# Multiple Objective Optimization for Video Streaming

by

Tanır Özçelebi

A Dissertation Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Electrical & Electronics Engineering

Koç University

December, 2006

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a Ph.D. thesis by

Tanır Özçelebi

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. A. Murat Tekalp

_____

Assist. Prof. M. Oğuz Sunay

_____

Assist. Prof. Yücel Yemez

_____

Prof. M. Reha Civanlar

_____

Prof. Bülent Sankur

Date: _____

*To my parents*

## ABSTRACT

In this thesis, we propose Multiple Objective Optimization (MOO) frameworks for efficient video streaming.

Firstly, we introduce pre-roll delay-distortion optimization (DDO) for uninterrupted content-adaptive video streaming over low capacity, constant bitrate (CBR) channels using MOO. Content analysis is used to divide the input video into shots with assigned relevance levels. The video is adaptively encoded and streamed aiming minimum pre-roll delay and distortion with the optimal spatial and temporal resolutions and quantization parameters for each shot. With buffer and distortion constraints, the bitrate of unimportant shots is reduced to achieve an acceptable quality in important shots.

Secondly, we introduce a cross-layer optimized video rate adaptation and scheduling scheme to achieve maximum "application layer" Quality-of-Service (QoS), maximum video throughput (video seconds per transmission slot), and QoS fairness for wireless video streaming. Using the MOO framework, these objectives are jointly optimized such that the user with i) the least remaining playback time, ii) highest available video throughput and iii) maximum video quality is served.

Finally, we propose an adaptive framework for compression and streaming of stereo video using the existing network infrastructure. We employ content-adaptive stereo video coding (CA-SC), where additional compression is achieved by spatial and/or temporal downsampling depending on the content. An end-to-end streaming system where the end-users can view the video in mono or stereo mode depending on their display capabilities is implemented and MOO formulations are proposed.

The improvements achieved are demonstrated with experimental results.

# ÖZETÇE

Bu tez raporunda verimli video akıtımı için Çok Hedef-İşlevli Eniyileme (MOO) şemaları sunulmaktadır.

İlk olarak, düşük ve sabit kapasiteli ağlarda kesintisiz içerik uyarlamalı video akımı için gecikme-bozunum eniyilemesi metodu sunulmaktadır. Giriş videosu içerik analizi yapılarak çeşitli ilgililik seviyelerine ayrıştırılmaktadır. Video en küçük gecikme ve bozunum hedeflenerek uyarlamalı olarak her sahne için en iyi uzaysal ve zamansal çözünürlük ve nicemlemeyle kodlanmakta ve akıtılmaktadır. Arabellek ve bozunum sınırlamalarıyla birlikte, önemsiz sahnelerin bit hızı düşürülmekte ve önemli kısımların kalitesi arttırılmaktadır.

Sonra en yüksek "uygulama katmanı servis kalitesinde", en yüksek video kapasitesinde (zaman sekmesi başına video saniyesi) ve servis kalitesinde adil kablosuz video akıtımı için çapraz-katmanlı eniyilenmiş bir video bit hızı uyarlama ve kullanıcı çizelgeleme şeması sunulmaktadır. Bu hedefler her zaman sekmesinde i) en küçük oynatma zamanına, ii) en yüksek video kapasitesine ve iii) en yüksek video kalitesine ulaşan kullanıcı seçilerek MOO ile eniyilenmektedir.

Son olarak, var olan ağlarda stereo videolarin kodlanması ve akıtımı için uyarlamalı bir metod önerilmektedir. Fazladan sıkıştırmanın uzaysal ve zamansal ölçeklemeyle sağlandığı içerik uyarlamalı stereo video kodlama uygulanmaktadır. Kullanıcıların gösterim imkanları dahilinde mono veya stereo video izleyebildikleri uçtan uca bir akıtım sistemi tanıtılmakta, MOO problem formüllemeleri önerilmektedir.

Ulaşılan kazanımlar deneysel sonuçlarla gösterilmektedir.

## ACKNOWLEDGMENTS

I would like to thank my thesis advisors Prof. A. Murat Tekalp and Prof. M. Reha Civanlar who have been a great source of inspiration and provided me with constructive criticism and research freedom during my studies. I am grateful for the critical reading of this thesis by the members of my thesis committee and for their valuable comments. Finally, I would like to thank my family for providing me the morale support that helped me in hard days of my research.

This thesis has been supported by The Scientific and Technological Research Council of Turkey (TÜBITAK) and by European Commission within FP6 under Grant 511568 with the acronym 3DTV.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## NOMENCLATURE

| | |
|---|---|
| MOO | Multiple Objective Optimization |
| DDO | Delay-Distortion Optimization |
| CBR | Constant Bitrate |
| VBR | Variable Bitrate |
| QoS | Quality of Service |
| CA-CS | Content Adaptive Stereo Video Coding |
| HVS | Human Visual System |
| AVC | Advanced Video Coding |
| GoP | Group of Pictures |
| RDO | Rate-Distortion Optimization |
| UMA | Universal Multimedia Access |
| VBV | Video Buffer Verifier |
| HRD | Hypothetical Reference Decoder |
| AMP | Adaptive Media Playout |
| PSNR | Peak Signal to Noise Ratio |
| LP | Linear Programming |
| HDR | High Data Rate |
| 1xEV-DO | 1x Evolution-Data Optimized |
| SVC | Scalable Video Coding |
| FEC | Forward Error Correction |
| ARQ | Automatic Repeat reQuest |
| MVC | Multiview Video Coding |
| SDP | Session Description Protocol |
| VCL | Video Coding Layer |
| NAL | Network Abstraction Layer |
| RTP | Real-time Transport Protocol |

# Chapter 1

# BACKGROUND AND MOTIVATION

Over the last few decades, the efforts for finding an efficient digital representation for video and communicating it over a network have gained a lot of interest from the society. The communication networks such as the Internet have become much faster than they used to be due to the tremendous developments in the physical backbone, especially in the developed countries. This progress has made new, more advanced and more interesting services possible for service providers resulting in higher user satisfaction. Among these new services, live and on-demand video streaming services such as Internet TV, video conference, and video databases like YouTube [1] and Google Video [2] are becoming more and more popular. The quality of video experience achieved by different service providers constitutes the cutting edge for their market share. The perceived video quality and the service speed are the determining factors in doing such a comparison between different service providers for users.

The perceived video quality can be defined as the closeness of the compressed, transmitted and decompressed video content to the original video sequence in terms of their spatial features (picture quality) and temporal features (fluent and uninterrupted play) visible to the naked eye. Ideally, the spatial and temporal features that are perceivable by the Human Visual System (HVS) would be exactly the same for the original and the reproduced video sequences. However, the video quality may be degraded in two phases in such services, namely; quality degradation while i) video coding/compression, and ii) video streaming. The former can be defined as the science of representing videos with the least amount of information (bits stored) and the maximum visual quality, and it is still a very hot research topic presently. The latter tries to satisfy the constraints of the communication channel and the end devices while dealing with packet losses and bit errors that occur in the physical layer.

Traditionally, these two issues have been approached somewhat independently from each other in order to make the system design and upgrade easier, which came at the expense of suboptimality in the achieved video experience. The video encoding algorithms in the literature show very little or no awareness to the communication environmental issues in general, except for one specific case, i.e. when the communication channel has Quality-of-Service (QoS) guarantees such as *constant* and *sufficient* bandwidth support. In this exceptional case, the encoder at the server side can simulate a replica of the decoder at the receiving client device and prevent unwanted pauses in the video playback. On the other hand, such network-layer QoS and bandwidth is very difficult to achieve at the same time in today's communication networks especially considering wireless environments and their shared nature. If the network-layer quality of service and/or the high bandwidth requirements of the video streaming service can not be provided, there is still a higher level trick the service provider can employ in order to improve the user experience, i.e. optimized adaptive temporal video rate (quality) allocation. In this thesis, we propose Multiple Objective Optimized (MOO) video streaming system designs that consider both temporal video rate allocation and transmission issues simultaneously.

In this chapter, firstly, background information about the state-of-the-art compressed video representation and temporal rate allocation (control) techniques and the evolution towards multiview video representations are presented. Secondly, the motivation for content adaptive video rate adaptation is given. Finally, the streaming issues known to both monocular and multiview video representation cases in the literature are discussed.

## *1.1 Video Compression and Temporal Rate Allocation*

The video compression techniques in the literature can be divided mainly into two categories, namely lossless and lossy video compression. In lossless video coding algorithms, the compressed and recovered video bitstream is the bit-by-bit identical of the original input video data. This is the ideal case when the video perceptual quality point of view is considered alone. On the other hand, the compression efficiency of such video coding techniques is quite low and they are impractical for most applications. Therefore, the application area of lossless video coding is very limited (e.g. video archiving). On the other hand, the lossy video encoders intentionally discard data in order to achieve much lower bit rates. As

a consequence, the video quality drops and there is a trade-off between the video perceptual quality and the compression efficiency that needs to be optimized, which in general is called the rate-distortion trade-off. Today, the popular lossy video compression techniques are able to achieve much higher compression rates than that of lossless encoders with very little degradation (unnoticeable in some cases) in the overall perceptual video quality, making them more suitable for streaming purposes. The typical methods to achieve this goal in lossy video coding are to take advantage of the Human Visual System (HVS) characteristics, image statistics, spatial correlations within a frame, temporal correlations between consecutive frames and information theory (entropy coding).

### *1.1.1 Monocular Video*

The state-of-the-art monocular lossy video codecs (encoder-decoder pairs) such as MPEG2, H.263 and MPEG4 all follow the same philosophy for achieving high compression rates and they have been quite successful indeed. The recent and the most advanced video coding standard up-to date is the H.264 MPEG-4 Part 10 or also known as the Advanced Video Coding (AVC) standard, which was introduced by the Joint Video Team (JVT). The JVT Group was founded as a partnership of the well known ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The H.264/AVC is a more advanced standard than its ancestors MPEG2, H.263 and MPEG4 Part 2 in that, it uses more sophisticated methods for motion estimation, compensation, block matching etc., supporting high perceptual quality at low bit rates. The overall computational complexity is higher, but this is bearable with modern CPU's and chips in case of hardware implementation. Although the H.264/AVC standard codec and its modified versions is widely used throughout this thesis, we will not dive into the standard specifications in high detail as it has already been well documented by the standardization bodies.

Although the level of compression rates achieved with today's encoders is sufficient for most of the personal storage related applications, extreme caution is still required for commercial and industrial storage purposes, and more importantly streaming services. Having an effective temporal bitrate allocation (control) algorithm has long been recognized as one of the main product differentiators between various video encoders. The shared and unstable nature of communication channels, and the temporal variations in the scene complexity

(coding difficulty) of videos make proper temporal video bitrate allocation an interesting research problem. Rate control algorithms consist of group-of-pictures (GoP)-level and frame-level bit allocation strategies and an encoder buffer fullness control strategy. In this thesis, we will mostly concentrate the codec parts that are directly related to the video streaming task, e.g. rate control (buffer management) and adaptivity issues.

The size of the decoder buffer at the receiving side of a video streaming system must be considered while encoding at the transmitting side as it puts an upper limit on the initial pre-roll delay. On one hand, if the pre-roll delay is kept too long, the receiving buffer may overflow, causing the received video packets to be dropped before they are put in the buffer queue. On the other hand, the receiving buffer may as well get empty during video play if this delay is kept too short, causing the video playback to pause temporarily. Note that both of these situations are extremely unwanted as they cause inefficiency in both the video experience and the network utilization. In order to prevent this, the received buffer is generally modeled and simulated at the transmitting side *while encoding* in the state-of-the-art video codecs [3, 4] *assuming a CBR channel behavior* with a specified channel capacity. Hence, if the given assumption is correct, it can be verified that the bitstream generated by the encoder can be played without interruptions at the receiving side.

Clearly, the larger the decoder buffer, the more the encoder has freedom to allocate bits across frames and GoPs for better performance at the expense of larger pre-roll delay. On the extreme case, with no constraint on the buffer size, the entire video can be downloaded before starting playback, which corresponds to the maximum pre-roll delay. In this thesis, we present a multiple objective optimization (MOO) framework and a linear programming based solution for the optimal allocation of bits across semantically defined GoP's (shots) to obtain a tradeoff between maximum visual quality and minimum pre-roll delay for a CBR channel under certain constraints.

State of the art encoders perform operational rate-distortion optimization (RDO) for mode selection, which can be considered as macro-block (MB)-level bit allocation. For example, within the H.264/AVC video coding standard [5, 4], Lagrangian optimization is used for determining the optimal encoder modes (Intra/Inter/Skip mode decision) and quantization levels for different parts of a video for a given rate with minimal overall distortion [6]. In [7], the optimal value of the Lagrange parameter is found by determining an approximate

rate-distortion (RD) curve and then differentiating the distortion with respect to the coding rate. A rate control algorithm should consider absorbing instantaneous changes in the encoding rate of the source video for transmission over a constant bitrate (CBR) channel using limited size buffers. Encoder buffer fullness control is achieved by adjusting the quantization parameter to avoid buffer overflow and underflow, without considering resulting distortion, e.g., the leaky-bucket method [8, 9].

### *1.1.2 Multiview Video*

It is crucial for visual communication to be realistic as far as the user is concerned. Single view video technologies have made a lot of progress since the invention of the black-and-white television as explained above. As the color depth and spatial resolution is increased, the video is perceived as more and more realistic by the Human Visual System. In order to improve the level of realism, it is reasonable to add depth information to the video. Depending on general understanding of object shapes in nature, perspective, occlusion, shading and more 3D clues, a human being can extract some depth information from monocular videos. However, this is indeed not enough for getting an exact three-dimensional (3D) feeling. To extract full depth information, HVS requires that the scene to be observed is simultaneously viewed from at least two different positions (angles) in three-dimensional coordinates. This is why a person with one eye shut has difficulty discriminating between depths of different objects in 3D space.

Three-dimensional video technology, although unfamiliar to most people around the world, is certainly not a new technology. Its roots go back to almost a century ago. The first 3D movie called "Power of Love" was first shown at the Ambassador Hotel in Los Angeles in September 1922. In this movie, the anaglyph technology was used where a couple of cameras are used in parallel to take slightly shifted versions of the same scene. The frames from both cameras are printed in two different colors, overlapped and then viewed using special glasses with different color filters on each eye. In November 1952, "Bwana Devil" was the first 3D movie of Hollywood to be shown all around the United States. The movie was a huge economic success for the producers, although it was shown to be one of the worst theatrical movies ever by many film authorities. Then comes the critical question: What was the reason that three dimensional movies have not become more popular over the years.

The answer is that the registration process of the anaglyph frames are not identical to that done by the human eyes and brain, and it causes headaches and sickness in stomach after 30 minutes on average. Another drawback of stereoscopic video has been the necessity of using stereo glasses to watch it. However, the recent achievements 3D display technologies such as lenticular monitors avoid such a requirement and will definitely give an acceleration to the studies in this field. The details of these issues are beyond the scope of this thesis. It is enough to state that the 3D technology will become more popular in the coming years for now.

Knowing that the future of the multiview technology is safe in terms of user satisfaction/demand, the researchers from around the world concentrate on possible applications and their quality attributes. It is straightforward to see that multiview video streaming is the next generation of today's video streaming applications and they will require efficient compression without sacrificing much from visual quality. Considering that, as the number of view angles (cameras) are increased for a particular multiview scene, the overall encoding rate would increase which would make life more difficult for us. On the other hand, this situation is not unbearable as long as some conditions are satisfied. Note that, all efficient monocular video compression technologies such as JPEG2000, MPEG2 and H.264/AVC make use of the high spatial and temporal correlation within the video in order to achieve high compression without losing much from visual quality [10, 11]. For the case of stereo video coding, in general, stereo image pairs are taken by two cameras standing close to each other and facing similar directions, and this causes the stereopairs to include a lot of common objects. Most of the time, the background is also the same for the left and right images as shown in Figure 1.1. For this reason, there is a huge amount of correlation between the left and right images in a stereopair and this correlation should also be used for further compression and a similar methodology can be applied for higher number of cameras. Therefore, directly applying monocular video coding methods to a stereo scene (multi-view in general) is suboptimal. In order to achieve optimality, all view angles should be considered simultaneously instead of one by one.

A predictive video encoding system contains several stages such as motion estimation and compensation, disparity estimation and compensation in the case of stereo coding, transform, quantization and, finally, entropy coding. Therefore, the overall performance of

Figure 1.1: An example stereo pair (Tsukuba from the Middlebury College Stereo Vision Page) including many common objects and a common background.

the system can be improved by adjusting a limited number of variables used in the above mentioned stages. It is observed that, especially in stereo video coding, the better the displacement (disparity) prediction is, the more efficient the overall compression rate becomes. Furthermore, using bit allocation and quantization strategies that are optimal for stereo coding instead of applying monocular encoding on left and right images separately would increase the overall system performance considerably. Coding left and right image sequences separately by using single-view video coding without considering binocular redundancies is trivial and suboptimal. Just like the ordinary monocular coding strategies of today, high compression rates can be achieved for stereo coding by making use of the temporal, spatial and binocular redundancies in the source video.

As opposed to the general idea, the bit rates of an equivalent quality (using the same encoding parameters) stereo video and the single-view version of it do not differ drastically, as all modern coding techniques depend on eliminating redundancy in the video and stereo-pairs usually possess strong correlation amongst themselves. Typically, two views cost only $1.2 \sim 1.3$ times the number of bits used by only a single view.

*Rate Distortion Optimization Among Stereopairs*

Several stereo image/video coding schemes have been introduced in the literature [12, 13, 14, 15, 16, 17, 18]. The schemes proposed in [15] consist of two main parts. The first part is the efficient estimation and compensation of disparity vectors, and the second part is an appropriate optimal bit allocation strategy.

Let the rate-distortion optimization problem between stereopairs be as explained in [15, 18]. There are many techniques to determine the disparity vector field between two images. One of the well known techniques is the *Iterative Lucas-Kanade Optical Flow Algorithm* [19]. The details of a pyramidal implementation of the Lucas-Kanade algorithm is given in [20] and explained in Appendix D. However, the main purpose of video coding is to compress the video. Therefore, using very accurate methods for disparity estimation is not our primary concern.

Block-based disparity estimation and compensation can be done as in Figure 1.2. The blocks in the target frame are searched for within the reference frame. The disparity estimation/compensation block calculates the disparity vector field (DV) and disparity compensated difference (DCD) frame between the encoded and decoded version of the reference image and the target image. After that, the resulting disparity vector field and disparity compensated difference frames are encoded according to the rate-distortion policy used.

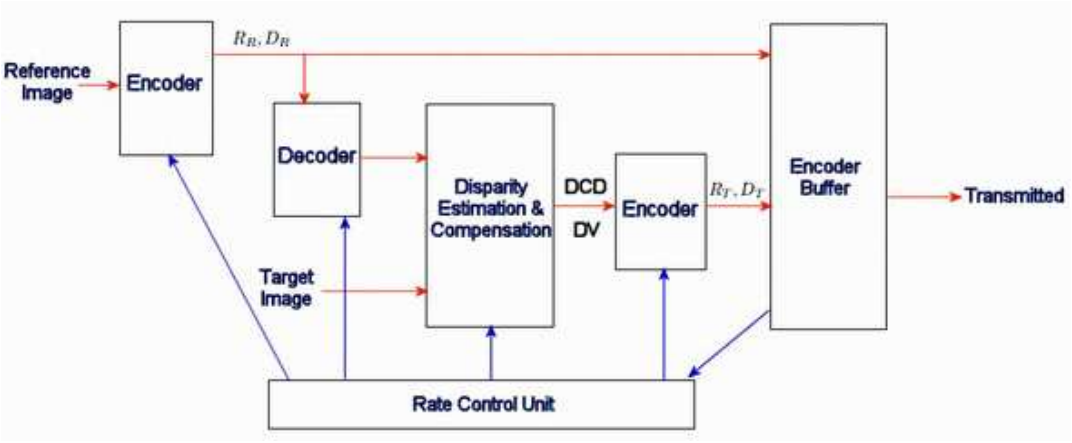



Figure 1.2: Encoder block diagram for stereo video.

If the focal rays of the stereo camera pair are parallel and they are orthogonal to

the stereo baseline, i.e. if they satisfy the epipolar constraint, then all the motion vectors (disparity vector estimates) between left and right images have to be horizontal and parallel to each other. This is obviously a special case of the motion estimation procedure and it can lead to more accurate disparity vector estimations.

Simultaneously assigning bits to an image pair from a common bit budget is called *dependent bit allocation*. The details of dependent bit allocation are explained in [17]. For finding the optimal bit allocation strategy, the total distortion measure, which can be defined in a variety of ways, has to be minimized while staying within the overall bit budget supplied for both left and right images.

Let the left image, $F_R$ of the stereopair be the reference image and the right image, $F_T$ be the target image. Of course, the selection of the reference and target frames could be the other way around as well. Here, the reference frame $F_R$ is used to estimate and compensate the target frame $F_T$. The disparity vector field (DV) between the reference image and the target image is computed. Afterwards, the computed disparity vector field is used to find an estimate for the target image from the reference image. The difference of the original target frame, $F_T$, from the disparity estimated and compensated version, $\hat{F}_T$ is called the disparity compensated difference (DCD) frame. Let the available number of bits for the whole stereopair be $R_{total}$. Then the rate-distortion optimization formulation can be stated as the one which tries to minimize the overall distortion measure ($D$) under the bit budget constraint.

$$\min_{(DV,Q_R,Q_T)} \{D\} \tag{1.1}$$

subject to

$$R_R + R_T \leq R_{total}$$

where $Q_R$ and $Q_T$ denote the quantization parameters used in the reference and the target frames respectively. DV is the resulting disparity vector field, and $R_R$ and $R_T$ are the corresponding bit rates of the reference and the target frames, respectively.

*Perceptual Quality of Stereo Video*

There are two different theories about the effects of unequal bit allocation between left and right images, namely *fusion theory* and *suppression theory*. In fusion theory, it is believed that the stereo distribution must be equally made for the best human perception. On the other hand, in suppression theory, it is believed that the highest quality image in the stereopair determines the overall perception performance. Therefore, according to this theory, we can compress the target image as much as possible to save bits for the reference image, so that the overall distortion is the lowest. If we assume that the overall distortion measure of a stereopair will be a weighted average of the individual images, we can define weighting coefficients between right and left image distortion values to take different amount of contributions from each picture into account.

## *1.2 Video Content Analysis and Adaptive Rate Allocation*

In classical rate control, all GoPs are treated equally, and frame-level bit allocation is based on the frame type and a complexity measure, sometimes with multiple passes over the video, but without considering the semantics of picture content. In the H.264/AVC reference encoder [5], the GoP borders are determined according to a predefined pattern of frames, and the same target bitrate is used for each GoP given the available channel rate. As a result, the video quality varies from GoP to GoP depending on the video content. The problem with this approach is that in some applications, e.g., wireless video, the total bit budget is not sufficient to encode the entire content at an acceptable quality. Video segments with high motion and/or small details may become unacceptable when all GoPs are encoded at the same low rate. Inter-GOP rate control schemes, that is variation of the target bitrate from GoP-to-GoP, have been proposed to offer uniform video quality over the entire video. For example, in [21], an optimal solution for the buffer constrained adaptive quantization problem is formulated. In [22], rate-distortion characteristics of the encoded video are used to find the frame rate and quantization parameters that provide the minimum distortion under rate constraints. The minimization operation is done in an iterative manner so that the measured distortion is smaller than the previous iteration at each step. However, these methods do not consider the semantics of the video content either in GoP definition or in GoP target bitrate allocation.

As the available computing power at the encoders increases, so does the level of sophistication of the encoders and their associated control techniques. By using appropriate content analysis, it is now possible to define GoPs according to shot boundaries, and allocate target bit rates to each GoP based on the shot type considering the "relevance" or "semantics" of each type of shot. Such a rate control scheme will be called "content-adaptive rate control." In content-adaptive rate control, video will be encoded according to a pre-specified or user defined relevance-distortion policy. In effect, we accept a priori that some losses are going to occur due to the high compression ratios needed, and we force these losses to occur in less relevant parts of the video content. We note that "relevance of the content" is highly context (domain) dependent. For example, in the context of a soccer game, the temporal video segments showing a goal event and the spatial segments around the ball are definitely more important than any other part of the video. There are a variety of other domains, such as other sports videos and broadcast news, where the relevance of the content can easily be classified. In content adaptive video coding, temporal segmentation policy used has a major effect on the overall efficiency and rate distribution among temporal segments. There exist techniques for automatically locating such content [23, 24, 25, 26, 27]. In [28], a summarization of the available multimedia access technologies that support Universal Multimedia Access (UMA) is presented. Segmentation and summarization of audio-video content are discussed in detail and the transcoding techniques for such content are demonstrated. The details of automatic content analysis is explained in Appendix C.

Content adaptive rate allocation ideas have been introduced in the literature before. In [29], the input video is segmented and encoded as two streams for different relevance levels with "predetermined bit rates," namely, the high target bitrate (highly relevant) and the low target bitrate (less relevant) streams. The less relevant shots are then encoded such that they are shown as still images at the receiving side and the more important shots are encoded at full quality. In this pioneering work, the decision to restrict the number of the relevance levels to two and the determination of the relative bit allocations are done in an ad-hoc manner. Quality of Service (QoS) is required for continuous playback to be guaranteed and low and high rates are determined by the client buffer size and the channel bandwidth. The server buffer size required is set afterwards, which effectively determines the pre-roll delay.

There are also techniques that divide the input video into segments by considering various statistics along these segments that affect the ease of coding without taking into account any relevance issues. For example, in [30] MPEG-7 metadata are used for video transcoding for home networks. Concepts like "difficulty hints" and "motion hints" are described. Difficulty hints are a kind of metadata that denotes the encoding difficulty of the given content. The motion hints describe the motion un-compensability metadata, which contains information about the GOP structure, frame rate and bitrate control and also the search range metadata that reduces the complexity of the transcoding process. In this work, boundaries of the temporal segments of the content are determined by the points where the motion un-compensability metadata makes a peak and then the video is transcoded using the difficulty hints. Here, GOP size is varied according to the motion un-compensability metadata. A hybrid scaling algorithm using a quality metric based on the features of the human visual system is introduced in [31], which tries to make full utilization of the communication channel by scaling video in either temporal or spatial dimensions. In this work, frame rate of the encoded video is reduced at scenes where motion jitter is low (high temporal resolution) and all the frames are kept for scenes with high motion at the expense of reduced spatial resolution.

### *1.3 Video Streaming*

Although video streaming with entertainment quality over the Internet seemed impossible only a few years back, nowadays it is a commonly used application for both live and on-demand video over high-speed networks. Here we make the definition of entertainment quality video as television quality or higher with no noticeable fluctuations in perceptual quality within the video duration. The recent advances in coding techniques [5, 4] have also made it possible to broadcast/simulcast such video content for mobile users over wireless channels.

The practical applications developed so far in video communication such as streaming video over the Internet, digital broadcast and teleconferencing are all built for monocular video technologies and are becoming more and more popular nowadays. For example, videoconferencing already occupies a huge market share as a technology that offers a wide range of applications from distance-learning to peer-to-peer video chat over the Internet and is

available with almost all commonly used Instant Messengers (IM) of today.

The transmission channel bandwidth is one of the most important limitations on the quality of the video content in a video transmission application. In a download and play scenario, if videos are not compressed efficiently, the download time for the client could be undesirably long such that the viewer would lose patience and stop the video downloaded process. From a client point of view, this is obviously a loss of not only his time but also his client device resources such as bandwidth, CPU, dynamic memory, storage space etc. for the download duration. From a service provider's point of view, this is an extremely unwanted situation as their primary objective is to distribute their content with maximum client satisfaction.

Compression efficiencies of the recent video codecs are still insufficient for transmission of entertainment quality video in low-speed wired and wireless environments where the encoding rate/quality of the video needs to be changed over the course of transmission for uninterrupted view. Video coding for streaming with such rate control (adaptive video coding) is called *information quality video* and is widely studied in the literature for the monocular case. Information quality video should be employed only when the users are not able to receive entertainment quality content due to client and network resource limitations. Here, we mainly concentrate on streaming of information quality video sequences with temporal fluctuations in quality as dictated by the system constraints and the semantic or statistical content of the scenes.

This thesis report is organized as follows: Firstly, in Chapter 2, the proposed Delay-Distortion Optimization (DDO) framework for monocular video streaming over low capacity constant bitrate channels using the MOO scheme is presented. In Chapter 3, a cross-layer multiple objective optimization framework for wireless monocular video streaming is introduced. In Chapter 4, a method for efficient compression and real-time streaming of binocular video using the existing network infrastructure and MOO formulations for stereo video streaming are presented. Finally, in Chapter 5, conclusions are drawn.

Appendix B discusses the perceptual quality measures, i.e. the distortion measures that are employed in our DDO framework. Appendix A explains the MOO scheme used in detail. Appendix C discusses the video content analysis techniques for monocular and multi-view videos in the literature. Appendix D gives a summary of the Lucas-Kanade optical flow algorithm.

Chapter 2

# MULTIPLE OBJECTIVE OPTIMIZATION FOR VIDEO STREAMING OVER IP

In this chapter, a Multiple Objective Optimization (MOO) problem for monocular video coding/streaming over a constant bit rate (CBR) channel will be formulated and solved. We propose a new pre-roll delay-distortion optimization (DDO) framework that allows determination of the minimum pre-roll delay and distortion while ensuring continuous playback for on-demand content-adaptive video streaming over limited bitrate networks. The input video is first divided into temporal segments, which are assigned a relevance weight and a maximum distortion level, called relevance-distortion policy, which may be specified by the user. The system then encodes the input video according to the specified relevance-distortion policy, whereby the optimal spatial and temporal resolutions and quantization parameters, also called encoding parameters, are selected for each temporal segment. The optimal encoding parameters are computed using a novel, multiple objective optimization formulation, where a relevance weighted distortion measure and pre-roll delay are jointly minimized under maximum allowable buffer size, continuous playback, and maximum allowable distortion constraints. The performance of the system has been demonstrated for on-demand streaming of soccer videos with substantial improvement in the weighted distortion without any increase in pre-roll delay over a very low-bitrate network using H.264/AVC encoding.

## *2.1 Introduction*

Pre-roll delay is a vital parameter in video streaming since it provides some level of protection against network throughput variations, as well as allowing flexible rate allocation in video coding. If it is chosen too small, pauses in video playback due to network throughput variations and/or unacceptable video quality due to strict rate control in video coding would result. An unnecessarily large pre-roll delay, which in the limit leads to the *download-and-play* solution, requires a very long initial wait, thus eliminating the benefit of streaming,

and is usually found objectionable by users. Therefore, video streaming applications should strike the right balance between pre-roll delay and video distortion. This issue becomes even more significant in content-adaptive video streaming over low-bitrate networks, where different bit rates (sometimes larger than the network throughput) shall be allocated to different temporal video segments (shots) according to their importance.

Streaming video over low-bitrate networks, such as 3G and beyond wireless systems, remains to be a challenging problem even with Quality of Service (QoS) support. Content adaptive video coding has been introduced as a potential solution to this problem [29], where the video is parsed into semantic temporal segments. Important temporal segments are encoded at a high enough bitrate, while the rest is transmitted at a very low bitrate (e.g., as key frames and audio). However, in this early work, the low and high bit rates are determined according to client buffer size and channel bandwidth in an ad-hoc manner. There also exist a number of content-adaptive transcoding strategies: Content-adaptive multimedia access technologies that support Universal Multimedia Access (UMA) are explained in [28] and [32]. In [28], assuming that each spatial region of interest $R_i$ of a video segment has an importance hint $0 \leq I_i \leq 1$, and a spatial resolution hint $0 \leq S_i \leq 1$, the optimization problem is formulated as finding a set of regions $R_i$ and a rescaling factor $L$ such that the overall *fidelity score* of the rescaled set is maximized and the minimum bounding rectangle surrounding the cropped and rescaled set $R_i$ fits the screen size of the receiving device. A method, where transcoding policies are determined by the content author is described in [33]. Depending on the capabilities of the client, versions of content with various resolutions and modalities are produced off-line, and the version that maximizes a subjective measure of fidelity is selected.

In [34], new performance measures for semantic adaptation, namely *Viewing Quality Loss* and *Bitrate Cost Increase*, are discussed. Object or event based segments of the input video are automatically classified into relevance levels. The unequal bit allocation strategy between important and unimportant temporal segments is determined by the semantic statistics (size and number of relevant and irrelevant segments) of the input video and the target bitrate. If a relevant segment is misclassified, a loss of quality occurs and is denoted by *Viewing Quality Loss.* Conversely, if an unimportant segment is misclassified, an unnecessarily high bitrate will be used, referred as *Bitrate Cost Increase.*

We recognize that some content adaptive techniques; including the one proposed here, yield temporal variations in quality, which may be unacceptable for entertainment-quality video. On the other hand, over very low bandwidth networks, if such techniques are not used, then almost no valuable visual information may be delivered. For example, when a soccer video is encoded at low bit rates with uniform quality, there may be severe distortions to the extent that the ball and the players are not visible and pitch lines are lost in the most important scenes (e.g., goals). Content adaptive coding facilitates best effort transmission of such relevant information instead of enforcing an average and low quality for the entire video segment. This chapter addresses optimal bit allocation between different temporal segments to minimize distortion and pre-roll delay under pre-set quality-level and continuous playback constraints. An alternative approach was introduced in [35], where rate-distortion optimized video summarization and transmission over packet lossy networks with minimum video distortion has been studied.

The classical approach to ensure continuous video playback for a fixed target encoding rate relies on the buffer management strategy of the underlying codec system, and determines the pre-roll delay as a function of the decoder buffer size (a hardware constraint). For example, the Video Buffer Verifier (VBV) model [3] of MPEG and the Hypothetical Reference Decoder (HRD) model [4] in H.264/AVC [5, 36] verify that the bitstream generated by an encoder can be played-back continuously at the decoder given the decoder buffer size and pre-roll delay for a constant bitrate (CBR) channel with a specified rate. However, the effects of the pre-roll delay or the decoder buffer size on the overall distortion are not specified in these models. With software decoders for streaming applications, hardware constraints become less important while the pre-roll delay becomes a main performance parameter (which then determines the required buffer size). In [37], an adaptive media playout (AMP) scheme was proposed as a means to ensure continuous playback, where the client device can adaptively change playout speed of the content in order to prevent buffer overflow and underflow. In [38], AMP framework is combined with the well-known rate-distortion optimized (RDO) [6] streaming. Although AMP addresses continuous playback issue in an ad-hoc manner, in low-bitrate streaming applications with non-uniform bitrate allocation among temporal segments, optimum determination of pre-roll delay under continuous playback constraint remains as an important concern.

In content-adaptive video coding and transcoding, temporal shot detection and relevance assignment methods may have significant effect on the overall performance. Most effective methods are highly context (domain) dependent. For example, in the context of a soccer game, the temporal video segments showing a goal event and the spatial segments around the ball are more important than any other part of the video. In a tennis game, breaks given between sets are not as relevant as the in-game strife. Television news reports can be segmented as anchorperson shots, news footage and commercial breaks. For movies, temporal shot detection and content analysis may facilitate bitrate assignment as a function of coding difficulty and existence of special effects. There exist several techniques in the literature for automatically analyzing such content [23, 24, 25, 26, 27]. Automatic content analysis is beyond the scope of this thesis, and we assume appropriate content analysis tools are available.

Another essential part of our framework is the definition of distortion and semantic relevance measures for video content. Although Peak Signal-to-Noise Ratio (PSNR) is the most commonly accepted distortion metric in the literature, it is not always a good indicator of perceptual quality when spatial and temporal resolutions are varied in rate allocation; hence the need for richer quality metrics [39, 40]. Blockiness and flatness measures have been more closely linked with perceptual quality. Insufficient frame rate due to frame skipping can also be considered as a source of perceptual disturbance, especially when there is high motion in the clip. Several other perceptual quality metrics have been proposed in the literature [41, 42, 43, 44, 45]. It is not the objective of this work to develop new video quality metrics, but rather to employ recently published such measures in our problem formulation.

This chapter offers the following main contributions: *i*) A new *delay-distortion optimization* (DDO) framework for content-adaptive video streaming using multiple-objective optimization (MOO), which allows studying trade-offs between pre-roll delay and distortion is proposed in Section 2.2. *ii*) A new off-line content-adaptive streaming solution for video-on-demand using this framework, where the best trade-off between spatial and temporal video resolutions (for encoding), and encoder quantization parameters for delay-distortion optimization is provided in Section 2.3. The method proposed in this chapter is an off-line procedure for rate allocation to each temporal segment which is applicable to finite length video clips. The main application is on-demand video streaming over limited

bandwidth networks with QoS where acceptable video quality must be delivered with minimum delay. First, we encode each temporal segment (also referred as GoP) individually with multiple target bit rates. Rate-distortion optimization (RDO) is used while encoding each segment [3, 4, 5]. Our proposed solution determines the target rate, and spatial and temporal resolutions for each GoP to achieve the least overall distortion and pre-roll delay for the video according to a user specific relevance-distortion policy, given the temporal segment boundaries. Finally, selected bitstreams for each GoP are pasted together using a bitstream assembly unit. The proposed framework can be used with any video codec, including the state of the art H.264/AVC encoder.

The chapter is organized as follows: Section 2.2 discusses the problem formulation for off-line delay-distortion optimized content-adaptive streaming. Section 2.3 presents a particular linear programming (LP) solution. Section 2.4 presents experimental results, where we observe considerable improvements in visual quality and user utility for a variety of bit rates using our bit allocation approach. In Section 2.5, conclusions are drawn. The main principle of the MOO approach used in our solution is overviewed in Appendix A.

## *2.2 Problem Formulation*

In this work, we assume that a video clip has already been partitioned into $N$ temporal segments. Our goal is to send more relevant temporal segments with high perceptual quality and minimum pre-roll delay over a CBR channel with bitrate *Rch* given a specific *relevance-distortion policy*, and never to send any content under an acceptable perceptual quality level. Clearly an acceptable quality (lower distortion) can be attained by increasing pre-roll delay (encoding at a rate higher than *Rch*). We first introduce the relevance-distortion policy for content-adaptive video streaming in Section 2.2.1. Section 2.2.2 addresses the relationship between pre-roll delay and distortion for the case of variable target bit rates for each segment, under continuous playback constraint. We then formulate selection of the best encoding parameters for each segment as a multiple objective optimization problem, to minimize the perceptual coding distortion and pre-roll delay at the receiver in Section 2.2.3, where maximum buffer size, continuous playback and the maximum perceptual distortion (per segment) constraints are taken into account.

### *2.2.1 Relevance-Distortion Policy*

A relevance-distortion policy assigns a relevance level $w_n$ and a maximum allowed distortion $D_n^{max}$ for each temporal segment $n$ according to its content. This policy may be universal, set at the server side, or may be user-specific, provided in a user profile. In applications where not all temporal segments may be equally interesting to a user, relevance levels can depend on the semantics of the content; in other contexts, relevance levels may be assigned according to low level features or coding difficulty. That is, a user's level of interest in certain shots (e.g., goals in soccer, touch-downs in football) in a given video context can either be set to default values derived from general opinions, or it can be signaled by a specific user prior to streaming session. It is also possible to compute relevance level automatically from audio and video features [46, 47]. For example, in a sports video, if we assume that audio signal energy increases whenever an important event occurs since the voice of the commentator and/or the noise of the audience will increase, then the relevance level of video segment $n$ may be defined by

$$w_n = \frac{E_n}{E_{global}} \tag{2.1}$$

where $E_n$ denotes the average audio energy of segment $n$, and $E_{global}$ is the average audio energy for entire video. The relevance factors are normalized between 0 and 1. We note that our framework does not depend on any specific method of determining the relevance factors.

It is important to specify a suitable distortion measure for video. This measure can be PSNR, perceptual quality measures, or a combination of both. We specify maximum allowed distortion levels $D_n^{max}$ for each video segment, as a function of the relevance of each segment, such that we would not transmit a video segment at a quality less than this specified level for an acceptable video experience.

### *2.2.2 Delay-Distortion Optimization with Continuous Playback Constraint*

In this section, we consider how to ensure continuous playback in content-adaptive video encoding/streaming, where different target bit rates, $R_1, \ldots, R_N$ will be assigned to different temporal segments of the video. Assume that the duration of video, $TD$ seconds, has been divided into $N$ temporal segments, and that the target bitrate $R_n$ for each segment $n$ is

fixed. The minimum buffer size $B_n$ to account for within-segment bitrate variations is an input to the reference decoder buffer verifier model of the particular codec that we use to code the corresponding segment. Hence, for continuous playback, the pre-roll delay must be chosen to guarantee that the receiver buffer must have at least $B_n$ bits at the start of each segment $n$ for the entire duration. Therefore, a necessary condition for $T_{pre}$ is to satisfy

$R_{ch} \cdot T_{pre} + R_{ch} \cdot \sum_{i=1}^{n} TD_i \geq \sum_{i=1}^{n} R_i \cdot TD_i + B_{n+1}$ for all $0 \leq n \leq N$

where $TD_i$ denotes the duration of segment $i$ and $B_{N+1} = 0$ [1].The first term on the left hand side is the number of bits accumulated in the decoder buffer during the pre-roll period, the second term is the total bits received until playback of segment $n$ is complete, the first term on the right hand side is the total bits drawn from the decoder buffer for playback of first $n$ segments, and the second term is the number of bits that must be present in the buffer before the start of segment $n$+1 to make sure continuous playback during segment $n$+1 according to the reference decoder buffer verifier model of the particular codec. Therefore, a necessary condition for continuous playback for the whole video can be stated as:

$$T_{pre} \geq \max_{0 \leq n \leq N} \left\{ \frac{(\sum_{i=1}^{n} R_i \cdot TD_i + B_{n+1}) - (R_{ch} \cdot \sum_{i=1}^{n} TD_i)}{R_{ch}} \right\} \tag{2.2}$$

$$= \max_{0 \leq n \leq N} \left\{ \sum_{i=1}^{n} \left[ TD_i \left( \frac{R_i}{R_{ch}} - 1 \right) \right] + \frac{B_{n+1}}{R_{ch}} \right\} \tag{2.3}$$

Observe that the value of $T_pre$ to ensure continuous playback depends on how target bit rates are assigned to different temporal segments, hence the given relevance-distortion policy, although the average bitrate and duration of the clip are the same. This is demonstrated by a simple example below.

*Example:* A video clip, with duration $TD$ and $N$=2 segments, shall be encoded in two different ways:

a) First segment, with duration $TD_1 = \frac{2}{3}TD$ is encoded at $R_1 = 96$ kbps and second segment $TD_2 = \frac{1}{3}TD$ at $R_2 = 32$ kbps; b) First segment, with duration $TD_1 = \frac{1}{3}TD$ is encoded at $R_1 = 32$ kbps and second segment $TD_2 = \frac{2}{3}TD$ at $R_2 = 96$ kbps, as depicted in Figure 2.1. The average bitrate for both cases is the same (74.67 kbps). Assuming the channel bitrate is $R_{ch} = 64$ kbps, let us now calculate *Tpre* required for continuous playback for each case.

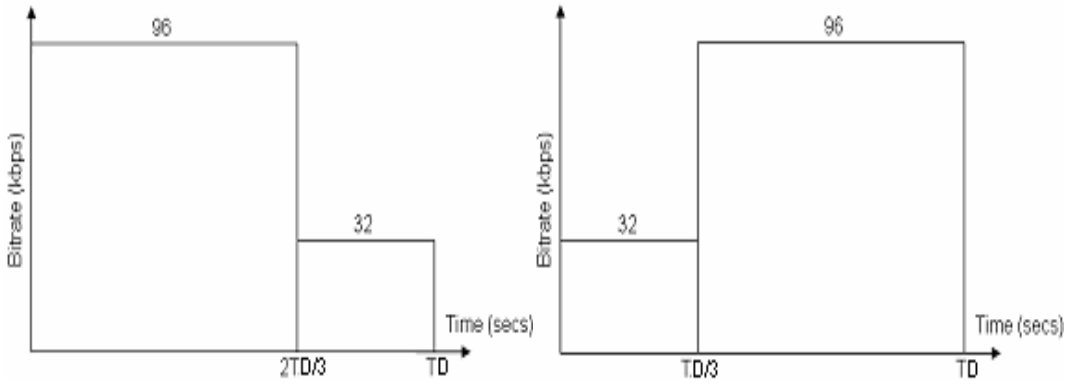
[1]Summations are assumed to be zero when the lower index is larger than the upper index.

Case a) The minimum pre-roll delay is given by

$$T_{pre} \geq \max\left\{\frac{B_1}{64}, \frac{2}{3}TD\left(\frac{96}{64}-1\right)+\frac{B_2}{64}, \frac{2}{3}TD\left(\frac{96}{64}-1\right)+\frac{1}{3}TD\left(\frac{32}{64}-1\right)\right\}$$
$$= \max\left\{\frac{B_1}{64}, \frac{1}{3}\,TD+\frac{B_2}{64}, \frac{1}{6}\,TD\right\} = \max\left\{\frac{B_1}{64}, \frac{1}{3}\,TD+\frac{B_2}{64}\right\}$$

Case b) The minimum pre-roll delay is given by

$$T_{pre} \geq \max\left\{\frac{B'_1}{64}, \frac{1}{3}TD\left(\frac{32}{64}-1\right)+\frac{B'_2}{64}, \frac{1}{3}TD\left(\frac{32}{64}-1\right)+\frac{2}{3}TD\left(\frac{96}{64}-1\right)\right\}$$
$$= \max\left\{\frac{B'_1}{64}, -\frac{1}{6}\,TD+\frac{B'_2}{64}, \frac{1}{6}\,TD\right\}$$




**(a)** **(b)**

Figure 2.1: A video clip with $N$=2 segments is encoded in two different ways.

We observe that the required minimum pre-roll delay can differ depending on how rate is allocated to each segment even though the average encoding rates and channel conditions are the same. In this setup, the pre-roll delay for case (a) could be more than twice the pre-roll delay for case (b) depending on the values of $B_0$, $B'_0$, $B_1$, and $B'_1$. We note that these values will depend on the coding pattern (IBBBPBBBP...) and encoding parameters used for the temporal segments.

Hence, in content-adaptive (variable target bit rates for segments) video streaming systems, there exists a trade-off between pre-roll delay and relevance-distortion policy used, similar to the well-known rate-distortion trade-off in fixed target bitrate encoding/streaming systems. In streaming applications with segment-based content-dependent target bit rates; however, applying classical RDO solution to each segment with different target bit rates does

not necessarily guarantee the best overall visual experience and minimum pre-roll delay for continuous playback. It is well known that larger values of *Tpre* provide more flexibility in assigning higher rates to particular segments, as well as more latitude in the allocation of rates $R_1, \ldots, R_N$ to each segment, hence a better visual experience at the expense of a larger buffer requirement and initial wait time. Therefore, we propose a delay-distortion optimization (DDO) formulation in the following to strike a compromise between pre-roll delay and overall distortion to obtain the best pre-roll delay vs. distortion performance.

### *2.2.3 Multiple Objective Optimization Formulation*

Given a relevance-distortion policy, we propose a multiple objective optimization formulation for delay-distortion optimization, where the optimal encoding parameters, hence the rates $R_1, \ldots, R_N$ for each segment are determined to minimize the pre-roll delay and weighted overall distortion, $D$, at the receiver subject to maximum acceptable average distortion $D_n^{max}$ for each segment $n$ and a maximum buffer size constraint. That is,

$$\min_{\{R_1,R_2,\ldots,R_N\}} \{T_{pre}\} = \min_{\{R_1,R_2,\ldots,R_N\}} \left\{ \max_{0\le n\le N} \left\{ \sum_{i=1}^{n} TD_i \left( \frac{R_i}{R_{ch}} - 1 \right) + \frac{B_{n+1}}{R_{ch}} \right\} \right\} \tag{2.4}$$

$$\min_{\{R_1,R_2,\ldots,R_N\}} (D) = \min_{\{R_1,R_2,\ldots,R_N\}} \left\{ \sum_{n=1}^{N} w_n \cdot D_n \cdot TD_n \right\} \tag{2.5}$$

**jointly** subject to

$$D_n \le D_n^{\max}, n = 1, \ldots, N \tag{2.6}$$

and

$$B_{n+1} \le R_{ch} \cdot T_{pre} + R_{ch} \cdot \sum_{i=1}^{n} TD_i - \sum_{i=1}^{n} R_i \cdot TD_i \le B^{\max} \text{ for all } n = 0, \ldots, N \tag{2.7}$$

where $D_n$ and $w_n$ denote the average distortion and relevance measure for temporal segment $n$ respectively, and $B^{max}$ is the maximum buffer size at the decoder. Minimization is performed over values of $R_n$ for each temporal segment $n$.

The objective function in Eqn. 2.4 is derived from the continuous playback constraint in variable target bitrate scenario explained in Section 2.2.2 and aims to minimize the initial

wait time. The constraint given by Eqn. 2.7 denotes the necessary condition to guarantee continuous playout, and it imposes that there is no buffer overflow or underflow at shot boundaries. We make the following observations and note that the proposed formulation includes some well-known solutions as special cases.

1. $D_n^{max}$ constraints (Eqn. 2.6) are not used, then distortion of a particular segment can be unacceptable. For example, the ball or field lines may be distorted in low-bitrate sports streaming.

2. If buffer size constraint (Eqn. 2.7) is not used, arbitrary $D_n^{max}$ constraints can be satisfied at the expense of increased pre-roll delay $T_{pre}$ by encoding at a rate higher than the channel rate $R_{ch}$.

3. If objective $T_{pre}$ (Eqn. 2.4) is not minimized, then the optimal solution approaches the download and play solution.

4. If objective $D$ (Eqn. 2.5) is not minimized, then it may result in under-utilization of the channel bandwidth when the minimum value of *Tpre* is zero, with the trivial solution such that $D_n = D_n^{max}$, for all $n$ where each segment is encoded with the worst allowable distortion. The multiple objective optimization solution allows allocation of the excess rate in certain segments to achieve a smaller distortion in the future segments.

5. It is not possible to simply minimize the average rate subject to distortion constraints (Eqn. 2.7) and achieve the minimum pre-roll delay. See the example in Section 2.2.2.

6. If no feasible solution exists, because the conflicting maximum distortion $D_n$=$D_n^{max}$ (Eqn. 2.6) and maximum buffer size $B^{max}$ (Eqn. 2.7) constraints cannot be satisfied simultaneously, then we try discarding the segment with the least relevance value and/or shortest duration, and try again.

### *2.3 An Off-Line Delay-Distortion Optimization Solution*

In this section, we provide a particular off-line solution to the delay-distortion optimization problem formulated in Section 2.2 using the H.264/AVC video codec.

### *2.3.1 Linear Programming Solution*

In our solution, the rates $\{R_1, \ldots, R_N\}$ will be indirectly determined as a function of a set of encoding parameters, the frame rate (temporal resolution), picture size (spatial resolution), and quantization parameter (SNR resolution), which are the independent optimization variables for each segment.

We assume that the frame rate, picture size and quantization parameter for each segment is quantized to certain pre-determined levels for a total of $K$ possible combinations. Each of the $N$ segments, with semantic relevance factors $\{w_1, w_2, \ldots, w_N\}$, has been coded off-line using these $K$ combinations of spatial resolutions, frame rates, and quantization parameters. In our study, the PSNR and blockiness measures are computed in comparison to the original video at the highest spatial resolution after spatial interpolation of the encoded-decoded video as needed. The average perceptual distortion measures for each segment are given by $\{D_1^1, D_1^2, \ldots, D_1^K, D_2^1, D_2^2, \ldots, D_2^K, \ldots, D_N^1, D_N^2, \ldots, D_N^K\}$, where the subscript denotes the segment count and the superscript denotes a particular combination of coding parameters. Each $D_n^k$ has been calculated as a weighted sum of PSNR and blockiness measures (increasing PSNR has a negative effect on distortion) given by

$$D_n^k = \frac{Blk_n^k - Blk_{\min}}{Blk_{\max} - Blk_{\min}} - \frac{PSNR_n^k - PSNR_{\min}}{PSNR_{\max} - PSNR_{\min}} \tag{2.8}$$

where $Blk_{\min}$, $Blk_{\max}$, $PSNR_{\min}$ and $PSNR_{\max}$ denote the minimum and the maximum of blockiness and PSNR measures [41], achieved respectively, computed over all shots. A motion jitter measure to account for insufficient frame rate, if included, can be computed as the difference of average motion vector lengths between full frame rate and the current frame rate. Bitrates corresponding to the above distortions;

$$\{R_1^1, R_1^2, \ldots, R_1^K, R_2^1, R_2^2, \ldots, R_2^K, \ldots, R_N^1, R_N^2, \ldots, R_N^K\}$$

are also computed for each combination of these encoding parameters. The quantization step sizes for both the intra and inter coded frames are determined as in [4]. The resulting $(R_n^k, D_n^k)$ pairs for each coding parameter set $k$ and segment $n$ are depicted in Figure 2.2.

If the original video is pre-processed to change its spatial and temporal resolution, the distortion measures outlined above become functions of spatial and temporal resolutions selected for the video segment to be encoded as well as the quantization parameters.

Hence, selection of the optimal distortion implicitly selects the best spatial and temporal resolution to be used, in addition to the optimal quantization parameter. Therefore, the problem of finding the optimal set of encoding parameters for each segment is then equivalent to finding a particular path on the coding parameter set index versus segment index graph shown in Figure 2.2, such that Eqn. 2.4 and Eqn. 2.5 are minimized subject to Eqn. 2.6 and Eqn. 2.7. Each feasible path in Figure 2.2 yields a pre-roll delay and overall distortion pair $(T_{pre}, D)$, which corresponds to a point on the two-dimensional delay-distortion graph depicted in Figure 2.3.

To find the optimal path, we first determine the utopia point (see Appendix A), which is defined as the delay-distortion point obtained by optimizing each objective function individually while ignoring the other. More specifically, we first ignore the delay objective function (Eqn. 2.4) and find the solution that gives the minimum distortion. This returns the encoding parameter set that yields the point $Opt_1 = (T_{max}, D_u)$ in Figure 2.3. Next, we ignore the distortion objective function (Eqn. 2.5) and find the encoding parameter set that gives the minimum pre-roll delay, hence the point $Opt_2 = (T_u, D_{max})$ shown in Figure 2.3. The point $U = (T_u, D_u)$ is called the utopia point.

Next, we determine the set of Pareto-optimal solution points. A delay-distortion pair $(T_{pre}, D)$ is called a Pareto-optimal solution if the value of the distortion can not be decreased without increasing the value of pre-roll delay, and vice versa. The set of *Pareto-optimal* points is shown by the curve in Figure 2.3. In order to find a set of Pareto-optimal solution points, the horizontal axis is uniformly quantized in the interval $[Tu, Tmax]$ using $Q$ levels, and minimum distortion values for the quantized pre-roll delay values are determined using linear programming, where each quantized pre-roll delay is used as an upper bound constraint, disregarding the delay objective function (Eqn. 2.4). The best compromise solution can only be determined after finding all such constrained solutions and forming the Pareto-optimal curve. Similarly, it is possible to do the quantization on the distortion axis and find the minimum pre-roll delays for $Q$ distortion constrained optimization problems. Software packages exist for solving such linear programming problems. In our study, we used General Algebraic Modeling System (GAMS) Integrated Development Environment[2] software.

Finally, the best compromise (optimal) path, hence the set of encoding parameters for

[2]GAMS Development Corporation, http://www.gams.com

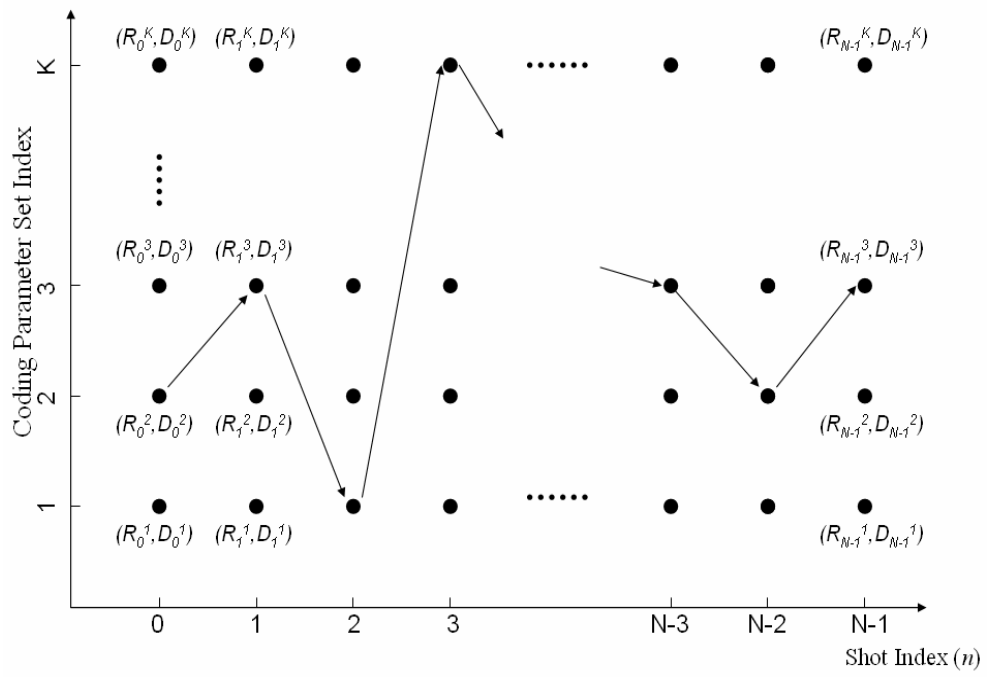



Figure 2.2: Optimal path along coding parameter set vs. the segment index plane.

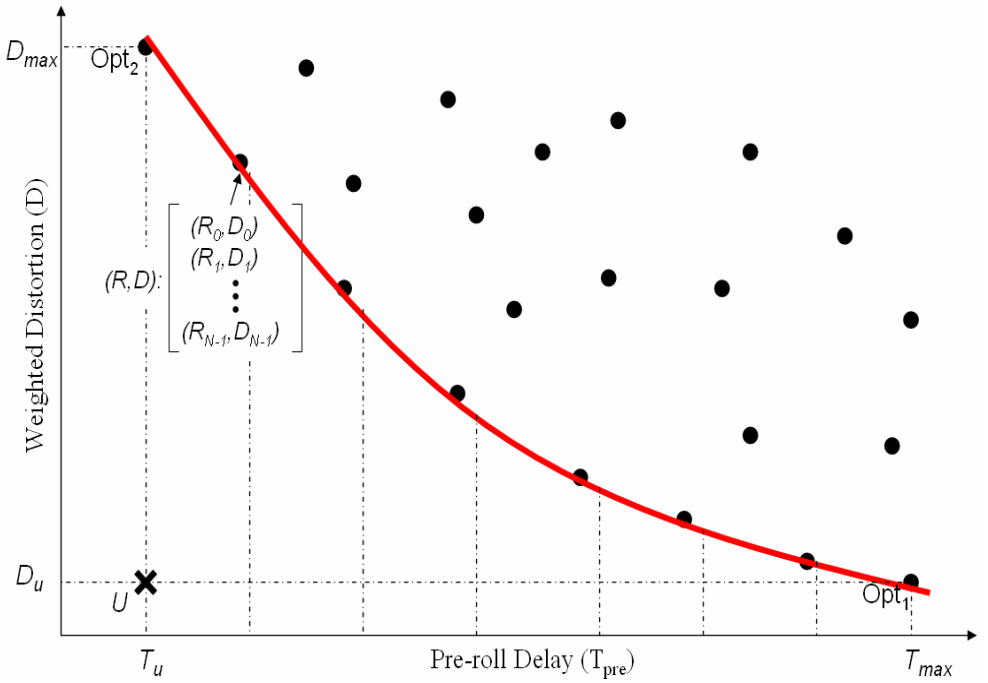



Figure 2.3: Sample points on the delay-distortion plane corresponding to paths and the Pareto-optimal curve.

each segment, is chosen as the path that corresponds to the closest solution to the *utopia point*, $U = (T_u, D_u)$, among all Pareto-optimal solutions using a suitable distance measure. An example MOO problem and its solution have been demonstrated in Appendix A.

It is well-known that an LP problem can be solved in polynomial time using optimization methods in the literature such as the projective method [48]. In order to find the Pareto-optimal curve, we need to apply the LP procedure $Q$ times (the number of quantization levels on the delay-axis). Therefore, the computational complexity of the optimization process is $Q$ times that of the LP procedure, which is polynomial time in the number of temporal segments $N$. For example; this computation takes approximately 30 minutes when $N$=1080 and $Q$=6 for a soccer game of 90 minutes.

### *2.3.2 Overall System Summary*

The operation of the proposed encoder and decoder is shown in Figure 2.4. The content analysis and shot classification module performs shot boundary detection and classification of each shot into certain pre-defined semantic content types. The output of the module is $N$ temporal segments each with a relevance measure, $w_n$, $n = 1, \ldots, N$. The pre-processor converts each segment into pre-selected spatial and temporal resolution format choices. The standard encoder encodes each input segment $In$ with all possible encoding parameter sets ($K$ spatial/temporal resolution and quantization parameter choices) resulting in $K \times N$ output segments. The output of the standard encoder for the $i^{th}$ segment and $j^{th}$ encoding parameter set is a bitstream with rate-distortion pair $(R_i^j, D_i^j)$. After this stage, all rate-distortion pairs for each temporal segment along with user-defined relevancy levels and available channel bandwidth information are fed to the MOO module. The optimal encoding strategy is then decided to minimize both pre-roll delay and overall perceptual distortion of the transmitted video. This solution requires $K$ different coding results for each of the $N$ shots. For example, we can select 2 frame rates, 2 spatial resolutions and 3 quantization parameters in a typical application, which results in $K$=12. Then the storage requirement is 12 times the size of the whole compressed video stream, which is 791 MBytes for a 90 minutes soccer video ($N = 1080$) when the average encoding bitrate is 100 kbps over all $K$ encoding schemes. Although quantization parameter is embedded in the encoded bit stream, spatial resolution and frames per second may need to be sent as side information

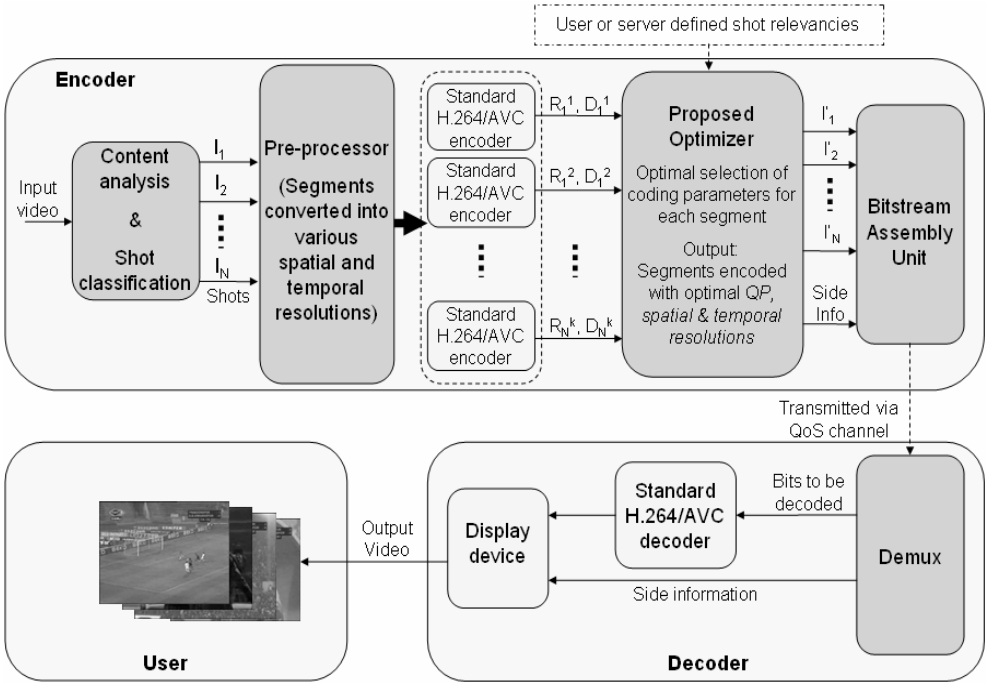



Figure 2.4: Block diagrams of the proposed encoder/streamer and decoder.

so as to synchronize when they are changed.

The operation of the decoder is straightforward. If the coding standard used supports spatio-temporal resolution changes, the resulting compressed bitstreams will be standards compliant. However, we may need a specialized display module to display all pictures at a standard spatial resolution. The display module may use the side information, consisting of the spatial and temporal resolution of each GoP to display the entire video using a single spatial and temporal resolution.

In the H.264/AVC reference encoder, the HRD model assumes that the video will be drained by a CBR channel with a rate equal to the video encoding rate. Since in our proposed system, the target bit rates assigned to each segment varies, and for some segments the target encoding bitrate can be more than the channel rate, additional logical encoder buffer will be needed to store the excess bits produced. Because bits transmitted during the pre-roll time need to be stored at the decoder side, an identical additional logical buffer will be required at the decoder as well to ensure the proper operation of the proposed variable target rate system. The required additional logical buffers at the encoder and decoder are

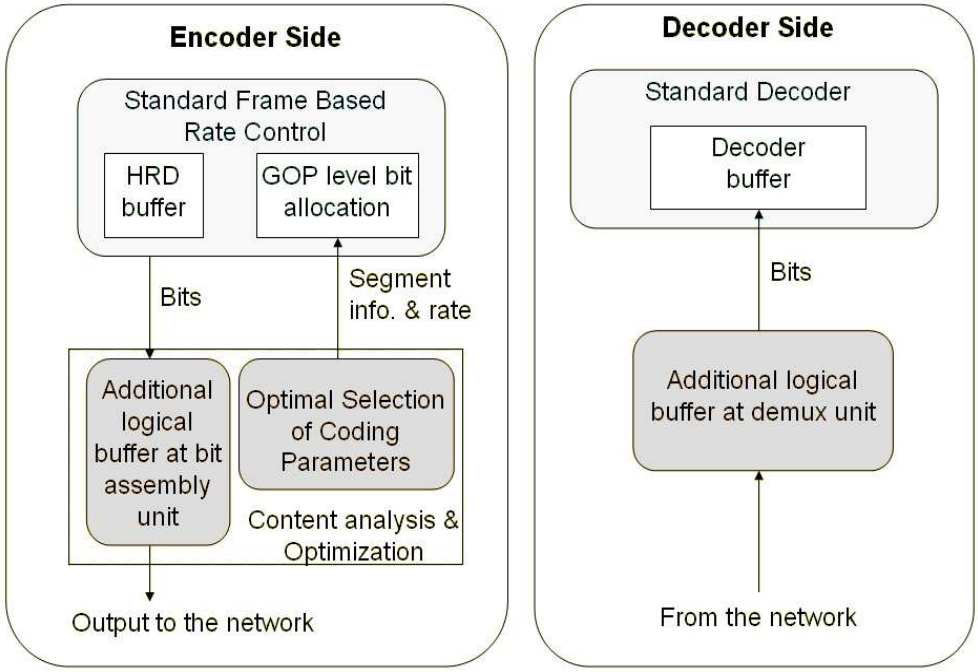



Figure 2.5: Additional "logical" buffers are used to provide continuous playback with variable GoP-target-bit rates.

illustrated in Figure 2.5. Here, the "logical" buffers demonstrate the necessary increase in the size of the codec buffers to realize DDO rate allocation. In an actual implementation, the "logical" buffers can be realized by simply increasing the codec buffer sizes accordingly.

### *2.4 Experimental Results*

In our experiments, we used H.264/AVC codec software JM 7.4 provided by the Joint Video Team (JVT) to encode each video segment using a number of fixed quantization parameters. We selected a 20 seconds soccer video clip, which is 352 $\times$ 288 and 25 fps. The video is segmented into $N = 4$ shots using the content analysis technique of [24]. The first shot is a goal event that is of great interest to most users, the second shot is a scene where the players cuddle to celebrate the goal. The audience is shown on the third shot and finally the team coach is seen on the last shot. We encoded each segment using spatial resolutions of $176\times144$ and $96\times80$, temporal resolutions of 25 fps, 12.5 fps and 6.25 fps and quantization parameters (QP) that vary between 17 and 36 for a total of $K$=232 combinations. Here, $K$ is chosen

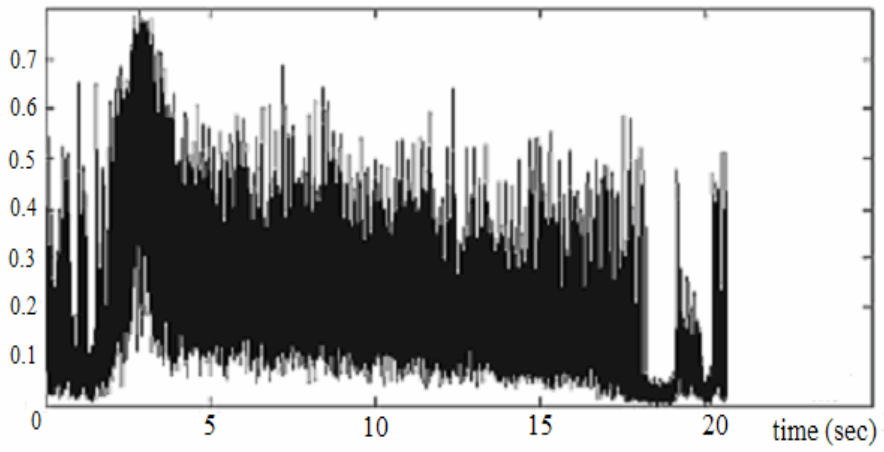



Figure 2.6: Audio energy distribution of the whole video.

large to better study the trade-off along the convex Pareto-optimal curve on a fine scale. However, $K$ can be reduced significantly by limiting the choice of quantization parameters to 2-3 values without noticeable performance degradation as shown in Figure 2.11. We computed the total bits (rates) and distortion values (as a linear combination of the PSNR and blockiness measures given by Eqn.2.8) for each combination.

User-specified relevance values for the four shots used in our experiments and refined (final) weights scaled using audio information (audio energy distribution function given in Figure 2.6) are shown in Table 2.1. Note that, in our formulation, ratios of weights (to each other) rather than the weights themselves are important. The relevance values can vary between different users. For example, if a user doesn't want to see parts of video where only the audience is shown, the weight of that shot should be set to zero. In this case, the optimal encoding result may not include this irrelevant shot at all.

Figure 2.7 shows a comparison of QCIF resolution key frames from different types of shots encoded by the proposed DDO rate allocation technique (each GoP coded utilizing RDO) and the standard RDO codec (JM 7.4 from the JVT group) at the same rate. The encoding rate is 37.57 kbps, the channel rate and the available physical receiving buffer size are assumed to be 25 kbps and 50 kBytes, respectively; resulting in an average encoding rate of 37.57 kbps and an overall delay of 10.06 seconds for the content adaptive codec at the receiving side. While the ball and lines of the field are quite noticeable in the content adaptive DDO rate-allocated clip, we can't see the ball and certain parts of the pitch lines


Figure 2.7: Sample frames from each of the 4 shots: left column are DDO coded and right column are RDO coded.

| Shot | Relevance | Average Audio Energy | Scaled Weight |
|---|---|---|---|
| 1 | 1 | 0.2689 | 0.759 |
| 2 | 0.25 | 0.2238 | 0.158 |
| 3 | 0.125 | 0.1582 | 0.056 |
| 4 | 0.125 | 0.0745 | 0.026 |

Table 2.1: Weights given by the user and refined (scaled) by audio information.

| Shot | Scaled Weight | Resolution | FPS | Bitrate | Duration |
|---|---|---|---|---|---|
| 1 | 0.759 | $176 \times 144$ | 6.25 | 68.25kbps | 4.97 sec. |
| 2 | 0.158 | $96 \times 80$ | 6.25 | 28.58kbps | 9.61 sec. |
| 3 | 0.056 | $96 \times 80$ | 6.25 | 27.4kbps | 2.86 sec. |
| 4 | 0.026 | $176 \times 144$ | 6.25 | 23.25kbps | 2.56 sec. |

Table 2.2: Optimal set of parameters for the video segments.

in the standard RDO encoded version at 37.57 kbps. Also, for the $2^{nd}$ shot, the blocking artifacts are very disturbing in the standard encoded version. For the last two shots, both coding schemes show similar performances. Figure 2.8 and Figure 2.9 show the quantization parameters and corresponding distortion measures, respectively, at each frame for both coding schemes.

*Buffer Requirements:* The proposed content-adaptive (DDO) results are compared with the variable bitrate (VBR) coded (using constant picture resolution and quantization factor for the whole video) and constant bitrate (CBR) coded (using H.264/AVC rate control [5]) versions of 120 seconds long video obtained by cascading six identical replicas of the original video. We illustrate the instantaneous decoder buffer occupancies of DDO, and regular VBR and CBR solutions with equal average bitrate in Figure 2.10, where we assure that the pre-roll delays are sufficient to guarantee continuous playback for each case for a fair comparison. The horizontal axis in Figure 2.10 denotes the time elapsed after the encoder side starts streaming. For our content adaptive DDO solution, the changes in the buffer level can be

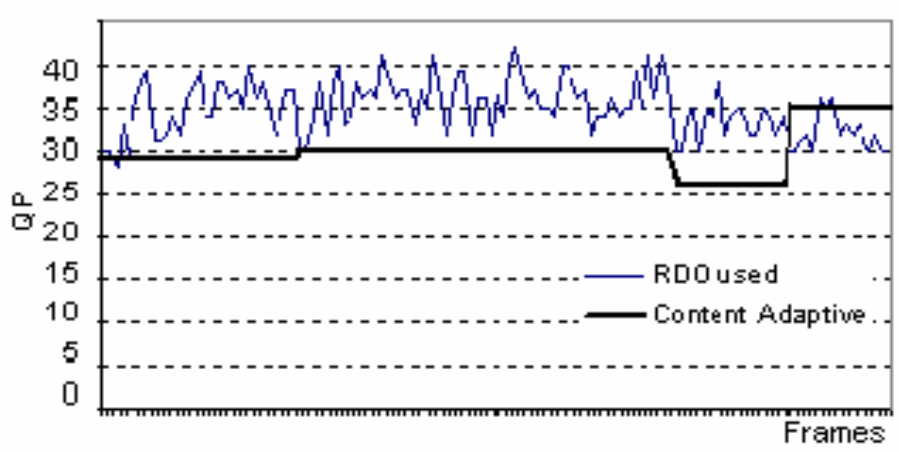



Figure 2.8: Quantization parameter values used in each frame.

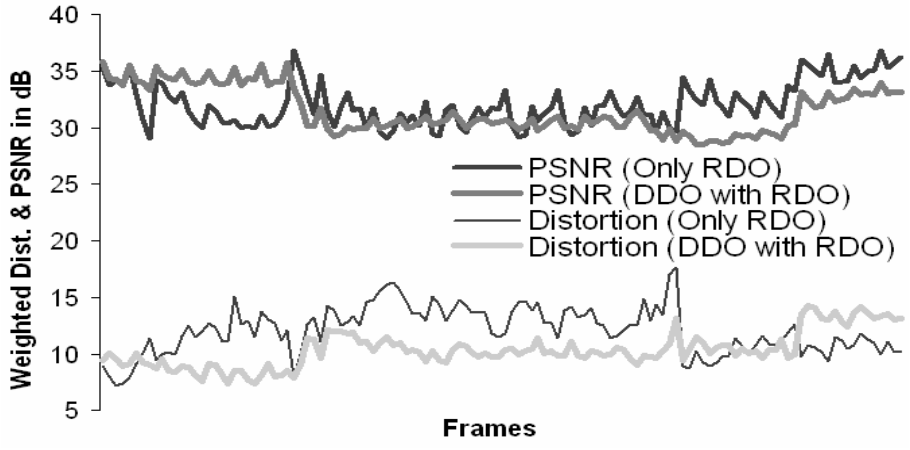



Figure 2.9: PSNR and weighted distortion of individual frames.

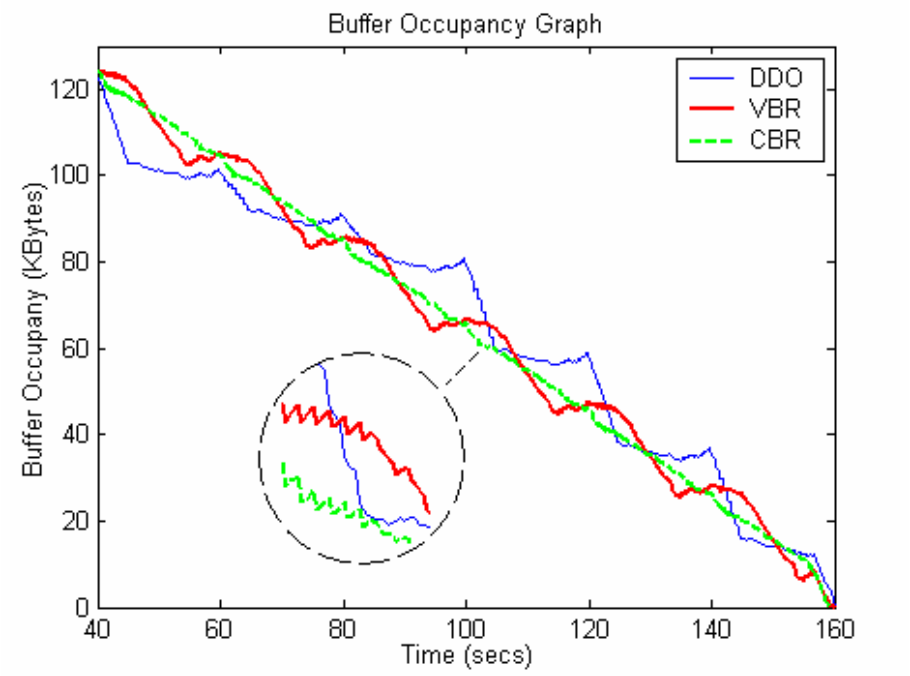


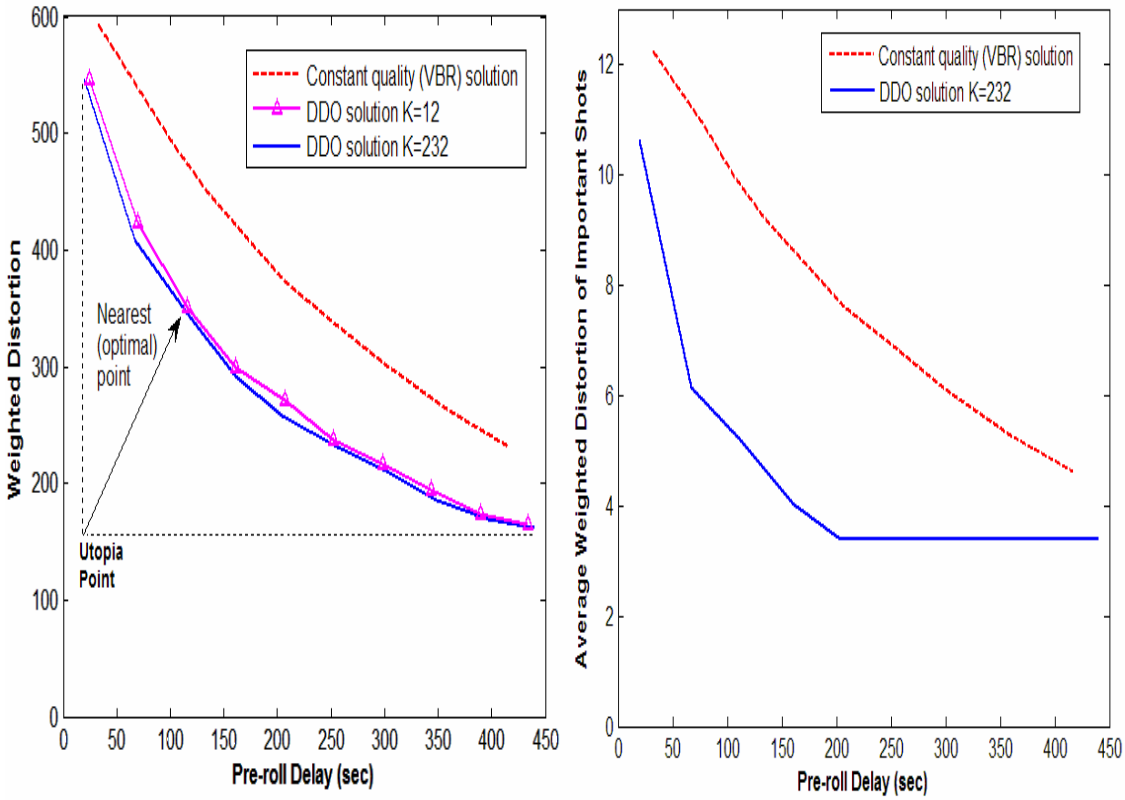

Figure 2.10: Buffer occupancy graph for the 120 seconds video after pre-roll time.

either steep or slow depending on whether a high relevance or a low relevance segment is displayed. In our solution, the maximum physical client buffer size is set to 300 kBytes; although the maximum buffer level observed (necessary and sufficient client buffer size) is found to be 125 kBytes with a pre-roll delay of 40 seconds. The resulting average encoding bitrate throughout the video is 33.33 kbps. For this example, at the same bitrate, VBR and CBR solutions require approximately the same buffer size as our solution; hence, our solution provides higher video quality in important temporal segments without incurring additional pre-roll delay and buffer requirements over the standard CBR solution with RDO.

If the buffer constraint is kept too small, it may not always be possible to come up with a feasible solution, unless concessions are made on one or more of the constraints/objective functions. Note that the DDO solution can not be dominated by either CBR or VBR solutions both in pre-roll delay and distortion, since the optimal DDO solution would approach the better one of these solutions in a worst case scenario. In cases where there exist feasible solutions, our framework would find the optimal solution.

*Delay-Distortion Trade-off:* For a 25 kbps constant bitrate channel, the delay-distortion

curves for the 120 seconds video with no buffer constraints imposed are shown in Figure 2.11. For equal pre-roll delays, the proposed solution shows better weighted distortion performance on average, especially at the important temporal segments, for which the video PSNR gain is around 4.5 dB compared to the VBR solution. Note that, as the pre-roll delay increases, the required buffer size at the client size also has to increase. As a result, the larger the receiver buffer is, the more flexibility the encoder side has on GoP level bit allocation, increasing the overall video quality, as seen in Figure 2.11.




**(a)** **(b)**

Figure 2.11: Comparison of VBR encoding and the proposed method for delay-distortion performances (a) over all segments and (b) over important segments for the 120 seconds long video.

In order to illustrate how the minimum required decoder buffer size is affected by the individual shot durations, we now construct a 1200 seconds (20 minutes) long video where

the duration of each shot in the 120 seconds long video is made ten times larger (shot durations up to 96.1 seconds). For the same target bitrate (33.33 kbps), if the same bit allocation strategy among shots is applied as for the 120 seconds video, the required buffer size would be 1250 kBytes. On the other hand, if we re-run our optimization algorithm for the 1200 seconds video under the same conditions with a maximum physical buffer size constraint of 1000 kBytes., this results in an average video encoding bitrate of 30.03 kbps, a pre-roll delay of 243.9 seconds, and a minimum required buffer size of 762.2 kBytes. For the same average-bitrate video, the download-play approach would result in 1443.8 seconds of delay and 4512 kBytes of storage space. Hence, the pre-roll delay and buffer requirements of our method do not grow linearly with overall video or individual shot durations. On the contrary, the maximum physical buffer size constraint in the optimization formulation causes the average encoding bitrate to drop when necessary.

The results for 120 seconds and 1200 seconds videos, shown in Table 2.3 indicate that the required buffer sizes are well within the capabilities of today's clients. With the continuous playback guarantee, the pre-roll delay and buffer size requirements of the standard RDO solution is very close to ours. However, the video quality is much higher in the important temporal segments in our solution. Hence, our solution provides higher video quality in important temporal segments without incurring additional pre-roll delay and buffer requirements (penalty) over the standard RDO solution.

The results presented here are provided as a proof of concept. Improvements in weighted PSNR and pre-roll delay may vary with the video content and specific relevance-distortion policy adopted.

### *2.5 Conclusions*

This chapter introduces a new MOO framework for delay-distortion optimization (DDO) in content-based adaptive GoP-level rate allocation for video streaming over resource-limited networks using linear programming. Semantic relevancy of shots has been taken into account in determination of encoding parameters for each shot. Clearly, video with unacceptable quality is by definition of no use for anyone. On the other hand, there are users who will wait to watch video at an acceptable quality as manifested by the streaming applications on the Internet. What we accomplished in this chapter is that, we developed a technique

| | | | Download-Play Solution | | Proposed Solution | |
|---|---|---|---|---|---|---|
| Channel Throughput | Average Bitrate | Video Duration | Delay | Req. Buffer Size | Delay | Req. Buffer Size |
| 25 kbps | 33.33 kbps | 120 sec | 160 sec | 500 kBytes | 40 sec | 125 kBytes |
| 25 kbps | 30.08 kbps | 1200 sec | 1443.8 sec | 4512 kBytes | 243.9 sec | 762.2 kBytes |

Table 2.3: Buffer requirements.

to reduce this waiting time to levels much lower than that of download and play, keeping the relevant quality at an acceptable level over low bandwidth channels. The proposed method not only maximizes perceptual quality of relevant parts in the video, but also minimizes the pre-roll (initial playback) delay at the receiving side. It outperforms the performance of regular bit allocation schemes in the relevant shots (4.5 dB gain), while still providing an acceptable quality for other shots with quite affordable buffer requirements. The proposed framework does not depend on a particular video coding technology.

Chapter 3

# MULTIPLE OBJECTIVE OPTIMIZATION FOR CROSS-LAYER WIRELESS VIDEO STREAMING

In this chapter, we present a cross-layer optimized video rate adaptation and user scheduling scheme for multi-user wireless video streaming aiming for maximum quality of service (QoS) for each user, maximum system video throughput, and QoS fairness among users. These objectives are jointly optimized using a multiple objective optimization (MOO) framework that aims to serve the user with the least remaining playback time, highest delivered video seconds per transmission slot and maximum video quality. Experiments with the IS-856 (1xEV-DO) standard numerology and ITU Pedestrian A and Vehicular B environments show significant improvements over the state-of-the-art wireless schedulers in terms of user QoS, QoS fairness, and the system throughput.

## *3.1 Introduction*

A wireless system that enables on-demand video streaming has unique design challenges compared to its wired counterpart, due to the time-varying nature of the wireless channel and scarcity of the system resources which makes it impossible to guarantee any video specific Quality-of-Service (QoS). In a cellular network with multiple users streaming various videos, achieving optimal sharing of system resources and allocating optimal video rate to each user simultaneously so that the highest possible application layer QoS is provided to each user in a fair manner while maximizing spectral efficiency of the overall system is a current research problem.

In the past, the cellular network used to deliver only voice data over a circuit-switched network. On the other hand, in general, voice data is bursty, while circuit-switched networks are dedicated to a user in the whole duration of use. This leads to under-utilization of the available bandwidth when users are not effectively using the communication channel. A better way of utilizing the channel capacity is to spare some of the bandwidth for data

other than voice and to let the users share this bandwidth in a systematic and adaptive manner. In the simplest case, a Time Division Multiple Access (TDMA) scheme where each user is allocated a time-slot periodically (round robin mechanism) can be applied [49]. The TDMA scheme is also suboptimal and results in under-utilization of the communication channel bandwidth. Presently, in the modern wireless telecommunication systems such as the High Data Rate (HDR) protocol, the base station constantly probes the instantaneous channel conditions of all the users in the network and tries to schedule the users with the best instantaneous conditions. These new systems are merely based on building intelligent time-division multiplex scheduling algorithm overlays on the existing physical layer based on instantaneous user demands and capabilities.

Standardized 2.5 and 3G systems (e.g., cdma2000, UTRAN, and EGPRS) try to provide video services by building on the air interface of the old 2G systems, such that existing 2G resource allocation basics are inherited and further improved. However, these improvements over the voice-centric 2G systems are not enough to provide support for high data rate and less delay intolerant services such as video streaming since resource requirements for packet data are significantly different from that of voice. For this reason, there is need for adaptive and efficient system resource sharing schemes unique to high-speed packet data access over wireless channels. Among such techniques, the opportunistic multiple access scheme in which all system resources are allocated (scheduled) to only one user at a given pre-defined time slot is shown to be optimal in terms of average system throughput in frequency flat fading channels [50]. In this scheme, adaptive coding and modulation need to be employed for each scheduled user such that optimal spectral efficiency is achieved. The main focus of this chapter is on wireless systems that employ opportunistic multiple access with adaptive coding and modulation. Examples of such systems are 3G extensions, such as 1xEV-DO for cdma2000 and HSDPA for WCDMA.

The essential target of cross-layer optimization is to provide a vertically optimized video communication system, in which, resources such as power, spectrum, time and code space are optimally and dynamically shared by system subscribers.

The scheduling algorithm has a major impact on the system performance in opportunistic multiple access systems. For delay tolerant data, it is possible to increase the system throughput significantly by making use of the time-varying characteristics of the wireless

system, provided that the channel characteristics are continuously tracked and accurately and quickly fed back to the transmitter. On the other hand, such capability may become very limited when the data is less tolerant to delay, as in video streaming. Well known scheduling algorithms for opportunistic multiple access systems are maximum C/I (carrier-to-interference ratio), first in first out (FIFO), proportionally fair (PF) [51] and exponential [52] schedulers. The maximum C/I scheduler, also called the maximum rate scheduler, assigns the user with the best channel condition to maximize the overall system throughput. The downside is the lack of fairness among subscribers, since users who are relatively further away from the base station (BS) will always suffer from lack of service, while users that are closer to it will almost always utilize all of the system resources. The FIFO scheduler selects the user who has waited the longest to receive data in the network. Apparently, this algorithm behaves optimally in terms of fairness in the number of time slots assigned per user. However, it may suffer from low throughput performance. Furthermore, fairness in slot assignment does not necessarily mean equal average data throughput for all users. The PF scheduler assigns the user with the best channel condition improvement relative to its own mean. This algorithm keeps track of every user's average available channel data rate over a given time window. At every time slot, the ratio of each user's available channel throughput to its average over that time window is calculated. The user whose ratio is the maximum is assigned for that time slot. The exponential scheduler attempts to add a certain level of fairness in terms of service latency to the PF scheduler, so that no user is left without service for long periods of time.

Existing 2G-3G wireless systems employ the Open Systems Interconnect (OSI) layered design, where the interfaces between layers are fixed; hence, design of an individual layer does not consider constraints of other layers. For example, in video streaming, resource allocation at the Medium Access Control (MAC) layer and video source coding at the Application Layer are handled independently. This makes the design of individual layers easier at the expense of suboptimal system performance. Indeed, the general purpose scheduling algorithms discussed above for the opportunistic multiple access system pay no regards to the application layer. Similarly, recent video coding technologies such as H.264/AVC [53] and scalable coding (SVC) [54] perform rate allocation without any regards to other OSI layers. Wireless systems provide users with rapidly varying data rates due to fast

channel signal-to-interference plus noise ratio (SINR) variations, which can best be exploited using an application-layer fair opportunistic multiple access scheme. At the same time, the wireless link also suffers from relatively slower oscillations in its average throughput due to shadowing effects, which can be exploited by adapting the video source coding rate accordingly. Hence, further improvements are possible for wireless systems by considering the interplay between different OSI layers with a cross-layer design.

There have been several works addressing cross-layer design of video streaming systems in the literature, which propose to adapt the source coding rate and/or system resource allocation among users in response to feedback from multiple layers. They have all been aimed to maximize either the system resource utilization or the perceived video quality, but not both of them jointly. In [55] an adaptive video rate control scheme for real-time video streaming using scalable video coding is introduced. Using the statistics of packets flowing through the network (packet drop percentage, round-trip-time, etc.) the current channel state is estimated and additional video enhancement layers are sent through the channel if conditions get better, resulting in better video quality. In [56], a joint source coding and rate adaptation scheme to achieve energy efficient video streaming is presented, where the number of macro-blocks (MB) in each packet, coding parameters of MB, transmission rate and scheduling of the packets are determined according to distortion-constrained minimization of energy required to successfully send the packet. In [57], a packet scheduling framework for wireless video streaming using an error-prone feedback is introduced. By observing the packet losses using the ACK/NACK messages and channel statistics, an optimal transmission strategy for the upcoming packets is determined. In [58], several abstracted parameters from different OSI protocol layers are used as decision variables in the optimization of a single objective function whose parameters depend on system design targets. Here, the results obtained for different objective function parameters may be different. Since only one objective function is considered in the optimization formulation, this scheme suffers from either service fairness or average system performance. Recently, we introduced a cross-layer scheduling framework for video streaming over the 1xEV-DO system, where not only the current system throughput capabilities but also the receiver buffer levels of individual users are optimized simultaneously [59]. However, source coding rate adaptation was not addressed in that work.

A possible approach for video rate adaptation is to store several versions of the same content, each encoded at a different rate, and switch among them as necessitated by the network conditions [60]. This is particularly suitable for video on-demand, where encoding is off-line and there is sufficient space to store multiple encodings. Another well-known approach is layered video coding, also called scalable video coding [54]. This method provides a base layer coded at a lower rate, as well as one or more enhancement layers. The base layer can be decoded independently, and enhancement layers, which can only be decoded if the base layer decoding is successful, refine the video quality. Rate adaptation is achieved by changing the number of enhancement layers transmitted [61]. A variation called fine grain scalability allows rate/quality tradeoffs at much finer granularity. Both approaches have been demonstrated to be useful in achieving good network utilization and high video quality [62, 63]. Several papers that overview these concepts, and extend them with techniques, such as frame skipping or coefficient dropping [64, 65] can be found in the literature. Alternatively, it is possible to employ advanced rate control to vary the video rate arbitrarily on the fly while real-time encoding.

In this chapter, we present a new cross-layer, multiple-objective optimization (MOO) framework for joint video rate adaptation and system resource allocation (user scheduling) for multi-user wireless video streaming systems. The MOO framework jointly considers "*application-layer QoS*" of the individual users, "*application-layer QoS fairness*" among all users, as well as the overall "*video throughput*" towards a best compromise solution. The video throughput is defined as the delivered video seconds per transmission second, which depends on both the channel data throughput and video encoding rate. In constant bitrate video encoding, video throughput is linearly related to the channel throughput. In Section 3.2, we introduce the application and physical layer related objective functions, including application-layer QoS fairness, and the problem formulation. In Section 3.3, we provide experimental results for the wireless opportunistic multiple access scheme for the 3G 1xEV-DO system [66]. Finally, in Section 3.4, we draw conclusions.

## *3.2 Optimization Criteria and Problem Formulation*

The optimization criteria used in the MOO framework are modeled in Sections 3.2.1-3.2.3, and formulation of the optimization problem is presented in Section 3.2.4, where we seek

to find a best compromise operating point such that any one of the objectives cannot be further improved without worsening the others by a bigger margin. This solution will provide a means to decide which user to schedule at a given time slot and what video source coding rate to use for that user.

### *3.2.1 Application-Layer QoS for Each User*

The quality of encoded video is generally measured in terms of the Peak-Signal-to Noise-Ratio (PSNR). In the proposed framework, we consider a system where the modulation and coding parameters are set so that the physical layer operates at the conventional 1 percent packet error rate. However, even this 1 percent packet error rate can cause a significant degradation in the PSNR of the received video stream. To ensure correct reception of all physical layer packets, we also employ Automatic Repeat reQuest (ARQ) at the physical layer so that every erroneous physical layer packet is retransmitted until it is received correctly. This clearly comes at the expense of buffer underflows and consequently, pauses in the playback. We assume a video-on-demand scenario, where pauses will not cause any loss of content; in other words, the playback will resume at the same position where the pause occurred. Therefore, the PSNR of received video will be identical to that of the transmitted video, and we will assess the perceived received video quality in terms of both the PSNR and the number of pauses. Alternatively, we could limit the number of retransmissions and deal with lost packets using error concealment methods [67] at the receiver, which would reduce the total wait time at the expense of a decrease in the received video PSNR.

The PSNR for user $i$ is directly related to the mean video encoding bitrate, $\mu_i(k)$, for that user. Adaptation of this mean video encoding rate may be beneficial especially when transmission is over a time-varying channel. This is because: i) continuous playback may be maintained, if the channel characteristics worsen for a particular user, at the expense of a lowered perceptual quality; ii) video quality may be increased at times when a user experiences a better than average channel condition. In this chapter, we focus on stream-switching method for video rate adaptation [60], where we switch between various streams of the same video, each encoded with a different $\mu_{i,l}(k)$, which denotes the encoding rate for the $l$'th video stream for user $i$ at the $k$'th time slot. We employ H.264/AVC [53] encoding with a GoP size of 12 frames. Therefore, the mean encoding rate may be switched once in

every $12^{th}$ frame.

One of the objectives of our framework is maximization of the video encoding rate for each user, thereby maximizing the user PSNR. The transmitter is allowed to vary the mean video encoding rate in response to the feedback received from the users' on their observed channel characteristics as well as buffer fullness levels, which indicates whether the users' will experience pauses in their playbacks. Therefore, in order to maximize $\mu_{i,l}(k)$, the scheduler needs to select the user $i$ and its $l$'th video stream that results in $\mu^*(k) = \max_{i,l} \mu_{i,l}(k)$ , at all times.

### *3.2.2 Average Video Throughput for All Users*

We define the duration of video content delivered to the scheduled user per transmission second as the *video throughput*, which is an important service quality parameter that needs to be maximized. Note that, in a generic wireless data communication system that does not consider application QoS specifically, it is desirable to maximize the average channel capacity to achieve spectral efficiency. In case of variable bitrate (VBR) video coding, the maximization of channel capacity is not equivalent to the maximization of video throughput. However, they would be equivalent in case of constant bitrate (CBR) video streaming, since the user with the highest data throughput would also be able to receive the longest video segment into its buffer at any given time slot. The maximization of the downlink video throughput is possible via available achievable data rate feedback from all users at each time slot, given that the video encoding rates are known at the server side. Hence the downlink video throughput improvement can be achieved at the expense of increased uplink channel overhead.

Assume that there are $M$ users with streaming video requests in the wireless system. Let $k$ $(1 \leq k \leq \infty)$ denote the discrete time slot index for scheduling. Let $\lambda_i(k)$ be the transmission bitrate supported by the wireless channel for user $i$ if scheduled at time slot $k$ and $a_i(k)$ be a binary variable that takes the value "1" if the user $i$ is scheduled at time slot $k$, and "0" otherwise. Note that, the video encoding rate $\mu_{i,l}(k)$ is allowed to vary from GOP to GOP in order to achieve a tradeoff between increasing video PSNR and decreasing the number of pauses. Also note that $\mu_{i,l}(k)$ can be varied only at the scheduling slot indices $k$ that correspond to the beginning of a new GOP for user $i$.

Let the video throughput of the $i$'th user for the $k$'th time slot be denoted by $t_{i,l}(k)$ if the $l$'th video stream is selected for transmission. Then,

$$t_{i,l}(k) = \frac{a_i(k) \cdot \lambda_i(k)}{\mu_{i,l}(k)} \tag{3.1}$$

Now, let the average system video throughput up to the $n$'th time slot be denoted by $t(n)$ . We can calculate the average video throughput in a recursive manner in terms of its previous value as follows:

$$t(n) = \frac{1}{n}\left((n-1)\cdot t(n-1) + \sum_{i=1}^{M} \frac{a_i(n)\cdot\lambda_i(n)}{\mu_{i,l}(n)}\right) = \frac{(n-1)\cdot t(n-1)}{n} + \frac{1}{n}\cdot\sum_{i=1}^{M} a_i(n)\cdot t_{i,l}(n) \tag{3.2}$$

For large values of $n$, the first term on the right hand side becomes approximately equal to $t(n-1)$. Then, the video throughput enhancement due to scheduling the $i^{th}$ user at time slot $n$ to transmit the $l^{th}$ video stream, $\Delta t_i(n)$, can be approximated as:

$$\Delta t_i(n) = t(n) - t(n-1) \cong \frac{1}{n}\cdot\frac{\lambda_i(n)}{\mu_{i,l}(n)} = \frac{1}{n}\cdot t_{i,l}(n) \tag{3.3}$$

where the only differentiating factor amongst users is the instantaneous video throughput, $t_{i,l}(n)$ at time $n$. Therefore, in order to maximize the value of $t(n)$, the scheduler needs to select the user $i$ and associated $l^{th}$ video stream with the highest instantaneous video throughput, $t^*(n) = \max_{i,l} t_{i,l}(n)$, at all times.

### *3.2.3 Application-Layer QoS Fairness*

In the literature, equating the system access time, equating the received average data rate, and equating the observed average delay across users have all been used as fairness measures. We classify such fairness criteria as link-layer fairness. It is apparent that link-layer fairness pay no regards to the specific QoS requirements of the application. Ultimately, a system should aim to provide service that satisfies its QoS requirements for all users, regardless of their current channel conditions. We define such a measure of fairness as application-layer QoS fairness. Hence, an application-layer QoS fair wireless video streaming system should aim to provide high PSNR video with minimum number of pauses for all of its users. While maximization of PSNR requires increasing the encoding rate $\mu_{i,l}(k)$, minimization

of number of pauses requires decreasing $\mu_{i,l}(k)$, which sets up an interesting optimization problem.

Video streaming applications employ a finite buffer at the receiver, and playback begins when the buffer reaches a pre-defined fullness level, resulting in a pre-roll delay. Hence, minimization of number of pauses is also related to the pre-roll delay. We define the "total wait time" as the sum of the pre-roll delay and duration of all pauses. Let $\theta_i(k)$ be the total remaining video playback time in seconds for user $i$ at time slot $k$, in case it is never scheduled again. We assume that the application cannot vary the video display rate, i.e., adaptive playout methods are beyond the scope of this chapter. Then, $\theta_i(k)$ may be computed by the user by counting the number of frames in its buffer at the $k^{th}$ time slot, $f_i(k)$. This is done by parsing the received stream, and locating the startcodes for each frame. Once $f_i(k)$ is determined, the remaining playback time $\theta_i(k)$ can easily be computed as

$$\theta_i(k) = \frac{f_i(k)}{\Omega} \tag{3.4}$$

when a constant frame rate of  Hz (frames per second) is used. Then, application layer fairness, i.e., minimizing the number of pauses observed during playback for each user, can be achieved by scheduling user $i$ that has the smallest remaining video playback time, $\theta^*(k) = \min_i \theta_i(k)$.

### *3.2.4 Problem Formulation*

We have three objectives for the desired system operation, namely, at time slot $n$, the proposed system should schedule user $i$ and video stream $l$ such that all active users experience high video PSNR with minimum number of playback interruptions, while the system enjoys a high average video throughput. Then, the optimization formulation for scheduling a user at time slot $n$ and deciding on its source data rate is given by,

1. Select the user $i$ and the associated video stream $l$ that provides the highest video encoding data rate, $\mu_{i,l}(n)$:

$$\max_{i,l}(\mu_{i,l}(n)) \tag{3.5}$$

2. Select the user $i$ and the associated video stream $l$ that provides the maximum available average system video throughput:

$$\max_{i,l}(t_{i,l}(n)) \tag{3.6}$$

3. Select the user $i$ whose remaining video playback time is the smallest:

$$\min_{i}(\theta_i(n)) \tag{3.7}$$

*jointly* subject to buffer constraints,

$$0 \leq B_i(n) \leq BufferSize \tag{3.8}$$

for all $i$ where $B_i(n)$ is the number of bits in the $i^{th}$ user's buffer at the $n^{th}$ time slot and *BufferSize* is the buffer size of the users.

If we assume that these three objectives are equally important to the user, their values can be scaled to an equal range (e.g., the range [0,1]). In case of unequal importance among the objectives, values of $\mu_{i,l}(n)$, $t_{i,l}(n)$ and $\theta_i(n)$ can be scaled to ranges $[0,w_1]$, $[0,w_2]$ and $[0,w_3]$, respectively, where $w_p$ is the importance weight of the $p^{th}$ objective.

In the proposed framework, we assume that quantized information on channel quality and remaining playback times for each user are available at the base station for each time slot by means of a physical and application layer feedback. The remaining playback times can be computed at the server side via an infrequent 1-bit application layer feedback from each user as explained in Section 3.3.1. Buffer overflows can be detected similarly. Availability of this information is useful for not only scheduling, but also intelligent video source code adaptation. The details of the uplink overhead caused by the physical and application layers feedback are discussed and demonstrated by experimental results in Section 3.3.

The three objectives stated in (3.5)-(3.7) may be conflicting with each other. For example, it is possible to have a user that provides the highest video throughput while having a large remaining playback time in its buffer, contradicting objectives (3.6) and (3.7). Similarly, the objectives of maximum remaining time in the buffer and high video data rate are

contradictory. For this reason, the optimization should attempt to find the best compromise solution in the Pareto-optimal sense. Such optimization is called multiple-objective optimization and is described in Appendix A.

## *3.3 Experimental Results*

Extensive simulations have been conducted to assess the performance of the proposed cross-layer multiple objective optimization for joint scheduling and video rate adaptation. We use IS-856 (1xEV-DO rev. 0) numerology [66] in the simulations to provide realistic results. Details of the simulation platform are given in Section 3.3.1. Results are presented to compare the proposed framework (when there is no video rate adaptation) with the traditional schedulers from the literature in Section 3.3.2. Results with video rate adaptation are shown in Section 3.3.3. Sensitivity of the system performance when the operating point deviates from the optimal one is discussed in Section 3.3.4.

### *3.3.1 Simulation Platform*

The simulations are composed of three stages: i) System level simulations, ii) physical layer simulations, and iii) joint scheduling and video rate adaptation simulations.

System level simulations model a 3-tier cellular layout with a cell radius of 1 km. Here, the first three tiers have 6, 12 and 18 cells centered around the cell of interest, respectively. Different videos of 183 seconds total duration are assumed to be demanded by a maximum of 32 users in the center cell. These users are repeatedly and randomly dropped into the center cell uniformly over a period of 1 second, which corresponds to 600 slots for the IS-856 system. The simulation sampling rate is set at 600Hz, which corresponds to one sample per time-slot. For each time-slot, the ITU Pedestrian A and Vehicular B wireless channel models [68] have been used to calculate the received signal-to-noise ratio for each user. Interference level is determined assuming that all base stations in the 3-tier layout always transmit at full power. The ITU models take path-loss, shadowing, multipath fading, and mobility into account. Gudmundson's model has been used to model the autocorrelation of the shadow fading [69].

The physical layer simulations have been conducted using Agilent's Advanced Design System (ADS 2004A) program. Here the IS-856 system is simulated to calculate the nec-

essary signal-to-noise ratio for each supported transmission rate so that a maximum of 1% packet error rate is achieved. IS-856 is originally designed to provide packet switched data to multiple users over a bandwidth of 1.25 MHz by providing service to only a single user at a given time. A time slot of 1.67 ms is defined for this operation. The active user is chosen according to a desired scheduler. The data rate of the scheduled user is selected according to its observed channel conditions. According to Shannon's capacity formula, the channel capacity increases logarithmically with signal power, i.e. signal-to-noise ratio (SNR). The probability of symbol error in transmission changes with respect to the SNR as shown in Figure 3.1.

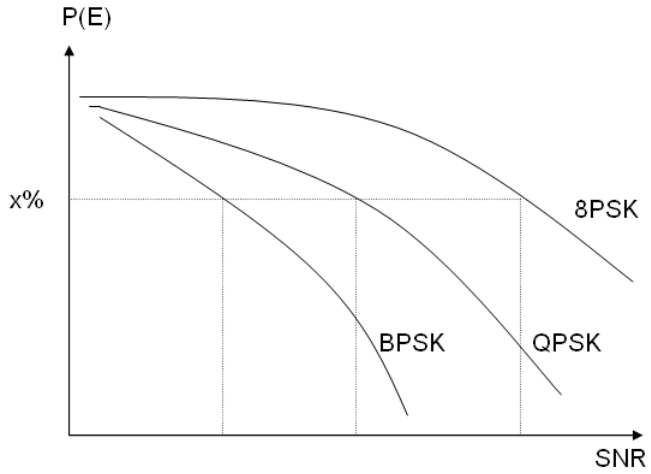



Figure 3.1: Changes in probability of symbol error in transmission with respect to the SNR.

Therefore, given the base station power and packet error rate constraints, the instantaneous signal-to-noise ratios (SNR) of each user can be converted into maximum available user data rates. An example case with 3 users is shown in Figure 3.3.1 where $t$ denotes the time slot index.

The data channel rate for a single user can take on values in the range from 38.4 to 2457.6 kbps. To enable this variability, the system uses 1/3 and 1/5 rate Turbo codes and QPSK, 8-PSK and 16-QAM modulation schemes adaptively. Also repetition and puncturing provide finer grain coding. After scrambling, modulation and repetition, the transmission packet is de-multiplexed into 16 blocks. Each of these blocks is spread using one of the orthogonal 16 Walsh codes. The final transmission packet is the sum of these 16 blocks. Four distinct

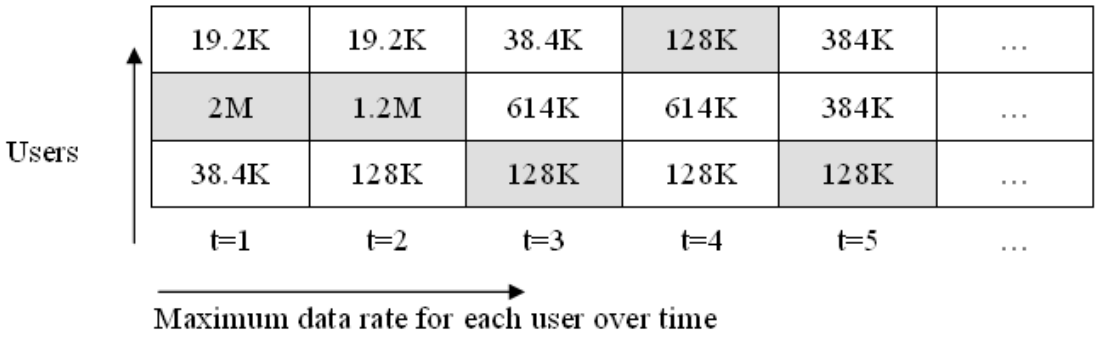



Figure 3.2: An example 3-user scenario and the maximum deliverable data rates due to error rate constraints and channel (SNR) conditions. The selected user for each time-slot is shown with gray color.

transmission packet sizes are described and each supported data rate maps onto one of these packet sizes. The transmission packets may span multiple time slots depending on the data rate. The data rates, the corresponding transmission packet sizes, modulation and coding parameters as well as the required signal-to-noise ratios obtained by the physical layer simulations are tabulated in Table 3.1 for the IS-856 system.

Once all user signal-to-noise ratio levels are determined for each time-slot, joint scheduling and video rate adaptation simulations are conducted. Here, the multiple-objective optimization is performed for the objectives of (3.5)-(3.7) to find the best compromise operating point for each time-slot.

To aid the IS-856 system in scheduling, all users need to report their achievable data rate levels every 1.667 ms. Users transmit a 4-bit feedback to describe one of the 13 available data rates as shown in Figure 3.3. In the proposed cross-layer framework, an additional feedback is necessary from each user to aid the base station calculate the remaining video playout time in the buffer of each user. An infrequent 1-bit flag that is transmitted when a user experiences a pause in the playback and then again when the playback is resumed. Since the system is designed to maximize the remaining playback time in the buffer of each user the probability of a pause in the playback is small and thus, for practical purposes, the amount of additional feedback necessary is very small. This statement is confirmed with the simulation results that are presented in the next section.

| Rate (kbps) | Slots | Transmission Packet Size (bits) | Modulation | Coding Rate | SNR (dB) (Ec/Io) |
|---|---|---|---|---|---|
| 38.4 | 16 | 1024 | QPSK | 1/5 | -11.68 |
| 76.8 | 8 | 1024 | QPSK | 1/5 | -9.31 |
| 153.6 | 4 | 1024 | QPSK | 1/5 | -6.14 |
| 307.2 | 2 | 1024 | QPSK | 1/5 | -2.96 |
| 614.4 | 1 | 1024 | QPSK | 1/3 | -0.77 |
| 307.2 | 4 | 2048 | QPSK | 1/3 | -3.94 |
| 614.4 | 2 | 2048 | QPSK | 1/3 | -0.88 |
| 1228.8 | 1 | 2048 | QPSK | 1/3 | 3.55 |
| 921.6 | 2 | 3072 | 8-PSK | 1/3 | 1.58 |
| 1843.2 | 1 | 3072 | 8-PSK | 1/3 | 7.73 |
| 1228.8 | 2 | 4096 | 16-QAM | 1/3 | 3.62 |
| 2457.6 | 1 | 4096 | 16-QAM | 1/3 | 11.19 |

Table 3.1: Required SNR values for the IS-856 system.

### *3.3.2 System Performance with No Video Rate Adaptation*

We first consider a system with no video rate adaptation. In this scenario, each user may view a different video, where playback starts after an initial pre-roll delay, i.e., after a user receives 6 seconds of video. We assume all videos are encoded at a constant average bitrate. We simulate the average and worst case number of pauses per playback second, $PN$, as well as the average and worst case total wait-times, $T_w$, for 32 active users, each with a buffer size of 1000 kbits.

Results, obtained for the proposed system as well as the state of the art schedulers for various average video coding rates are shown in Figures 3.4 and 3.5, for the ITU Pedestrian A and Vehicular B channels, respectively. The buffer size constraint of (3.8) is applied to all schedulers such that a scheduled user is not served if its buffer is already full. The schedulers select the second ranked user in this case. In both channel scenarios, the pro-

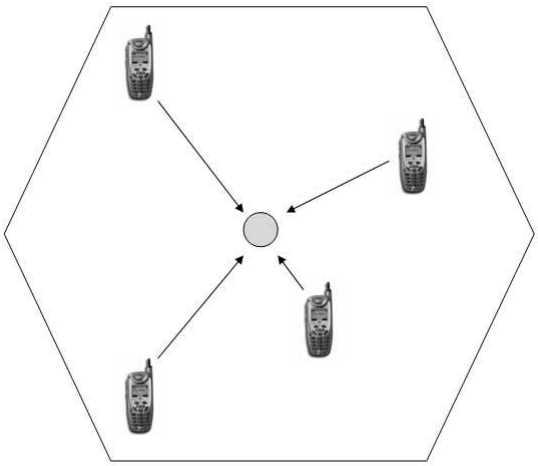


Figure 3.3: All users provide feedback (channel status) to the base station.

posed multiple-objective optimized scheduler outperforms all others in both the number of pauses and the total wait-time significantly. In fact, for video transmissions of up to 60 kbps, the average number of pauses observed using the proposed scheduler is nearly zero. The number of pauses for the worst behaving user in this case is only 2 over the course of a 183 second video. For video rates of 80 kbps, the average number of pauses is 44% and 72% of that of the second-best scheduling algorithm for the pedestrian and vehicular channels, respectively. Similarly, the average total wait-time for the same average video rate is 52% and 78% of that of the second-best scheduling algorithm for the pedestrian and vehicular channels, respectively. More importantly, the proposed framework provides streaming video specific QoS enhancements without sacrificing the overall system throughput, where we obtain an 11% improvement for both vehicular and pedestrian channels when compared to the second-best scheduling algorithm. Table 3.2 provides values also for the system goodput which is defined as the net data rate used for video transmission. The goodput excludes the headers and frame-fill inefficiencies from the system throughput.

### *3.3.3 System Performance with Video Rate Adaptation*

We assume that rate adaptation for videos is achieved by switching amongst 12.5 frames-per-second (fps) pre-encoded bit streams at mean rates of 50, 60, 70 and 80 kbps. Switching among different bit streams is possible every 12th frame, i.e., in 0.96 second periods. In this

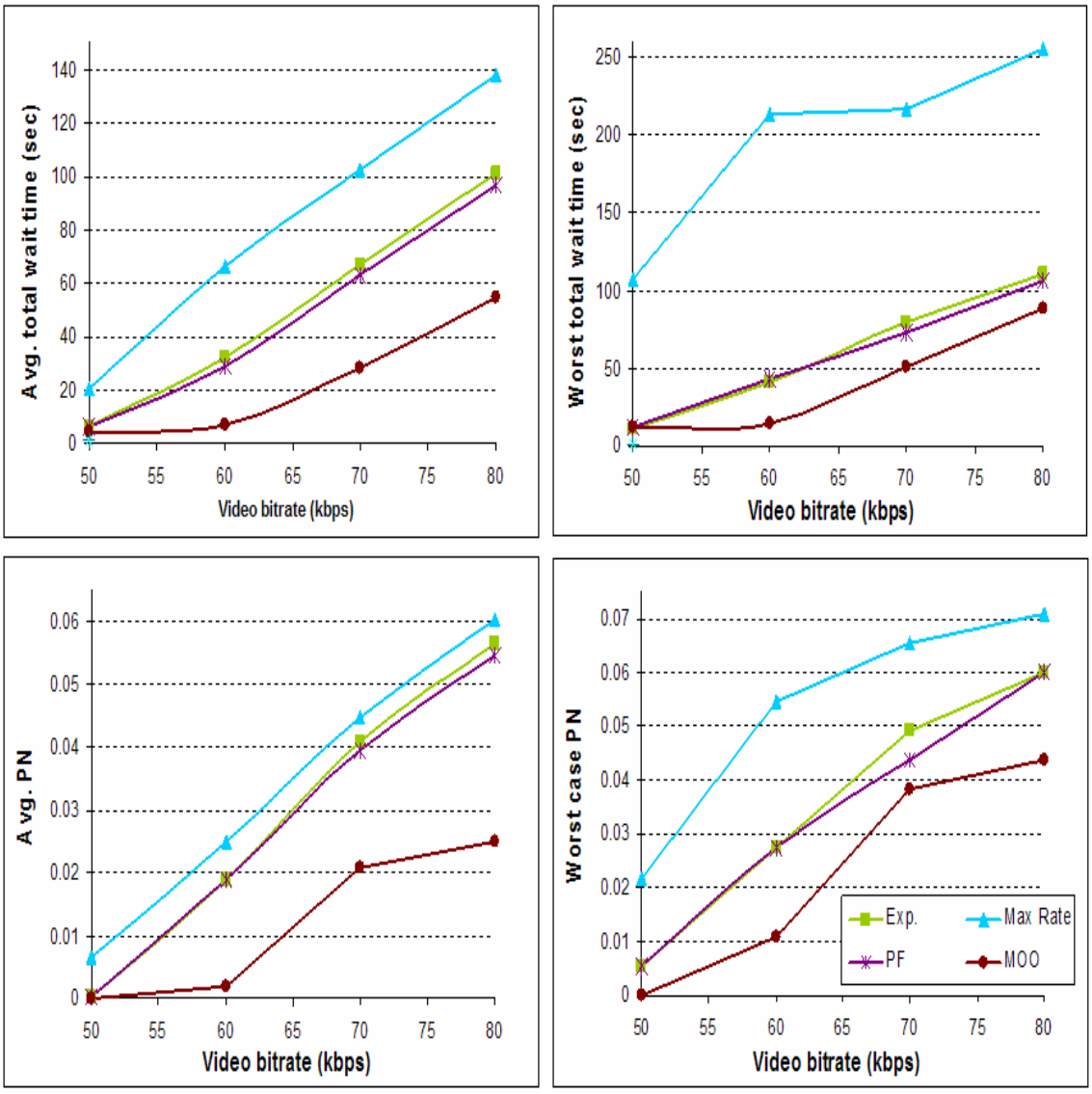



Figure 3.4: Average and worst case total wait time and number of pauses per play-second (PN) computed over all 32 users vs. constant video rate for ITU Pedestrian A environment.

scenario, we repeat the simulations described in the previous section to compute the average and worst-case number of pauses and total wait-time. The results are tabulated in Table 3.2 comparing the performances of the proposed framework with and without video rate adaptation to those of the traditional schedulers from the literature.

We observe that video rate adaptation further improves the performance of the proposed framework over the case with no rate adaptation. For example, for average video rates of 60 kbps and 50 kbps, video rate adaptation results in an average number of pauses that is 50% and 89% of that of the non-adaptive scheme, for the pedestrian and vehicular channels, respectively. Similarly, the average total wait-time figures for the same average video rates are 75% and 99% of that of the non-adaptive scheme for the pedestrian and vehicular channels, respectively. Rate adaptation also results in a further 10% increase of the system throughput over the non-adaptive system for the pedestrian channel. For the vehicular channel, no further gain is observed.

The PSNR levels of the received videos by the 32 users have a mathematical average of 31.12 dB with a standard deviation of 0.065, for the Pedestrian A environment. The received video PSNR for the best and the worst users are 31.24 dB and 30.99 dB, respectively. Thus, one can conclude that the proposed framework succeeds in providing application-level fairness for the streaming video service among all users.

### *3.3.4 Sensitivity Analysis*

The optimum operating point in a MOO problem with two objective functions (described as $s^0$ in Appendix A) is a pair $(i,j)$ where $i$ denotes the user index and $j$ denotes the associated video coding rate. Associated with the operating point is a triplet of values for the objectives, namely, the video coding rate, the remaining playback time and the video throughput. To assess the sensitivity of these values to departures from the optimum operating point we first rank all operating points with increasing distances from the utopia point. If the sensitivity analysis is to be conducted for the video coding rate objective, then we define the operating points that are nearest ranked to the optimal point and having a larger or smaller video coding rate as $s^1$ and $s^{-1}$, respectively. Obviously if one of these points were to be employed instead of the optimum point, the overall system performance will change. The results, tabulated in Table 3.3, are obtained for the ITU Pedestrian A

| Environment: ITU Pedestrian A | Avg. video rate: 60 kbps, Initial buffer: 6 video seconds, Buffer size: 1000 kbits | | | | | |
|---|---|---|---|---|---|---|
| Scheduler | Avg. Tw (sec) | Max. Tw (sec) | Avg. PN | Max. PN | Capacity (kbps) | Goodput (kbps) |
| MOO with rate adaptation | 5.3024 | 13.8067 | 0.0010 | 0.0055 | 2183.2 | 1939.1 |
| MOO with CBR video | 7.0929 | 14.0800 | 0.0020 | 0.0109 | 2145.2 | 1901.7 |
| Proportionally Fair | 29.1139 | 41.3967 | 0.0188 | 0.0273 | 1902.5 | 1662.7 |
| Exponential | 32.2392 | 40.6050 | 0.0213 | 0.0273 | 1876.7 | 1639.6 |
| Maximum Rate (C/I) | 68.5008 | 214.500 | 0.0264 | 0.0546 | 1736.4 | 1507.1 |
| Environment: ITU Vehicular B | Avg. video rate: 50 kbps, Initial buffer: 6 video seconds, Buffer size: 1000 kbits | | | | | |
| Scheduler | Avg. Tw (sec) | Max. Tw (sec) | Avg. PN | Max. PN | Capacity (kbps) | Goodput (kbps) |
| MOO with rate adaptation | 24.6716 | 39.3533 | 0.0177 | 0.0327 | 1632.1 | 1428.1 |
| MOO with CBR video | 24.7248 | 39.4583 | 0.0199 | 0.0327 | 1632.9 | 1429.1 |
| Proportionally Fair | 48.2845 | 53.2200 | 0.0319 | 0.0327 | 1501.7 | 1280.6 |
| Exponential | 50.0746 | 54.2767 | 0.0326 | 0.0382 | 1489.8 | 1270.1 |
| Maximum Rate (C/I) | 73.5168 | 149.500 | 0.0331 | 0.0491 | 1416.4 | 1204.3 |

Table 3.2: Performances of various schedulers.

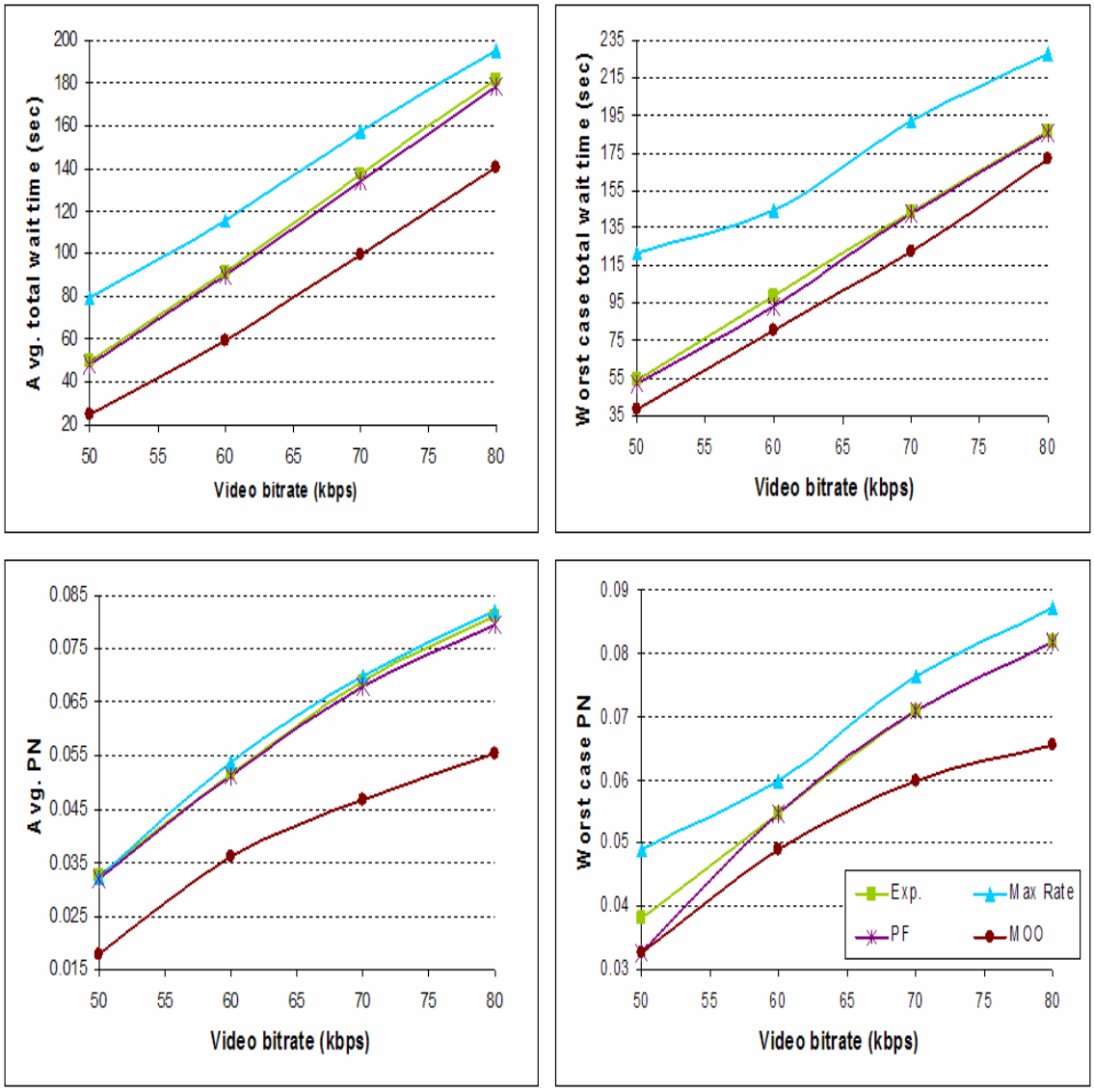



Figure 3.5: Average and worst case total wait time and number of pauses per play-second (PN) computed over all users vs. constant video rate for ITU Vehicular B environment.

| Sensitivity Item | Decision Point | Avg. video rate (kbps) | Avg. video PSNR (dB) | Avg. Tw (sec) | Avg. PN | Channel Capacity (kbps) | Goodput (kbps) |
|---|---|---|---|---|---|---|---|
| *Optimal Solution* | $s^0$ | 59.4354 | 30.99 | 5.3024 | 0.0010 | 2183.2 | 1939.1 |
| *Video Throughput* | $s^{-1}$ | 59.6684 | 30.95 | 77.2666 | 0.0164 | 1483.3 | 1290.3 |
| | $s^1$ | 61.2491 | 30.92 | 15.1865 | 0.0060 | 2162.0 | 1917.8 |
| *Remaining Play Time* | $s^{-1}$ | 62.9041 | 31.18 | 10.5974 | 0.0065 | 2240.7 | 1995.3 |
| | $s^1$ | 50.2418 | 30.35 | 32.3995 | 0.0232 | 1565.4 | 1357.4 |
| *Video Rate* | $s^{-1}$ | 57.1756 | 30.82 | 4.7188 | 0 | 2147.3 | 1903.3 |
| | $s^1$ | 69.2657 | 31.56 | 982.9725 | 0.0039 | 1110.2 | 963.97 |

Table 3.3: Sensitivity analysis.

environment when the sensitivity to changes of the video coding rate is investigated. It is observed that if the provider is more interested in reducing the number of pauses rather than providing a very high PSNR, it may choose $s^{-1}$ as the operating point which results in a 0.26 dB per user video quality loss on average. In return, a zero average number of pauses is achieved.

### *3.4 Conclusions*

In this chapter we present a cross-layer optimized video adaptation and user scheduling scheme for wireless video streaming over packetized networks aiming for maximum video throughput, maximum user QoS, as well as video QoS fairness. We optimize the application and physical layer objectives jointly using a Multiple Objective Optimization framework that aims to serve the user with the least remaining playback time, highest video quality and the highest video throughput. The proposed framework may be used with or without video

coding rate adaptation.

Simulations conducted using the IS-856 numerology over ITU Pedestrian A and Vehicular B channels show that the proposed system without video rate adaptation achieves significant improvements over the state-of-the-art wireless schedulers in terms of user QoS and application-layer QoS fairness. These gains are achieved without sacrificing the overall system throughput; on the contrary, the proposed framework provides gains on the throughput as well when compared to the schedulers that are considered.

When the system is allowed to use video coding rate adaptation, we observe further gains in the overall system performance. The proposed video adaptation algorithm is able to track long term changes in the pedestrian environment well and gains in all three objectives are observed. However, these changes are very fast in the vehicular environment and thus the gains achieved by video adaptation are less pronounced.

The proposed framework runs in real-time and requires a modest increase in the size of the feedback that is regularly sent by each user. However, this increase is negligibly small for the video data rates considered in this paper.

Chapter 4

# MULTIPLE OBJECTIVE OPTIMIZATION FOR STEREO VIDEO STREAMING

This chapter addresses efficient compression and real-time streaming of stereoscopic video over the current Internet. We first propose content-adaptive stereo video coding (CA-SC), where additional coding gain, over that can be achieved by exploiting only inter-view correlations, is targeted by down-sampling one of the views spatially or temporally depending on the content, based on the well-known theory that the human visual system can perceive high frequencies in 3D from the higher quality view. We also developed stereoscopic 3D video streaming server and clients by modifying available open source platforms, where each client can view the video in mono or stereo mode depending on its display capabilities. The performance of the end-to-end stereoscopic streaming system is demonstrated using subjective quality tests.

## *4.1 Introduction*

The average Internet connection has become much faster than it used to be over the last few years due to the tremendous developments in the physical backbone, especially in the developed countries. This progress has made new, more advanced and more interesting services possible for the Internet service providers resulting in higher user satisfaction. Among these new services, live and on-demand web based video streaming services such as Internet TV, video conference, and video databases like YouTube [1] and Google Video [2] are becoming more and more popular these days. The recent advances in video coding techniques [5, 4] provide higher compression efficiency and increased bandwidth provided by Internet Service Providers (ISP's) make this kind of service easier to distribute, which attracts even more interest from the users. The brutal competition among different streaming video providers leads to increased quality of such service.

Scientists from all over the world are currently trying to evolve the video streaming

technology into the most realistic and the most popular remote user experience ever by introducing the third dimension (3D) into it [70] where the depth feeling is added to the conventional 2D video. This obviously means increased data rates for high quality compressed content, which comes at the expense of further bandwidth and delay requirements. Therefore, network bandwidth and delay are still important constraints for today's and future's video streaming applications.

The simplest kind of multiview video, i.e. the stereoscopic video, consists of two video sequences captured by closely located (similar to the distance between two eyes) cameras, i.e. the bandwidth requirements of a stereoscopic video is much higher than the monoscopic video. However, the close distance between the cameras results in high redundancy between the two views. Thus an efficient coding scheme can be developed by exploiting the redundancies between these two views.

There are many research and standardization activities for stereoscopic video compression based on exploiting inter-view redundancy. Early work in this area resulted in the MPEG-2 multi-view profile [16]. Later [71] propose modifications to MPEG-2 multiview profile for improving the compression efficiency again based on the correlation between two views. Recently, new stereoscopic video codecs based on H.264 are introduced in [72, 73, 74]. A new standard for multi-view video coding (MVC) is currently under development under the auspices of Joint Video Team (JVT) [75]. This paper proposes making use of the psycho-visual redundancy depending on the characteristics of the video content in order to achieve additional compression efficiency.

In monoscopic video compression, it is a common practice to sub-sample the chrominance channels, since the HVS is less sensitive to variations in chrominance values. Similarly, in the theory of stereo perception, it is reported that the HVS can perceive high frequency information in 3D from one of the views even if the other view is low pass filtered [76]. Hence, spatial and temporal subsampling can be performed to reduce the bandwidth requirements. To this effect, we propose a content adaptive approach for temporal and spatial downsampling of one of the views to achieve better compression with higher perceptual quality in our H.264/AVC based multi-view codec.

We also propose an end-to-end stereoscopic 3D video streaming architecture using the proposed content-adaptive multi-view coding and modifications to available open source monoc-

ular streaming platforms. There are several open source monocular video streaming platforms, including the Darwin Streaming Server [77], GPAC [78] and VideoLAN Client/Server [79]. Apple QuickTime Streaming Server (QSS) and its open source version Darwin Streaming Server (DSS) supports streaming of H.264 [80] coded video wrapped inside MPEG-4 [81] or 3GPP file format across the Internet using RTSP and RTP protocols [82]. In order to stream media over RTP, these systems need special information about the media files, which is carried in a hint track. Another project called GPAC, which is developed as a multimedia framework based on MPEG-4 standard, also supports streaming H.264 coded media files inside MPEG-4 file format [78]. In addition to these two platforms, VideoLAN Client (VLC) also provides streaming capabilities. VLC supports H.264 video in local playback and streaming when encapsulated in MPEG-TS file format over RTP. Recently, a 3DTV prototype system, similar to our system, with real-time acquisition, transmission and autostereoscopic display of dynamic scenes has been offered by MERL. Multiple video streams are encoded and sent over a broadband network. The 3D display shows high-resolution stereoscopic color images for multiple viewpoints without special glasses. This system uses light-field rendering to synthesize views at the correct virtual camera positions [83].

The rest of the chapter is organized as follows: In Section 4.2, we present the proposed content-adaptive multi-view video encoding. Section 4.3 describes the overall streaming system in detail. Results are given in Section 4.4, and stereo multiple-objective optimization formulations are proposed in Section 4.5. Finally, Section 4.6 provides our concluding remarks.

## *4.2 Content-Adaptive Stereo Video Coding*

There are different theories about the effects of unequal bit allocation between left and right video sequences, such as the fusion theory and suppression theory [84, 85, 18]. According to fusion theory, the stereo bitrate (hence distortion) needs to be equally allocated between the views for the best human perception. Contrarily, according to suppression theory, the highest quality view in a stereo-video determines the overall perception performance. Therefore, the target (right) sequence can be compressed as much as possible to save bits for the reference (left) sequence, so that the overall perceived distortion is the lowest. The proposed content-adaptive stereo codec (CA-SC) is motivated by the suppression

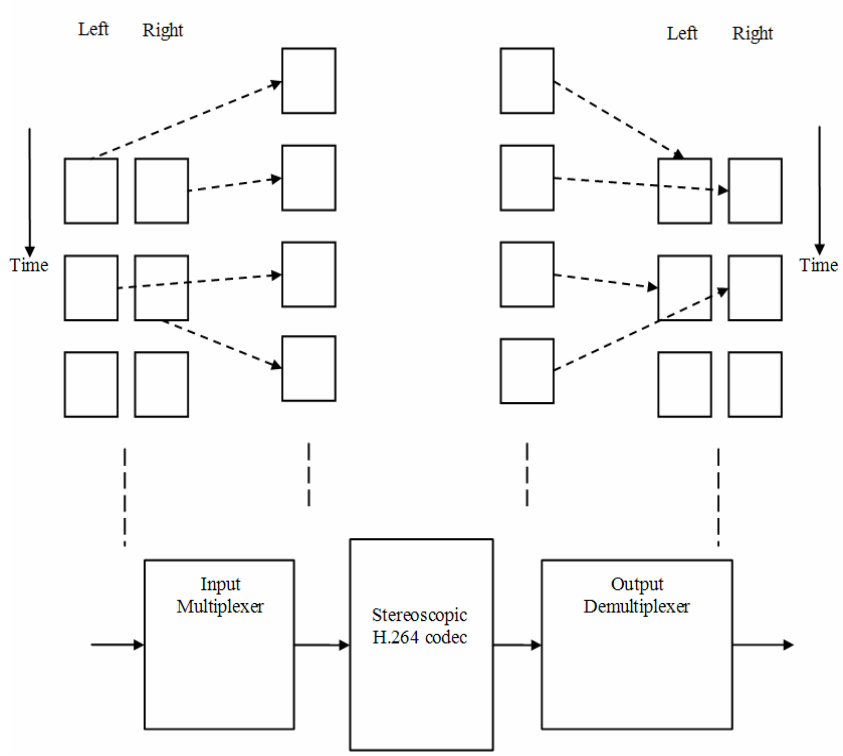



Figure 4.1: Stereoscopic encoder

theory and reduces the frame (temporal) rate and spatial resolution of the target (right) sequence adaptively according to its content-based features.

The principle behind content adaptive video coding is to parse video into temporal segments. Each temporal segment can be encoded at different spatial, temporal and SNR resolution (hence at a different target bitrate) depending on its low and/or high-level content-based features. Even though this approach has been used for monoscopic video encoding [86, 29, 28, 33], there are no such studies in the literature for content-adaptive stereoscopic coding. The proposed CA-SC is an extension of the stereo codec (SC) in [74] which is based on AVC/H.264. We note that CA-SC can also be developed as an extension of the recently standardized MVC codec [75]. The codec structure is shown in Figure 4.1.

In stereoscopic coding, in the compatible mode, any standard H.264/AVC decoder can decode the sequence as a monoscopic sequence since left channel is coded independent of the right channel. In order to improve the coding efficiency without significant perceptual quality loss, we added three modes to the encoder for down-sampling the right-view only:

They are the spatial, temporal, and content-adaptive scaling modes.

### *4.2.1 Spatial Scaling*

The spatial scaling mode corresponds to downsampling the right view by a predefined scale prior to encoding. The implementation of downsampling consists of both decimation and low-pass filtering in order to prevent aliasing. For spatial scaling the following filters are used:

*13-tap downsampling filter:*

{ 0,2,0,-4,-3,5,19,26,19,5,-3,-4,0,2,0 } /64

*11-tap upsampling filter:*

{ 1, 0,-5, 0, 20, 32, 20, 0,-5, 0, 1 } /64

Filters are applied to all Y, U, and V channels and in both horizontal and vertical directions. The picture boundaries are padded by repeating the edge samples. These filters are currently used in Scalable Video Coding extension of H.264/AVC [87] and explained in [88]. In order to keep filtering process simpler in both encoder and decoder, we have implemented downscaling by factors of 2 (dyadic sampling) in both dimensions. Although the spatial scaling is applied to the right view only in our stereoscopic codec, during the motion estimation left frames are also scaled the same amount for proper estimation.

### *4.2.2 Temporal Scaling*

Temporal scaling mode corresponds to the decimation of the right view in time, i.e. frame dropping in the right sequence. The implementation of temporal downsampling is done by sending all the macro-blocks of dropped frame as skipped mode of the H.264 standard. In our codec notation, temporal scaling of n denotes encoding 1 frame out of n frames and dropping the remaining n-1 frames.

### *4.2.3 Content Adaptive Scaling*

In content adaptive video scaling, we first divide the right video into temporal segments (shots or sub-shots) using well-known temporal segmentation methods [89]. We then classify the temporal segments (shots) into 4 categories as determined by their low-level attributes such as the amount of motion and spatial detail within the segment. Shots with high

temporal activity (high motion) need to be encoded at full temporal resolution for a smooth viewing experience. On the other hand, if a somewhat stationary shot is being encoded, the temporal sampling rate can be reduced to a lower value without any loss of perceptual quality. Likewise, shots with high spatial detail should not be reduced to lower spatial resolutions for the sake of perceptual quality, while it is harmless to do such downsampling in case of low spatial detail.

There have been several studies on temporal content adaptation and classification in the literature over the past years [90, 89], which can be used to determine which temporal region belongs to what spatial and temporal class in any given video segment. The spatio-temporal attributes may vary even from GoP to GoP within a shot. In the following, we define *Spatial Scene Complexity* and *Temporal Activity* measures for the classification of temporal segments.

The *Spatial Scene Complexity Measure* is calculated as the pixel variance across the temporal segment.

Assume that the number of frames (either left or right sequence) in a specific stereoscopic video temporal segment is $N$. Let the pixel resolution of each frame be $H\,W$ where the symbol $H$ denotes the height and the symbol $W$ denotes the width of the frames in the number of pixels. In order to calculate the spatial complexity measure of a single frame $i$, we first find the mean of horizontal and vertical squared pixel value differences as follows:

$$E\left[d_i^2\right] = \frac{\sum_{h=0}^{H-2}\sum_{w=0}^{W-1}\left\{\left(p_i(h+1,w)-p_i(h,w)\right)^2\right\} + \sum_{h=0}^{H-1}\sum_{w=0}^{W-2}\left\{\left(p_i(h,w+1)-p_i(h,w)\right)^2\right\}}{W(H-1)+H(W-1)}$$

where $h$ represents the vertical pixel coordinates and $w$ represents the horizontal pixel coordinates in within the frame. We then calculate the square of the mean pixel difference as indicated by:

$$E\left[d_i\right]^2 = \left(\frac{\sum_{h=0}^{H-2}\sum_{w=0}^{W-1}\left(p_i(h+1,w)-p_i(h,w)\right) + \sum_{h=0}^{H-1}\sum_{w=0}^{W-2}\left(p_i(h,w+1)-p_i(h,w)\right)}{W(H-1)+H(W-1)}\right)^2$$

Therefore, the pixel variance within the frame can be calculated as follows:

$$\sigma_i^2 = E\left[d_i^2\right] - E\left[d_i\right]^2$$

Hence the scene complexity, $\sigma$, of the temporal segment can be calculated as the average pixel variance across its frames:

$$\sigma = \frac{1}{N} \sum_{i=0}^{N-1} \sigma_i^2$$

A segment is classified as having low spatial scene complexity if this measure is below some threshold, and as having high spatial scene complexity otherwise. The *Temporal Activity Measure* is calculated for each frame by taking the average absolute motion vector length (encoder output) in a temporal segment. The intra block codings within inter-coded frames (P or B frames) are considered to have the maximum motion vector length within the allowable search range. The intra coded I-frames are left out of this temporal activity measure computation. After this, the results found are scaled by their respective spatial resolutions in order to be consistent across different stereoscopic videos.. A shot is classified as having low temporal activity, if this measure is below a threshold; and classified as high temporal activity if it exceeds the threshold. Therefore, one can classify temporal video segments into four categories with specific low-level temporal and spatial attributes as shown in Figure 4.2.

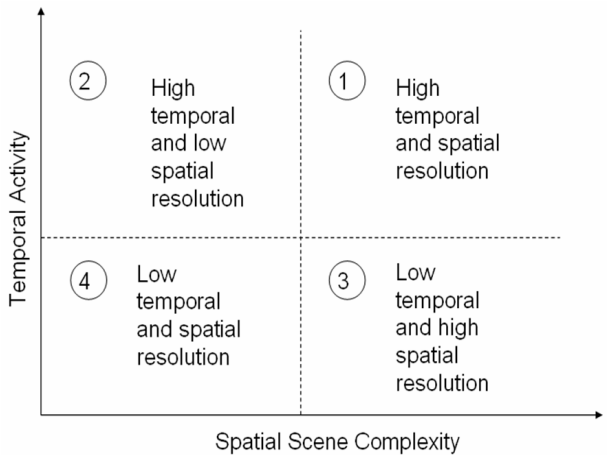



Figure 4.2: Temporal and spatial formats appropriate for the right view according to low-level features.

Here the appropriate spatial and temporal formats for the right view only belonging to each class are as follows:

- Type 1: High Spatial and Temporal Activity: Do not scale the spatial and temporal formats

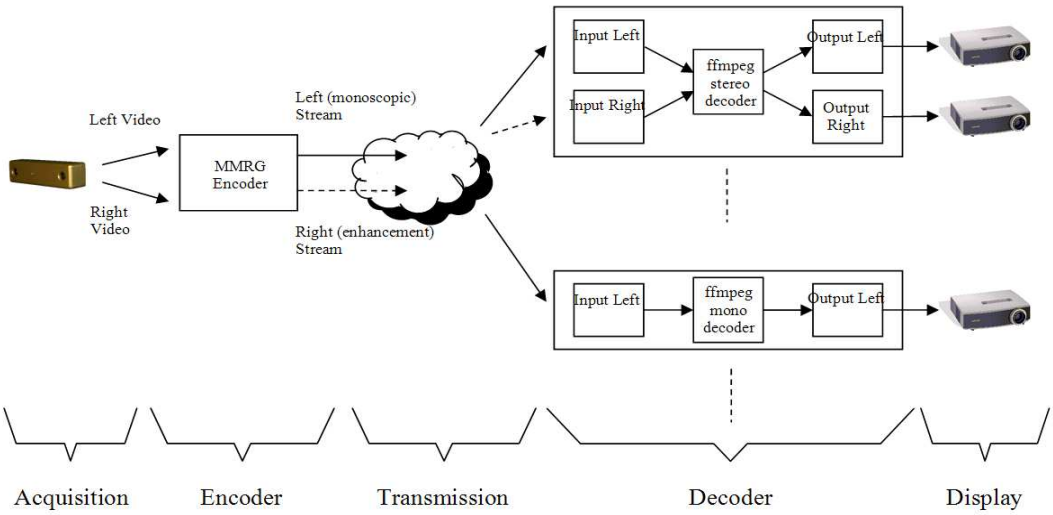



Figure 4.3: End-To-end system overview

- Type 2: Low Spatial and High Temporal Activity: Apply spatial scaling but not temporal scaling
- Type 3: High Spatial and Low Temporal Activity: Apply temporal scaling but not spatial scaling
- Type 4: Low Spatial and Temporal Activity: Apply both temporal and spatial scaling

## *4.3 End-to-End Stereo Video Streaming System Overview*

An overview of the proposed system architecture is shown in Figure 4.3. Stereo video is encoded off-line by using the proposed content-adaptive SC, which has been described in Section 4.2. The server, which is detailed in Section 4.3.1, streams the encoded bitstream over the Internet. The end users can view either monoscopic or stereo streams based on their display capabilities using the client, as described in Section 4.3.2.

### *4.3.1 Server*

The server employs the protocol stack RTP/UDP/IP, and can serve multiple clients, described in Section 4.3.2, simultaneously. Session description protocol (SDP) is used to ensure interoperability with the clients.

Since the proposed CS-SC encoder is a modification of the H.264/MPEG-4 AVC design, it supports a Video Coding Layer (VCL), which efficiently represents the video content, and a Network Abstraction Layer (NAL), which provides header information for particular transport layers (such as Real Time Transport Protocol) or storage media. All data are contained in NAL units, each of which contains an integer number of bytes. The format of NAL units is the same for both packet-oriented transport and bitstream delivery. The only difference is each NAL unit can be preceded by a start code prefix in a bitstream-oriented transport layer [80].

The encoded stereo bitstream contains NAL units of both left and right views. These NAL units are packetized for independent streaming over two separate channels using Real-time Transport Protocol (RTP) [82]. The sender side packet format is implemented based on the RTP payload format for H.264 video [91]. Three packetization modes are defined in this payload format. We implemented Single NAL Unit mode and Non-Interleaved mode which are intended for low-delay applications. We used FU-As (Fragmentation Unit without Decoding Order Number) packetization structure to transfer NALUs the sizes of which are exceeding the network MTU. We fragmented the NALUs on the application layer instead of relying on the IP layer fragmentation. Other packets with smaller sizes are sent in Single NAL unit packets.

In both of these packetization modes, the transmission order of the RTP packets shown by the sequence numbers is taken as the decoding order of the NAL units. Since our encoder does not support B frames, packet structures, which do not contain decoding order numbers are usable in our application. Timestamps carried in the RTP header are used to determine the decoding order of the frames. The RTP timestamps are used to synchronize the frames. The display application arranges the play-out time by using the relative order of the frames positioned by the RTP timestamps. Since we stream two video files, we set related frames to the same timestamp supposing same sampling rate for videos with a 90 kHz clock. In addition, the H.264 parameter sets are fundamental parts for video coding. A more reliable transfer is required for their transmission and receiver must receive them before the decoding process. So we transfer them out-of-band to the receiver side reliably prior to the actual RTP sessions.

For the interoperability of the stereo video server and the client on the receiver side,

we used session description protocol (SDP) [92]. For indicating stereo view, an additional session attribute is used in order to specify stereo data and which channel is the left and which one is the right. Moreover, for future extension of stereo streaming to multi-view streaming, the session descriptor also can be used. Currently, we define a new attribute "view" which gives the address and the port information of the other sessions broadcasting extra views of the video.

*a=view : mono*

*a=view : stereo < address-Left > < port-Left >*

*< address-Right > < port-Right >*

*a=view : multi < address > < port > < address > < port > ,*

*< address > < port > < address > < port > ,*

*< address > < port > < address > < port > , ...*

*w*here "mono" for monoscopic, "stereo" for stereoscopic and "multi" for multi-view gives the view type and " < address > < port > < address > < port > " pair gives the access information of two corresponding views of the multi-view video.

### *4.3.2 Clients*

We have implemented three clients for different types of display systems: i) Client-1 supports an in-house polarized 3D projection display system; ii) Client-2 supports the autostereoscopic Sharp 3D laptop, iii) Client-3 supports a monocular display to demonstrate backwards compatibility. For all client implementations, we modified the open source software VideoLAN Client (VLC). VLC is a highly portable multimedia player for various audio and video formats and streaming protocols [79]. We modified its stream receiver to support raw H.264 streaming over RTP and used it as a player. The modified VLC handles packets of left and right views using two separate threads. Then, corresponding decoder for H.264 coded data is opened by the player. We send NALU units received in RTP packet payload directly to the decoder after de-packetization.

As the decoder, open source H.264 decoder implementation in the FFmpeg library [93] is used with MVC modifications. Before sending to the decoder, the data is buffered in order to synchronize related left and right frames. The decoder decodes and sends the decoded picture to the video output modules. The video output units visualize the left and right

frames in a synchronized manner by using the time information in the RTP timestamps.

Finally, the in-house 3D projection display system uses a pair of Sharp MB-70X projectors as shown in Figure 4.4. Light from one of the projectors is polarized in clockwise direction and light from the other projector in counter-clockwise direction using circular polarization filters. Both projectors are aligned to project onto a special silver screen covered with a neutral grey reflective dielectric material to preserve the polarization of light during reflection. The users wear glasses which have matching filters with the projectors to ensure that light from one projector is only seen with one eye. This enables us to feed left and right images to left and right eye of the subject to create the illusion of 3D. The projectors were driven by a single high-end PC with two display outputs using the extended desktop feature. This setup results in a virtual desktop of 2048x768 pixels, each projector displaying only one half of the extended desktop at 1024x768 native resolution. Using this setup, left and right videos can be easily shown on the left and right halves of the extended desktop, such that they exactly overlap with each other on the silver screen.

Figure 4.4: The stereoscopic display system

## *4.4 Experimental Results*

### *4.4.1 Subjective Quality Tests*

In order to investigate the effects of spatial, temporal and content adaptive scaling in stereoscopic videos, we employ the DSCQS (Double Stimulus Continuous Quality Scale) Method [94]. In this test method, assessors which are chosen among non-experts and inexperienced assessors should be used. The evaluation should be on a continuous scale ranging from 0 to 100. The method can be applied in two variants:

*Variant1:* Each assessor is let to switch between two conditions, A and B (two stereoscopic images or videos), one of which is always the source and the other is the tested condition applied on the source. The identity of the images, whether it is the source or the test condition, should be known by the experimenter but not by the assessors. After evaluating the conditions the assessor moves to the next pair of images or videos.

*Variant2:* Multiple assessors are shown two conditions, A and B (two stereoscopic images), consecutively one of which is always the source and the other is the tested condition applied on the source. The identity of the images, whether it is the source or the test condition, should be known by the experimenter but not by the assessors. The next pair of conditions is shown after the assessors establish an opinion.

*Analysis Method:* For the analysis of the test results, each evaluation is graded between 0-100 and the difference between the scores of source image and the test condition is calculated to find the score of that test condition on that image by the assessor. After all these scores are calculated, the values are normalized to fit in 0-100. And as a final step, to find the scores of each algorithm (test condition) the average of all the scores over the assessors and images are taken. Scores of the algorithms can be compared with their closeness to the number to which zero score is mapped during the normalization process.

### *4.4.2 Experiments*

In the experiments, we investigated effects of spatial, temporal and content adaptive scaling in stereoscopic videos. In order to meet time requirements of assessment test, we use only 4 video sets with 8 algorithms.

**Assessors:** 21 assessors (13 female, 8 male with average age 24) with ages ranging from

19 to 36, volunteered to participate in the experiment. The participants were non-experts in the area of picture quality and were screened for color vision, stereo depth perception and visual acuity.

Each assessor is well informed on the test process and test materials (possible quality defects) before the test and they are assisted during the whole test procedure. DSQCS test method with the second variant mentioned above is used as the test methodology. At each step two video sequences, original left and right videos and processed left and right videos are used. We will call those 4 videos an evaluation pair. In the experiments, original videos are also repeated as a processed video in order to test the performance of the test.

At the beginning of the test, 5 random evaluation pairs are shown to the assessors and these 5 evaluation pairs are not evaluated since they provide stabilization of the perception of assessors. The test material is shown in a random order for each assessor. The randomization is done both among evaluation pairs and among the set of video sequences in the pair.

**Test Material**: As the test material, four different stereoscopic video pairs are used: balloons (720x480, 25 fps, 10 seconds), botanical (960x540, 15 fps, 5 seconds), flowerpot (720x480, 25 fps, 10 seconds), train_tunnel (720x576, 25 fps, 10 seconds). The temporal activity values for each frame of the tested videos can be seen in figures 4.5, 4.6, 4.7 and 4.8 whereas the spatial scene complexity values are illustrated in Figure 4.9. Eight different algorithms are applied on these videos as shown in Table 4.1.

| | |
|---|---|
| SIMUL | Simulcast coding |
| S1T1 | Stereo coding, no spatial, no temporal scaling |
| S1T2 | Stereo coding, no spatial, temporal scaling 2 for right frames |
| S1T2L | Stereo coding, no spatial, temporal scaling 2 for left and right frames |
| S1T3 | Stereo coding, no spatial, temporal scaling 3 for right frames |
| S2T1 | Stereo coding, spatial scaling 2, no temporal scaling |
| S4T1 | Stereo coding, spatial scaling 4, no temporal scaling |
| S4T3 | Stereo coding, spatial scaling 4, temporal scaling 3 for right frames |

Table 4.1: Algorithms applied to test videos.

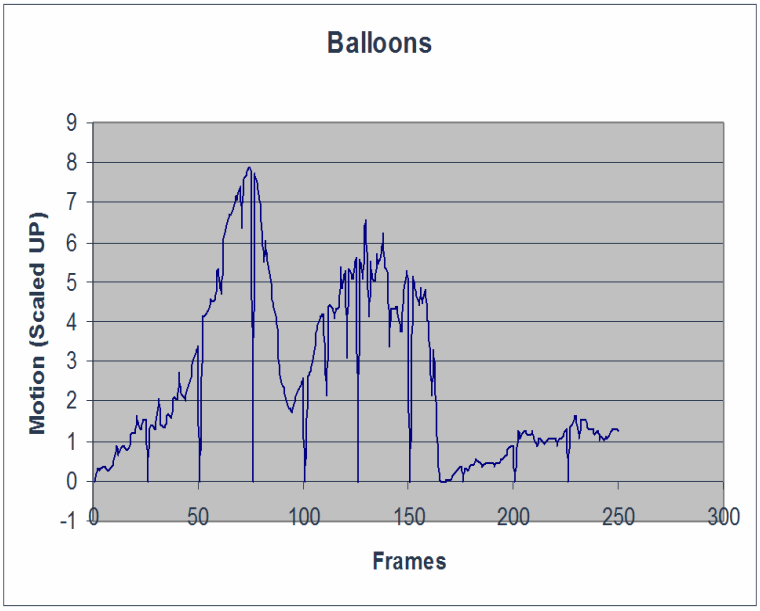



Figure 4.5: The Temporal Activity values for Balloons sequence.

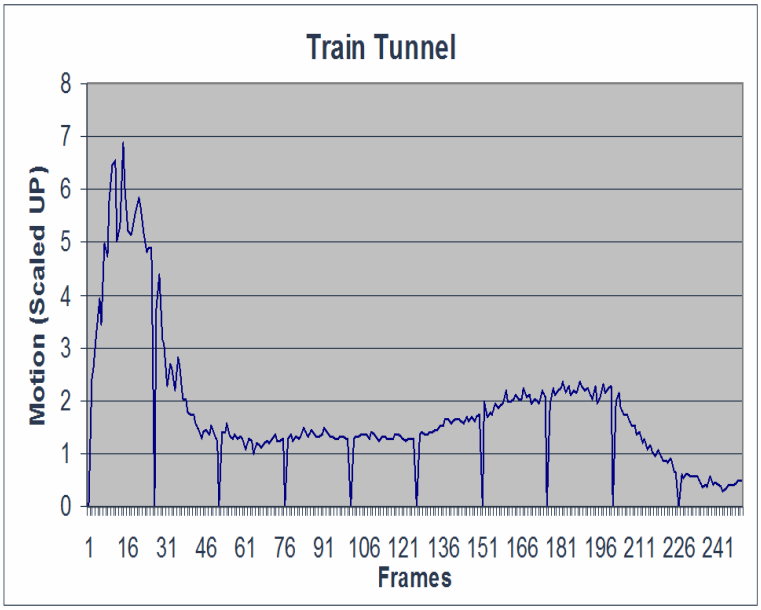



Figure 4.6: The Temporal Activity values for Train Tunnel sequence.

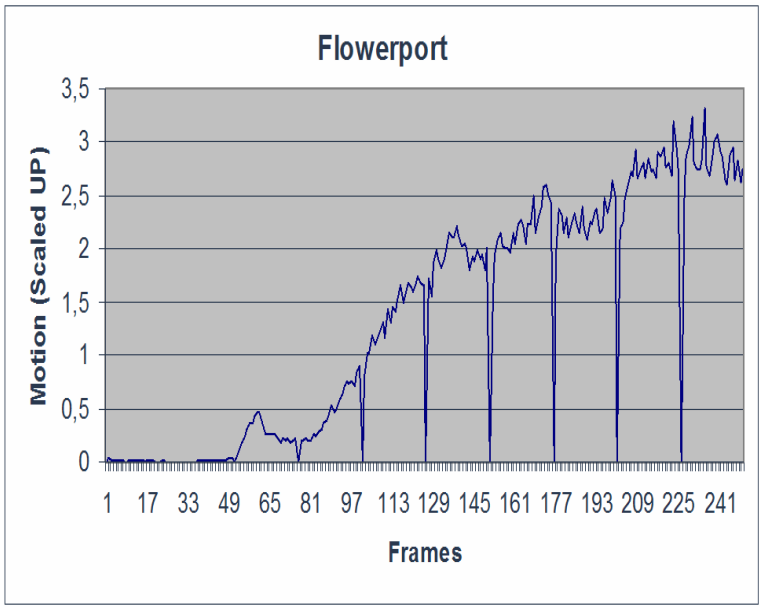



Figure 4.7: The Temporal Activity values for Flowerpot sequence.

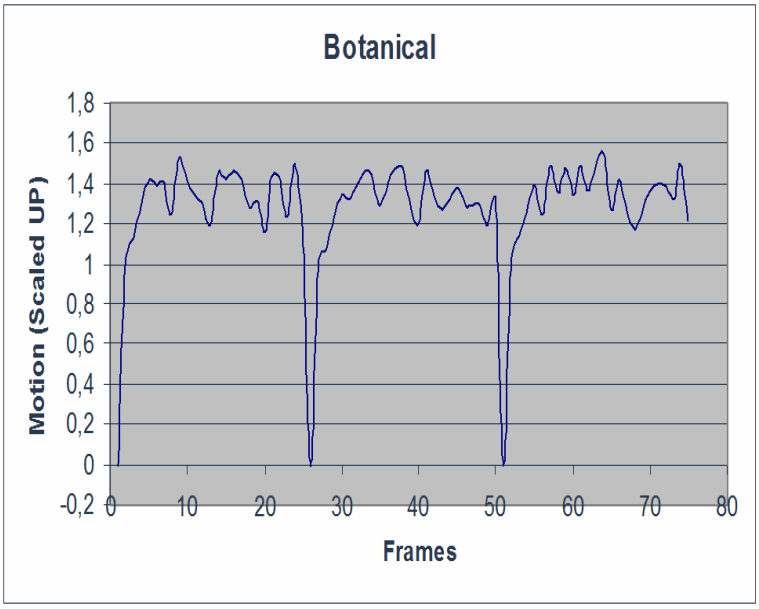



Figure 4.8: The Temporal Activity values for Botanical sequence.

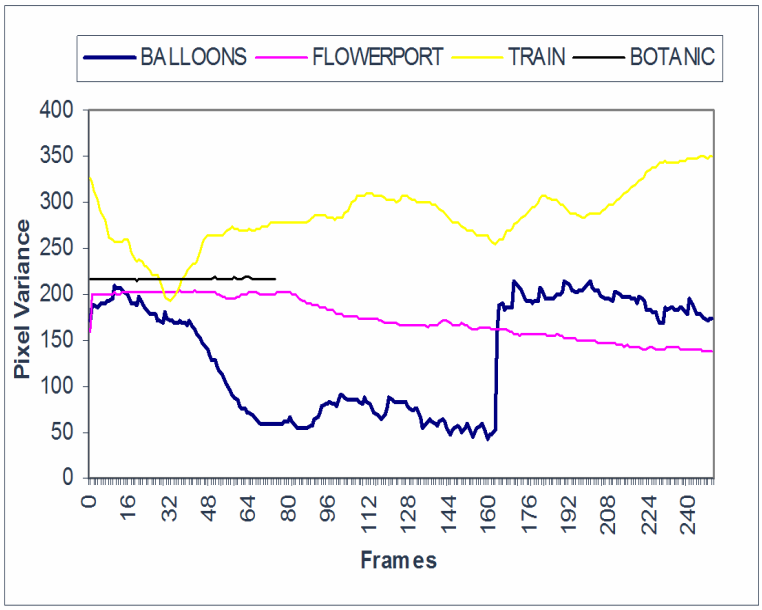



Figure 4.9: Pixel variance (spatial scene complexity) values of each frame of the test videos.

The algorithm ADAP is tested only on the Balloons sequence since its motion vs pixel variance values of GOPs are the most scattered among the other test sequences as in Figure 4.13. As a result, a total of 42 evaluation pairs, including first 5 stabilizing pairs, are shown to the assessors and it is assured that each test does not take more than 30 minutes.

### *4.4.3 Results*

All the test videos are encoded with the modes explained in Table 4.1. Intra period of 25 and Quantization Parameter (QP) of 28 are used while encoding. Total bitrate for simulcast coding is interpreted as twice the data required compared to single view coding and the bit rates of all other algorithms are normalized accordingly and can be found in Table 4.2.

The resulting average PSNR values of the sequences for different algorithms can be seen in Figure 4.10. PSNR values are all in dB and calculated according to the following formulas where $D_l$ and $D_r$ represent the distortions in right and left images [95]:

$$PSNR_l = 10\log_{10}\frac{255^2}{D_l/2}, PSNR_r = 10\log_{10}\frac{255^2}{D_r/2}, PSNR_{all} = 10\log_{10}\frac{255^2}{(D_l+D_r)/2}$$

By only spatial subsampling of right video with 2 in both dimensions we have approximately matched 1.2 times the single view bitrate. By applying both spatial and temporal

| | BALN | FLOW | BOTA | TRAIN | Average |
|---|---|---|---|---|---|
| SIMUL | 2.000 | 2.000 | 2.000 | 2.000 | 2.000 |
| S1T1 | 1.901 | 1.927 | 1.452 | 1.881 | 1.790 |
| S1T2 | 1.606 | 1.692 | 1.289 | 1.601 | 1.547 |
| S1T2L | 1.324 | 1.450 | 0.923 | 1.336 | 1.258 |
| S1T3 | 1.489 | 1.586 | 1.228 | 1.492 | 1.449 |
| S2T1 | 1.242 | 1.267 | 1.065 | 1.252 | 1.207 |
| S4T1 | 1.091 | 1.095 | 1.012 | 1.085 | 1.071 |
| S4T3 | 1.053 | 1.069 | 1.006 | 1.049 | 1.044 |

Table 4.2: Normalized bit rates of the algorithms.

scaling on right frames, we can nearly code the stereoscopic video at single view bit rates.

After all the assessors finish the test, the scores are evaluated and normalized according to [94]. Average MOS for each algorithm and confidence intervals are shown in Figure 4.11. Due to the normalization, 0 (best quality) is mapped to 38, and the success of the algorithms can be measured by closeness of their mean to 38. This mapping is due to wrong evaluations of the assessors giving better scores to the distorted sequences than the original and expected in general. Simulcast (SIMUL) coding and stereo coding without scaling (S1T1) have similar or better performances over original video. Since QP is low, reconstructed video quality is visually lossless (with average PSNR of 36 dB) and misjudgment is expected for these algorithms as well. Also DCT based coded images are reported [96] to be preferred by assessors comparing to original.

We can see that scaling with 3 or 4 in both spatial and temporal domain, results are not acceptable. According to the bitrate and MOS (Mean Opinion Score), only spatial scaling looks like the optimum solution. Spatial scaling by 4 corresponds to 16:1 reduction in image size; therefore its performance is not acceptable. Spatial scaling with non-dyadic factors and better filters for upsampling might keep the visual quality at desired levels with bitrate similar to single view coding bit rates.

According to the video characteristics (slow motion video), temporal scaling in either

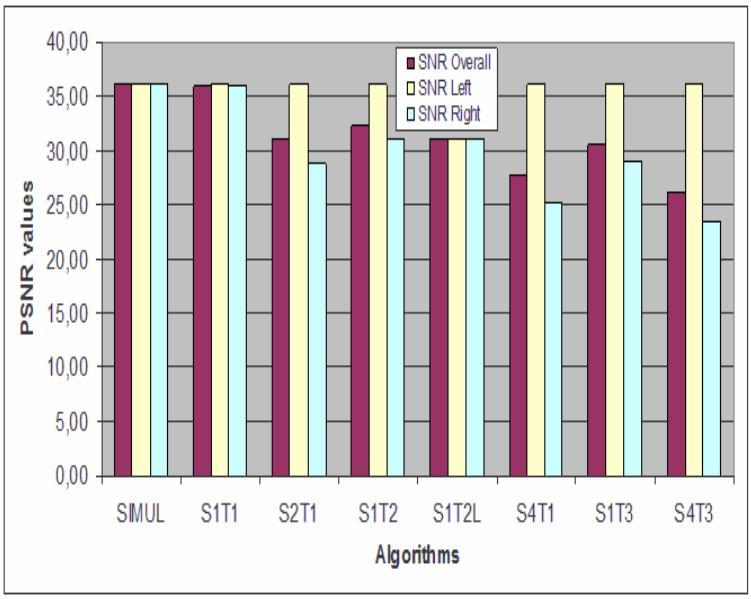



Figure 4.10: PSNR values of the algorithms.

right channel or both channels gives better results (see Figure 4.11).

By analyzing the characteristics of the video in each GOP, appropriate scaling can be applied to decrease bitrate without visual quality degradation. The temporal (motion) and spatial (pixel variance) features of the GOPs of the test sequences can be seen in Figure 4.13.

### *4.4.4 Streaming System Performance*

For transmission and display process, we have implemented all the modules and run the system with already encoded files. In order to cope with packet losses, frequent intra frames are inserted and frames are coded in slice mode. Although the system is tested by encoding each frame as a single slice, number of slices can be increased or fixed size slices can be used according to the network state. The system is initially tried on local area network with zero packet losses. H.264 packet loss resilience techniques and loss concealment techniques will be added to the system for real Internet use.

The H.264/AVC coded video increased the efficiency of bandwidth usage and this also affects the quality of the views. We coded videos as 25 fps and we inserted one intra frame per 12 frame. Two different quantization values were used, one has Y channel limit value

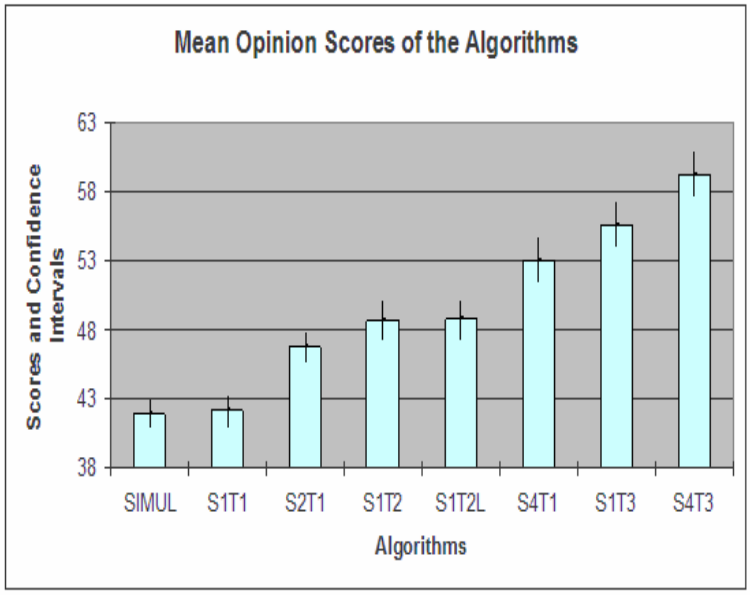



Figure 4.11: Mean Opinion Scores and confidence intervals of the algorithms.

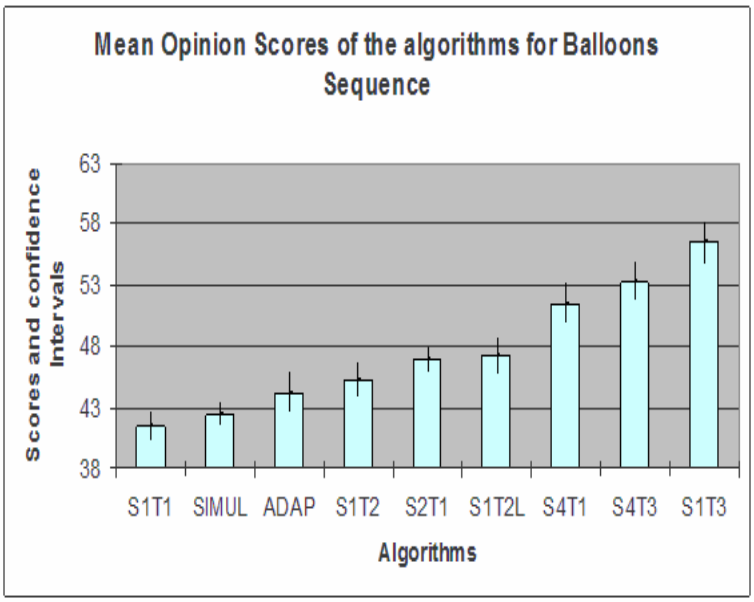



Figure 4.12: Mean Opinion Scores and Confidence intervals of the algorithms including the content adaptive scaling algorithm for Balloons sequence.

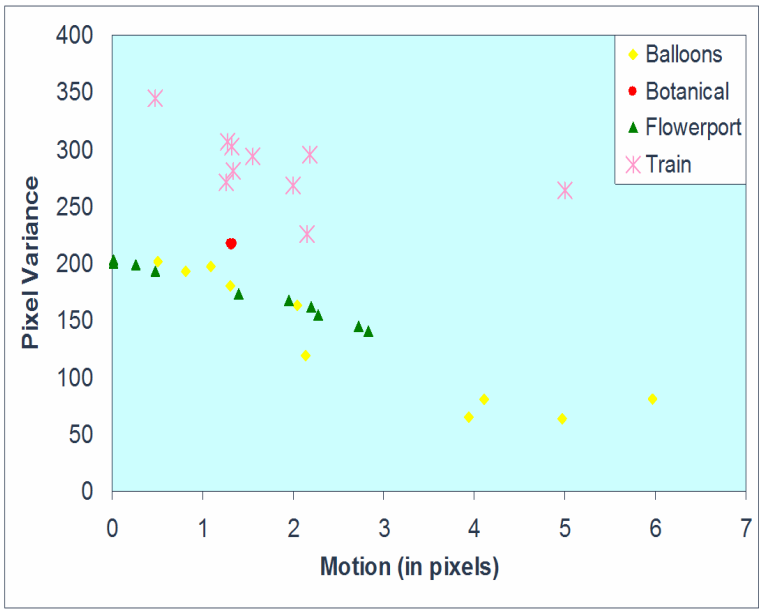



Figure 4.13: Motion vs. pixel variance averages of each GOP for each test sequence.

of 38.27 dB and other one has 33.47 dB. For 320x240 video, their bandwidth usage were 744.665 and 415.335 kbits/sec respectively.

The system scalability was also another factor for the system design. The data can be multicasted from anywhere and the users can view as mono or stereo depending on their connection capacity and display system. Moreover the player functionality and integrity also increases the usage of system with the future improvements on different file formats and codec standards.

### *4.5 Multiple Objective Optimization Formulations for Stereo*

As discussed in Appendix A, multiple objective optimization theme can help us avoid trivial and suboptimal encoding strategies. The gains achieved by spatial and temporal scaling of one view (left or right) in stereo videos demonstrated in this chapter can be further improved by multiple-objective optimization as proposed here.

In a multiple objective optimization problem, it is typical to have multiple variables that determine the nature of the optimization problem. Some of these variables can be used to define the objective functions and the others can be used to define the constraining set. For this reason, we usually have the freedom to come up with various optimization formulations

for the same problem or similar problems. In this section, we list possible content-adaptive rate control optimization formulations for stereo videos.

The basic logic presented in Chapter 2 can be applied to the stereoscopic video, with the difference that it is now also possible to change coding parameters between the two views.

It is a desirable future of a video streaming system to be adaptable in the sense that the video encoding (bitrate allocation) is done according to the coding complexity (difficulty) and/or the content relevance (importance). Assigning a general target bitrate to all parts of a video results in worse user utility in video streaming. This is not only due to the fact that the coding difficulty and complexity vary continuously in the duration of the video, but also because the user's level of interest in each possible type of shots may differ dramatically. The more important and/or complex shots need to be encoded using more number of bits for the sake of visual quality. Therefore, in order to increase the overall user utility, shot classification should also take part at the encoder side, which in return increases the number of optimization variables to be used considerably.

*Formulation 1:*

In the simplest case, where the left and right images are treated equally in terms of their contribution to the overall distortion measure, the following formulation can be used:

$$\min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})} \{R\} = \min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})} \sum_{i=1}^{n} \{R_{R_i} + R_{T_i}\} \tag{4.1}$$

$$\min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})} \{D\} = \min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})} \sum_{i=1}^{N} w_i \cdot TD_i \cdot \{D_{R_i} + D_{T_i}\} \tag{4.2}$$

jointly subject to

$$D_i = D_{R_i} + D_{T_i} \leq D_i^{max} \text{ for all } n = 0, \ldots, N$$

where $R_{R_i}$, $R_{T_i}$ and $D_{R_i}$, $D_{T_i}$ pairs are the corresponding bitrate and distortion pairs for the reference and the target frame sequences for the $i$'th shot respectively. The time duration of shot $i$ is denoted by $TD_i$. This problem formulation aims at finding the best encoding rate sequence (hence the encoding parameters as in Chapter 2 for single view) among shots $1 \leq i \leq N$ for a stereo video with $N$ pre-defined temporal segments. The distortion of the stereo shot number $i$ is constrained by $D_i = D_{R_i} + D_{T_i} \leq D_i^{max}$.

*Formulation 2:*

When the left and right images are treated unequally in terms of distortion, as described by the suppression theory [84, 85, 18], the above formulation can be modified as follows:

$$\min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})}\{R\} = \min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})}\sum_{i=1}^{n}\{R_{R_i}+R_{T_i}\} \tag{4.3}$$

$$\min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})}\{D\} = \min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})}\sum_{i=1}^{N} w_i \cdot TD_i \cdot \{D_{R_i}+\alpha \cdot D_{T_i}\} \tag{4.4}$$

jointly subject to

$$D_i = D_{R_i} + \alpha \cdot D_{T_i} \leq D_i^{max} \text{ for all } n = 0, \ldots, N$$

where $\alpha$ is a weighting coefficient between right and left image distortion values to take different amount of contributions to distortion from each image into account.

*Formulation 3:*

We can also make the first minimization over the pre-roll delay with buffer limitations as in Chapter 2 instead of the average bitrate, as shown below:

$$\min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})}\{T_{pre}\} = \min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})}\max_{1\leq n\leq N}\left\{\sum_{i=1}^{n} TD_i\left(\frac{R_{R_i}+R_{T_i}}{R_{ch}}-1\right)+\frac{B_{n+1}}{R_{ch}}\right\} \tag{4.5}$$

$$\min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})}\{D\} = \min_{(R_{R_1},R_{T_1},\ldots,R_{R_N},R_{T_N})}\sum_{i=1}^{N} w_i \cdot TD_i \cdot \{D_{R_i}+\alpha \cdot D_{T_i}\} \tag{4.6}$$

**jointly** subject to

$$D_i = D_{R_i} + \alpha \cdot D_{T_i} \leq D_i^{max} \text{ for all } n = 0, \ldots, N$$

$$B_{n+1} \leq R_{ch} \cdot T_{pre} + R_{ch} \cdot \sum_{i=1}^{n} TD_i - \sum_{i=1}^{n}(R_{R_i}+R_{T_i}) \cdot TD_i \leq B^{\max} \text{ for all } n = 0, \ldots, N$$

where $T_{pre}$ denotes the initial pre-roll delay needed for uninterrupted stereo video playback.

### *4.6 Conclusions*

In this chapter, we have described our implementation of an end-to-end stereoscopic video streaming system using content-adaptive multi-view coding and modifications to available

open source monocular streaming platforms. Our proposed content adaptive approach for temporal and spatial downsampling of one of the views yields better compression with higher perceptual quality. The performance of the proposed approach is tested on several stereo sequences using subjective quality tests. The system is initially tried on local area network with zero packet losses. Finally, we have proposed several multiple-objective formulations that can be exploited for optimize the streaming experience for stereo video.

# Chapter 5

# CONCLUSIONS AND DISCUSSION

In this thesis, we proposed novel multiple-objective optimization (MOO) frameworks for monocular and binocular video streaming to determine the best bitrate allocation in a video sequence. For the monocular video representation, this bitrate allocation is considered to be only in the temporal direction. On the other hand, the bitrate allocation between left and right images is also taken into account for the binocular video case.

Firstly, the DDO framework where an optimal spatial and temporal resolution for each semantically defined GoP is selected to achieve the least overall distortion and pre-roll (initial) delay according to a user specific relevance/distortion policy was presented. The DDO method interfaces with a standard encoder by specifying the target bit rates and the spatial and/or temporal resolutions for each GoP, allowing a study of trade-offs between pre-roll delay and perceptual distortion. The proposed scheme outperforms the regular bit allocation schemes in the most relevant shots (4.5 dB gain) and provides reasonable quality for the others. The buffer requirements are found to be easily affordable by today's hardware technology. If the coding standard used supports spatio-temporal resolution changes, the resulting compressed bitstreams will be standards compliant. However, we may need a specialized display module to display all pictures at a standard spatial resolution. This is an off-line video encoding framework that can be used for video-on-demand services over low capacity networks.

Secondly, a cross-layer optimized video adaptation and user scheduling scheme for wireless video streaming was introduced for packet-networks. The MOO objectives of this scheme were to select the user-video bitrate pair at each time slot such that the maximum video throughput, maximum user QoS, and video QoS fairness are achieved. This is possible by selecting the pair with the least remaining playback time, highest video quality and the highest video throughput within the MOO framework. The experiments carried out in the IS-856 standard and ITU Pedestrian A and Vehicular B environments with no video

adaptation show significant gains of the proposed system over the state-of-the-art wireless schedulers in terms of application-layer QoS and QoS fairness with higher overall system throughput. The gain achieved is further improved by video rate adaptation, especially in the Pedestrian A environment. The proposed framework runs in real-time and requires a modest increase in the size of the feedback that is regularly sent by each user. However, this increase is negligibly small for the video data rates considered in this scheme.

Finally, an end-to-end stereoscopic video streaming system was implemented using open source components with minor modifications. We were able to show that the encoding bitrate of a stereo video can be much lower than double the encoding bitrate of an equivalent quality monoscopic video encoded with the same encoding parameters and that this bitrate can be as low as 1.2 times that of the monoscopic video using the Human Visual System properties. This is mainly due to the *redundancy elimination* strategy of the modern video codecs and the strong correlation existing between stereo-pairs.

Appendix A

# OVERVIEW OF MULTIPLE-OBJECTIVE OPTIMIZATION (MOO)

## *A.1 Multiple-Objective Optimization (MOO)*

The MOO concept was introduced by Pareto where the solution of an optimization problem with the objective/cost function set $F = \{f_1, f_2, \ldots, f_P\}$, $s^*$, is called globally Pareto-optimal (also non-dominated/non-inferior) if any one of the objective function values cannot be improved without degrading other objective values. Let us assume that the optimization problem in hand consists of $P$ distinct and possibly conflicting objective functions. Without any loss of generality, let us assume that the problem in hand requires all the objective functions to be minimized. Then, a Pareto-optimal solution $s^*$ exists if there exists no other feasible solution $s$ that satisfies

$$f_p(s) \leq f_p(s^*), \forall p \in \{1, \ldots, P\} \tag{A.1}$$

with at least one strict inequality. This means, there cannot exist a feasible solution that is at least as good as a Pareto-optimal solution in all objective functions and strictly better in one or more objective functions, i.e., a Pareto-optimal solution cannot be dominated by any other feasible solution. In our delay-distortion optimization formulation (see Section 2.2.3), $P = 2$ and the objectives are given by (2.4)-(2.5). In our cross-layer optimization formulation, (see Section 3.2), $P = 3$ and the objectives are given by (3.5)-(3.7).

It is possible to have multiple Pareto-optimal solutions in multiple-objective optimization problems ($P \geq 2$). However, unlike the single objective problems, the multiple Pareto-optimal solutions do not necessarily result in a unique functional value. In many cases, as different objective functions represent different system aspects on a specific scale, variance and units of measurement, it is difficult to discriminate between these Pareto-optimal points and determine which one is better than the other. However, using the relative importance weights for all of the objective functions, $w_p$'s, a so called *best compromise solution* can be determined. For example, in the proposed cross-layer framework, the aim is to schedule

the user and the associated video source data rate such that the user provides the maximum instantaneous video throughput, and has the minimum remaining time before possible buffer underflow. Note that, the scales, the measurement units and the variances of video throughput, quality and remaining playback time all differ from each other.

There exist several solution techniques to this problem in the literature. Minimizing the weighted sum of functions [97] is one of the most popular solution methods. However, this method needs accurate selection of the scalar weights which is a very difficult task in most cases [98]. The equality constraint method that minimizes objective functions one by one by simultaneously specifying equality constraints on the other objective functions was presented in [99]. In the goal programming technique [100], only one objective is minimized while constraining the other objectives to be less than their target values. This technique cannot be used to generate the Pareto-optimal set of solutions effectively since the suitable selection of the objective target values can be quite difficult. The normal-boundary intersection (NBI) method [101] tries to enumerate an even distribution of Pareto-optimal points on the Pareto-optimal curve even for the case of objectives with very different scales. NBI may generate points that do not actually belong to the Pareto-set if the feasible region is non-convex. In multi-level programming, objective functions are first ordered due to their importance and then single objective optimization methods are applied in this order recursively, reducing the sample set at each step. Here, the optimal solutions for the most important objective function are found, forming the new sample set for the next important objective function and so on. Although this is a very useful method when there is a certain hierarchy among objectives, the continuous tradeoff between objective functions is disregarded, lowering the overall performance.

In order to determine the best compromise solution among the objective functions, $f_p$'s, we first rescale their values to an interval $[0,w_p]$, where $w_p$ is the importance weight of the $p^th$ objective function using the following equation:

$$f_{p,scaled}(n) = w_p \frac{f_p(n) - f_{min}(n)}{f_{max}(n) - f_{min}(n)} \tag{A.2}$$

Hence, the video throughput, user remaining playback time, and video rate values are all normalized to form a three-dimensional solution space. Note that, ideally the optimizer would select higher video bit-rates when the user remaining playback times are high and

lower the bit-rates when they are low. For this purpose, the weight of the $3^{rd}$ objective function for maximizing the video rate, $w_3$, can be dynamically changed at each time slot according to the average remaining playback time for all users in the system, $\overline{\theta}(n)$, i.e. $w_3 = \overline{\theta}(n)/\theta_{max}$ where $\theta_{max}$ is the maximum possible remaining playback time which is equal to the ratio of the buffer size to the slowest available video coding rate.

In MOO problems, an *infeasible* point that optimizes all of the objective functions *individually* is called the *utopia point*. Hence, the utopia point, *U(n)*, for the three-dimensional scaled video throughput, remaining playback time, and video rate solution space is as follows:

$$U(n) = (\max(t_{i,l,scaled}(n)) \ , \ \min(\theta_{i,scaled}(n)) \ , \ \max(\mu_{i,l,scaled}(n))) \tag{A.3}$$

Figure A.1 shows an example of a scaled feasible solution set for $P$=2 objective functions, where both objectives are being minimized and the feasible solutions are depicted by dots. The best compromise solution is found as the *feasible* point that is closest to the utopia point in the Euclidian-distance sense.

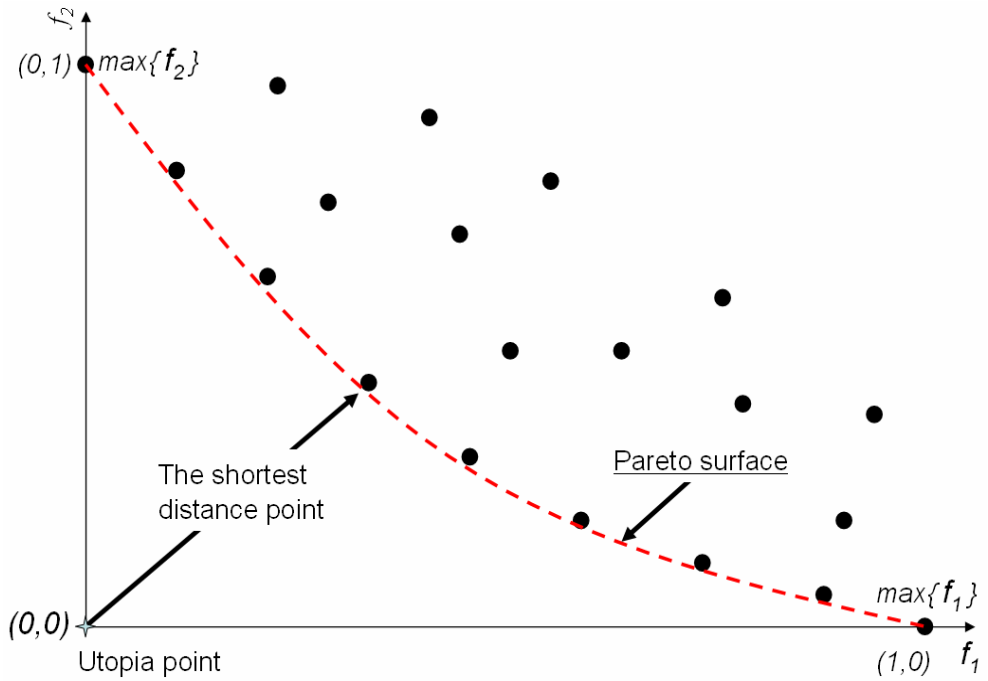



Figure A.1: The solution whose objective values are closest to the utopia point is chosen.

In the proposed framework, an exhaustive search proves to be computationally feasible to determine the utopia point, since for a system with $M$ active users we only need $3 \times (M-1)$ comparisons resulting in a complexity of order $M$.

It is also possible to generate a solution that is better than the actual best compromise solution for one objective function, but worse for the others. This actually corresponds to fine-tuning the optimization decisions in favor of a selected optimization criterion along the Pareto-surface. For example, we can come up with a solution that has lower video quality with better continuous playback performance and vice versa. Knowing the client preferences, the server side may prefer to skip the original optimal solution and offer different solutions by utilizing this property as shown in Figure A.2. This decision depends on the answers to the following two questions:

1. How much of performance degradation can be tolerated by a client in each objective function for the sake of performance improvement in another objective function?

2. What is the sensitivity of this tradeoff?

A thorough treatment of multiple-objective optimization (MOO) techniques can be found in [102, 103].

### *A.2 Example: A Simple MOO Problem and Its Solution*

This section presents a simple example to demonstrate the optimal solution generated by a MOO formulation. Suppose that we would like to solve the following MOO problem:

$$\min_{x,y} f(x,y) = \min_{x,y}\{x \cdot y\} \tag{A.4}$$

$$\min_{x,y} g(x,y) = \min_{x,y}\left\{\frac{200}{x} + \frac{200}{y}\right\} \tag{A.5}$$

jointly subject to

$$x \in [1,20] \quad \textit{and} \quad y \in [1,20]$$

The sketch of the functions f(x,y) and g(x,y) for the region of interest is shown in Figure A.3.

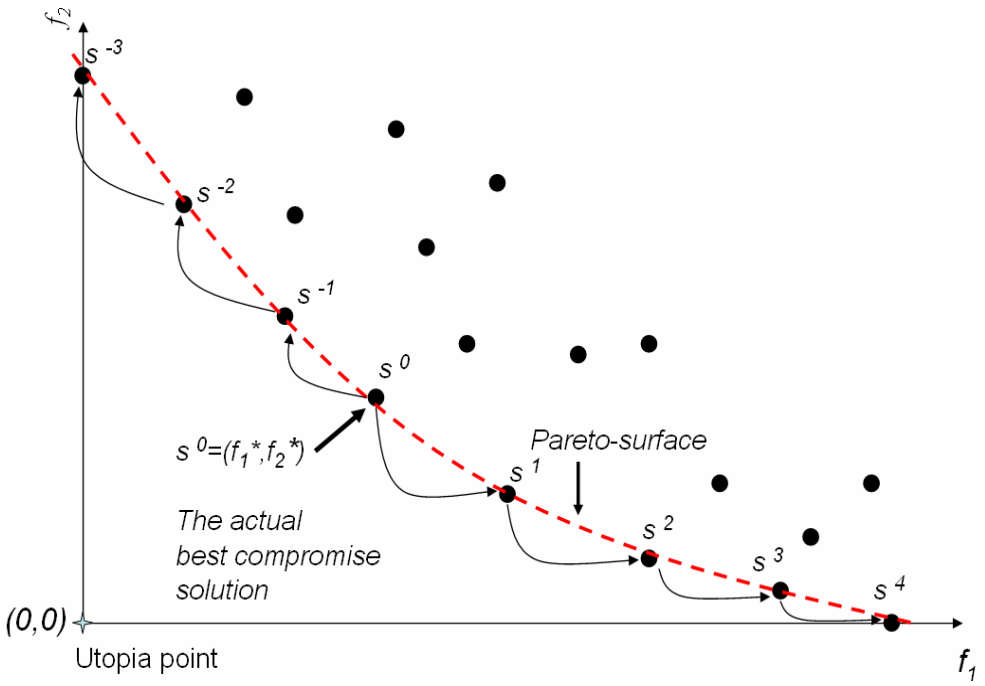



Figure A.2: Fine-tuning of the optimization decisions along the Pareto-optimal surface.

The point $(x, y) = (1, 1)$ minimizes $f$ with a minimum value of $f_{min} = 1$ while $g$ attains its maximum value, $g_{max} = 400$ at this point. The other endpoint $(x, y) = (20, 20)$ minimizes $g$ with a minimum value of $g_{min} = 20$, while $f$ attains its maximum value $f_{max} = 400$ at this point.

A solution is called Pareto-optimal if any one of the objective values cannot be improved without degrading other objective values. In other words, a Pareto-optimal solution cannot be dominated (outperformed in all the objective functions) by any other feasible solution. In order to draw the Pareto-optimal curve for our example, we take $Q$ equally spaced samples in the interval $[f_{min}, f_{max}]$. For every sample, we find the minimum value that the other cost function g can achieve, and plot the Pareto-optimality trade-off curve shown in Figure A.4.

An infeasible point that minimizes both of the objective functions individually, the point $(f_{min} = 1, g_{min} = 20)$ for the example presented here, is called the *utopia point*. The best compromise solution is defined as the point on this curve that is closest to the utopia point $(f = 1, g = 20)$ in the Euclidian-distance sense after proper scaling (subtracting the mean

and dividing by the standard deviation) of all feasible points. In our example, the closest point to the utopia point on this curve can be found as $(f = 38.21, g = 64.71)$. The corresponding x and y values are determined as $x = y = 6.181$.

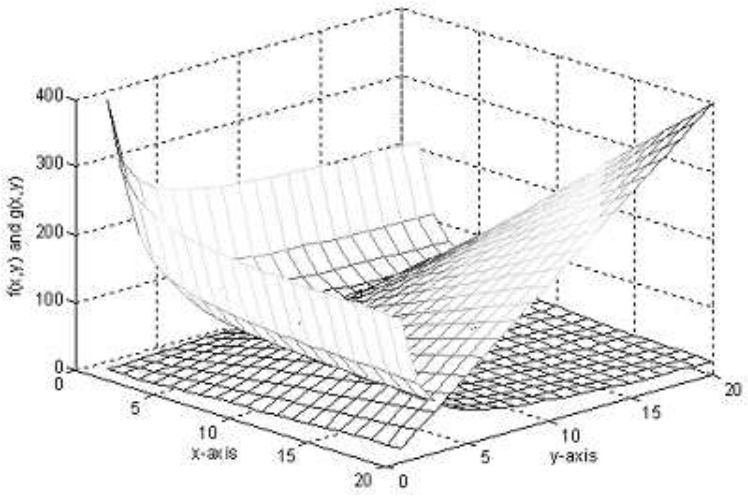



Figure A.3: Sketch of the two functions f and g in the region of interest.

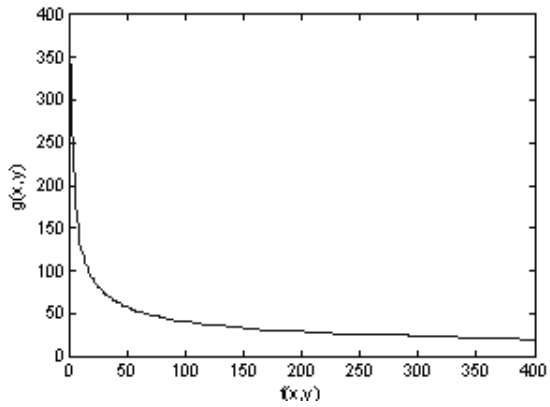



Figure A.4: Minimum values that the cost function g can take for possible values of $f$ in the interval $[f_{min}, f_{max}]$.

# Appendix B

# PERCEPTUAL QUALITY MEASURES

In this appendix, we define our particular selection of distortion measures to use in the proposed Delay-Distortion Optimization (DDO) framework. In particular, we relate distortion to perceptual quality measures. We note that the proposed MOO scheme is not coupled with the specific distortion measure selection, and can be used with other distortion measures as well.

In the DDO framework, we employ a weighted combination of PSNR and a blockiness or flatness measure to quantify distortion. Perceptual video quality measures are determined at the encoder (server) side, which has access to uncompressed video or a very high-quality compressed version. Therefore, we use referenced measures, which needs to employ the original version of the video sequence in hand.


Figure B.1: Organization of blocks.

Our blockiness and flatness measures are modified versions of those proposed in [45], which compares pixel intensity variations across boundaries of blocks and within blocks. For $M \times N$ blocks, a horizontal blockiness measure, $BM_h$, between blocks A and B (depicted in Figure B.1) for both the original and the encoded versions are computed as follows:

$$\boldsymbol{BM_h} = \begin{cases} \frac{BD1_h}{BD3_h} & \text{if } BD3_h \neq 0 \\ 0 & \text{if } BD3_h = 0 \end{cases}$$

where $BD1_h$ and $BD3_h$ refer to one-pixel inter-block difference and cumulative difference over $\pm 3$ columns across the block boundary, respectively, which are defined by:

$$BD1_h = \gamma_1 \cdot \sum_{i=1}^{N} |a_{i1} - b_{iM}| \tag{B.1}$$

$$BD3_h = \gamma_2 \cdot \sum_{i=1}^{N} \left( \sum_{j=M-3}^{M-1} |b_{i(j+1)} - b_{ij}| + \sum_{j=1}^{3} |a_{i(j+1)} - a_{ij}| \right) \tag{B.2}$$

Here $a_{ij}$ and $b_{ij}$ denote values offF pixels in blocks A and B, respectively, $\gamma_1$ and $\gamma_2$ are normalization factors. The effective horizontal blockiness of a certain block $BM_h^{eff}$ caused by lossy compression is:

$$BM_h^{eff} = max\left\{(BM_h^{enc} - BM_h^{org}), 0\right\} \tag{B.3}$$

where $BM_h^{enc}$ and $BM_h^{org}$ are the horizontal blockiness measures of the same block in the encoded and the original clips, respectively. The effective vertical blockiness measure, $BM_v^{eff}$, between blocks A and C is defined similarly. Then, the effective blockiness measure for block A is computed as the average of the horizontal and vertical effective blockiness measures between blocks A and B, and A and C; and an overall effective blockiness measure for a frame is defined as the average of the effective blockiness measures of all blocks within that frame. Similarly, a horizontal flatness measure, $F_h$, between blocks A and B is defined as:

$$F_h = \gamma_3 \cdot \sum_{i=1}^{N} \left( \sum_{j=M-3}^{M-1} z(b_{i(j+1)}, b_{ij}) + \sum_{j=1}^{3} z(a_{i(j+1)}, a_{ij}) \right) + \gamma_3 \cdot \sum_{i=1}^{N} z(a_{i1}, b_{iM}) \tag{B.4}$$

where

$$\boldsymbol{z(\alpha, \beta)} = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{if } \alpha \neq \beta \end{cases}$$

and $\gamma_3$ is a normalization factor. A vertical flatness measure, $F_v$, is computed likewise. The effective flatness measure of a block can be computed by the same procedure used in effective blockiness. Finally, the perceptual measure of a block is given by the maximum of the effective blockiness and flatness measures that are appropriately scaled, and the overall perceptual measure of a frame is the average of these measures for all blocks that fall within that frame.

Hence, the overall distortion measure for the $i^{th}$ shot $D_i$ is a weighted combination of PSNR and perceptual measures $Blk_i$ (blockiness) for that shot and is given by

$$D_i = \frac{Blk_i - \mu_{Blk}}{\sigma_{Blk}} - \frac{PSNR_i - \mu_{PSNR}}{\sigma_{PSNR}} \tag{B.5}$$

where $\mu_{Blk}$, $\mu_{PSNR}$ and $\sigma_{Blk}$, $\sigma_{PSNR}$ denote the mean and the standard deviation of PSNR and blockiness measures, respectively, computed over all shots.

We note that the correlation between adjacent frames gets smaller for lower frame rates, which causes the encoded video to have a lower PSNR and higher blockiness measures when compared to a version encoded at a higher frame rate with the same quantization parameter. Therefore, we do not employ an extra motion jitter measure to take frame rate into account while computing our overall quality measure.

Appendix C

# VIDEO CONTENT ANALYSIS

The tremendous increase in the number of available multimedia applications in the last few years has emerged video analysis techniques for efficient summarization, search and browsing of video content. Especially, applications like finding similar shots/pictures in a given video content, video summarization and searching for the videos of a specific event or person has been attracting great amount of attention recently. We define a "camera shot" to be a *sequence of images* with similar content in a video segment.

## *C.1 Analysis of Temporally/Spatially Structured Videos such as Sports and News Reports*

In general, it is a challenging task to teach a computer to semantically analyze a video with random structure. In order to make this task easier, video analysis usually has been tried on video contents that have a certain *predefined* temporal and spatial structure, such as sports and news report videos. One can think of a soccer game as consisting of three major events. The first shot type is the so called *long shots*, in which many players and the ball interacting on a big portion of the soccer field is shown. The second one is the *close-up shots*, in which players, referee or team coach is zoomed in and finally, the third one is the *audience shots*, where the audience is shown. Detecting only these three events according to some scene attributes is much more convenient than trying to detect events that that have been neither predefined, nor temporally/spatially structured, in terms of system performance and computational efficiency.

Automatically indexing an input video can be done in two main steps. The first step is determining the shot boundaries. The second step is identifying shots with semantically distinct meanings and classifying the shots with similar content into the same category. To do this, image properties has to be extracted from each shot to come up with semantic descriptions. However, it is not always easy to carry out such a task for all types of video

contents. For instance, it is very difficult to semantically analyze some contents such as movies and documentary films, even manually. On the other hand, the shot types for some contents like sports and news report videos are relatively easier to be categorized automatically, because certain shot types repeatedly appear in the duration of the overall video and one can actually predetermine the types of shots that are likely to occur. Efficient automatic or semi-automatic content analysis and temporal segmentation methods for sports and news report videos have been intensively studied in the literature [23]-[26].

### *C.2 Movie Content Analysis*

Content-based movie analysis and indexing approach has been attracting a lot of interest recently and there are great amounts of investments being made in multimedia-information. A single shot can no longer be a valid retrieval unit for movies. Therefore, in order for us to have good understanding of content-based movie description and for better content browsing, an event based segmentation algorithm as in [104] is needed instead of shot based segmentation. The content-based movie analysis and indexing scheme of [105] tries to extract semantically meaningful events in movies and to identify target speakers in movie dialogs. [105] proposes to extract movie events and speaker identity at the semantic level depending on fusion of audio information with visual information. In this work, three types of events are assumed to exist in a movie. These event types, which are considered to be the most informative parts of the movie, are 2-speaker dialogs, multiple-speaker conversations, and hybrid events, that is scenes with less dialog and more action. The information bearing audiovisual event attributes extracted are then used to make browsing, abstraction and indexing of the movie possible. Visually and temporally close shots are grouped together into a shot sink. Afterwards, these sinks are classified as periodic, partly-periodic or non-periodic using unsupervised K-means algorithm. At the end, one of the three event types is assigned to each shot group.

### *C.3 Semantic Relevance Measure*

In some video domains, not all the shots are equally interesting to a user, hence bit rate allocation may be done according to semantic relevance of the content; in other video domains, video analysis can be employed to determine bit rate allocation according to

coding difficulty of the content. For example, in a soccer game, shots in the vicinity of goals may be more interesting than others. In a tennis game, breaks given between sets are not as relevant as the in-game strife. Television news reports can be segmented as anchorperson shots, news footage and commercial breaks. For movies, temporal segmentation and content analysis may facilitate bit rate assignment as a function of coding difficulty, existence of special effects, etc.

Given the shot boundaries and shot types, the relevance factor $w_{i,v}$ of each shot $i$ can be determined according to a pre-specified or user defined relevance-distortion policy, where more relevant shots will be encoded with less distortion. That is, a user's level of interest in certain types of shots (goals, breaks, commercials etc.) can either be set to default values by our system, or can be signaled by the user prior to the video transmission. Given that a finite evaluation scale will be sufficient for our purposes, the semantic relevance factors are specified between 0 and 1.

Generally, the audio information can also be used in assessing the relevancy of a video segment [46]-[47]. In sports video, we can assume that the audio signal energy will increase whenever an important event occurs since the voice of the commentator and the noise that the audience makes are going to increase.

The overall relevance factor $w_i$ of the video segment $i$ can be adjusted by

$$w_i = w_{i,v} \cdot (E_{i,avg}/E_{global}) \tag{C.1}$$

where $w_{i,v}$ is the relevance factor determined by the video content only, $E_{i,avg}$ is the average audio energy of the $i^{th}$ segment, and $E_{global}$ is the average energy for the entire video.

### *C.4 Monocular Video Analysis*

Two of the most important applications of video analysis are video browsing and retrieval. It is a very challenging task to index and organize video databases since the overall database size and complexity is generally huge. There has been important studies on automatic/semi-automatic temporal segmentation of video into shots [106]. Generally, the summary of each shot is given by a key frame as shown in Figure C.1 and some global features associated with it such as the color histogram [107, 108, 109, 110]. Therefore, the more attributes we can extract out of a video and the shots it includes, the better such summarization

algorithms will work. There are also interactive summarization methods such as Scene Transition Graphs (STG), in which both the temporal flow of the video and the image content are illustrated as explained in [111]. However, the way human beings observe and interpret a moving scene or a video is more complex than just temporal activities and events within that scene. The objects included and their interactions with each other are also very important for our understanding of the scene/shot. Therefore, an object-based video analysis and summary method would be of much better use for us. On the other hand, it is very difficult to develop a common object-based description method for all video domains since each video domain may include different types of objects and various types of interactions between them. Therefore the video analysis techniques used in this thesis will be applied to some specific video domains such as soccer and news report videos.

Figure C.1: An example set of key frames representing different types of shots in a soccer game.

As explained in [24], firstly, crucial objects in the shot have to be identified in object-based video indexing. For example, these crucial objects may be the ball, the players and the soccer field in a soccer game. Afterwards, other lower level attributes and information such as color, texture and semantics of each object are added. It is important to account for the time-varying attributes of the objects such as their shapes and movement. Therefore, only a few key frames from an entire shot are not descriptive enough, since those key frames do not necessarily reflect average values of the time varying attributes.

### *C.5 3-D Video Analysis*

Multi-view video technology can provide more efficient handling and control of video objects since it enables extra image attributes related to the depth information as explained in [90]. In [90], a general scene complexity free framework for non-linear representation of three dimensional videos is introduced. In this work, color segments that belong to close depth levels are merged to obtain precise contours of the scene objects. This approach is valid as long as the parts of each video object have similar depth values. The first step in the algorithm is to compute the disparity field, which leads to the depth information. However, the depth information alone is not enough for accurate video object segmentation. For this reason, color and depth information are fused together to determine object boundaries (contours) more accurately, so that a better content-based segmentation is achieved.

The size of feature vectors extracted this way would be different for each frame obviously. In order to prevent this, color levels are also quantized along with depth levels in to pre-determined color-depth classes to form a multidimensional histogram. After applying a shot boundary detection method [89] to the whole stereoscopic video sequence, the most representative frame for each shot is chosen as the one that has the minimum correlation measure (as defined in [90]) among all frames within that shot.

In [112], a combination of active contours and retrainable neural networks is used. By examining the depth information extracted out of disparity knowledge, an active contour is initialized on the boundary of each segment (on the same depth level) to detect the surrounded object. The detected video objects form the retraining set and the neural network weights are adapted accordingly. Afterwards, the trained network is applied to the rest of the frames in the same shot to fully detect the positions of the initially found video objects.

# Appendix D

## LUCAS-KANADE OPTICAL FLOW ESTIMATION

The essential steps of the algorithm are the following:

- Let $u$ and $v$ be two corresponding points in the left image $I$ and the right image $J$, respectively. First we need to build the pyramid representations of both images.

- Let the initial guess of the displacement vector at the top level (level m) be given by:

$$g^{L_m} = [g_x{}^{L_m} \; g_y{}^{L_m}] = [0 \;\; 0]^T$$

- In a for loop, starting from level m to level 0:

  - Find the location of the tracked point on the $L_{th}$ level, $u^L = u/2^L$.
  - Find the gradient in the x and y directions as follows:

$$I_x(x,y) = \frac{I^L(x+1,y) - I^L(x-1,y)}{2}$$

$$I_y(x,y) = \frac{I^L(x,y+1) - I^L(x,y-1)}{2}$$

  - Then the spatial gradient matrix $G$ is computed as given by:

$$G = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2(x,y) & I_x(x,y)\, I_y(x,y) \\ I_x(x,y)\, I_y(x,y) & I_y^2(x,y) \end{bmatrix}$$

  - Here the $K$ iterations of Lucas-Kanade method can be initialized. Initial displacement guess for the $k^{th}$ iteration is a zero vector:

$$\overline{V}^0 = [0 \;\; 0]^T$$

  - Compensated image difference is computed as:

$$\delta I_k(x,y) = I^L(x,y) - J^L(x + g_x^L + \nu_x^{k-1}, y + g_y^L + \nu_y^{k-1})$$

– When the image mismatch vector is defined as below, the optical flow $\overline{\eta}^k$ at the $k^{th}$ iteration step and the initial guess for the next iteration can be found easily.

$$\overline{b}_k = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I_k(x,y)\, I_x(x,y) \\ \delta I_k(x,y)\, I_y(x,y) \end{bmatrix}$$

$$\overline{\eta}^k = G^{-1}\, \overline{b}_k$$

$$\overline{\nu}^k = \overline{\nu}^{k-1} + \overline{\eta}^k$$

– Resulting optical flow for the level $L$ is given by $d^L = \overline{\nu}^K$ and the initial guess for the next lower level is computed as:

$$\mathbf{g}^{L-1} = [g_x^{L-1} \;\; g_y^{L-1}]^T = 2\,(\mathbf{g^L} + \mathbf{d}^L)$$

- The overall flow vector $d$ is found as:

$$d = g^0 + d^0$$

For the Lucas-Kanade method to work fine, the flow vectors have to be small since the algorithm uses the first order Taylor expansion.

The resulting optical flow is perfect when we use only 2 levels and is given in Figure D.2 and the optical flow vectors for the Tsukuba image pair is also shown on Figure D.3.

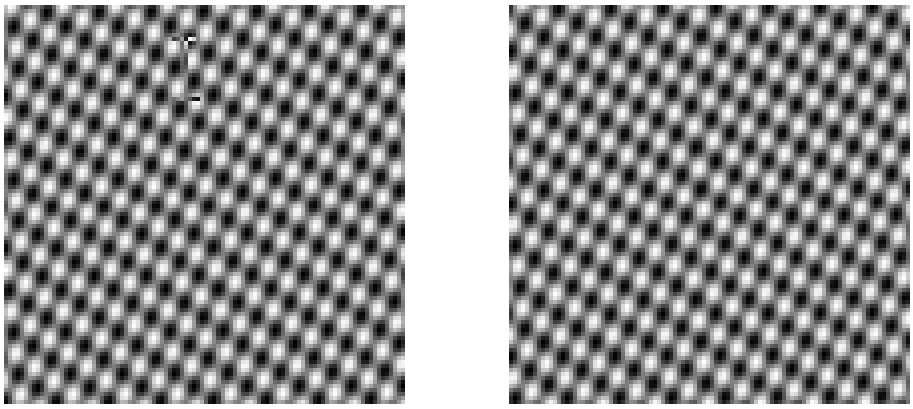


Figure D.1: An example stereo image pair.

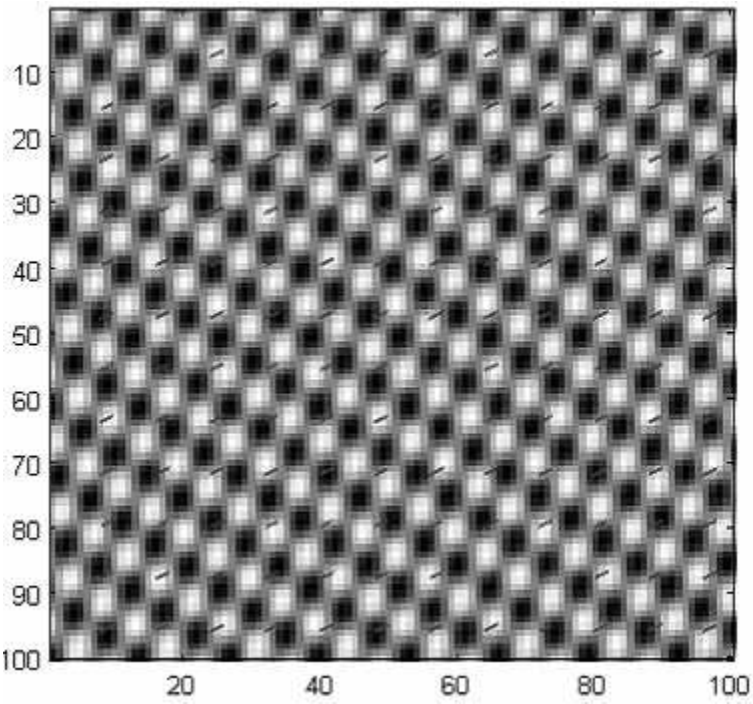



Figure D.2: Resulting optical flow of the stereopair.

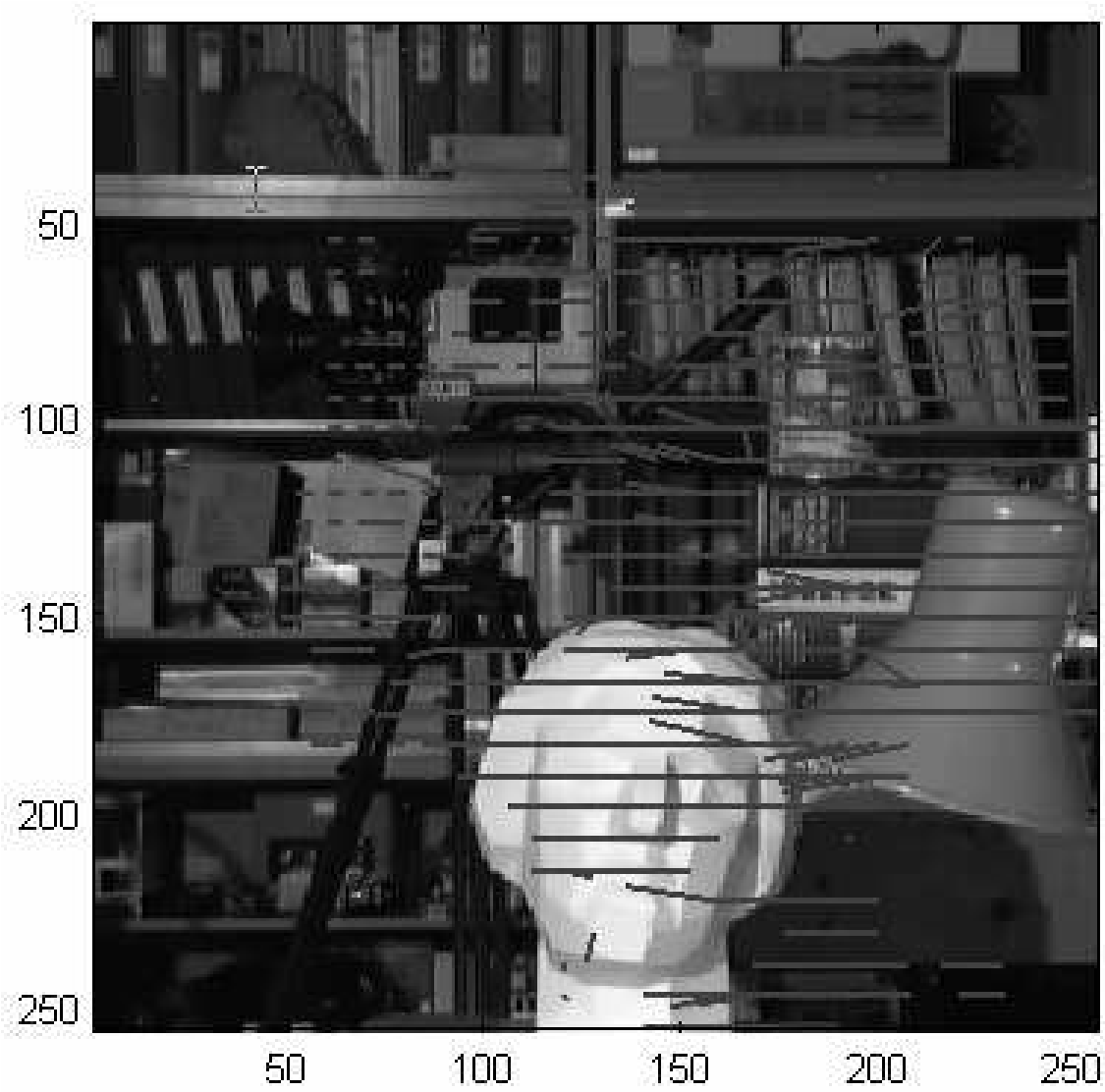



Figure D.3: Optical flow vectors of the Tsukuba stereopair.

# BIBLIOGRAPHY

[1] http://www.youtube.com.

[2] http://video.google.com.

[3] L. Zhao and C.-C. Jay Kuo. Buffer-constrained R-D optimized rate control for video coding. *IEEE International Conference on Acoustic, Speech and Signal Processing, Hong Kong*, 3:89–92, April 2003.

[4] S. Ma, W. Gao, F. Wu, and Y. Lu. Rate control for JVT video coding scheme with HRD considerations. *ICIP*, 3:793–796, 2003.

[5] T. Wiegand, G. Sullivan, and A. Luthra. Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T rec. H.264 — ISO/IEC 14496-10 AVC). May 2003.

[6] G.J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 15:74–90, November 1998.

[7] S. Kadono and N. Yokoya. RD optimization technique of quantization for MPEG-4 intra-frame coding. *Picture Coding Symposium*, 2001.

[8] Ping Li, W.S. Lin, S. Rahardja, X. Lin, X.K. Yang, and Z.G. Li. Geometrically determining leaky bucket parameters for video streaming over constant bit-rate channel. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2004.

[9] C.-Y. Hsu and A. Ortega. Joint encoder and VBR channel optimization with buffer and leaky bucket constraints. *Symposium on Multimedia Communications and Video Coding, Brooklyn, NY*, October 1995.

[10] K.R. Rao and P. Yip. Discrete cosine transform-algorithms, advantages, applications. *Academic Press, Inc. London*, 1990.

[11] Moving Pictures Expert Group. MPEG-2 test model 5, doc. ISO-IEC / JTC1 / SC29 / WG11 / MPEG93. March 1998.

[12] I. Dinstein, M.G. Kim, J. Tzelgov, and A. Henik. Compression of stereo image and the evaluation of its effects on 3-D perception. *SPIE*, 1153, 1989.

[13] I. Dinstein, M. Guy, J. Rabany, J. Tzelgov, and A. Henik. On stereo image coding. $9^{th}$ *International Conference on Pattern Recognition*, 1:357–359, November 1988.

[14] P. D. Gunatilake, M. W. Siegel, and A. G. Jordan. Compression of stereo video streams. *Stereoscopic Displays and Virtual Reality Systems*, 2177:258–268, February 1994.

[15] W. Woo. Rate-distortion based dependent coding for stereo images and video: Disparity estimation and dependent bit allocation. *Ph.D. thesis, University Of Southern California*, December 1998.

[16] A. Puri, R. V. Kollarits, and B. G. Haskell. Basics of stereoscopic video, new compression results with MPEG-2 and a proposal for MPEG-4. *Image Communication: Signal Processing*, 10:201–234, 1997.

[17] M. G. Perkins. Data compression of stereopairs. *IEEE Transactions on Communication*, 40:684–696, 1992.

[18] W. Woo and A. Ortega. Optimal blockwise dependent quantization for stereo image coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:861–867, September 1999.

[19] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conerence on Artificial Intelligence*, pages 674–679, 1981.

[20] J-Y. Bouguet. Pyramidal implementation of the Lucas-Kanade feature tracker description of the algorithm. *Intel Corp. Microprocessor Research Lab.*, 2003.

[21] A. Ortega. Optimal trellis based buffered compression and fast approximations. *IEEE Transactions on Image Processing*, 3, January 1994.

[22] A. Vetro, Y. Wang, and Sun H. Rate-distortion optimized video coding considering frameskip. *Proceedings of the IEEE International Conference on Image Processing*, 3:534–537, October 2001.

[23] D. Yow, B.-L. Yeo, M. Yeung, and B. Liu. Analysis and presentation of soccer highlights from digital video. *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 1995.

[24] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12:796–807, June 2003.

[25] M.G. Brown, J. T. Foote, G. J. F. Jones, K. S. Jones, and S. J. Young. Automatic content-based retrieval of broadcast news. *Proceedings of the 3rd Int'l Conf. on Multimedia (ACM Multimedia 95), ACM Press, New York*, pages 35–43, 1995.

[26] V. Tovinkere and R. J.Qian. Detecting semantic events in soccer games: Toward a complete solution. *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, August 2001.

[27] M. Bertini, A. Del Bimbo, R. Cucchiara, and A. Prati. Semantic video adaptation based on automatic annotation of sport videos. *Proceedings of 6th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR), NY, USA*, pages 291–298, October 2004.

[28] P. van Beek, J. R. Smith, T. Ebrahimi, T. Suzuki, and J. Askelof. Metadata-driven multimedia access. *IEEE Signal Processing Magazine*, 20:40–52, March 2003.

[29] S.F. Chang, D. Zhong, and R. Kumar. Real-time content-based adaptive streaming of sports video. *IEEE Workshop on Content-Based Access to Video/Image Library, Hawaii*, pages 139–146, December 2001.

[30] T. Suzuki and P. M. Kuhn. Mpeg-7 metadata for segment based video transcoding. *Picture Coding Symposium*, 2001.

[31] C. Kuhmunch, G. Kuhne, C. Schremmer, and T. Haenselmann. A video-scaling algorithm based on human perception for spatio-temporal stimuli. *Proceedings of SPIE, Multimedia Computing and Networking (MMCN)*, pages 13–24, January 2001.

[32] Special issue. *IEEE Signal Processing Magazine*, 20, March 2003.

[33] R. Mohan, J. R. Smith, and S. Chung. Adapting multimedia Internet content for universal access. *IEEE Transactions on Multimedia*, 1:104–114, March 1999.

[34] M. Bertini, A. Del Bimbo, R. Cucchiara, and A. Prati. Semantic video adaptation based on automatic annotation of sport videos. *Proceedings of 6th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR), NY, USA*, pages 291–298, October 2004.

[35] P. Pahalawatta, Z. Li, F. Zhai, and A. K. Katsaggelos. Rate-distortion optimized video summary generation and transmission over packet lossy networks. *SPIE Image/Video Comm. and Proceedings, San Jose, CA*, January 2005.

[36] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G.J. Sullivan. Rate-constrained coder control and comparison of video coding standards. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:688–703, July 2003.

[37] E. Steinbach, N. Farber, and B. Girod. Adaptive playout for low-latency video streaming. *Proceedings of IEEE International Conference on Image Processing (ICIP), Thessaloniki, Greece*, 1:962–965, October 2001.

[38] M. Kalman, E. Steinbach, and B. Girod. R-D optimized media streaming enhanced with adaptive media playout. *Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Lausanne, Switzerland*, 1:869–872, August 2002.

[39] J.-W. Lee, A. Vetro, Y.Wang, and Y.-S. Ho. Bit allocation for MPEG-4 video coding with spatio-temporal tradeoffs. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:488–502, June 2003.

[40] S. Liu and C.-C. J. Kuo. Joint temporal-spatial bit allocation for video coding with dependency. *IEEE Transactions on Circuits and Systems for Video Technology*, 15:15–26, January 2005.

[41] S. Winkler, A. Sharmaa, and D. McNally. Video quality and blockiness metrics for multimedia streaming applications. *Proceedings of the International Symposium on Wireless Personal Multimedia Communications, , Aalborg, Denmark*, pages 547–552, September 2001.

[42] C. J. van den B. Lambrecht and O. Verscheure. Perceptual quality measure using a spatio-temporal model of the human visual system. *Proceedings of the SPIE, San Jose*, 2668:450–461, 1996.

[43] S. A. Karunasekera and N. G. Kingsbury. A distortion measure for blocking artifacts in images based on human visual sensitivity. *IEEE Transactions on Image Processing*, 4:713–724, 1995.

[44] M. Miyahara, K. Kotani, and V. R. Algazi. Objective picture quality scale (PQS) for image coding. *IEEE Transactions on Communications*, 46:1215–1226, September 1998.

[45] F. Pan, X. Lin, S. Rahadja, W. Lin, E. Ong, S. Yao, Z. Lu, and X. Yang. A locally adaptive algorithm for measuring blocking artifacts in images and videos. *Image Communication: Signal Processing*, 19:499–506, July 2004.

[46] Y. Wang, Z. Liu, and J. Huang. Multimedia content analysis using both audio and visual clues. *IEEE Signal Processing Magazine*, 17:12–36, November 2000.

[47] T. Zhang and J. Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on Speech and Audio Processing*, 9:441–457, May 2001.

[48] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[49] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson. Cross-layer design for wireless networks. *IEEE Communications Magazine*, 41:74–80, October 2003.

[50] R. Knopp and P. A. Humblet. Multiple accessing over frequency selective fading channels. *Proceedings of IEEE PIMRC, Canada*, October 1995.

[51] A. Jalali, R. Padovani, and R. Pankaj. Data throughput of CDMA-HDR: A high efficiency high data-rate personal communications system. *Proceedings IEEE VTC, Japan*, May 2000.

[52] S. Shakkottai and A. Stolyar. Scheduling algorithms for a mixture of real-time and non-real-time data in hdr. *Proceedings ITC-17, Brazil*, 2001.

[53] T. Wiegand, G. Sullivan, and A. Luthra. *ITU-T Rec. H.264—ISO/IEC 14496-10 AVC*, 2003.

[54] H. Schwarz, D. Marpe, T. Schierl, and T. Wiegand. Combined scalability support for the scalable extension of h.264/avc. *Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Netherlands*, July 2005.

[55] C.-M. Huang, Y.-T. Yu, and Y.-W. Lin. An adaptive control scheme for real-time media streaming over wireless networks. *Proceedings of AINA*, pages 373–378, 2003.

[56] C. E. Luna, Y. Eisenberg, R. Berry, T. N. Pappas, and A. K. Katsaggelos. Joint source coding and data rate adaptation for energy efficient wireless video streaming. *IEEE Journal on Selected Areas in Communications*, 21, December 2003.

[57] D. Tian, X. Li, G. Al-Regib, Y. Altunbasak, and J. R. Jackson. Optimal packet scheduling for wireless video streaming with error-prone feedback. *Proceedings of IEEE WCNC*, 2004.

[58] L.-U Choi, W. Kellere, and E. Steinbach. Cross-layer optimization for wireless multi-user video streaming. *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2004.

[59] T. Ozcelebi, M. O. Sunay, A. M. Tekalp, and M. R. Civanlar. Cross-layer design for real-time video streaming over 1xev-do using multiple objective optimization. *Proceedings of IEEE Globecom, St. Louis, USA*, November 2005.

[60] X. Zhou and C.-C. J. Kuo. Enhanced video stream switching schemes for h.264. *IEEE International Workshop on Multimedia Signal Processing, China*, October 2005.

[61] T. Ahmed, A. Mehaoua, R. Boutaba, and Y. Iraqi. Adaptive packet video streaming over IP networks: A cross-layer approach. *IEEE Journal on Selected Areas in Communications*, 23:385–401, February 2005.

[62] S.F. Chang and A. Vetro. Video adaptation: Concepts, technologies and open issues. *Proceedings of the IEEE*, 93:148–158, January 2005.

[63] I. Radulovic, P. Frossard, and O. Verscheure. Adaptive video streaming in lossy networks: Versions or layers? *Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Taiwan*, June 2004.

[64] A. Ortega. Compressed video over networks. *M.-T. Sun and A. R. Reibman, eds, pp. 343-382, Marcel Dekker, New York, NY*, 2000.

[65] C.C.J. Kuo and H. Song. Rate control for low-bit-rate video via variable encoding frame rates. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:512–521, April 2001.

[66] IS-856 cdma2000. High Rate Packet Data Air Interface Specification. *TIA Std.*, November 2000.

[67] Y. Wang and Q.-F. Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE*, 86:974–997, May 1998.

[68] ITU. Guidelines for evaluation of radio transmission technologies for IMT-2000. *Recommendation ITU-R, M.1225*, 1997.

[69] M. Gudmundson. Correlation model for shadow fading in mobile radio systems. *Electronics Letters*, 27:2145–2146, November 1991.

[70] http://www.3dtv research.org/.

[71] S. Oh, Y. Lee, W. Woo, V. Roca, and F. Rousseau. Scalable stereo video coding for heterogeneous environments. *Proceedings of MIPS, LNCS 3311*, pages 72–83, 2004.

[72] B. Balasubramaniyam, E. Edirisinghe, and H. Bez. An extended H.264 codec for stereoscopic video coding. *Proceedings of SPIE*, 2004.

[73] P. Merkle, H. Schwarz, T. Hinz, A. Smolic, and T. Wiegand. Multi-view video coding based on H.264/AVC using hierarchical B-frames. *Proceedings of Picture Coding Symposium (PCS), Beijing, China*, April 2006.

[74] C. Bilen, A. Aksay, and G. B. Akar. A multi-view video codec based on H.264. *Proceedings of IEEE International Conference on Image Processing (ICIP)*, October 2006.

[75] Karsten Mller, P. Merkle, Aljoscha Smolic, and Thomas Wiegand. Multiview coding using AVC. *MPEG Meeting - ISO/IEC JTC1/SC29/WG11, Bangkok, Thailand, MPEG06/M12945*, January 2006.

[76] L. B. Stelmach, W. J. Tam, D. Meegan, and A. Vincent. Stereo image quality: Effects of mixed spatio-temporal resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 10:188–193, 2000.

[77] http://www.apple.com/.

[78] http://gpac.sourceforge.net/.

[79] http://www.videolan.org/vlc/.

[80] ITU-T ISO/IEC 14496-10. Recommendation H.264: Advanced video coding for generic audiovisual services. May 2003.

[81] ISO/IEC International Standard 14496 (MPEG-4). Information technology - coding of audio-visual objects. January 2000.

[82] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. *RFC 3550*, July 2003.

[83] A. Vetro, W. Matusik, H. Pfister, and J. Xin. Coding approaches for end-to-end (3d tv) systems. *Proceedings of Picture Coding Symposium (PCS)*, December 2004.

[84] B. Julesz. Foundations of cyclopeon perception. *The University of Chicago Press*, 1971.

[85] I. Dinstein, M.G. Kim, A. Henik, and J. Tzelgov. Compression of stereo images using subsampling transform coding. *Optical Engineering*, 30:1359–1364, September 1991.

[86] T. Ozcelebi, A. M. Tekalp, and M. R. Civanlar. Delay-distortion optimization for content-adaptive video streaming. *to appear in IEEE Transactions on Multimedia*, 2007.

[87] J. Reichel, H. Schwarz, and M. Wien. Scalable video coding - working draft 3. *JVT-R201, Bangkok, Thailand*, January 2006.

[88] C. A. Segall. Study of upsampling/down-sampling for spatial scalability. *JVT-Q083, Nice, FR, PL*, October 2005.

[89] B. L. Yeo and B. Liu. Rapid scene analysis on compressed videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 5:533–544, December 1995.

[90] N. D. Doulamis, A. D. Doulamis, Y. S. Avrithis, K. S. Ntalianis, and S. D. Kollias. An optimal framework for summarization of stereoscopic video sequences. *Proceedings of International Workshop on Synthetic - Natural Hybrid Coding and Three Dimensional Imaging, Santorini, Greece*, 1999.

[91] S. Wenger, M. M. Hannuksela, T. Stockhemmer, M. Westerlund, and D. Singer. RTP payload format for H.264 video. *RFC 3984*, February 2005.

[92] M. Handley and V. Jacobson. SDP: Session description protocol. *RFC 2327*, April 1998.

[93] http://ffmpeg.sourceforge.net/.

[94] Rec. ITU-R BT.500-11. Methodology for the subjective assessment of the quality of television pictures. 2002.

[95] N. V. Boulgouris and M. G. Strintzis. A family of wavelet-based stereo image coders. *IEEE Transactions on Circuits and Systems for Video Technology*, 12:898–903, 1999.

[96] A. Schertz. Source coding of stereoscopic television pictures. *Proc. IEE Inter. Conf. Image Processing and Its Applications, Maastricht, the Netherlands,*, pages 462–464, April 1992.

[97] L. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8:59–60, 1963.

[98] I. Das and J. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14:63–69, 1997.

[99] J. Lin. Multiple objective problems: Pareto-optimal solutions by method of proper equality constraints. *IEEE Transactions on Automatic Control*, 21:641–650, 1976.

[100] J. P. Ignizio. Goal programming and extensions. *Lexington Books, Massachusetts*, 1976.

[101] I. Das. Nonlinear multicriteria optimization and robust optimality. *Ph. D. Thesis, Dept. of Computational and Applied Mathematics, Rice University, Houston, TX, USA*, 1997.

[102] H. Papadimitriou and M. Yannakakis. Multiobjective query optimization. *Proceedings of PODS, California, USA*, 2001.

[103] Y. il Lim, P. Floquet, and X. Joulia. Multiobjective optimization considering economics and environmental impact. *ECCE2, Montpellier*, October 1999.

[104] A. Hanjalic, R. L. Lagendijk, and J. Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:580–588, June 1999.

[105] Y. Li, S. Narayanan, and C. C. J. Kuo. Content-based movie analysis and indexing based on audiovisual clues. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:1073–1085, August 2004.

[106] F. Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal on Visual Communication and Image Representation*, 8:146–166, 1997.

[107] R. Lienhart, W. Effelsberg, and R. Jain. VisualGrep: A systematic method to compare and retrieve video sequences. *Proceedings of SPIE*, 3312:271–282, 1998.

[108] M. M. Yeung, B. L. Yeo, W. Wolf, and B. Liu. Video browsing using clustering and scene transitions on compressed sequences. *Proceedings of SPIE*, 2417:399–413, 1995.

[109] S. W. Smoliar and H. J. Zhang. Content-based video indexing and retrieval. *IEEE Multimedia*, 1:62–72, 1994.

[110] F. Arman, R. Depommier, A. Hsu, and M.Y Chiu. Content-based browsing of video sequences. *ACM Multimedia*, pages 77–103, August 1994.

[111] B. L. Yeo and M. M. Yeung. Classification, simplification and dynamic visualization of scene transition graphs for video browsing. *Proceedings of SPIE*, 3312:60–70, 1998.

[112] K. Ntalianis, A. Doulamis, N.Doulamis, and S. Kollias. Unsupervised stereoscopic video object segmentation based on active contours and retrainable neural networks. *Signal Processing, Computational Geometry and Vision, World Scientific and Engineering Academy and Society Press*, 2002.