen Energien E_i für jedes Feature eines Decodiervorgangs. Diese Features, wurden je nach Video entsprechend N_i oft ausgeführt , sodass die Gesamtenergie wie folgt angegeben werden kann:

$$E_{ges} = \sum_{i=0}^{\mathrm{M}-1} E_i \cdot N_i \tag{2.3}$$

Eine direkte Verwendung des Zusammenhangs aus Gleichung 2.2 hatte ein negatives Ergebnis zur Folge, sodass eine leichte Variation dieser verwendet wurde:

$$T_{ges} = [\sum_{i=0}^{\mathrm{M}-1} (1.9493 \cdot E_i) \cdot N_i] - 0.1041 \tag{2.4}$$

Der Faktor 0.1041 wurde lediglich einmal abgezogen und zwar von der spezifischen Energie E_0. Dieser Faktor kennzeichnet die benötigte Energie zum Initialisieren und Beenden eines Decodiervorgangs.
In Abbildung 2.1 ist der Zusammenhang zwischen spezifischer Energie und der daraus geschätzten Zeit für ein Video dargestellt.
Abbildung 2.2 zeigt die Relation zwischen geschätzter spezifischer Energie und gemessener Energie bzw. zwischen geschätzter und gemessener Zeit für eine Vielzahl an Videos. Der blaue Balken beschreibt dabei die aufsummierte geschätzte spezifische Energie. Die wahre gemessene Energie, wird durch den hellblauen Balken repräsentiert. Der gelbe Balken zeigt die Zeit an, die aus den spezifischen Energien berechnet wurde. Die tatsächlich gemessene Zeit für ein Video, wird durch den roten Balken beschrieben. Abbildung 2.3 zeigt einen Ausschnitt aus dem vorherigen Graphen für eine genauere Betrachtung. Dabei ist der lineare Zusammenhang deutlich erkennbar. Die absolute Abweichung ist weniger interessant als die relative bzw. mittlere Abweichung über alle Videos, da man dadurch bessere Aussagen über das Zeitmodell treffen kann. Die relative Abweichung wird folgendermaßen berechnet:

$$\epsilon_{E_{rel}} = \frac{(E_{est} - E_{\text{ground}-\text{truth}})}{E_{\text{ground}-\text{truth}}} \tag{2.5}$$

Analog dazu wird die relative Abweichung für die Zeit berechnet.
Mittlere und maximale Abweichung werden anhand der relativen Abweichung wie folgt angegeben:

$$\begin{aligned} \epsilon_{E_{mean}} &= \frac{1}{\mathrm{M}} \sum_{i=0}^{\mathrm{M}-1} |\epsilon_{E_{rel_i}}| \\ \epsilon_{E_{max}} &= \max_i |\epsilon_{E_{rel_i}}| \end{aligned} \tag{2.6}$$

Abbildungen 2.4 bis 2.7 zeigen die verschiedenen Abweichung von Energie und Zeit. Es ist keine Überraschung, dass die Abweichungen von Zeit höher ausfallen, als die der Energie. An manchen Stellen ist eine relative Abweichung von über 15% für die Zeit zu erkennen, was eine Verbesserung des Zeitmodells erfordert.
Die maximale Abweichung unterscheidet sich ungefähr um das doppelte, wohingegen der Unterschied in der mittleren Abweichung bei 2% liegt.

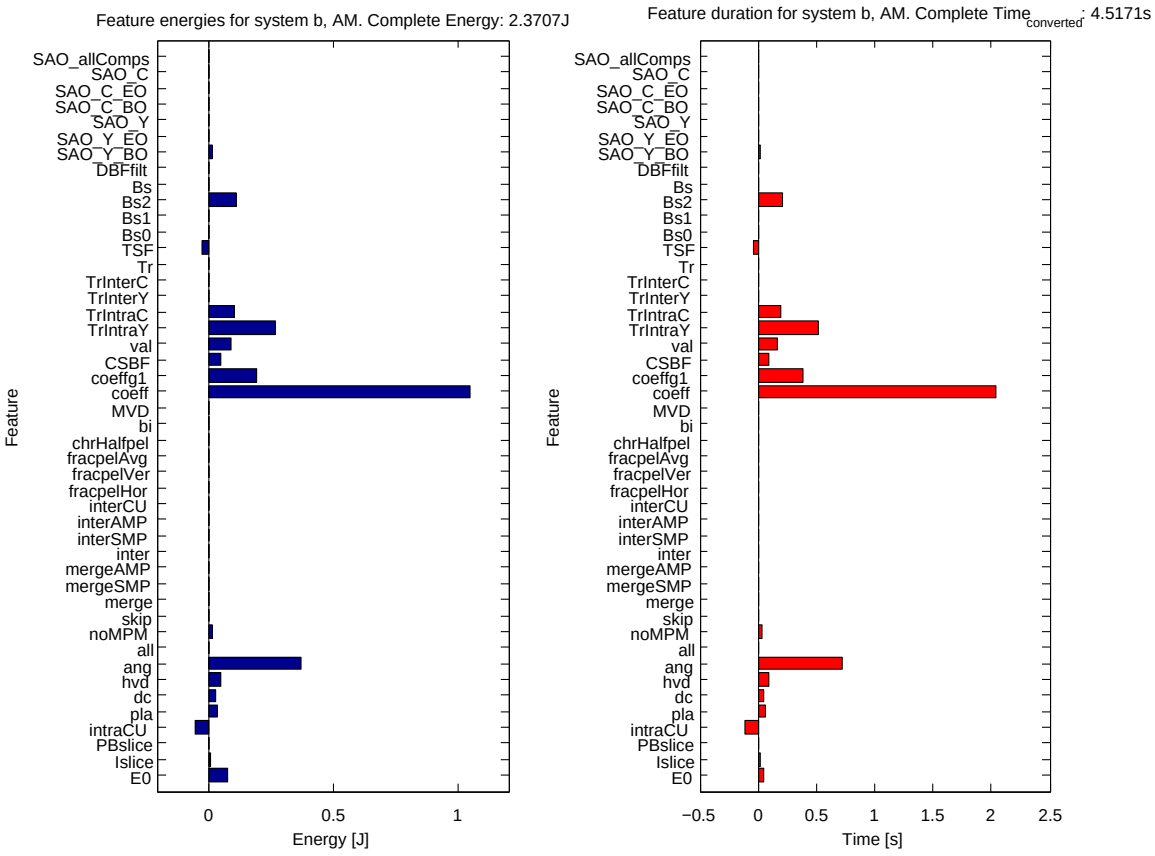

Abbildung 2.1: geschätzte Energie und Zeit für Features

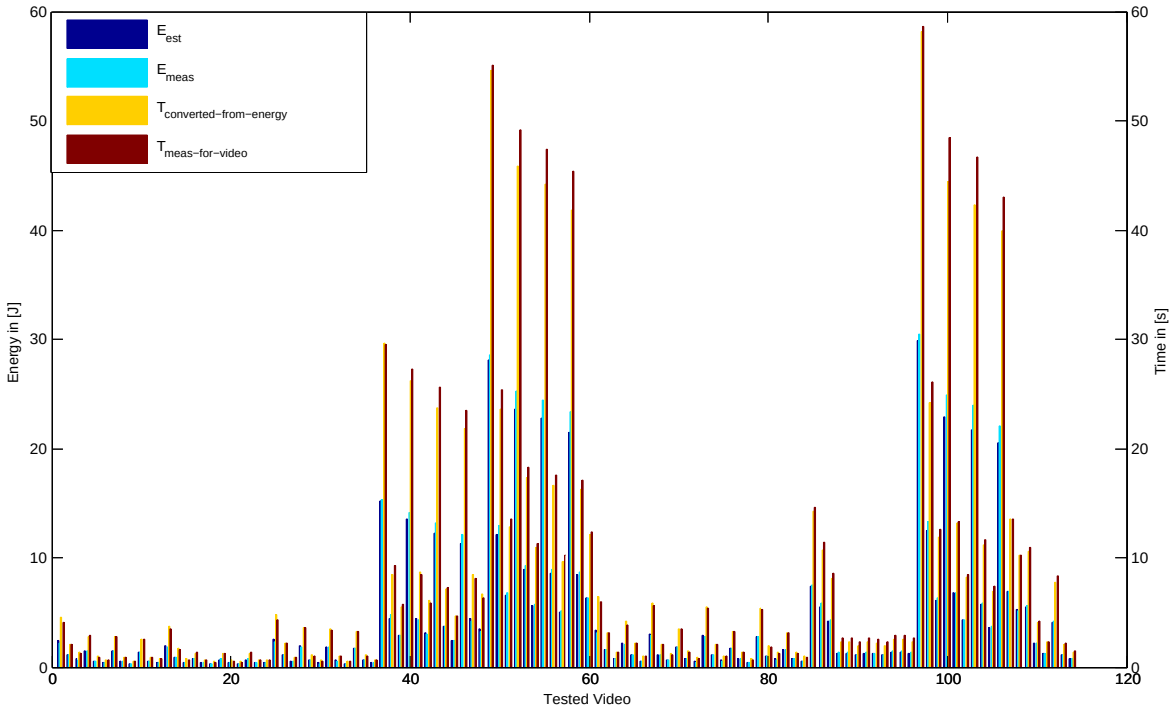

Abbildung 2.2: Zusammenhang zwischen Energie und Zeit

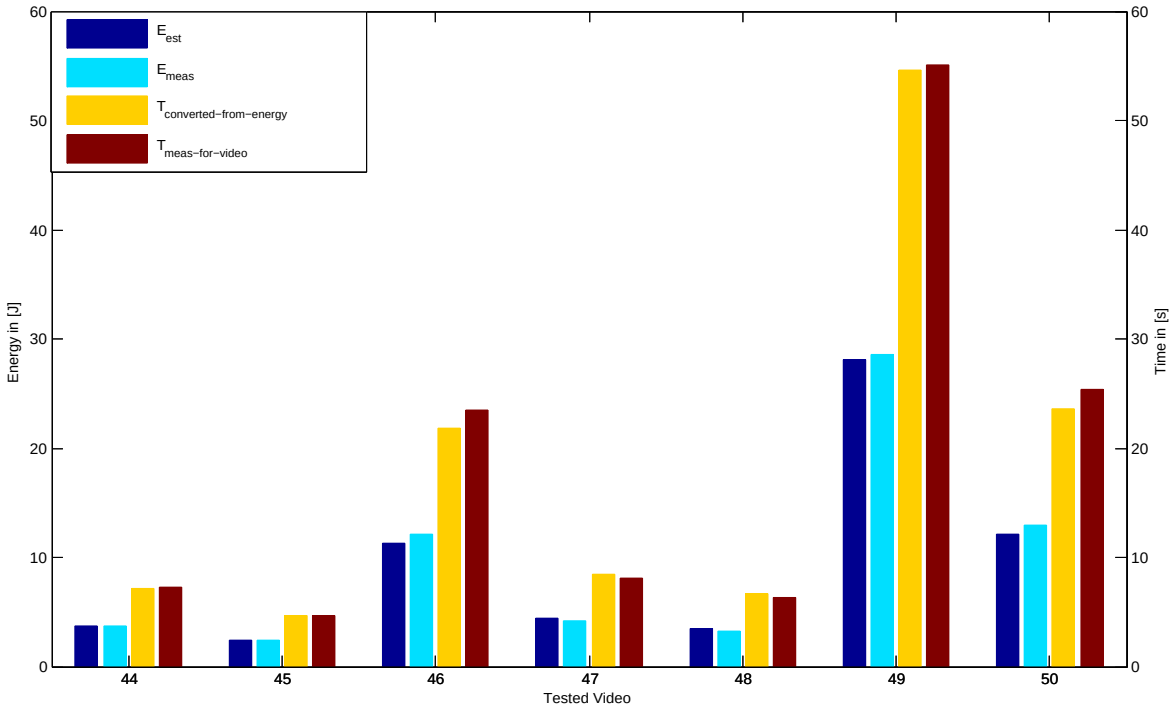

Abbildung 2.3: Ausschnitt aus Zusammenhang

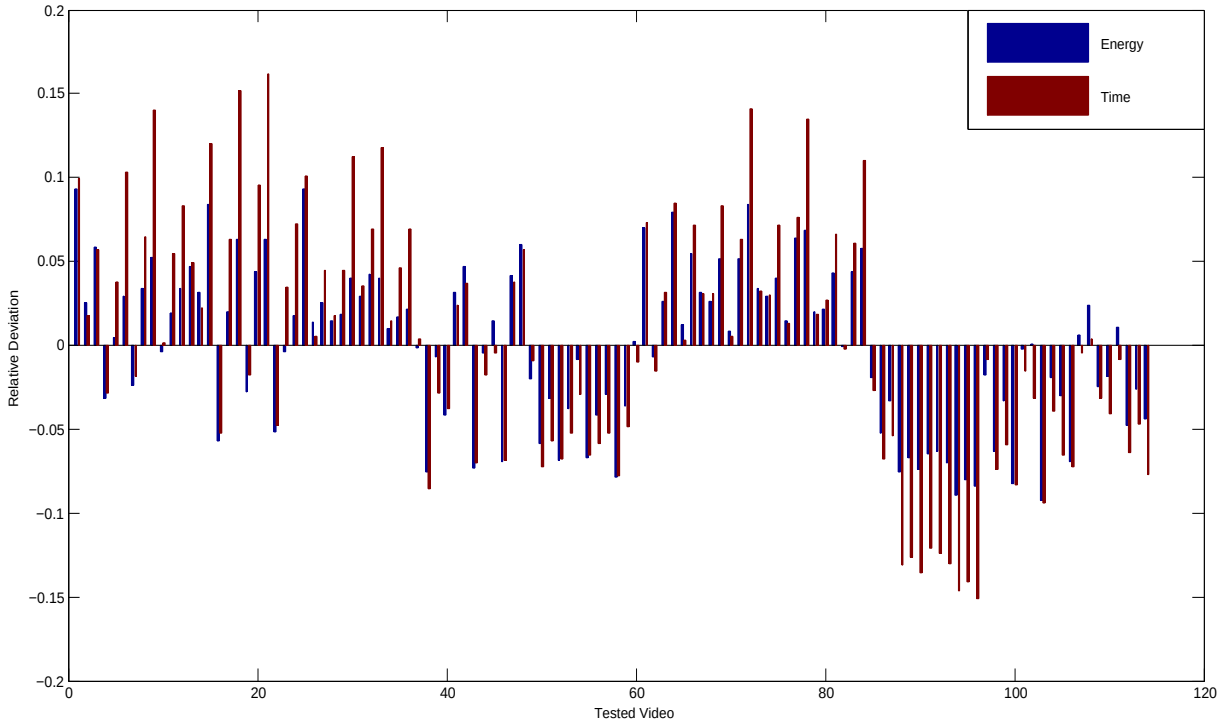

Abbildung 2.4: Relative Abweichung für Energie und Zeit

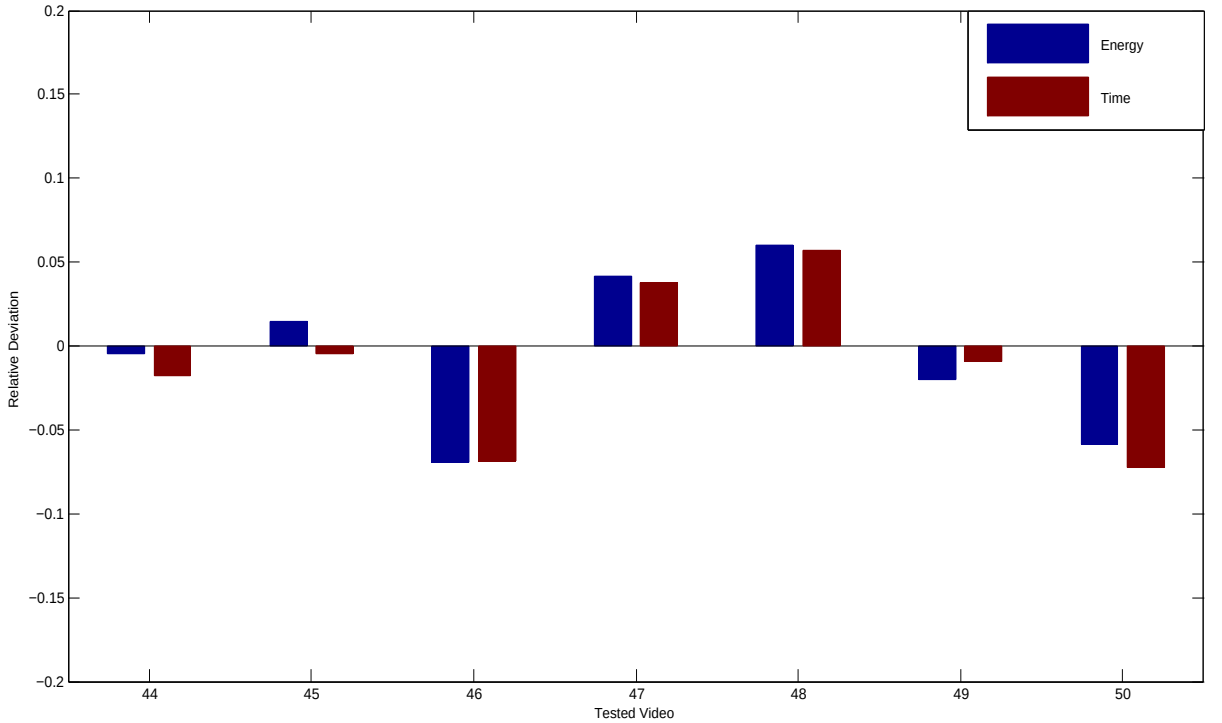

Abbildung 2.5: Ausschnitt aus rel. Abweichung

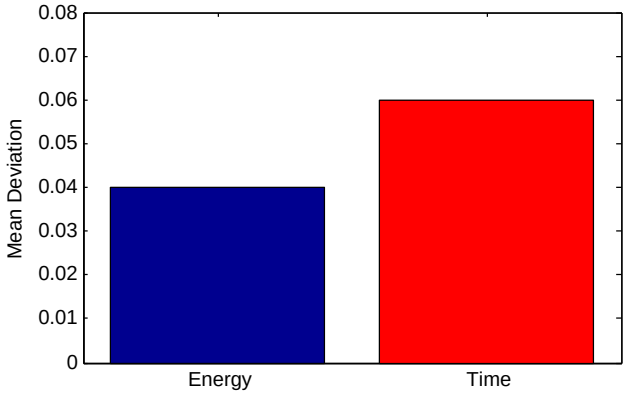

Abbildung 2.6: Mittlere Abweichung

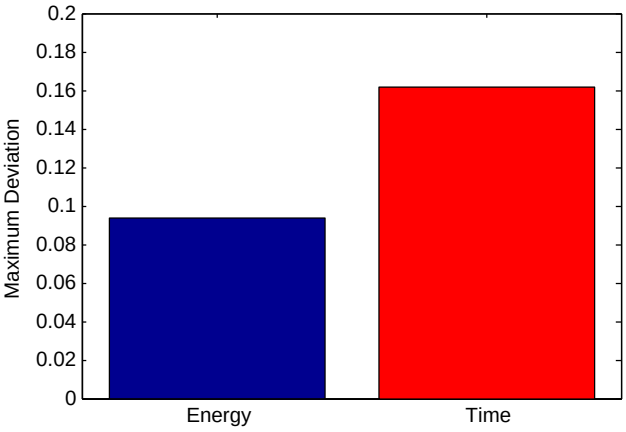

Abbildung 2.7: Maximale Abweichung

2.3 Vergleich mit geschätzter Zeit einzelner Features

Bisher wurde die umgerechnete Zeit mit der gemessenen Zeit eines Videos verglichen. Im nächsten Schritt wird die spezifische Zeit für die einzelnen Features aus [3] geschätzt. Diese Zeiten werden anschließend analog zur Gleichung 2.1 mit ihrer Häufigkeit multipliziert und schließlich aufsummiert, um eine geschätzte Gesamtdauer zu bestimmten.
Anschließend werden sowohl umgerechnete,spezifische und tatsächlich gemessene Zeit miteinander verglichen. In Abbildung 2.8 und 2.9 sind die unterschiedlichen Prozesszeiten für eine Vielzahl an Videos bzw. einen Ausschnitt zu sehen. Der blaue Balken zeigt die Zeit an, die aus der Energie umgerechnet wurde. Die über alle Features aufsummierte Zeit ist in hellgrün angegeben. Der rote Balken kennzeichnet die tatsächlich gemessene Zeit für das entsprechende Video. An manchen Stellen, wie beispielsweise das 44-te Video liegen die drei Zeiten sehr nah zusammen. Beim Video 55 ist allerdings ein Unterschied zwischen geschätzten und wahren Wert von bis zu 3-4 Sekunden festzustellen.

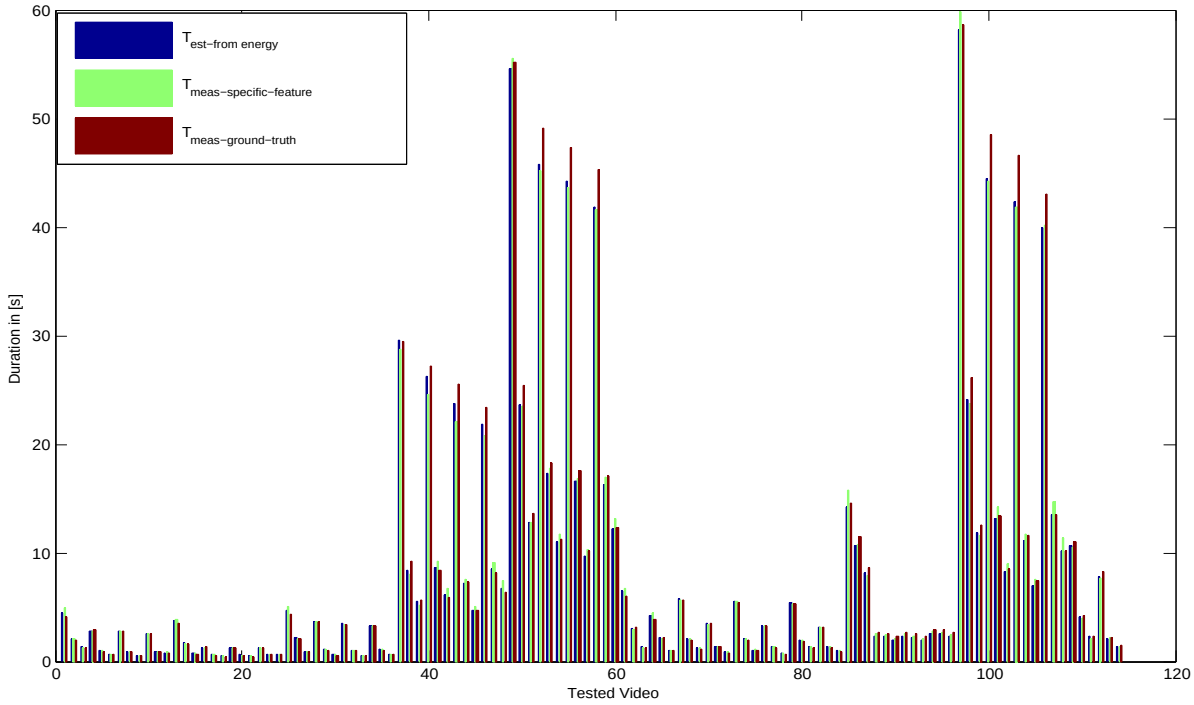

Abbildung 2.8: Umgerechnete,spezifische und gemessene Zeit

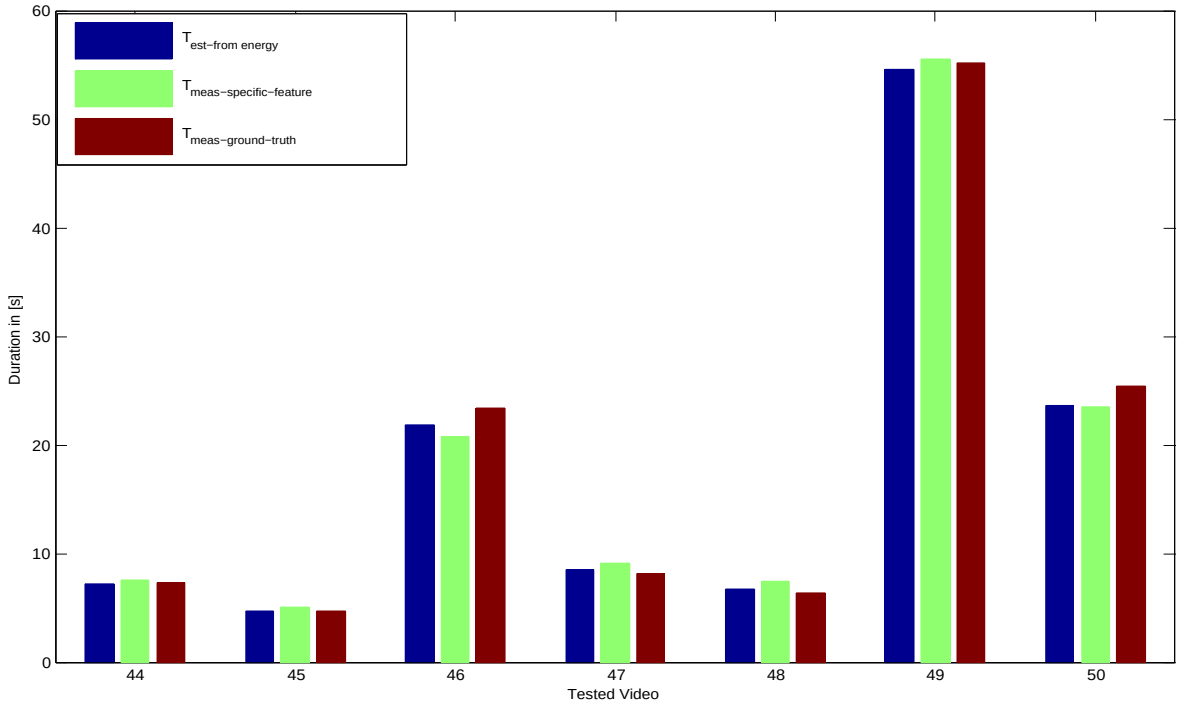

Abbildung 2.9: Ausschnitt aus Zusammenhang

In [3] wurde der relative Fehler berechnet um die einzelnen System und Implementierungen zu vergleichen. Diese Idee wurde ebenfalls für die Zeit aufgegriffen. Der relative Fehler lässt sich wie folgt berechnen:

$$\epsilon = \frac{\hat{T} - T_{\text{ground-truth}}}{T_{\text{ground-truth}}} \tag{2.7}$$

Die Variable $\hat{T}$ beschreibt eine geschätzte Zeit. In unserem Fall ist das einmal die aus den Energien abgeleitete Zeit und die geschätzte spezifische Zeit für die einzelnen Features. Abbildungen 2.10 und 2.11 zeigen die gruppierten vorkommenden relativen Fehler für eine Behältergröße von 1%.
Ebenfalls interessant ist der mittlerer und der maximale Fehler, die analog zu 2.6 berechnet werden. Die Ergebnisse dafür, sind in der Tabelle 2.3 dargestellt. Die Abweichung der Zeit aus der konvertierten Energie ist geringer, da diese manuell angepasst wurden.

	$\bar{\epsilon}$	ϵ_{max}
$T_{\text{converted}}$	5.98%	16.19%
T_{specific}	6.49%	20.08%

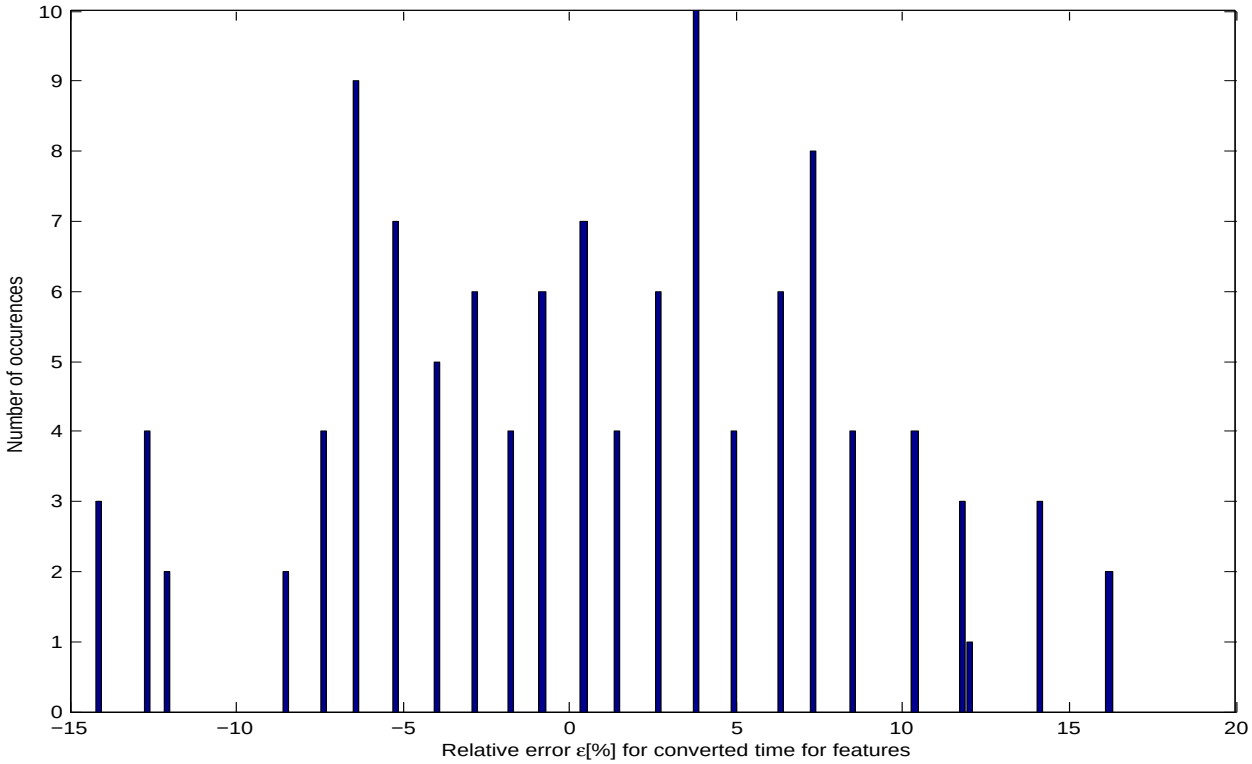

Abbildung 2.10: Relativer Fehler für umgerechnete Zeit

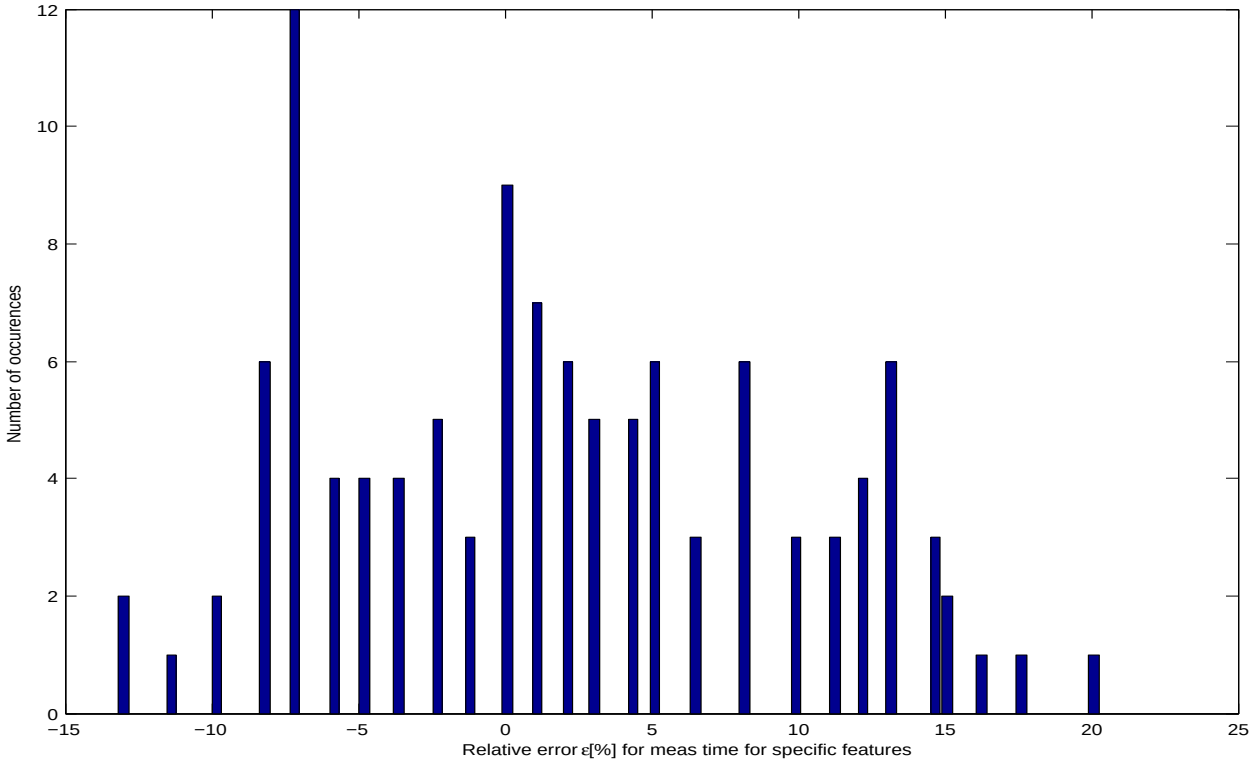

Abbildung 2.11: Relativer Feher für spezifische Zeit

2.4 Optimierung des Zeitmodells

Bei Betrachtung der Abbildung 2.8, sind bei einigen Videos große Abweichungen zu den realen Meßwerten zu sehen.
Deshalb wurde eine Methode zur Optimierung entwickelt, die die größten fünf Energien betrachtet und dessen Wert mit einem Skalar modifiziert. Bei dem Skalar wurde jeder Wert aus einem angegebenen Intervall bei allen möglichen Permutationen der fünf größten Energien durchprobiert. Es wurde sich für die Skalare entschieden, bei denen die relative Abweichung zum wahren Wert möglichst klein war.
Leider brachte diese Methode nur eine Verbesserung von einem Prozent. Der Grund hierfür könnte die Wahl des Intervalls sein, welches im Versuch [0.81.2] betrug. Eine Möglichkeit der Verbesserung wäre die Betrachtung aller Features und dessen Permutationen mit allen Werten aus einem Intervall.

2.4.1 Optimierung mit Hilfe von lsqcurvefit

Die Matlabfunktion lsqcurvefit löst ein nichtlineares Datenanpassungsproblem im Sinne der kleinsten Fehlerquadrate. Es werden drei Parameter benötigt. Einmal die Funktion, die angepasst wird und Eingangs- bzw. Ausgangsdaten. Die Eingangsdaten sind die einzelnen Features, wohingegen die Ausgangsdaten die tatsächlich gemessene Zeit für mehrere Videos repräsentieren. Die Funktion bestimmt einen Ausgangswert, indem eine Linearkombination aus alle Features aufsummiert wird.
Die Gleichung 2.8 soll noch einmal dies verdeutlichen. Die t-Werte sind Konstanten, die eine feste Zeit für ein bestimmtes Feature bei einmaliger Ausführung ist. Die n-Werte sind Häufigkeiten, wie oft ein bestimmtes Feature aufgerufen wird.

$$y_k = t_0 * n_0 + t_1 + n_1 + ... + t_m * n_m \tag{2.8}$$

Bei der Optimierung mit lsqcurvefit war eine Reduktion der mittleren Abweichung festzustellen. Es wurde sowohl für die Open-Source Implementierung des H265-Standards libde265 als auch die x86-Architektur optimiert. Die untere Tabelle zeigt die mittlere Abweichung nach der Optimierung mit lsqcurvefit.

	libde265	x86-Architektur
$\bar{\epsilon}$	0.3575 %	0.3098 %

Literaturverzeichnis

[1] Elisabeth Walencik *Energieverbrauch und Prozesszeit in der Videocodierung*, 22. August 2014.

[2] Christian Herglotz, Dominic Springer, und André Kaup, *Modeling the Energy consumption of HEVC P- and B-Frame Decoding.*

[3] Christian Herglotz, Dominic Springer, Marc Reichenbach, und André Kaup, *Modeling the Energy Consumption of the HEVC Decoding Process.*