Ö. D. ÖNÜR

A COMPLEXITY-UTILITY FRAMEWORK FOR OPTIMIZING QUALITY OF EXPERIENCE FOR VISUAL CONTENT IN MOBILE DEVICES

ÖZGÜR DENIZ ÖNÜR

FEBRUARY 2012

METU 2012

A COMPLEXITY-UTILITY FRAMEWORK FOR OPTIMIZING QUALITY OF EXPERIENCE FOR VISUAL CONTENT IN MOBILE DEVICES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖZGÜR DENİZ ÖNÜR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2012

Approval of the thesis:

A COMPLEXITY-UTILITY FRAMEWORK FOR OPTIMIZING QUALITY OF EXPERIENCE FOR VISUAL CONTENT IN MOBILE DEVICES

submitted by **ÖZGÜR DENİZ ÖNÜR** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. A. Aydın Alatan _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Gözde Bozdağı Akar _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. A. Aydın Alatan _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Levent Onural _____
Electrical and Electronics Engineering Dept., Bilkent University

Prof. Dr. Tolga Çiloğlu _____
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Çağatay Candan _____
Electrical and Electronics Engineering Dept., METU

Date: 08/02/2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

A COMPLEXITY-UTILITY FRAMEWORK FOR OPTIMIZING QUALITY OF EXPERIENCE FOR VISUAL CONTENT IN MOBILE DEVICES

Önür, Özgür Deniz
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. A. Aydın Alatan

February 2012, 121 pages

Subjective video quality and video decoding complexity are jointly optimized in order to determine the video encoding parameters that will result in the best Quality of Experience (QoE) for an end user watching a video clip on a mobile device. Subjective video quality is estimated by an objective criteria, video quality metric (VQM), and a method for predicting the video quality of a test sequence from the available training sequences with similar content characteristics is presented. Standardized spatial index and temporal index metrics are utilized in order to measure content similarity. A statistical approach for modeling decoding complexity on a hardware platform using content features extracted from video clips is presented. The overall decoding complexity is modeled as the sum of component complexities that are associated with the computation intensive code blocks present in state-of-the-art hybrid video decoders. The content features and decoding complexities are modeled as random parameters and their joint probability density function is predicted as Gaussian Mixture Models (GMM). These GMMs are obtained off-line using a large training set comprised of video clips. Subsequently,

decoding complexity of a new video clip is estimated by using the available GMM and the content features extracted in real time. A novel method to determine the video decoding capacity of mobile terminals by using a set of subjective decodability experiments that are performed once for each device is also proposed. Finally, the estimated video quality of a content and the decoding capacity of a device are combined in a utility-complexity framework that optimizes complexity-quality trade-off to determine video coding parameters that result in highest video quality without exceeding the hardware capabilities of a client device. The simulation results indicate that this approach is capable of predicting the user viewing satisfaction on a mobile device.

Keywords: Video Adaptation, Decoding Complexity, Video Content Characteristics, Quality of Experience.

ÖZ

MOBİL CİHAZLARDA GÖRSEL İÇERİK İÇİN TECRÜBE NİTELİĞİ ENİYİLEMESİ AMAÇLI KARMAŞIKLIK VE FAYDA TEMELLİ YAKLAŞIM

Önür , Özgür Deniz
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. A.Aydın Alatan

Şubat 2012, 121 sayfa

Mobil cihazlarda video izlenirken en yüksek tecrübe niteliği sağlayacak video kodlama parametrelerinin belirlenmesi için, öznel video kalitesi ve video çözme karmaşıklığı birlikte en iyilenmiştir. Öznel video kalitesi, nesnel bir kriter olan video kalite metriği (VQM) kullanılarak modellenmiş ve bir video'nun kalitesinin, kalite değerleri önceden ölçülmüş bir eğitim seti içinden benzer içerik özelliklerine sahip videolar kullanılarak kestirilmesini sağlayan bir yöntem sunulmuştur. İçerik benzerliğinin ölçülmesi için standartlaştırılmış uzamsal ve zamansal index metrikleri kullanılmaktadır. Belirli bir donanım için videonun çözme karmaşıklığını videolardan elde edilen içerik özellikleri kullanarak modelleyen istatistiki bir yöntem sunulmaktadır. Toplam çözme karmaşıklığı, modern video çözücülerinde bulunan ve yoğun işlem gücü gerektiren kod parçalarının karmaşıklıklarının toplamı şeklinde modellenmektedir. İçerik özellikleri ve çözme karmaşıklıkları rassal değişkenler olarak modellenmiş ve aralarındaki bileşik olasılık yoğunluk fonksiyonları Gauss Karışım Modelleri (GMM) kullanılarak elde edilmiştir. GMM ler çok sayıda videodan oluşan bir eğitim seti kullanılarak belirlenmiştir. Yeni bir videonun çözme

karmaşıklığını ölçmek için önceden hesaplanmış olan GMM ler ve videodan gerçek zamanlı olarak çıkarılan içerik özellikleri kullanılmaktadır. Ayrıca her cihaz için bir kere yapılacak video çözme deneyleri kullanılarak mobil cihazların video çözme kapasitesinin belirlenmesini sağlayan özgün bir yöntem geliştirilmiştir. Son olarak, kompleksite-kalite dengesini en iyileyerek çözme karmaşıklığı kullanılacak cihazın donanım kapasitesini aşmayacak şekilde erişilebilecek maksimum kalitede videoların elde edilmesini sağlayacak kodlama parametrelerinin belirlenmesi için fayda-karmaşıklık temelli bir yöntem önerilmektedir. Simulasyon sonuçları bu yaklaşımın kullanıcıların mobil cihazlardan video izlerken elde ettikleri tatmini kestirmek için kullanılabileceğini göstermektedir.

Anatar Kelimeler: Video Uyarlama, Video Çözücü Karmaşıklığı, Video İçerik Özellikleri, Tecrübe Kalitesi.

To my wife, my parents, and my sister

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. A. Aydın Alatan for his guidance, advice, criticism, encouragements and insight throughout the research.

I am also indepted to the members of my thesis committee, Prof. Dr. Levent Onural, Prof. Dr. Gözde Bozdağı Akar, Prof. Dr. Tolga Çiloğlu and Assist. Prof. Dr. Çağatay Candan for their support and suggestions which improved the quality of the thesis.

I would also like to thank my partners at Mobilus Ltd. who patiently waited for me to finish my studies.

Last but not the least; I would like to thank my family for loving and supporting me in all my endeavors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The processing capabilities of mobile terminals, such as Personal Digital Assistants (PDA), tablet computers and cellular phones, have increased at an unprecedented rate during the previous decade. Accompanied by the much anticipated spread of broad band wireless access, this has brought about a wealth of new possibilities for novel consumer services. Among the most exciting killer applications of this era is the pervasive access to rich multimedia content on mobile terminals.

1.1 Motivation and Problem Definition

Delivering multimedia content to terminals with diverse processing capabilities through heterogeneous networks is challenging. The problem is intensified by the fact that the end users have unique preferences and the representation of a content that is desirable for a user might be unsatisfactory for another. It is apparent that a particular representation of content would be satisfactory for a very limited number of use cases. Consequently, it is mandatory to be able to adapt the multimedia content depending on the requirements of the consumption scenario. The factors that need to be considered while determining the best representation of the content include network characteristics (maximum bandwidth or bit error rate of transmission channel), terminal characteristics (Central Processing Unit (CPU) capacity, available video codecs, color capability, display resolution), natural environment (ambient noise, illumination conditions), video content characteristics (amount of motion, amount of spatial detail) and user preferences. In [1] the research challenges outlined above are described in detail.

In most modern video distribution systems, high quality versions of video clips are stored on a media server. When a video clip is requested by a particular client, the video bit stream is modified so that it is suitable for the current consumption scenario (network conditions, client capacity etc.). Generally, the resource requirements of the high quality clip need to be decreased by using video adaptation algorithms in order to make sure that the content can be 'successfully' delivered to the client. The process of modifying a given representation of a video into another representation, in order to change the amount of resources required for transmitting, decoding and displaying video is defined as *video adaptation* [2][3].

One of the most important factors that determine the success of a video adaptation system is its ability to retain an acceptable amount of video quality, while reducing the resource requirements.

Video quality can be measured by using a plethora of tools and methods [4][5][6]. Since the ultimate consumers of video content are humans, the ultimate judge of video quality is the human subjective opinion. However, it is always difficult to find human subjects to participate in video quality tests, adhering to strict testing standards is tedious and is rarely done, and the results usually cannot be generalized for different terminals and testing environments. In practice, objective measures are commonly utilized for video quality measurement due to these difficulties involved in subjective testing methods. The validity of the objective methods is directly related to their correlation with human opinion. It is well established that conventional objective metrics fail to measure the human satisfaction accurately. However, recently developed objective metrics, such as Structural Similarity Index Metric (SSIM) [4] or Video Quality Metric (VQM) [5], have shown significant correlation with subjective data [6]. In this dissertation, VQM metric is used for modeling subjective video quality. The justification for using VQM is presented in Chapter 2.

Regardless of the method utilized to measure video quality, the subjective user satisfaction pertaining to video content is named as the *utility* of the video. The first reference to *utility* in the context of video adaptation appears in [7]. In a more theoretical approach, a conceptual framework that models adaptation, as well as resource, utility and the relationships in between, are also presented [8]. A content-based utility function predictor, in which the system extracts compressed domain features in real time and uses content-based pattern classification and regression to obtain a prediction to the utility function, is first proposed in [9].

In [10], a novel method to determine an optimal video adaptation scheme, given the properties of an end-terminal, on which the video is to be displayed, is presented. In this approach, *Utility Theory* [11] is utilized to model a strictly subjective quantity, *satisfaction*, a user will get from watching a certain video clip. In [12], the multidimensional adaptation problem is considered. The utility of video clips is determined using subjective video evaluation experiments and the results are tested using a scalable video codec (MC-3DSBC [13]). However, the processing capabilities of user terminals are not taken into consideration and this limits the usefulness of the results. In addition, most of the evaluated video content is evaluated by only five assessors, and thus, the results cannot be used to make statistical generalizations.

In [14] a system that aims to deliver multi-view video over peer-to-peer (P2P) networks is presented. The scalable video coding (SVC) extension of the H.264 standard is utilized. Each view is coded with two signal to noise ratio (SNR) scalability layers i.e. a base layer and an enhancement layer. A video adaptation decision engine is capable of adapting the bit-stream depending on the amount network resources available. If the network resources are not sufficient, the adaptation engine adjusts stream bandwidth either by selectively discarding the enhancement layer of some views, or, if the resources are even more scarce, by completely discarding some of the views. However, this approach only takes into account network characteristics and ignores device capabilities while adapting

content. In addition, the use of SVC limits its applicability since mobile devices are not able to decode SVC content and an extra step of converting SVC to baseline H.264 is required.

In [15], an end to end video adaptation architecture that enables on-the-fly content adaptation and enriched Perceived Quality of Service (PQoS) by dynamically combining different content layers, views and representations of the same video stream transmitted from multiple sources (different servers or peers for P2P) and received from multiple diverse paths and networks is presented. The MPEG-21 framework is utilized for cross-layer metadata exchange, while a Session Description Protocol (SDP) is preferred for low end terminals. The framework performs video adaptation based on network characteristics, the terminal requirements and the user preferences. However, this approach does not .take into account the terminals processing capabilities, it only considers requirements like available codecs, screen resolution etc., thus it is not capable of adapting video resource requirements according to device decoding capacity.

In addition to video quality, another important factor that determines the success of a video adaptation system is the resource requirements of the adapted video. After all, the need for adaptation arises, when the original video is too complex; i.e., it requires more resources to transmit and decode the content than what is available. If the resource requirements of the adapted video still exceed the resources available in a particular usage environment, the performed adaptation becomes useless regardless of the resulting video quality. The most resource critical component of a video delivery system is the transmission channel. With the growth of ubiquitous access to multimedia, wireless networks have become the main medium for video transmission. Wireless channels are highly error prone and their characteristics change rapidly making capacity prediction a challenging task. Consequently, a proper resource allocation is quite difficult. The challenges involved in delivering high quality media content through wireless channels is out of the scope this

dissertation. This work focuses instead on the resource requirements related to the decoding capacity of the end user terminal.

On the other hand, considering the mobility paradigm, multimedia data is increasingly accessed from mobile devices. Mobile device resources, such as battery life, CPU capacity, display size, video codec etc. are limited compared to stationary devices. Thus, it is mandatory to take into account the available device resources while adapting the content.

The computational complexity of video decoding is by far the most demanding factor on device resources. Modern decoders require significant processing power; the H.264 video decoder is twice as complex as the MPEG-4 Part 2 (Simple Profile) decoder [16].The concerns related to decoding complexity are twofold. First, if the video is too complex, the decoding capacity of the device will not be sufficient to decode the video in real time. This will result in severe artifacts like frame dropping and user satisfaction will be adversely affected. Secondly, a complex video requires significantly more CPU cycles to decode compared to a simpler video. Running CPU at higher frequencies requires higher CPU voltage, and thus, consumes more battery power [17].

There has been substantial amount of research on H.264 video encoding/decoding complexity. Unfortunately, decoding complexity is platform specific and methods that can determine decoding complexity a priori on a multitude of platforms are non-existent. In [18], a method is proposed to measure the decoding complexity of baseline H.264 streams. The decoder is separated into functional blocks and the number of times each block needs to be executed to decode a certain video clip coded at a certain bit-rate is calculated. Next, the number of basic operations (e.g. add, subtract) for the execution of each block is measured. Using processor specific information, such as number of Arithmetic Logic Units (ALU), and the types of operations each ALU is capable of performing, the clock frequency that is needed to decode the video in real time is measured. However, most mobile devices employ

specialized hardware for common video decoding operations and this significantly affects the decoding complexity on the device. Thus, the number of ALUs and the basic operations that they can perform is not sufficient to assess the video decoding performance of a mobile device.

As set forth in the above discussion, there are two critical factors that determine the success or failure of video adaptation algorithms. Resulting video quality and decoding complexity. In order to be able to provide a satisfactory end user experience, quality and complexity should be considered jointly. The optimal video adaptation scheme is the one that can produce video clips with high quality and relatively low complexity.

This thesis aims to provide the highest possible Quality of Experience (QoE) for an end user watching a video clip on a resource limited mobile device. Quality of Experience is a new paradigm that aims to determine the video quality via a user-centric approach. QoE focuses on the satisfaction of the user related to the content rather than the content quality itself [19]. In order to achieve this aim, video encoding parameters that will result in video clips having high quality and low decoding complexity need to be determined. These coding parameters are determined by jointly considering video quality, decoding complexity and device processing capability in subsequent chapters.

The next section introduces the studies in the literature that have jointly considered quality and complexity and thus have endeavored to solve the problem that is also the subject of this thesis.

1.2 Joint Optimization of Complexity and Utility

As discussed previously, in order be able to devise a useful video adaptation scheme, video decoding complexity and resulting video quality needs to be considered

jointly. The research in this area is in its infancy; nevertheless, some of the efforts in this direction have already made some progress [20]-[24]. In this section, such pioneering efforts will be discussed and their shortcomings will be presented.

In [20], a statistical framework is utilized for modeling the relationship between various content features and video decoding complexity. Decoding execution time is used as the complexity metric. The following five content features from a compressed bit-stream are used in order to model the content characteristics [20] :

1. The percentage of decoded nonzero transform coefficients,
2. The percentage of decoded non-zero motion vectors out of the maximum possible motion vectors per frame,
3. The percentage of nonzero interpolated fractional pixel positions,
4. The sum of magnitudes of the non-zero transform coefficients,
5. The sum of the run lengths of zero coefficients.

The video decoding complexity is claimed to comprise of four main components, namely *motion compensation, fractional interpolation, entropy decoding* and *inverse transformation* complexities. Each complexity is modeled in terms of one or more of the content features [20]. The relevant features for each complexity component are determined by a statistical pruning process. Each complexity component and its related content characteristics are defined as random parameters. The joint density of the random vector, whose elements are component complexity values, and the corresponding content feature random parameters are modeled using Gaussian Mixture Models (GMM). The parameters of the probability density function (*pdf*) are determined using the observations of the random vector from a training set of sequences through the expectation maximization (EM) algorithm [20]. When the decoding complexity of a new video sequence needs to be determined, the content features are extracted in real time and the most likely value for the complexity is obtained using the joint density.

Once the decoding complexity is modeled in terms of content characteristics, the optimal coding point that yields the highest video quality under complexity and distortion constraints need to be determined. In order to achieve this goal, complexity-distortion optimization is performed by a Lagrangian formulation [21]:

$$\{j^*(i), {\lambda_r}^*, {\lambda_c}^*\} \forall b_i = \underset{j(i),\lambda_r,\lambda_c}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left({D_i}^{j(i)} + \lambda_r {R_i}^{j(i)} + \lambda_c {C_i}^{j(i)} \right) \right\} \qquad (1)$$

$$\mathrm{R_{GOP}} < \mathrm{R_{max}} \text{ and } \mathrm{C_{GOP}} < \mathrm{C_{max}} \ ,$$

where *i* is the index for the adaptation unit (i.e. frame or GOP), The term *j(i)* is a particular adaptation operation, after the adaptation operation is performed the video has rate ${R_i}^{j(i)}$, distortion ${D_i}^{j(i)}$ and complexity ${C_i}^{j(i)}$, R_{max} and C_{max} are the limits on rate and complexity, respectively. $j^*(i)$ is the adaptation operation that minimized the right side of Equation 1. Note that (1) both $j^*(i)$ and $j(i)$ are not numbers, rather they are used to denote possible adaptation operations. For instance $j(i)$ could indicate dropping all *B* frames or dropping 30% of quantization parameters. On the other hand all other symbols are real numbers. An algorithm to solve the above equation is given in [21].

The main drawback of the approach in [21] is that the Gaussian models are platform (hardware) specific and the rather resource intensive task of GMM estimation process needs to be repeated for each specific hardware platform. Another drawback is that the content characteristics are not taken into account, while computing the distortion or the decoding complexity. The accuracy of video quality prediction can be increased significantly by exploiting the intuitive assumption that videos with similar content characteristics have coinciding content quality when encoded with the same coding parameters. Finally, the use of PSNR as the distortion metric is another shortcoming. As already discussed, MSE-based metrics are not sufficiently correlated with subjective human opinion. There are novel metrics that aim to solve these shortcomings are proposed in the subsequent chapters of this dissertation.

In [22], a rate-distortion complexity framework is utilized to jointly model the distortion and the power consumption of a complexity scalable encoder. The encoding procedure is claimed to be made up of three main components [22] (see Appendix-A):

1. Motion Estimation (ME) / Motion Compensation (MC)
2. Mode Selection
3. Entropy Coding

The complexity-scalable encoder is parameterized, in other words, the encoder complexity can be controlled by changing predefined encoding parameters. The complexity parameter set C is defined as $= \{C_1, C_2, \dots C_N\}$, where each complexity parameter C_i is used to control the complexity of a particular function of the encoder. Each C_i takes values in the interval [0,1]. A larger value of C_i indicates higher complexity; thus, C_i value of 0 indicates no complexity, whereas a value of 1 indicates full complexity.

Only ME/MC and, mode selection modules are parameterized in [22]. The parameters for the ME/MC module are selected as the size of the motion vector search window, motion estimation precision and the number of search points during motion estimation, whereas the parameters for mode selection are INTRA ratio parameter, number of available coding modes, etc.

The optimization problem is defined as [22]

$$min \left[\frac{P(R,C)}{D(R,C)}\right] \; s.t. P(R,C) \leq P_C \;\; , \tag{2}$$

where $P(R,C)$ is the power consumption and $D(R,C)$is the video distortion at coding bit-rate R and parameter set C. P_C is the maximum power that can be allocated to video decoding.

It is claimed that higher complexity results in lower distortion, while consuming more energy, whereas lower complexity has lower power consumption, but results in higher distortion [22]. Thus, the objective functions are in conflict with one another and for such a Multiple Objective Optimization (MOO) problem and there is no unique solution.

Assuming independent identically distributed (i.i.d.) memoryless source typical Rate Distortion (R-D) model under the MSE distortion criterion is [22] (see Appendix-B for a brief overview about Rate-Distortion Theory):

$$D(R) = \varepsilon^2 \sigma_x^2 e^{-\alpha R}, \alpha > 0 \ , \tag{3}$$

where ε^2, σ_x^2, α are real numbers. ε^2 is a source dependent parameter and it is equal to 1, if the source is uniformly distributed. The term σ_x^2 denotes signal variance, and α is also a source dependent parameter that equals to 1.386 for uniform, Gaussian and Laplacian distributions [23].

In order to account for the effect of complexity for the case when there is a single complexity parameter, a complexity multiplier is added to Eq. (3). In this case the distortion model becomes [22].

$$D(R, C) = \varepsilon^2 \sigma_x^2 c^{-\beta} e^{-\alpha R} = \gamma c^{-\beta} e^{-\alpha R} \ . \tag{4}$$

The contribution of the complexity to the *R-D* formulation is modeled as an exponential function, since it is observed that the decrease in distortion quickly saturates with increasing complexity. Furthermore, the complexity associated with each encoder component affects video distortion differently. For instance, the video distortion is less sensitive to changes in motion estimation complexity than it is to changes in DCT/Quantization complexity. Thus, each C_i has different impact on the

distortion. Parameter β accounts for this difference. A larger β indicates that a change in the corresponding complexity has more effect on the overall distortion.

For the case with multiple complexity parameters the overall distortion is defined as:

$$D(R,C) = \gamma e^{-\alpha R} \prod_{i=1}^{N} C_i^{\beta i} \quad . \tag{5}$$

In order to simplify the MOO problem, it is separated into two single objective problems [22] :

1. Minimize the distortion, while not allowing the power to be higher than a certain threshold, i.e. *min D(R,C), P(R,C)* $\leq P_c$.
2. Minimize the power consumption, while keeping the distortion under a certain threshold, i.e. *min P(R,C), D(R,C)* $\leq D^*$.

An algorithm is proposed to start coding with all C_i=1, i.e. at maximum encoder complexity. Then, the values of C_i are progressively decreased for each frame until the distortion $D(R,C)$ exceeds a predetermined threshold. As soon as the distortion exceeds the threshold, all C_i are rapidly increased. This increase in complexity decreases the distortion. Once the value of $D(R,C)$ goes below the threshold, the complexity is again gradually decreased. The complexity is also automatically increased at shot boundaries regardless of the value of $D(R,C)$. Applying this algorithm to all frames in a sequence significantly reduces the overall encoding complexity while not causing a significant increase in average distortion.

It is shown that the algorithm [22] yields power consumption gains of up to 75%. The main drawback of this approach is that it models the complexity of an encoder instead of the decoder. For our purposes, the complexity of the encoder is irrelevant, since encoding is very rarely done at the client side, especially for mobile devices. It is difficult to apply this analysis to the decoder as is, since the main complexity

components investigated in this work, i.e. motion estimation and mode selection, are not present at the decoder side Another disadvantage of [22] is that the algorithm for encoder complexity control is only applicable to the propriety encoder developed by the authors; hence, it is difficult to generalize those results to a generic codec.

In a realistic video adaptation scenario, the complexity at the decoder should be controllable by the algorithms that can be applied to already available codecs. One such approach is presented next.

In [12] and [24], the multidimensional adaptation (MDA) problem is considered in two dimensions, spatial adaptation and temporal adaptation. It is observed that metrics, such as PSNR, are not suitable for choosing the best MDA operation, since they cannot totally account for spatial or temporal resolution change. There are some approaches in the literature utilizing analytical solutions to approximate the effect of spatio-temporal resolution change on video quality [25]-[28]. However, these approaches are mostly ad-hoc and their success has been limited. A utility function (UF) is used to model the relationship between the video utility and the required resources for decoding when the video is subject to various adaptation operations. A classification based approach is followed rather than strict analytical modeling. Videos are classified into distinct content classes using extracted content features. For each video class, a separate UF is obtained [12],[24].

The MDA selection problem is formulized as [12],[24] :

$$\tilde{a} = \underset{a}{\operatorname{argmax}}\, \mathrm{U}(a),\ R(a) \leq R_{\mathrm{o}}\ , \tag{6}$$

where $U(a)$ is the utility of the adapted video, $R(a)$ is the resources required for displaying the video after the adaptation operation and $R_o$is the resource constraint of the usage environment.

Two types of adaptation operations are considered, frame dropping (FD) and coefficient dropping (CD). FD provides coarse grained adaptation, whereas CD provides fine grained adaptation. Thus FD-CD provides flexibility for adjusting spatio-temporal quality. While performing CD the coefficients that will be dropped need to be chosen so as to minimize the distortion. For this purpose, a Lagrangian optimization is performed [29].

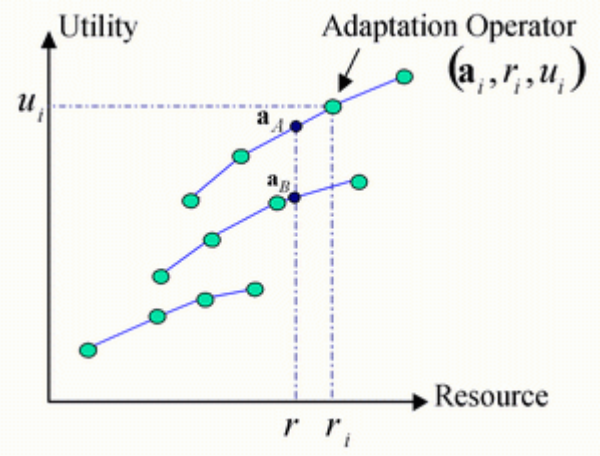

Figure 1: FD-CD adaptation [29].

Figure 1 illustrates FD-CD adaptation concept. The utility function of a particular video clip is plotted for various adaptation operations. Each connected curve represents a FD operation (no frame dropping, drop only B frames, drop all frames except I frame), while each of the blue circles indicate a CD operation (drop 10%, 20%, 30% etc. of the transform coefficients.). If all the curves are known, it is easy to determine the adaptation operation resulting in highest video utility for a given rate constraint. However, it is often not easy to obtain the utility function. Three different approaches are possible:

1. Brute Force Method: For each adaptation point transcode the video and measure the utility and the required resources. This method is obviously too complex and not suitable for any real-time implementation.

2. Analytical Modeling: Obtain approximate R-D curves by using statistical distribution models [30][31]. Very difficult to obtain realistic curves under such assumptions. A distribution that fits a certain codec might not fit another.
3. Content Based Prediction: Extract video features and map features to utility functions. This is the followed approach [24], since it provides a good balance between computational complexity and prediction accuracy.

The utility function is formulated as [24]

$$\mathrm{F}^{\mathrm{UF}} = G(\mathrm{F}^{\mathrm{CF}}) \ , \tag{7}$$

where *G* is the mapping function that maps the content features to utility functions and both F^{UF} and F^{CF} are real valued. *G* function is calculated separately for each content class using linear regression.

The drawback of the approach in [24] is that it does not take into consideration the end user terminals that the videos will be viewed on. The general concept of 'resource' is not sufficient to account for different hardware and software architectures that are utilized by mobile devices. Moreover, subjective opinion scores are used as a measure of utility. Although, subjective tests are the most reliable way to measure video quality, they need to be repeated for every video codec, every hardware architecture and each content class which is not applicable in practice.

Notwithstanding the significant amount of research in this field, an end-to-end system that provides video having satisfactory subjective quality without exceeding the resource requirements of the usage environment still fails to exist. In this dissertation, we propose a video adaptation scheme that is capable of jointly modeling video quality and decoding complexity and determining the optimal adaptation operation that results in optimal complexity-quality trade-off.

1.3 Main Contributions of the Thesis

The main aim of this dissertation is to jointly optimize video quality and decoding complexity in order to determine the video coding parameters that will provide maximum video quality while minimizing the decoding complexity.

The main contributions of this thesis can be summarized as follows:

- Joint optimization of video quality and decoding complexity, for delivering the optimal representation of video to mobile terminals, taking into account video content characteristics and device capability.

- Modeling objective video quality metrics utilizing a supervised learning algorithm on training data for which the values of the metrics are pre-computed off-line.

- Accurately modeling video decoding complexity as a function of content features extracted from the bit-stream in real time.

- Devising a method to determine the video decoding capacity of mobile terminals by exploiting a set of decodability experiments that are performed once for each device.

- Providing an end-to-end framework for quality-complexity optimization for video delivery to resource constrained mobile terminals.

1.4 Thesis Outline

This thesis is organized as follows:

Chapter 2 introduces objective and subjective video quality evaluation methods. VQM and SSIM metrics are outlined in detail. A method to model the video quality using training data for which the value of VQM metric is pre-computed is proposed. Chapter 3 introduces the concept of video decoding complexity. A method for estimating the video decoding complexity by utilizing content features extracted from the video bit-stream is described. Chapter 4 describes a novel algorithm that jointly optimizes video quality and decoding complexity. Decodability experiments that measure the decoding capacity of mobile terminals is presented. Finally, a novel algorithm that predicts the subjective quality of video clips utilizing video quality, decoding complexity and device decodability is presented. Chapter 5 presents a summary, conclusions and future research directions.

Thesis organization is illustrated by the figure below:

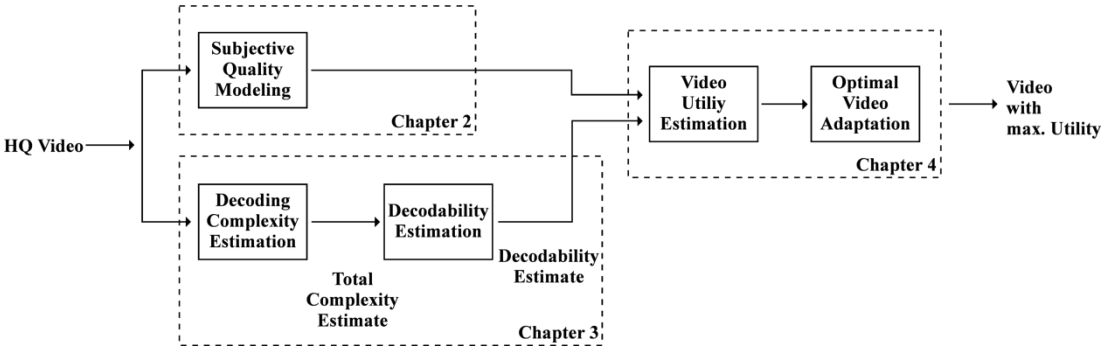

Figure 2 : Thesis Organization

CHAPTER II

VIDEO QUALITY

Digital video technology has dramatically increased the pervasiveness of video content. Video data is accessed anywhere and anytime through wired or wireless channels from a diversity of devices. This ubiquity has brought new challenges for the video coding community. The content has to pass through many stages of processing before finally reaching its destination. Video is first encoded at the source and then transmitted through a channel and then finally decoded on the end terminal. All of these stages (encoding, transmission channel, decoding) introduce distortions to the original content. In order to cope with these problems video coding algorithms are constantly evolving to increase coding efficiency, error resilience, etc. However, with the increase in numbers and complexity of video codecs, it is becoming more and more important to be able to evaluate the effectiveness of the coding algorithms and measure the quality of the produced video. Consequently, new video quality measurement metrics have been devised in recent years. These metrics can be used to dynamically monitor and adjust delivered video quality in order to increase the quality of service provided to the end user. Video quality metrics also help to further improve and optimize the coding algorithms.

There are two main types of video quality assessment metrics, objective and subjective. These two types of metrics and the associated video quality measurement methods are discussed in the subsequent two sections.

2.1 Measuring Video Quality: Subjective vs. Objective Methods

Consumers are the ultimate receivers of video content and thus, they should be the arbiter of video quality. This implies that the measurement of video quality should involve selecting a sufficiently large group of humans and eliciting their opinion on the content quality. Video quality evaluation depending on human judgment is called as *subjective quality assessment*. Typically, the evaluators use a predefined grading scale to gauge the perceived quality of video clips. The average grade over all human subjects then constitutes the subjective quality of the video sequence.

Quality is a very subjective concept and opinions of human evaluators might be affected from the test environment, mood of evaluator, time of day, fatigue, etc. Thus, subjective video evaluation experiments are deemed valid, only when performed in accordance with strict standards [32][33]. Subjective quality assessment is time consuming and expensive as it is very hard to enlist a large group of human subjects to participate in the tests. Furthermore, the results of the tests performed in different laboratories do not always agree with each other and thus, it is very difficult to determine an absolute quality score for a particular video clip. These complications have led to the design of numerical algorithms that seek to predict human subjective scores. Such algorithmic assessment of quality is referred to as *objective quality assessment* [6]. Subjective and objective video quality evaluation methods are described in detail in the following sections.

2.2 Subjective Video Quality

As discussed previously, measuring subjective video quality is time consuming and expensive. Nevertheless, subjective video quality assessment is quite important, since it provides a ground truth against which the performance of objective algorithms can be judged.

Results of subjective tests are very sensitive to test methodology and the testing environment. The International Telecommunication Union (ITU) has standardized the testing procedures, so that the results across different laboratories can be aggregated and used jointly to evaluate objective algorithms. Commonly used ITU testing procedures are described in detail in the next section.

2.2.1 Subjective Video Quality Testing Methods

Different test methods exist for determining the subjective video quality. Methods that are most commonly used [32][33] are briefly described below:

Double Stimulus Continuous Quality Scale (DSCQS): For each sequence, the reference picture and the test picture are presented to the assessor in a random order (i.e. the assessor does not know which one is the reference and which one is the test). The assessor is asked to rate both pictures using to a continuous grading scale. Usually grading is done by putting a mark on a straight line, where one end of the line denotes the highest quality and the other end the lowest quality.

Double Stimulus Impairment Scale (DSIS): For each test sequence first a reference picture is presented to the assessor and the assessor is explicitly notified that it is the reference. Then the test picture is presented and the assessor is asked to grade the impairment in the test picture compared to the reference. Grading is done on a discrete impairment scale with 5 or 7 grading levels.

Single Stimulus (SS): The assessors are only presented a single video and are asked to grade the video on a five point grading scale.

Double Stimulus Binary Vote (DSBV): Very similar to DSIS but the assessors are only asked to decide whether the test sequence contains a discernible impairment or not.

Regardless of the testing methodology used, the test sessions should be made up of three phases [4].

Training Phase:

- During the training phase written instructions (so that exactly the same set of instructions can be given to each assessor) should be provided to the assessors that describe the testing methods and the grading scales used.
- The training phase should also include 2-3 sequences that will get the assessor acquainted with the timing and the amount of quality variation between test videos that are likely to be encountered during the test. The samples used for the training session should have similar levels of impairments to the actual sequences that will be used in the test but they should not be the same video sequence.

Stabilization Phase:

- The first five sequences of each session should be used for stabilization. These sequences should contain some of the best and some of the worst quality videos so that the entire impairment range is presented to the assessor.
- The grades given to these five sequences should not be taken into account and these sequences should later be presented again in the test.
- The assessors should not know that they are in stabilization phase.

Testing Phase:

- If any reference sequences are used, they should be in ITU-R 601 format (uncompressed 4:2:0 YUV).The sequences used in the testing phase should be about 10 seconds long.

- The assessors should be given a limited amount of time to complete the grading. Usually 10 seconds of grading time is ideal.
- In general, at least 15 assessors participating in the test. However 4-8 assessors are sufficient to provide indicative results.

Comparison of DSIS and DSCQS methods

DSIS and DSCQS methods are commonly utilized for video quality evaluation on mobile devices. DSCQS by its nature gives relative results as the assessors do not know beforehand which sequence is reference and which sequence is a test sequence. Therefore, DSCQS is usually preferred, when the quality of the reference and the quality of the test sequences are similar [4], whereas the DSIS is usually used, when the reference picture has clearly a higher quality compared to the test sequence.

In our previous work [10][34], a method for determining the subjective satisfaction of users pertaining to watching a video clip on a mobile device has been proposed. User satisfaction is modeled in terms of content characteristics (amount of motion and spatial detail) and video coding parameters (bit-rate, frame rate and spatial resolution). User satisfaction models are obtained utilizing subjective video evaluation experiments performed according to ITU recommendation ITU-R BT.500-11[32]. *Double Stimulus Impairment Scale* (DSIS) method described in the recommendation is employed in order to determine the user satisfaction on watching a particular video clip (encoded with predetermined values of bit-rate, frame rate and resolution) on a particular mobile device. In accordance with the DSIS method, evaluators are presented with an unimpaired reference video together with an impaired version which is processed using the system under test. The reference videos are presented in YUV 4:2:0 format on a desktop computer. The impaired videos are coded with the reference implementation of the H.264 codec using a set of frame rates and QP values and are viewed on the mobile device that is being tested. The users are asked to evaluate the difference between the original reference and the videos viewed in the mobile device. A discrete grading scale of 1 to 5 is used. A

grade of 5 indicates that the video being tested has no visible difference from the reference video, whereas a grade of 1 indicates that the video is severely impaired.
Since video content plays a major role in modeling the user satisfaction, different utility models are constructed for videos having different content characteristics. In [49][34], video clips are classified into 4 distinct content classes according to their level of spatial detail and motion activity and a unique satisfaction model is obtained for each content class. In order to determine the subjective quality of a previously unknown video clip, it is first assigned to one of the content classes depending on the values of its SI and TI metrics. Once the correct model is determined, the subjective quality of the clip can be determined for an arbitrary set of coding parameters.

As discussed in Chapter 1, although subjective quality evaluation provides accurate results, the difficulties associated with subjective testing outweighs its advantages and usually objective video quality measurement is preferred. Next section describes objective quality metrics in detail.

2.3 Objective Video Quality

Significant amount of research has been done on objective video quality evaluation methods and numerous objective metrics have been developed [4]-[6]. Objective quality metrics are classified according to their requirements on the availability of the original image for computing the distortion of the processed image. The metrics that require the complete original image to be available are called *full reference*, whereas those that require only features extracted from the original image are called as *reduced reference* metrics. Finally, the metrics that do not require any information other than the undistorted image are called *no reference* or *blind* metrics.

Mean Square Error (MSE) and its related metric Peak Signal to Noise Ratio (PSNR) are the most commonly used full reference metrics. The reason for their wide adoption is due to their computational simplicity, their clear physical interpretation

and their mathematical convenience for optimization purposes. However, it is known that MSE and PSNR do not correlate well with subjective human scores [35]. Many other objective quality assessment methods that aim to mimic the subjective quality more accurately exist. A means of testing the effectiveness and accuracy of objective quality metrics in predicting human subjective opinion is necessary in order to further improve these algorithms. For this purpose, the results of objective quality assessment tests are commonly compared to available subjective test results.

Video Quality Experts Group (VQEG) is one of the organizations that undertake projects aimed at developing and continuously improving objective video quality metrics. A typical VQEG project contains two parallel evaluations of test video material, one by human observers and the other one by objective quality metrics, which are meant to predict the subjective scores [36]. The results of some VCEQ projects are discussed next.

In VQEG Full Reference Television (FRTV) Phase I [37] tests performed in year 2000; 10 leading video quality assessment algorithms were compared and their correlation with human subjective scores were studied. It was observed that all the metrics were statistically indistinguishable from PSNR. However, the FRTV Phase I database is dated and the tests were done for TV, and thus, contain interlaced videos which affect the performance of the algorithms. Phase I tests included distortions from earlier codecs, such as H.263 and MPEG-2 that exhibit different error patterns compared to H.264 [5].

The VQEG FRTV Phase II [38] tests, which were performed in 2003, demonstrate significantly different results. VQEG FRTV Phase II tests contained two experiments, one restricted to 525-line video and the other restricted to 625-line video. The subjective testing was performed by three independent tests. VQM algorithm exhibited very strong correlation with subjective scores. It achieved a Pearson correlation coefficient of 0.938 for 525-line tests (VQM was the only objective metric to achieve a Pearson coefficient higher than 0.9) and 0.886 for 625-

line tests. VQM was among the four methods that performed statistically better than the others for the 625-line tests. VQEG experiments, as well as other efforts in the literature [5], indicate that two classes of algorithms outperform other objective quality metrics in terms of their correlation with subjective scores. These are the Structural Similarity Index Metric (SSIM) [3] and the VQM [4].
The SSIM and VQM algorithms are described in more detail in the following sections.

2.3.1 Structural Similarity (SSIM) Index

Under the assumption that the human visual system (HVS) is highly adapted to extract structural information from the viewing field in an image, SSIM metric uses the change in structural information as a measure of perceived distortion, instead of using the amount of error between the reference signal and the distorted image [4]. As is apparent from the definition, the SSIM index is designed for measuring the similarity of still images rather than video sequences.

The SSIM Index is composed of three components : *luminance*, *contrast* and *structure* [3]. In order to measure the objective quality of an image signal *x*, it is compared to another image signal, *y***,** which is assumed to have perfect quality. Since signal, *y*, has perfect quality, the similarity measure between *x* and *y* will determine the objective quality of *x*. The general form of the overall similarity measure is given as [3]:

$$S(x,y) = f(l(x,y), c(x,y), s(x,y)) \ , \tag{8}$$

where $l(x,y), c(x,y), s(x,y)$ stand for the luminance, contrast and structure comparison functions, respectively, each measuring the similarity between the two images for the corresponding features.

The luminance of the signals *x* is estimated as the mean intensity given by

$$\mu_x = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} x_{\mathrm{i}} \ , \tag{9}$$

where x_i is the intensity (i.e. *Y*-component only) of the pixel *i* in image *x* and *N* is the total number of pixels. The luminance comparison function $l(x, y)$ is thus a function of $\mu_x$and μ_y.

The signal contrast is estimated by using the standard deviation given as

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum_{\mathrm{i}=1}^{N}(x_{\mathrm{i}} - \mu_x)^2} \ . \tag{10}$$

The contrast function $c(x, y)$is thus a function of σ_x, σ_y. Finally, the structure comparison function is obtained as a function of $(x - \mu_x)/\sigma_x$ and $(y - \mu_y)/\sigma_y$. The resulting closed form expression for the comparison functions in (8) are obtained [3] as below:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{{\mu_x}^2 + {\mu_y}^2 + C_1} \ , \tag{11}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{{\sigma_x}^2 + {\sigma_y}^2 + C_2} \ , \tag{12}$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \ , \tag{13}$$

where C_1,C_2, and C_3 are constants and σ_{xy} is the covariance between *x* and *y*. The details related to the derivations for the constants can be examined in [3]. Finally, the SSIM index is defined as

$$\mathrm{SSIM(x, y)} = \mathrm{l(x, y)\ c(x, y)\ s(x, y)} \ . \tag{14}$$

Following the success of the SSIM metric given as described in (14), many variants have been proposed. Below, is a short summary of its variants [4]:

Frame-SS-SSIM: The single-scale structural similarity index. Identical to the original index defined in (14). Its value is calculated for each frame, and then, averaged over all frames.

Frame-MS-SSIM: The multi-scale SSIM index corrects the viewing distance dependence of SS-SSIM and accounts for the multi scale nature of both natural images and HVS. MS-SSIM yields results that are more correlated with subjective scores than the SS-SSIM. Its value is also calculated for each frame and then averaged over all frames.

Speed-Weighted SSIM (SW-SSIM): Incorporates a temporal weighting scheme. Assigns a weight to each frame taking into account the amount of motion in the frame. The SS-SSIM values are then averaged across all frames using the assigned weights.

P-SS-SSIM: Objective algorithms treat different parts of an image equally and obtain the overall quality score by averaging the quality from different regions. However, humans attach more significance to regions with poor quality then to regions with high quality. Even when there is a small number of poor quality regions in image, humans perceive the image as low quality. P-SS-SSIM assigns greater weights to regions with poor quality, and thus, correlates better with human perception. This metric is also calculated on a frame by frame basis.

In [4], the VQM metric is compared against the group of SSIM metrics described above. These metrics are tested initially under varying distortion types and then under different compression rates. The Spearman Rank ordered correlation coefficients (SROCC) between the metrics and the human subjective scores are utilized in order to judge the success of the algorithms. SW-SSIM, VQM and MS-

SSIM perform better than the others across different distortion types and VQM, MS-SSIM, P-SSIM perform better across compression rates. Overall VQM metric performs better than all the SSIM metrics. Figure 3 summarizes the results. As can be observed from the figure VQM metric outperforms all the other metrics, however it is also the most computationally complex one. In the remainder of this thesis the VQM metric is used to model perceived subjective quality. The details of the VQM metric are presented in the next section.

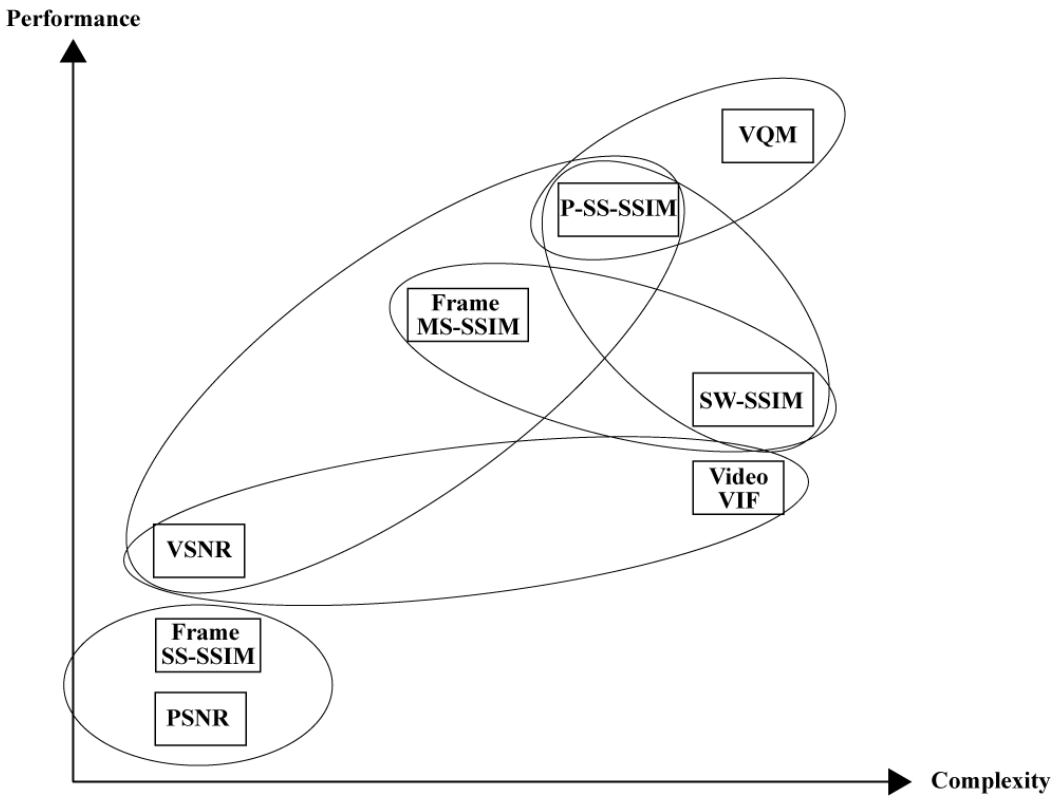

Figure 3 : Performance vs Complexity of objective Video Quality Assesment algorithms [5]

2.3.2 Video Quality Metric (VQM)

The VQM metric has been developed by the National Telecommunication and Information Administration (NTIA) [39]. VQM was the only video quality estimator that was in the top performing group for both the 525-line and 625-line tests carried

out as a part of VQEG FRTV Phase II [38] experiments. Subsequently, VQM has been standardized by the American National Standards Institute (ANSI) in the updated version of T1.801.03 – 2003, "Digital transport of one-way video signals – Parameters for objective performance assessment" [40]. VQM has also been included in two ITU standards [41][42].

VQM computes the video quality in four stages [5]:

- *Calibration:* The distorted video is aligned with the reference video. Spatial alignment, valid region extraction, gain and level offset calibration and temporal alignment are performed.
- *Quality Feature Extraction:* A quality feature is defined as a quantity of information associated with, or extracted from a spatiotemporal (S-T) sub region of a video stream. Conceptually, all the features are extracted using the same procedure. A perceptual filter is applied to the video stream in order to enhance some property of video quality (i.e. edge strength). Then, features are extracted from S-T sub regions utilizing a mathematical function, such as standard deviation. Finally, a perceptibility threshold is applied to the extracted features.
- *Quality Parameter Calculation:* While quality features quantify some perceptual aspect of a video stream, quality parameters compare original and processed features to obtain a measure of video distortion. First, the processed feature value of each S-T region is compared to the corresponding original feature using comparison functions that emulate the perception of impairments. Next, perception based error pooling functions are applied across space and time.
- *VQM Calculation:* VQM can be calculated using various models; i.e. General Model, Low Bandwidth Model, Developer's Model, Television Model, Video Conferencing Model, Fast Low bandwidth Model, PSNR Model. During VQEG FRTV phase II tests discussed above, the performance of the General

Model was tested. The General Model is also the one standardized by ANSI and ITU [40][41][42]. All of the models use a linear combination of an identical set of quality features.

The VQM general model uses seven independent parameters. Four of these parameters are based on features extracted from spatial gradients of the luminance component, whereas two of them are based on features extracted from the chrominance components and one is based on the product of features that measure contrast and motion extracted from the luminance component. The quality parameters utilized by the General Model are briefly described below [5]:

si_loss: Detects a decrease or loss of spatial information (i.e. blurring). Always takes a negative value.

hv_loss: Detects a shift of edges from horizontal and vertical orientation to diagonal orientation. This might be the case, if the horizontal and vertical edges suffer more blurring than the diagonal edges.

hv_gain: Detects a shift of edges from diagonal to horizontal. This might be the case, if the distorted video contains tiling or blocking artifacts.

chroma_spread: Detects changes in the spread of the distribution of two dimensional color samples.

si_gain: Measures improvements to quality that result from edge sharpening or enhancements. Note that si_gain is the only quality improvement parameter in the model. In other words, higher value of si_gain indicates an higher quality sequence.

ct_ati_gain: Product of a contrast feature, measuring the amount of spatial detail, and a temporal information feature, measuring the amount of motion present in the S-T

region. Impairments will be more visible in S-T regions that have a low product than in S-T regions that have a high product.

chroma_extreme: Detects severe localized color impairments, such as those produced by digital transmission errors**.**

The VQM value for the General Model is calculated as a linear combination of the above quality parameters as specified below.

$$\begin{aligned} VQM \;=\; & -0.2097\, si_{loss} + 0.5969\, hv\_loss + 0.2483\, hv\_gain \\ & + 0.0192\, chroma\_spread - 2.3416\, si\_gain \\ & + 0.0431\, ct\_ati\_gain + 0.0076\, chroma\_extreme \end{aligned} \qquad (15)$$

Figure 4 illustrates the correlation between subjective scores and VQM general model. The subjective tests were performed between 1992 and 1999 in accordance with ITU standards [23][24]. A total of 1536 sequences were used. The Pearson linear correlation coefficient between subjective scores and the VQM values was 0.948. The linearity of the results in Figure 4 indicates that VQM results are highly correlated with VQM scores. In the next section, VQM General Model is used to model video quality.

2.4 Modeling Video Quality

As discussed previously, utilizing objective metrics for video quality prediction is much more practical than subjective quality evaluation. Nevertheless, objective metrics are also computationally complex and it is not feasible to compute their value for each video whose quality needs to be determined. Furthermore, the values of the objective metrics will be different for the same video scene encoded with different parameters (bit-rate, frame rate, spatial resolution), and thus, the metrics will need to be recomputed, whenever there is a change in video coding parameters.

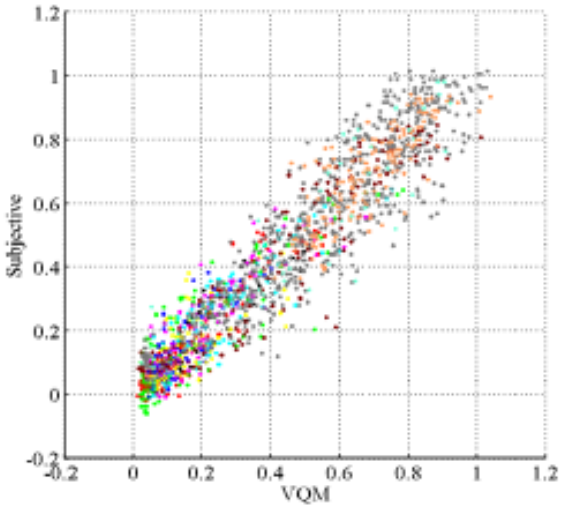

Figure 4 : Subjective Quality vs VQM [5]

Consequently, it is necessary to decrease the computational complexity associated with objective quality assessment. One possible approach is fitting analytical models to objective quality data and using these models for subsequent quality estimation. The main problem with such an approximation is that generally simpler models lead to less accurate quality predictions. An alternative method, based on the idea that sequences having similar content characteristics will have similar objective quality when coded with the same encoding parameters, is presented next.

2.4.1 Video Content Characteristics and Video Quality

Video content characteristics play a crucial role on quality. For instance, it is apparent that jerky motion would not create the same level of perceptual impairment on a simple head and shoulder scene, as the level of impairment it would create on a fast motion sports clip. Thus, it is very important to be able to determine the content characteristics of a video clip before modeling its quality.

In this dissertation, it is proposed that sequences sharing similar content characteristics will have similar objective quality, when encoded with identical encoding parameters (i.e. bit-rate, frame rate and resolution). A training set of videos whose objective quality is measured in advance can be used to predict the objective quality of a new video whose content characteristics are similar to one or more videos in the training set. Utilizing a large enough training set, it should be possible to determine the objective quality of a video clip quite accurately, as will be demonstrated in this section.

ITU Spatial Perceptual Information (SI) and Temporal Perceptual Information (TI) metrics [33] will be utilized in order to measure content similarity. It is argued that sequences with similar SI and TI values should have similar objective quality. The details related to the SI and TI metrics are described next.

SI metric is based on the Sobel filter. Initially, each video frame is filtered with the Sobel operator. Then, the standard deviation over pixels is computed for each filtered frame. Thus, a set of standard deviation values are obtained. The maximum value in the set is the SI value for the investigated sequence. The extraction of the SI metric can be formulized as below.

$$SI = \max_{time}\{std_{space}[Sobel(F_n)]\} \ , \tag{16}$$

where F_n is the frame at time n, Sobel is the Sobel operator, std_{space} is the standard deviation of the pixel intensities of frame F_n and $\max_{time}$ depicts taking the maximum value in time.

On the other hand, TI metric is based on the difference between pixel values at the same location in space at successive times or frames. This difference is called as the *motion difference feature* and defined as

$$M_n(i,j) = F_n(i,j) - F_{n-1}(i,j) \ , \tag{17}$$

where F_n is the frame at time *n*, *(i,j)* denote the spatial coordinates of the pixel within the image and M_n is the motion difference feature. TI is computed as the maximum in time of the standard deviation over space of M_n *(i,j)* over all *i,j* given as:

$$TI = \max_{time}\{std_{space}[M_n(i,j)]\} \ . \tag{18}$$

The SI and TI values of 10 sequences (*Akiyo*, *Bus*, *Coast*, *Flower*, *Foreman*, *Mobile*, *Mother*, *Soccer*, *Waterfall*) are plotted in Figure 4. The sequences are all in YUV 4:2:0 format with common intermediate format (CIF) resolution. The sequences are divided into 16 frame long video chunks and each data point (i.e. dot) in Figure 4 corresponds to a particular 16 frame long video chunk. Bus sequence has 150 frames, waterfall sequence has 260 frames, flower sequence has 250 frames, and all the other sequences have 300 frames.

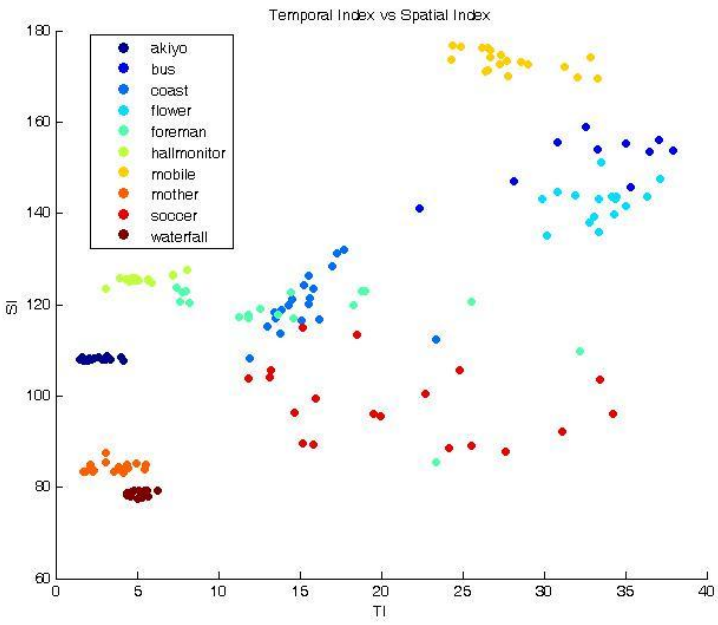

Figure 5 : SI and TI values for sequences *Akiyo, Bus, Coast, Flower, Foreman, Mobile, Mother, Soccer* and *Waterfall*

Figure 5 indicates a strong self-clustering of sequences, i.e. the chunks from a particular sequence show a strong tendency to fall into the same neighborhood in the graph. Only the *Foreman* and *Soccer* sequences do not exhibit this behavior. This is expected, since both *Foreman* and *Soccer* sequence content characteristics change dramatically within the video clip.

A method for predicting the objective video quality of a previously unknown sequence utilizing training sequences with similar content characteristics is presented in the next section.

2.4.2 Predicting Objective Video Quality Using Training Data

As discussed previously, it is not feasible to compute the value of VQM metric for each sequence whose quality needs to be determined. In order to overcome this problem, a method for predicting the VQM value of a previously unknown sequence using a set of training sequences whose VQM values have already been determined is proposed. In this manner, a prediction on human subjective quality assessment could be achieved for any content encoded with arbitrary parameters and such an evaluation could be utilized during quality-complexity optimization framework.

In order to able to make a prediction for VQM, a training set is required and the training set consisting of the 10 sequences described in Section 2.4.1 is utilized. The videos are divided into small chunks, each 16 frames long and coded with three different quantization parameter (QP) [43] values (15, 25, 35) and three different frame rates (15 fps, 20 fps, 30 fps). There are a total of 175 chunks, and each one is coded with nine different coding parameters (three QP values x three frame rate values); thus, there are a total of 1575 video clips in the training set. The VQM value for each clip is calculated using the BVQM tool which is available online [44].

Before proceeding with the details of the VQM prediction method, there is an important point that should be discussed. VQM metric is calculated using raw YUV

sequences. This approach makes VQM 'blind' for the encoding frame rate, since, as long as the camera frame capture rate is constant, the YUV data will not change with changing encoding frame rate. In other words, as long as VQM is concerned, the first *N* frames of a particular sequence encoded at 15 fps are indistinguishable from the first *N* frames of the same sequence encoded at 30 fps. The sequences utilized in the training set were originally captured at 30 fps. Thus, in order to measure the VQM value for frame rates other than 30 fps, the frame rate agnostic YUV data should be altered somehow in order to simulate different frame rates. A frame repeating procedure is proposed in order to measure VQM values for 15 fps and 20 fps sequences against the original rate of 30 fps. By repeating YUV frames as appropriate, an effect similar to what would have been observed by a subject, if the frames were originally captured at the reduced frame rates, is obtained. For instance, since the camera has captured the original YUV sequence at 30 fps, to simulate a 15 fps YUV sequence, every frame was repeated once, whereas to simulate a 20 fps YUV sequence every other frame was repeated. Table 1 illustrates this concept.

Table 1 : Manually Changing Frame Rate - Frame Orderings For Different Frame Rates

Frame Rate	Frame Numbers of First 16 Frames
30 FPS	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16
20 FPS	1-2-2-4-5-5-7-8-8-10-11-11-13-14-14-16
15 FPS	1-1-3-3-5-5-7-7-9-9-11-11-13-13-15-15

It is proposed that such a frame repeating procedure also simulates the user experience during watching a video clip at a reduced frame rate or watching a video clip having frame dropping distortions. This result is due to the fact that when the video is played back at a reduced rate or there are dropped frames, the frames that are already decoded remain displayed on the device screen longer (until new frames are available to replace them) than they would remain for full frame rate videos. Hence, such a visual perception results in the same impairment as the impairment produced by repeating frames in the YUV sequence as described in Table 1. Consequently, the VQM value obtained by repeating frames is expected to properly mimic the

perceptual distortion observed in the video stream with dropped frames or reduced frame rate.

Returning to the discussion of the VQM prediction algorithm, the VQM values for all 1575 video clips are measured by using the BVQM tool. The SI and TI values of the 175 chunks are calculated using the definitions in (9)-(11). The SI and TI values for video clips belonging to the same video chunk are assumed to be the same regardless of the encoding frame rate and QP. When the VQM value of a new sequence needs to be determined, the Euclidian distance between its SI and TI values and the SI and TI values of all the video chunks in the training set are calculated. The video chunk with the smallest SI and TI distance is determined, to obtain a nearest neighbor classifier. As discussed previously, all video chunks have 9 representations (each coded with combinations of 3 frame rates and 3 QP values) in the training set. Among the video clips with the smallest SI and TI distance, the VQM value of the instance coded at the same frame rate and QP as the new video clip is taken as the VQM estimate.

In order to evaluate the performance of the VQM prediction algorithm, the prediction error for each distinct sequence (i.e. *Akiyo, Bus, Coast, Flower, Foreman, Mobile, Mother, Soccer* and *Waterfall*) in the training set is calculated. For this purpose, all the video clips belonging to each one of the above sequences is removed from the training set sequentially, then the VQM values for the removed clips are estimated using the algorithm presented above.

The prediction error is measured by using the relative percentage error (RPE) metric given below:

$$\text{RPE} = \frac{\sum_{i=1}^{N} |VQM_i - \widetilde{VQM}_i|}{\sum_{i=1}^{N} VQM_i} \text{x}100 \ , \qquad (19)$$

where the N is the total number of video clips whose VQM is to be estimated, $VQM_i$is the actual value of the VQM data and $\widetilde{VQM}_i$is the predicted VQM value.

The VQM prediction errors for each sequence and the average error over all sequences are given in Table 2.

Table 2 : VQM Predition Errors

Sequence	Prediction Error (%)
Akiyo	8,77
Bus	15,67
Coast	14,65
Flower	40,27
Foreman	16,33
Hallmonitor	26,87
Mobile	43,18
Mother	11,16
Soccer	13,90
Waterfall	14,08
Average	**20,48**

It is expected that the prediction error should either remain the same or decrease further by utilization of larger training sets. To demonstrate this argument, 7 more unique sequences are added to the training set. The added sequences are *Bridge-close, Container, Football, Harbour, Highway, Silent* and Tempete. Moreover, a 600 frame version of the soccer sequence is used instead of the 300 frame version. Thus, the number of video chunks has been increased to 359 and the total number of video clips now equals 3231. The SI and TI plot of this new training set is given in Figure 6. The resulting prediction errors for this new extended training set are given in Table 4 presents the value of the correlation coefficients between VQM and PSNR. As it can be observed from the results in Table 4, the VQM and PSNR values have significant correlation. However, the correlation is not strong enough to imply a completely linear relationship. Notice that the results are in Table 4 negative; this result is due to the fact that the higher quality levels are indicated by lower VQM values, whereas they are indicated by higher PSNR values and vice versa. In Chapter 4, both VQM and PSNR will be utilized in subjective video quality, i.e. video utility, prediction and their prediction performance will be compared performance of PSNR.

Table 3. The results in Table 4 presents the value of the correlation coefficients between VQM and PSNR. As it can be observed from the results in Table 4, the VQM and PSNR values have significant correlation. However, the correlation is not strong enough to imply a completely linear relationship. Notice that the results are in Table 4 negative; this result is due to the fact that the higher quality levels are indicated by lower VQM values, whereas they are indicated by higher PSNR values and vice versa. In Chapter 4, both VQM and PSNR will be utilized in subjective video quality, i.e. video utility, prediction and their prediction performance will be compared performance of PSNR.

Table 3 illustrate that the prediction error tends to decrease as the training set is enlarged. Another interesting point to note is that the prediction errors related to some sequences do not change at all, while others decrease significantly. Juxtaposing the SI and TI plots in Figure 5 and Figure 6, it can be observed that for the sequences which are similar in the SI and TI distance sense to the 7 sequences that were added, the prediction error decreases. For sequences that are not similar to the 7 sequences the prediction error does not change. This observation is precisely as expected, since as the number of SI and TI points in a certain region of the training set increases, better training sequence matches with similar content characteristics and VQM values can be obtained for the test sequences with SI and TI values in that region.

The results presented in Tables 2 and 3 are sufficiently accurate for our purposes. Thus, the approach presented in this chapter will be used to predict the VQM values in the subsequent chapters.

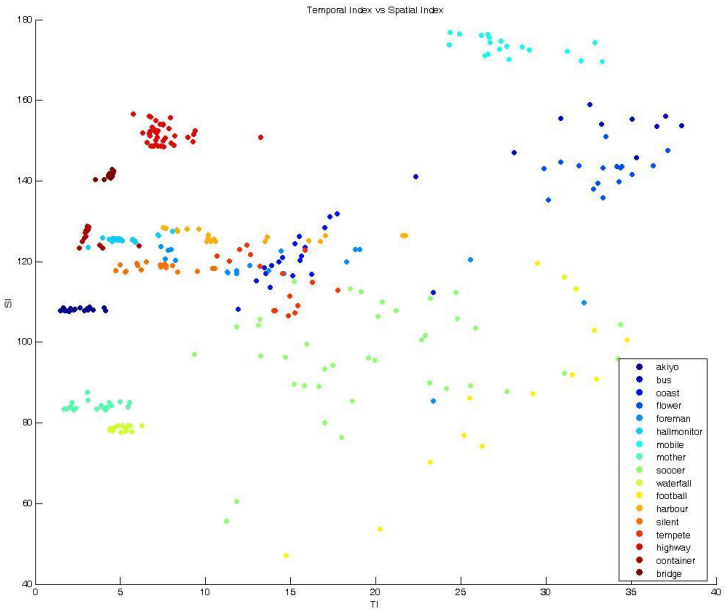

Figure 6 : SI andTI values for the Extended Training Set

Before proceeding with decoding complexity estimation in the next chapter, it is informative to see the correlation between the VQM values and the PSNR for the videos in the training set. A correlation coefficient of ± 1 indicates a perfectly linear relationship whereas a coefficient of 0 indicates that the increase or decrease of one of the values does not imply a corresponding increase or decrease for the other value.

Table 4 presents the value of the correlation coefficients between VQM and PSNR. As it can be observed from the results in Table 4, the VQM and PSNR values have significant correlation. However, the correlation is not strong enough to imply a completely linear relationship. Notice that the results are in Table 4 negative; this result is due to the fact that the higher quality levels are indicated by lower VQM values, whereas they are indicated by higher PSNR values and vice versa. In Chapter 4, both VQM and PSNR will be utilized in subjective video quality, i.e. video utility, prediction and their prediction performance will be compared performance of PSNR.

Table 3 : Prediction Errors for 17 Sequence Training Set

Sequence	Prediction Error (%)
Akiyo	9,70
Bus	15,67
Coast	14,31
Flower	40,27
Foreman	22,71
Hallmonitor	13,85
Mobile	43,18
Mother	11,16
Soccer	12,42
Waterfall	14,08
Average	**19,74**

Table 4 : Correlation Coefficient Between VQM and PSNR Values of the Training Data

Sequence	PSNR VQM Corr.Coeff.
Akiyo	-0.99
Bus	-0.83
Coast	-0.93
Flower	-0.93
Foreman	-0.86
Hallmonitor	-0.97
Mobile	-0.94
Mother	-0.97
Soccer	-0.81
Waterfall	-0.94
Average	-0.92

2.5 Summary and Discussions

In this chapter, various methods of video quality measurement are introduced. Specifically subjective quality measurement and objective quality measurement methods are presented. It is argued that using subjective quality metrics is not practical, as performing subjective experiments are difficult and expensive. The

VQM metric, an objective metric which has high correlation with subjective scores is utilized. It is further argued that even the VQM metric is too complex to be used in a real time video delivery scenario.

Thus, a method for predicting the objective video quality of a previously unknown sequence utilizing training sequences with similar content characteristics is presented. It is proposed sequences sharing similar content characteristics will have similar objective quality when encoded with identical encoding parameters (i.e. bit-rate, frame rate and resolution). A training set of videos whose objective quality is measured in advance can be used to predict the objective quality of a new video whose content characteristics are similar to one or more videos in the training set. ITUs Spatial Perceptual Information (SI) and Temporal Perceptual Information (TI) metrics are utilized in order to measure content similarity. It is proposed that sequences with similar SI and TI values will have similar objective quality.

Utilizing a training set of video clips, the prediction performance of the proposed algorithm is tested. For a training set of 10 distinct sequences, a prediction error value of 20,48% is obtained, whereas for a training set of 17 video clips, this error decreases to 19,74.

Finally, the correlation between PSNR and VQM values are presented. It is observed that even though the two values have significant correlation, they do not exhibit a strict linear relationship.

CHAPTER III

VIDEO DECODING COMPLEXITY

The consumption of rich multimedia data on mobile devices is becoming commonplace and it is expected that mobile platforms will be the primary means of access to multimedia in the coming years. Delivering multimedia data to mobile terminals involves devices with diverse processing capabilities, heterogeneous data networks, and different user preferences. This makes determining the correct representation of video that will provide a satisfactory multimedia experience very difficult. One of the key points for providing the best quality of experience (QoE) to end users is to make sure that the resources required to decode the provided video do not exceed their devices capabilities. In order to accomplish this, the computational complexity of decoding the video stream should be calculated and compared to the device processing capacity. In case the complexity is found to exceed the device capacity, the video should be transcoded to a simpler format prior to sending. The aim of this chapter is to model the video decoding complexity in terms of video content features that can be extracted from the content.

H.264 (MPEG-4 Part 10 - AVC) is the most advanced video codec that is available for commercial applications. The H.264 standard is described in detail in Appendix-A. In this chapter, the approach for determining decoding complexity is demonstrated by using the H.264 codec. Nevertheless, the proposed method can easily be applied to all modern hybrid codecs.

3.1 Decoding Complexity in Hybrid Video Decoders

Conventional video codecs, such as MPEG-1/2/4, are all made up of individual blocks of code each designed to exploit a particular aspect of the temporal or spatial redundancy in video content in order to decrease the size of the resulting bitstream. Main blocks of code typically found in modern codecs are *motion compensation, inverse transform, entropy decoding* and *deblocking filter* blocks. The amount of complexity that is associated with each block depends on video content characteristics. The total complexity of decoding a video sequence can be expressed as the aggregation of complexities resulting from each of these code blocks.

The complexity of decoding a video clip on a particular hardware platform can be denoted as the *decoding complexity* (DC). In this context, DC represents the total time or the total number of processor cycles spent by a particular decoder in order to fully decode a video stream (For a CPU with known frequency, the total execution time can be calculated from the total number of processor cycles or vice versa).

The decoding complexity of a particular video clip on a particular hardware platform can be precisely measured by using resource monitoring tools, such as *Vtune* by Intel [45] and *Real View Development Studio* (RVDS) [46] by ARM. These software tools measure the number of processor cycles that is required to execute any piece of code, and thus, they can be used to accurately determine the decoding complexity, as well as to pinpoint parts of code that require the most processing power.

Below listing indicates the percentage of processor cycles required to decode the *Mother* sequence (Baseline Profile, CIF resolution, 300 frames) by the H.264 reference decoder JM Version18 [43]. Each value in the parenthesis next to a function indicates the percentage of the processor cycles of the calling function that are spent by its child function. For instance, Listing 1 indicates that the decode_slice function uses up 54% of the total processor cycles spent by the calling function

decode_one_frame. The decode_one_frame function spends 98% of all the processor cycles used in the decoder.

Listing 1: Percentage of processor cycles for H.264 High Profile H264 JM18

- decode_one_frame(98%)
  - decode_slice (54%)
    - decode_one_macroblock(82%)
    - read_one_mb_p_slice(10%)
  - exit_picture(45%)
    - DeblockPicture(98%)

In the JM H264 implementation, decode_one_macroblock function contains the inverse transform and motion compensation methods, whereas the read_one_mb_p_slice method contains *Context Adaptive Variable Length Coding* (CAVLC) entropy decoding methods. Further analysis indicates that for the *Mother* sequence %27 of the total decoding processor cycles is used during motion compensation, whereas 40% is used for deblocking, 10% for entropy decoding and 13% is used for the inverse transform. In other words, the aforementioned 4 functional blocks account for the 90% of the total processing power required for decoding the video clip. This ratio has been observed to remain mainly constant for the most of the sequences analyzed in this thesis. Thus, it is justifiable to use the aggregation of the complexities associated with these coding blocks as the total decoding complexity.

3.1.1 Relating Decoding Complexity and Content Features

As mentioned previously, it is possible to determine the decoding complexity of a video clip very accurately by using software tools, such as *Vtune*. However, running *Vtune* increases the total execution time of the decoder and requires significant amount of resources. Thus it is not feasible to run *Vtune* each time the decoding complexity of a new video clip needs to be determined. Furthermore, the results provided by *Vtune* are platform specific, that is, *Vtune* must be run on the target

platform alongside the decoder in order to determine decoding complexity. Apparently, running resource intensive software, like *Vtune*, on mobile platforms is not practical. Thus, a method for predicting the decoding complexity needs to be devised.

In this chapter, a method for predicting video decoding complexity without using resource monitoring tools is proposed. The decoding complexity is modeled as the sum of the complexities associated with the functional blocks previously described. The decoding complexity for each functional block will be modeled by using content features extracted from the bit-stream in real time [20]. The content features that will be used for predicting decoding complexities are proposed as below:

- F_{TN}: The number of nonzero transform coefficients. The number of non-zero transform coefficients depends on the amount of spatial detail in video content. Hence, it is a good indicator of spatial content complexity.
- F_{MB}: Total Number of coded motion vectors. It is possible to code up to 16 motion vectors for each macroblock in H.264. Depending on the motion complexity of the scene only a subset of these motion vectors are coded. Thus, the total number of coded motion vectors is a good indication of motion complexity.
- F_{BR}: Content bit-rate.

F_{TN} will be used to model inverse transform complexity and entropy decoding complexity, whereas F_{MB} will be used for modeling motion compensation complexity. Finally, a combination of F_{MB} and F_{BR} will be used to model deblocking complexity. It should be emphasized that these features are platform-independent and depend only on the content itself, rather than the decoding architecture used. That is, the F_{TN} and F_{MB} values are constant for a sequence coded at a particular quantization parameter (QP) and bit-rate regardless of the decoding platform.

In order to determine whether these features are reliable indicators of decoding complexity, the correlation coefficient between the decoding complexities and the values of the content features is computed for the training set described in Section 2.4.1. The YUV sequences in the training set are first encoded with JMH.264 encoder using the baseline profile and then decoded using the same codec. *Vtune* is launched alongside the decoder in order to measure the complexity of the decoding process.

It should be noted that *Vtune* uses two different modes of operation to measure complexity. One mode, called as the *call graph mode,* records the system clock each time the execution of a function starts or ends. Using the time difference between these measurements, *Vtune* determines the total execution time of the function. However, the execution overhead associated with this *call graph mode* is quite high, and hence the measured execution times are not very accurate. The main advantage of this mode is that, in addition to measuring execution times, *Vtune* also constructs the *execution call graph* that contains the caller and callee relationships between all the functions of a given executable. The other mode of *Vtune* operation is the *sampling mode*. In most of modern CPU's, there are special registers that hold the position of the execution pointer. That is the name of the function that is being currently executed is held in a separate register. In sampling mode, *Vtune* periodically samples the value of this register. The number of samples taken during the execution of each function is used as an indicator of the time spent by the CPU on this function. The sampling mode has a low overhead and the computational complexity measurements performed by using the sampling mode are more accurate than the ones performed using the call graph mode. In this thesis, the sampling mode is used for all complexity calculations.

The correlation coefficients computed using the sampling mode are given in Table 5.

Table 5 : Correlation Coefficients between Decoding Complexities And Content Features

Decoding Block	Feature	Correlation Coefficient
Inverse Transform (C_{IT})	F_{TN}	0,909
Entropy Decoding ($C_{ED)}$	F_{TN}	0,906
Motion Compensation (C_{MC})	F_{MB}	0,988
Deblocking Filter (C_{DB})	F_{MB}-F_{BR}	0,610

As it can be seen from the above table, inverse transform and entropy decoding complexities exhibit substantial correlation with the F_{TN} feature; this relation is expected since these complexities are highly dependent on the content spatial detail. Motion compensation complexity is strongly correlated with F_{MB}. The deblocking filter is correlated with F_{MB}- F_{BR}, however this correlation is not as strong as the correlation between the other content features and decoding complexities. These results show that the complexity components C_{IT}, C_{ED}, C_{MC} have an almost linear relationship with the corresponding content features. The correlation coefficient associated with the deblocking complexity is much smaller indicating that its relationship with F_{MB}-F_{BR} is not linear. It will be shown next that these features can be used to model the total decoding complexity with sufficient accuracy.

3.2 Complexity Modeling with GMM

A statistical approach is followed in order to model the relationship between the content features and decoding complexities. Random variables corresponding to each

content feature (F_{TN}, $F_{MB,}$ F_{BR}) and each complexity (C_{IT}, C_{ED}, C_{MC}, C_{MC}) are assumed. Furthermore, a random vector containing feature random parameters and the corresponding complexity random parameter is constructed for each decoding block. The joint densities of these random vectors are then estimated in order to model the feature-complexity relationship. The random vectors are defined as:

$$\begin{aligned} \mathbf{X_{IT}} &= [\mathbf{C_{IT}}\mathbf{F_{TN}}] \ , \\ \mathbf{X_{ED}} &= [\mathbf{C_{ED}}\mathbf{F_{TN}}] \ , \\ \mathbf{X_{MC}} &= [\mathbf{C_{MC}}\mathbf{F_{MB}}] \ , \\ \mathbf{X_{DB}} &= [\mathbf{C_{DB}}\mathbf{F_{MB}}\mathbf{F_{BR}}] \ , \end{aligned} \tag{20}$$

where $\mathbf{X_{IT}}$, $\mathbf{X_{ED}}$, $\mathbf{X_{MC}}$,$\mathbf{X_{DB}}$ are the random vectors for inverse transform, entropy decoding, motion compensation and deblocking, respectively. The proposed method for the estimation of the joint probability density functions (pdfs) of these random vectors in Eq. (20) is presented next.

The problem of density estimation has been studied extensively in literature and a variety of methods exists for tackling this problem [47].Inspired by the approach in [20], Gaussian Mixture Models (GMM), which is a parametric density estimation method, is utilized in order to model the joint pdfs. In GMM formulation, it is assumed that the density to be estimated can be expressed as a linear combination of Gaussian distributions. Thus, the joint pdf belonging to any one of the random vectors defined in Eq. (20) can be formulated as

$$\mathrm{f}_{\boldsymbol{x_j}}(\mathrm{X}) = \sum_{\mathrm{i}=1}^{m_j} \omega_{\mathrm{i},j}\, \emptyset_{i,j}(\mathrm{X}) \ , \tag{21}$$

Where $j \in \{\mathrm{IT}, \mathrm{ED}, \mathrm{MC}, \mathrm{DB}\}$, $\boldsymbol{x_j}$ is any one of the random vectors defined in (20), $\emptyset_{i,j}$ is the Gaussian distribution corresponding to the i'th component of the GMM for $\boldsymbol{x_j}$, $\omega_{\mathrm{i,j}}$ is the corresponding weighting factor and m_j is the number of Gaussian

components in the GMM for $\boldsymbol{x_j}$. $\omega_{i,j}$'s are chosen such that their sum for each *j* is equal to 1.

$$\sum_{i=1}^{m_j} \omega_{i,j} = 1 \ . \tag{22}$$

Each $\emptyset_{i,j}$ in (21) can be expressed as

$$\emptyset_{i,j}(X) = \frac{1}{(2\pi)^{d/2}|\Sigma_{i,j}|^{1/2}} e^{-\frac{1}{2}(x-\mu_{i,j})^t \Sigma_{i,j}^{-1}(x-\mu_{i,j})} \ , \tag{23}$$

where $\mu_{i,j}$ is the mean vector, $\Sigma_{i,j}^{2}$ is the covariance matrix of the *i*'th Gaussian distribution of the GMM for the random vector $\boldsymbol{x_j}$ and *d* is the dimension of the gaussian distribution. Note that the *d*=3 (i. e. $\boldsymbol{x_j}$ is three dimensional) for *j*=DB and *d=2* (i. e. $\boldsymbol{x_j}$ is two dimensional) for all other complexity components.

In order to obtain the GMMs for the random vectors given in (20), the values of $\omega_{i,j}$, $\mu_{i,j}$, $\Sigma_{i,j}$, and m_j need to be determined. The Expectation Maximization (EM) algorithm is employed in order to determine the values of these parameters. The values of the content features and the associated complexities are extracted from the training videos using *Vtune*. The EM algorithm uses these observations of the random vector $\boldsymbol{x_j}$ in order to determine the values of the model parameters. The observations are given in matrix form as

$$\overrightarrow{X_j} = \begin{bmatrix} X_j^{\,0} \\ X_j^{\,1} \\ X_j^{\,2} \\ \vdots \\ X_j^{\,N} \end{bmatrix} . \tag{24}$$

Each X_j^k in Eq. (24) is a 1xM vector defined as:

$$X_j^k = \left[C_j^k \quad F_j^k\right] , \tag{25}$$

where C_j^k is k^{th} observation of the j^{th} content complexity measured by *Vtune* for a particular video clip in the training set, and F_j^k is the associated content feature vector.

EM is an iterative algorithm. The values of all the model parameters that are going to be estimated by EM need to be manually initialized. The initial values for $\omega_{i,j}$ are set to be equal and sum to 1 as given in (22), the only set of $\omega_{i,j}$ that satisfy these two conditions are as given in the equation below,

$$\omega_{i,j} = \frac{1}{m_j} , \forall i \in [1, m_j]. \tag{26}$$

Initial mean values are randomly chosen to be equal to one of the sample vectors $X_j^k$where $k \in [1, N]$. Initial covariance matrices are all diagonal, where the diagonal element at (*n*,*n*) of the covariance matrix is taken to be equal to the variance of n^{th} column of $\overrightarrow{X_j}$.

The EM algorithm is executed 100 times and is allowed 1000 iterations to converge. The run that yields the largest likelihood is declared as the GMM that will be utilized as the joint density estimate. The details related to determination of the value of the number of Gaussian components m_j for each random vector are given in the next section.

Once the joint densities are obtained by using the above GMM formulation, the decoding complexity for given content features can be determined. The decoding

complexity is the one that maximizes the joint pdf for the given feature vector. This relation is illustrated below:

$$\mathrm{C_j^*(X) = argmax_C\, f_{X_j}(X|F = F_0)}\ . \qquad (27)$$

In the above equation, F_0 is the observed feature vector and C_j^* is the estimated complexity value that maximizes the joint pdf for the given feature value.

Once the most likely values for all inverse transform, motion compensation, entropy decoding, deblocking complexities are obtained, the overall decoding complexity can be simply stated as the sum of individual complexities.

$$\mathrm{DC = {C_{IT}}^* + {C_{MC}}^* + {C_{ED}}^* + {C_{DB}}^*}\ , \qquad (28)$$

where DC is the estimate for the total decoding complexity.

The next section presents the simulation results obtained using the density estimation method outlined below.

3.3 Complexity Prediction Tests

In this section, the decoding complexity of a set of sequences are predicted by using the method described in Section 3.2 . The actual values of the complexities are computed via *Vtune* and the error between the predictions and the actual values are calculated.

The actual complexity values of the 10 sequences that make up the data set described in Section 3.1 are calculated via *Vtune*. The sequences in this set are encoded by using the JM encoder Version 18 in H264 baseline profile. Then, they are decoded using the same codec. *Intel Vtune Software Analysis Tool* is used to collect decoding

complexity data. The procedure is carried out on a Intel Centrino Duo 2000 MHZ PC.

Content features and decoding complexities are calculated individually for each GOP of the sequences in the data set in order to increase the number of data points available. There are a total of 1575 GOPs in the data set, and thus, when Vtune is run for all the GOPs, 1575 pairs of content feature and decoding complexity values are obtained. Then, in order to determine the prediction error for each unique sequence, the GOPs belonging to that sequence are utilized as the test set, while the remaining data is used as the training set. The GMMs are obtained utilizing the data in the training set, and subsequently the content feature values of the GOPs are used together with the obtained GMMs in order to determine the decoding complexities associated with the GOPs in the test set.

One important parameter of the GMM that needs to be determined is m_j, i.e. the number of Gaussian components to be used in the formulation. In order to determine m_j, the negative log likelihood (*NlogL*) and the Bayesian Information Criterion (BIC) metrics [48] are utilized. *NlogL* is an indicator for the fidelity of the model fit. The lower the value of *NlogL* the better the Gaussian models account for the observed data. The BIC is used to determine over fitting. The BIC is calculated as [48]

$$\mathrm{BIC} = 2\mathrm{xNlogL} + \mathrm{zlog(n)} \ , \qquad (29)$$

where *z* is the number of parameters to be estimated to obtain the model and *n* is the number of observations. Since lower *NlogL* values indicate better fit to the data, the first term decreases with increasing modeling accuracy whereas the second term increases with increasing model complexity. Lower BIC values indicate that there is a good balance between accuracy and model complexity, whereas higher BIC values

indicate either that the model does not accurately account for the observed data or that the model is too complex.

BIC, *NlogL*, and prediction error values for the inverse transform complexity GMM and the results are given in Figure 7, Figure 8 and Table 6, respectively.

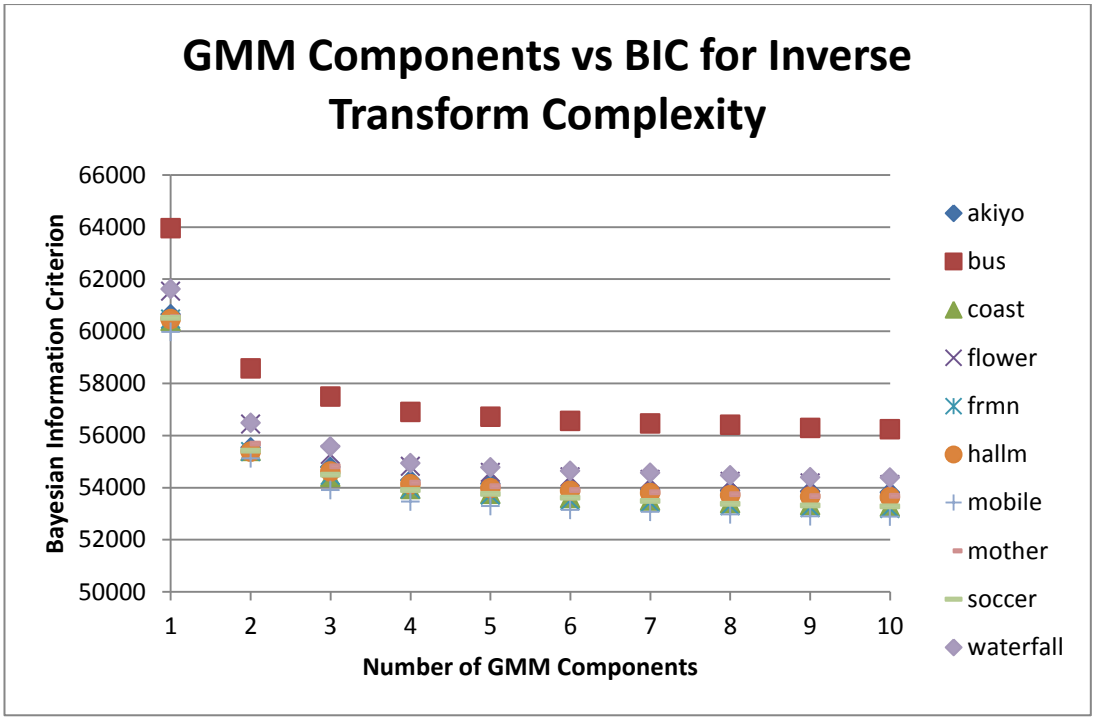

Figure 7 : GMM Components vs BIC for Inverse Transform Complexity

The *NLogL* and BIC values given above indicate that the GMMs model the data more accurately as the number of components in the GMM increases. Nevertheless, the increase in modeling accuracy saturates after the number of components are equal to 4 or larger. On the other hand, the lowest average prediction error is obtained, when the number of components is 3. The reason prediction error increases after 3 components, even though the modeling accuracy is still improving is that the model starts memorizing the training data, rather than generalizing the pattern of relationship between the complexities and content features. Since the test data is not included in the training set, memorization of the training data decreases the

prediction accuracy. The number of components for modeling of inverse transform complexity is chosen as the number which yields the lowest prediction error, i.e. $m_{IT} = 3$.

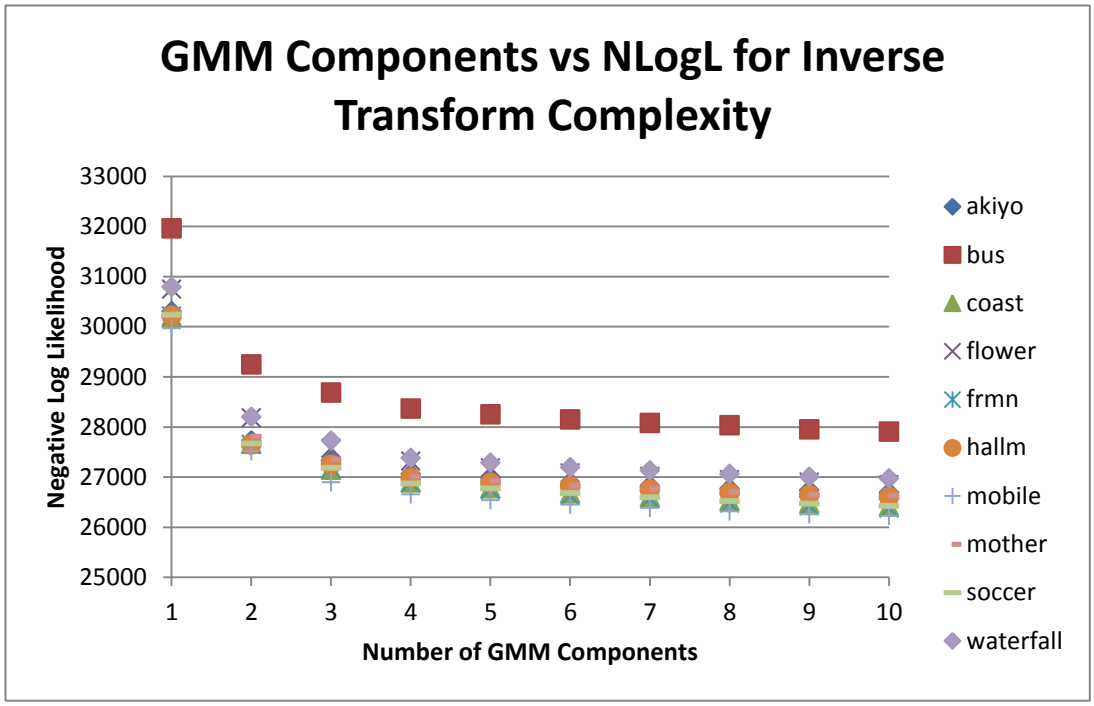

Figure 8 : GMM Components vs NLogL for Inverse Transform Complexity

Table 6 : Prediction Error for Inverse Transform complexity for varying number of GMM Components

Sequence	**Number Of Gaussian Components**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
akiyo	65	64	41	50	47	47	48	44	43	43
bus	17	12	12	20	21	22	22	21	23	20
coast	29	21	21	24	20	27	30	28	32	28
flower	30	23	25	22	25	25	32	34	32	26
foreman	35	21	20	17	19	18	19	20	21	23
hallmntr	32	25	19	18	20	21	27	27	27	23
mobile	26	19	17	15	21	21	21	21	18	17
mother	51	27	24	27	26	26	26	28	28	29
soccer	25	15	15	19	17	19	22	22	21	22
waterfall	34	21	21	19	23	24	25	24	28	29
Average	34,4	24,8	**21,5**	23,1	23,9	25	27,2	26,9	27,3	26

Prediction error values for motion compensation, entropy decoding and deblocking complexities are shown in Tables Table 7, Table 8 and Table 9. The BIC and *NLogL* plots for these complexity blocks are almost identical to the plots for inverse transform given above and thus, they are omitted.

Table 7 : Average percentage of motion compensation complexity prediction error for varying number of GMM components

Sequence	**Number Of Gaussian Components**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
akiyo	37	24	30	27	27	24	23	23	23	24
bus	13	13	13	14	14	14	14	14	13	13
coast	11	11	11	12	12	12	13	13	13	13
flower	9	10	9	10	10	11	10	10	11	10
foreman	18	19	18	17	17	18	17	18	18	18
hallmntr	48	39	43	41	41	41	41	40	41	45
mobile	29	29	29	28	29	28	28	28	28	27
mother	14	12	13	12	13	13	13	13	12	12
soccer	12	13	13	14	14	14	14	14	14	14
waterfall	9	8	9	10	7	9	10	10	10	11
Average	20	**17,8**	18,8	18,5	18,4	18,4	18,3	18,3	18,3	18,7

Table 8 : Average entropy decoding complexity prediction error for varying number of GMM components

Sequence	**Number Of Gaussian Components**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
akiyo	40	26	17	18	18	18	18	18	18	18
bus	8	5	5	10	10	11	11	11	9	8
coast	14	9	7	10	10	11	11	10	10	11
flower	13	10	11	11	12	13	13	13	12	12
foreman	18	12	10	9	9	9	10	11	11	11
hallmntr	17	16	11	11	10	14	10	14	15	14
mobile	10	7	6	5	5	8	8	8	10	8
mother	31	15	6	8	8	8	8	8	8	8
soccer	14	9	8	9	9	10	10	10	10	10
waterfall	17	14	10	10	10	10	10	10	12	7
Average	18,2	12,3	**9,1**	10,1	10,1	11,2	10,9	11,3	11,5	10,7

As illustrated in Table 7, the lowest prediction error for motion compensation complexity is attained, when the number of components in the GMM is 2. Following the reasoning outlined for the inverse transform complexity modeling, the number of GMM components for motion compensation complexity is chosen as 2; i.e., $m_{MC} = 2$. Similarly analyzing the data in Table 8 and Table 9, the number of GMM components for entropy decoding complexity is selected as $m_{ED} = 3$ and the number of components for deblocking complexity is chosen as $m_{DB} = 5$.

Table 9 : Average deblocking complexity prediction error in different content classes for varying number of GMM components

Sequence	**Number Of Gaussian Components**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
akiyo	35	36	44	22	13	16	20	22	41	40
bus	36	32	13	28	27	38	27	39	37	42
coast	36	35	18	31	28	34	29	37	39	36
flower	54	59	65	57	58	57	54	52	52	51
foreman	32	36	20	19	24	25	26	28	26	26
hallmntr	19	17	30	35	23	27	25	31	39	27
mobile	57	56	57	52	50	46	45	49	51	51
mother	64	69	69	64	60	79	86	95	87	93
soccer	34	34	21	30	27	27	28	22	24	21
waterfall	81	92	37	35	36	41	89	74	93	95
Average	44,8	46,6	37,4	37,3	**34,6**	39	42,9	44,9	48,9	48,2

3.4 Summary and Discussions

In this chapter, the video decoding complexity is modeled in terms of content characteristics by using a statistical approach. It is argued that, in hybrid video decoders, the decoding complexity arises from independent decoding blocks, namely *inverse transform, motion compensation, entropy decoding and deblocking*. GMM's are utilized in order to estimate the joint pdfs of content features and component complexities. A separate GMM is used to model the content-complexity characteristics of each decoding block. An off-line training set is used to estimate the

GMM parameters by using EM algorithm. Finally, simulation results that illustrate the GMM fit accuracy are presented for each decoding complexity block. The simulation results demonstrate that although the GMM models the training data more accurately, as the number of GMM components increases, the prediction error starts increasing when the number of components exceed a threshold. The reason for this observation is due to the fact that the GMM starts memorizing the training data and fails to generalize the relationship between the content features and the complexities as the number of components are increased beyond a certain limit. The number of components that are going to be used for each decoding block is taken as the number that minimized the prediction error.

CHAPTER IV

UTILITY-COMPLEXITY FRAMEWORK

4.1 Rate Distortion Optimization

The Rate-Distortion (R-D) theory is concerned with the task of representing a source with the fewest number of bits possible for a given reproduction quality [49].

The classical rate distortion approach aims to determine the theoretical bound on signal fidelity for a given bit-rate, given that the source signal adheres to a probability density function. The bounds obtained by the classical R-D theory are valid for any encoding system.

To be able to derive the bounds, it is very important to characterize the data source accurately. That is, the underlying probability distributions need to be determined. In practice, these bounds are likely to be found only for simple statistical source models. For example, such bounds have been known for independent identically distributed scalar sources with Gaussian, Laplacian and generalized Gaussian distributions. The closed form R-D function is even more difficult to obtain, it is only derived for Gaussian sources. For other sources numerical optimization methods have to be used to estimate the R-D function [49].

Although the bounds on theoretical R-D performance are important for the design of video encoding systems, there are two concerns that need to be considered:

1. Complexity - Can a practical algorithm be constructed whose performance approach the theoretical bounds, and if yes, how much computational power or memory is required?
2. Source Data Model - How accurate does the assumed source distribution, model the actual data? Are the bounds based on this model realistic?

To guarantee that a realistic compression scheme is designed, the operational R-D approach should be used instead of theoretical R-D. The operational R-D approach deals with finding the best encoding parameters in an R-D sense for a specific encoding system performing compression on inputs characterized by a set of data or a probability model. Details of an operational R-D system are given in Appendix-B.

The proposed approach described in subsequent sections is inspired by operational R-D approach.

4.2 Complexity-Distortion Theory

Complexity-Distortion theory (CD) was formulized in [50]. It substitutes a universal *Turing Machine*, instead of the Shannon's decoder in Rate Distortion Theory and uses Kolmogorov complexity to obtain the complexity distortion functions. The details of the theory are out of the scope of this work and the user is referred to the reference for further information. The application of CD theory to video adaptation will be studied next.

A framework for Rate-Distortion-Complexity modeling for video adaptation is proposed in [21]. The complexity of a video stream is first modeled by *generic complexity metrics* (GCM) depending only on the content characteristics. Then these pre-computed GCMs are mapped to real complexity metrics (RCM) that explicitly take into account specific terminal architectures and available resources.

It is impractical to pre-compute complexity for every possible receiver architecture. Consequently, a generic complexity model that captures the abstract GCMs of the employed decoding algorithm depending on the content characteristics and transmission bit-rate is employed [21]. An abstract receiver called as *Generic Reference Machine* (GRM) capable of only performing *add*, *multiply*, *assign* operations is defined. GCMs are derived by computing the average number of times the different GRM operations are executed.

While applying the Lagrangian optimization outlined in Appendix-B to this scenario, it is assumed that each video group of pictures is partitioned into *N* independently coded *adaptation units* (AU). These can be individual frames or slices or macroblocks. Let the set of AUs that correspond to the decoded resolution and frame rate of a GOP be denoted as $\{b_1, b_2, \ldots b_N\}$. Each independent AU is associated with a set of distortion points $\{R_i^{j(i)}, D_i^{j(i)}\}$ with $j(i)$ indicating the corresponding adaptation point. R-D optimization that aims to minimize the overall distortion in the GOP under the rate constraint R_{max} can be formulated as a Lagrangian optimization;

$$\{j_r^*(\mathrm{i}), \lambda_\mathrm{r}^*\} \forall b_i = \arg\min_{\mathrm{j(i)},\lambda} \left\{ \sum_{\mathrm{i}=1}^{\mathrm{N}} \left({D_i}^{j(i)} + \lambda {R_i}^{j(i)} \right) \right\} : R_{GOP} < R_{max} \quad .(30)$$

The Lagrangian multiplier λ must be adjusted until the value $\lambda = \lambda_\mathrm{r}^*$ is obtained, where the rate corresponding to the selected $j_r^*(i)$ is approximately equal to $R_{max.}$

The proposed architecture to jointly model video quality and video decoding complexity is described in detail in the remainder of this chapter.

4.3 Proposed Complexity Constrained Utility Optimization

The aim of this dissertation is to maximize the overall Quality of Experience (QoE) of an end user watching a video clip on a resource limited mobile device. In this dissertation, QoE is used synonymously with *video utility*, i.e. the subjective user

satisfaction pertaining to video content. On the other hand, *video quality* refers to the inherent content quality that depends on coding bit-rate, frame rate and spatial resolution. In a perfect environment, i.e. error-free networks, and mobile terminals with limitless processing power, video utility would be equivalent to video quality. However typical video consumption scenarios are never perfect and high quality videos do not always lead to better user experiences. As discussed in Chapter 3, the amount of resources required for decoding and displaying video is called as *decoding complexity*. Increasing the video quality usually increases the decoding complexity of the video stream. Since mobile devices are inherently resource limited, increasing the *decoding complexity* beyond the devices capabilities will result in an ill-fated decoding process. In other words, the device will not able to perform the necessary computations to decode the stream in real-time and this situation will result in temporal distortions, such as dropped frames and motion jerkiness. These distortions are bound to reduce the video utility significantly. A higher user satisfaction could probably be attained by coding the video with reduced complexity (i.e. with reduced spatial resolution, limited bit-rate or frame rate), so that the resources required for decoding do not exceed the device capabilities. Consequently, it is clear that blindly increasing video quality does not always lead to increased user satisfaction, hence, quality and decoding complexity should be jointly optimized in order to obtain maximum QoE.

The proposed approach for maximizing video utility by jointly optimizing video quality and decoding complexity is described in the following subsections.

4.3.1 Video Quality

The first step of the proposed approach is measuring video quality. It has previously been discussed in Chapter 2 that video quality can be measured either objectively or subjectively. Subjective measurement comprises designing experiments involving human evaluators to determine the perceived video quality, whereas objective measurement utilizes algorithms that automatically compute quality. Subjective

measurements are accepted as the ultimate judge of video quality, whereas objective methods try to come up with results that are as close to subjective values as possible. However, objective methods are widely preferred over subjective ones as it is very difficult to find human experimenters and designing subjective experiments that will yield consistent results on different laboratories requires strict standardization of experimental procedures.

In this dissertation, the VQM metric is utilized for modeling video quality. It is claimed that the use of VQM metric does not significantly decrease the accuracy of video quality prediction, as it is shown [5][6] that the VQM metric is highly correlated with subjective results. The details of modeling video quality using the VQM metric are presented in Chapter 2.

4.3.2 Decoding Complexity

The second step of the proposed approach is predicting the decoding complexity. In order to achieve this, the method presented in Chapter 3 is utilized. . Overall decoding complexity is proposed to comprise of the complexities of 4 main coding blocks; i.e. *inverse transform*, *motion compensation*, *entropy decoding* and *deblocking filter*. Such a decomposition of decoding complexity does not assume the utilization of a specific decoder, since these coding blocks are present in all modern hybrid codecs. In order to predict the complexity associated with each coding block, specific content features, which are shown to have significant correlation with decoding complexities of the coding blocks, are extracted from the video in real time. Content features and associated complexities are computed for a training set of sequences. Then the joint statistics of the content features and the decoding complexities obtained from the training set is used to determine the complexity of a new sequence with known content features. The total decoding complexity for a video clip is obtained by summing the complexities associated with individual coding blocks. The details of decoding complexity estimation are given in Chapter 3.

4.3.3 Video Clip Decodability

Video quality can be modeled using VQM quite accurately, but it should be kept in mind that VQM only reflects the inherent content quality; it does not keep track of whether the computational resources of the device are sufficient to properly decode the video in real time. Consequently, in order to predict video utility, video decoding complexity and viewing device capacity should also be taken into account in addition to VQM.

While predicting utility, the effect of distortions resulting from insufficient computing power is accounted for utilizing the concept of *decodability.*. The *decodability* of a video clip on a particular client device can be predicted by comparing the video decoding complexity with the computational capacity of the device. If the decoding complexity is less than the device capacity the video is said to be *decodable*, whereas if the complexity is more than the device capacity the video is *not decodable*. The severity of the decoding distortions for not decodable videos depends on the extent of the decoding complexity. For videos with decoding complexities only slightly higher than device capacity, the associated distortions may be indiscernible to the human observer.

The decoding complexity of a video clip obviously depends on the coding parameters (frame rate, resolution, bit-rate etc.) in addition to the content itself. If the decoding complexity of a video clip can be calculated for different values of coding parameters, the set of parameters that will yield a decodable representation of a video on a particular device can be obtained.

In Section 4.5.1, subjective tests that measure the decodability of a video clip on a mobile device are presented. Subsequently, in Session 4.5.1, a method for predicting the decodability score of a video clip utilizing its decoding complexity is proposed.

4.3.4 Modeling Subjective Quality

As discussed in the previous section, when the device resources are sufficient to decode the video in real time, the video utility can be modeled using only inherent content quality. However, when device resources are not sufficient, video decoding complexity and device resources should also be taken into consideration. The decodability concept, again introduced in the previous section, accounts for the effect of video decoding complexity and device capacity on video utility.

Building on these concepts, it is proposed that the utility can be modeled as a linear combination of video quality and decodability as:

$$\widehat{U} = a\,DB(DC, R) + bQ + c\,, \tag{31}$$

where $\widehat{U}$ is the predicted video utility, *DB* is the video decodability, *DC* is the video decoding complexity, *R* is the device resource capacity and *a*, *b* and *c* are real valued constants. *Q* is the objective video quality which will be modeled by using the VQM metric. Note that $a = 0$ for videos whose decoding complexity is less than the device capacity since, in that case, video utility prediction is considered equivalent to objective video quality.

Eq. 31 summarizes one of the most important contributions of this thesis. We claim that, in order to determine the utility for videos with decoding complexities higher than the device capacity, it is sufficient to add a decodability term to the utility equation. We also claim that the decodability is a function of video decoding complexity and device resources. These claims will be verified in Section 4.6 by using subjective quality experiments to measure the actual value of *U* and then using Eq. 31 to obtain the prediction $\widehat{U}$.

4.3.5 System Architecture

The proposed architecture is presented in Figure 9. An off-line video training pool is utilized in order to estimate model parameters for both video quality (VQM) estimation and decoding complexity prediction (GMM) models. In addition, decodability experiments, which determine the video decoding capacity of mobile devices, must be performed for each device. When a new video clip is requested from the video streaming server by the mobile device, initially the SI and TI metrics are extracted from the video content and VQM values for the clip are predicted for various QP and frame rate values. Then, for each video representation (i.e. each QP - frame rate pair), content features related to decoding complexity are extracted from the bit-stream and the decoding complexities for each coding block are estimated by using the GMMs that are already obtained during off-line training. The total complexity is obtained as the sum of complexities of the coding blocks. Using the decodability tests performed on the device and the total decoding complexity of the video clip, the decodability score of the video clip for the particular device is determined. Using the decodability score and the VQM value obtained for each different bit-rate and frame rate, the subjective quality of each video representation is estimated. The representation with the highest subjective quality estimate is chosen as the optimal representation of the video i.e. the representation that will yield the highest quality of experience. A transcoder module is used to transcode the video to the determined optimal representation and to deliver the transcoded video to the end terminal.

The details of decodability and utility predictions are presented in subsequent chapters.

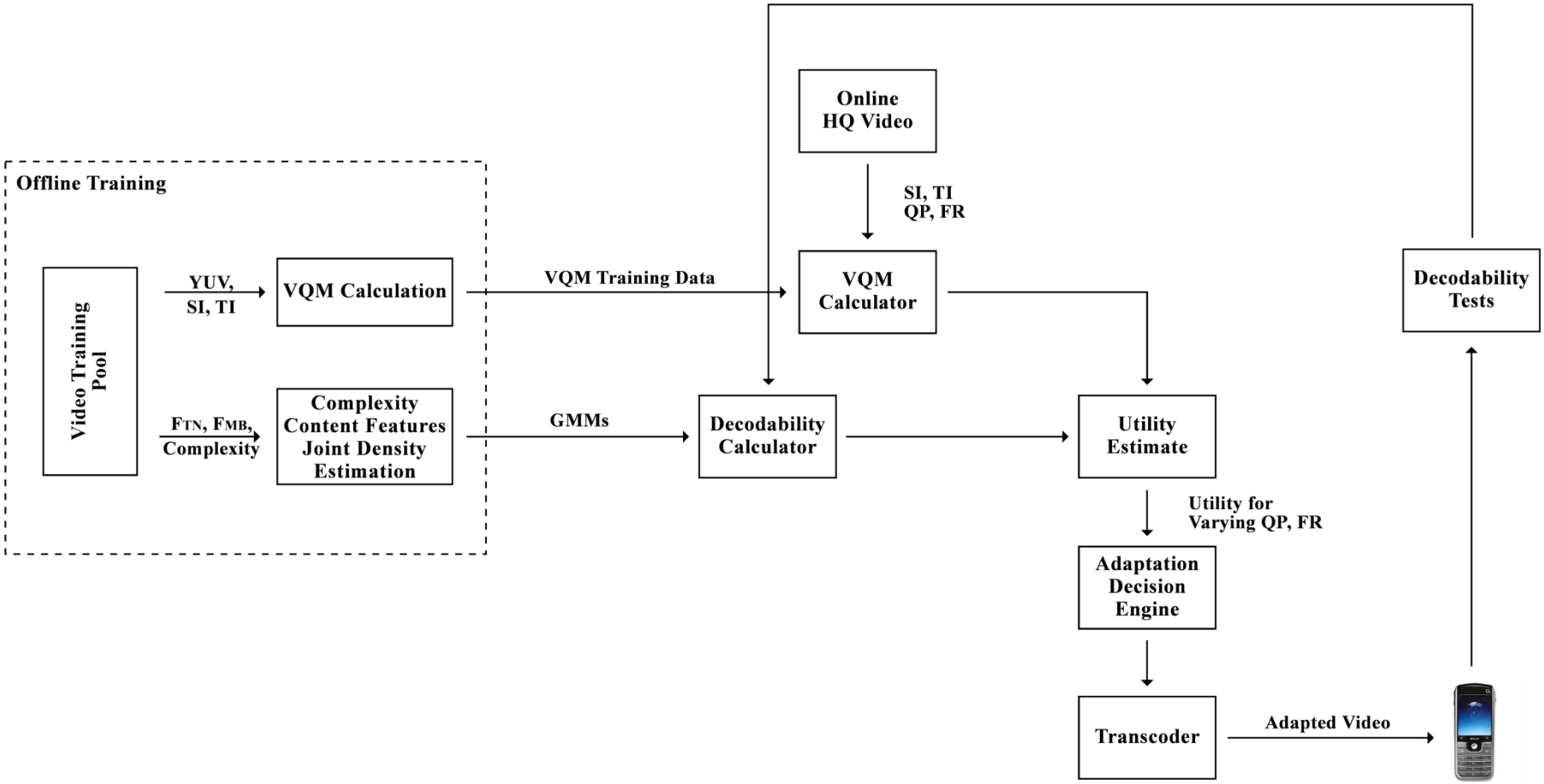

Figure 9 : Proposed System for Determining Optimal Adaptation Operation

4.4 Quality Complexity Joint Optimization

In order to be able to deliver the best QoE to end users, video quality and decoding complexity need to be jointly optimized. This is a multi-objective optimization (MOO) problem. Generally higher quality videos are more complex and low complexity videos have poor quality. Thus, minimizing complexity and maximizing quality are conflicting goals. The solution of the above problem can be quite complex; furthermore, there is no unique solution for such a MOO problem. There are pareto-optimal solutions for which one of the objectives cannot be improved without sacrificing from the other. The ideal choice among pareto-optimal solutions is usually application specific and can change as the video consumption scenario changes.

As discussed previously, most modern handsets have advanced processing capabilities. For these devices, the main concern is the available battery power rather than the processing power. On the other hand, for less capable devices available processing power becomes critical for real time decoding. In the light of the above discussion, it is clear that the ideal solution to the quality complexity joint optimization problem depends on the usage scenario. Taking this into account, we divide the MOO problem into two single objective optimization problems.

1. Given that the available processing power is sufficient for real time decoding, minimize the battery power consumption while keeping the video quality above a certain threshold.

   It is very difficult to measure the consumed battery power while decoding a video clip. Hence decoding complexity should be minimized instead of power consumption, since the two quantities are quite correlated [22] and decoding complexity can be measured using the approach presented in Chapter 3.

Thus the constrained optimization problem can be formulated as:

$$\min \mathrm{DC}(QP, FR) \quad \text{s.t. } \mathrm{Q}(QP, FR) > U_0 \ , \tag{32}$$

where *QP* is the video quantization parameter, *FR* is the frame rate and U_0 is the video quality threshold.

2. Given that there is an upper bound on the complexity of a video that can be decoded in real time (i.e. for mobile devices having limited processing capabilities), maximize the video quality while keeping the decoding complexity below a certain threshold.

   This version of the optimization problem can be formulated as

$$\max \mathrm{Q}(QP, FR) \quad \text{s.t. } \mathrm{DC}(QP, FR) < C_0 \ , \tag{33}$$

   where C_0 is the complexity threshold. This threshold needs to be determined separately for each mobile device.

   It is quite difficult to determine the value of C_0 accurately for a mobile device, since the actual decoding capacity depends on many factors like hardware architecture, operating system utilities, etc. More importantly, a constant complexity threshold C_0, for which there is no deterioration in utility while $\mathrm{DC}(R, S) < C_0$ and there is a sudden decrease in utility immediately after complexity exceeds C_0, does not exist in practice. Video utility deteriorates gradually, rather than abruptly, as the decoding complexity is increased beyond the device capacity. In this dissertation, instead of trying to determine a constant C_0 value for each device, the decodability concept is utilized in order to model the extent of deterioration in QoE as decoding complexity of video clips is

increased beyond the device decoding capacity. The decodability experiments are described in the next section.

In order to solve the first problem, the Lagrange multiplier method will be utilized. The Lagrangian for this problem can be written as,

$$\mathrm{L}(R,S,\lambda) = \mathrm{DC}(R,S) + \lambda(\mathrm{VQM}(R,S) - \mathrm{U}_0) \ . \tag{34}$$

Investigating the critical points, we obtain the following relations:

$$\begin{aligned} &\lambda(\mathrm{VQM}(\mathrm{R},\mathrm{S}) - \mathrm{U}_0) = 0 \ , \\ &\frac{\partial \mathrm{DC}}{\partial \mathrm{R}} + \lambda \frac{\partial \mathrm{VQM}}{\partial \mathrm{R}} = 0 \ , \\ &\frac{\partial \mathrm{DC}}{\partial \mathrm{S}} + \lambda \frac{\partial \mathrm{VQM}}{\partial \mathrm{S}} = 0 \ . \end{aligned} \tag{35}$$

From the first condition in Eq. (35), either λ=0 or $\mathrm{VQM}(\mathrm{R},\mathrm{S}) - \mathrm{U}_0 = 0$. If λ=0, from the second and third conditions, the following relations are obtained: ∂DC/∂R = 0 and ∂DC/∂S=0. Assuming that the derivatives can not be zero, it follows that λ cannot be zero; thus, we have $VQM(R,S) = U_0$ as the solution. This result indicates that we have to code the video at the minimun acceptable quality in order to minimize complexity.

The solution for the second problem can be obtained using a similar approach. The solution in that case is to code the video with the maximum decoding complexity, i.e. $\mathrm{DC}(R,S) = C_0$, that the device can tolerate, in order to maximize video quality. As discussed previously this problem will be dealt with using a different approach in this dissertation. Rather than coding the video at C_0, the amount of detorioration in video utility as the *DC* exceeds C_0 will be predicted, and the video representation with the highest utility will be selected as the optimal representation. Such a method allows choosing video representations that have a larger DC value than C_0, this is especially

meaningfull when video representations exist with DC value slightly higher than C_0 and having utility significantly larger than other videos with DC value less than C_0.

It is generally not possible to encode a particular video, so that its quality or decoding complexity will exactly equal some value, such as U_0 or C_0. Usually, a high quality version of the video clip is available at the server and certain adaptation operations need to be applied to the bitstream in order to reduce its complexity and meet the constraints of the optimization scenario. Several types of adaptation operations i.e. frame droppping, resolution reduction, etc., can be utilized in order to reduce the resource requirements.

FD-CD adaptation scheme described in Chapter 1 will be utilized in order to transcode a video clip to the desired quality or computational complexity level. Frame dropping will be used for coarse adaptation and coefficient droppping will be used for finer adaptation. It is assumed that an adaptation engine that is capable of encoding/transcoding the video to 3 different frame rates (15, 20 and 30 frames per second) and 3 different QP values (15, 25, 45) is readily available. The implementation details of such an encoder/transcoder is out of the scope of this thesis, but various implementations exist in the literature [29].

The optimal video coding parameters for the usage scenarios described in the two aforementioned optimization problems will be determined by searching among the possible combinations of FD-CD adaptations. For the first problem, only the FD-CD operations that produce videos having quality higher than U_0 are considered. Among these, the optimal FD-CD operation will be the one that produces the video having the lowest decoding complexity. For the second problem, optimal FD-CD operation will be the one that produces the video with highest utility.

4.5 Predicting Decodability

As discussed in the previous section, the *QoE* of an end user watching a video clip on a mobile device is significantly influenced by the decoding complexity of the video clip and the computational resources available on the device. Typically, decoding a high quality video stream requires resources in excess of the device capacity. This is bound to hinder the user experience, since the video cannot be decoded properly in real time and artifacts, such as frame dropping are observed. Consequently, it is very important to determine whether a given representation of a video clip can be decoded in real time on a particular mobile device. One approach to achieve this would be to calculate the maximum decoding complexity that the device can accommodate by using information about the devices hardware and software specifications (CPU, DSP, OS etc.),. However, the decoding performance of a mobile device depends on many factors and it is not possible to reliably obtain a constant complexity threshold C_0 (see problem 2 in Section 4.4 for a particular device that is applicable to all scenarios.

In this dissertation, an alternative method to determine the real time decodability of a video clip on a mobile device is proposed. The mobile device is considered as a black box that takes compressed video as input and provides decoded video as output. A set of videos with varying resource requirements is played back on the device and the presence of artifacts that arise from lack of processing resources are investigated. According to the number and magnitude of these artifacts, each video representation (a video clip coded at a particular bit-rate and frame rate) is given a decodability score for the particular mobile device.

The decodability score for the video clips can be measured using objective or subjective methods. An objective method could involve an algorithm implemented in the mobile device that measures the amount of frame lag or number of dropped frames, while the video is being displayed. The subjective methods could involve subjective video evaluation tests that ask the evaluators to rate the impairments that

result from artifacts caused by insufficient device resources. In this dissertation, the subjective method is utilized. The details of the subjective video quality experiments are given in Section 4.5.1

After determining the decodability score, the decoding complexity of each video representation is calculated by using the method described in Chapter 3. The statistical relationship between the decoding complexity and the decodability score is used as the basis for determining the decodability score of a previously unknown video clip for a particular mobile device. In other words, decoding complexity and decodability score are assumed to be random variables, and an estimate for their joint density is obtained using the measurements from the video clips in the training set. In order to predict the decodability score for a new video clip, it is sufficient to calculate its decoding complexity; the decodability that maximizes the joint density given the decoding complexity is taken as the decodability score. The details of the estimation procedure are described in the next Section 4.5.3 .

4.5.1 Subjective Tests for Measuring Decodability

In order to determine the decodability score of a video clip on a mobile device, subjective video evaluation experiments are performed on a Nokia N 81 mobile phone. The *Double Stimulus Impairment Scale* (DSIS) method is used during the experiments. A total of 10 evaluators participated in the tests. The tested videos are all stored locally on the mobile devices in order to eliminate the effects of the transmission network on perceived video quality. When the effects of the network are eliminated, motion related artifacts (e.g. frame drop, motion jitter etc.) arise exclusively from insufficient device resources. Thus, in order to determine the decodability score of the test videos, the evaluators are asked to rate videos only according to the number and magnitude of motion related artifacts and disregard other non-motion related artifacts, such as blockiness, ringing, etc. Reference videos are played back on a desktop computer, whereas the test videos are displayed on the

mobile device for which the decodability scores are to be obtained. The grading is based on the difference between the smoothness of motion (i.e. lack of motion artifacts) of the reference and the test videos. Videos are rated on a scale of 1 to 5, a grade of 5 indicating that no motion artifacts are observed during playback, and thus, the video is subjectively pleasing, and a grade of 1 indicating that the observed artifacts are severe and significantly hinder the user satisfaction.

The complexity of the videos presented in the subjective tests is measured using the Intel *Vtune* Analyzer. The details of the complexity measurements are presented in Chapter 3.

Figure 10 presents the relationship between the decodability scores and the decoding complexities obtained by using the described method. The results in Figure 10 indicate that the decoding complexity is a good indicator of real time decodability. It can be observed that the sequences having the lowest decoding complexity are the ones that are decodable without any distortion (i.e. sequences that have a decodability score of 5) and as the decoding complexity is increased beyond a certain limit the decodability scores of the videos start decreasing. These results encourage the utilization of decoding complexity for predicting the decodability of a video clip on a mobile device. A method for predicting the decodability scores using the decoding complexities is presented in 4.5.3 .

4.5.2 Statistical Analysis of Subjective Test Results for Decodability

The results of any subjective video evaluation experiment should be presented together with the 95% confidence intervals [33]. The 95% confidence interval is given as

$$\left[\mu_{i,j} - \delta_{i,j}, \mu_{i,j} + \delta_{i,j}\right] , \qquad (36)$$

where $\mu_{i,j}$ is the average opinion score over all observers i and all video clips j. $\delta_{i,j}$ is given as [32]

$$\delta_{i,j} = 1{,}96\frac{S_j}{\sqrt{N}}\ , \tag{37}$$

where N is the total number of observers and $S_j$is the standard deviation for each video clip is given by [32].

$$S_j = \sqrt{\sum_{i=1}^{N}\frac{(\mu_j-\mu_{i,j})^2}{N-1}}\ . \tag{38}$$

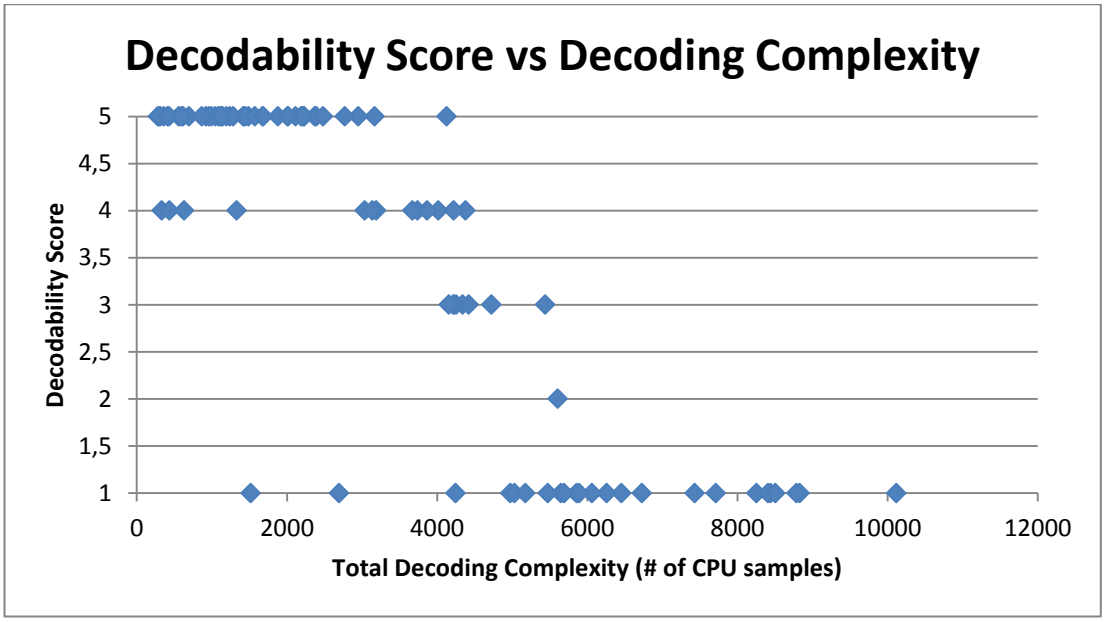

Figure 10 : Decodability Scores vs Total Decoding Complexity for Nokia N 81

The 95% confidence interval signifies that the absolute error between the experimental mean and the true mean (i.e. the mean value for a very large number of observers) will be within the confidence interval with 0.95 probability as long as the distribution of scores meet certain requirements.

For the decodability score subjective experiments, the average 95% confidence interval is found to be 0.0770. Such a small value for the confidence interval indicates that the results successfully account for the statistical spread of the opinions of the evaluators.

The data provided by the participants of the subjective test are also analyzed for consistency. The data provided by each user is subjected to observer screening procedures outlined in [32]. For this purpose, the *kurtosis coefficient* [32] which is the ratio of the second moment to the fourth moment is calculated. The kurtosis coefficient is given by

$$\beta_{2i,j} = \frac{m_4}{(m_2)^2} \ , \tag{39}$$

where $m_k$is given by

$$m_k = \frac{\sum_{i=1}^{N}(x_{i,j}-\mu_{i,j})^k}{N} \ . \tag{40}$$

The procedure that was followed in order to determine whether the scores given by an observer are consistent with rest of the scores is given below:

For each observer *i*, determine Q_i, P_i where

If $2 \leq \beta_{2i,j} \leq 4$ then:

$$\text{if } x_{i,j} \geq \mu_{i,j} + 2S_{i,j} \text{ then } Q_i = Q_i + 1 \ , \tag{41}$$

$$\text{if } x_{i,j} \leq \mu_{i,j} - 2S_{i,j} \text{ then } P_i = P_i + 1 \ , \tag{42}$$

Else:

$$x_{i,j} \geq \mu_{i,j} + \sqrt{20}S_{i,j} \text{then} Q_i = Q_i + 1 \ , \tag{43}$$

$$x_{i,j} \leq \mu_{i,j} - \sqrt{20}S_{i,j} \text{then} P_i = P_i + 1 \ , \tag{44}$$

If

$$\frac{P_i+Q_i}{i\ j} \geq 0.05\ . \tag{45}$$

and

$$\left|\frac{P_i-Q_i}{P_i+Q_i}\right| < 0.3\ . \tag{46}$$

Then reject observer *i.* otherwise include observer *i*'s score in experiment results.

None of the observers were rejected as a result of the observer screening tests for the decodability score experiments. All the Q_i and P_i scores are zero, except for 1 observer who has a P_i score of 1. This result indicates that the observer was voting slightly more negatively, when compared with the other observers; however, the negativity is not strong enough to disqualify him.

The results in Figure 10 clearly indicate that there is a strong inclination in users to vote a video either as totally decodable (i.e. with decodability score 5) or strongly undecodable (i.e. with decodability score 1). The number of intermediate scores is fewer than the extreme scores by a large margin. Figure 11 illustrates the histogram of the results. As it can be seen from the figure, the histogram is skewed towards the extremes i.e. the frequency of decodability scores of 1 and 5 are much larger than the frequency of the other votes. There is actually only 1 result with score 2. These results suggest that for the decodability experiments, a 3 level grading scale could have been used, instead of the 5 level grading scale without losing generality. Nevertheless, the current 5 level grading scale will be used in the remainder of this thesis.

The next section present an algorithm that aims to predict the decodability score of a video clip using its decoding complexity.

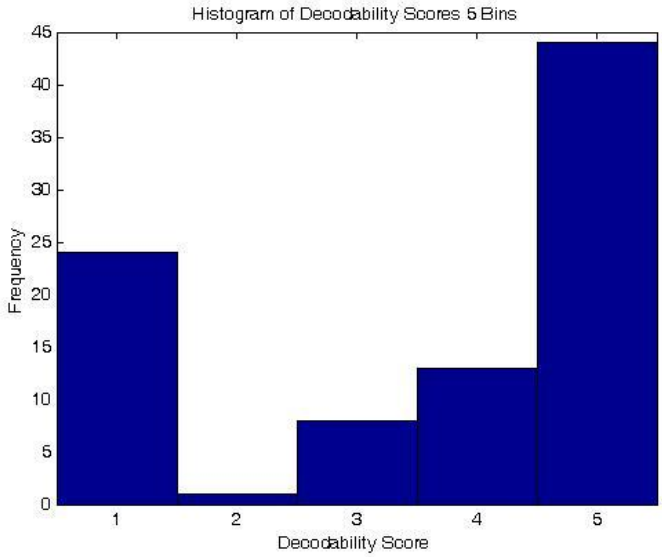

Figure 11 : Histogram of Decodability Scores for 5 bins

4.5.3 Predicting Decodability Utilizing Decoding Complexity Statistics

The decodability score of a video clip on a particular mobile device needs to be estimated. In order to be able to predict the decodability score, the joint probability density of total complexity and decodability is estimated by using a kernel based estimation technique [52]. A Gaussian kernel is used with a heuristically selected bandwidth of 0.04 times the maximum value of input total complexity. The data obtained from subjective decodability experiments presented in Section 4.5.1 is used as the training data for the kernel estimator. Figure 12 shows the joint density estimated utilizing the training data presented in Figure 10.

Figure 12 also illustrates that the least complex sequences have a decodability score of 5, since for the lowest complexity videos the joint pdf peaks around decodability score 5. As the complexity increases, the joint density first peaks around decodability score of 4 and as the complexity is increased further the pdf peaks are more pronounced around decodability scores 3 and 1.

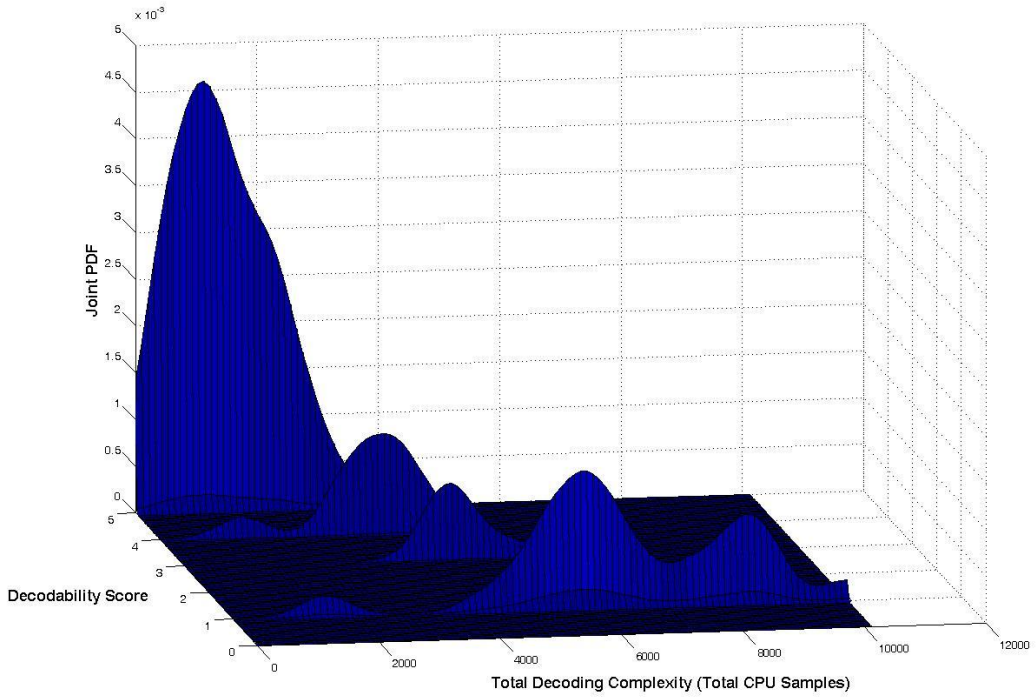

Figure 12 : Decodability vs Complexity Kernel Estimate

Once the joint statistics is obtained, it is possible to estimate the decodability score for a given complexity in an optimal way. Following *maximum a posteriori* (MAP) formulation [53], for a given total complexity value, the decodability score is estimated as the score that maximizes the joint density estimate. That is the decodability score is obtained using the MAP estimate:

$$DB = \underset{db}{\operatorname{argmax}}\, f_{db,dc}(db|dc = dc_i) \qquad (47)$$

where DB is the estimate for the decodability score and db and dc are the random variables for decodability and complexity respectively. The parameter dc_i is the observed total complexity value.

The decodability prediction errors for the data set described in Section 2.4.1 is given below.

Table 10 : Decodability Prediction Error

Sequence	Prediction Error
Akiyo	1.37%
Bus	17.95%
Coast	15.87%
Flower	28.28%
Foreman	15.24%
Hall monitor	20.82%
Mobile	10.24%
Mother	1.69%
Soccer	15.32%
Waterfall	34.79%
AVERAGE ERROR	16.16%

The results in Table 10 indicate that the decodability values can be predicted quite accurately using the method presented above.

The next section presents a method to predict the subjective video quality using video quality and decodability scores.

4.6 Subjective Quality Prediction

In order to determine the representation (i.e. encoding parameters) of a video clip that will yield the highest *utility* when viewed on a mobile device, the *utility* of each different representation of the video clip needs to be determined.

If the mobile device is capable enough to decode all representations of the video clip, utility is only affected by distortions like blockiness and ringing that arise from insufficient coding bandwidth. However, if device resources are not sufficient for real time decoding, the utility will also be affected by distortions that arise from lack of device resources, such as frame dropping. Thus, in order to predict utility accurately, it is necessary to take into account the decoding complexity, and the device decoding capacity in addition to the content quality.

The most reliable method of measuring subjective quality is through subjective evaluation experiments. However, as described previously, subjective experiments are quite difficult to perform. Thus, it is desirable to predict subjective quality by using alternative methods. In Section 4.6.2 a method for predicting subjective video quality is presented. In order to measure the performance of this method, subjective video evaluation tests which establish the ground truth for subjective quality is performed. The subjective tests are presented in Section 4.6.1

4.6.1 Subjective Video Quality Evaluation Tests

In this section, subjective tests that accurately measure the perceived quality of video clips displayed on a mobile device are performed.

Subjective video quality evaluation tests are performed on a set of 90 video clips. The set is comprised of 10 unique sequences coded with 3 different QP values (15, 25, 45) and 3 different frame rates (15, 20, 30 fps). *Single Stimulus Impairment Scale* method is used during evaluation. The experiments are performed on a Nokia N-81 mobile phone. A grading scale of 1 to 5 is utilized, where a grade of 1 indicates that the videos subjective quality is very poor whereas a grade of 5 indicates that the video is as subjectively pleasing as possible for the platform under test. A total of 10 evaluators participated in the tests. The subjective quality of each video clips was taken as the Mean Opinion Score (MOS) of the grades given by the evaluators.

In a subjective video quality evaluation experiment, it is important to have evenly distributed scores (i.e. the number of grades in each level of the grading scale should be on the same order of magnitude) [6]. Having an even distribution indicates the sequences used in the test adequately span the entire range of possible quality levels. Figure 13 illustrates the histogram of the subjective grades given in the test. It can

easily be observed that the histogram is not skewed and the distribution of results is quite homogeneous.

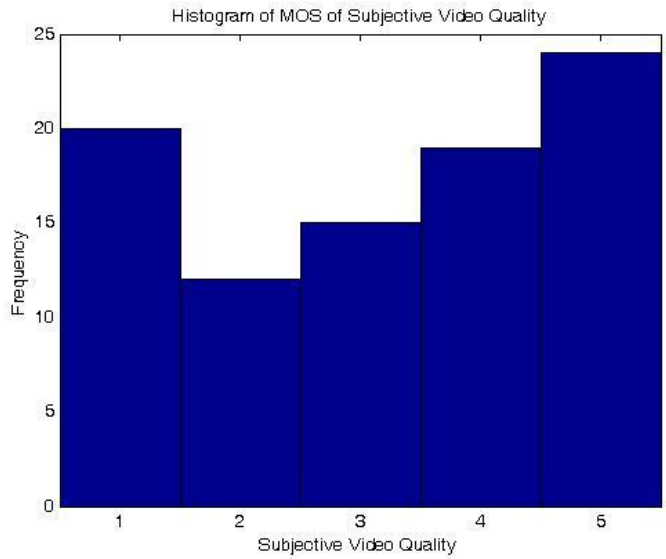

Figure 13 : Histogram of MOS Subjective Video Quality

The average 95% confidence interval is found to be 0.1020. Note that the 95% confidence interval values are significantly larger than the confidence interval values for the decodability tests. This result is expected, since it is much more challenging to give an overall score to the satisfaction on watching a video clip rather than to grade only the presence of motion related artifacts. As a result, the user opinions have a larger spread and the number of observers needed to obtain statistically meaningful results is higher. Nevertheless, the confidence interval values obtained indicate that the experiments sufficiently accounted for the spread in user opinion scores. The observer screening techniques described in Section 4.5 are applied and none of the observers are rejected by the algorithm.

In the next section, an algorithm for predicting the subjective quality given the values of content quality, decoding complexity and device decoding capacity is presented.

4.6.2 Predicting Subjective Quality Utilizing Decodability and Complexity

In order to predict the subjective quality of the video clips used in the subjective quality experiments described above, two sets of video clips are formed. One set is used as the test data and the other as the training set. The test set is made up of the same 90 sequences (10 unique sequences each coded at 3 different QP and 3 different frame rates) used in the subjective quality evaluation experiments. The training set is made up of the same 10 sequences coded with the same values of QP and frame rate, but the video clips are divided into 16-frame chunks, yielding a total of 1575 video chunks. VQM and decoding complexity values are computed separately for each chunk. All the tests presented below are performed such that the test and the training sets are always mutually exclusive. When the subjective quality of a particular test sequence is to be determined, all chunks belonging to the test sequence are removed from the training set. Thus, the actual size of the training set is always less than 1575.

Initially, the training set data is processed in order to obtain the statistical models that will be used to predict video quality and the decoding complexity values of the videos in the test set. The following values are calculated for the sequences in the training set:

- SI,TI values (A single SI, TI value is calculated for a particular 16 frame portion of a video clip regardless of its bit-rate and frame rate. That is the same SI,TI value is assigned to all video chunks with different QP and frame rate values, as long as they belong to the same video clip and the same 16 frame portion.),
- VQM values,
- Content Features (Number of non-zero transform coefficients and number of non-zero motion vectors),
- Gaussian mixture models for inverse transform, motion compensation, entropy decoding, and deblocking complexities.

The subjective quality of a test sequence is predicted by using the algorithm below:

The first step in predicting the subjective video quality of a test video clip is estimating its VQM value. In order to predict the VQM value of the test sequence, a *nearest neighbor* approach is followed. The video chunk in the training set that is the closest to the test sequence is determined and its VQM value is used as the prediction for the VQM value of the test sequence. In order to determine the closest chunk, the SI and TI values of the test sequence are calculated. Then, the video chunks in the training set, whose SI and TI values are closest in an Euclidian sense to the SI and TI value of the test sequence are determined. The set of closest video chunks contains at least 9 different elements, since each distinct video clip (i.e. a particular 16 frame portion of a particular video) is coded with 9 different encoding parameters (3 frame rates x 3 QP values) and thus, it has 9 representations in the training set. All 9 of these representations have the same SI and TI value as discussed previously. Among the 9 representations, the one which has the same frame rate and the same QP value as the test sequence is selected as the closest to the test data. The VQM value of the training data is utilized as the prediction for the VQM value of the test data. The details of VQM estimation are given in Chapter 2.

The second step is determining the decoding complexity. The decoding complexity is proposed to be comprising of 4 components, namely inverse transform, motion compensation, entropy decoding and deblocking complexities. These complexities represent the main blocks of code found in most hybrid video coders. In order to estimate decoding complexity, the relationship between content features and the complexity components are exploited. A joint density is estimated for each complexity component. A parametric density estimation method i.e. GMMs is utilized for this purpose. The GMMs are obtained from the video clips in the training set. Once the GMMs are obtained, the content features of the test sequence can be used to predict the component complexity values, i.e. the complexity that maximizes the joint density for the given content features is taken as the complexity prediction. The number of non-zero motion vectors content feature is used for estimating the

motion compensation complexity and deblocking complexity, whereas number of non-zero transform coefficients feature is used for estimating the inverse transform and entropy decoding complexities. Once the component complexities are determined, the total complexity is obtained as the sum of component complexities. The details of complexity estimation are presented in Chapter 3.

The third step is estimating the decodability of the test clip for the mobile device. Decodability estimation is presented in detail in Section 4.5 and thus will not be repeated here.

The fourth and final step is using decodability scores and the VQM data to predict the subjective video quality. As described in Section 4.3.4 , it is assumed that the utility values can be predicted by using a linear combination of VQM values and the decodability scores. Modifying Eq. (31) using VQM to represent video quality we obtain the equation below:

$$\widehat{U} = a\ x\ DB(DC, R) + b\ x\ VQM + c\ \ , \qquad (48)$$

where $\widehat{U}$ is the prediction for video utility, DB is decodability score, *VQM* is the value of the *VQM* metric, and *a, b* and *c* are real values constants. An important point about the above equation is that $a \leq 0$, since for mobile devices that can decode all representations of a video in real time, video utility is only a function of VQM and thus *a*=0; whereas for mobile devices that have limited processing capabilities, the decodability term accounts for the deterioration in video utility resulting from temporal distortions.

The values of *a, b, c* given in Eq. 26 are determined by using Surface Fitting Toolbox of MATLAB. In order to determine the values of *a, b* and *c* for a particular sequence, the utility, VQM and decodability values belonging to the other 9 unique sequences in the test set are utilized as the training data for curve fitting. In other words, while

determining the values of *a*, *b*, *c* for a particular sequence, none of the videos belonging to that sequence is present in the training set. The values of *a*,*b*,*c* obtained for the 10 sequences are given in the table below.

Table 11 : *a*,*b*,*c* Values for each Sequence with Sequence Removed from Training Set

Sequence	a	b	c
Akiyo	-1.55	0.57	1.63
Bus	-2.00	0.61	1.64
Coast	-1.96	0.60	1.69
Flower	-1.98	0.63	1.57
Foreman	-1.77	0.64	1.44
Hallmonitor	-2.14	0.63	1.66
Mobile	-1.86	0.57	1.78
Mother	-1.39	0.57	1.61
Soccer	-1.88	0.60	1.64
Waterfall	-1.87	0.67	1.45

Using the values of *a*,*b*,*c* given in Table 11 above, the subjective quality scores are predicted. The *relative prediction error* (RPE) metric given in Eq. 49 is utilized in order to compute the prediction error.

$$\mathrm{RPE} = \frac{\sum_{i=1}^{N} |\hat{U}_i - U_i|}{\sum_{i=1}^{N} U_i} \mathrm{x}\ 100\% \qquad (49)$$

The per sequence prediction errors are given in Table 12.

An important point to consider is the effect of the specific objective metric (i.e. VQM) in prediction performance. Table 13 illustrates the prediction results when PSNR is utilized instead of VQM for subjective quality prediction.

Table 12 : Subjective Quality Prediction Error - VQM

Sequence	Prediction Error
Akiyo	12.76%
Bus	33.13%
Coast	25.94%
Flower	30.45%
Foreman	23.84%
Hall monitor	29.53%
Mobile	25.96%
Mother	17.99%
Soccer	22.68%
Waterfall	35.48%
AVERAGE ERROR	25.78%

Table 13 : Subjective Quality Prediction Error - PSNR

Sequence	Prediction Error
Akiyo	12.80%
Bus	37.69%
Coast	21.33%
Flower	24.96%
Foreman	24.75%
Hall monitor	26.23%
Mobile	32.64%
Mother	20.25%
Soccer	24.61%
Waterfall	40.64%
AVERAGE ERROR	26.59%

Comparing the results in Table 12 and Table 13, it can be observed that using PSNR instead of VQM does not result in a significant loss of prediction accuracy. The

reason for this observation is the presence of temporal distortions (frame dropping etc.) caused by inadequate device decoding capacity. These distortions affect the human subjective opinion significantly, whereas the effect of the spatial distortions such as blockiness, noise etc. is relatively unimportant. As a result, the evaluators tend to overlook spatial distortions in video clips which have no temporal distortions. This situation could significantly decrease the effect of the specific objective metric on prediction accuracy, since the precision of the utilized objective metric is relevant only for measuring spatial distortion.

4.7 Determining Optimal Adaptation Operation

In the previous section we have demonstrated how to predict *subjective quality* of a video sequence, by using its total complexity and VQM values. The final stage of the proposed approach is choosing the optimal adaptation operation that is going to be applied to the video clip before being sent to the client terminal. As described in Section 4.4 , there are two different cases for which the optimal adaptation operation needs to be determined.

1. The resources of the mobile device are sufficient to decode the video clips – minimize the overall complexity as long as the video quality is above a certain threshold.
2. The resources of the mobile device are not sufficient to decode the video in real time – maximize video quality as long as the decoding complexity is below a certain threshold.

For the first problem, i.e. the case where mobile device is strong enough to decode any video that is played on it, the VQM value will be used as the utility estimate, since the decodability score is not expected to affect the quality score (i.e. all the decodability scores will be equal to 5). However, for the second case, the linear

combination of VQM and the decodability score will be used as the utility estimate as described in the previous section.

It is expected that for Case 1, the video quality will initially improve with increasing complexity until at a certain point the improvement in quality saturates and further increasing the complexity will not significantly improve the quality. The optimal adaptation point will be the one that has the lowest complexity among the points that have subjective quality prediction larger than a threshold U_0. It should be kept in mind that the candidate adaptation points are representations of the same video sequence coded with different video coding parameters. Thus, it is obvious that the highest subjective quality that can be attained will differ from sequence to sequence. As a result, rather than specifying a constant value of U_0, it makes more practical sense to take U_0 as the highest quality value attained by the candidate points. It is also reasonable to look for the optimal adaptation operation among the ones that yield a quality value within a small interval around U_0, i.e. U_0-δ rather than looking only at points that have quality exactly equal to U_0. This enables the algorithm to make a better choice in the R-D sense, especially if there are adaptation points that yield a quality value that is slightly less than U_0, but have complexities which are much smaller than the complexity of the points that yield a video quality of U_0.

Figure 14 illustrates the relationship between VQM and total complexity for Case 1. The values in the y-axis are obtained using the transform (1-VQM) x 5. This normalization is performed in order to transform the VQM values to a similar range with the subjective quality scores (i.e. VQM values have a range from 0 to 1 with 0 having the highest quality, whereas subjective quality scores have a range from 1 to 5 with 5 having the highest quality.). As it can be observed from Figure 13, the quality value saturates after a certain value of complexity as expected, and further increasing complexity beyond this point does not cause a significant improvement in quality. It should be noted that each point in the graph represents one of the 10 sequences described earlier coded with a certain QP and frame rate value. Since there are 3 different frame rates and 3 different QP values, each sequence is represented by 9 points in the graph.

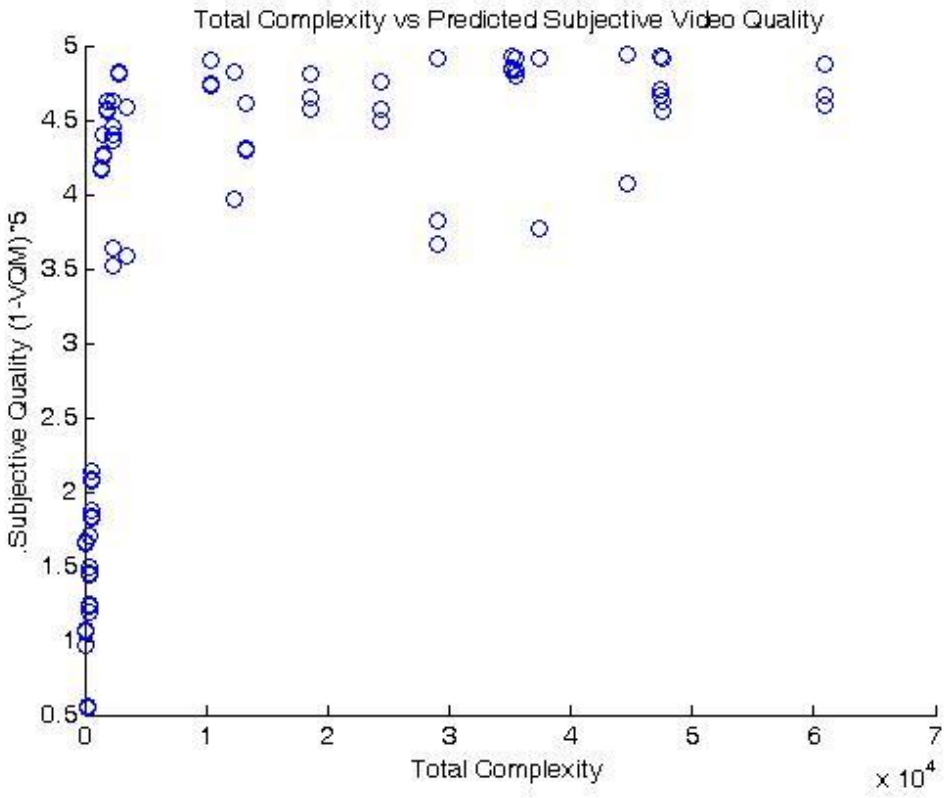

Figure 14 : Total Complexity vs Predicted Subjective Video Quality

For Case 2, the quality estimate is a linear combination of VQM and decodability scores as specified in Eq. (26). Unlike Case 1, where VQM is directly used as the quality estimate, the combination of VQM and decodability scores can account for the decrease in subjective video quality that occurs when the video complexity exceeds device processing capacity. It is expected that increasing the complexity will initially increase the video quality up to a certain point at which the complexity of the video will have reached the devices decoding capacity. Increasing the complexity further will result in a sharp decrease in quality, since the device will not be able to decode the video in real time. The optimal adaptation operation is the one that yields the highest subjective quality U_{max}. However, again in order to make a better choice in the R-D sense, it is reasonable to investigate all adaptation points that yield a quality value within a small interval around U_{max} i.e. [U_{max} ,U_{max}-δ]. The optimal

adaptation operation is the one which yields a video with subjective quality value within the interval [U_{max}, U_{max}-δ] and has the lowest complexity value among others that have subjective quality values within the same interval. The value for δ is chosen heuristically as 0.15. The results for the sequences in the test set are illustrated in Figure 15.

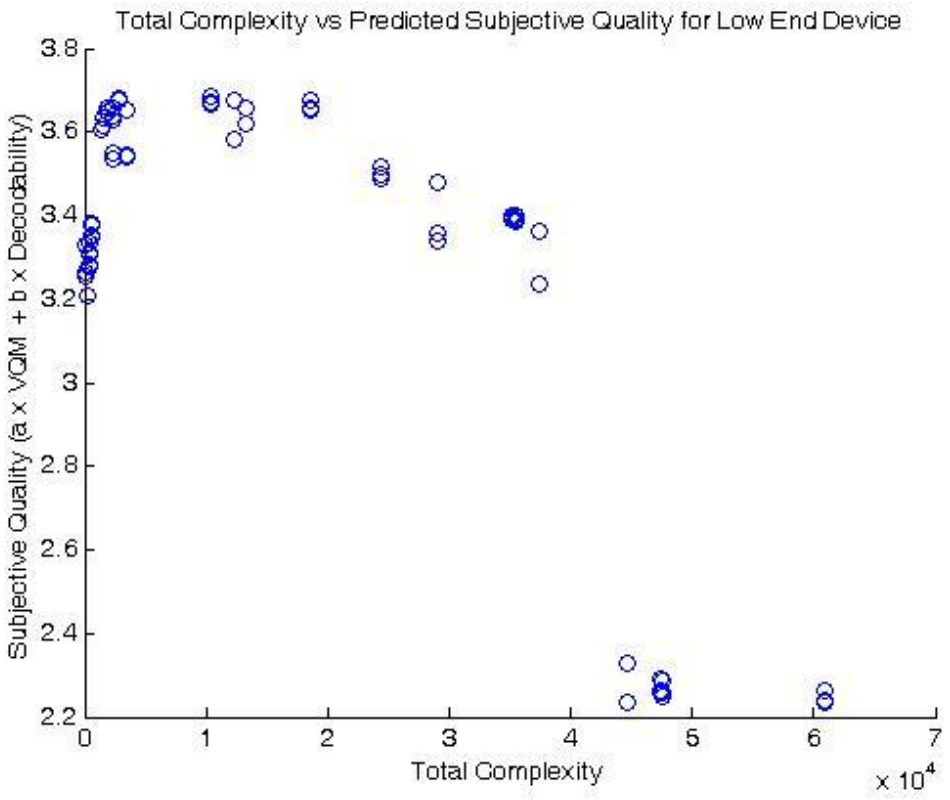

Figure 15 : Total Complexity vs Predicted Quality for Low End Device

The following algorithm summarizes the proposed method for determining the optimal adaptation operation for a test sequence:

i. For high capacity mobile devices:
    a. VQM values are used as the subjective video quality estimate

b. Let A_i be one of the 9 permissible adaptation operations that produces videos with 3 QP and 3 frame rate values. Then the optimal adaptation operation $A^*$is given by:

$$A^*[V] = V^*\ s.t, \tag{50}$$
$$U^* \in [U_0 - \delta, U_0 + \delta]\ and\ DC^* = \min(DC_i)\ \forall i\ \in \{A_1, A_2, \dots, A_9\}\ ,$$

where A^* is the optimal adaptation operation, V is the original video sequence, V^* is the adapted video. U^* is the utility of V^*, U_0 is the minimum quality threshold, DC^* is the decoding complexity of V^*, A_i's are the permissible adaptation operations and DC_i are the decoding complexities of the videos that were adapted using A_i's

ii. If the resource capacity of the mobile is limited:
    a. The relation given in Eq. (48) is used to obtain the subjective quality estimate by using the VQM and decodability scores
    b. the optimal FD-CD operation will be the one that produces adaptation points that yield a quality value within a small interval around U_{max}, i.e. $[U_{max} - \delta, U_{max}]$ where U_{max} is the maximum quality value attained by representations of the video clip and δ is a heuristically determined threshold. That is:

$$\mathrm{A^*[V] = V^*\ s.t\ ,} \tag{51}$$
$$\mathrm{U^* \in [U_{max} - \delta, U_{max}]\ and\ DC^* = \min(DC_i)\ \forall i\ \in \{A_1, A_2, \dots, A_9\}\ \ ,}$$

where U_{max} is the maximum utility attained by the adaptation operations A_i.

Table 14 lists the optimal adaptation operations that are determined using the Subjective video quality scores vs. the ones that are determined using the algorithm above.

Some of the rows in Table 14 have multiple results separated by slashes. These indicate that the corresponding adaptation operations have equal utility. For *Akiyo,*

Mother and *Waterfall* sequences, the adaptation operation which yields the highest predicted utility value is not chosen as the optimal operation since within an interval $\delta = 0{,}15$ of the utility value for the highest utility video (i.e. $[U_{max} - \delta, U_{max}]$) there are one or more video clips having much lower decoding complexity. Thus, the video representation having the lowest decoding complexity and among the representations with $U\epsilon\ [U_{max} - \delta, U_{max}]$ is chosen as the optimal representation.

Table 14 : Optimal Coding Parameters Using SubjectiveTests vs Proposed Algorithm

Video Clip	**Optimal Coding Parameters based on Subjective Tests**			**Optimal Coding Parameters estimated by the Proposed Algorithm,** $\delta = 0.15$		
	QP	**FR**	**Score**	**QP**	**FR**	**Score**
Akiyo	15	15	5	15	15	4,21
Bus	25	15	4,88	25	15	4,08
Coast	25	15	4,75	25	15	4,24
Flower	25	15	4,25	25	15	3,84
Foreman	25/15	15/15	4,63	25	15	4,23
Hallmonitor	25/25	15/30	4,75	25	30	4,49
Mobile	25	15	4,25	45	30	3,38
Mother	25	20	4,88	25	20	4,17
Soccer	25	15	4,5	25	15	4,38
Waterfall	25	15	4,88	25	15	4,27

As can be observed from Table 14, for 9 out of 10 videos, the result of the subjective video quality experiments agree with the results of the proposed prediction algorithm. These results indicate that the proposed algorithm can be used to predict the subjective results quite accurately. Although, all the subjective experiments are performed on a specific mobile phone, the algorithm is applicable to all platforms as long as decodability experiments (Figure 10) are performed for each platform. As discussed previously, it is possible to construct objective decodability tests as

opposed to the subjective tests provided in this thesis. Using objective decodability tests is advisable in order to apply the algorithm to various mobile devices, since performing subjective tests separately for each device is quite difficult.

4.8 Summary and Discussions

The aim of this chapter is to determine the optimal adaptation operation that yields a video with high quality and low decoding complexity. For this purpose, the VQM and decoding complexity values obtained in the previous chapters are utilized together with the decodability scores obtained from subjective evaluation experiments.

It is argued that the joint quality-complexity problem could be divided into two simpler single objective optimization problems. The first problem is applicable to mobile devices, which have very high processing capabilities and are assumed to be able to decode any video clip in real time without any artifacts. For these devices, the optimal adaptation operation can be obtained by minimizing the total complexity (e.g. in order to maximize battery power), while keeping the video quality above a certain threshold. The second problem is applicable to mobile devices with limited processing capabilities. For these devices, the optimal adaptation operation can be obtained by maximizing the video quality (e.g. in order to provide a seamless viewing experience), while keeping the decoding complexity below a certain threshold.

Decodability experiments are performed in order to determine quantity and extent of motion related artifacts that arise from lack of device resources, while decoding videos with varying resource requirements. It is argued that the decodability experiments can be either objective or subjective. In this chapter, subjective decodability experiments are performed.

The subjective quality of a video clip is predicted as a linear function of VQM values and decodability scores. Subjective video quality experiments are performed in order to build a ground truth for subjective quality. Then, the results of the proposed prediction algorithm are compared with the results of the subjective experiments. It is shown that for 90 video clips, the subjective quality value can be predicted with 25% error. The same results are also utilized to determine the optimal adaptation operation. It is shown that for 9 out of 10 cases the prediction algorithm chooses the same adaptation operation as the subjective ground truth. These results indicate that the proposed algorithm can be used for accurately predicting the subjective video quality.

CHAPTER V

SUMMARY, CONCLUSIONS AND FUTURE DIRECTIONS

5.1 Summary

The advances in mobile networks and processing capabilities of handheld terminals have made ubiquitous access to rich multimedia data possible. However, there are many challenges that need to be overcome before interoperable solutions, that make access to multimedia transparent to end users, are widely deployed. One of the most important challenges is delivering multimedia content in a format that is tailored according to end terminal processing capabilities, transmission network conditions and content characteristics. This dissertation aims to present an end-to-end solution that enables delivering video with high subjective quality to end users, while minimizing the resources required for decoding the resulting bit-stream.

There are two main methods of measuring video quality, i.e. subjective quality measurement and objective quality measurement. Subjective measurement represents the perceived subjective quality by end users and is thus the ultimate measure of quality. However there are many difficulties associated with subjective testing and in practice objective metrics are utilized in order to model video quality.

There are two objective metrics that exhibit significant correlation with subjective scores, i.e. the VQM metric and the SSIM metric. VQM metric is used to measure video quality, since it has been shown to have the highest correlation with subjective opinion scores. It is proposed that sequences sharing similar content characteristics should have similar objective quality when encoded with identical encoding parameters (i.e. bit-rate, frame rate and resolution). A training set of videos whose

VQM values are measured in advance can be used to predict the objective quality of a new video whose content characteristics are similar to one or more videos in the training set. In this context, content similarity is measured using ITUs SI and TI metrics.

Video decoding complexity of the H.264 reference decoder is modeled in terms of video content characteristics. Video decoding complexity is divided into four components, as. *inverse transform*, *motion compensation*, *entropy decoding* and *deblocking*. The complexity of each component is modeled individually by using relevant content features that are extracted from the compressed bit-stream in real-time. The joint probability distribution of content features and component complexities for the videos in the training set are estimated via Gaussian Mixture Models. Then, in order to determine the decoding complexity of a new video clip, the GMMs estimated from training data and the content features extracted from the current video clip are utilized.

Decodability experiments are performed in order to determine quantity and extent of motion related artifacts that arise from lack of device resources, while decoding videos with varying resource requirements. It is argued that the decodability experiments can be either objective or subjective. In this dissertation, subjective decodability experiments are performed. The joint statistics of decodability and decoding complexity are obtained for a training set of videos. Using these joint statistics, the decodability of test sequences are predicted utilizing their decoding complexity values.

Video decoding complexity and video quality are then jointly optimized. The multiple objective optimization problem of simultaneously minimizing the decoding complexity and maximizing video quality is split into two single objective problems.

1. Given that the available processing power is sufficient for real time decoding, minimize the decoding complexity while keeping the video quality above a certain threshold.
2. Given that there is an upper bound on the complexity of a video that can be decoded in real time (i.e. for mobile devices having severely limited processing capabilities), maximize the video quality while keeping the decoding complexity below a certain threshold.

The solution to the above problems using Lagrangian formulation is presented.

Finally, an algorithm for predicting the optimal video coding parameters that result in a video having maximum possible quality while having decoding complexity less than the device decoding capacity is presented. It is argued that using simple adaptation operations, such as frame dropping and transform coefficient dropping, video streams that have the desired coding parameters could be obtained. In order to assess the performance of the prediction algorithm, subjective quality tests are performed and the video coding parameters that result in video clips with highest subjective quality scores are determined. It is observed that for 9 out of 10 sequences, the results of the prediction algorithm agree with the subjective experiments.

5.2 Conclusions

This thesis aims to provide the highest possible Quality of Experience (QoE) for an end user watching a video clip on a resource limited mobile device. In order to achieve this aim, video encoding parameters that will result in video clips having high quality and low decoding complexity should be determined. These coding parameters are determined by jointly considering video quality, decoding complexity and device processing capability.

Video quality is measured via VQM metric; there are two main results that are obtained while modeling video quality. The first one is that video quality is highly correlated with ITUs SI and TI metrics. In other words, videos with similar SI,TI values have similar objective quality. The second result is that even a simple classifier, such as nearest neighbor algorithm, can be used to predict the video quality with sufficient accuracy provided that a large enough training set is utilized.

Video complexity statistical modeling is performed using Gaussian Mixture Models. Inverse transform, motion compensation, entropy decoding, and deblocking complexities are estimated using content features extracted from the bit-stream. With the exception of deblocking complexity, all complexities are estimated using one dimensional content features. For deblocking complexity, no single feature yields sufficient prediction accuracy. Thus, two content features are used simultaneously for deblocking complexity estimation.

Decodability experiments that determine whether a video clip with a given predicted complexity can be decoded in real time on a particular mobile platform are performed. It is observed that videos having low complexity indeed have high decodability scores, whereas videos with high complexity have low decodability scores. This result indicates that the predicted complexity accurately mimics the real life hardware platform specific decoding complexities of the video clips.

Finally, decoding complexity and video quality are jointly optimized. Based on the simulations, it is argued that a simple linear combination of video quality and decodability score can be used to model the subjective video quality with satisfactory accuracy.

In conclusion, the proposed approach is quite promising as it allows accurate prediction of the subjective quality score of video clips using much simpler and practical methods. Using this approach, it is possible to maximize user QoE while watching a video clip on a mobile device.

5.3 Future Directions

Optimizing video delivery to mobile devices is a very dynamic field of research, new devices, video delivery methods and network improvements are being developed incessantly. Some of these technologies gain wide adoption while others disappear even before hitting the market. Consequently, future research directions change with the shifting technology. In the context of this thesis, the most important direction of future research is related to determining the video decoding capabilities of mobile terminals. In Chapter 4, it is claimed that instead of performing subjective decodability experiments it is possible to measure the decodability of a video clip programmatically. Utilizing objective decodability metrics, instead of subjective experiments would significantly increase the applicability of our approach. Thus, it is necessary to develop objective decodability metrics for at least one mobile platform and show that the results of the objective metrics are indeed correlated with the subjective results.

One of the most important features of video adaptation algorithms is being able to execute in real time. In this thesis, SI, TI metrics are utilized in order to measure the similarity of video content. Although, SI, TI metrics are effective for content modeling, they are full reference metrics and the video stream needs to be decoded upto pixel domain in order to calculate SI, TI values. This severely limits the algorithms usability, since these metrics cannot be calculated in real time. Identifying content features that can be obtained from the compressed bit-stream and can discriminate content as accurate as SI, TI metrics is an important future research direction.

APPENDIX-A

THE H.264 STANDARD

In 1998, the Video Coding Experts Group (VCEG) ITU-T SG16 Q.6 issued a call for proposals on a project called H.26L with the target to double the coding efficiency, that is halving the bit rate for a given fidelity, in comparison to any other existing video coding standard for a broad variety of applications. In December 2001 VCEG and MPEG formed a Joint Video Team (JVT) to work on this standard, and the specification was finally completed meeting the target efficiency requirements in March 2003 119 [54]. The completed projects new name was H.264. Today, the H.264 standard is amended to MPEG-4 (part 10) and is one of the most efficient video codecs in existence.

The new standard is designed to address a broad range of application areas (broadcast, conversational, video on demand etc.), and these applications should be deployable on existing and future networks. To address the need for flexibility the H.264/AVC design employs a layered architecture. There are two main layers: the video coding layer (VCL) and the network abstraction layer (NAL). The VCL is designed to efficiently represent the actual video data. The NAL formats the VCL representation and provides header information according to the transport layer to be used.

NAL

NAL facilitates the ability to map H.264 VCL data to transport layers such as

- RTP/IP (both for conversational and streaming),
- File formats like ISO mp4 for storage and the MMS format,

- H.32x for wired/wireless conversational services,
- MPEG-2 systems for broadcasting.

NAL Units

The coded video data is organized into NAL units. A NAL unit is a data packet that contains an integer number of bytes. The first byte is a header byte indicating the type of data in the NAL units, and the remaining bytes are the payload. NAL unit structure has a generic format that can be used for both packet oriented and bit-stream oriented transport systems. A series of NAL units that are generated by an encoder is called a NAL Unit Stream. Payload in a NAL unit can contain "emulation prevention bytes" used to prevent accidental occurrences of start code prefixes.

VCL and NON-VCL NAL Units

There are two types of NAL units: VCL NAL units and Non-VCL NAL units. VCL NAL units contain data from the video pictures. Non-VCL NAL units contain associated additional information like parameter sets.

Parameter Sets

Parameter sets contain important header information that can be applied to a large number of VCL NAL units. Parameter sets are also classified into two main types: Picture Parameter Sets and Sequence Parameter Sets. Picture parameter sets apply to the decoding of one or more individual pictures within a coded video sequence. Sequence parameter sets apply to a series of consecutive coded video pictures called as a coded video sequence.

Each VCL NAL unit contains an identifier that refers to the relevant picture parameter set. Each picture parameter set contains an identifier that refers to the relevant sequence parameter set. This allows omitting the header information in the

parameter set, if the parameters to be specified have the same values with the ones already specified inside the sequence parameter set. The parameter set can be transmitted within the same channel that carries the VCL NAL units ("in-band transmission"). Alternatively, they can be sent from a separate more reliable channel ("out-of-band transmission").

Access Units

Access Units are a set of NAL Units in a specified form. The decoding of each access unit results in one decoded picture. The structure of an access unit is presented in Figure 2. Following an optional access unit delimiter, there is a Supplemental Enhancement Info (SEI) block that contains data such as picture timing information. The SEI block is also optional. Primary Coded Picture consists of a set of VCL NAL units comprised of slices or slice partitions that represent the samples of the video picture. Primary coded picture block is mandatory.

Following the primary picture, redundant coded picture block contains VCL NAL units that contain redundant representations of areas or whole of the same video picture. The redundant coded picture is used for recovering from loss or corruption of the data in the primary coded picture. Decoders are not required to decode the redundant coded pictures if they are present. An end of sequence NAL unit is present if the coded picture is the last picture in a coded video sequence. An end of stream NAL unit may be present, if the coded picture is the last coded picture in the entire NAL unit stream.

Coded Video Sequence

A series of pictures that is independently decodable and uses only one sequence parameter set is called as a Coded Video Sequence. Consists of a series access units that are sequential in the NAL unit stream. At the beginning of the coded video sequence, there is an Instantaneous Decoding Refresh (IDR) access unit, which

contains an intra coded picture. In this respect, a coded video sequence is quite similar to the Group of Pictures (GOP) coding structure used in the previous coding standards. A NAL unit stream may contain one or more coded video sequence.

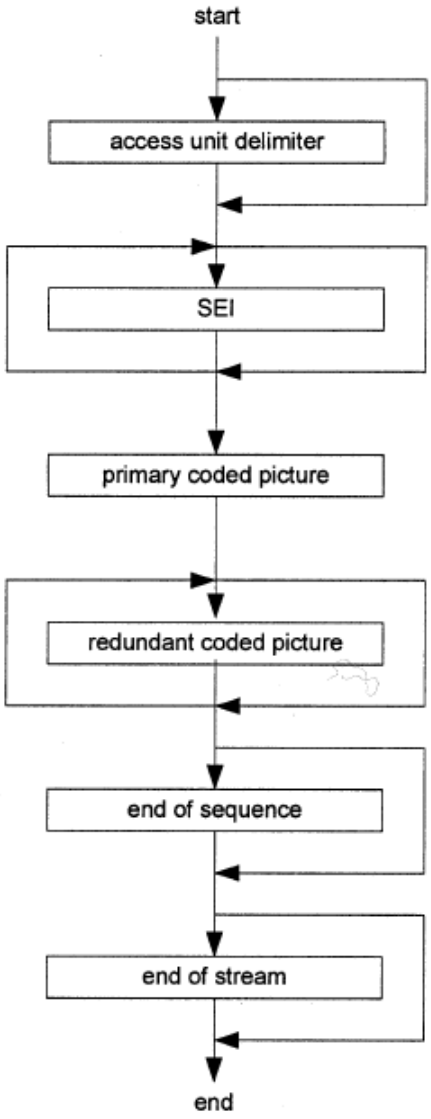

Figure 16 : NAL Access Unit [54]

VCL

A coded video sequence in H.264 consists of a sequence of coded pictures. A coded picture can either represent an entire frame (progressive) or a single field (interlaced). H.264 uses the YCbCr color space with 4:2:0 sampling and 8 bits per

sample. A picture is partitioned into fixed size macroblocks that each covers a rectangular picture area of 16x16 samples of luma and 8x8 samples of each of the two chroma components.

Slices and Slice Groups

A picture may be split into one or more slices in H.264. Slices are essentially a group of macroblocks which are processed in the order of a raster scan (when not using Flexible Macroblock Ordering (FMO)). Slices are self contained in the sense that the values of the samples from the video sequence can be recovered, given the picture parameter sets and the reference pictures. Some information across other slices may be needed to apply a deblocking filter across slice boundaries.

A slice group is a set of macroblocks defined by a macroblock to slice group map. This map consists of a slice group identification number for each macroblock in the picture. A picture can be split into many macroblock scanning patterns using FMO. Slice groups can be partitioned into slices such that a slice is a sequence of macroblocks within the slice group that are processed in the order of raster. The case of not using FMO is equivalent to FMO with the whole picture consisting of a single slice group.

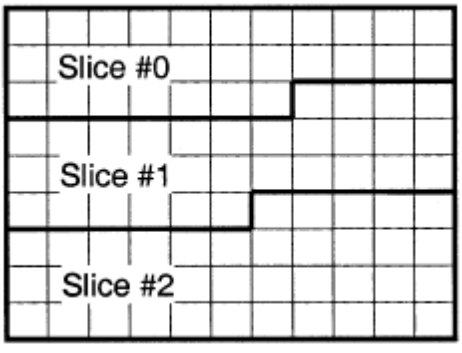

Figure 17 : Subdivision of a picture into slices without using FMO [54]

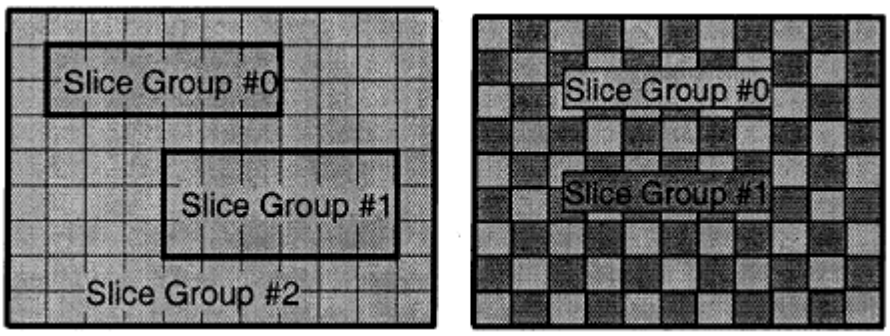

Figure 18 : Subdivison of a frame into slices with FMO [54]

Regardless of whether FMO is used or not, the slice can be coded using different coding types (I slice, P slice, B slice, SP slice, SI slice).

All luma and chroma samples of a macroblock are either spatially or temporally predicted and the prediction residue is encoded using transform coding. For transform coding, each color component of the residual signal is divided into 4x4 macroblocks. Transform coefficients are then quantized and entropy coded.

Intra Frame Prediction

In contrast to previous coding standards, Intra prediction is always conducted in the spatial domain in H.264 (In H263+ and MPEG-4 it is performed in the transform domain). Intra prediction is done by referring to neighboring samples of previously coded blocks which are to the left and above the block being predicted. Intra prediction is not used across slice boundaries in order to keep all the slices independent from each other. Each macroblock can be transmitted in one of the several coding types depending on the slice coding type.

Inter Frame Prediction

Motion compensated coding types are specified as p-macroblock coding types. Each p-macroblock type corresponds to a specific partitioning of the macroblock into the

block shapes used for motion compensated prediction. Partitions with luma block sizes of 16x16, 16x8, 8x16 and 8x8 are supported. 8x8 partitions can further be divided into 8x4, 4x8 or 4x4 samples. Thus, if a macroblock is coded using four 8x8 partitions and each 8x8 partition is split into 4x4 partitions, 16 motion vectors may be transmitted for a single p-macroblock.

The motion compensation accuracy is in units of one quarter of the distance between luma samples. In case of the motion vector pointing to an integer sample position, the prediction signal consists of the corresponding samples of the reference picture. Otherwise, the corresponding sample is obtained using interpolation to generate non-integer sample positions.

Since coding order and display order are totally decoupled from each other, and in contrast to the previous coding standards, other pictures can reference B pictures for motion prediction, the only substantial difference between B and P slices is that B slices are coded in a manner, in which some macroblocks may use a weighted average of two distinct motion compensated prediction values for optimizing the prediction signal. In H.264, it is also possible to use multiple reference pictures for P pictures, but this step cannot be performed within the same macroblock. Note that in previous coding standards only the frame just before the frame to be predicted in coding order could be used for prediction of P pictures.

APPENDIX -B

OPERATIONAL RATE-DISTORTION FUNCTION

Consider a scalar quantizer followed by an entropy coder as the specific encoding system. The quantizer is completely defined by its quantization bins, the reproduction level for each bin, and the associated code words for each reproduction level. For this case, if we consider all quantization levels for a given system and source, one can define an operational distortion curve. This curve is obtained by designing the best encoder/decoder pair for each bit rate, and plotting the distortion obtained for the designed pair. The points in this curve are operational in the sense that they are directly achievable by the selected implementations and for the given set of test data. This approach is not very practical, since it is not possible to design a decoder/encoder pair for each rate, and also since the encoder should perform well for every decoder and not only a decoder that is specifically designed for it.

Operational Rate-Distortion (R-D) approach is particularly useful for the case, where there are a finite set of choices for the encoding parameters (as is the case for MPEG and JPEG). In this case, each R-D point corresponds to a specific combination of these parameters applied to each element in a particular set of data. An example operational R-D curve is given in Figure 19.

Operational R-D significantly simplifies the concerns for complexity of R-D approach. To solve the accurate data modeling dilemma, Transform Coding (which is used in today's video coding standards, such as DCT in MPEG-1/2/4) is utilized. Transform coding eases the process of modeling the data, since the signal is first decomposed into its frequency components and much simpler models can be used to model each of these frequency bands instead of trying to model the whole signal.

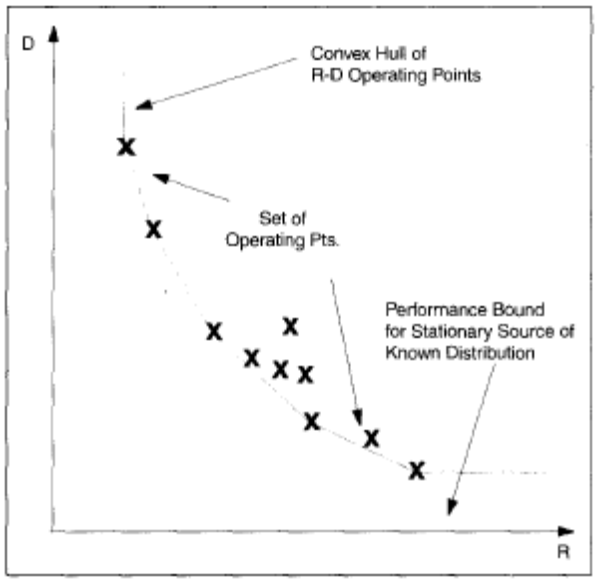

Figure 19 : The operational R-D curve [49]

B.1 R-D for Standards Based Video Coding

In all standards based applications, the encoder can select parameters that will result in various levels of R-D performance. This leads to a situation where the number of operating points is finite, and thus, the operational R-D bound is determined by the convex hull (Explained in Section 4.2.2) of the set of all operating points. In typical image and video coding process involving these standards, the encoding task of selecting the best operating point from a discrete set of operating points agreed upon a priori by a fixed decoding rule is referred to as *syntax constrained* optimization [49].

In this case, the R-D optimization is not concerned with obtaining the optimal operating point for all inputs characterized by a probability density or a particular data set. The task is confined to obtaining the best point for a particular input (that is the input to be encoded), given the constraints imposed by the coding framework.

There are some points that the encoder has to consider during the R-D optimization process. One of the first issues that need to be addressed is the selection of the coding unit on which the optimization is going to be performed. For instance, it is possible to consider video frames as the coding unit. If frames are chosen as the coding units, then frame-wise rate and distortion will be measured for each frame in the sequence, and optimal operating points will be chosen for each frame according to a cost function. It is also possible to choose finer coding units, such as slices or macroblocks.

Complexity is one of the other points that need to be considered, while performing R-D based optimization on images and video. There are two main sources of complexity associated with R-D optimization. First source is the extraction of R-D data from the content; several encode/decode operations may need to performed to obtain the R-D performance of coding alternatives. If multiple encode/decode cycles are unacceptably complex, models of R-D performances of coding alternatives could be employed, instead of determining the actual data. The second source of complexity comes from the search of the optimal point among the determined R-D points. Even if the R-D data is well known, the search for the best operating point is in itself a complicated task. The complexity brought forth by the R-D optimization can be justified, if the quality improvements are significant, especially since encoding is usually performed only once but decoding is done many times.

Another issue is the selection of the cost function. Both rate and distortion can be a part of the objective functions to be optimized. The objective functions can easily be computed for each coding unit, but determining the cost function needs additional consideration, when the problem further involves optimization for a set of coding units. For instance, the R-D optimization procedure might target minimizing the average MSE across all the coding units considered. However, for some scenarios, it is possible that average distortion can be less important than for example maximum distortion. Several alternatives have been proposed to determine an overall cost

function for a group of coding points; these functions include average MSE, minimax approaches, and approaches based on lexicographic optimization [49] .

B.2 Lagrangian Optimization

If the case, where rate and distortion can be measured independently for each coding unit is considered, that is the R-D data for coding unit *i* can be computed without requiring that other coding units to be encoded, then the rate distortion optimization can be formulated by using Lagrange multipliers as

$$min(\sum_{i=1}^{N} d_{ix(i)} + \lambda r_{ix(i)}) \ , \quad \text{subject to } \sum_{i=1}^{N} r_{ix(i)} < R_T \ , \tag{52}$$

where $x_{(i)}$ is the operating point of each coding unit *i*, d_i and r_i are the associated distortion and rate respectively. R_T is the total bit budget of the bit-stream .

For each coding unit, the point on the R-D curve that minimizes $d_{ix(i)} + \lambda r_{ix(i)}$, is that point at which the line having slope λ is tangent to the convex hull of the R-D characteristic. This is illustrated in Figure 20.

For the case where λ = 0, only the distortion is minimized disregarding the rate. Generalizing this result, it can be stated that: when using Lagrangian optimization, selection of a small λ indicates favoring low distortion over rate constraint and this causes the resultant operating point to have a high bit rate and low distortion, and choosing a large λ means favoring low bit rate over quality, causing the resultant operating point to have high distortion and low rate. This result can also easily be confirmed by investigating Figure 20 below.

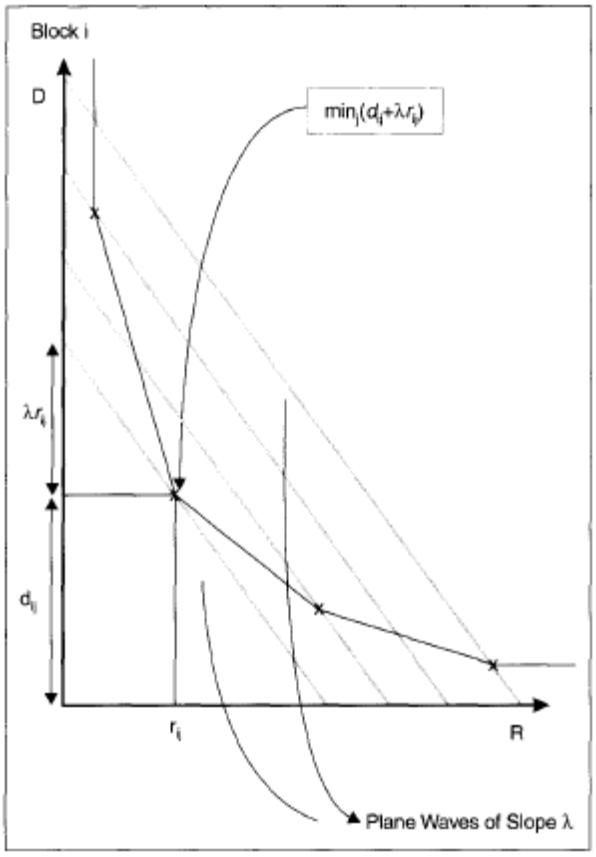

Figure 20 : For each coding unit, to minimize $d_{ix(i)} + \lambda r_{ix(i)}$ for a given λ is equivalent to finding the point in the R-D characteristic that is "hit" first by a "plane wave" of slope λ

The case, where λ is chosen to be the same for each coding unit is referred to as constant slope optimization. By choosing a constant λ, all coding units yield the same marginal return for an extra bit in the rate distortion trade off. This means that the reduction in MSE for using one extra bit for a given coding unit would be equal to the MSE increase incurred in using one less bit for another unit. This indeed is the only logical equilibrium point for the optimization algorithm, since if different λ's were chosen for different coding units, then we would need to allocate all the bits to the coding unit having the highest λ to obtain the optimal solution.

B.3 Lagrangian R-D Optimization for Encoding Decisions

R-D procedures can be used for making decisions at the encoder side. R-D optimization can be applied to bit rate control, motion estimation and INTRA/INTER/SKIP mode decisions.

Bit Rate Control

The overall bit rate of a video encoder is determined by its prediction mode decisions, MV choices and DFD coding fidelity. Particularly, the DFD fidelity, i.e. the fidelity of the residual, is the most important factor for bit rate control. DFD fidelity is controlled by choosing a quantization step size for the transformed difference signal. A larger step size results in a lower bit rate and a larger amount of distortion. Thus, the choice of step size is closely related to the choice of the relative emphasis to be placed on rate and distortion; i.e. λ. Control over λ in a well-optimized encoder can provide excellent means of bit rate control.

Motion Estimation

R-D optimization can be used to measure the rate-fidelity performance of candidate motion vectors during motion vector search. The criterion for a particular motion vector is the minimization of a Lagrangian cost function wherein the distortion, represented as the prediction error is Sum of Squared Differences (SSD) or Sum of Absolute Differences (SAD), is weighted against the number of bits associated with the motion vector using a Lagrange multiplier.

Motion estimation can therefore be viewed as the minimization of the Lagrangian cost function

$$J_{MOTION} = D_{DFD} + \lambda_{MOTION} R_{MOTION} \ . \qquad (53)$$

Motion vector block size can also be chosen using R-D optimization. In [52], motion estimation is performed by minimizing J_{MOTION} in the above equation. In the first part

of the procedure, an integer pixel accurate displacement vector was obtained within a search range -15,+15. Then, given this vector, its surrounding half pixel positions were checked for improvement by evaluating the above R-D equation. Then, the impact of motion compensation block size on coding performance was tested. For the test, 3 different cases were considered for 16x16 macroblock prediction modes.

Case 1: Inter coding with 1 motion vector for each block (INTER)
Case 2: Inter coding with 4 motion vectors for each block (INTER +4V mode)
Case 3: Selectively picking Case 1 or Case 2 for each macroblock using rate constrained Lagrange optimization.

Case 1 can achieve better prediction than Case 2 at the lowest bit rates, since it can represent a moving area with one fourth as many motion vectors. However, Case 2 achieves better performance at higher bit rates since it can represent finer motion detail. Case 3 can adaptively choose the proper block size as needed, so it obtains the best prediction at virtually all the bit rates. Case 1 has better prediction than Case 3 at the lowest rates, since it does not require the extra bit per macroblock to distinguish between the two modes.

INTRA/INTER/SKIP Mode Decision

R-D optimization can also be used for the macroblock mode decision. If we consider the various macroblock modes for H.263 INTRA, SKIP, INTER, INTER+4V (Inter prediction with 4 motion vectors per block) and if we further assume that the bit rate and the distortion of the residual coding stage is controlled by the selection of a quantizer step size Q, then the rate distortion optimized mode decision refers to the minimization of the following Lagrangian function.

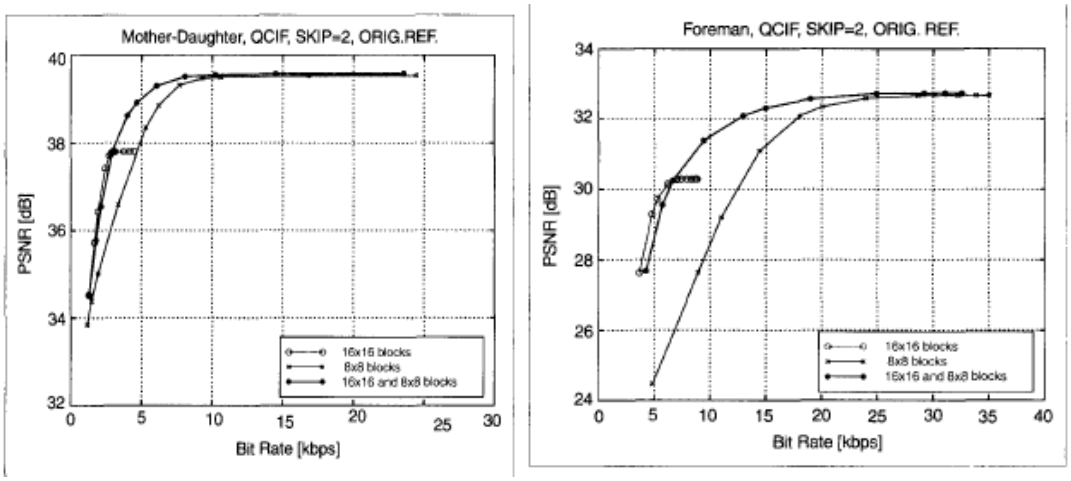

Figure 21 : PSNR vs Bit Rate spent on motion vectors for the 3 different macroblock size modes [50]

$$J(A, M, Q) = D_{REC}(A, M, Q) + \lambda_{MODE} R_{REC}(A, M, Q) \ , \qquad (54)$$

where M ϵ { INTRA, SKIP, INTER, INTER+4V} indicates a mode chosen for a particular macroblock, Q is the selected quantizer step size, D_{REC}(A,M,Q) is the SSD between the original macroblock A and its reconstruction, and RREC(A,M,Q) is the number of bits associated with choosing M and Q.

REFERENCES

[1] FP-7 ICT Next Media, Future Media Internet Coordination action, "Deliverable D2.1 Report on current research and business targets V1.0",2010.

[2] Y. Neuvo , J. Yrjanainen, "Wireless Meets Multimedia – New Products and Services," Proceedings of IEEE, ICIP, 2002.

[3] S. F. Chang, Anthony Vetro, "Video Adaptation: Concepts, Technologies and Open Issues," Proceedings of the IEEE, vol. 93, no. 1, Jan. 2005.

[4] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[5] M. Pinson and S. Wolf. "A New Standardized Method for Objectively Measuring Video Quality," IEEE Transactions on Broadcasting, vol. 50, no. 3, pp. 312-322, Sep. 2004.

[6] A. K. Moorthy, K. Seshadrinathan, R. Soundararajan, A. C. Bovik, "Wireless Video Quality Assessment: A Study of Subjective Scores and Objective Algorithms," IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 4, Apr. 2010.

[7] P. Bocheck, Y. Nakajima and S.-F. Chang, "Real-time Estimation of Subjective Utility Functions for MPEG-4 Video Objects," Proceedings of IEEE Packet Video Workshop (PV'99), Apr. 1999 New York, U.S.A.

[8] S. F. Chang, "Optimal Video Adaptation and Skimming Using a Utility-Based Framework," Tyrrhenian International Workshop on Digital Communications Sep. 2002, Capri Island, Italy.

[9] Y. Wang, J.-G. Kim, S.-F. Chang, "Content-Based Utility Function Prediction For Real-Time MPEG-4 Video Transcoding," Proceedings of IEEE, ICIP 2003, Barcelona Spain.

[10] Ö. D. Önür, A. A. Alatan, "Optimal Video Adaptation for Resource Constrained Mobile Devices Based on Utility Theory," WIAMIS 2004, Portugal.

[11] A. L. Golub, *Decision Analysis: An Integrated Approach*, John Wiley and Sons Inc, 1997.

[12] Y. Wang, M. van der Schaar, S. F. Chang and A. C. Loui, “Classification Based Multidimensional Adaptation Predoction for Scalable Video Coding Using Subjective Quality Evaluation,” IEEE Transactions On Circuits And Systems For Video Technology, vol. 15, no. 10, pp. 1270-1279, 2005.

[13] S. J. Choi and J. W. Woods, “Motion Compensated 3D subband coding of video,” IEEE Transactions on Image Processing, vol. 8, no. 2, pp. 155-167, Feb. 1999.

[14] FP-7 ICT - Distribution Of Multi-view Entertainment using content aware Delivery Systems, “D2.2 Interim reference system architecture report,” 2010.

[15] FP-7 ICT - Seamless Content Delivery, “D2.2 Interim reference system architecture report,” 2010.

[16] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, “Video Coding with H.264/AVC: Tools, Performance and Complexity,” IEEE Circuits and Systems Magazine, pp. 7-29, First Quarter 2004.

[17] W. Yuan, K. Nahrstedt, S. V. Adve, D. L. Jones, R. H. Kravets, “GRACE-1:Cross layer Adaptation for Multimedia Quality and Battery Energy,” IEEE Transactions on Mobile Computing vol. 5, no. 7, Jul. 2006.

[18] M. Horowitz, A. Joch, F. Kossentini,” H.264/AVC Baseline Profile Decoder Complexity Analysis,” IEEE Transactions On Circuits And Systems For Video Technology, vol. 13, no. 7, pp. 704-716, Jul. 2003.

[19] S. Winkler, P. Mohandas, “The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics,” IEEE Transactıons On Broadcastıng, vol. 54, no. 3, Sep. 2008.

[20] N. Kontorinis, Y. Andreopoulos, M. v.d. Schaar “Statistical complexity framework for video decoding complexity modeling and prediction,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 7, pp. 1000-1013, JULY 2009.

[21] M. Van der Schaar, Y. Andreopoulos, “Rate-Distortion-Complexity Modeling for Network and Receiver Aware Adaptation,” IEEE Transactions on Multimedia, vol. 7, no. 3, Jun. 2005.

[22] Y. Liang, I Ahmad, “Power and distortion optimization for pervasive video coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 10, pp. 1436-1447, Oct. 2009.

[23] H.-M. Hang and J.-J. Chen, “Source model for transform video code and its application, part I and II,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 2, pp. 287–311, Apr. 1997.

[24] Y. Wang, J.G. Kim, S.F. Chang, H.M. Kim, “Utility-Based Video Adaptation for Universal Multimedia Access (UMA) and Content-Based Utility Function Prediction for Real Time Video Transcoding,” IEEE Transactions on Multimedia vol. 9, no. 2, Feb. 2007.

[25] A. Eleftheriadis, *Dynamic Rate Shaping of Compressed Digital Video*, Ph.D. dissertation, Graduate School of Arts and Sciences, Columbia Univ., New York, 1995.

[26] E. C. Reed and J. S. Lim, “Optimal multidimensional bit-rate control for video communications,” IEEE Transactions on Image Processing, vol. 11, no. 8, pp. 873–885, Aug. 2002.

[27] A. Vetro, Y. Wang, and H. Sun, “Rate-distortion optimized video coding considering frameskip,” in Proceedings of IEEE, ICIP, pp. 534–537, Oct. 7–10, 2001, Vancouver, BC, Canada.

[28] P. Yin, M. Wu, and B. Liu, “Video transcoding by reducing spatial resolution Proceedings of IEEE, ICIP, pp. 972–975, Sep. 10–13, 2000.

[29] Y. Wang, J.-G. Kim, and S.-F Chang, “MPEG-4 Real Time FD-CD Transcoding,” Columbia Univ. DVMM Group, Tech. Rep. 208-2005-2, 2003,New York.

[30] H.-M. Hang and J.-J. Chen, “Source model for transform video code and its application, part I and II,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 2, pp. 287–311, Apr. 1997.

[31] E. C. Reed and J. S. Lim, “Optimal multidimensional bit-rate control for video communications,” IEEE Transactions on Image Process., vol. 11, no. 8, pp. 873–885, Aug. 2002.

[32] Recommendation ITU-R BT.500-11 “Methodology for the Subjective Assessment of the Quality of TV Pictures,” 2002.

[33] Recommendation ITU-R P910. Subjective Video Quality Assessment Methods for Multimedia Applications,” 1999.

[34] Ö.D.Önür, A.A. Alatan “Video Adaptation Based on Content Characteristics and Hardware Capabilities” chapter of Advances in Semantic Media Adaptation and Personalization SMAP, 15 Jan. 2009, Auerbach.

[35] B. Girod, “What’s wrong with mean-squared error,” Digital Images and Human Vision, A. B. Watson, Ed. 1993, pp. 207–220, MIT Press , Cambridge, MA.

[36] G. Cermak, M. Pinson, and S. Wolf, “The Relationship Among Video Quality, Screen Resolution, and Bit-rate,” IEEE Transactions On Broadcasting, vol. 57, no. 2, Jun. 2011.

[37] Video Quality Experts Group (VQEG), “Final Report from the Video Quality Experts Group on the Validation of Objective Quality Metrics for Video Quality Assessment,” 2000, Available: http://www.its.bldrdoc.gov/vqeg/projects/frtv phase I, last accessed on 01/09/2011.

[38] Video Quality Experts Group (VQEG), “Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment, phase II,” 2003, Available: www.vqeg.org, last accessed on 08/09/2011, last accessed on 01/09/2011.

[39] S. Wolf and M. Pinson, “Video quality measurement techniques,” NTIA Report 02-392, Jun. 2002. Available: www.its.bldrdoc.gov/pub/n3/video/index.php, last accessed on 08/09/2011.

[40] ANSI T1.801.03 – 2003, “American National Standard for Telecommunications – Digital transport of one-way video signals – Parameters for objective performance assessment,” American National Standards Institute, 2003.

[41] Preliminary Draft New Recommendation “Objective perceptual video quality measurement techniques for digital broadcast television in the presence of a full reference,” Recommendations of the ITU, Radio communication Sector.

[42] Draft Revised Recommendation J.144, “Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference,” Recommendations of the ITU, Telecommunication Standardization Sector.

[43] JSVM Software Manual, JSVM 9.8, Joint Video Team 2007.

[44] VQM Metric, Institute for Telecommunication Sciences, Boulder Colorado Available: http://www.its.bldrdoc.gov/vqm/, last accessed on 03/09/2011

[45] VTUNE by Intel Corporation, Available: http://software.intel.com/en-us/articles/intel-vtune-amplifier-xe/, last accessed on 03/10/2011

[46] Real View Development Suite Arm Corporation, Available: http://www.arm.com/products/tools/software-tools/index.php, last accessed 01/09/2011.

[47] D. W. Scott, S. R. Sain, “Multi-Dimensional Density Estimation”, Elsevier Science Aug. 2004.

[48] Schwarz, Gideon E., "Estimating the dimension of a model," Annals of Statistics vol. 6, no.2, pp. 461–464, 1978.

[49] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video Compression: An Overview,” IEEE Signal Processing Magazine, pp. 23-50, November 1998.

[50] G. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” IEEE Signal Processing Magazine, pp. 74-90, November 1998.

[51] D. M. Sow, A. Eleftheriadis, “Complexity Distortion Theory,” IEEE Transactions on Information Theory, vol. 49, no. 3, Mar. 2003.

[52] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973, New York.

[53] H. Stark, J.W. Woods, *Probability and Random Processes with Applications to Signal Processing*, Prentice Hall, 2002, New Jersey.

[54] T. Wiegand, G. J. Sullivan, G. Bjontegaard and Ajay Luthra, “Overview of the H.264/AVC Video Coding Standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13 no.7 July 2003.

VITA

Özgür Deniz Önür was born in Ankara, Turkey, in 1979. He received his B.S. and M.S degrees in Electrical and Electronics Engineering, from Middle East Technical University, Ankara, Turkey in 2001 and 2003, respectively. He has worked as a senior researcher for The Scientific and Technological Research Council of Turkey from 2001 to 2005. Since then, he is a partner in Mobilus Ltd. His research interests include video adaptation, video quality modeling, video decoding complexity estimation.

His publications are as follows:

Book Chapter

- Ö.D.Önür, A.A. Alatan "Video Adaptation Based on Content Characteristics and Hardware Capabilities" Chapter of Advances in Semantic Media Adaptation and Personalization SMAP, 15 Jan 2009, Auerbach.

Pending Patent

- Optimal video adaptation for resource constrained mobile devices based on subjective utility models. onur ozgur deniz (tr); alatan a aydin (tr) patent number : WO2006126974.

Conference Papers

- Özgür D. Önür, A. Aydın Alatan, "A Complexity-Utility Framework Utilizing Statistical Video Decoding Complexity Prediction For Mobile Devices", IEEE MQoE 2011, USA.
- Özgür D. Önür, A. Aydın Alatan, "A Complexity-Utility Framework for Modeling Decoding Complexity Towards Optimizing Subjective Quality of

Video for Mobile Devices.” ACM Multimedia 2010 Workshop - Mobile Video Delivery (MoViD), 2010, Rome, Italy.

- Özgür Deniz Önür, A. Aydın Alatan, “Video Adaptation Based on Content Characteristics and Hardware Capabilities.” "SMAP 2007", London, UK.
- Ö. D. Önür, A. A. Alatan, " Video Adaptation for Transmission Channels by Utility Modelling "IEEE ICME 2005, Amsterdam Holland.
- Ö. D. Önür, A. A. Alatan, " Optimal Video Adaptation for Resource Constrained Mobile Devices Based on Utility Theory, " WIAMIS 2004, Portugal.
- Ö. D. Önür, A. A. Alatan,"Optimal Video Adaptation Using Subjective Utility Functions," Cost 276 Workshop 2004, Ankara Turkey.
- Ö.D.Önür, P. Drege, A. Perkis, A. A. Alatan, Runar Solberg" Delivering Adapted Content to Terminals Using MPEG-21 Digital Items", Cost 276 Workshop 2004, Ankara Turkey.