UNIVERSITY OF CALIFORNIA, SAN DIEGO

Perceptual Quality Assessment for Compressed Video

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Signal and Image Processing)

by

Kai-Chieh Yang

Committee in charge:

Professor Pankaj K. Das, Chair
Professor Clark C. Guest, Co-Chair
Professor Robert Bitmead
Professor William S. Hodgkiss
Professor Truong Nguyen

2007

The dissertation of Kai-Chieh Yang is approved, and it is acceptable in quality and form for publication on microfilm:

Co-Chair

Chair

University of California, San Diego

2007

To my family

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Completing a Ph.D has been a memorable journey in my life. It's consisted of extensive literature surveys, scientific exploration, and creative innovations. During this process, I received immeasurable help and inspiration from many people. First, I would like to express my appreciation to my advisers; Prof. Pankaj K. Das and Prof. Clark C. Guest, for giving me this opportunity to complete my Ph.D study, and I especially want to thank Prof. Clark C. Guest for his elaborate review of all my writing, technical comments, and many encouragements. Also, Prof. Truong Nguyen served as one of my committee members and provided me a golden opportunity to work with his students. I would like to thank my other committee members, Prof. Robert Bitmead and Prof. William S. Hodgkiss, for their participation, time, and valuable comments.

Several respectful friends, Dr. Hao-Hong Wang, Dr. Yen-Chi Lee, Dr. El-Maleh, Khaled, and Prof. Wei-Si Lin, selflessly shared their experiences and knowledge with me to shape and realize many ideas. Ai-Mei Huang helped facilitating my research by providing some valuable data. Moreover, I want to thank the people who participated in my subjective tests for their time and patience.

From a more personal perspective. I would like to thank my parents for giving me such a healthy body and good education. Finally, my most sincere gratitude goes to my wife for her strong support and great sacrifice.

PUBLICATIONS

Chapter 3

Kai-Chieh Yang, Clark C. Guest and Pankaj K. Das, “Human Visual Attention for Compressed Video”, *Proc. IEEE International Symposium on Multimedia*, pp. 525-532, Dec. 2006

Kai-Chieh Yang, Clark C. Guest and Pankaj K. Das, “Hierarchy Visual Attention Map”, *Proc. SPIE, Human Vision and Electronic Imaging XII*, Vol. 6492, Feb. 2006

Chapter 4

Kai-Chieh Yang, Clark C. Guest and Pankaj K. Das, “Perceptual Sharpness Metric for Compressed Video”, *Proc. IEEE International Conference of Multimedia and Expo*, pp. 777-780, July. 2006

Kai-Chieh Yang, Clark C. Guest, Pankaj Das, “Motion Blur Detection by Support Vector Machine”, *Proc. SPIE*, Vol. 5916, pp. 261-273, Aug. 2005

Chapter 5

Kai-Chieh Yang, Gokce Dane, and Khaled El-Maleh, “Temporal Quality Evaluation for Enhancing Compressed Video”, to appear at *Proc. IEEE 16th International Conference on Computer Communications and Network*, pp. 1160 - 1165, Aug., 2007

Kai-Chieh Yang, Clark C. Guest, Khaled El-Maleh and Pankaj K. Das, “Perceptual Temporal Quality Metric for Compressed Video”, *IEEE Transactions on Multimedia*, Volume 9, Issue 7, pp. 1528 - 1535, Nov. 2007

Kai-Chieh Yang, Clark C. Guest, Khaled El-Maleh and Pankaj K. Das, “Perceptual Temporal Quality Metric for Compressed Video”, 3rd International Workshop on *Video Processing and Quality Metric for Consumer Electronics*, Jan. 2007

Chapter 6

Kai-Chieh Yang, Ai-Mei Huang, Truong Nguyen, Clark C. Guest, and Pankaj K. Das, “A New Objective Quality Metric for Frame Interpolation Used in Video Compression ”, submitted to *IEEE Transaction on Broadcasting*

Kai-Chieh Yang, Ai-Mei Huang, Truong Nguyen, Clark C. Guest, and Pankaj K. Das, “New Objective Quality Metric for Frame Interpolation Using in Video Compression”, in *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. II - 177 - II - 180, Oct. 2007

VITA

1999	BS., Yuan-Ze University, Taiwan, R.O.C.
1999-2001	Manager of Simulation-and-Training team in advanced armor training center, Taiwan
2001-2002	Research Assistant of Magnetronic and Material Physics Lab. Nation Taiwan University, Taiwan
2004	MS., University of California - San Diego, Electrical and Computer Engineering Department
2005-present	Video quality analyst in Qualcomm Inc.
2007	Ph.D, University of California - San Diego, Electrical and Computer Engineering Department

ABSTRACT OF THE DISSERTATION

Perceptual Quality Assessment for Compressed Video

by

Kai-Chieh Yang
Doctor of Philosophy in Electrical Engineering
(Signal and Image Processing)

University of California San Diego, 2007

Professor Pankaj K. Das, Chair
Professor Clark C. Guest, Co-Chair

With multimedia research burgeoning, video applications have become essential to our daily life. However, as the compression becomes more aggressive, too much data loss results in degrading perceived video quality for viewers. Therefore, an accurate quality measurement is important to improve or preserve the quality of compressed video. This dissertation focuses on measuring the quality degradations that are caused by compression. We specifically target distortions with impact above the human perceptual threshold, which are also called artifacts. This type of distortion usually appears in a structured form. This characteristic makes quality assessment highly content dependent and many existing metrics fail in this regard. Some previous research has tried to raise the accuracy of video quality assessment by considering human visual system (HVS) effects, or human visual attention factors. However, both HVS and human visual attention have very strong interaction in the video quality assessment process, and none of the existing quality measurement research takes both of them into account. In addition, cognitive factors significantly influence the visual quality assessment process, but they have been ignored in current quality assessment research. Based on these realizations, a new video quality assessment philosophy is introduced in this thesis. It considers the characteristics of artifacts, effects from HVS, visual attention, and cognitive non-linearity. First, a new human visual module is proposed, it takes both visual masking and attention effects into account. Its unique design makes embedding this visual module in any video quality related applications very easy. Based on this new human visual module, a blurriness metric is designed which includes cognitive characteristics. This new blurriness metric does not rely on edge information, and is more robust at assessing heavily compressed video data. A metric for artifacts introduced by motion compensated field interpolation (MCFI) is also implemented. It is the first metric ever designed for measuring the spatial quality of temporally interpolated frames. From a temporal quality perspective, a novel temporal quality metric is designed to measure the temporal quality degradation caused by both uniform and non-uniform distributed frame loss. Experimental data shows these metrics significantly outperform the existing metrics.

1

Introduction

The most obvious shortcoming of existing quality metrics of compressed video is in treating quality measurement as a signal measurement issue without considering visual and cognitive effects. However, video quality is ultimately judged by human eyes using a complicated cognitive process. Quantifying quality of compressed video by only considering the presence or absence of video data will severely deviate from what humans perceive. Therefore, performance of video quality metrics can be dramatically improved by including those visual and cognitive effects. This improvement is demonstrated in this thesis by one new human visual module and three quality metrics.

1.1 Motivation

In past decades, video technology has developed quickly. As video materials migrate from analog to digital format, delivering video content has become an important, almost must-have, part of daily human life. Video signals can be delivered via various kinds of media, such as storage devices or networks. Among these, the most rapidly growing service is delivering video materials by wireless network. This service allows the subscriber to fetch or produce multimedia content almost any time and anywhere. Other reasons for its popularity are the increasing availability of computational power in hand held devices and mature wireless transmission technology. As this service becomes more and more popular, users' expectations of video quality have increased rapidly as well. Hence, the ability to deliver adequate quality to end users becomes increasingly important. Digital video data distributed through communication networks is subject to various kinds of distortions during acquisition, compression, processing, transmission, and reproduction. For example, lossy video compression techniques, which are almost always used to reduce the bandwidth needed to store or transmit video data, may degrade the quality during the source coding process. In another instance, digital video bitstreams delivered over error-prone channels, such as wireless channels, may be received imperfectly due to impairment occurring during transmission. Packet-switched communication networks can cause loss or severe delay of received data packets, depending on network conditions and the quality of service. All these transmission errors can result in distortions in the received video data.

In order to achieve the best balance between compression efficiency and human perceived quality, a number of different video encoding standards have been established. The Moving Picture Experts Group (MPEG), for example, has developed a number of standards including MPEG-1, MPEG-2 and MPEG-4. Other examples include the International Telecommunication Union (ITU) H.263 standard, and the emerging ITU H.264 standard. These video encoding standards generally support improved transmission efficiency of video sequences by encoding data in a compressed manner. However, most of the codecs are designed to compress video data by maximizing the Peak-Signal-to-Noise-ratio (PSNR), which is a normalized root mean square error between original and compressed video frames. Larger PSNR is taken to mean higher fidelity between original and compressed signals. But the PSNR has been proven not to correspond very well to human perception because of 1) human visual factors, 2) characteristics of quality degradation, and 3) added improving signal (i.e. additional signal used to sharpen image). Human visual factors cover the effects from the human visual system (HVS) and human visual attention. These effects reflect human sensitivities to various quality degradations in different spatial-temporal locations. In addition, quality degradation can be classified into several groups by its perceptibility for human eyes. The unique aspects of different types of quality degradation must be considered and the metric designed accordingly. Finally, in certain situations, human perceived quality can be improved by adding some additional signal (i.e. sharpening is usually implemented by adding some high frequency signals). However, the pixel-difference based quality metrics usually treat this as noise and thus give lower quality scores. Therefore, a perceptual quality assessment system that takes account of all the concerns above is highly desirable.

A perceptual video quality assessment system can be used either off-line or real-time. In off-line applications, the video quality assessment system can be used as an independent metric that provides insight to help the codec designer develop a coding strategy that minimizes the perceptual quality distortion. In real-time applications, the quality assessment system can be embedded with the video data receiver to estimate viewers' quality feedback. This information is used to guide a quality post-processing algorithm to enhance video quality in real-time. Also, this information can be sent back to the network server or transmission side to prevent quality loss by tuning the transmission or compression parameters. With this concept, perceptual video quality assessment system provides a balance between compression efficiency and human perceived quality.

As compressed video quality has became an important issue for the entire multimedia industry, some recently developed codecs have included quality enhancement modules in their base profile. The H.264 standard, the most advanced video codec, already includes a de-blocking filter as a built in post-processing module to reduce the most annoying compression artifact: *blockiness*, which usually given into an abnormal tiling structure on compressed images [4]. The fundamental working principle of the de-blocking filter is to apply a strong low pass filter to remove the abnormal blocking structure. Although this module is designed to adaptively low-pass filter compressed video according to some compression parameters, it still can not avoid increasing the annoying level of another important artifact: *blurriness*, which can be referred to as fuzzy or unclear content representation [4]. Hence, although blockiness can be effectively removed, blurriness has become the most pronounced artifact, and an accurate measurement of this kind of quality distortion is a very critical issue.

From a temporal quality aspect, as a video sequence is transmitted through a bandwidth-limited and error-prone channel, the video temporal smoothness at the receiving end might be degraded by:

1. The encoder skipping some frames during encoding in an attempt to reduce the data rate, while decoder might not be able to play all received frames because of limited computational capability.

2. An error-prone channel, in which packet loss may corrupt the video data and an entire frame may be lost. In the case of substantial frame loss, the viewer may observe motion freezing because most video decoders automatically repeat the last frame received before a dropped frame.

Since frame skipping is an important function in a codec's design, and frame loss caused by communication errors is unavoidable, a reliable way to measure the temporal quality is important. Once the level of temporal quality degradation is known, a post-processing algorithm, frame interpolation, can be applied to enhance the temporal quality by producing the missing frame after decoding. Since the interpolated frame is produced in an un-conventional way (separate from video compression), the nature of quality distortion caused by frame interpolation is very different from the quality degradation caused by compression. Thus, the quality metrics designed for compression artifacts can not be applied to assessing the quality degradation caused by frame interpolation. Therefore, a suitable metric for temporally interpolated frames is highly desired.

Most existing blurriness metrics use the pixels around content edges to quantify the strength of blurring artifacts. However, as an image is degraded by blurriness, edge detection will not be correct and the accuracy of blurriness estimation will decrease. In addition, several important human visual and cognitive factors that are crucial to blurriness estimation are not considered in most blurriness metrics. Based on these concerns, we will present a visual module that emulates various visual sensitivities to blurring artifact at different spatial and temporal locations with consideration of the effects from the HVS and visual attention. In the following, a novel blurriness metric will be proposed based on this visual module without relying on the edge information. In temporal quality assessment, most research estimates temporal quality based on the ratio of the number of lost and original frames. The final temporal quality output is calculated by adjusting this ratio by motion activity. This approach assumes that lost frames are distributed evenly through the whole sequence, and frame loss occurs in a fixed frequency fashion. However, in real applications, the temporal location of frame loss is uncertain and the duration of each frame loss instance varies. In this case, the temporal quality of a sequence is non-uniform and the temporal quality contrast of each individual frame loss occasion usually results in much more profound quality impact compared to the uniform case. Therefore, a new temporal quality metric will be introduced, including the attribute of temporal quality contrast. For the temporally interpolated frames, most related research in frame interpolation measures the algorithms' performance by PSNR or other fidelity measurements. However, because of the unique characteristics of the artifacts introduced during frame interpolation, such fidelity measurements can not accurately quantify the quality degradation caused by frame interpolation. Therefore, a more suitable metric for frame interpolation artifacts is proposed in this thesis with the consideration of visual attention, perceptual sensitivity to local quality contrast, and characteristics of the artifacts introduced by frame interpolation.

1.2 Contributions

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

- A Visual Blurriness Sensitivity Map (VBSM) including human visual attention and masking factors is defined . It not only considers the various sensitivities for different spatial-temporal locations introduced by visual attention, but also takes into account the suppression effect from visual masking. These characteristics make the VBSM suitable for quality assessment and enhancement related applications. In addition, the VBSM works in the frequency domain and as a block-based unit. Because most codecs transform video data into the frequency domain using a block-based compression, VBSM can easily be embedded into any block-based codec design and extended to other applications.

- A Perceptual Blurriness Metric (PBM) is designed to estimate the level of blurriness caused by compression. It can work without accessing original video data. Also, PBM employs the VBSM to emulate the various blurriness sensitivities at different spatial-temporal locations and adjusts the local blurriness score accordingly to improve the accuracy of blurriness measurement. Unlike many blurriness related metrics, PBM is insensitive to content type and is able to carry out cross-sequence comparisons. Moreover, PBM takes into account the non-linearity of the cognitive blurriness estimation process to make the objective blurriness scores closer to the blurriness humans perceive.

- An extensive study and comparison of designs, strengths, and weaknesses of several well known blurriness metrics is presented in this thesis. A performance comparison of all blurriness metrics, including PBM, is performed. Results show that most metrics work fine with video sequences compressed by the MPEG-4 codec, but except for PBM fail in estimating the blurriness introduced by the H.264/AVC codec. Since H.264/AVC is the most popular codec in modern multimedia industry, the success of PBM is a major advantage over other existing metrics.

- A temporal quality metric, the Perceptual Temporal Quality Metric (PTQM), is proposed and demonstrated. This metric can provide accurate temporal quality estimation without the original video data. Most existing temporal quality metrics use the number of frames lost in one second to measure temporal quality. Experimental results show this approach can only cover the case with evenly distributed frame loss, but is not sufficient to capture the temporal quality degradation caused by non-uniformly distributed frame loss, which happens more often than the former in practical scenarios. PTQM can successfully provide accurate measurements for both cases. In addition, PTQM outputs a hierarchical temporal quality report that includes the temporal quality for each frame and segment, up to the entire sequence. This characteristic allows users to freely zoom in or out to know the exact temporal quality from local to global scope. This property permits more flexibility for PTQM to combine with encoder design for a more adaptive temporal quality enhancement mechanism (i.e. adaptive frame skipping, and frame interpolation). Finally, because temporal quality is not only influenced by the number of lost frames, but also the level of motion, PTQM combines scene cut detection with motion estimation to achieve more accurate motion mapping for temporal quality estimation.

- Detailed investigation of the artifacts introduced by frame interpolation techniques is presented. The process of frame interpolation is very unlike the process of conventional video compression. The appearance of quality distortion caused by frame interpolation is also very different. Thus, conventional quality measurement methods can not be directly applied to measure the quality degradations introduced during frame interpolation. This thesis presents the first work to investigate this type of quality impairment. Furthermore, this analysis also addresses several weaknesses of the quality metrics that are commonly used for evaluating the spatial quality of interpolated frames.

- Based on the investigation of quality degradation caused by frame interpolation, a new metric, the Perceptual Frame Interpolation Quality Metric (PFIQM), is designed. The visual attention model in PFIQM is based on the characteristics of artifacts caused by frame interpolation. Since the accuracy of quality measurement for many widely used frame interpolation metrics is often decreased by some unnoticeable distortion induced during frame interpolation, the PFIQM differentiates and disregards this type of distortion to ensure high quality prediction performance. In addition, contrasted with compression artifacts, the quality distortion caused by frame interpolation often aggregates in a small region. This leads many objective metrics to give high quality scores since the distorted area is small, but it actually produces a large impact on visual perception because of its high quality difference to neighboring temporal regions. This effect is considered in PFIQM. Subjective comparison shows that PFIQM's outputs are closer to human perceived quality than other metrics. A systematic performance comparison shows that PFIQM significantly outperforms all the other metrics.

1.3 Structure of this thesis

This thesis is organized as follows: Chapter 2 provides detailed background information about video quality assessment and video compression techniques. The focus of Chapter 3 is to explain the process of constructing the VBSM. In Chapter 4, a detailed investigation of several existing blurriness metrics is presented, and a novel blurriness metric, PBM, is described. In Chapter 5, the phenomena of additional temporal quality degradation caused by local temporal quality difference is demonstrated and a new temporal metric, PTQM, is implemented. The characteristics of quality degradation caused by frame interpolation are discussed in detail and a new metric, PFIQM, designed specifically for interpolated frames is developed in Chapter 6. Finally, a summary and the conclusions of this thesis are presented in Chapter 7.

2

Background

In this chapter, several aspects of video quality are reviewed. Section 2.1 gives an introduction to digital video compression. The factors that cause video quality degradation are described at Section 2.2. Finally, representative video quality assessment research is reviewed in Section 2.3.

2.1 Digital Video Compression

Video compression has two important benefits: 1) It makes feasible transmission of video data through a variety of mediums, such as network or storage devices. Without compression, the size of raw (uncompressed) video data dramatically limits the usefulness of these transmission devices. For example, current Internet throughput rates are insufficient to acceptably handle raw video data. The most popular video storage device, Digital Versatile Disk (DVD), can carry only very limited raw video data at television-quality resolution and frame rate. Thus, DVD-Video storage would not be practical without video and audio compression. 2) Video compression enables more efficient use of transmission and storage resources, resulting in more versatile multimedia applications. Even in a high bitrate transmission channel, it is more desirable to send a multiple channel high-resolution compressed video than to send a single channel, low-resolution stream. Even with constant advance for increasing the capacity of transmission devices, people still desire higher-resolution, better quality video data through these mediums. Therefore, video compression is an essential component of multimedia services for the foreseeable future.

The fundamental concept of information compression is *redundancy removal*. Redundancy can be understood as the components that are not necessary for complete reproduction of the data. The fundamental concept of compression is to avoid allocating too many resources to targets with similar representation. In a *lossless* compression scheme, the compression target's redundancy is removed so that the original signal can be perfectly reconstructed at the receiver. This sounds great at first, but with current lossless compression techniques, it can only achieve a very moderate amount of compression. The other more practical compression scheme is *lossy* compression. This can achieve a larger compression ratio (i.e. ratio of the size of the raw data to the size of the compressed data) with the penalty that the decoded signal is not identical to the original. In summary, information compression is used to achieve as much compression as possible while minimizing the quality degradation of the reconstructed signal.

The key aspect of digital video compression is to reduce the video data size by removing different types of redundancy. The redundancy of video data can be categorized into *temporal* and *spatial* attributes. In the temporal domain, there is usually a high correlation (similarity) between frames of video that were captured around the same time. Temporally adjacent frames (sucessive frames in time order) are often highly correlated, especially if the temporal sampling rate (frame rate) is high. Figure 2.1 shows a pair of frames captured in a consecutive manner using a frame rate of 30 frame per second (fps). It shows clear evidence of temporal redundancy, since most of the image remains unchanged between these two frames. A simple compression method for removing temporal redundancy is to utilize the frame difference, where only pixel differences between successive frames are coded. Higher compression can be achieved using motion estimation, a technique for describing a frame based on the content of nearby frames by means of motion vectors. By compensating for the movements of objects in this manner, the differences between frames can be further reduced. Figure 2.2 shows an example of spatial redundancy. In the region marked by a cross, there is very little variation in the content of the image and thus there is significant spatial redundancy.

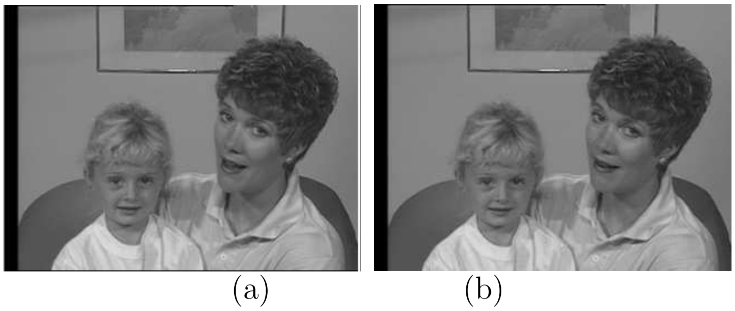


Figure 2.1: Example of temporal redundancy where (a) is the $(n-1)$th frame and (b) is the nth frame

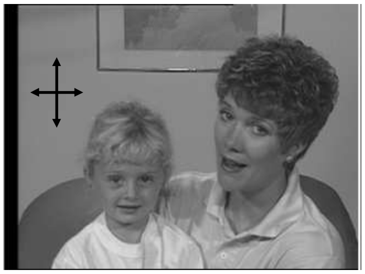


Figure 2.2: Example of spatial redundancy where the marked region has very little content variation

Video Compression Fundamentals

The major video compression standards released since the early 1990s have been based on the same generic design. Figure 2.3 shows the system diagram of a generic framework for video compression. Any codec that is compatible with H.261, H.263, MPEG-1, MPEG-2, MPEG-4, and H.264 standards [5–14] implements this set of basic coding and decoding functions [3]:

1 Mode decision: A video sequence is composed of two types of frames, namely *intra*-, and *inter*- frames. An intra-frame is usually referred as an I-frame. A general intra-coding process is shown in Fig. 2.4. This coding method is very similar to the JPEG image compression standard [5, 6]. The I-frame data goes through the transform, quantization, and entropy coding stages without any temporal redundancy removal. The inter-frames utilize temporal redundancy to achieve further video data compression. This step looks for similar content in selected reference frames and compress only the differential information. A gen-

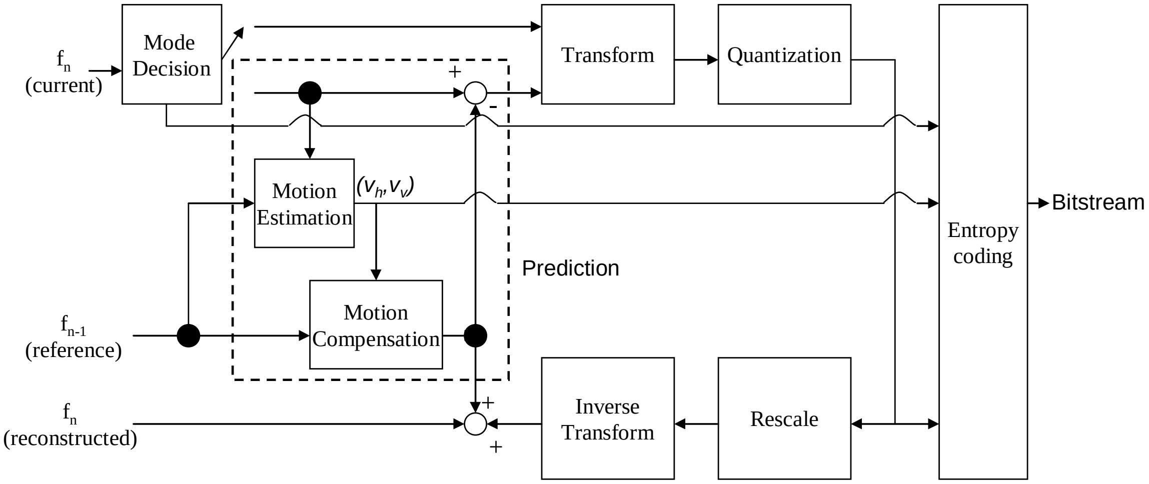


Figure 2.3: Generic system diagram for video compression

eral system diagram of inter coding is shown in 2.5. Distinct from intra-coding, the inter-coding requires the current frame and the reference frame for temporal redundancy removal, which is processed by the prediction module. Then, the predicted data are transformed, quantized, and entropy coded. The inter-frames can be further separated into forward predicted (P), and bi-directionally predicted (B) frames. The structure of a sequence of frames is typically arranged in IBBPBBPBBPBBPBBIBBP....and so on, as shown in Fig. 2.6. As depicted in Fig. 2.6, P-frames are compressed by searching for similar content from past reconstructed reference frames (i.e. I- or P-frames). The B-frame take both the past and future reconstructed frames as reference frames to carry out temporal redundancy removal. The mode decision module selects one of these three compression modes for each frame based on the structure of Group of Pictures (GOP). Together, I- and P- frames are called anchor frames. They are used as the basis for temporal redundancy removal in bi-directionally coded B-frames.

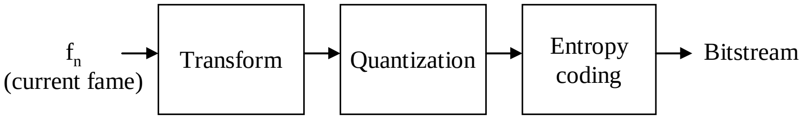


Figure 2.4: System diagram of intra-coding

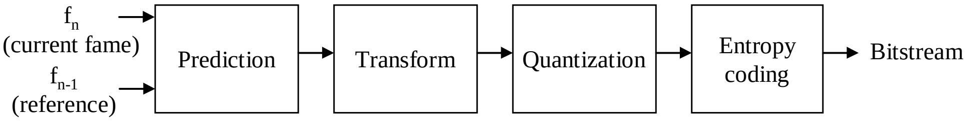


Figure 2.5: System diagram of inter-coding

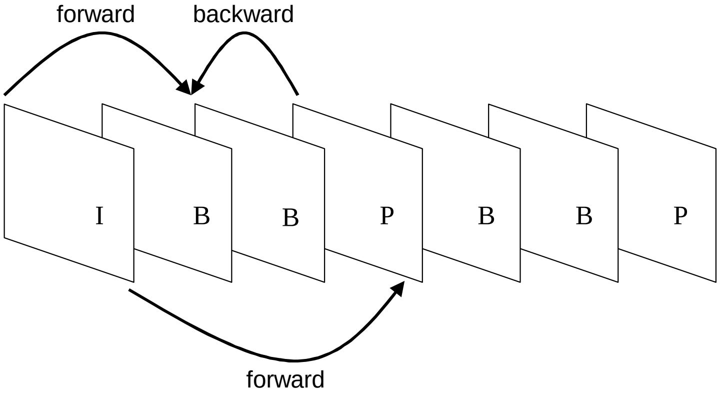


Figure 2.6: Structure of a sequence of frames and their relationship when carrying out temporal redundancy removal

2 Prediction: This process reduces the size of the data by removing temporal redundancy. The temporal redundancy reduction between transmitted frames is carried out by forming a predicted frame and subtracting this from the current frame. The output of this process is a residual (difference) frame and the more accurate the prediction process, the less energy is contained in the residual frame. The residual frame is compressed and transmitted to the decoder which independently re-creates the predicted frame, adds the decoded residual, and reconstructs the current frame. The predicted frame is created from one or more past or future frames. However, the correlation between current and reference frames decreases as spatial displacement of moving objects on these two frames occurs; this results in a large residual. In order to reduce the energy of residual data, the content of the reference frame is re-arranged according to the motion trajectory of moving objects. This process is called - *motion estimation*. An illustration is shown in Fig. 2.7. The displacement of moving objects between two frames is a *motion vector*, which is denoted as $(\mathrm{v}_h, \mathrm{v}_v)$ in horizontal and vertical directions respectively. Consider f_{n-1} and f_n are the $(n-1)$th and nth frames respectively, where f_n is the current frame and f_{n-1} is a reference frame

for f_n. Take the (i'_0, j'_0)th block in nth frame, $f_n(i'_0, j'_0)$, as an example. The residual data of $f_n(i'_0, j'_0)$ is calculated by taking the difference between the block with the smalles difference in the reference frame, $f_{n-1}(i'_M, j'_M)$, and the block in current frame - $f_n(i'_0, j'_0)$. This process is known as *motion compensation.* The motion vectors and residual data are sent to the following compression modules suitable for transportation of the data format. This prediction process is only applied on P- and B-frames, by which a set of prediction values is created (often based in part on an indication sent by the mode decision module).

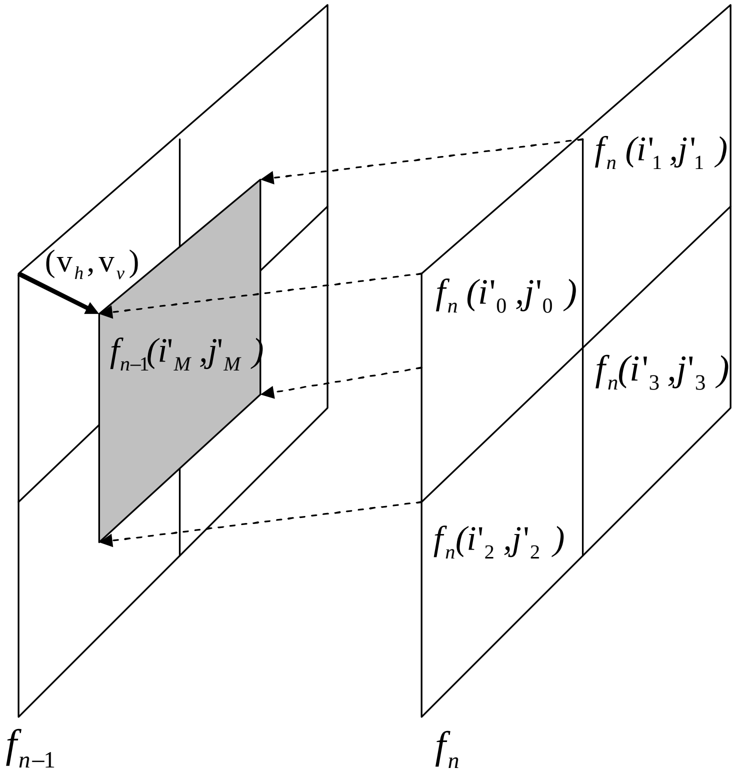


Figure 2.7: Illustration of the motion estimation process. White blocks represent blocks in the current frame subject to motion estimation and the gray region is the block in the reference frame that has the least difference to current block.

3 Transformation: The frames or motion-compensated residual data are transformed into the frequency domain to facilitate removal of the psychovisual redundancies by decomposing the data into several independent sets. Also, human eyes have very limited ability to recognize data in some spatial frequency bands

(i.e. high frequency), which means data in those bands have less impact on visual quality. Hence, transforms allow us to separate the data in those bands from the bands that carry more significant information. More resources can be allocated on those important bands by reducing the data precision in the bands that are less important. This can decrease the compressed data size without sacrificing appreciable visual quality. The choice of transform depends on a number of criteria: 1) Data in the transform domain should be decorrelated, 2) the transform should be reversible, and 3) the transform should be computationally tractable. The transforms can be separated into block- and image-based. The block-based transforms include the Karhunen-Love transform (KLT), which is an optimal decorrelator, and the Discrete Cosine Transform (DCT), which has performance close to that of a KLT when applied to highly correlated auto-regressive sources. Each of these operate on blocks of an image or residual samples, Hence the image is processed in units of a block with $N_B \times N_B$ pixels, where N_B is chosen for effectiveness and computational efficiency. Block transforms have low memory requirements and are well-suited to compression of block-based motion compensation residuals. However, they tend to introduce some artificial blocking artifacts. Image-based transforms operate on the entire image of a frame. The most popular image transform is the Discrete Wavelet Transform (DWT). Image transforms such as the DWT have been shown to out-perform block transforms for still image compression, but tend to have higher memory requirements since the whole image is processed as a unit. Also, it does not work well with block-based motion compensation. For these reasons, the DCT has become the most popular transform for video compression techniques. Using the DCT to transform the (i', j')th block with dimensions $N_B \times N_B$ in the frame f_n into the frequency domain is given by

$$F_n(i', j', u, v) = \sum_{i=0}^{N_B-1} \sum_{j=0}^{N_B-1} f_n(i', j', i, j) k(i, j, u, v), \tag{2.1}$$

$$k(i, j, u, v) = \alpha(u) \cdot \alpha(v) \cos[\frac{(2i+1)u\pi}{2N_B}] \cos[\frac{(2j+1)v\pi}{2N_B}], \tag{2.2}$$

where

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N_B}}, & \text{for u} = 0 \\ \sqrt{\frac{2}{N_B}}, & \text{for u} = 1,\, 2 \ldots N_B - 1, \end{cases} \tag{2.3}$$

and likewise for v, where (i, j) represents the index of a pixel, and u, v represent the DCT coefficient indices within a block. The DCT coefficients can be grouped into DC and AC as shown in Fig. 2.8. DC refers to $F_n(i', j', u, v)$ with $(u, v) = (0, 0)$, which is the DCT coefficient of the lowest band. The rest of the DCT coefficients are AC, where $F_n(i', j', u, v)$ for all $(u, v) \in (1, \cdots, N_B, 1, \cdots, N_B)$.

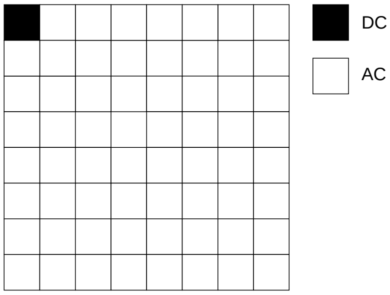


Figure 2.8: Illustration of a DCT block with dimensions 8 × 8, and the location of the DC and AC coefficients.

4 Quantization: After transformation, the numerical precision of the transform coefficients is reduced in order to decrease the data size. The degree of quantization applied to each coefficient is usually determined by the visibility of the resulting distortion to a human observer. For example, high-frequency coefficients can be more coarsely quantized than low frequency coefficients. A quantization process is composed of two parts: 1) scaling, and 2) re-scaling. A general example of scaling and re-scaling is:

$$\begin{aligned} F_{n,scaled} &= round(\frac{F_{n,org}}{QP}) \\ F_{n,re-scaled} &= F_{n,scaled} \cdot QP, \end{aligned} \tag{2.4}$$

where QP denotes the Quantization Parameter, which controls the degree of quantization , $F_{n,org}$, $F_{n,scaled}$, and $F_{n,re-scaled}$ represent the original, scaled, and re-scaled data respectively. If the QP is large, the range of quantized values is small and can therefore be efficiently represented (highly compressed) during transmission, but the fidelity between the re-scaled and original values is decreased. If the step-size is small, the re-scaled values match the original signal more closely, but the larger range of quantized values reduces compression efficiency. A *scalar quantiser* maps one sample of the input signal to one quantized output value and a *vector quantiser* maps a group of input samples to a group of quantized values.

5 Entropy coding: This is a process by which discrete-valued source symbols are represented in a manner that takes advantage of the relative probabilities of the various possible values of each source symbol. A well-known type of entropy code is the variable-length code (VLC), which involves establishing a tree-structured code table that uses short binary strings to represent symbol values that are highly likely to occur and longer binary strings to represent less likely symbol values. The best-known method of designing VLCs is the well-known Huffman code method, which produces an optimal VLC. A somewhat less well-known method of entropy coding that can typically be closer to optimal than VLC coding, and can also be more easily designed to adapt to varying symbol statistics, is the newer technique referred to as arithmetic coding.

MPEG-4 and H.264/AVC are the codecs used for simulation in this thesis. An overview of these two codecs will be presented in the following sections, and a comparison of these two codecs is shown in Table 2.1.

Table 2.1: Comparison of MPEG-4 and H.264/AVC from Ref. [3]

Comparison	MPEG-4	H.264/AVC
Number of profiles	19	3
Compression efficiency	Medium	High
Support for video streaming	Scalable coding	Switching slices
Motion compensation minimum block size	8 × 8	4 × 4
Motion vector accuracy	half or quarter-pixel	quarter-pixel
Transform	8 × 8 DCT	4 × 4 DCT approximation
Built-in de-blocking filter	No	Yes

Overview of the MPEG-4 compression

In Fig. 2.3, the data flow of a compression process of the MPEG-4 codec can be separated into 1) the encoding process, and 2) the decoding process. The encoding flow is as follows [3]:

1 An input video frame f_n is presented for encoding and is processed in units of a macroblock (corresponding to a 16 × 16 luminance region and associated chrominance samples).

2 The f_n is compared with a reference frame, for example the previous encoded frame: f'_{n-1}. A motion estimation function finds a 16 × 16 region in f'_{n-1} (or a sub-sample interpolated f'_{n-1}) that closely matches the current macroblock in f_n. The offset between the current macroblock position and the chosen reference region is used to calculate the a motion vector.

3 Based on the chosen motion vector, a motion compensated prediction is generated.

4 The predicted region is subtracted from the current macroblock to produce a residual or difference macroblock.

5 The residual macroblock is transformed using the DCT. Typically, the residual macroblock is split into 8 × 8 or 4 × 4 sub-blocks and each sub-block is transformed separately.

6 The coefficients of each sub-block are quantized.

7 The quantized DCT coefficients are re-ordered and coded.

8 Finally, the coefficients, motion vectors, and associated header information for each macroblock are entropy encoded to produce the compressed bitstream.

The reconstruction data flow is as follows:

1 Each quantized macroblock is re-scaled, and inverse transformed to produce a residual. This is usually not identical to the residual data before quantization.

2 The motion compensated prediction is added to the re-scaled residual to produce a reconstructed macroblock and the reconstructed macroblocks are saved in the frame buffer: f'_n.

Overview of H.264/AVC Compression

H.264/AVC is the newest video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. It is designed primarily to support efficient and robust coding and transport of rectangular video frames. Common applications include two-way video communications (video-conferencing or video-telephony), coding for broadcast and high quality video, and video streaming over packet networks. Support for robust transmission over networks is built in and the standard is designed to facilitate implementation on as wide a range of processor platforms as possible. The general structure of a H.264/AVC codec is shown in Fig. 2.9. Comparing Fig. 2.9 against the general compression structure in Fig. 2.3, the H.264/AVC has one more function - Intra prediction. This technique also existed in MPEG-4, but H.264/AVC uses it more efficiently. The following paragraphs describe it in more detail.

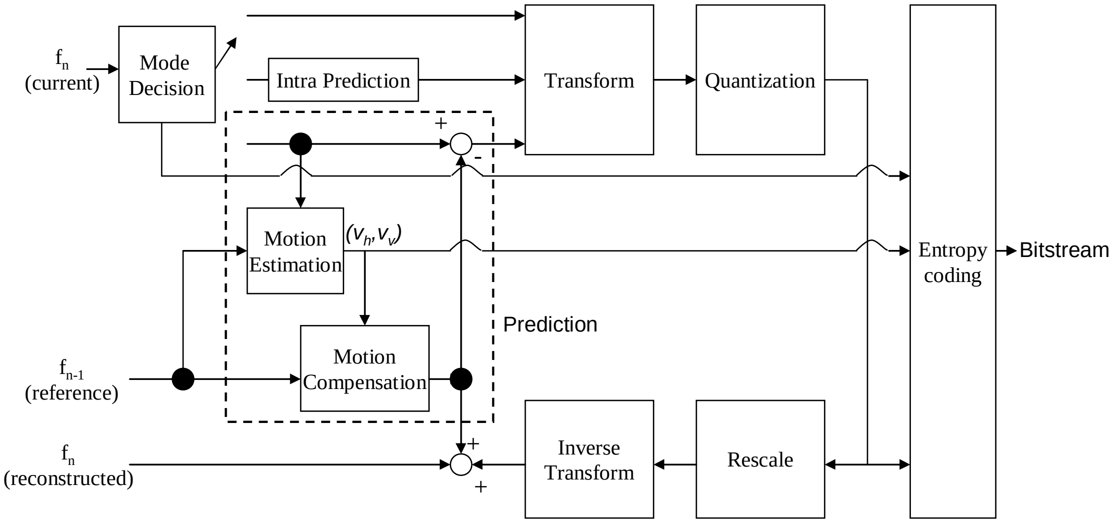
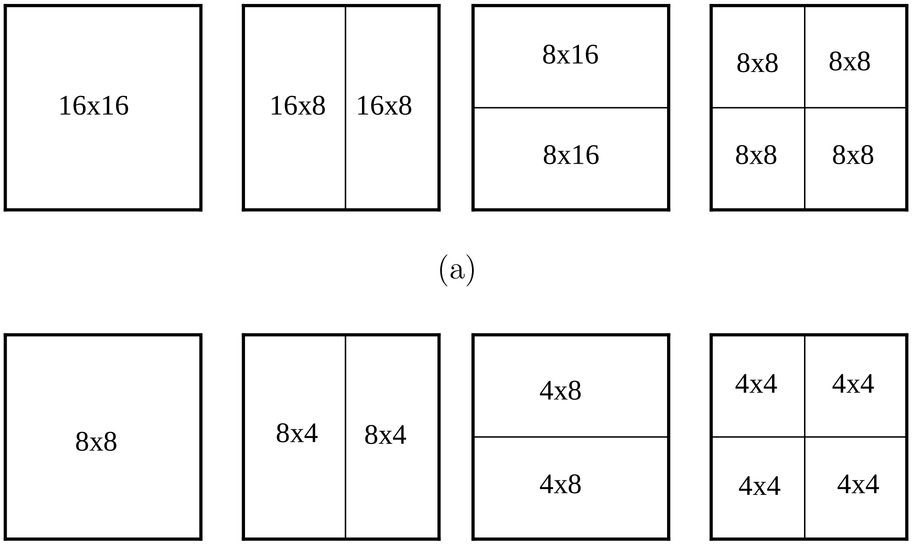


Figure 2.9: A general system diagram of H.264/AVC codec.

Relative to prior video coding methods, some highlighted features designed in H.264/AVC for enhancing coding efficiency include the following [15] :

1) Variable block-size motion compensation with small block sizes: The H.264/AVC standard supports more flexibility in the selection of motion compensation block sizes and shapes than any previous standard, with a minimum luma motion compensation block size as small as 4×4. This characteristic allows the encoder to carry out motion compensation more adaptive to object shape. All possible macroblock partitions for H.264 are shown in Fig. 2.10. A 16×16 macroblock can be divided into four different partitions as shown in Fig. 2.10(a). Furthermore, the 8×8 partition has four additional sub-partitions as shown in Fig. 2.10 (b).



(a)

(b)

Figure 2.10: Illustration of different size blocks in H.264 motion compensation, (a) shows a set of macroblock partitions, and (b) shows the sub-partitions of an 8×8 macroblock partition.

2) Quarter-sample-accurate motion compensation: Most prior standards (in MPEG-1, MPEG-2, and H.263) enable half-sample motion vector accuracy at most. The new design improves on this by adding quarter-sample motion vector accuracy, as first found in an advanced profile of the MPEG-4 Visual (part 2) standard, but further reduces the complexity of interpolation processing compared to the prior design. The quarter-sample motion vector refers to the use of spatial displacement motion vector values that have greater than integer pre-

cision, thus requiring the use of interpolation for the searching target when performing motion prediction. This is being done by first creating a half pixel frame by up-sampling 2 × 2 using a 6 tap FIR filter. Then generate a quarter pixel frame from the half pixel frame by up-sampling 2 × 2. Finally, the motion vectors with quarter-sample accuracy are obtained by carrying out a block search motion estimation on the quarter pixel frame.

3) Motion vectors over picture boundaries: While motion vectors in MPEG-2 and its predecessors were required to point only to areas within the previously-decoded reference picture, the picture boundary extrapolation technique first found as an optional feature in H.263 is included in H.264/AVC.

4) Multiple reference picture motion compensation: Predictively coded pictures (called P pictures) in MPEG-2 and its predecessors used only one previous picture to predict the values in an incoming picture. The new design extends the enhanced reference picture selection technique found in H.263++ to enable efficient coding by allowing an encoder to select, for motion compensation purposes, among a larger number of pictures that have been decoded and stored in the decoder. The same extension of referencing capability is also applied to motion compensated bi-prediction, which is restricted in MPEG-2 to using two specific pictures only (one of these being the previous I- or P- frame in display order and the other being the next I- or P- frame in display order).

5) Decoupling of referencing order from display order: In prior standards, there was a strict dependency between the ordering of pictures for motion compensation referencing purposes and the ordering of pictures for display purposes. In H.264/AVC, these restrictions are largely removed, allowing the encoder to choose the ordering of pictures for referencing and display purposes with a high degree of flexibility, constrained only by the total memory capacity bound imposed to ensure decoding ability. Removal of this restriction also enables removing the extra delay previously associated with bi-predictive coding.

6) Decoupling of picture representation methods from picture referencing capability: In prior standards, pictures encoded using some encoding methods (namely

bi-predictively-encoded pictures) could not be used as references for prediction of other pictures in the video sequence. By removing this restriction, the new standard provides the encoder more flexibility and, in many cases, the ability to use a picture for referencing that is a closer approximation to the picture being encoded.

7) Weighted prediction: A new innovation in H.264/AVC allows the motion compensated prediction signal to be weighted and offset by amounts specified by the encoder. This can dramatically improve coding efficiency for scenes containing fades, and can be used flexibly for other purposes as well.

8) Improved inference for skipped and direct motion: In prior standards, a skipped area of a predictively-coded picture could not move in the scene content. This had a detrimental effect when coding video containing global motion, so the new H.264/AVC design instead infers motion in skipped areas. For bi-predictively coded areas (called B slices), H.264/AVC also includes an enhanced motion inference method known as direct motion compensation, which improves further on prior direct prediction designs found in H.263++ and MPEG-4 Visual.

9) Directional spatial prediction for intra-coding: A new technique of extrapolating the edges of the previously-decoded parts of the current picture is applied in regions of pictures that are coded as intra (i.e., coded without reference to the content of some other picture). This improves the quality of the prediction signal, and also allows prediction from neighboring areas that were not coded using intra coding. Every luminance macroblock has three different intra prediction - intra_16 $\times$ 16, intra_4 $\times$ 4, and intra_PCM(transmit the image data directly without prediction). Figure 2.11 shows some sample modes of Intra_4 $\times$ 4. The Intra_4$\times$4 has eight directional Intra prediction modes and one DC mode, which simply takes the average of the referenced data regardless of direction.

10) In-the-loop de-blocking filter: Block-based video coding produces artifacts known as blocking artifacts. These can originate from both the prediction and residual difference coding stages of the decoding process. Application of an adaptive de-blocking filter is a well-known method of improving the resulting video

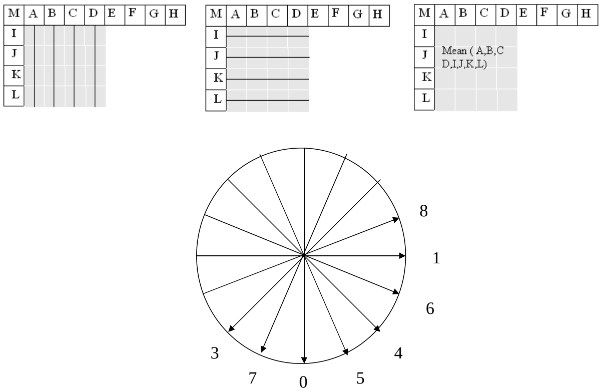


Figure 2.11: Samples of different modes of Intra_4×4 prediction, and all possible Intra prediction directions for Intra_4 × 4, where the blocks with alphabetic characters are the referred data during Intra prediction and the gray blocks are the predicted data.

quality. When designed well, this can improve both objective and subjective video quality. Building further on a concept from an optional feature of H.263, the de-blocking filter in the H.264/AVC design is brought inside the motion-compensated prediction loop, so that this improvement in quality can be used in inter-picture prediction to improve the ability to predict other pictures as well. The de-blocking filter functions as a low-pass filter to smooth out the abrupt pixel value changes around blocks' boundaries. Figure 2.12 shows an example of pixel value change around a block boundary, where p0-p3 and q0-q3 represent the pixels on the left and right hand side of a block boundary. In order to avoid removing details that belong to original content, the de-blocking filter will be applied on the block boundary only if all the following criteria are true:

(a) $|\text{p0-q0}| < \alpha \cdot QP$,

(b) $|\text{p1-q0}| < \beta \cdot QP$,

(c) $|\text{q1-q0}| < \alpha \cdot QP$.

where α and β determine the strength of de-blocking filter and higher value means stronger filtering.

11) Small block-size transform: All major prior video coding standards used a transform block size of 8 × 8, while the new H.264/AVC design is based primarily

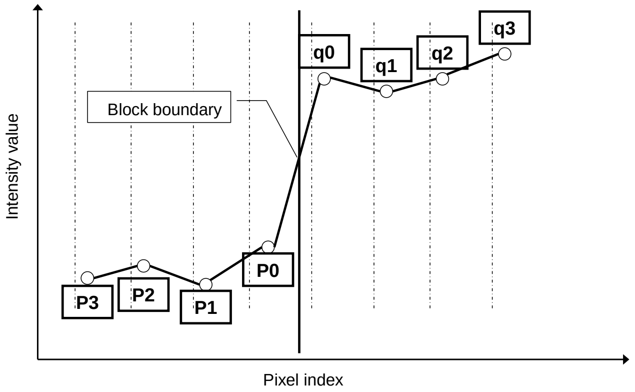


Figure 2.12: The p0-p3 and q0-q3 represent pixels at the left and the right hand side of a block boundary.

on a 4 × 4 transform. This allows the encoder to represent signals in a more locally-adaptive fashion, which reduces artifacts known colloquially as ringing. (The smaller block size is also justified partly by the advances in the ability to better predict the content of the video using the techniques noted above, and by the need to provide transform regions with boundaries that correspond to those of the smallest prediction regions.)

12) Hierarchical block transform: While in most cases, using the small 4 × 4 transform block size is perceptually beneficial, there are some signals that contain sufficient correlation to call for some method of using a representation with longer basis functions. The H.264/AVC standard enables this in two ways: 1) by using a hierarchical transform to extend the effective block size, used for low-frequency chroma information to an 8 × 8 array and 2) by allowing the encoder to select a special coding type for intra coding, enabling extension of the length of the luma transform for low-frequency information to a 16 × 16 block size in a manner very similar to that applied to the chroma.

13) Exact-match inverse transform: In previous video coding standards, the transform used for representing the video was generally specified only within an error tolerance bound, due to the impracticality of obtaining an exact match to the ideal specified inverse transform. As a result, each decoder design would produce slightly different decoded video, causing a drift between encoder and decoder representation of the video and reducing effective video quality. Build-

ing on a path laid out as an optional feature in the H.263++ effort, H.264/AVC is the first standard to achieve exact equality of decoded video content from all decoders.

14) Arithmetic entropy coding: An advanced entropy coding method known as arithmetic coding is included in H.264/AVC. While arithmetic coding was previously found as an optional feature of H.263, a more effective use of this technique is found in H.264/AVC to create a very powerful entropy coding method known as CABAC (context-adaptive binary arithmetic coding).

15) Context-adaptive entropy coding: The two entropy coding methods applied in H.264/AVC, termed CAVLC (context-adaptive variable-length coding) and CABAC, both use context-based adaptivity to improve performance relative to prior standard designs.

2.2 Sources of Digital Video Degradation

From a visual perceptibility perspective, video quality distortion can be separated into sub-, near-, and supra-threshold categories according to its perceptibility to human vision [16]. This threshold can be thought of as the visual perceptibility to distortion or, more simply, the minimum distortion a human observer will notice. The sub- and near-threshold distortions are types that are either not or only slightly able to be perceived by human eyes respectively. Supra-threshold distortion generally appears in a structured form and is known as an *artifact*. This type of distortion is very irritating to human perception and largely dominates subjective quality judgements.

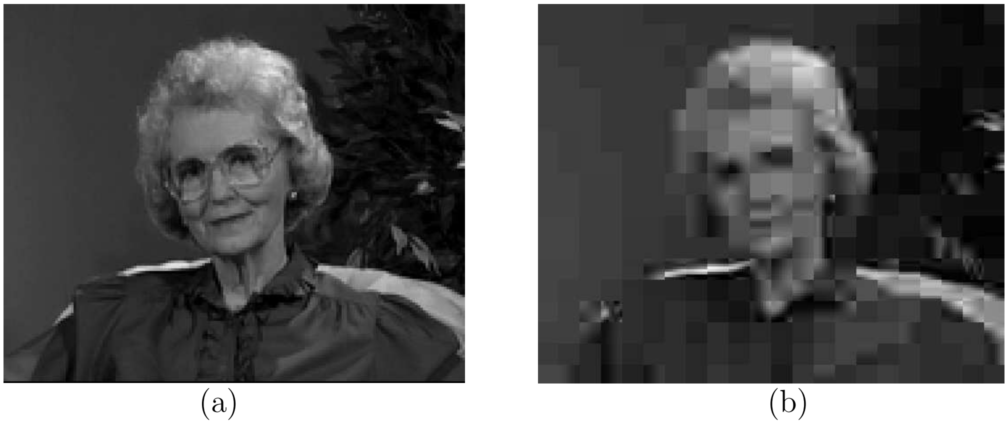
In general, causes of digital video quality distortion are compression and transmission errors. In compression, the video quality is mainly degraded by information loss during video data size reduction. Also, some post-processing (i.e. the de-blocking filter) might be applied on video data during or after decompression. Its purpose is to reduce artifacts caused by the source coding process, but it also can introduce other artifacts (e.g. blurriness). With regard to transmission errors, most transmission networks produce errors, and this results in another type of information loss. The appearance of this type of information loss is quite different than the information loss caused by compression. Therefore, the quality degradations caused by these two sources are discussed separately.

2.2.1 Compression Artifacts

Most compression standards rely on a block-based DCT with motion compensation and subsequent quantization of the coefficients. In such coding schemes, compression distortions are often caused by the quantization of the transform coefficients. Although other factors affect the visual quality of the stream, such as motion prediction or decoding buffer size, they do not introduce any distortion, but affect the encoding process indirectly through the quantization scale factor. Artifacts are commonly categorized as:

1 Blockiness: This refers to a abnormal block pattern in the decompressed sequence. It is caused by the independent quantization of individual blocks (usually 8×8 pixels in size) in block-based DCT coding schemes, leading to discontinuities at the boundaries of adjacent blocks. The blocking artifact is often the most prominent visual distortion in a sequence compressed in a block-based fashion because of its regularity and the extent of the pattern. Figure 2.13 shows an example of the blocking artifact, where Fig. 2.13(a) is the original image and Fig. 2.13(b) is the heavily compressed image. As we can see, an abnormal tiling artifact appears at the block boundaries and it severely degrades the compressed image's quality.

2 Blurring manifests itself as a loss of spatial detail and a reduction of edge sharpness. Blurring artifacts can be classified into three types - (1) motion blur, (2) out-of-focus blur, and (3) compression blur. The first type of blurriness is caused by camera aim moving during video capture [17], or the long response time Liquid Crystal Display (LCD). Appearance of this type of blurriness is the same with the output of a directional low pass filtering; the energy of high frequency signal components smear out along the camera motion direction. The second type of blurriness is usually caused by misplacing the camera focus, and makes the image detail smeared isotropically around each pixel. The third type of



(a) (b)

Figure 2.13: Samples of blocking artifact. (a) the original image, and (b) the heavily compressed image with blocking artifact.

blurriness usually occurs during compression. It is caused by the suppression of high-frequency coefficients during the image/video data redundancy removal process. The blurriness introduced by compression can be caused by: (1) aggressive data reduction: quantization is the main cause of data size reduction in image/video compression. It eliminates the high frequency data on the assumption that the human vision system has higher tolerance to high frequency data loss. However, as quantization becomes too aggressive, many spatial details are lost and the video is perceived as fuzzy. (2) Strong post-processing: Some recently developed codecs include a quality enhancement module in their basic profile. H.264, the most advanced video codec, already includes a de-blocking filter as its built-in post-processing module to reduce the most annoying compression artifact - blockiness. The working principle of the de-blocking filter is to apply a strong low pass filter to remove abnormal block structure. The taps and strength of the de-blocking filter are controlled by compression parameters, such as quantization step size. Nevertheless, strong filtering also results severe high frequency loss and blurriness is introduced. Figure 2.14 shows an example of this type of bluriness, where Fig 2.14(a) is the original image, and Figure 2.14(b) is the image compressed by H.264/AVC with QP=45 and post-processed by a strong de-blocking filter. Clearly, Fig. 2.14(b) omits many details compared to Fig. 2.14(a), specially in the high texture regions (i.e. spectators).

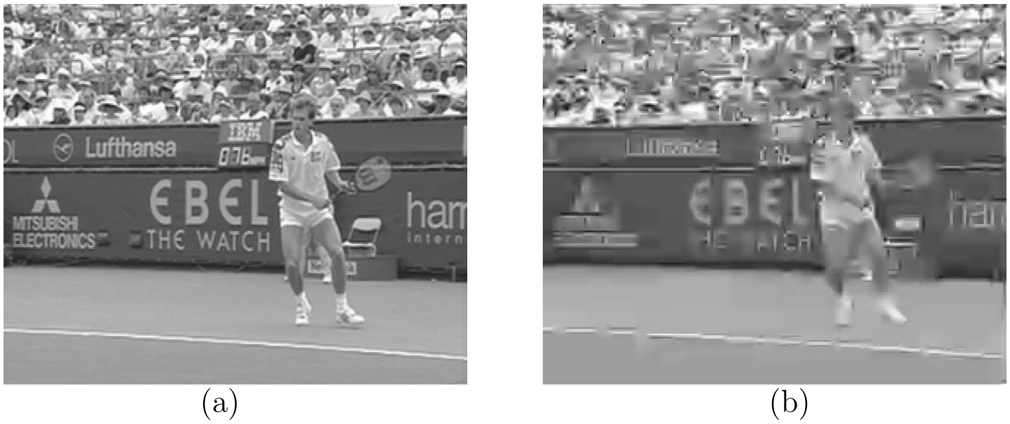
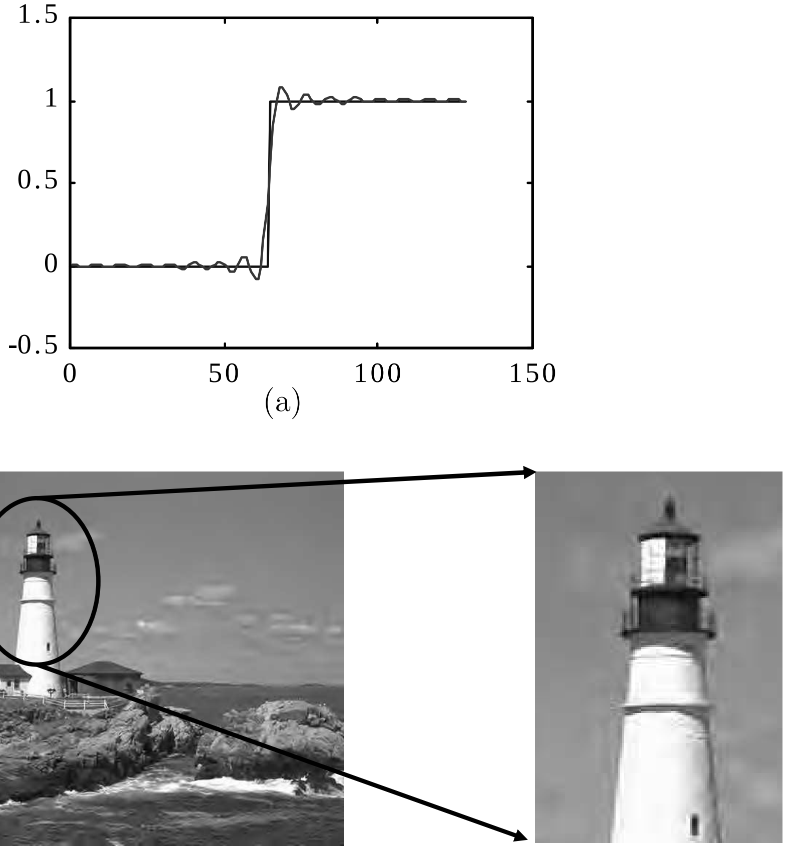


Figure 2.14: Examples of blurring artifact where (a) is the original image, and (b) is the heavily compressed image with severe blur.

3 The ringing artifacts is a manifestation of Gibbs phenomenon in the case of two-dimensional signals. The Gibbs phenomenon named after the American physicist J. Willard Gibbs, is the peculiar manner in which the Fourier series of a piecewise continuously differentiable periodic function behaves at a jump discontinuity. It is a consequence of trying to approximate a discontinuous function with a partial (i.e. finite) sum of continuous functions. A finite sum of continuous functions is, by definition, continuous, and therefore cannot approximate the discontinuity (and the area "near" it) to within any arbitrarily chosen accuracy. An infinite sum of continuous functions can be discontinuous, and hence, does not exhibit the Gibbs phenomenon. In two-dimension data, this phenomenon leads to the values of pixels' oscillating around edges in compressed images. Figure 2.15(a) shows a one-dimensional illustration of pixel oscillation around edges, where the horizontal and vertical axis represent pixel location and pixel value respectively, the straight line represents the pixels before Gibbs phenomenon occurs, and the oscillating curve represents the pixel values reconstructed from frequency data with severe high frequency energy loss. Ringing is most evident along high-contrast edges in otherwise smooth areas (i.e. a large jump of a discontinuous function). It is a direct result of quantization leading to high-frequency irregularities in the reconstruction. Figure 2.15(b) shows an example of ringing. In Fig. 2.15(b), the tower has very strong edges against the flat sky. The image on right hand side is a magnified version of the tower.

As we can see, there are many pixel value oscillations around the tower's edges. This is the so called ringing artifact.



(a)

(b)

Figure 2.15: Samples of ringing artifact. (a) the illustration of Gibbs phenomenon, and (b) the compressed image with ringing artifact.

4 Color bleeding is the smearing of color between areas of strongly differing chrominance. Its cause is ringing in chrominance data, which results from the suppression of high-frequency coefficients of the chroma components. Due to chroma sub-sampling, color bleeding extends over an entire macro-block. Figure 2.16 shows an example of color bleeding. Obvious color interference can be found around regions with high color difference.

5 Flickering appears when a scene has high texture content. Texture blocks are compressed with varying quantization factors over time, which results in a visible flickering effect.

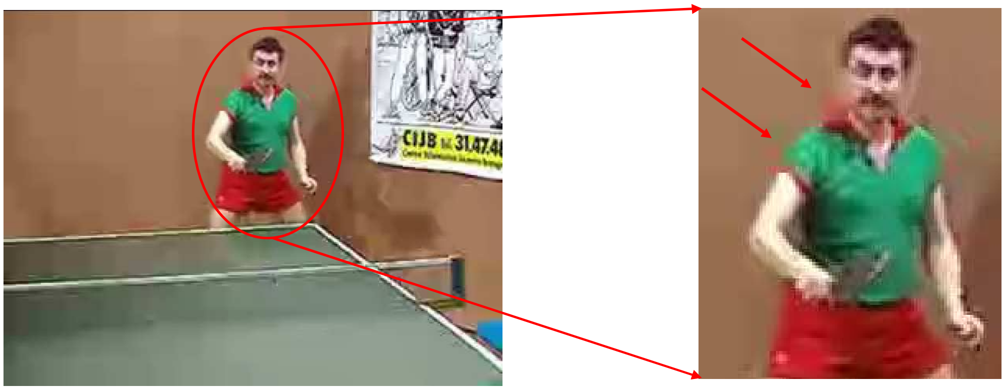


Figure 2.16: An example of color bleeding, where the left image is the highly compressed image, and the image on right hand side is zoomed in on the regions with significant color bleeding.

6 Ghost is a result of temporal low-pass filtering of the source video sequence, and it appears as a blurred remnant trailing behind fast moving objects. The tendency of temporal low-pass filtering is to eliminate the noise in the original sequence with the assumption that the noise in each frame has very low correlation along temporal axis. The method to achieve noise removal is to represent the current frame with a weighted average of its neighboring frames. However, if a video sequence contains high motion objects, which is not noise but has low correlation compared to non-moving regions. Then part of the moving object from neighboring frames might appear in the current frame, but with lower intensity. It is so called ghost artifact. Figure 2.17 shows an example of ghost artifact. The foreman's face is a mixture of the mouth and nose from the neighboring and current frames.

7 Jitter refers to irregular frame dropping. It can be introduced by either compression or the presence of transmission errors. On the decoder side, video frames can be dropped by a playback system that is not efficient enough to decode and display each video frame at the required speed. On the encoder side, frames may be dropped because of a sudden increase of motion in the video content, which can cause the encoder to discard frames in order to prevent an increase of the encoding bit rate while maintaining a certain level of picture quality. Figure 2.18(a) shows an example of jitter, where the black bars represent the

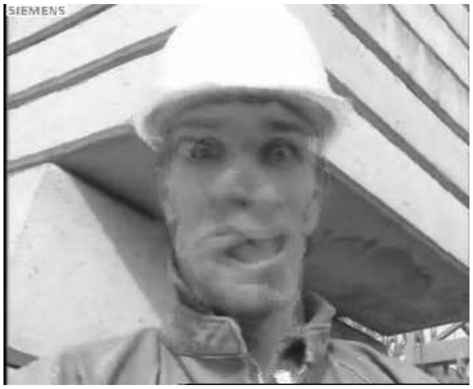


Figure 2.17: Example of ghost artifact, where the man's face is a mixture of nose and mouth from previous and current frames.

frames received successfully, dashed bars are the lost frames, and the numbers represent the index or time stamp of received frames. We can observe that the interval of frame loss is not consistent, and hence, viewers will perceive irregular motion freezing.

8 Jerkiness is the result of regularly skipping video frames to reduce the amount of video information that the system is required to encode or transmit. This creates motion perceived as a series of distinct snapshots, rather than smooth and continuous motion. Figure 2.18(a) shows an example of jerkiness. The frame loss occurs regularly every two frames and also the number of lost frames is consistently two. Viewers will perceive consistent discontinuous motion.

2.2.2 Transmission Errors

A very important source of video impairment comes from the transmission of the video stream over an error-prone channel. Digitally compressed video is primarily transferred over packet-switched networks. In this scenario, two main types of transmission impairments can occur. Packets can be lost or they can be delayed to the point where they are not received in time for decoding. Both will result in the same effect on the decoded information: a portion of the video stream is missing. This partial loss of information can have a dramatic impact on users' perceived quality since the loss of a single packet can result in a corrupted macro-block. Corrupted information can subsequently spread both spatially to neighboring blocks and tem-

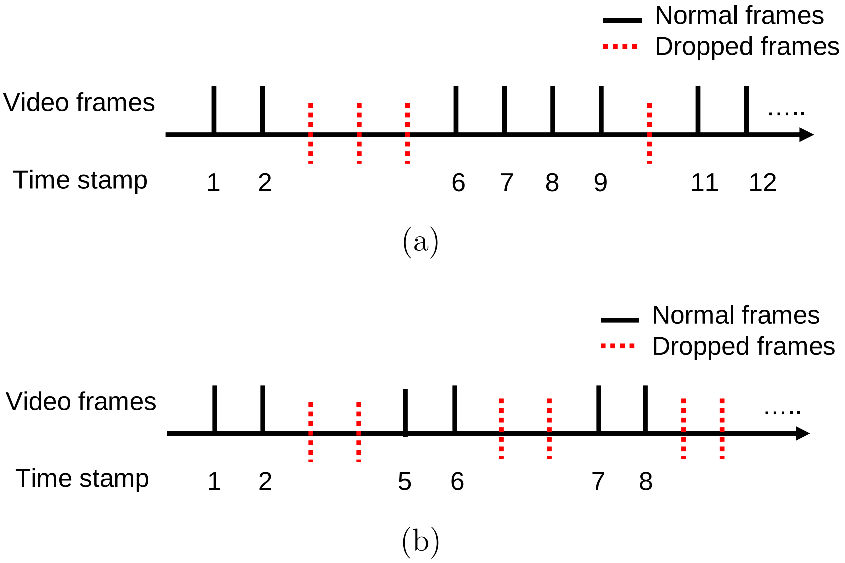


Figure 2.18: An illustration of video temporal quality degradation, where black bars represent the frame received successfully, dashed bars are the lost frames, and the numbers represent the index or time stamp of received frames. Figure (a) represents jitter artifact, and (b) represents jerkiness artifact. Humans will perceive irregular motion freezing in the case with jitter artifacts, and consistent frame freezing in the case with jerkiness artifacts.

porally over adjacent frames because most video encoders use differential predictive coding and motion compensation. The loss or corruption of a single macro-block can therefore affect the stream up to the next re-synchronization point (e.g. next slice, next intra-coded frame). The visual impact of such losses varies between video decoders depending on their ability to deal with corrupted streams. Some decoders hardly recover from certain errors, while others will apply more or less complex error concealment mechanisms. However, in some applications, decoders will choose to entirely discard the frame that has corrupted or missing information and repeat the previous video frame instead, until the next valid decoded frame is available. This is an entirely different situation from error concealment scenarios since one or several complete video frames are missing. No additional spatial degradations are introduced but frame repetition and frame drop occur. Thus, both jitter and jerkiness might occur.

2.3 Overview of Video Quality Assessment

Generally speaking, video quality can be measured in two ways: subjective and objective methods.

2.3.1 Subjective Video Quality Assessment

Subjective testing is the most reliable method because it measures the most direct response from end users. It requires viewers, e.g. members of the public, to view video clips and assign a quality score. Average quality for a processed video sequence is known as Mean Opinion Score (MOS). Two methods, single stimulus continuous quality evaluation (SSCQE) and double stimulus continuous quality scale (DSCQS), have been demonstrated to give the most repeatable and stable results, provide consistent viewing configurations and subjective reports, and have consequently been adopted as parts of an international standard, ITU-R BT. 500 [18], by the International Telecommunications Union (ITU). If the SSCQE and DSCQS tests are conducted on multiple human subjects, the scores can be averaged to yield the MOS. The standard deviation between the scores may also be useful to measure the consistency between subjects.

1 *Single Stimulus Continuous Quality Evaluation*: In the SSCQE method, subjects continuously indicate their impression of the video quality on a linear scale that is divided into five segments. The five intervals are marked with adjectives to serve as guides. An example of grading value and semantic meaning is shown in Fig. 2.19. The viewers are instructed to move a slider to any point on the scale that best reflects their impression of quality at that instant of time, and to track the changes in the quality of the video using the slider.

2 *Double Stimulus Continuous Quality Scale* The DSCQS method is a discrimination based method, and has the extra advantage that the subjective scores are less affected by adaptation and contextual effects. In the DSCQS method, a reference and a compressed video form a test case. The procedure for a test case is illustrated in Fig. 2.20, where period T1 shows either reference or test image data, T2 shows a gray image for buffering. The DSCQS method first presents

the reference, then the test data to participants. Subsequently, this image pair is repeated but in random order and participants vote on a quality score. The length of each video data is restricted to a small number of seconds; 10 to 15 sec is recommended. Participants evaluate both test and reference sequences using sliders similar to those for SSCQE. The difference between the scores of the reference and the distorted sequences gives the subjective impairment judgment, and it is defined as Differential-Mean-Opinion-Score (DMOS).

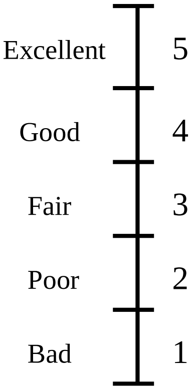


Figure 2.19: Example of subjective grading and its corresponding semantic expression.

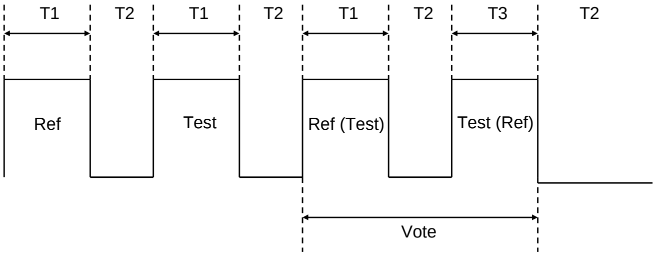


Figure 2.20: Procedure of DSCQS subjective test, where period T1 that shows either reference or test image data, and T2 shows a gray image for buffering.

However, results of subjective testing can be affected by several experimental conditions (i.e. lighting condition, display order of testing materials, etc.). The experiment needs to be designed carefully. In addition, in order to increase the reliability of subjective data, the number of samples (i.e. participants) can not be few. Hence, carrying out a subjective test is very resource consuming and inconvenient. It is neither a practical nor scalable solution for a live application. Objective methods provide the alternative solution.

2.3.2 Objective Video Quality Assessment

The goal of objective image/video quality assessment research is to design quality metrics that can predict subjective image/video quality without carrying out subjective experiments. A good objective metric should correlate well with subjective data. Since the objective video quality evaluation method consumes less resources than the subjective method, it permits objective metrics to be applied to more versatile applications. Hence, an objective image/video quality metric can be employed for the following functions:

1 It can be used to monitor image quality for quality control systems. For instance, an image/video acquisition system can monitor the quality metric and automatically adjust related parameters to obtain the best quality image and video data rate. A network video server can control the quality of video streaming by monitoring feedback of the quality of the digital video transmitted over the network.

2 It can be used to help the system designer while developing new image/video processing systems and algorithms. If multiple video processing systems are available for a specific task, then a quality metric can help in determining which one of them provides the best quality results.

3 It also can be embedded into an image and video processing system to optimize the algorithms and allow the algorithm to adjust its parameters autonomously. For instance, in a visual communication system, a quality metric can be employed to optimize the parameters of the post-processing and bit allocation algorithms on the encoder side for optimal reconstruction, error concealment, and post-processing algorithms on the decoder side.

Objective image and video quality metrics can be classified according to the availability of the original image or video signal, which is considered to be distortion-free, or perfect quality, and may be used as a reference for comparison to a distorted image or video signal. Most of the proposed objective quality metrics in the literature assume that the undistorted reference signal is fully available. Although ”image and video quality” is frequently used for historical reasons, the more precise term for this type of metric would be image and video similarity or fidelity measurement, or full-reference (FR) image and video quality assessment. It is worth noting that in many practical video service applications, the reference images or video sequences are often not accessible. Therefore, it is highly desirable to develop measurement approaches that can evaluate image and video quality blindly. Blind or no-reference (NR) image and video quality assessment turns out to be a very difficult task, although human observers usually can effectively and reliably assess the quality of distorted image or video without using any reference. There exists a third type of image quality assessment method, in which the original image or video signal is not fully available. Instead, certain features are extracted from the original signal and transmitted to the quality assessment system as side information to help evaluate the quality of the distorted image or video. This is referred to as reduced-reference (RR) image and video quality assessment.

The most well known FR objective image and video distortion/quality metrics is peak signal-to-noise ratio (PSNR), which is defined as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \tag{2.5}$$

where MSE is the mean-square-error, which is given by

$$MSE = \frac{1}{N_C \cdot N_R} \sum_{i=1}^{N_R} \sum_{j=1}^{N_C} (f_x(i,j) - f_y(i,j))^2, \tag{2.6}$$

N_C and N_R are the numbers of rows and columns of an image, and $f_x(i,j)$ and $f_y(i,j)$ are the values of (i,j)th pixel of the reference and degraded images respectively. PSNR is widely used because of its simplicity and clear physical meaning. However, it also has been widely criticized for its poor correlation with subjective quality measurement. In other words, video content with the same PSNR value can result in different quality opinion scores. Similar approaches have tried to estimate delivered video quality using network parameters [19,20], such as the amount of packet loss. However, packet loss levels will not always give a meaningful quality score. The same level of packet loss can produce different types of degradations and therefore different levels of quality. Quality determined by the customer's perception is much more complex than the statistics that a typical network management system can provide, e.g. bit error rates or levels of packet loss. Aside from PSNR, quality evaluation research can be grouped approximately into three major categories, which are fidelity, artifacts and hybrid approaches [21, 22]. In fidelity approaches, the metrics are established based on knowledge of the human visual system (HVS) [23–30]. Figure 2.21a provides an example system diagram. The visual quality is interpreted from the perceivable distortion associated with several human visual factors, such as contrast sensitivity [31–33], pattern masking [34–36], etc. The artifacts approach [37,38] estimates the quality based on a priori knowledge of the same features, such as compression artifacts, and obtains the final quality by ad-hoc integration and is shown in Fig. 2.21b. Generally, the artifacts approach works well in some specific applications and permits a computationally efficient implementation. In designing a quality metric based on the artifacts approach, the accuracy of measuring the quality degradation caused by compression is crucial for ensuring good performance. The hybrid approach has been discussed in [28,39]. This approach predicts final quality by combining both fidelity and artifacts measurement; a system diagram of this approach is shown in Fig. 2.21c. The pooling module takes the perceptual saturation and artifacts masking phenomenon into account.

The development of individual metrics for specific compression artifacts can be separated into two major groups [4], which are the spatial and temporal domains. In the spatial domain, several well known artifacts and the related metrics, such as blockiness, blurriness and ringing etc., have been extensively studied in Refs. [1, 17, 40–47]. References [1, 17, 40–42, 48] focus on measuring the blurring artifact. Caviedes and Oberti [40] extend the work in Ref. [48] from measuring sharpness(i.e. the opposite of blurriness) of images captured under a microscope to compressed video. They compare the distribution of the compressed image's high frequency coefficients against a Gaussian model and utilize this deviation as a sharpness measurement. Marichal *et al.* [1] designed another blurriness estimation method in frequency domain by observing the distribution of the ratio of AC to DC coefficients. Marziliano and *et al.* [41] estimate the blurriness based on the width of edges; a blurred image has wide edge widths. Yang *et al.* [17] detect the blurriness caused by motion with a machine learning approach. The results show that the metric can detect motion blurriness but lacks resolution in determining an accurate blurriness level. In a later paper, Yang *et al.* [42] measure the blurriness by observing high frequency energy associated with

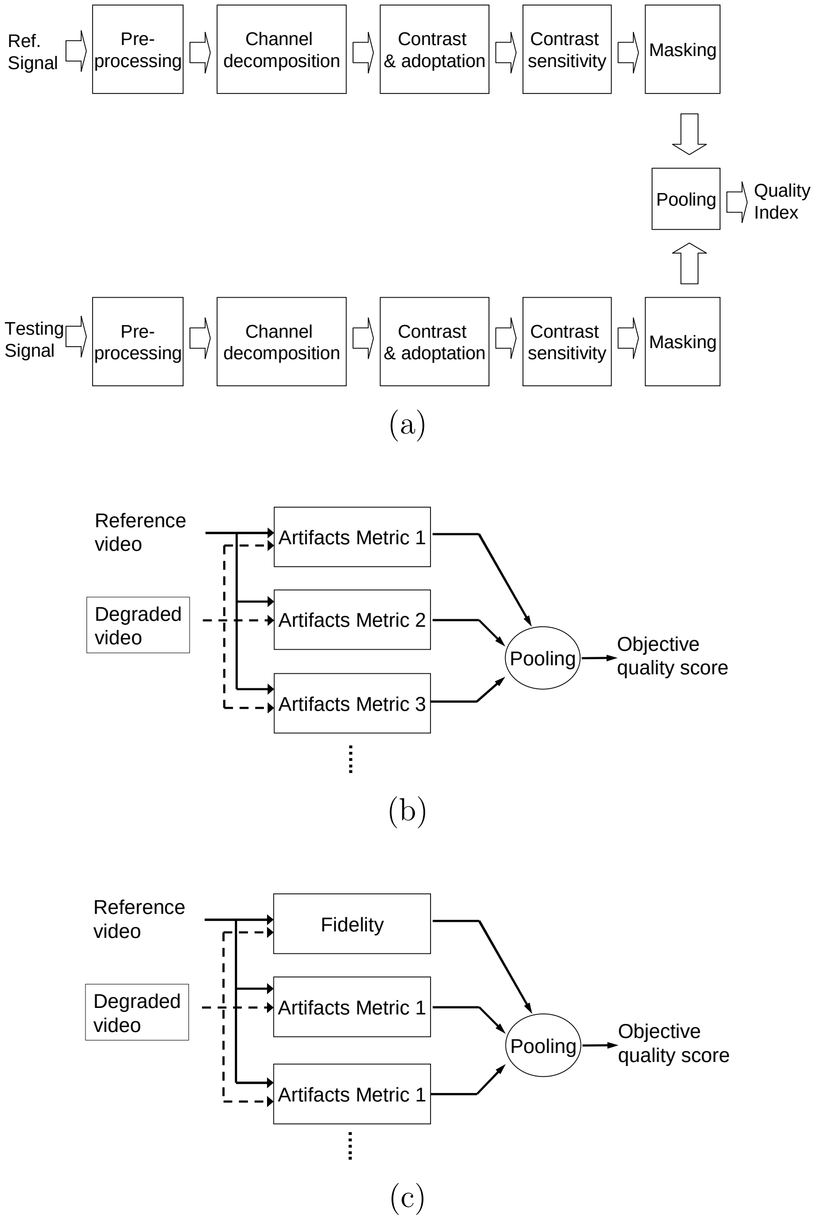


Figure 2.21: General system diagrams of (a) Fidelity, (b) Artifacts and (c) Hybrid approaches to video quality assessment

the effects from human visual factors. The focus of Ref. [43–46] is to determine the amount of blockiness. Wu [43, 44] proposed the first non-reference blockiness metric that considers texture and luminance masking. Karunasekera [45] designed a blockiness metric with an elaborate implementation of many HVS modules. Gao in Ref. [46] proposed a non-reference blockiness metric, and also designed a de-blocking algorithm guided by this metric. Marziliano [41] proposed a ringing metric that estimates the pixel oscillation by measuring the pixel variation around each edge pixel. Oguz *et al.* [47] implemented a ringing metric that also use pixel variation to determine the severity of ringing artifacts. But in addition to that, the estimated ringing level is adjusted by texture, and luminance masking effects.

In objective temporal quality metric designs, Feghali *et al.* [49] use frame rate as the scaling factor to adjust PSNR and output a spatial-temporal quality score. In Refs. [50] and [51], a jerkiness metric based on frame rate and motion activity is proposed. This metric in Ref. [50] is further applied to guiding a new transcoder. In Ref. [52], the inter-frame correlation is used to determine the location of lost frames. Some post-processing is conducted on the number of lost frames to extract several indices, such as the duration of group dropping and the number of group dropping occurrences, etc. The final temporal quality score is determined based on the ad-hoc analysis of those indices. Pastrana-Vidal and Gicquel [53] proposed a non-reference objective metric for measuring fluidity impairments in video service. This metric responds to their previous work [54] and takes the density of group dropping into account. Lost frames are detected by inter-frame dissimilarity on the decoder side. After thresholding, noticeable fluidity breaks are obtained. Each fluidity break is weighted by a function of the pixel variation in the last frame at the end of the freeze and the first frame appearing after the freeze. This stage tries to map the fluidity break to different types of motion. Afterward, the fluidity break is further adjusted by a function of the fluidity break density. The paper claims that the contribution to temporal quality degradation of the fluidity breaks with more occurrences is less significant. In other words, with the same amount of frame loss, the temporal quality with scattered fluidity breaks is better than with aggregated fluidity breaks. Watanabe et al. [55] studied subjectively the temporal distortion with different combinations of small group dropping with a fixed amount of frame loss. Based on that, they tuned a logarithmic function specifically for different combinations of frame loss in each sequence. This work provides important evidence that the same amount of frame loss within one sequence could lead to different levels of subjective temporal degradation through different combinations of aggregated frame loss. It shows the prediction accuracy of the logarithmic function can be improved by tuning parameters according to the duration of each grouped frame loss.

2.3.3 Performance Evaluation of Objective Quality Metrics

Since most objective quality metrics' outputs are content dependent, and also for the sake of data interpretation convenience, metrics' outputs are normalized by the max and min value of each sequence by

$$\mathcal{VQR}_n = 1 + 4\frac{VQR_n - \min(VQR)}{\max(VQR) - \min(VQR)} \tag{2.7}$$

where $\mathcal{VQR}_n$ represents a normalized metric's output for the nth frame, which ranges from 1 to 5, and higher value means better quality, VQR_n denotes a metric's raw output for the nth frame, and VQR is a set of the metric's raw outputs for a sequence. According to the Phase II Final Report from Video Quality Experts Group (VQEG) [56], the relationship between the metrics' outputs and the subjective quality ratings, $DMOS_s$, may not be linear, as subjective testing can have nonlinear quality testing compression at the extremes of the test range. In order to remove any nonlinearity caused by subjective rating process and to facilitate comparison of metrics in a common analysis space, normalized metrics' outputs are mapped by a nonlinear regression function as

$$DMOS_{o,n} = \frac{b(1)}{1 + exp[-b(2) \times (\mathcal{VQR}_n - b(3))]} \tag{2.8}$$

where $DMOS_{o,n}$ denotes the mapped objective score for the nth frame, and b is a set of parameters obtained by fitting the $\mathcal{VQR}$ of each metric against $DMOS_s$. As a result, each metric has a set of b parameters, and the corresponding $DMOS_o$ represents the objective scores that are closest to subjective ratings. Therefore, the best performance of each metric can be obtained by this mapping process.

After normalization and nonlinear transformation, outputs of all metrics range from 1 to 5 and higher value means better quality. The $DMOS_o$ are compared with the $DMOS_s$ values by computing the similarity using the following metrics:

(a) Pearson correlation coefficient (C_P): This metric is used to estimate the model prediction accuracy, which is the ability of the objective metric to predict subjective ratings with minimum average error,

$$C_P = \frac{\sum \Delta DMOS_{o,n} \cdot \Delta DMOS_{s,n}}{\sqrt{\sum \Delta DMOS_{o,n}^2 \cdot \Delta DMOS_{s,n}^2}}, \tag{2.9}$$

where $\Delta DMOS_{o,n} = DMOS_{o,n} - \overline{DMOS_o}$ and $\Delta DMOS_{s,n} = DMOS_{s,n} - \overline{DMOS_s}$, where $\overline{DMOS_o}$, $\overline{DMOS_s}$ are the mean values of mapped objective and subjective scores respectively. Larger C_P means higher prediction accuracy.

(b) Spearman rank order correlation coefficient (C_S): This coefficient is designed to determine the level of monotonicity by measuring the correlation of the decreasing(increasing) trend of both variables independent of the magnitude. C_S is given by

$$C_S = 1 - 6 \sum \frac{(DMOS_{o,n} - DMOS_{s,n})^2}{N(N^2-1)}, \tag{2.10}$$

where N is the number of the data point. Larger C_S means better prediction performance.

(c) Root-Mean-Square-Error (C_R): Root-Mean-Square-Error (RMSE) is the square root of the mean squared difference between objective and subjective values, and is given by

$$C_R = \sqrt{\frac{1}{N} \sum_n (DMOS_{o,n} - DMOS_{s,n})^2}. \tag{2.11}$$

Lower C_R means less deviation between subjective and objective data and better prediction performance.

In this chapter we have reviewed the fundamentals of digital video compression for the MPEG-4 and H.264/AVC standards. The nature of spatial and temporal artifacts introduced by aggressive compression or by transmission errors. Next, subjective and objective methods for video quality assessment were described. This included a review of relevant standards that have been established, as well as metrics for evaluating the accuracy of quality determination.

3

Visual Blurriness Sensitivity Map (VBSM)

3.1 Introduction

Automatic segmentation of attention-getting regions is a very interesting but challenging problem, and it can benefit a wide range of multimedia applications. For example, in video telephony applications the video encoder can allocate more resources to object regions that might be of interest to a viewer, to code them at higher quality to achieve better perceptual quality. As another example, for video quality evaluation, the segmented object can be allocated higher sensitivity when pooling the local quality values into a global (i.e. frame or sequence level) quality value. In [57], Itti *et al.* presented a saliency-based computational model for visual attention. The fundamental idea is that human visual attention focuses on objects having features distinct from their surroundings. In the first stage, several salient features are extracted based on psychovisual knowledge. Afterward, the salient features are combined to produce a saliency map. Finally, a winner-take-all neural network is used to determine the high attention locations in the saliency map. Yang *et al.* [58] proposed a perceptual sensitivity map based on several bottom-up human visual system (HVS) factors and skin tone. Furthermore, this sensitivity map is applied to bits allocation for a video telephony application. Lu *et al.* [59] introduced a visual attention module that can be deployed in several video quality evaluation systems. The attention module is produced by combining several bottom-up and top-down features. In the pooling stage, the dependence between features is considered. Then motion masking caused by camera motion is emulated to adjust the importance of each frame.

In summary, most of previous research only considers positive visual stimuli (i.e. visual interesting regions), but pay very little attention on negative visual stimuli (i.e. visual masking). As described in Sec. 2.3.2, masking is an important aspect of the HVS in modeling the interactions between different image components present at the same spatial/temporal location. Masking refers to the fact that the presence of one image component (i.e. texture) will decrease the visibility of another image signal (i.e. artifacts). The mask generally reduces the visibility of the image signal in comparison to the case where the mask is absent. Therefore, the visual attention model can not be applied to video quality evaluation directly, it must be combined with visual masking effects.

Since artifacts have various appearances, the visual masking model for each artifact is also slightly different. For example, blocking artifacts have less visibility in high texture regions, but on the other hand, those regions are important to determine blurring artifacts. Hence, the human visual module must have enough flexibility to include or adjust the masking components according to different characteristics of the evaluated artifacts. Based on this reason, a novel and extendable human visual module has been designed to emulate the perceptibility of blurring artifacts in the spatial and temporal domains by considering both human visual attention and masking phenomena.

3.2 Visual Blurriness Sensitivity Map

When evaluating the blurriness of a video, observers tend to allocate different attention to various spatial and temporal locations. This tendency is driven by several bottom-up (i.e., luminance contrast) and top-down (i.e. skin tone) stimuli. In addition, the blurriness at some spatial/temporal locations can not be correctly recognized because of masking phenomena. Less blurring sensitivity should be assigned to those regions. Hence, human visual attention and masking factors lead to different weights on various spatial and temporal locations. These weights are numerically emulated by a *Visual Blurriness Sensitivity Map (VBSM)*. The VBSM is composed of two major parts - 1)*Human Visual Attention Map (HVAM)* from Ref. [60], and 2) luminance and motion masking components. Figure 3.1 shows the procedure for obtaining a VBSM, where the white blocks are responsible to determine the level of human visual interest and the gray blocks are used to suppress the response using masking effects. In order to accommodate the system design of the blurriness metric that is going to be introduced in Chapter 4, and avoid computational complexity without sacrificing the accuracy of blurriness sensitivity determination, the input image is processed by a non-overlapped block-based DCT and separated into 64 sub-bands as absolute values using

$$F_n(i',j',u,v) = |\sum_{i=0}^{N_B-1}\sum_{j=0}^{N_B-1} f_n(i',j',i,j)k(i,j,u,v)|, \tag{3.1}$$

$$k(i,j,u,v) = \alpha(u)\cdot\alpha(v)\cos[\frac{(2i+1)u\pi}{2N_B}]\cos[\frac{(2j+1)v\pi}{2N_B}], \tag{3.2}$$

where

$$\alpha = \begin{cases} \sqrt{\frac{1}{N_B}}, & \text{for u = 0} \\ \sqrt{\frac{2}{N_B}}, & \text{for u = 1, 2} \ldots N_B - 1, \end{cases} \tag{3.3}$$

$f_n(i',j',i,j)$ represents the image data of (i,j)th pixel in (i',j')th block in nth frame, F_n is the image data in frequency domain, which is used as input for VBSM determination, u, v represent the DCT coefficient indices within a block, and N_B is the dimension of each DCT block. This type of input gives VBSM more flexibility to be combined with other video related applications, such as sharpening or bits-allocation. In the first layer of Fig. 3.1, several attention features, such as intensity, chrominance, texture, and motion, denoted as $\mathcal{I}_n^+, \mathcal{BG}_n^+, \mathcal{BY}_n^+, \mathcal{T}_n^+, \mathcal{M}_n^+$ respectively, are calculated. Meanwhile, the luminance masking value, $\mathcal{L}^-$, and camera motion activity are estimated. In the second layer, the outputs of attention and luminance masking features that belong to the same object are grouped together by post-processing in the spatial domain. Then, the values of spatially post-processed visual features and camera motion are temporally post-processed to avoid sudden changes in visual features' output along the time axis. Finally, camera motions are used to estimate the motion masking

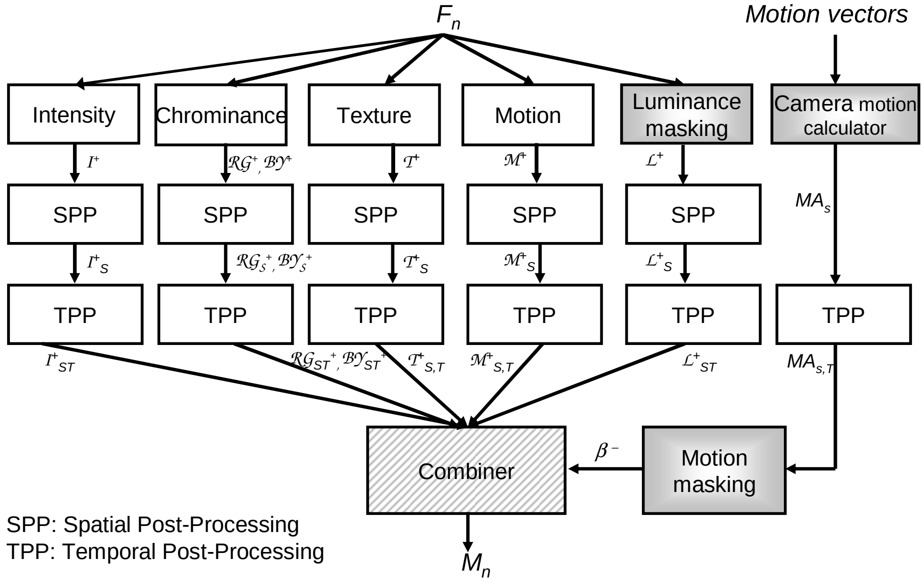


Figure 3.1: System diagram of Visual Blurriness Sensitivity Map (VBSM)

value, β^- and combined with all spatial-temporal post-processed visual features to form a VBSM.

3.2.1 Intensity attention feature

The response of human eyes depends much less on absolute luminance than its local variations. This property is known as the *Weber-Fechner law*. The intensity attention feature, $\mathcal{I}^+$, is a measure of this relative variation of the luminance at each pixel location and is given by

$$\mathcal{I}_n^+(i',j') = \max(|F_{n,I}(i',j',0,0) - F_{n,I}(i'_w,j'_w,0,0)|) \tag{3.4}$$

where $\mathcal{I}_n^+(i',j')$ denotes the intensity visual attention value of (i',j')th block in the nth frame, $(i'_w,j'_w) \in w$ and w is a 3×3 mask centered at the (i',j')th block.

3.2.2 Chroma attention feature

The chromatic attention feature is constructed based upon [57], which is the so called "color-double-opponent" system. It shows that human cortex activity is excited by one color (e.g., red) and suppressed by another color (e.g., green). That means the difference between a pair of specific color channels provides information for determining attention. The chromatic opponents, red/green and blue/yellow, denoted as $\mathcal{RG}^+$ and $\mathcal{BY}^+$ respectively, are estimated by

$$\mathcal{RG}^+_n(i',j') = \max(|RG_n(i',j',0,0) - GR_n(i'_w,j'_w,0,0)|) \tag{3.5}$$

$$\mathcal{BY}^+_n('i,j') = \max(|BY_n(i',j',0,0) - YB_n(i'_w,j'_w,0,0)|) \tag{3.6}$$

and

$$RG_n(i',j',0,0) = F_{n,R}(i',j',0,0) - F_{n,G}(i',j',0,0) \tag{3.7a}$$

$$BY_n(i',j',0,0) = F_{n,B}(i',j',0,0) - F_{n,Y}(i',j',0,0) \tag{3.7b}$$

$$GR_n(i'_w,j'_w,0,0) = F_{n,G}(i'_w,j'_w,0,0) - F_{n,R}(i'_w,j'_w,0,0) \tag{3.7c}$$

$$YB_n(i'_w,j'_w,0,0) = F_{n,Y}(i'_w,j'_w,0,0) - F_{n,B}(i'_w,j'_w,0,0) \tag{3.7d}$$

where $F_{n,R}$, $F_{n,G}$, $F_{n,B}$, and $F_{n,Y}$ denote the DCT coefficients of the red, green, blue, and yellow color channels respectively, where $F_{n,Y} = \frac{F_{n,R}+F_{n,G}}{2} - \frac{|F_{n,R}-F_{n,G}|}{2} - F_{n,B}$. The DCT coefficients of color channels can be obtained through the following conversion:

$$\begin{bmatrix} F_{n,R}(i',j',0,0) \\ F_{n,G}(i',j',0,0) \\ F_{n,B}(i',j',0,0) \end{bmatrix} = \begin{bmatrix} 1.16 & 0 & 1.59 \\ 1.16 & -0.39 & -0.81 \\ 1.16 & 2.01 & 0 \end{bmatrix} \tag{3.8}$$

$$\times \begin{bmatrix} F_{n,I}(i',j',0,0) - 128 \\ F_{n,Cb}(i',j',0,0) - 1024 \\ F_{n,Cr}(i',j',0,0) - 1024 \end{bmatrix}, \tag{3.9}$$

where $F_{n,Cb}$, $F_{n,Cr}$ are the DCT coefficients of up-sampled Cb and Cr chromatic video data respectively. The color conversion matrixes in the pixel domain are given by ITU-R Recommendation BT.601-4 [61] as

$$\begin{bmatrix} f_R \\ f_G \\ f_B \end{bmatrix} = \begin{bmatrix} 1.16 & 0 & 1.59 \\ 1.16 & -0.39 & -0.81 \\ 1.16 & 2.01 & 0 \end{bmatrix} \times \begin{bmatrix} f_I - 16 \\ f_{Cb} - 128 \\ f_{Cr} - 128 \end{bmatrix}. \tag{3.10}$$

The multiplication matrix and value shift constants from Equation (3.10) are transformed by Equation (3.1) into the color conversion matrix as shown in Equation (3.9).

3.2.3 Texture attention feature

The energy in high frequency bands is used to estimate the texture level of each block. Edge structure is more important than fine texture since the former has more structural texture pattern and draws more human attention than the latter. Therefore, only the medium frequency bands that are denoted as **M** in Fig. 3.2 [62] are used to determine the texture visual attention region. The value for the texture attention feature of each block, $\mathcal{T}^{+}$, is estimated by the magnitude of medium bands as

$$\begin{aligned} \mathcal{T}_n^{+}(i',j') &= \sum_{i=2}^{6} F_{n,I}(i',j',i,0) + \sum_{j=2}^{6} F_{n,I}(i',j',0,j) \\ &+ F_{n,I}(i',j',1,2) + F_{n,I}(i',j',2,1) \\ &+ F_{n,I}(i',j',2,2) + F_{n,I}(i',j',3,3). \end{aligned} \quad (3.11)$$

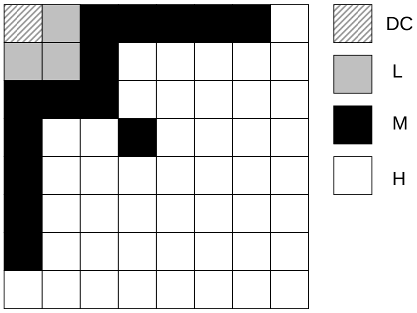


Figure 3.2: Illustration of 8×8 block DCT coefficients and different frequency bands, where DC, L, M, and H denote the DC, low, median, and high frequency bands respectively.

3.2.4 Motion attention feature

Motion is the major difference between video and still images. In video, a set of images is played sequentially and moving regions usually draw most of viewers' attention, while other features are not very conspicuous.

A common way to recognize moving regions is by using the magnitude of the motion vectors. However, because of rate control and compression efficiency optimization, motion vectors do not always match the true motions. Besides, the motion vector information is only available in inter-coded blocks; they are not available in the intra-coded blocks.

In order to cope with these concerns, a robust and low complexity background subtraction method is adopted from [63] to estimate the movement in each block. This approach assumes that the background scene belongs to regions with very minor movement and the foreground is in noticeable motion. However, this assumption is not always applicable. For a surveillance camera, the unmoving regions are determined to be background and the regions with movement are classified as foreground. But for a tracking camera, the determination is the other way around; the unmoving regions belong to foreground, because the aim of the camera moves along with the object and the background regions are in motion. Therefore, instead of foreground/background, we use the terms *moving-region(MR)* and *non-moving-region(NMR)* for the sake of clarity.

The motion of each block is estimated by observing the DC value distribution along the temporal axis of the R, G and B color channels. Figure 3.3 shows the procedure for the determination of motion attention regions. In order to find the likelihood belonging to NMR of the nth frame, the DC values of all blocks for all three color channels of nth and previous frames are given as input into a mixture Gaussian model. The current nth frame is compared against each of the previous frames and the deviations are given as input into a mixture Gaussian function given by

$$Pr_n(i',j') = \frac{1}{T}\sum_{t=1}^{T}\prod_{c}\frac{1}{\sqrt{2\pi\sigma_{n,c}^2}}e^{-\frac{1}{2}\frac{D_{t,c}^2}{\sigma_{n,c}^2}}, \tag{3.12}$$

where Pr represents the likelihood to NMR and a larger $Pr_n(i',j')$ means (i',j')th block in nth frame is more likely to be considered as part of the NMR, $c \in (R, G, B)$, $D_{t,c}$ is the deviation between the current and the tth prior frames in cth color channel, where $D_{t,c} = F_{n,c}(i',j',0,0) - F_{n-t,c}(i',j',0,0)$, T is the number of previous frames, which is set to 5 according to Ref. [63] and experimental data, and $\sigma_{n,c}^2$ is the variance of cth color channel of a given DC value. This variance can also be thought of the width of the mixture Gaussian kernel. In order to estimate $\sigma^2_{n,c}$ against some slight movement of background, the median value, $m_{n,c}$ of the DC value difference of all blocks, $|F_n(i', j', 0, 0) - F_{n-1}(i', j', 0, 0)|$, for a consecutive pair (F_n, F_{n-1}) in the sample is calculated independently for each color channel. The reason behind using the median of the absolute difference of all blocks' DC value of two consecutive frames is that DC values change as different objects appear on the same block but at different times; this is defined as the objects' movement. As measuring the deviation between two consecutive frames, most DC value pairs $[F_n(i', j', 0, 0), F_{n-1}(i', j', 0, 0)]$ have similar values except a few pairs with object movement. However, using the same reason, some slight and un-meaningful movement (e.g. leaves with random movement in the background) might also cause DC value differences, but this should be lower than the DC value difference caused by real object movement. In order to distinguish these two different movements, the median value of the deviation of DC value pairs has been recognized as the most robust threshold to filter out un-meaningful movements [63]. Let's assume that the pixel value distribution of an image can be approximated by a Gaussian distribution $N(0, \sigma^2)$, then the distribution for the deviation of two images is also Gaussian: $N(0, 2\sigma^2)$. Since the distribution is symmetric, the median of the absolute deviations $m_{n,c}$ is equivalent to a quarter of the deviation distribution as

$$\Pr(N(0, 2\sigma^2) > m_{n,c}) = 0.25, \tag{3.13}$$

and therefore the standard deviation is estimated as

$$\sigma_{n,c} = \frac{m_{n,c}}{0.68\sqrt{2}}. \tag{3.14}$$

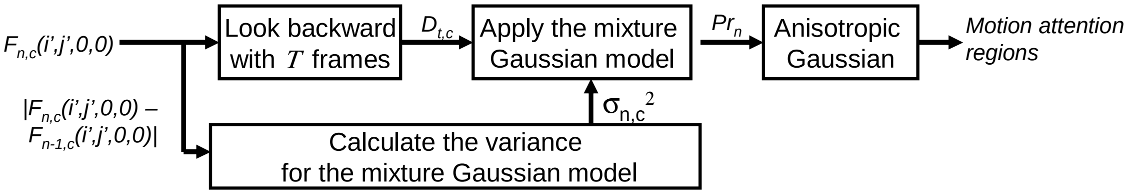

$F_{n,c}(i',j',0,0)$
Look backward with T frames
$D_{t,c}$
Apply the mixture Gaussian model
Pr_n
Anisotropic Gaussian
Motion attention regions
$|F_{n,c}(i',j',0,0) - F_{n-1,c}(i',j',0,0)|$
$\sigma_{n,c}^2$
Calculate the variance for the mixture Gaussian model

Figure 3.3: Procedure for the determination of motion attention regions.

When the camera tracks a moving object, the aim of the camera is on the object and the object usually stays in the central region of the video images [64]. Most of the NMR locates around the image central. On the contrary, the NMR for a static camera usually stays near the boundary of the image. An anisotropic Gaussian kernel given by

$$P_2(i',j') = \frac{1}{\sqrt{2\pi}(\sigma_x^2+\sigma_y^2)/2} e^{-\frac{1}{2}(\frac{(i'-i'_c)^2}{\sigma_x^2}+\frac{(j'-j'_c)^2}{\sigma_y^2})} \tag{3.15}$$

is used to differentiate these two types of camera motion by observing the locations of high motion regions, where (i'_c, j'_c) represents the index of the spatial central point of an image, and σ_x^2, σ_y^2 control the widths of the Gaussian distribution in the horizontal and vertical directions respectively. Suggested by Ref. [64], the values of σ_x^2 and σ_y^2 are set to 800 and 500 respectively. Figure 3.4 shows the outputs of Equation (3.15), where brighter intensity represents higher attention. The visual sensitivity aggregates in the central region and it decreases toward the image boundary. Hence, as more high motion occurs around the image boundary, output from Equation (3.15) will be smaller and the camera motion will be classified as a tracking camera. The block indices (i',j') in Equation (3.15) are replaced by (i'_o, j'_o), which are the block indices with considerable motion as $Pr_n(i'_o, j'_o) \leq 0.8$. The static and tracking camera motions are differentiated by

$$\text{camera motion} = \begin{cases} \text{static} & \text{if } G(n) > T_2 \\ \text{tracking} & \text{if } G(n) \leq T_2 \end{cases} \tag{3.16}$$

where $G(n) = \frac{1}{(N_C/8-1)\cdot(N_R/8-1)} \sum_{i'_o,j'_o} P_2(i'_o, j'_o)$, where N_C and N_R are the width and height of an image; T_2 is the threshold to determine the camera motion type; it is empirically set to 0.08. Because visual attention caused by motion changes with camera motion, the final motion attention regions are given by

$$\mathcal{M}_n^+(i',j') = |\eta - Pr_n(i',j')|, \tag{3.17}$$

where

$$\eta = \begin{cases} 1 & \text{if camera motion} = \text{static} \\ 0 & \text{if camera motion} = \text{tracking} \end{cases} \tag{3.18}$$

3.2.5 Luminance Masking

Luminance masking is the phenomenon that human eyes have less discrimination ability under lighting conditions that are too bright or too dark. The blurriness level

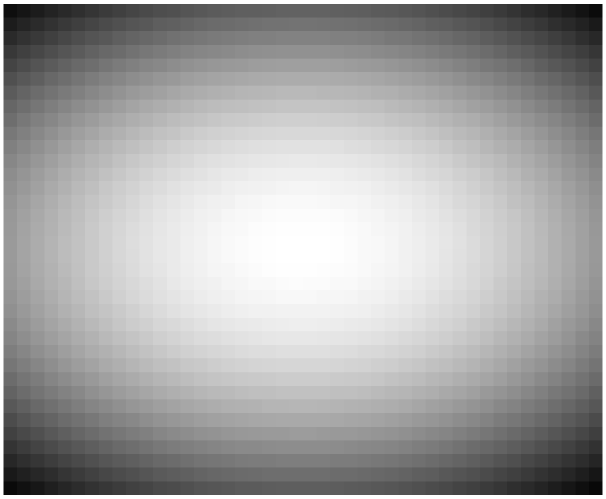


Figure 3.4: Output of a anisotropic Gaussian function. Brighter regions represent stronger visual sensitivity and it decreases as spatial location moves toward boundary.

is less noticeable, or incorrectly judged, under those conditions. Luminance masking is given by

$$\mathcal{L}_n^-(i',j') = \begin{cases} a_1 - c_1[1 - e_1 F_{n,I}(i',j',0,0)]^{k_1} & \text{if } F_{n,I}(i',j',0,0) \geq 127 \\ a_1 - c_2[e_1 F_{n,I}(i',j',0,0)]^{k_2} & \text{Otherwise} \end{cases}$$

where parameters $a_1 = 1.096$, $c_1 = 0.36$, $c_2 = 0.95$, $e_1 = 9.75E-04$, $k_1 = 3$, and $k_2 = 2$ are obtained from [65]. Figure 3.5 shows varying blurriness sensitivity caused by luminance masking. Note that the blurriness sensitivity is highest when the average luminance value in an eight bit scale lies between 100 and 130. Experimental results from [66] indicate that image content can be most correctly distinguished within this lighting range, and the sensitivity in very low light is higher than very bright light.

3.2.6 Camera Motion Activity Estimator

Motion can be roughly separated into local and global types. Local motion refers to the relative motion between objects and the camera. It induces human visual attention as described in Sec. 3.2.4. Global motion is usually generated by camera movement. According to [67], global motion can be further separated into several

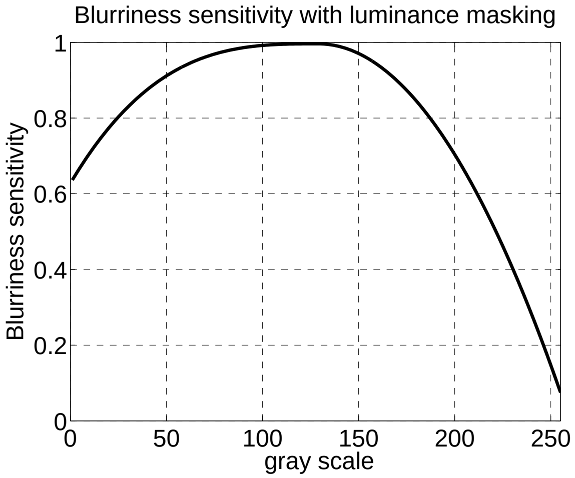
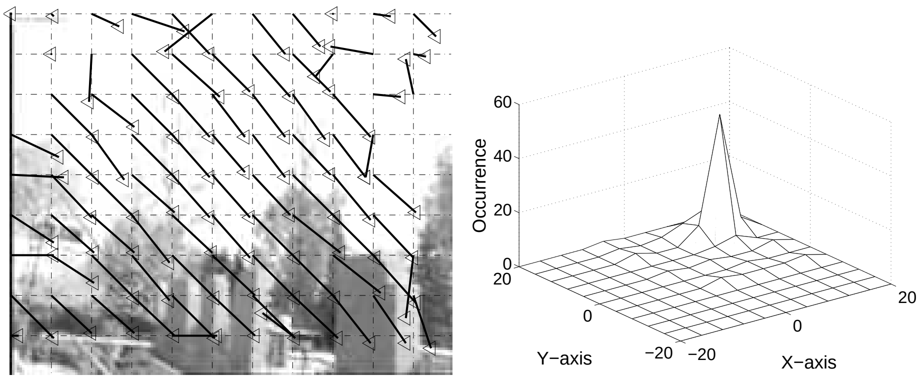


Figure 3.5: Blurriness sensitivity with different average luminance values within eight bit scale

types, such as rotation, shifting, panning, etc. Shifting motion caused by a fast camera aim change from one scene to another functions as a spatial low-pass filtering [68] to eliminate some high spatial frequency signals that are very important to blurriness assessment. This effect will be considered in Section 3.2.8. Before that, the motion vectors that are responsible for shifting motion must be extracted to estimate the motion activity.

Although motion vectors are not always consistent with real motion, they can still be utilized for motion activity estimation since it does not require very accurate motion information. For shifting motion, most of the motion vectors point in a similar direction and that makes the distribution of all motion vectors concentrate in some interval, as shown in Fig. 3.6. Based on this characteristic, a quantity, MA_s, the average of the amplitude of the motion vectors that have the highest coherence in the motion direction of one frame, is defined to represent the strength of camera shifting motion activity. Figure 3.7 shows the procedure of determining the global motion activity. The motion vector information and a predefined histogram step size are used as inputs. After calculating the occurrences in both horizontal and vertical directions, the occurrences in each bin is normalized by the total number of motion vectors and converted into a probability scale. The probabilities are compared against a threshold to filter out the bins with high probability. If no bin has high probability, then $MA_s = 0$. Otherwise, the amplitude of the bin that has highest probability is used as the final MA_s.



(a) (b)

Figure 3.6: Examples of (a) motion vectors with shifting motion and (b) distribution of motion vectors, where x- and y- axes represent the motion in the horizontal and vertical directions.

3.2.7 Spatial and Temporal Post-Processing

In order to ensure that the responses of all attention features are within a consistent range at the final feature composition stage, the output from each attention feature is normalized by

$$X_n(i', j') = \frac{x_n(i', j') - \min[x_n(I', J')]}{\max[x_n(I', J')] - \min[x_n(I', J')]}, \tag{3.19}$$

where $X_n(i', j')$ represents the normalized value of an attention feature which ranges from 0 to 1, $x_n \in (\mathcal{I}_n^+, \mathcal{BG}_n^+, \mathcal{BY}_n^+, \mathcal{T}_n^+, \mathcal{M}_n^+)$, $I' \in (0, \ldots, N_R/8 - 1)$, and $J' \in (0, \ldots, N_C/8 - 1)$.

A two-dimensional median filter first sorts a set of two-dimensional data into one dimension with either decreasing or increasing order vector. Then the middle value of the sorted vector is the output of the median filtered data. This filter is applied to each block to group the attention responses that belong to the same object together:

$$X_{n,S}(i', j') = \text{median}_1[X_n(i'_w, j'_w)], \tag{3.20}$$

where median_1 denotes one iteration of median filtering, and $X_{n,S}$ represents the spatially post-processed data.

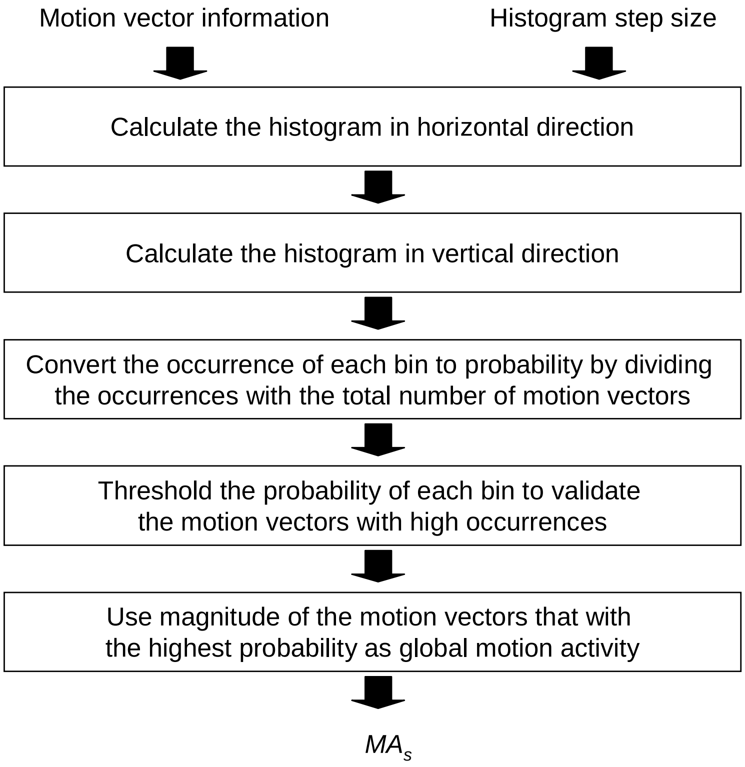

Motion vector information
Histogram step size
Calculate the histogram in horizontal direction
Calculate the histogram in vertical direction
Convert the occurrence of each bin to probability by dividing the occurrences with the total number of motion vectors
Threshold the probability of each bin to validate the motion vectors with high occurrences
Use magnitude of the motion vectors that with the highest probability as global motion activity
MA_s

Figure 3.7: Procedure of global motion activity determination.

Since the content of the current video frame is expected to be similar to its neighboring frames, high attention response contrast of a block in the current frame to the blocks at the same spatial location but in neighboring frames will result in a sudden attention value change. A one iteration temporal median filter is applied to each block along the temporal axis as

$$X_{n,ST}(i',j') = \text{median}_1[X_{n_w,S}(i',j')] \tag{3.21}$$

to moderate this effect, where $X_{n,ST}$ represents the spatially and temporally post-processed normalized attention value, $n_w \in w_t$ and w_t represents a group of frames centered at the nth frame.

3.2.8 Motion Masking Function

It is generally believed that two temporal mechanisms exist in human vision, one is transient and the other is sustained [69]. The transient channel usually happens as high motion camera shifting occurs and it introduces strong motion blur [17]. In this case, the spatial content is low-pass filtered, and most of the spatial detail is lost. On the other hand, the sustain channel generally refers to static camera motion and no motion blurriness occurs. As a result, most visual detail information is carried in the sustained channel. In order to imitate this human visual effect, a binary motion masking mechanism is implemented as

$$\beta^{-}(n) = \begin{cases} 0 & \text{if } MA_{s,T}(n) \geq T_s \\ 1 & \text{Otherwise} \end{cases}, \tag{3.22}$$

where $MA_{s,T}(n)$ is the temporally post-processed shifting motion activity of nth frame, and T_s is the threshold for $MA_{s,T}(n)$ to decide the visibility of high frequency content, where T_s is determined experimentally and it is set to 8 for QCIF(176×144) and 11 for QVGA(320×240) image size. Using (3.22), blurriness scores from frames with high shifting motion are disregarded due to motion masking.

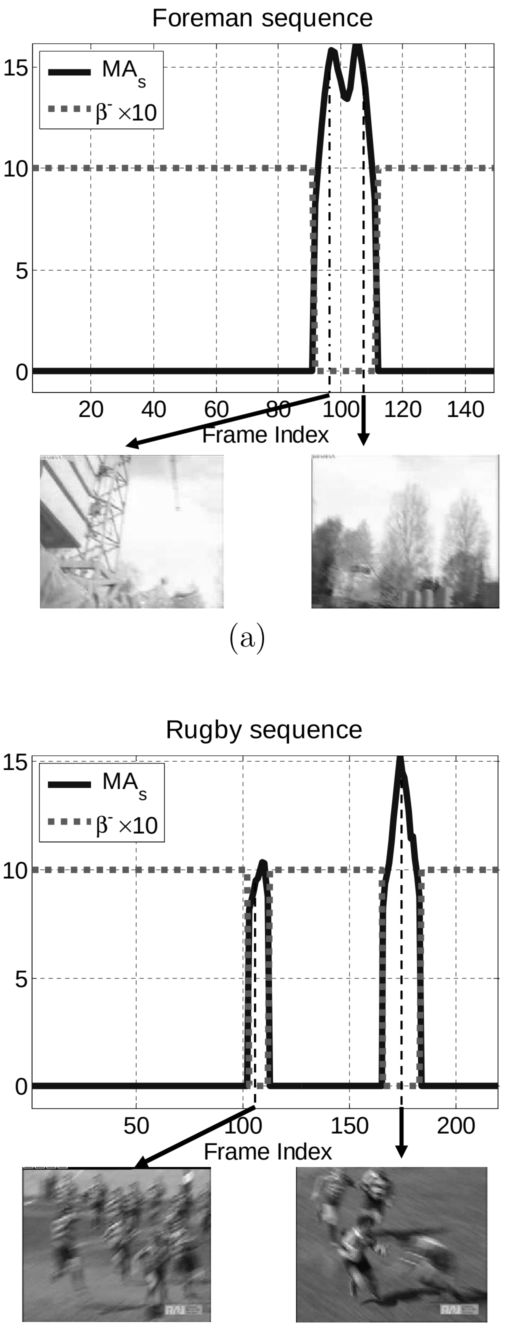
Figure 3.8 shows examples of $MA_{s,T}$ vs. β^{-} and corresponding snapshots. Note that β^{-} is multiplied by 10 for illustration purposes. In Fig. 3.8, details in the frames with high $MA_{s,T}$ are masked by motion and the texture content is barely perceptable. The corresponding β^{-} of those frames decreases to 0, meaning the blurriness caused by compression of those frames is not visible and they are not taken into account in sequence level blurriness estimation.

3.2.9 Combiner

The final VBSM is derived by a linear combination of all attention and masking features and is given by

$$\begin{aligned} M_n(i',j') &= \frac{\beta^{-}(n)\mathcal{L}^{-}_{n,ST}(i',j')}{5} \cdot [\mathcal{I}^{+}_{n,ST}(i',j') + \mathcal{RG}^{+}_{n,ST}(i',j') \\ &+ \mathcal{BY}^{+}_{n,ST}(i',j') + \mathcal{T}^{+}_{n,ST}(i',j') + \mathcal{M}^{+}_{n,ST}(i',j')], \end{aligned} \tag{3.23}$$

where $\mathcal{L}$ functions as a suppression parameter to adjust the attention value of each block, and β^{-} binarily determines the blurriness visibility of each frame.



(a)

(b)

Figure 3.8: Example of camera shifting motion activity vs. motion masking for (a)FOREMAN and (b) RUGBY sequences at frame rate 15fps.

3.3 VBSM performance evaluation

Figure 3.9 shows several image samples, their corresponding VBSMs are shown in Fig. 3.10. Fig. 3.9(a-c, g, h) are QCIF size and Fig. 3.9(d-f) are QVGA size. Figure 3.9(a) represents content of a moving bus tracked by a camera, Figure 3.10(a) shows that the bus has been allocated the highest blurriness sensitivity of all regions. Data from Fig. 3.9(b, c) depict sports type content with high object movement. The corresponding VBSMs in Fig. 3.10(b, c) show the highest blurriness attention aggregates on the players. Fig. 3.9(d) comes from a panning camera that moves along with a slow moving boat, with high texture background and foreground. Results from Fig. 3.10(d) show that not only the boat but also the adjacent rocks have been assigned high blurriness sensitivity because the former is the main attention object and the latter belong to high structural texture regions. Figure 3.9(e, f) use a static camera with slow moving talking-head and object, respectively. In Fig. 3.10(e, f), the main interesting objects have high blurriness sensitivity but the flat regions that have the least blurriness sensitivity are assigned very low blurriness attention. Figures 3.9(g, h) are from the same sequences as Fig. 3.9(b, c) but with high shifting motion. As shown in Fig. 3.10(g, h), both Fig. 3.9(g, h) receive zero blurriness attention since they are heavily masked by motion.

Furthermore, a deeper evaluation of the performance of VBSM is carried out by injecting blurriness into image *with* and *without* VBSM. On the average, two cases are designed to reduce the same percentage high frequency energy, but case 1 assigns a varying percentage of high frequency loss to each block according to VBSM, and case 2 truncates high frequency energy of all blocks using a uniform percentage. Thus, the magnitude of all frequency bands of a blur-injected block are adjusted by

$$F'(i',j',u,v) = F(i',j',u,v) \cdot \text{Mask}_{8\times 8}(u,v) \tag{3.24}$$

where $\text{Mask}(u_1,v_1) = 1$ and $\text{Mask}(u_2,v_2) = 0.8 \times th$, where $u_1,\ v_1 \in (1,2,3)$, and $u_2,\ v_2 \in (4,\cdots,8)$. This indicates that only the high frequency band energies are adjusted by the value of $0.8 \times th$, and th is the key factor that controls the way that energy of high frequency bands will be reduced. It is given by

$$th = \begin{cases} M_n(i', j') & \text{for case 1} \\ \frac{\sum_{i'} \sum_{j'} M_n(i',j')}{(N_R/8-1)\cdot(N_C/8-1)} & \text{for case 2.} \end{cases}$$

As a result, both cases reduce high frequency energy by the same percentage but with different energy loss allocation. Finally, the image that has high frequency energies truncated is obtained by inverse DCT as $f' = \text{IDCT}(F')$, where IDCT is the inverse DCT. In case 1, with guidance from VBSM, blurriness sensitive blocks have less high frequency energy reduction, while case 2 eliminates the same percentage of high frequency energy for all blocks regardless of their importance to visual perception. Blur-injected frames of CONTAINER are shown in Fig. 3.11. In case 1, the salient objects (i.e. boat and pole) have a sharper appearance than case 2. This is evidence that the VBSM can accurately determine the regions that are significant to visual blurriness perception.

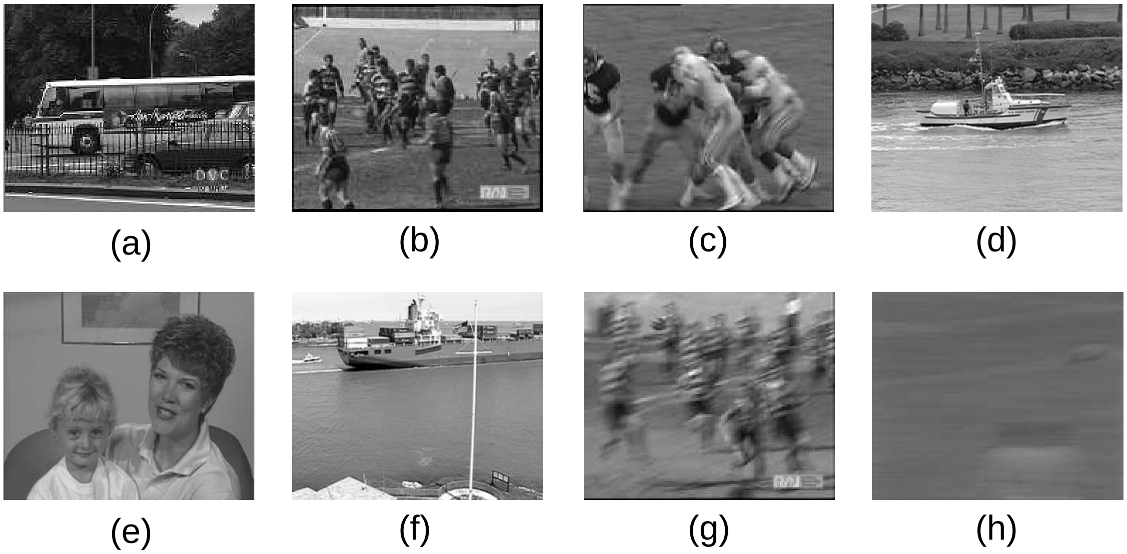


Figure 3.9: Sample image data of (a)BUS, (b)RUGBY, (c)FOOTBALL, (d)COASTGUARD, (e)MOTHER DAUGHTER, (f)CONTAINER, (g)RUGBY with high shifting motion, and (h)FOOTBALL with high shifting motion

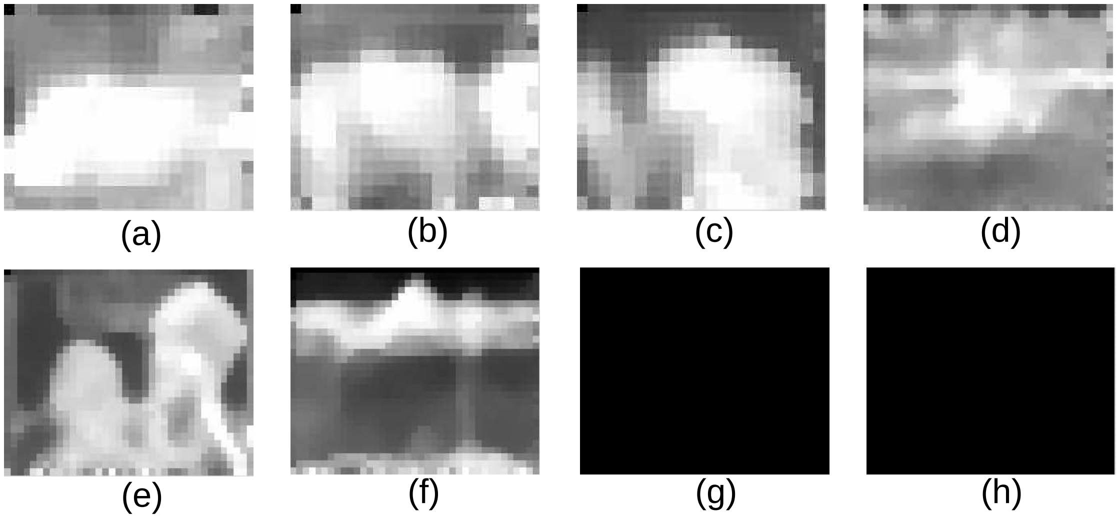
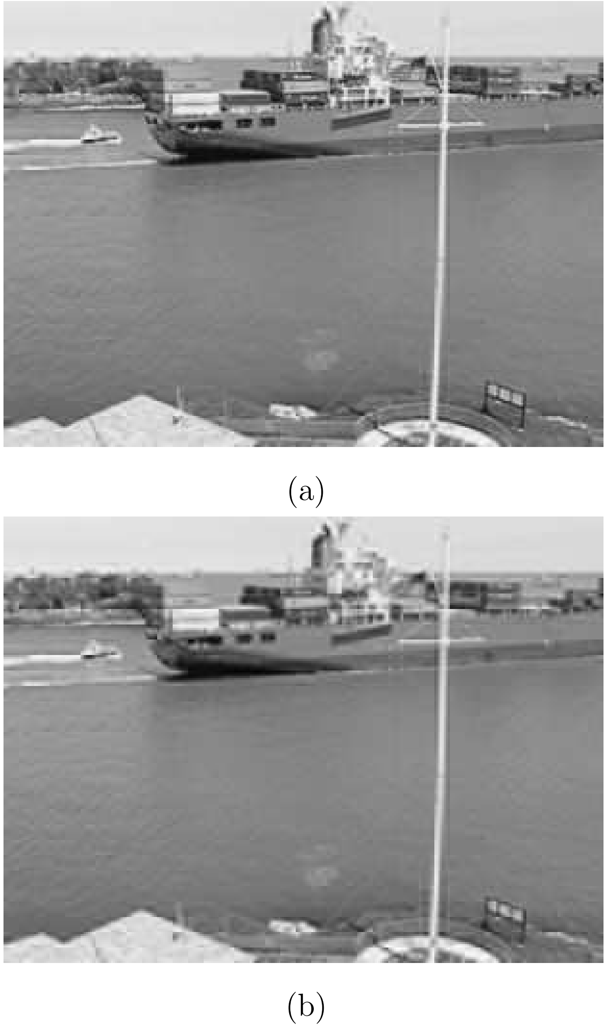


Figure 3.10: VBSMs of (a)BUS, (b)RUGBY, (c)FOOTBALL, (d) COASTGUARD, (e)MOTHER DAUGHTER, (f)CONTAINER, (g)RUGBY with high shifting motion, and (h)FOOTBALL with high shifting motion

3.4 Summary

A new framework for determining human visual sensitivity to blurring artifacts has been proposed. It considers both human visual attention and several human visual effects. Not only the positive stimuli (i.e. visual attention) but also the negative stimuli (i.e. visual masking effect) have been taken into account. Because of these characteristics, the output visual sensitivity maps are very suitable for video quality assessment or enhancement related applications. Simulation results show that the estimated visual sensitivity maps highly correspond to human visual determination. In addition, a blur-injection experiment has been carried out to evaluate the performance of VBSM by removing the same percentage of information from two images with the same contents, but one loses information with guidance from the VBSM, and the another one in a uniform fashion. Subjective evaluation shows that the former has better visual quality than the latter. It confirms that the VBSM can accurately determine the visual sensitivities in a video sequence.

In the next Chapter, the VBSM will be applied to video quality assessment; specifically to blurriness estimation. In future, possibility of more applications will be explored, such as bits allocation, and video quality enhancement. For bits allocation, the compression ratio of each local compression unit (i.e. macroblock) can be ad-



(a)

(b)

Figure 3.11: Blur-injected samples of CONTAINER for (a) VBSM directed, and (b) uniformly spread.

justed according to VBSM, with more bits assigned to the regions with higher visual sensitivity. For quality enhancement, the VBSM can be used to direct the quality enhancement algorithm (e.g. sharpening) to improve the quality of regions that are important to visual quality and skip the parts with low visual significance. In this way, the computation complexity can be decreased dramatically but the video quality is still effectively enhanced.

PUBLICATIONS

Kai-Chieh Yang, Clark C. Guest and Pankaj K. Das, “Human Visual Attention for

Compressed Video", *Proc. IEEE International Symposium on Multimedia*, pp. 525-532, Dec. 2006

Kai-Chieh Yang, Clark C. Guest and Pankaj K. Das, "Hierarchy Visual Attention Map", *Proc. SPIE, Human Vision and Electronic Imaging XII*, Vol. 6492, Feb. 2006

4

Perceptual Blurriness Metric (PBM)

4.1 Introduction

Blurriness, one of the most pronounced artifacts in video quality assessment, dominates the first impression of compressed video or image signals. In this Chapter, a new blurriness metric that includes human cognitive and visual effects will be presented. It does not require access to original video sequence. This metric estimates the blurriness level of a compressed video sequence based on the presence of high frequency signal components. A numerical visual blurriness sensitivity map (VBSM) as described in Chapter 3 is adopted to assign a weight to each spatial and temporal location according to human visual attention and human visual system factors (i.e. motion, luminance masking). The local blurriness measure is adjusted based on the VBSM and forms a sequence of blurring score. In order to ensure the blurriness assessment is independent of content, the blurring score is normalized by Quantization Parameter information from the bitstream. Finally, a cognitive module is applied to emulate human perceptual non-linearity and saturation effects, and produce a final blurring value. Detailed performance analysis of each sub-module is reported to demonstrate the importance of each. Moreover, two performance tests:(1) objective and (2) psychovisual experiments are carried out. Experimental results confirm high accuracy of blurriness prediction for the proposed metric.

Quantization is the core aspect of data size reduction used in video compression. The basic principle of quantization is to represent the video data with a finite number of levels. This results in more compact data after entropy coding. The precision of quantized data is determined by the quantization step size and is often denoted by the *Quantization Parameter*(QP), where larger QP means larger quantization step size and smaller compressed data size. However, the consequence of large QP is greater information loss and impaired perceived quality for viewers. As described in Chapter 2, video quality distortion can be separated into sub-, near-, and supra-threshold categories according to its perceptibility to human vision [51]. The sub- and near-threshold classes refer to types of distortion that are below or slightly above just-noticeable-difference (JND) respectively. Supra-threshold distortion generally appears in a structured form and is known as an *artifact*. This type of distortion is very irritating to human perception and dominates subjective quality assessment. Many researchers have attempted to improve video quality assessment accuracy by determining an appropriate JND [21, 65]. However, research in quantifying supra-threshold distortion is relatively sparse. One of the main challenges to supra-threshold distortion measurement is that its appearance varies with video content. Hence, human perceived annoyance can be different even though two video segments have the same error energy [70]. Many high level cognitive functions in the human brain and visual system are involved in artifacts perception, which complicates the measurement process.

Blockiness, ringing, and blurriness are the major artifacts caused by video compression [71]. Blockiness refers to an abnormal tiling structure in compressed images. It results from the independent quantization of blocks in the block based Discrete Cosine Transform (DCT) coding schemes, leading to discontinuities at the boundaries of adjacent blocks. Ringing is fundamentally associated with Gibbs phenomenon [71] and is most evident along high-contrast edges in smooth areas. Blurriness manifests as a loss of spatial detail and a reduction of edge acuteness. It is due to the suppression of high-frequency coefficients by coarse quantization.

Video data compressed by MPEG-1, MPEG-2, and MPEG-4 very often suffer from these three major artifacts. However, some emerging codecs, such as H.263 and H.264, have included in-loop post-processing as a part of the standard. The post-processing is like a low-pass filtering mechanism to remove possible spurious high frequency signal components in reconstructed video data. Consequently, blocking and ringing artifacts are effectively removed. But this post-processing often accidently eliminates some high frequency content that belongs to the original video data. Thus, it adds additional blurriness to the blurriness introduced by quantization. Therefore, blurriness has recently become the most pronounced artifact. Enhancing this aspect of video quality has drawn much research interest. A good blurriness metric can provide precise information about the strength and location of blurring artifact. With a proposed burriness metric as guidance, video coder designer can reduce the blurriness by adaptively allocating resources or intelligently post-processing video data. Thus, accurate blurriness assessment is crucial to video quality enhancement and a reliable blurriness metric is highly desired.

The flow of this chapter is arranged as follows. Section 4.2 provides an overview of related research for blurriness assessment. Based on the analysis of the related work, several weaknesses of current blurriness estimation approaches are discussed in Section 4.3. The proposed metric is introduced in detail in Section 4.4. Section 4.5 explains the experiment set up for parameter fitting and metric performance evaluation. The parameter fitting process for several important functions is described in Section 4.6. The experimental results are presented and analyzed in Section 4.7. Finally, the summaries of this work are given in Section 4.8.

4.2 Related Work

An organization of blurriness measurement research is shown in Fig. 4.1. First, edge pixels are extracted by observing the local peak in luminance gradient value. The edge pixels are those with a local maximum in first derivative of pixel values. In the second step, pixel value activities around each edge pixel are used to estimate the strength of blurring artifacts. The estimation process can be carried out in either the spatial or frequency domain. The former uses video pixel values directly and the latter uses the transformed spatial frequency data.

Using a spatial approach, Ref. [41,72,73] propose several metrics that measure the blurriness by estimating spatial activity around edges. These algorithms heavily rely

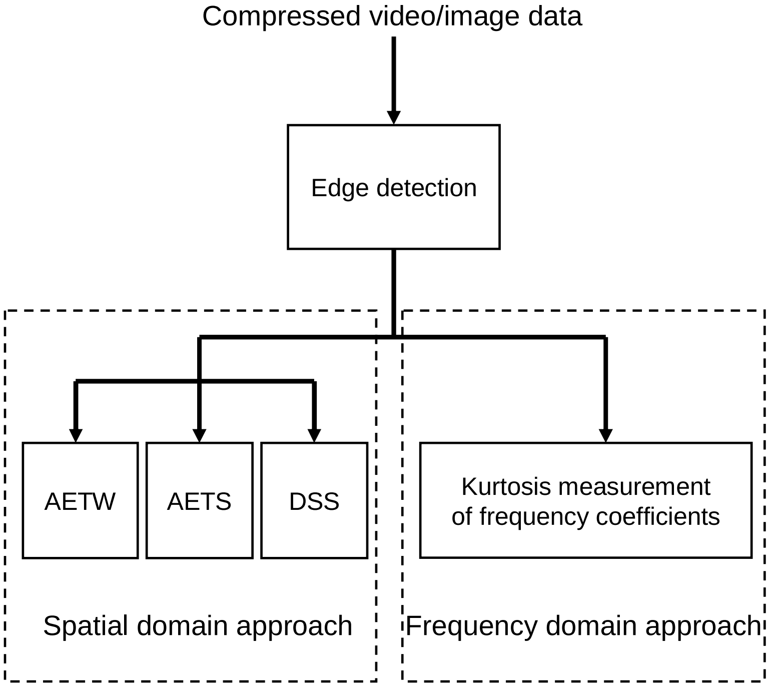

Compressed video/image data
Edge detection
AETW
AETS
DSS
Kurtosis measurement of frequency coefficients
Spatial domain approach
Frequency domain approach

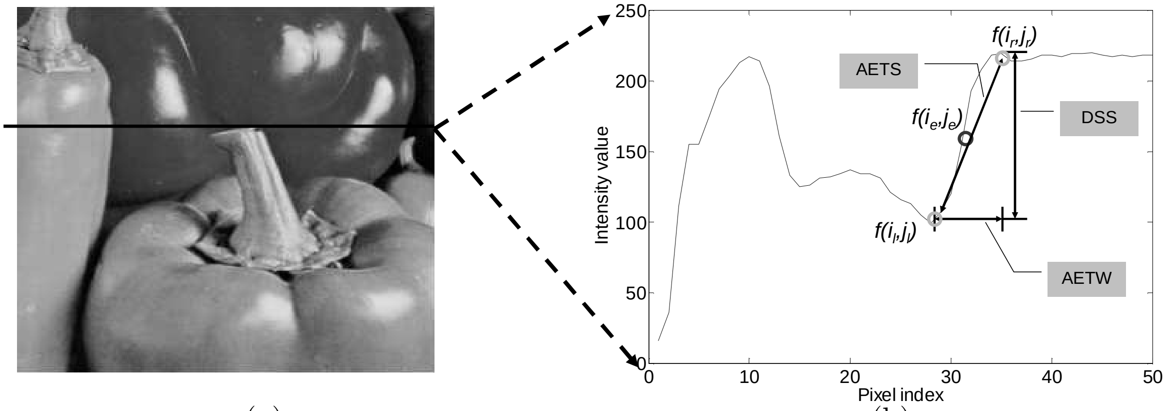
Figure 4.1: Summary of related blurriness assessment research.

on accurate edge detection and are designed for still image applications. Figure 4.2 shows an example of blurriness estimation using edge profiles. Figure 4.2(a) is a sample image and Fig. 4.2(b) shows a row of pixels of the image, where (i_e, j_e) depicts the spatial location of an edge pixel, (i_l, j_l) and (i_r, j_r) represent respectively the closest left and right local minimum or maximum second order intensity derivative along the horizontal or vertical direction of (i_e, j_e). Consider $f(i_e, j_e)$, $f(i_l, j_l)$, $f(i_r, j_r)$ as the corresponding intensity values, the spatial edge profile based blurriness estimation methods can be summarized into the following three categories:

1 Average Edge Transition Width (AETW): Measure the strength of blurring artifact by observing the average width of all edge pixels as

$$AETW = \frac{1}{N_e} \sum_{i_e, j_e} |(i_l, j_l) - (i_e, j_e)| + |(i_r, j_r) - (i_e, j_e)|, \tag{4.1}$$

where $|(i_l, j_l) - (i_e, j_e)|$ and $|(i_r, j_r) - (i_e, j_e)|$ represent the edge widthes on both sides of an edge pixel on (i_e, j_e), N_e is the total number of all edge pixels. Larger AETW means that edges are wider and image is more blurred.



(a) (b)

Figure 4.2: Example of edge profile.

2 Digital Sharpness Scale (DSS) : Using the height of edges to estimate the blurriness as

$$DSS = \frac{1}{N_e} \sum_{i_e, j_e} |f(i_l, j_l) - f(i_e, j_e)| + |f(i_r, j_r) - f(i_e, j_e)|. \tag{4.2}$$

Higher DSS means that edges are more acute and image is less blur.

3 Average Edge Transition Slope (AETS) : Estimate the blurriness by observing the slope of edges as

$$AETS = \frac{1}{N_e} \sum_{i_e, j_e} \frac{DSS(i_e, j_e)}{AETW(i_e, j_e)}. \tag{4.3}$$

Sharp images contain edges with steeper slope, and hence, AETS value is higher.

Using a frequency domain approach, Caviedes [40] extends Zhang's work [48] to design a sharpness metric, the opposite to blurriness, based on measuring the departure of a probability distribution from a Gaussian(normal) shape of local DCT coefficients around each edge pixel by a statistical measure: *Kurtosis*. The Kurtosis measurement, β_2, for a variable, $\mathcal{X}$, is given by

$$\beta_2 = \frac{m_4}{m_2^2} = \frac{E[(\mathcal{X} - E(\mathcal{X}))^4]}{E^2[(\mathcal{X} - E(\mathcal{X}))^2]}, \tag{4.4}$$

where m_4 and m_2 are the 4th and 2nd moment of data $\mathcal{X}$, respectively, and $E(\mathcal{X})$ is the mean value. The Kurtosis of a Gaussian distribution is 3. The value of β_2 for a random

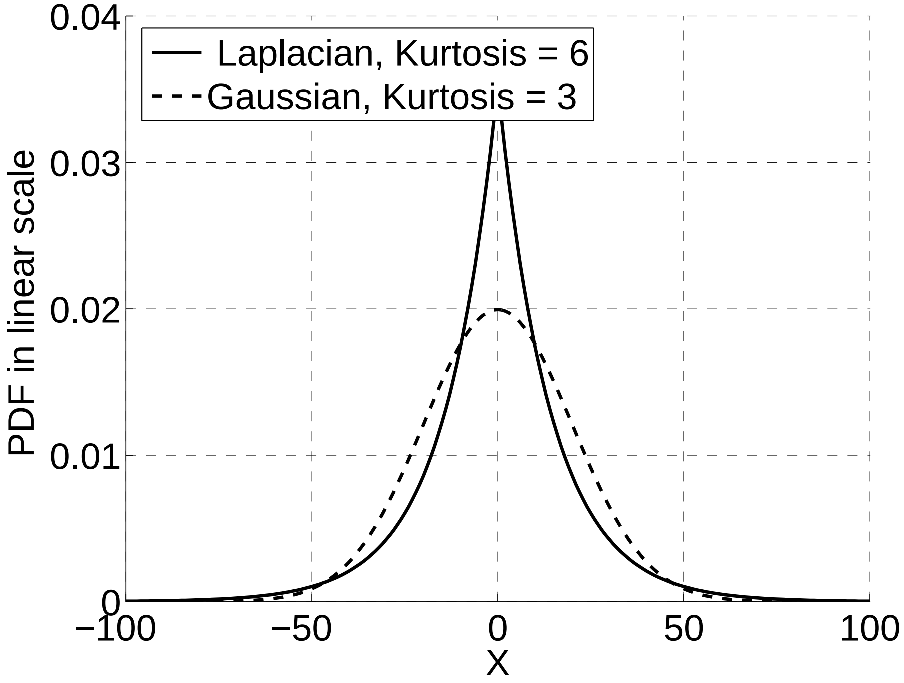


Figure 4.3: Example of different distributions and their Kurtosis value.

variable can be compared with 3 to determine whether its distribution is ”peaked” or ”flatted-topped” relative to a Gaussian distribution. Figure 4.3 shows probability density functions for Gaussian and Laplacian distributions. For the distribution that is more like the Laplacian, the β_2 will be higher than 3. On the contrary, for the distribution that is similar to Gaussian or flatter, the Kurtosis measurement will be close to 3 or less. Consider each DCT coefficient except the DC value as one histogram bin. After normalization, each DCT coefficient can be thought as a probability value that ranges from 0 to 1. For a sharp image, high frequency bands will contain more energy and the distribution of DCT coefficients will be closer to Gaussian. In this case, the Kurtosis measurement will be lower. For a blurred image, less energy exists in high frequency bands and the distribution of DCT coefficients will be closer to a Laplacian distribution, and hence, the Kurtosis value will be higher. Therefore, it can be concluded that β_2 will increase as the image suffers more blurriness. A more complete procedure of blurriness estimation using Kurtosis measurement is shown in Fig. 4.4. First, the locations of edge pixels are detected by calculating pixel

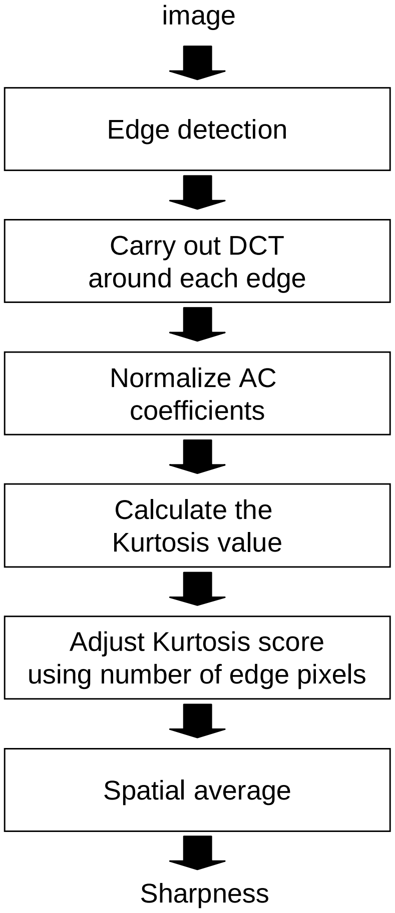


image
Edge detection
Carry out DCT around each edge
Normalize AC coefficients
Calculate the Kurtosis value
Adjust Kurtosis score using number of edge pixels
Spatial average
Sharpness

Figure 4.4: Procedure of sharpness estimation using Kurtosis measurement.

gradient values in both horizontal and vertical directions. An 8-by-8 block is grouped around each edge pixel and the DCT is carried out on each block. AC coefficients are normalized by the sum of all AC coefficients. The Kurtosis value is calculated using Equation (4.4) and it is adjusted by the number of edge pixels in each block. The sharpness score of a frame is the average of Kurtosis value of all edge pixels.

Marichal and *et al.* [1] designed another blurriness estimation method in the frequency domain by observing the distribution of the ratio of AC to DC coefficients. This metric is denoted as the *DCT-histogram* in the following presentation. The blurriness estimation procedure for this metric is shown in Fig. 4.5. First, the compressed image is transformed by the non-overlapped block-based DCT. Then the occurrence of non-zero values for each DCT coefficient of all blocks is calculated. The histogram

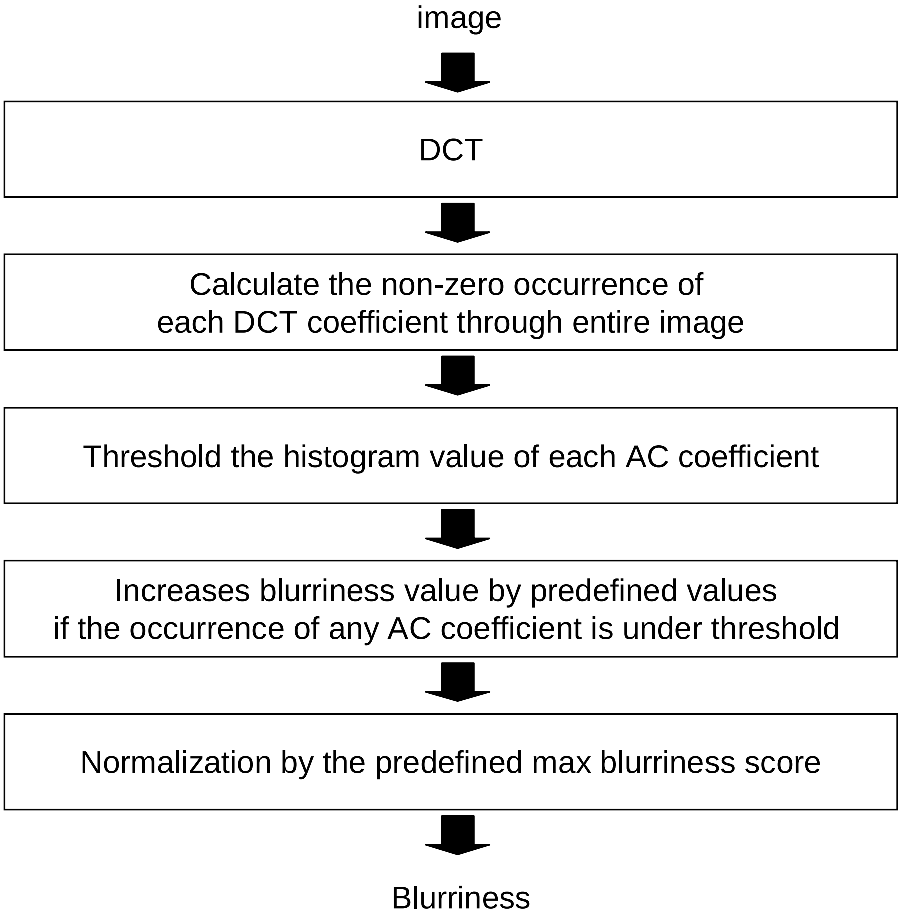


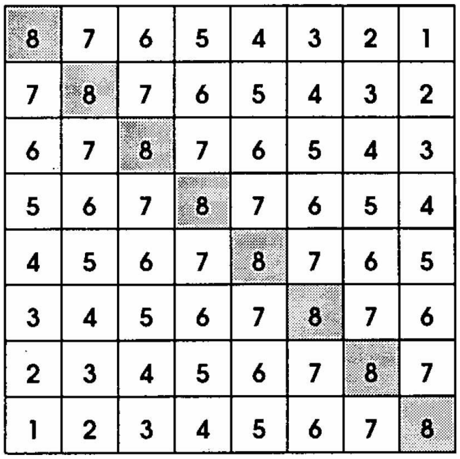
image
DCT
Calculate the non-zero occurrence of each DCT coefficient through entire image
Threshold the histogram value of each AC coefficient
Increases blurriness value by predefined values if the occurrence of any AC coefficient is under threshold
Normalization by the predefined max blurriness score
Blurriness

Figure 4.5: Procedure of blurriness estimation using the DCT-histogram approach.

for each AC coefficient is compared against a threshold that adapts to the histogram value of DC coefficient. If any AC coefficient that is less than the threshold, then blurriness increases according to the location of the coefficient by a predefined value as defined in Fig. 4.6. Finally, the ratio of the sum of the blurriness values produced by all AC coefficients to the sum of the values in Fig. 4.6 is the blurriness score of one block. The final blurriness score for entire image is average blurriness score of all blocks.

4.3 Problem Formulation

Most related work relies on edge information to determine the regions that are significant for blurriness assessment. However, edge detection can fail because of the



8	7	6	5	4	3	2	1
7	8	7	6	5	4	3	2
6	7	8	7	6	5	4	3
5	6	7	8	7	6	5	4
4	5	6	7	8	7	6	5
3	4	5	6	7	8	7	6
2	3	4	5	6	7	8	7
1	2	3	4	5	6	7	8

Figure 4.6: Predefined blurriness values for each AC coefficient [1].

presence of other compression artifacts and the absence of high frequency information due to heavy quantization. Accurate edge location might not be available and the accuracy of blurriness estimation is reduced.

Also, most previous work focuses on application to individual images and treats a video sequence as a set of independent images. These approaches assume that human eyes have the same sensitivity to blurriness regardless of varying spatial and temporal stimuli. However, the fact is that viewers pay the most attention to blurriness only in certain regions, and some blurring artifacts might not be perceived because of the presence of other signals. Therefore, when estimating blurriness, the sensitivity to local blurriness should vary with the level of human visual attention.

All the referenced blurriness metrics assume that the measured signal can be mapped to human perceived blurriness linearly. However, because of high-level human cognitive mechanisms, the visual blurriness assessment process is highly nonlinear and saturates as blurriness is either imperceivable or extremely severe.

Normalization is a challenging and unresolved issue in most blurriness estimation research. Because human perceived blurriness is highly influenced by content, it results in different annoyance level even with the same strength of erroneous signal. This causes objective measures to deviate from human perceived blurriness. Previous research does not have a good solution for this problem. Because of this weakness, the absolute blurriness value can not be interpreted directly.

Addressing the concerns above, a novel and effective blurring metric - *Perceptual Blurriness Metric(PBM)* is designed for compressed video data. Several notable characteristics of this metric are:

1. Non-Reference estimation: Video/image quality metrics can be classified into Full-Reference(FR), Reduced-Reference(RR), and No-Reference(NR) according to the accessability of original data [21,22]. The proposed PBM is a NR metric, which allows more flexibility for future deployment.

2. High blurriness estimation accuracy on both MPEG-4 and H.264/AVC: Our investigation shows that many blurriness metrics can handle the MPEG-4 encoded sequences but fail for H.264/AVC. Experimental results show that PBM not only works well with MPEG-4 but also dramatically outperforms all other metrics on the sequences encoded by H.264/AVC.

3. More robust estimation basis: The most important cause of blurriness is high frequency signal loss, so we use the energy of high frequency bands as the blurriness estimation basis. Also, different weights are applied to each band since each frequency band contributes a different level of impact for blurring artifact. Results show that this measure is more robust and generic than edge profile based schemes.

4. New visual module for supra-threshold video quality assessment: A novel and extendable human visual module introduced in Chapter 3 is adopted to emulate varying artifact perceptibility in the spatial and temporal domains by considering both human visual attention and masking phenomena. This is the first work that provides enough flexibility for adding other visual masking or attention features based on the characteristics of different types of artifact. Also, the proposed visual module works in the frequency domain, so it can be easily implemented with coder design for more applications (such as bits allocation and quality enhancement).

5. Content independent: The output of PBM remains within a fixed range regardless of the video content. This makes the calculated absolute blurriness score more interpretable.

6. Cognitive saturation effects are taken into account: Perceptual saturation and non-linearity are emulated to enable high accuracy of blurriness estimation.

4.4 The Proposed Metric

Figure 4.7 depicts the system diagram of PBM. Assuming the input is a video bitstream of a compressed video sequence, it is first decoded to raw data. The reconstructed video data of nth frame, f_n, is transformed into the frequency domain and decomposed into several sub-bands, denoted as F_n. Subsequently, all sub-bands are weighted according to their importance to blurriness estimation given by *Blurriness Sensitivity Vector*(BSV) and the weighted high frequency energy-HF_n is calculated. Meanwhile, the F_n and motion vector information from bitstream are given as input into *Visual Blurriness Sensitivity Map*(VBSM) to form a set of significance values, M_n, to all blocks. Afterward, the HF_n is combined with M_n and form a frame level blurriness score - $b_{n,0}$. The normalization module takes the QP information from bitstream and $b_{n,0}$ to estimate the texture level of video content, tex_n, and normalize the $b_{n,0}$ to b_n according to different texture level. Finally, both b_n and $text_n$ are input into a cognitive module to emulate visual saturation effect and obtain the final blurriness score for the entire sequence - B.

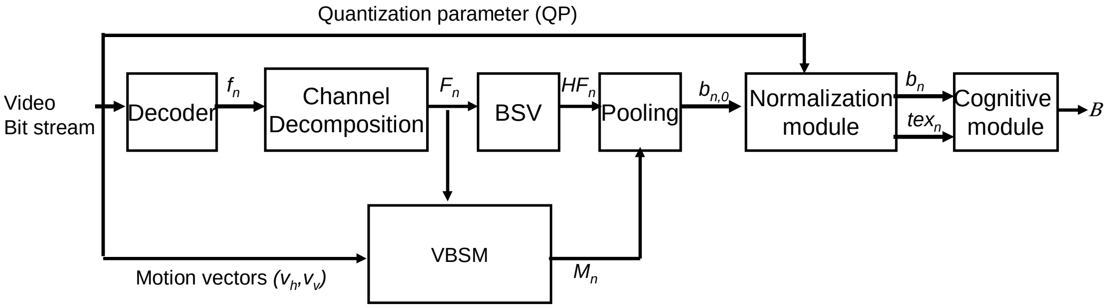


Figure 4.7: System diagram of PBM.

4.4.1 Channel Decomposition

The purpose of channel decomposition is to transform video data from pixel domain to frequency domain, the data with similar frequency are grouped into different sub-bands. The most well known block based transform, DCT, with block size 8 × 8 is used.

Since DCT is a widely used transform in video and image codecs, the DCT coefficients can be extracted directly from the bitstream. For the intra-coded frames, which are compressed using one of image standards (i.e. JPEG and JPEG-2000), the DCT coefficients can be obtained directly. Also, the DCT coefficients for inter-coded frames, which are compressed by motion compensation, can be reconstructed by inverse motion compensation with the corresponding motion prediction residual. The extraction can be easily carried out using a MPEG-4 codec. However, this is not entirely applicable in a H.264/AVC bitstream, because: 1) The DCT transform block size is no longer fixed at 8 × 8; it can be 4 × 4 as well. In the case of 4 × 4 block size, up-sampling DCT coefficients from 4 × 4 to 8 × 8 will introduce additional uncertainty, and 2) The intra-prediction has too many different modes (i.e. eight for 4 × 4, and four for 16 × 16 block size), and it is carried out in the pixel domain. These effects may degrade the channel decomposition accuracy of inverse motion compensation method and increase implementation complexity. Therefore, we still decode the bitstream fully and process a non-overlapped block-based DCT to separate the nth frame, f_n, into 64 sub-bands in absolute value using

$$F_n(i', j', u, v) = |\sum_{i=0}^{N_B-1} \sum_{j=0}^{N_B-1} f_n(i', j', i, j) k(i, j, u, v)|, \tag{4.5}$$

$$k(i, j, u, v) = \alpha(u) \cdot \alpha(v) \cos[\frac{(2i+1)u\pi}{2N_B}] \cos[\frac{(2j+1)v\pi}{2N_B}], \tag{4.6}$$

where

$$\alpha = \begin{cases} \sqrt{\frac{1}{N_B}}, & \text{for u} = 0 \\ \sqrt{\frac{2}{N_B}}, & \text{for u} = 1,\ 2 \ldots N_B - 1, \end{cases} \tag{4.7}$$

u, v represent the DCT coefficient indices within a block, and N_B is the dimension of each DCT block, which is set to 8 here.

4.4.2 Blurriness Sensitivity Vector

Only certain frequency bands are useful for blurriness estimation. Low frequency bands elicit too little sensitivity to blurriness, while the highest frequency bands are too sensitive to some high frequency compression artifacts (i.e., ringing). Therefore, each channel is weighted accordingly to emphasize bands that are important to blurriness estimation and suppress the others. Significant research effort has been spent on a similar topic: the *Contrast Sensitivity Function(CSF)* [74, 75]. However, it is mainly designed for determining sensitivity to near-threshold distortion (i.e. noise); its philosophy is not applicable to blurriness estimation. Therefore, a *Blur Sensitivity Vector (BSV)*, Ψ, has been designed as

$$\Psi_{1\times 8} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0.13 & 0.04 & 0 & 0.04 \end{bmatrix}, \tag{4.8}$$

to represent the importance of each band to blurriness estimation. Each element of Ψ corresponds to one band, and a higher value means this band is more sensitive to blurring artifacts. Values from (4.8) shows that the 4th band has the greatest effect on blurriness assessment. The values in Ψ are determined from data of a subjective experiment. Details of parameter determination process will be explained in Section 4.6.

The first column and row in a block of DCT coefficients represent frequencies associated with horizontal and vertical edges respectively. Human eyes have higher blurriness sensitivity to the edges in these two orientations. Also, luminance data contains the richest information for quality assessment. Thus, we emphasize the DCT coefficients for these two directions in luminance data, and estimate the high frequency energy of (i', j')th block, $HF_n(i', j')$, as

$$HF_n(i', j') = \frac{1}{2F_{n,I}(i', j', 0, 0)} \cdot \Psi \times \hat{F_{n,I}}(i', j') \tag{4.9}$$

where $F_{n,I}$ denotes the DCT coefficients of the luminance data, $F_{n,I}(i', j', 0, 0)$ is the corresponding DC value, and $\hat{F_{n,I}}(i', j')$ is given by

$$\hat{F_{n,I}}(i', j') = \begin{bmatrix} 2F_{n,I}(i', j', 0, 0) \\ F_{n,I}(i', j', 0, 1) + F_{n,I}(i', j', 1, 0) \\ \vdots \\ F_{n,I}(i', j', 0, N_B - 1) + F_{n,I}(i', j', N_B - 1, 0) \end{bmatrix}.$$

4.4.3 Visual Blurriness Sensitivity Map

When evaluating the blurriness of a video, observers tend to allocate different attention to various spatial and temporal locations. This tendency is driven by several bottom-up (i.e., luminance contrast) and top-down (i.e. skin tone) stimuli. In addition, the blurriness at some spatial/temporal locations is not recognized because of *masking* phenomenon. Less blurring sensitivity should be assigned to masked regions. Hence, human visual attention and masking factors lead to different blurriness sensitivities on various spatial and temporal locations. This phenomena is numerically emulated using a *Visual Blurriness Sensitivity Map (VBSM)* as introduced in Chapter 3. The VBSM is composed of two major parts: 1)the *Human Visual Attention Map (HVAM)* from work in [60], and 2)luminance and motion masking components. The final VBSM of nth frame is given by Equation (3.23).

4.4.4 Pooling

The preliminary blurring score, $b_{n,0}$, is the summation of product of blurriness sensitivity of each block given by Equation (3.23) and the amount of high frequency energy calculated from Equation (4.9) as

$$b_{n,0} = \sum_{i'=0}^{N_R/8-1} \sum_{j'=0}^{N_C/8-1} HF_n(i', j') \cdot M_n(i'.j'). \tag{4.10}$$

4.4.5 Normalization Module

A great challenge to blurriness assessment is that the range of blurring scores varies with content, which is mainly attributed by different texture levels in the original video. In the scenario that original video data is not accessible, selecting an appropriate normalization threshold is extremely difficult. Nevertheless, since quantization is the main cause of blurriness, the QP information is helpful to approximate the original texture level.

However, a question may arise - ”Since we know blurriness increases along with QP, why not simply exploit QP to measure blurriness instead of high frequency band magnitude?” This makes sense at first, however, the blurriness caused by compression is not only introduced by quantization, but also by pre- and post-processing (i.e. de-blocking filtering) and long distance motion compensation. For example, the deblocking filter low-passes the reconstructed video data after applying quantization and generates additional blurriness. In motion compensation, blurriness introduced by both quantization and in-loop de-block filtering of reference frames will propagate to the current frame. So the level of blurring artifacts in the current frame includes the blurriness caused by quantization and the blurriness from the reference frame. In both cases, QP can only reflect part of the blurriness distortion and is not sufficient to capture the true blurring annoyance. But with the help of QP information and the measure of high frequency energy obtained from Equation (4.10), we can trace back to the original image texture level and select appropriate normalization factors to maintain the output from Equation (4.10) in a fixed range.

Consider two compressed images with the same high frequency energy but quantized by different QP. The image quantized by higher QP must originally have more texture than the image quantized by lower QP, because the former loses more high frequency signal than the latter during quantization. Based on this property, the texture level of an original image is determined by

$$\mathrm{tex}_{n,0} \quad = \quad \Upsilon(\mathrm{QP}) \times \hat{b} \tag{4.11}$$

where $\mathrm{tex}_{n,0}$ represents the estimated texture level of the original nth frame, $\Upsilon_{1\times4}(\mathrm{QP})$ contains a set of polynomial parameters that varies with QP, and $\hat{b} = [b_{n,0}^3, b_{n,0}^2, b_{n,0}, 1]^T$, output from Equation (4.10).

Each QP corresponds to a $\Upsilon(\mathrm{QP})$, and MPEG-4 and H.264/AVC codecs have 30 and 50 different QP values respectively. Theoretically, both codecs require 30 and 50 sets of parameters respectively. That is neither a practical nor convenient solution. Therefore, we only define $\Upsilon(\mathrm{QP})$ for certain QPs and the texture level for other QPs are linearly interpolated using

$$\begin{aligned}
\frac{\text{tex}_{n,0} - \text{tex}_L}{\text{tex}_U - \text{tex}_L} &= \frac{\text{QP}_n - \text{QP}_L}{\text{QP}_U - \text{QP}_L} \\
\Rightarrow \quad \text{tex}_{n,0} &= \frac{(\text{QP}_n - \text{QP}_L)(\text{tex}_U - \text{tex}_L)}{\text{QP}_U - \text{QP}_L} + \text{tex}_L \\
\Rightarrow \quad \text{tex}_{n,0} &= \frac{(\text{QP}_n - \text{QP}_L)(\Upsilon(\text{QP}_U) \times \hat{b} - \Upsilon(\text{QP}_L) \times \hat{b})}{\text{QP}_U - \text{QP}_L} \\
&\quad + \Upsilon(\text{QP}_L) \times \hat{b}. \qquad (4.12)
\end{aligned}$$

where QP_L, QP_U, tex_L, and tex_U are the closest upper and lower QPs, and the corresponding outputs from Equation (4.11), respectively. Finally, $\text{tex}_{n,0}$ is bounded by

$$\text{tex}_n = \begin{cases} 1 & \text{if } \text{tex}_{n,0} \leq 1 \\ 10 & \text{if } \text{tex}_{n,0} \geq 10 \end{cases}, \qquad (4.13)$$

Because MPEG-4 and H.264/AVC have different ranges of QP, the $\Upsilon(\text{QP})$ of these two codecs are defined separately as

$$\Upsilon(\text{QP}_{H.264}) = \begin{bmatrix} 1.72E{+}5, & -1.833E{+}4, & 8.3E{+}2, & 0 \\ 2.02E{+}5, & -2.044E{+}4, & 8.7E{+}2, & 0 \\ 3.19E{+}5, & -2.874E{+}4, & 1.0E{+}3, & -10 \\ 3.72E{+}5, & -3.277E{+}4, & 1.1E{+}3, & 0 \\ 7.14E{+}3, & -2.002E{+}4, & 1E{+}3, & -3 \end{bmatrix}$$

and

$$\Upsilon(\text{QP}_{MPEG4}) = \begin{bmatrix} 1.73E{+}5, & -1.83E{+}4, & 8.4E{+}2, & 0 \\ 3.36E{+}5, & -2.9E{+}4, & 1.04E{+}3, & -10 \\ 4.02E{+}5, & -3.27E{+}4, & 1.09E{+}3, & -10 \\ 3.99E{+}5, & -3.21E{+}4, & 1.08E{+}3, & 0 \\ 4.76E{+}5, & -3.63E{+}4, & 1.16E{+}3, & -10 \\ 4.23E{+}5, & -3.4E{+}4, & 1.15E{+}3, & 0 \end{bmatrix}$$

where $\text{QP}_{H.264} \in [10, 20, 30, 40, 45]$, and $\text{QP}_{MPEG4} \in [2, 10, 15, 20, 25, 30]$. The parameter determination process will be discussed in Sec. 4.6.

After the texture level of original image is known from Equation (4.13), the normalization upper and lower bounds, ub_n and lb_n, are calculated by

$$\begin{bmatrix} \mathrm{ub}_n \\ \mathrm{lb}_n \end{bmatrix} = \Phi \times \begin{bmatrix} \mathrm{tex}_n^2 \\ \mathrm{tex}_n \\ 1 \end{bmatrix} \tag{4.14}$$

where

$$\begin{aligned} \Phi_{H.264} &= \begin{bmatrix} 0, & 1.7E{-}3, & 6.9E{-}3 \\ 1E{-}4, & 1.4E{-}3, & 3.9E{-}3 \end{bmatrix}, \\ \Phi_{MPEG4} &= \begin{bmatrix} 0, & 1.7E{-}3, & 7E{-}3 \\ 1.1E{-}3, & 5.2E{-}3, & 0 \end{bmatrix}. \end{aligned}$$

The process of parameter fitting for Φ will be discussed in Sec. 4.6.

Outputs of Equation (4.11) for both MPEG-4 and H.264/AVC are shown in Fig. 4.8. Each curve corresponds to one QP, and all of them behave in a similar way for either MPEG-4 or H.264/AVC. This suggests that the trend of texture level change of each QP case is the same, and hence, it is reasonable to use linear interpolation to obtain the texture level by neighboring QP information. Also, the shift between curves and the different range of x-axis data of MPEG-4 and H.264/AVC shows the necessity of providing Υ to these two codecs separately. The solid and dash lines in Fig. 4.9 represent the upper and lower bound for each texture level respectively. Notice that the distance between ub_n and lb_n for each texture level is very different in these two codecs.

Once the normalization upper and lower bounds are available, the output of Equation (4.10) is normalized by

$$b_n = \frac{4(b_{n,0} - \mathrm{lb}_n)}{\mathrm{ub}_n - \mathrm{lb}_n} + 1, \tag{4.15}$$

where b_n is the normalized blurriness score, which ranges from 1 to 5 and higher scores represent less blurriness.

4.4.6 Cognitive Module

Humans have limited resolution in judging blurriness when compressed video is extremely blurred or sharp. Human perception can only classify them as either *Very*

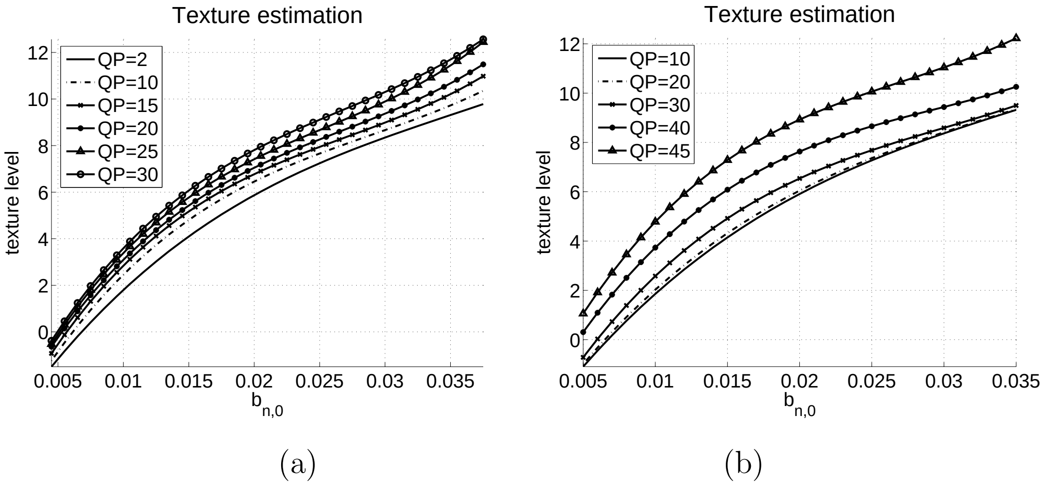


Figure 4.8: Estimation of original image texture for (a) MPEG-4, and (b) H.264/AVC codecs

blurred or *Sharp*. This is called *Saturation*. Also, viewer perceived blurriness does not linearly correlate with the raw objective blurriness measure (i.e. edge profile, high frequency energy) because human perception is a non-linear process. Therefore, we compensate for these phenomena by modulating the measured blurriness value from Equation (4.15) with a *Cognitive Module*.

First in the cognitive module, the very low texture cases is compensated using

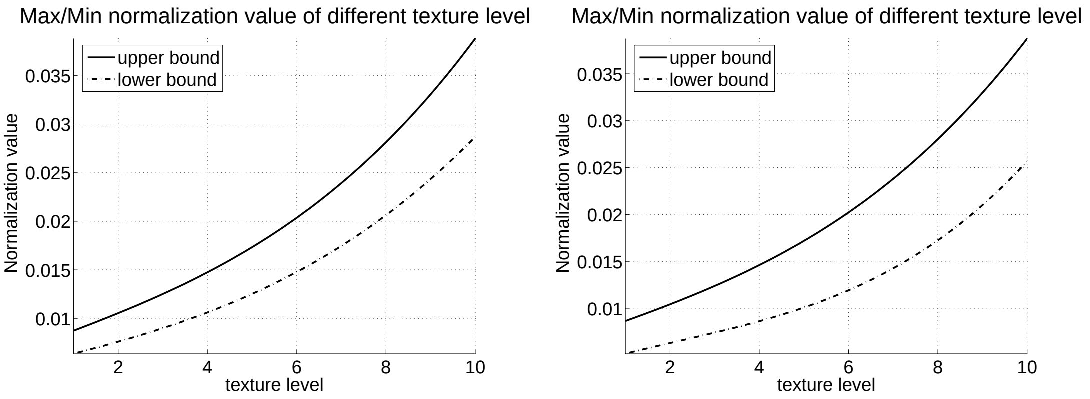
$$b'_n = \begin{cases} b_n + 0.25 & \text{if tex}_n \leq 1.85 \\ b_n & \text{Otherwise} \end{cases} . \tag{4.16}$$

Second, cognitive saturation is emulated by

$$b''_n = \begin{cases} 5 & \text{if } b_n \geq 4.14 \\ 1 & \text{if } b_n \leq 1.25 \\ b'_n & \text{Otherwise.} \end{cases} \tag{4.17}$$

Finally, a non-linear mapping function is applied to the blurriness score from Equation (4.15) using

$$b'''_n = \Theta \times \begin{bmatrix} b''^4_n \\ b''^3_n \\ b''^2_n \\ b''_n \\ 1 \end{bmatrix}, \tag{4.18}$$



(a) (b)

Figure 4.9: Normalization of upper and lower bound of each texture level for (a) MPEG-4, and (b) H.264/AVC codecs

where

$$\begin{aligned}\Theta_{MPEG4} &= [0,\ \ -2.1E\text{–}3,\ \ 1.07E\text{–}1,\ \ 3.84E\text{–}1,\ \ -1.4] \\ \Theta_{H264} &= [-1.2E\text{–}2,\ \ 6.3E\text{–}2,\ \ 6E\text{–}4,\ \ 9.086E\text{–}1,\ \ 1.4E\text{–}3],\end{aligned}$$

and the parameter determination process will be explained in Sec. 4.6, and b'''_n is the blur score of nth frame.

The final blurriness score of the entire sequence is

$$B = \frac{1}{N-1}\sum_{n=0}^{N-1} b'''_n, \tag{4.19}$$

where B is the blurriness score of a sequence, and N is the number of total frames.

4.5 Experimental Setup for PBM Tuning and Validation

4.5.1 Data Preparation

Two sets of video data are generated in this experiment, a training video database is used to determine several important parameters and a separate testing video database is dedicated to evaluate the metric's performance. All video sequences are sampled in YCbCr 4:2:0 with frame size either QVGA (320×240) or QVGA (176×144) and frame rate 30 frame per second(fps). These data are obtained from Ref. [76, 77]. Table 4.1 provides a brief description of all training and testing sequences. Shifting motion is quantified by the ratio of number of frames that have high shifting motion against the number of all frames.

MPEG-4 and H.264/AVC JM12 [78] are employed to generate various compression scenarios. In order to isolate the spatial quality degradation from temporal effects, frame skipping and rate control functions are turned off. The sequences are compressed using different QPs. The training set is encoded using $\mathrm{QP}_{H.264} \in [10, 20, 30, 40, 45]$, and $\mathrm{QP}_{MPEG4} \in [2, 10, 15, 20, 25, 30]$. The testing set is encoded by $\hat{\mathrm{QP}}_{H.264} \in [15, 25, 35, 45]$, and $\hat{\mathrm{QP}}_{MPEG4} \in [5, 13, 23, 27]$. We also use the H.264/AVC built in de-blocking filter with different strengths to simulate the blurriness caused by post-processing. The filtering strength is controlled by α and β, where $\alpha, \beta \in [-6, 0, 6]$ and higher values mean stronger low-pass filtering.

4.5.2 Objective Test Methodology

The metric's performance is objectively tested by two criteria adopted from [79] and one new criterion. These criteria are based on prior knowledge of video compression and the human visual system. A well-designed blurriness metric should have the ability to satisfy the following expectations:

E-1. With the same level of post-process filtering, blurring score should increase as QP increases.

E-2. For a fixed QP, the blurring value should increase along with $[\alpha, \beta]$.

E-3. Since human eyes have less sensitivity on the frames with high shifting motion, applying different QP on frames with high $MA_{s,T}$ should not influence the human-perceived blurriness level. Hence, the sequences that have high shifting motion, FOREMAN, FOOTBALL, and RUGBY are separated into two parts according to the strength of $MA_{s,T}$. Frames with high $MA_{s,T}$ are encoded using $QP_{hs} = [2, 10, 17, 25]$. The rest of the frames, which are static motion frames, are encoded with $QP_{ls} = 30$. The expectation is that the blurriness should not vary with different QP_{hs}. This experiment is only carried out with the MPEG-4 codec.

4.5.3 Subjective Test Methodology

The subjective experiments are carried out using the *Double Stimulus Impairment Scale (DSIS)* method recommended in ITU-R BT.500 [18]. This methodology is very similar to the *Double Stimulus Continuous Quality Scale* (DSCQS) described in Section 2.3, except different semantic meaning of voting scores, where DSIS is more suitable to evaluate the strength of specific compression artifact. Eight testers were employed to provide the quality value for the parameter fitting experiment. Another twenty viewers comprised of five females and fifteen males participated in the performance evaluation experiment, ages ranging from 25 to 50. Examiners are asked to score the video sequences' blurriness levels from 1 to 5 with the semantic meaning - "Very annoying", "Annoying", "Slightly annoying", "Perceptible" and "Imperceptible". The increment interval between level is 0.1. The presented video sequences are 10s long.

A test has *Introduction* and *Testing* sessions. The duration of the introduction session is 10 minutes long and it includes an eye test, testing purpose explanation, scoring method, showing sample sequences with different levels of blurriness, and a question period. The testing session is composed of several testing pairs. Each testing pair shows both original and blurred sequences. The viewers are asked to vote for the perceived blurriness difference between these two sequences and it is converted into Difference Mean Opinion Score (DMOS), where higher values represent less blurriness.

The performance of objective blurriness metrics is quantified by measuring the correspondence between the DMOS and the metrics' output as introduced in Section 2.3.3. The correspondence is measured by Pearson and Spearman correlation coefficients, Root-Mean-Square-Error (RMSE) [56], and are denoted as C_P, C_S, and C_R, between the metric's output and DMOS data. Higher C_P, C_S and lower C_R indicate better metric performance.

4.6 Parameter Fitting

4.6.1 Blurriness Sensitivity Vector

In order to determine the importance of each DCT coefficient, a band-blocking filter is designed to remove the same percentage of energy for each band. The band-blocking frequency runs from $\frac{1}{(N_B-1)}\sqrt{i^2+j^2}$ to $\frac{1}{(N_B-1)}\sqrt{(i+1)^2+(j+1)^2}$, where $i, j \in [1, 2, \cdots, 7]$. This band-blocking is applied to the 40th frame of all training sequences and the testers are asked to vote on the perceived blurriness. Each band-blocking interval has one DMOS and it is the average DMOS of all training samples of all testers. As shown in Fig. 4.10, the 3rd DCT AC coefficient, which is the 4th DCT coefficient, has the lowest DMOS. It reflects a fact that with the same percentage energy loss in each band, the 4th band introduces the most pronounced blurriness. In other words, the 4th band is very sensitive to blurriness. Thus, the values of the BSV are obtained using $\Psi = (5 - \text{DMOS})/4$.

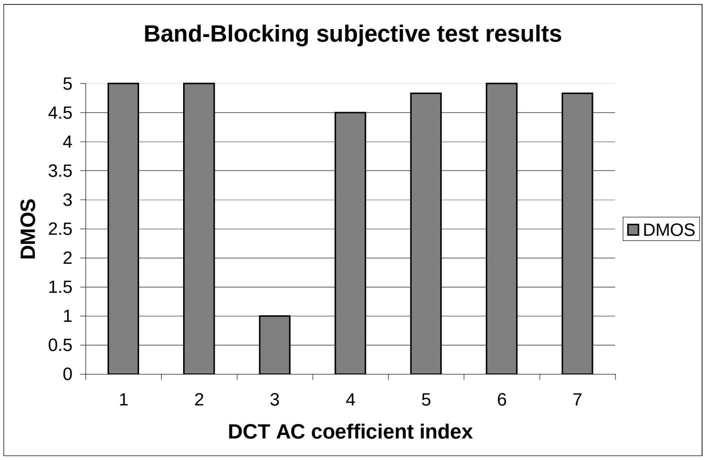


Figure 4.10: Subjective DMOS values of band-blocking filtered data

4.6.2 Normalization Module

Figure 4.11 illustrates the procedure for obtaining Υ and Φ. The procedure for determining values of Υ is shown in Fig. 4.11(a). First, several frames from the *original* training sequences are transformed by DCT and the summation of the 4th to 8th DCT coefficients are used for indicating the texture level of images before compression: tex'_n. Meanwhile, those frames are compressed using MPEG-4 and H.264/AVC with the given QP_{MEPG4} and $\text{QP}_{H.264}$ respectively, and the corresponding $b_{n,0}$ is calculated using Equation (4.10). Please note that the QP value in Fig. 4.11 represents either QP_{MEPG4} or $\text{QP}_{H.264}$. In the following, in order to find a function to map $b_{n,0}$ to tex'_n, a kernel function in polynomial form

$$\text{tex}'_n = \Upsilon_0 \cdot b^3_{n,0} + \Upsilon_1 \cdot b^2_{n,0} + \Upsilon_2 \cdot b_{n,0} + \Upsilon_3. \tag{4.20}$$

is trained using supervised learning, where $b_{n,0}$ and tex'_n are used as the training data and labels respectively. $R_{tex'_n,\Upsilon}$ the residual between left and right terms in Equation is given by (4.20)

$$R^2_{tex'_n,\Upsilon} = \sum_{z=1}^{N_t}[\text{tex}'_{n,z} - (\Upsilon_0 \cdot b^3_{n,0} + \Upsilon_1 \cdot b^2_{n,0,z} + \Upsilon_2 \cdot b_{n,0,z} + \Upsilon_3)]^2, \tag{4.21}$$

where $\text{tex}'_{n,z}$ and $b_{n,0,z}$ are the tex'_n and $b_{n,0}$ values of zth training data respectively, and N_t is the number of training data. The partial derivatives are

$$\begin{array}{rcl}
\dfrac{\partial R^2_{tex'_n,\Upsilon}}{\partial \Upsilon_3} & = & -2\displaystyle\sum_{z=1}^{N_t}[\text{tex}'_n - (\Upsilon_0 \cdot b^3_{n,0,z} + \Upsilon_1 \cdot b^2_{n,0,z} + \Upsilon_2 \cdot b_{n,0,z} + \Upsilon_3)] = 0, \\
\dfrac{\partial R^2_{tex'_n,\Upsilon}}{\partial \Upsilon_2} & = & -2\displaystyle\sum_{z=1}^{N_t}[\text{tex}'_{n,z} - (\Upsilon_0 \cdot b^3_{n,0,z} + \Upsilon_1 \cdot b^2_{n,0,z} + \Upsilon_2 \cdot b_{n,0,z} + \Upsilon_3)]b_{n,0,z} = 0, \\
\dfrac{\partial R^2_{tex'_n,\Upsilon}}{\partial \Upsilon_1} & = & -2\displaystyle\sum_{z=1}^{N_t}[\text{tex}'_{n,z} - (\Upsilon_0 \cdot b^3_{n,0,z} + \Upsilon_1 \cdot b^2_{n,0,z} + \Upsilon_2 \cdot b_{n,0,z} + \Upsilon_3)]b^2_{n,0,z} = 0, \\
\dfrac{\partial R^2_{tex'_n,\Upsilon}}{\partial \Upsilon_0} & = & -2\displaystyle\sum_{z=1}^{N_t}[\text{tex}'_{n,z} - (\Upsilon_0 \cdot b^3_{n,0,z} + \Upsilon_1 \cdot b^2_{n,0,z} + \Upsilon_2 \cdot b_{n,0,z} + \Upsilon_3)]b^3_{n,0,z} = 0.
\end{array} \tag{4.22}$$

These lead to the equations

$$
\begin{aligned}
\sum_{z=1}^{N_t} \mathrm{tex}'_{n,z} &= \Upsilon_0 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^3 + \Upsilon_1 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^2 + \Upsilon_2 \cdot \sum_{z=1}^{N_t} b_{n,0,z} + N_t \Upsilon_3, \\
\sum_{z=1}^{N_t} b_{n,0,z} \mathrm{tex}'_{n,z} &= \Upsilon_0 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^4 + \Upsilon_1 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^3 + \Upsilon_2 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^2 + N_t \Upsilon_3 \sum_{z=1}^{N_t} b_{n,0,z}, \\
\sum_{z=1}^{N_t} b_{n,0,z}^2 \mathrm{tex}'_{n,z} &= \Upsilon_0 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^5 + \Upsilon_1 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^4 + \Upsilon_2 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^3 + N_t \Upsilon_3 \sum_{z=1}^{N_t} b_{n,0,z}^2, \\
\sum_{z=1}^{N_t} b_{n,0,z}^3 \mathrm{tex}'_{n,z} &= \Upsilon_0 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^6 + \Upsilon_1 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^5 + \Upsilon_2 \cdot \sum_{z=1}^{N_t} b_{n,0,z}^4 + N_t \Upsilon_3 \sum_{z=1}^{N_t} b_{n,0,z}^3.
\end{aligned}
\tag{4.23}
$$

These equations can be written in a matrix format as

$$
\begin{aligned}
\begin{bmatrix} \sum_{z=1}^{N_t} \mathrm{tex}'_{n,z} \\ \sum_{z=1}^{N_t} b_{n,0,z} \mathrm{tex}'_{n,z} \\ \sum_{z=1}^{N_t} b_{n,0,z}^2 \mathrm{tex}'_{n,z} \\ \sum_{z=1}^{N_t} b_{n,0,z}^3 \mathrm{tex}'_{n,z} \end{bmatrix}
&= \begin{bmatrix}
\sum_{z=1}^{N_t} b_{n,0,z}^3 & \sum_{z=1}^{N_t} b_{n,0,z}^2 & \sum_{z=1}^{N_t} b_{n,0,z} & N_t \\
\sum_{z=1}^{N_t} b_{n,0,z}^4 & \sum_{z=1}^{N_t} b_{n,0,z}^3 & \sum_{z=1}^{N_t} b_{n,0,z}^2 & N_t \sum_{z=1}^{N_t} b_{n,0,z} \\
\sum_{z=1}^{N_t} b_{n,0,z}^5 & \sum_{z=1}^{N_t} b_{n,0,z}^4 & \sum_{z=1}^{N_t} b_{n,0,z}^3 & N_t \sum_{z=1}^{N_t} b_{n,0,z}^2 \\
\sum_{z=1}^{N_t} b_{n,0,z}^6 & \sum_{z=1}^{N_t} b_{n,0,z}^5 & \sum_{z=1}^{N_t} b_{n,0,z}^4 & N_t \sum_{z=1}^{N_t} b_{n,0,z}^3
\end{bmatrix} \\
&\times \begin{bmatrix} \Upsilon_0 \\ \Upsilon_1 \\ \Upsilon_2 \\ \Upsilon_3 \end{bmatrix},
\end{aligned}
\tag{4.24}
$$

and can be organized as

$$
\begin{aligned}
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
b_{n,0,1} & b_{n,0,2} & \cdots & b_{n,0,N_t} \\
b_{n,0,1}^2 & b_{n,0,2}^2 & \cdots & b_{n,0,N_t}^2 \\
b_{n,0,1}^3 & b_{n,0,2}^3 & \cdots & b_{n,0,N_t}^3
\end{bmatrix}
\begin{bmatrix} \mathrm{tex}'_{n,1} \\ \mathrm{tex}'_{n,2} \\ \vdots \\ \mathrm{tex}'_{n,N_t} \end{bmatrix}
&= \begin{bmatrix}
1 & 1 & \cdots & 1 \\
b_{n,0,1} & b_{n,0,2} & \cdots & b_{n,0,N_t} \\
b_{n,0,1}^2 & b_{n,0,2}^2 & \cdots & b_{n,0,N_t}^2 \\
b_{n,0,1}^3 & b_{n,0,2}^3 & \cdots & b_{n,0,N_t}^3
\end{bmatrix} \\
&\times \begin{bmatrix}
b_{n,0,1}^3 & b_{n,0,1}^2 & b_{n,0,1} & 1 \\
b_{n,0,2}^3 & b_{n,0,2}^2 & b_{n,0,2} & 1 \\
\vdots & \vdots & \vdots & \vdots \\
b_{n,0,N_t}^3 & b_{n,0,N_t}^2 & b_{n,0,N_t} & 1
\end{bmatrix}
\begin{bmatrix} \Upsilon_0 \\ \Upsilon_1 \\ \Upsilon_2 \\ \Upsilon_3 \end{bmatrix}
\end{aligned}
$$

Thus, Equation (4.24) can be represented as

$$\begin{bmatrix} \text{tex}'_{n,1} \\ \text{tex}'_{n,2} \\ \vdots \\ \text{tex}'_{n,N_t} \end{bmatrix} = \begin{bmatrix} b^3_{n,0,1} & b^2_{n,0,1} & b_{n,0,1} & 1 \\ b^3_{n,0,2} & b^2_{n,0,2} & b_{n,0,2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ b^3_{n,0,N_t} & b^2_{n,0,N_t} & b_{n,0,N_t} & 1 \end{bmatrix} \begin{bmatrix} \Upsilon_0 \\ \Upsilon_1 \\ \Upsilon_2 \\ \Upsilon_3 \end{bmatrix}. \tag{4.25}$$

In matrix notation, Equation (4.25) is given by

$$\bar{T}_{N_t \times 1} = \bar{B}_{N_t \times 4} \bar{\Upsilon}_{N_t \times 1}.$$

It can be solved by premultiplying the left hand side of above equation with the matrix transpose $\bar{B}^T_{N_t \times 4}$. This matrix equation can be solved numerically, or can be inverted directly if it is well formed, to yield the solution vector

$$\bar{\Upsilon}_{N_t \times 1} = (\bar{B}^T_{N_t \times 4} \bar{B}_{N_t \times 4})^{-1} \bar{B}^T_{N_t \times 4} \bar{T}_{N_t \times 1}.$$

The process for obtaining Φ is shown in Fig. 4.11. First, the training images are compressed with $\text{QP}'_{MPEG4} = 2, 30$ and $\text{QP}'_{H264} = 10, 45$. Then tex_n and $b_{n,0}$ are calculated from Equation (4.13) and (4.10) , respectively. Finally, tex_n and $b_{n,0}$ are used to train a quadratic polynomial kernel function

$$b_{n,0} = \Phi_0 \cdot \text{tex}^2_n + \Phi_1 \cdot \text{tex}_n + \Phi_2 \cdot \text{tex}^0_n. \tag{4.26}$$

The training process is the same as Equations (4.20 - 4.25). Coefficients trained by $\text{QP}'_{MPEG4} = 2$ or $\text{QP}'_{H264} = 10$ are used for estimating the upper bound for normalization, ub_n, and the coefficients trained by $\text{QP}'_{MPEG4} = 30$ or $\text{QP}'_{H264} = 45$ are used for estimating lower bound, lb_n.

4.6.3 Cognitive Module

This module is designed to modify the relationship between the DMOS value and b_n for different blurriness levels. The various blurriness levels are simulated by compressing the video data with different QPs. Furthermore, in order to isolate any temporal human visual factors (i.e., motion masking), only several frames from the training sequences are compressed instead of entire sequences. The selected images

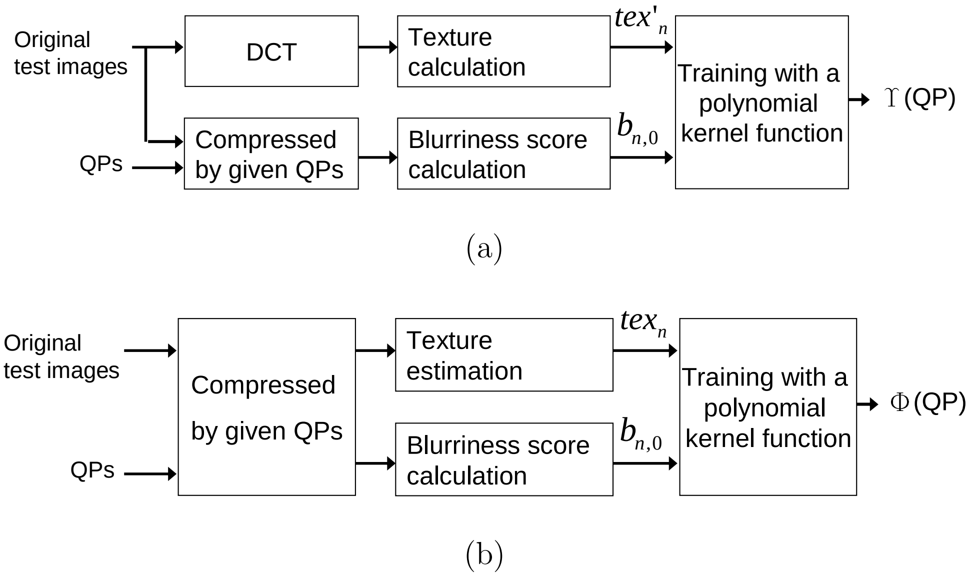


Figure 4.11: Training procedure for obtaining (a) Υ, and (b) Φ.

are compressed using MPEG-4 and H.264/AVC with the same QP_{MEPG4} and $\text{QP}_{H.264}$ used in Section 4.6.2. These compressed images were shown to viewers who scored the blurriness levels. The average DMOS for each QP through all samples is used as the training label and the corresponding average b_n obtained from Equation (4.15) is used as the training data. The training data and labels are employed first to determine the saturation thresholds in (4.17), and second to fit the coefficients of the quintic polynomials for H.264/AVC and MPEG-4 codecs, Θ_{MPEG4} and $\Theta_{H.264}$, respectively. The training process follows Equations (4.20 - 4.25).

4.7 Analysis of Experimental Results

4.7.1 Objective evaluation

Definitions of E-1, E-2, and E-3 have been given in Section 4.5.2. The E-1 expectation assumes that the metrics' output should increase monotonically with QP value regardless of type of codec. Performance on this point is quantified using the Spearman correlation coefficient, C_S. The metrics that are able to fulfill the E-2 expectation will be denoted as "Y", and "X" otherwise. The consistency of E-3 expectation is quantified by the average variance of metrics' output for all QP_{hs} and

QP$_{ls}$ combinations through the three sequences; higher value means that the metric is affected by QP$_{hs}$ and has worse performance. The objective evaluation results are shown in Table 4.2.

Overall, PBM has the highest score in E-1 expectation, Kurtosis is the second highest, and the DSS is third. Among them, the scores for Kurtosis are very close to PBM's. It indicates that these three metrics' outputs increases monotonically as QP increases. The DCT-histogram has a very low value in the E-1(H.264) case. AETS performs evenly in both E-1(MPEG-4) and E-2(H.264) but is always less than 0.8. All metrics are able to meet the E-2 expectation. For the E-3 expectation, PBM has the lowest variance. Thus, PBM is the only metric that is immune to the influence of different QP$_{hs}$.

Given the objective results, the DCT-histogram metric is excluded from the following performance evaluation process since it fails in E-1 requirement.

4.7.2 Subjective evaluation

Figure 4.12 presents a scatter plot of the objective blurring score and the corresponding DMOS for all testing cases. A curve fitted by the scattering data with a second order polynomial function is plotted on each of the figures. It is used to represent the correlation between the objective metric's outputs and the subjective data. Note that the output of all metrics except PBM have been normalized manually and converted to the same range and semantic meaning as the subjective rating, where higher value means less blurriness. A good metric should have a linear correlation between the x-axis and y-axis data, and hence, the curve should close to a straight line. The AETS and AETW have scattered distributions and both of them give most cases a very low objective value. DSS shows higher correlation with subjective DMOS compared to the previous two metrics, but its estimation is a bit overly optimistic since it gives many low DMOS cases high objective scores. The Kurtosis metric also has a very scattered distribution similar to AETS and AETW. Overall, PBM has much better correspondence than to all the other metrics. In some very extreme cases, such as the cases with very low and high DMOS, PBM still matches the subjective DMOS accurately. This indicates the importance of the saturation function in the Cognitive Module. In the cases with medium blurriness ($2 \leq DMOS \leq 4$), the high linear correspondence between PBM's output and DMOS shows the performance improvement from the non-linear mapping function in the Cognitive Module.

Tables 4.3 and 4.4 show the quantified performance - C_P and C_R of all test cases. Most metrics perform fine in the cases encoded by MPEG-4. PBM has the best performance and AETW is second, while AETS performs worst. It is interesting to see that the performance difference between PBM and AETW is marginal, and DSS has performance almost identical to Kurtosis. In the cases encoded by H.264/AVC, PBM has very high correlation and low C_R compared to all the other metrics. Ignoring PBM, only DSS has correlation more than 0.7, and other metrics have very low correlation. This is especially true of AETW, the second best metric in the case encoded by MPEG-4; its performance drops dramatically here. Hence, most metrics fail in predicting the blurriness caused by the H.264/AVC coder but PBM is still able to estimate it accurately. Overall performance is calculated by averaging the C_P and C_R of all test cases. It is worth mentioning that AETS, AETW, and Kurtosis have similar prediction accuracy, with $C_P < 0.7$. DSS performs better, but it is still worse than PBM. Finally, PBM has correlation close to 0.9 and significantly outperforms all the other metrics.

A detailed performance report for each sequence compressed by both coders is shown in Table 4.5. The experimental results show that AETS, AETW, and Kurtosis perform poorly in the FOREMAN and FOOTBALL sequences. That might be because of inaccurate edge profile determination due to high shift motion. Besides these two sequences, AETS also has very low performance on MOTHER DAUGHTER. It could be because of low texture and fewer edge points. DSS works fairly well in most cases. As with the two sequences mentioned before, the Kurtosis metric also has low prediction accuracy on BUS. PBM-raw denotes the output from (4.9), which only includes the weighted high frequency energy. Even though it is not adjusted by any perceptual module, it still outperforms all other metrics except full PBM. This shows that high frequency band energy is a better blurriness estimation basis compared to edge profile oriented approaches. After including all the perceptual modules, a complete PBM has correlation coefficient ranging from 0.85 to 0.97, and C_R from 1.35 to 2.35. None of the other metrics has better performance than PBM in any sequence. Comparing PBM with PBM-raw, the PBM has 0.06, 0.07, and 0.04 higher C_P than PBM-raw in FOREMAN, FOOTBALL, and RUGBY sequences respectively. These sequences have high motion and complicated content. The benefits brought by VBSM is more observable and it might explain the performance difference between PBM and PBM-raw. In the relatively simple sequence, MOTHER DAUGHTER, although PBM and PBM-raw have the same C_P, PBM still has lower C_R. Overall, PBM is better than PBM-raw, and another strength of PBM is that its output is normalized and it can be used for cross-sequence comparison.

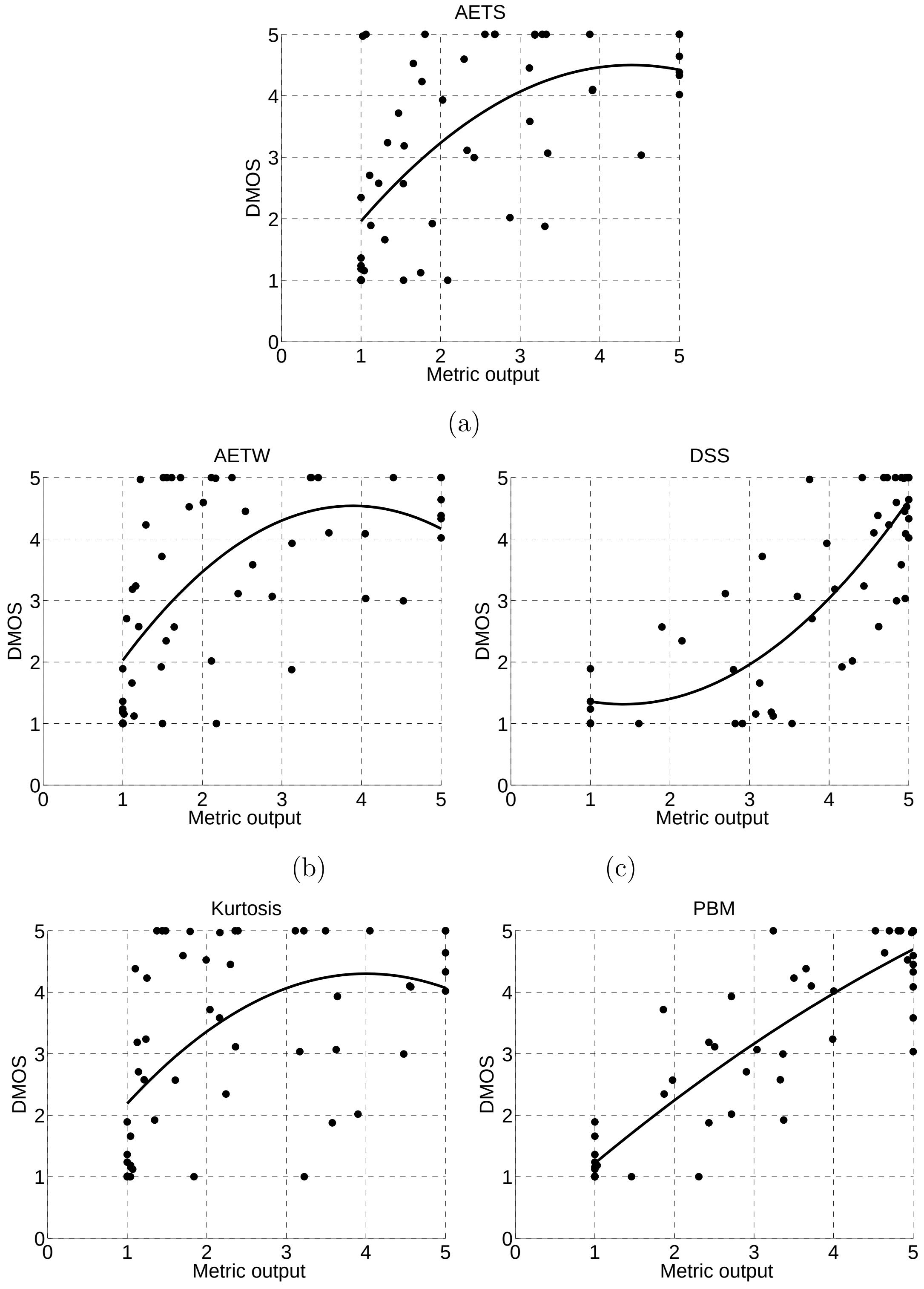
4.8 Summary

A reference-free human perception based blurriness metric for compressed video is proposed in this Chapter. It gauges blurriness level by measuring the energy in high spatial frequency bands. Several human visual and cognitive factors are included to enhance the blurriness prediction accuracy. Output of the proposed metric is content independent and the analysis of detailed experimental results shows significant performance improvement provided by each module. Both objective and subjective performance evaluation show high blurriness prediction accuracy for PBM compared to other metrics.

PUBLICATIONS

Kai-Chieh Yang, Clark C. Guest and Pankaj K. Das, “Perceptual Sharpness Metric for Compressed Video”, *Proc. IEEE International Conference of Multimedia and Expo*, pp. 777-780, July. 2006

Kai-Chieh Yang, Clark C. Guest, Pankaj Das, “Motion Blur Detection by Support Vector Machine”, *Proc. SPIE*, Vol. 5916, pp. 261-273, Aug. 2005



(a)

(b)

(c)

(d)

(e)

Figure 4.12: Subjective DMOS compared against metrics' output of (a)AETS, (b)AETW, (c)DSS, (d)Kurtosis, and (e)PBM

Table 4.1: Content description and format of training and testing sequences for PBM

Training sequences

Size	Name	Shift motion(%)	Object motion	Texture level	Content description
QVGA	CONTAINER	0	Low	Low	Static camera with a slow moving boat
	MOBILE	0	Low	Medium	Static camera aims at many moving objectives
	COASTGRD	0	Low	High	Slow panning camera follows slow moving boats
QCIF	CARPHONE	0	Low	Medium	Static camera with talking head

Testing sequences

Size	Name	Shift motion(%)	Object motion	Texture level	Content description
QVGA	FAMILY	0	Low	High	Static camera with multiple scenes
	FOREMAN	13	Medium	Medium	Talking head scene shift to another complex scene
	MOTHER DAUGHTER	0	Low	Low	Static camera with talking head
QCIF	FOOTBALL	20	High	Medium	Sports content with high motion scene shifting
	RUGBY	12	High	Medium	Sports content with high motion scene shifting
	BUS	0	Medium	High	Camera tracks a moving bus

Table 4.2: Results of objective performance evaluation of all blurriness metrics

	E-1(MPEG-4)	E-1(H.264/AVC)	E-2	E-3
PBM	0.98	0.98	Y	0.0014
AETS	0.78	0.77	Y	1.1
AETW	0.89	0.81	Y	0.27
DSS	0.94	0.90	Y	0.01
DCT-histogram	0.80	0.62	Y	0.52
Kurtosis	0.97	0.96	Y	0.1

Table 4.3: Correlation value, C_P, between all blurriness metric's outputs and subjective rankings.

	AETS	AETW	DSS	Kurtosis	PBM
MPEG4	0.73	0.85	0.81	0.81	0.87
H.264/AVC	0.55	0.50	0.78	0.53	0.91
Overall	0.64	0.67	0.79	0.67	0.89

Table 4.4: RMSE value, C_R, between all blurriness metric's outputs and subjective rankings.

	AETS	AETW	DSS	Kurtosis	PBM
MPEG4	2.60	2.38	2.74	2.59	1.64
H.264/AVC	5.23	5.78	3.45	5.66	2.12
Overall	3.91	4.08	3.09	4.13	1.88

Table 4.5: Detail quantitative performance, $C_P(C_R)$, of each blurriness metric and sequence

	FAMILY	FOREMAN	MOTHER DAUGHTER	BUS	FOOTBALL	RUGBY
AETS	0.76 (3.81)	0.41 (5.94)	0.44 (4.08)	0.75 (3.73)	0.60 (4.16)	0.80 (3.10)
AETW	0.78 (3.57)	0.50 (5.37)	0.79 (3.46)	0.68 (4.32)	0.57 (4.29)	0.77 (3.36)
DSS	0.82 (3.28)	0.81 (2.64)	0.78 (3.41)	0.81 (2.53)	0.71 (4.10)	0.91 (1.97)
Kurtosis	0.79 (3.73)	0.60 (4.75)	0.78 (3.55)	0.61 (4.72)	0.58 (4.06)	0.73 (3.60)
PBM-raw	0.94 (2.02)	0.82 (2.14)	0.88 (2.19)	0.84 (1.94)	0.83 (2.57)	0.93 (1.70)
PBM	0.95 (1.35)	0.88 (1.86)	0.88 (1.50)	0.86 (1.68)	0.9 (2.35)	0.97 (1.38)

5

Perceptual Temporal Quality Metric (PTQM)

5.1 Introduction

The quality degradation introduced by compression can be separated into two principal types of artifacts [4]: spatial and temporal. In the spatial domain, several key artifacts and their related metrics, such as blockiness, blurring and ringing, have been extensively studied [1, 40–43, 45]. Temporal degradation includes three major artifacts: flickering (blinking), motion jerkiness, and jittering. Flickering is usually caused by visual quality fluctuation at the same spatial position but in different temporal locations. The severity of flickering can be estimated by observing spatial quality fluctuation along the time axis. However, there is a lack of comprehensive research into quantifying jerkiness and jittering, which occurs very frequently in real time video communication, i.e. video telephony and broadcasting.

When a video sequence is transmitted through a bandwidth-limited and error-prone channel (e.g. packet-switched wireless network), the video playing smoothness at the receiving end suffers from several sources of degradation. The encoder might discard some frames during encoding in an attempt to reduce the data rate, while decoder might not be able to play all received frames because of limited computational capability. Moreover, in an error-prone channel, packet loss may corrupt the video data and an entire frame may be lost. The relationship between packet loss and temporal quality has been studied by Claypool and Tanner [80]. However, among these three degradation sources, the temporal quality loss caused by an error-prone network is marginal compared to the other two degradation sources. In the case of substantial frame loss, the viewer may observe frame freezing because most video decoders automatically repeat the last frame received before a dropped frame. Jerkiness is the result of regular frame dropping by the codec in order to down-sample the frame rate and meet the bits budget. When jerkiness occurs, viewers perceive regular frame freezing and discontinuous motion. Jitter is usually caused by irregular frame skipping by the codec or frame loss during data transmission. In this case, viewers perceive irregular frame freezing and various impacts of discontinuous motion. In this chapter, temporal quality is generalized into two types: (1) Regular and (2) Irregular frame loss. The regular case refers to frame loss distributed evenly through the whole sequence, while the latter one is the case where each dropping event has variable size and relative temporal location. With the same amount of frame loss, the irregular case tends to produce a greater impact on perceived temporal quality. In order to reduce the negative impact of frame dropping on viewers, several approaches, i.e., smart frame dropping and frame interpolation [81–83], have been extensively investigated. With the help of accurate temporal quality assessment, the performance of such enhancements can be improved dramatically. Subjective video quality testing is the most convincing approach because it represents the general user experience. Average response from all viewers is known as the Mean-Opinion-Score (MOS). However, subjective experiments require testers to view video clips and assign quality scores in a well controlled viewing environment; it is inconvenient and costly, and hence, it is neither a practical nor scalable solution for live applications. Therefore, objective methods provide an alternative feasible solution.

A novel objective temporal quality metric - *Perceptual Temporal Quality Metric (PTQM)* is proposed in this chapter. Instead of just using frame rate, we treat each frame as a possible dropping instance and measure the temporal quality for each of them. These individual temporal quality measures are pooled together to form a higher level temporal quality score. In addition, since the human visual system (HVS) perceives with contrast rather than absolute signal strength, we investigate the perceptual impact of local temporal quality fluctuations and propose an explicit mathematical model. Experimental results show this is an essential factor for temporal quality prediction in practical scenarios. Several notable contributions of the PTQM are summarized below:

1 PTQM is a Non-Reference (NR) metric. This characteristic allows PTQM to be combined with other applications, such as real time temporal quality monitoring and adaptive frame skipping, more easily.

2 Temporal quality degradation caused by both regular and irregular frame loss can be accurately measured.

3 PTQM outputs both local and global temporal quality measurement. This hierarchical temporal quality report provides more flexibility and detail for further temporal quality analysis and enhancement.

4 The proposed method can accurately estimate humans' perceived visual discomfort induced by temporal discontinuity under various combinations of scenes and motion activity.

This chapter is organized as follows. Section 5.2 provides a literature survey of temporal quality related research. Some important but not yet resolved issues of temporal quality assessment are discussed in Section 5.3. Section 5.4 presents the details of PTQM. In Section 5.5, the process of subjective experiments is introduced. The fitting process of several important parameters and the result of metric performance evaluation are reported in Section 5.6. Finally, we present summaries in Section 5.7.

5.2 Related Works

Key related works on objective temporal quality metric design include [49–53, 55]. Feghali *et al.* [49] uses frame rate as the scaling factor to adjust Peak-Signal-to-Noise-Ratio (PSNR) and output a spatial-temporal quality score. In Refs. [50] and [51], a jerkiness metric based on frame rate and motion activity is proposed. In Ref. [52], Montenovo *et al.* use inter-frame correlation to locate lost frames. Some post-processing is conducted on the number of lost frames to extract several indices, such as the duration of dropping and the number of dropping occurrences, etc. The final temporal quality score is determined based on an ad-hoc analysis of those indices. Pastrana-Vidal and Gicquel [53] proposed a no-reference objective metric for measuring fluidity impairments in video service. Here, fluidity means smoothness of motion; the lack of jerkiness and jitter. This metric responds to their previous work [54] and takes the density of dropping into account. Lost frames are detected by inter-frame dissimilarity on the decoder side. After thresholding, noticeable fluidity breaks are obtained. Each fluidity break is weighted by a function of the pixel variation of the last frame at the end of the freeze and the first frame appearing after the freeze. Afterward, the fluidity break is further adjusted by a function of the fluidity break density. The paper claims that the contribution to temporal quality degradation of the fluidity breaks with more occurrences is less significant. In other words, with the same amount of frame loss, the temporal quality with scattered fluidity breaks is better than with aggregated fluidity breaks. Watanabe et al. [55] studied the subjective effect on temporal distortion with different combinations of small dropping occasions with a fixed amount of frame loss. Based on that, they tuned a logarithmic function specifically for different combinations of frame loss in each sequence. This work provides important evidence that the same amount of frame loss within one sequence can lead to different levels of subjective temporal degradation through different combinations of aggregated frame loss. It shows the prediction accuracy of the logarithmic function can be improved by tuning parameters according to the duration of each grouped frame loss.

5.3 Problem Statement

Many of the previous works estimate the temporal quality based on average frame rate and motion activity. However, human observers usually have higher tolerance to regular frame dropping than irregular because of the well preserved correlation of the remaining frames. This allows human viewers to interpolate the missing content with an inherent cognitive ability. In the irregular case, frame loss does not occur on a fixed time schedule; it can occur in groups with various lengths and uncertain time slots. The irregular case introduces a more profound impact on video playing smoothness because of unexpected perceptual changes. Hence, estimating temporal quality based on the amount of frame loss and motion activity only is not sufficient; the temporal quality contrast should also be considered.

This problem has been discussed in Refs. [52, 53, 55]. In Refs. [52, 53], they tried to resolve this issue by observing the density of dropping occasions. However, they assume the lengths of dropping occasions are nearly identical and no dependency exists between dropping occasions. In real applications, as irregular frame loss occurs, the length of dropping occasions is rarely the same, and additional discomfort is introduced when sudden temporal quality changes occur. With the same amount of frame loss and the same dropping occasion frequency, very different sizes for each dropping occasion will induce dramatically different subjective impact. In Ref. [55], although a set of specific parameters are provided experimentally for each scenario, a general mathematical model is still lacking. All of the related works assume (1) the non-dropped frames do not contribute to the temporal degradation at all, and (2) the local temporal quality is independent of neighboring dropping occasions. Nevertheless, from subjective test, we have found that the sensitivity to each dropping occasion not only varies with the number of lost frames, but also with the local temporal quality contrast to neighboring dropping occasions. Higher contrast results in more pronounced discomfort. Therefore, we claim that each dropping occasion should not be treated independently. The sensitivity to each dropping occasion should be adjusted according to the inter-dropping-occasion dependency. In this chapter, we will present the results of several experiments conducted to investigate this issue. Based on the observations, a general and more accurate temporal quality metric is proposed.

5.4 System Description

Table 5.1 presents several important notations. Figure 5.1 illustrates the system diagram of PTQM and Fig. 5.2 depicts an example of the temporal quality estimation process at each stage. Each displayed frame is treated as a potential dropping occasion and assigned a dropping severity measurement - $s_{m,n}$, which represents a measurement of video flow discontinuity of nth frame in mth scene. In the first stage, time stamp information from the bitstream is used to identify the location of dropped frames and estimate the dropping severity of each of them. Meanwhile, the video sequence is separated into several segments based on content similarity, and the motion activity of each segment is calculated. Based on the motion activity, one of the predefined motion models is applied to each segment to adjust its dropping severity in the second stage. The following three stages are carried out by the *Temporal Fusion* module. Among them, the first stage emulates human sensitivity to different levels of temporal quality contrast, and weights each dropping severity by a *temporal quality fluctuation (TQF)* function. Thereafter, the dropping severity of each dropping occasion is transformed to frame level temporal quality, denoted as $q_{m,n} \in (1, 5)$, a higher value means better temporal quality. The temporal quality of all dropping occasions within a segment are pooled together to form the segment level temporal quality - q_m. Since the algorithm earlier applied different motion models to various scenes, the overall temporal quality, Q, is simply obtained by averaging temporal quality of all segments.

Table 5.1: Important notations for PTQM

m	Index of a video scene
n	Index of a dropping occasion
ma_m	Motion activity of the mth video segment
$s_{m,n}$	Dropping severity of the nth dropping occasion at the mth video segment
$s'_{m,n}$	Motion adjusted dropping severity of the nth dropping occasion at the mth video segment
$q_{m,n}$	Temporal quality of the nth dropping occasion at the mth video segment
q_m	Temporal quality of the mth video segment
Q	Temporal quality of whole video sequence

5.4.1 Dropping Severity Estimator

Any index that increases as the video plays can be used for estimating the video playing continuity. Here, we assume that each frame has its own time stamp information from the video bitstream. If the time stamps for consecutive received frames show a gap, it is evident that one or more intervening frames have been dropped. The length of the gap, as determined by the time stamps, permits determination of

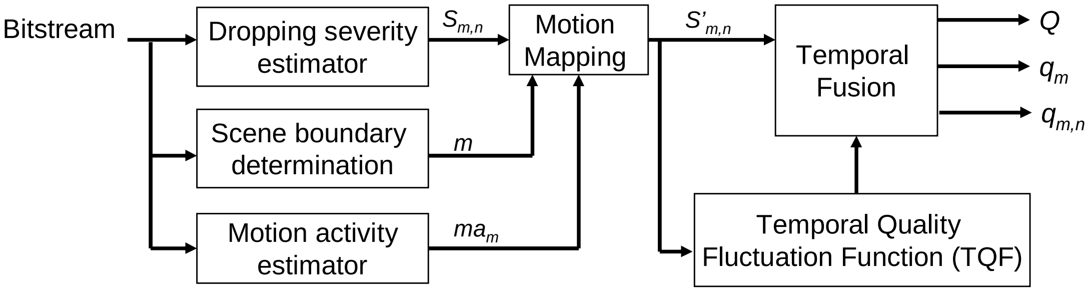

Bitstream
Dropping severity estimator
Scene boundary determination
Motion activity estimator
$S_{m,n}$
m
ma_m
Motion Mapping
$S'_{m,n}$
Temporal Fusion
Temporal Quality Fluctuation Function (TQF)
Q
q_m
$q_{m,n}$

Figure 5.1: System diagram of PTQM.

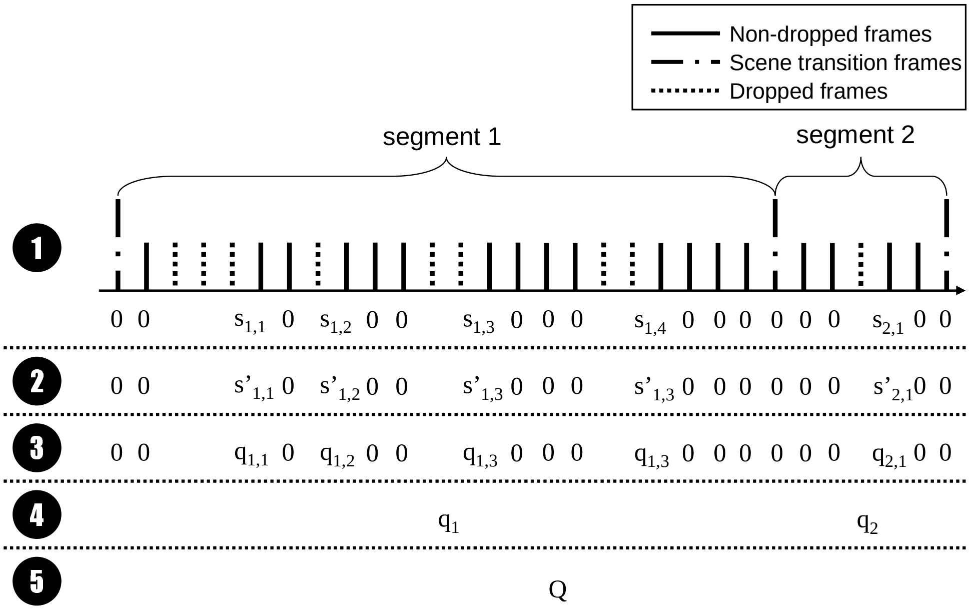

Non-dropped frames
Scene transition frames
Dropped frames
segment 1
segment 2
1
0 0 $s_{1,1}$ 0 $s_{1,2}$ 0 0 $s_{1,3}$ 0 0 0 $s_{1,4}$ 0 0 0 0 0 0 $s_{2,1}$ 0 0
2
0 0 $s'_{1,1}$ 0 $s'_{1,2}$ 0 0 $s'_{1,3}$ 0 0 0 $s'_{1,3}$ 0 0 0 0 0 0 $s'_{2,1}$ 0 0
3
0 0 $q_{1,1}$ 0 $q_{1,2}$ 0 0 $q_{1,3}$ 0 0 0 $q_{1,3}$ 0 0 0 0 0 0 $q_{2,1}$ 0 0
4
q_1
q_2
5
Q

Figure 5.2: Sample temporal quality estimation process at each stage of PTQM.

the number of consecutive frames that have been lost, which is defined as the length of the dropping occasion. The length can be zero or greater; zero means no frame dropping occurs. Furthermore, the length of the dropping occasion is normalized to a dropping severity , $s_{m,n}$, by

$$s_{m,n} = \frac{1}{R-1}\Big[\frac{\left|t_{m,n+1} - t_{m,n}\right|}{T} - 1\Big], \tag{5.1}$$

where $s_{m,n} \in (0,1)$, $t_{m,n+1}$ and $t_{m,n}$ are the time stamp of $n+1$th and nth frame in the mth segment respectively, T is the default time interval between frames at full frame rate, and the factor R is equivalent to the applicable maximum frame rate; it is used to normalize the frame dropping severity value across different frame rates and is set to 30fps in this case. As a result, a sequence that plays at 1fps would have dropping severity equal to 1, and a sequence playing at full frame rate has a dropping severity of 0.

5.4.2 Scene Boundary Determination

A video sequence may contain multiple scenes and each scene may be different in terms of the captured subject matter. Therefore, one of the reasons for doing scene boundary detection is that video sequences with similar content usually have consistent motion activity, and different motion activity level results in different temporal quality impact. In order to accurately estimate the temporal quality, motion activity for each video segment must be estimated separately. Another reason for scene boundary detection is that large displacements usually occur between the last frame of one scene and the first frame of the next scene; the motion activity is usually very high. However, since these kinds of motion do not contribute to temporal degradation at all, the motion activity caused by scene transitions should not be taken into account.

In order to detect the location of scene boundaries from the bitstream, the approach from Ref. [84] is adopted. This approach can effectively detect both abrupt and gradual scene changes using motion vectors, coding type and the DC value of each 8×8 macro block, with minimum decoding. In general, a motion vector points from a video block in one frame to a substantially similar or identical video block in another frame, providing an indication of displacement. As introduced in Chapter 2.1, there are two different coding types, which are *intra* and *inter* coding. The term *intra* coding means that no motion compensation is performed during compression and the image is compressed by a JPEG-like standard. The term *inter* coding includes the ability to use temporal redundancy to improve coding efficiency. The DC value is the DCT coefficient in the lowest frequency band. When video data is compressed, the frames are compressed as a unit for group-of-pictures (GOP). As shown in Fig. 5.3, within each GOP, the first frame is usually compressed as an I-frame. An I-frame is completely intra-coded, a P-frame is inter coded to perform motion compensation forward from the previous I- or P- frames, and in a B-frame, motion compensation is performed both forward and backward from the closest past and future I- or P-frames. According to the type of coding, scene change detection can be categorized into four different scenarios, which are

1 Scene change happens on an I-frame: In this scenario, the previous B-frame has very few backward motion vectors since the correlation between these two frames is low, and hence, the previous B-frame will favor forward motion compensation to decrease the energy of residual of motion compensation. This causes the ratio between number of forward to backward motion vectors , R_f, to be high. In addition, the absolute inter-frame variance difference $|\triangle\sigma^2|$ between current I-frame and its previous frame will be low, where frame variance σ^2 is obtained by estimating the variance of DC values of each 8×8 block. Based on these two characteristics, the scene boundary on the I-frame can be detected.

2 Scene change happens on a B-frame: When a scene change happens on B-frame, this frame will contain more backward motion vectors compared to the forward direction, because it has higher correlation with succeeding frames. Therefore, the ratio of the number of backward and forward motion vectors, R_b, will be high.

3 Scene change happens on a P-frame: A quantity, R_p, the ratio of the number of intra-coded blocks to inter-coded blocks in a P-frame, is used to determine if the scene boundary is located on a P-frame. Because the content of the current P-frame is very different from the content of the previous scene, the codec will prefer to use more intra-coding than inter-coding to increase compression efficiency. Thus, the R_p will be high when a P-frame happens to be the scene boundary.

4 Gradual scene change detection: This type of scene change does not have a clear difference between different scenes, it appears in a fading-in-then-out form. Thus, the coding type or number of different direction motion vectors are not sufficient to detect the scene boundary accurately. However, because of the scene fading-in-then-out characteristic, the scene change can be detected by observing a parabolic pattern of the curve of absolute frame variance $|\triangle\sigma^2|$.

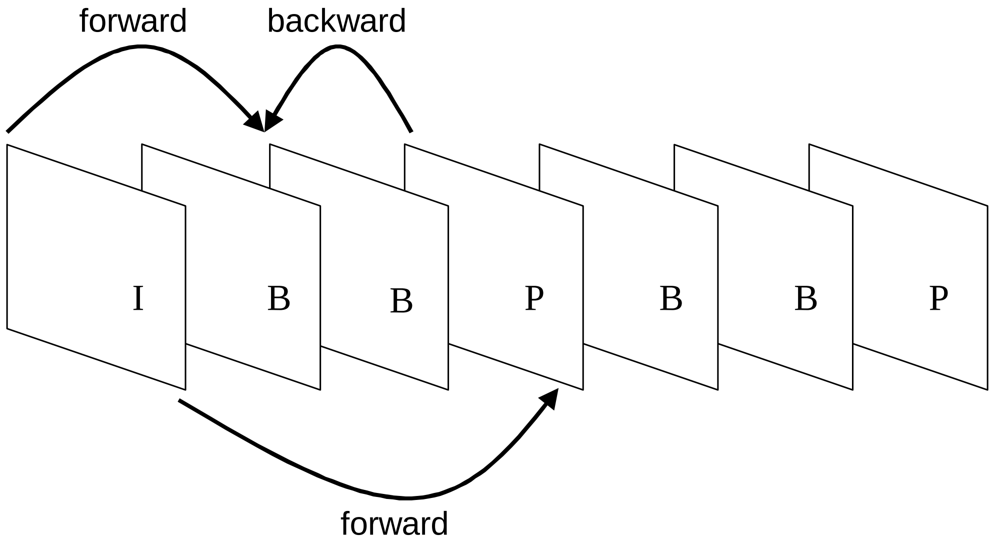


Figure 5.3: Structure of a GOP and the related direction of motion prediction of different types of inter frames.

5.4.3 Motion Activity Estimator

Along with the number of lost frames, motion activity is another important factor in temporal quality assessment. We determine the motion activity of each segment by averaging the magnitude of all eligible motion vectors within that scene; a motion vector indicates the displacement from a video block in one frame to a substantially similar or identical video block in another frame. Here, the motion vectors are obtained from the bitstream and the motion activity of the mth scene is

$$ma_m = min[\frac{1}{N \cdot N_{R,MB} \cdot N_{C,MB}} \sum_{n=1}^{N} \sum_{i'_{MB}=1}^{N_{R,MB}} \sum_{j'_{MB}=1}^{N_{C,MB}} MA(m, n, i'_{MB}, j'_{MB}), 10], \qquad (5.2)$$

where $ma_m \in (1, 10)$, N is the number of dropping occasions of each scene, $N_{R,MB}$ and $N_{C,MB}$ are the number of total eligible motion vectors in horizontal and vertical direction respectively, $MA(m, n, i'_{MB}, j'_{MB})$ is the magnitude of motion vectors in (i'_{MB}, j'_{MB})th block of nth frame in mth segment that satisfy $MA(m, n, i'_{MB}, j'_{MB}) > T_{ma}$, where $MA(m, n, i'_{MB}, j'_{MB})$ is given by

$$MA(v_{m,n,i}) = \sqrt{\mathrm{v}^2_{i'_{MB},j'_{MB},h} + \mathrm{v}^2_{i'_{MB},j'_{MB},v}}, \qquad (5.3)$$

and $(\mathrm{v}_{i'_{MB},j'_{MB},h}, \mathrm{v}_{i'_{MB},j'_{MB},v})$ depicts the motion vector at (i'_{MB}, j'_{MB})th block. The constant T_{ma} is a threshold to filter out un-noticeable motion vectors, and it is given by

$T_{ma} = \sqrt{1^2 + 1^2} = 1.414$; the value is derived from subjective observation. Moreover, some blocks are intra-coded (no motion compensation is performed during compression, the blocks are compressed by a JPEG-like standard) because the motion of those blocks is too large and the motion estimator can not find matched block within the search range. In that case, although those macro blocks have large displacement, the motion vector information is not available. Hence, a predefined motion activity, 22.62, is assigned to those blocks.

5.4.4 Motion Mapping

As reported in Refs. [50, 51, 85, 86], the same amount of dropping loss may introduce different temporal quality impacts because of different motion activity; higher motion activity usually results in larger temporal quality degradation. Therefore, the dropping severity $s_{m,n}$ should be mapped to $s'_{m,n}$ according to motion activity by

$$s'_{m,n} = s_{m,n}^{\alpha_T - (0.1 \cdot ma_m + \epsilon_T)}, \tag{5.4}$$

where constant α_T influences the slope and trend of the mapping function of each motion activity, and ϵ_T is the baseline of motion activity. The determination process of α_T and ϵ_T will be explained in Sec. 5.6.1.

Figure 5.4 presents the outputs of Equation (5.4), and each slice along the x-axis represents a mapping function corresponding to a motion activity - ma_m. As the motion activity increases, the slope and the non-linearity of the mapping function increases accordingly, which means the same $s_{m,n}$ is projected into higher $s'_{m,n}$. In other words, the same dropping severity is more noticeable for a high motion clip than for a low motion clip.

5.4.5 Temporal Fusion

With the considerations stated at Section 5.3, we adjust each motion mapped dropping severity $s'_{m,n}$ by the relationship of current dropping occasion to its neighboring frames within a scanning window. A *Temporal Quality Fluctuation (TQF)* module is designed to estimate the additional annoyance level of each dropping occasion caused by local temporal quality difference. Figure 5.5 shows the system diagram

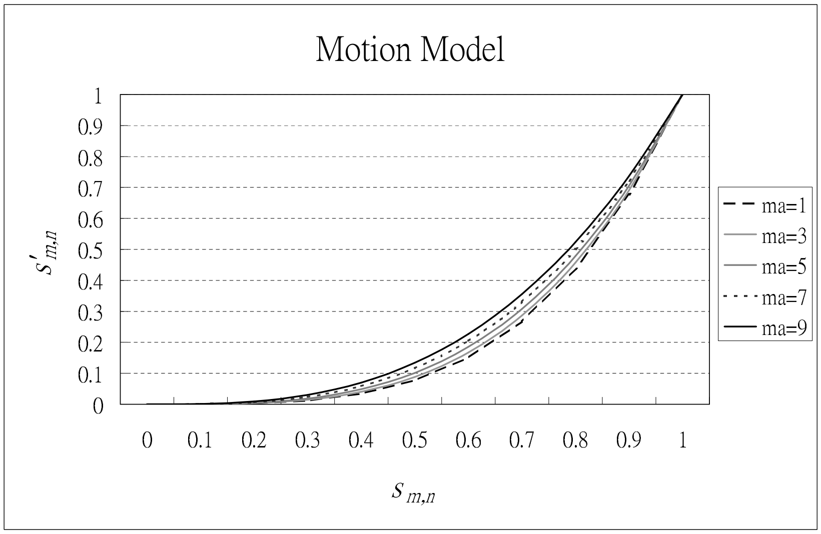


Figure 5.4: Motion model

of TQF. The quantity $s'_{m,n}$ is the input of the temporal fluctuation estimator, and the temporal fluctuation *tf'* is determined by

$$tf'_{m,n} = \frac{1}{UB(R)}[s'_{m,n} - \frac{1}{I}\sum_{i=1}^{I} s'_{m,n\pm i}]^{\beta}, \tag{5.5}$$

where R is equivalent to the applicable maximum frame rate, $UB(R)$ is the normalization function, I is the size of the scanning window (it is set to 3 here), β is a factor used to differentiate similar dropping occasions and it is set to 2. Since the range of fluctuation for each frame rate varies, we use $UB(R)$ to normalize *tf'* to $(0, 1)$, where the $UB(R) = [d(R) - \frac{1}{I}d(R)]^2$ and $d(R) = (30 - R)/29$. The scanning window direction could be backward or forward based on the availability of neighboring frames; because of human stronger working memory of previous dropping instances [87, 88], the default direction is backward (previous frames). But if not enough prior frames exist, such as a scene boundary, the missing frames will be compensated by scanning forward (future frames). Suggested by subjective data, human sensitivity to cases with different fluctuation of frame rate can be approximated by a logarithm function; the slope varies with frame rate and the magnitude varies with motion. Hence, a non-linear TQF

$$TQF(tf'_{m,n}) = \kappa\Big[1 - (1 - \frac{tf'_{m,n}}{\eta})^{\psi}\Big] \tag{5.6}$$

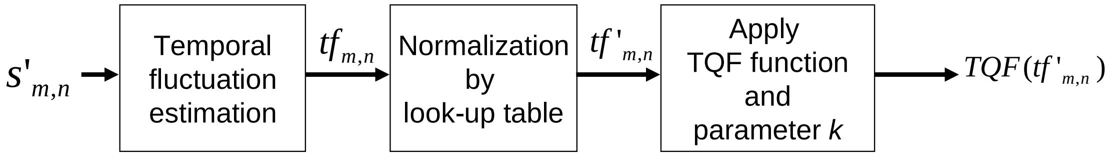

$s'_{m,n}$
Temporal fluctuation estimation
$tf_{m,n}$
Normalization by look-up table
$tf'_{m,n}$
Apply TQF function and parameter k
$TQF(tf'_{m,n})$

Figure 5.5: Procedure for estimating the impact of temporal quality fluctuation.

is applied on $tf'_{m,n}$, where

$$\psi = \begin{cases} 4 & \text{if } 20 \le R \le 30 \\ 8 & \text{if } 19 \le R \le 14 \\ 9 & \text{if } 1 \le R \le 13 \end{cases}, \tag{5.7}$$

constant η is used to fit the training data and it is assigned a value of 1.25, ψ influences the scaling nonlinearity of different fluctuation cases at different frame rate, κ balances the dominance between the effect of temporal fluctuation and the amount of consecutive frame loss; the value of κ is provided in Table 5.2. The procedure of obtaining parameters κ and ψ will be explained in Sec. 5.6.1. Output of Equation (5.6) functions as enlarging dropping severity, $s'_{m,n}$, obtained from Equation (5.4). Therefore, larger output of Equation (5.6) represents additional temporal quality degradation is introduced by local temporal quality difference. As shown in Table 5.2, κ decreases as frame rate decreases, which means the effect from quality fluctuation is less noticeable at low frame rates because the amount of frame loss is large enough to dominate perceived temporal quality. Figure 5.6 presents the outputs of TQF, and we can see that the linearity and maximum magnitude of TQF decrease as frame rate decreases. The curve is more linear at high frame rates, and becomes nonlinear at low frame rates. At low frame rates, because the length of each dropping occasion is large, even a small temporal quality contrast difference will result in a very noticeable quality change. Moreover, as the temporal quality contrast is larger than a certain threshold, human perceived quality will reach the quality baseline. Therefore, as frame rate decreases, TQF rises more quickly initially, but saturates earlier. Furthermore, since only limited perceptual quality can be achieved at low frame rates, even if there is no additional degradation from quality contrast, the TQF saturates at lower values for low frame rates than high frame rates.

After weighting by the TQF and normalizing, the temporal quality score of each

Table 5.2: Values of parameter κ in PTQM

	R∈[30,20)	R∈[20,10)	R∈[10,1]
Low motion	10	2.6	1.5
Medium motion	9.7	2.5	1.5
High motion	9.6	2.4	1.4

dropping occasion, $q_{m,n}$, is given by

$$q_{m,n} = \delta_T + \beta_T \cdot [1 - TQF(tf'_{m,n}) \cdot s'_{m,n}], \tag{5.8}$$

where δ_T, β_T denote the normalization factors to scale $q_{m,n} \in (1, 5)$, δ_T and β_T are set as 1 and 4 respectively. The scaling of $q_{m,n}$ is needed to match the MOS, and higher $q_{m,n}$ means better temporal quality. The temporal quality for each scene, q_m, is estimated by

$$q_m = \frac{1}{N} \sum_{n=1}^{N} q_{m,n}, \tag{5.9}$$

and the final temporal quality for whole sequence - Q is calculated by

$$Q = \frac{1}{M} \sum_{m=1}^{M} q_m, \tag{5.10}$$

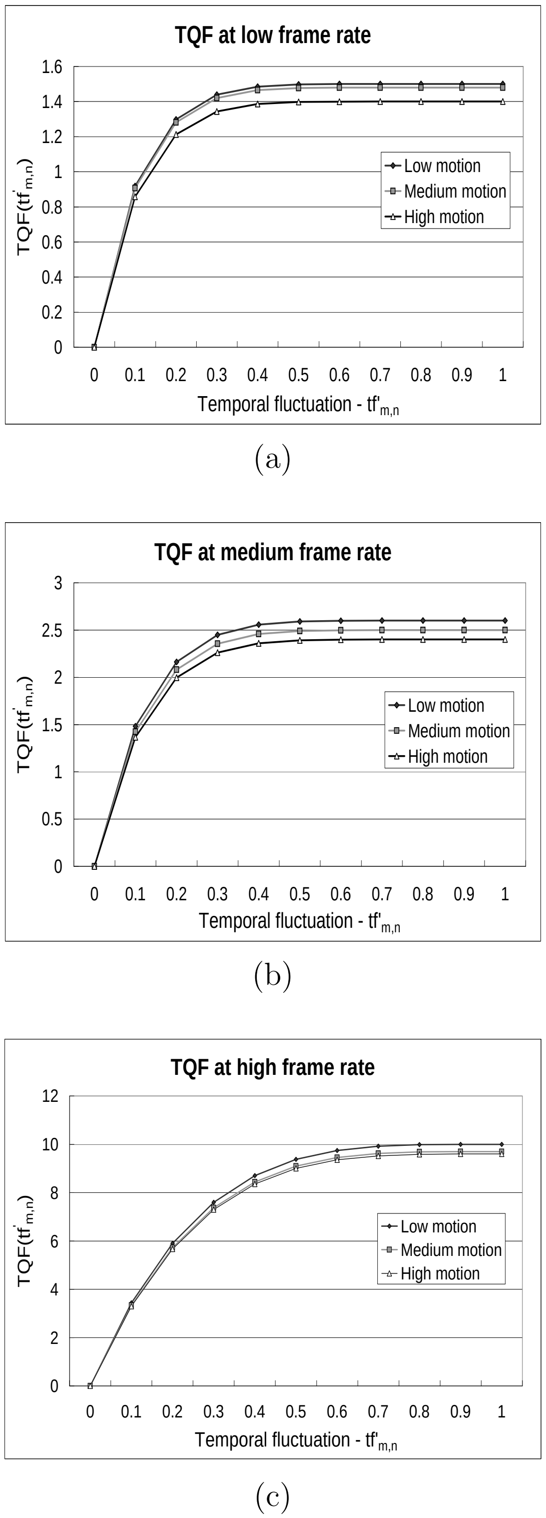
where M is the number of scenes in one sequence.

5.5 Experimental Set Up

Two different subjective tests have been conducted. The intention of the first test is to fit some important parameters as introduced in Sec. 5.6.1, while, in the second test, the subjective scores are used to evaluate the PTQM's performance in Sec. 5.6.2.

5.5.1 Testing Data

A total of six standard sequences from Ref. [76] are included, and these sequences can be classified into three groups based on their motion activity; low motion sequences: CONTAINER , MOTHER AND DAUGHTER, medium motion sequences: CARPHONE, HIGHWAY; and high motion sequences: RUGBY, FOOTBALL. More detail of each sequence can be found in Table 5.3. The original frame rate of all the



(a)

(b)

(c)

Figure 5.6: Output of TQF with (a) 10fps, (b) 15fps, and (c) 23fps.

sequences is 30fps and the duration of each sequence is 10 seconds as suggested by ITU-recommendation BT.500 [18]. The video sequences are sampled in YCbCr 4:2:0 with QCIF size(176×144). Among them, CONTAINER, HIGHWAY, and RUGBY are utilized for parameter fitting. All six sequences are used for metric performance evaluation.

Table 5.3: Description of testing sequences for performance evaluation of PTQM

CONTAINER	Still camera with slow moving ship
MOTHER AND DAUGHTER	Still camera with talking head
CARPHONE	Still camera with talking head and moving background
HIGHWAY	Camera in a moving vehicle
RUGBY	High motion sports
FOOTBALL	High motion sports

The testing cases are processed in two steps. First, the testing sequences are frame rate down sampled by dropping frames at a constant frequency. In order to simulate the temporal quality degradation without introducing any spatial distortion, testing sequences are not compressed. The lost frames are dropped artificially and replaced by duplicating the last frame before the frame loss occurs. In the second step, several sub-cases with different combinations of dropping occasion length and temporal location were generated for each frame rate and each sequence. The profile of all sub-cases can be denoted by the following formats:

1. $a - b$: There are a total of a dropping occasions within 1 second and each of them has b consecutive lost frames,

2. (a', b') : Two dropping occasions occur in 1 second and each of them has a' and b' consecutive dropping frames, respectively.

These notations will be used in Tables 5.4 and 5.5.

5.5.2 Experimental Methodology

These experiments were carried out using *Double-Stimulus Continuous Quality Scale(DSCQS)* [18] as described in Chapter 2.3. The sequences played at full frame rate (i.e., 30fps) serve as reference sequences. A pair of video sequences, comprised by one reference and one impaired video sequence with the same content, is shown twice for each testing session. The position of the reference sequence is changed in pseudo-random fashion. The video sequences are shown using a standard personal computer with Samsung 17' LCD displays. The lighting condition of the viewing environment and the viewing distance are adjusted to the testers' comfort. Viewers are asked to score the video sequences on a 1 to 5 scale, with the corresponding semantic meanings: Bad, Poor, Fair, Good, and Excellent. The final MOS data is the difference of MOS (DMOS) between reference and impaired sequences. The DMOS is converted into 1 to 5 and a higher DMOS indicates better temporal quality.

Two different groups of testers participated in the experiment. The first group had a total of eight examiners (all non-expert viewers). The DMOS data of this group is used for parameter fitting. Another group consisted of twenty examiners, which included twelve non-expert and eight expert viewers. This group was employed to evaluate the metric performance. In accordance with ITU-Recommendation BT.500 [18] , all DMOS data have been screened to remove the outliers and increase the data reliability.

5.6 Experimental Results

5.6.1 Parameter Fitting

The profiles of all test cases used for obtaining parameters are given in Table 5.4. Calculated by Equation (5.5), the low fluctuation cases have $tf' = 0.003$, and the medium and high fluctuation cases have $tf' = 0.25$ and 1 respectively.

Table 5.4: Profile of all cases at each frame rate for parameters determination of PTQM

	23fps	15fps	10fps
Low fluctuation	7-1	15-1	20-1
Medium fluctuation	(3,4)	(7,8)	(10,10)
High fluctuation	1-7	1-15	1-20

Figure 5.7 shows the subjective data. First observation shows that DMOS decreases as frame rate decreases. Also, with the same frame rate but different motion

activity, not surprisingly, lower motion sequences have higher DMOS than high motion sequences. Another important phenomenon is that for the same content sequence with the same frame rate and same motion, but different temporal quality fluctuation, the DMOS changes dramatically. These data provide strong evidence that the local temporal quality contrast must be considered when estimating temporal quality degradation. Furthermore, this set of subjective data is used for determining several parameters used in Equation (5.4) and (5.6).

Because $s_{m,n}$ from Equation (5.1) has a different range and opposite semantic meaning to DMOS, (i.e., higher $s_{m,n}$ represents worse temporal quality but DMOS works in the other way around), DMOS in Fig. 5.7 is converted using $s'' = 0.75 - DMOS/4$ to align the semantic expression with $s_{m,n}$. Equation parameters are determined using the following procedures.

Determination of α_T and ϵ_T

The parameters α_T and ϵ_T in Equation (5.4) account for the trend of the motion mapping function without considering temporal quality fluctuation. We define ${s''}_o$ to be the normalized DMOS with the lowest temporal quality fluctuation (test cases in the first row of Table 5.4) and the lowest motion activity of each testing sequence to get the best-fit parameters using a least square approach, and as a result, $\alpha_T = 10$ and $\epsilon_T = 7.7$.

Determination of ψ and κ

In Equation (5.6), ψ and κ regulate the nonlinearity and magnitude of TQF respectively. Regardless of the factor of motion, ψ can be determined by observing the relationship between each test case with a given frame rate. Since TQF amplifies the dropping severity according to different temporal quality fluctuations, Equation (5.6) is re-written as

$$\frac{TQF(tf')}{\kappa} = \frac{{s''}_\psi}{{s''}_o} = 1 - (1 - \frac{tf'}{\eta})^{\psi}, \tag{5.11}$$

where s''_{ψ} represents a s'' values of a training datum with a given tf' value and the same level of motion with s''_{o}. Taking the logarithm of both sides,

$$log(s''_{\psi}) - log(s''_{o}) = -\psi \cdot log(1 - \frac{tf'}{\eta}). \tag{5.12}$$

The optimal ψ for Equation (5.12), which provides minimum residual deviation between DMOS data and the right term of all test cases, is obtained using a recursive estimation algorithm. Since the nonlinear scaling behavior of TQF varies at different frame rates, a ψ is trained for a range of frame rates across all sequences with different motion level. Afterward, with the same frame rate, output of TQF not only varies with different tf', but also varies with different motion. This deviation is fine tuned by κ. Process of determining κ is similar to Equation (5.12), it is computed for each range of frame rates but different motion sequences as

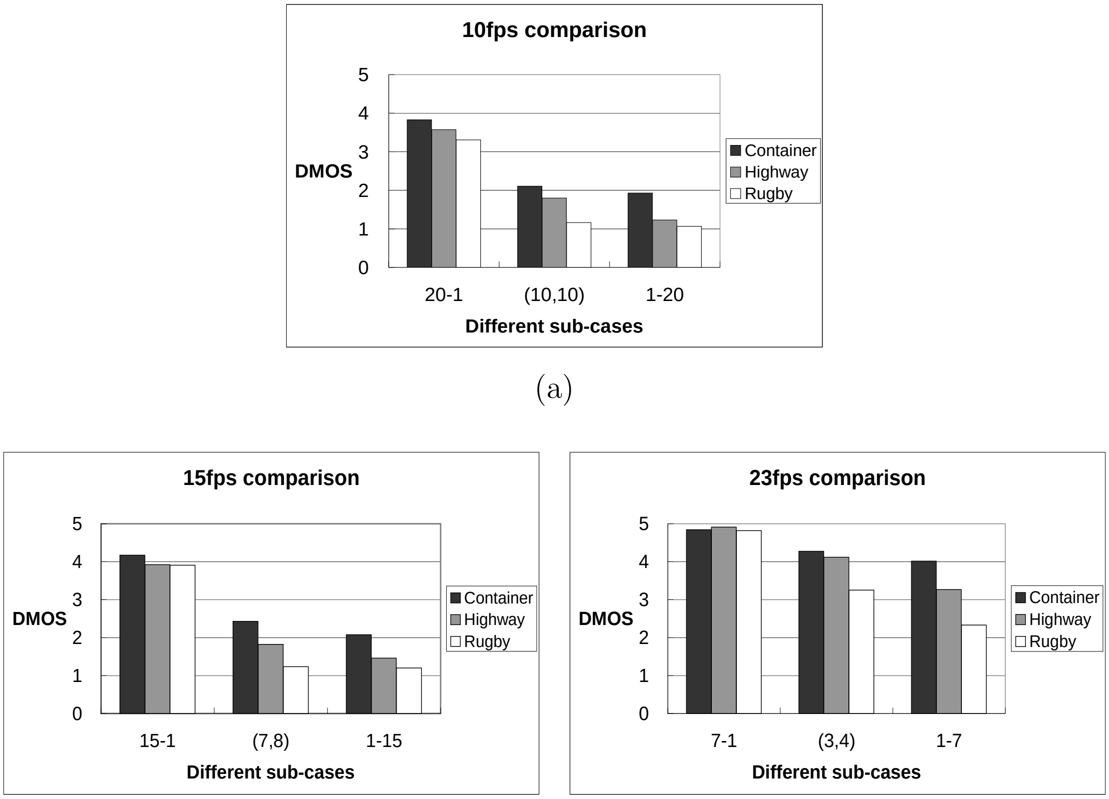
$$\frac{s''_{\kappa}}{s''_{o}} = \kappa \cdot [1 - (1 - \frac{tf'}{\eta})^{\psi}], \tag{5.13}$$

where s''_{κ} and s''_{o} belong to the same sub-case but with different motion activity, s''_{κ} represents a s'' value with corresponding tf' value but higher motion activity, and s''_{o} is similar to s''_{κ} but of the sequence with lowest motion activity. This training process is carried out through all different sub-cases across sequences with different level of motion within a range of frame rate. Thus, each range of frame rate has one κ value.

5.6.2 Metric Performance Analysis

Computational Complexity

There are three factors that can influence the computation load of the PTQM system: frame rate R, the total number of motion vectors of one frame P, and the number of dropping occasions, N. If quality is to be monitored continuously, then inevitably the processing load scales linearly with R. Thus it is not a useful indicator and will be ignored. Referring to Fig. 5.1, we will examine the complexity of each functional box, using the $\mathcal{O}$ notation. The Dropping Severity Estimator implemented in Equation (5.1) requires only differences between sequential received frame numbers, and so is $\mathcal{O}(1)$. The Motion Activity Estimator implements Equation (5.2).



(a)

(b)

(c)

Figure 5.7: Sample DMOS data for (a) 10fps, (b) 15fps, and (c) 23fps cases of frame loss with temporal quality fluctuation.

As P increases, the number of calculated motion vectors must increase, so there is linear dependence on P. Equation (5.2) also contains a sum whose limit is N. In its current form, the algorithm treats each frame as a potential dropping occasion, so N would be a stand-in for R. However, for optimized implementations, calculations might be triggered only for actual detected dropping, so a linear dependence on N can be included. Overall, the Motion Activity Estimator is $\mathcal{O}(PN)$. Scene Boundary Detection can rely on the same motion vectors developed for Motion Activity Estimation, so it is $\mathcal{O}(P)$. Motion Mapping uses output from the Motion Activity Estimation, but given that as input is itself $\mathcal{O}(1)$. The sum in Equation (5.5) for Temporal Fusion is over a fixed size window, so that is $\mathcal{O}(1)$. The normalization function $UB(R)$ does not change this. The Temporal Quality Function of Equation (5.6) is just a transformation of previously produced factors, and so is $\mathcal{O}(1)$.

Summarizing, the most computationally costly part of the PTQM system is the Motion Activity Estimator, which is $\mathcal{O}(PN)$. However, this is an operation that is common in video coding, and is not considered unreasonably expensive. The presence of N in the order notation is actually good news, indicating that as the frequency of dropping occasions decreases, the cost of this block tends toward zero, freeing resources for other uses.

Quality Prediction Accuracy

Testing cases in Table 5.5 are used for evaluating the performance of PTQM. Using several metrics from Chapter 2.3.3, the performance of PTQM is quantified by the Pearson and Spearman correlation coefficients, and Root-Mean-Square-Error (RMSE), denoted as C_P, C_S, and C_R, between the PTQM's output and DMOS data. Higher C_P, C_S and lower C_R indicate better metric performance.

Figure 5.8 compares the output of PTQM and DMOS data. Each mark represents a sub-case at different frame rate. Comparing PTQM against DMOS, we find that PTQM has very high linear correlation with DMOS data. Tables 5.6 through 5.8 show the quantified performance. On average, PTQM has C_P ranging from 0.92 to 0.97, and C_S ranges from 0.84 to 0.92. These high C_P and C_S validate the observation of high linear correlation in Fig. 5.8. The C_R ranges between 0.56 to 0.75. Since C_R

Table 5.5: Profile of testing cases for PTQM performance evaluation

23fps	15fps	10fps
7-1	15-1	10-2
(3, 2-2)	(6-2, 3)	7-3
(1-1, 5)	(3-4, 3)	5-4
(3, 4)	(2-3, 2, 7)	(10, 10)
(2, 4)	(7, 8)	(5, 15)
1-7	(4, 11)	(3, 12)
NA	1-15	1-20

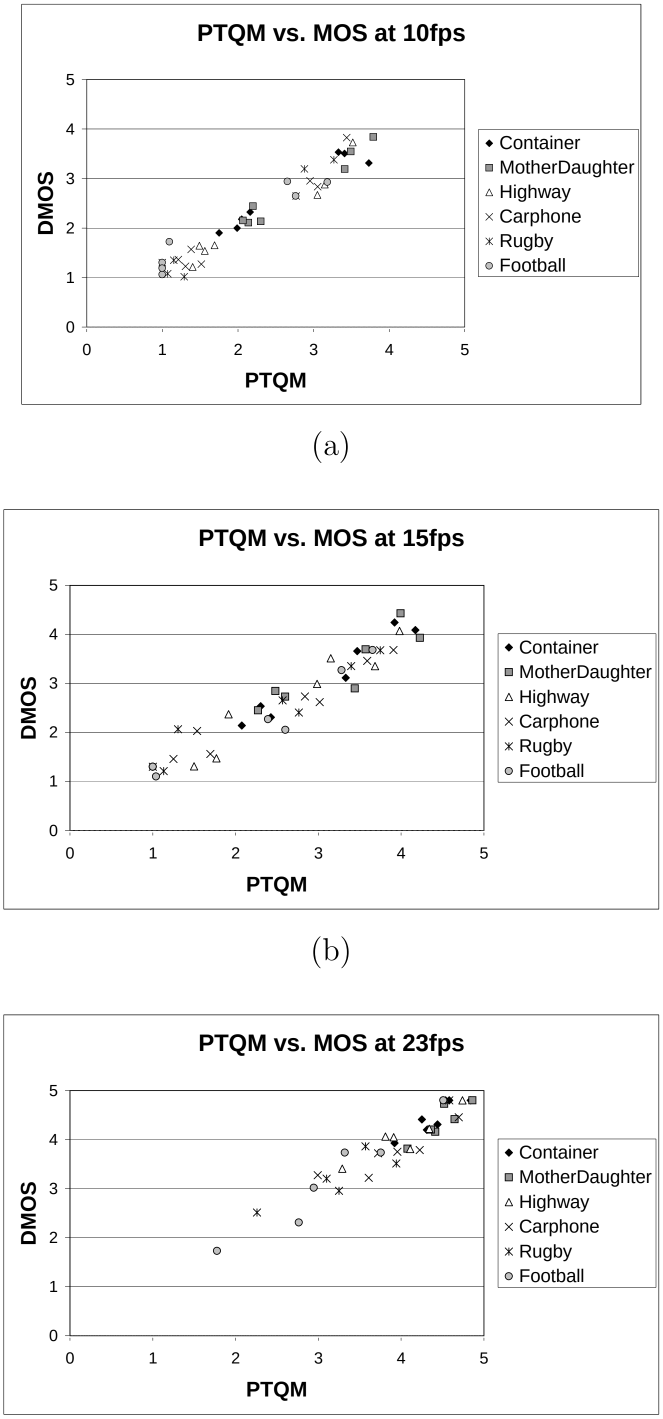
tends to estimate the absolute value difference while C_P and C_S estimate the trend similarity between two testing data sets, the C_P, C_S, and C_R should be interpreted jointly. Take for example, the case of RUGBY at 15fps and 10fps. Although it has higher C_R at 15fps, it still has very high correlation with $C_P = 0.96$ and $C_S = 0.93$ respectively. Another example is FOOTBALL at 15fps and 10fps. The case at 10fps has higher C_R than the one at 15fps, but both cases have a very similar C_P and C_S. Hence, it can be that the DMOS and PTQM fit very well and the C_R difference is marginal. Overall, the quantified correlation shows high correspondence between the objective and subjective data. We can summarize that PTQM is able to predict human perceived temporal quality accurately.

Table 5.6: Performance parameter for PTQM at 23fps

23fps	C_P	C_S	C_R
Container	0.91	0.81	0.32
Mother Daughter	0.89	0.89	0.51
Highway	0.91	0.77	0.45
Carphone	0.90	0.94	0.72
Rugby	0.92	0.89	0.69
Football	0.97	0.94	0.69
Average	0.92	0.87	0.56

5.7 Summary

A novel and reliable objective temporal quality metric - PTQM has been proposed. It considers the amount of frame loss, object motion, and local temporal quality contrast. Unlike conventional approaches, this metric produces not just sequence, but



(a)

(b)

(c)

Figure 5.8: Comparison of the PTQM's output and DMOS data at (a) 10fps, (b) 15fps, and (c) 23fps.

Table 5.7: Performance parameter for PTQM at 15fps

15fps	C_P	C_S	C_R
Container	0.97	0.93	0.51
Mother Daughter	0.89	0.93	0.88
Highway	0.96	0.96	0.76
Carphone	0.97	0.93	0.74
Rugby	0.96	0.93	0.90
Football	0.97	0.85	0.71
Average	0.95	0.92	0.75

Table 5.8: Performance parameter for PTQM at 10fps

10fps	C_P	C_S	C_R
Container	0.97	0.86	0.54
Mother Daughter	0.98	0.82	0.39
Highway	0.97	0.96	0.57
Carphone	0.98	0.79	0.57
Rugby	0.98	0.75	0.57
Football	0.96	0.85	0.83
Average	0.97	0.84	0.58

also scene and even frame level temporal quality measurement. This hierarchical temporal quality assessment is achieved by treating each frame as a potential frame loss occasion. It provides more freedom for integrating this metric with other applications in the future, and more insight into temporal quality analysis. Also, since motion is essential and content dependent for temporal quality assessment, the motion mapping mechanism has been improved by taking scene change boundary into account. The core of this work is that the PTQM can precisely estimate the temporal quality degradation caused by both regular and irregular type of frame loss by calculating the quality of each frame using temporal quality contrast and amount of frame loss. The subjective experiment shows high temporal quality prediction accuracy between the output of PTQM and subjective rating.

As a future plan, PTQM can be combined with spatial quality metrics to output a spatial-temporal quality score. This metric can serve as a guidance for designing several temporal quality enhancement algorithms, such as smart frame skipping and frame interpolation techniques, to improve the perceptual temporal quality while also controlling resource consumption.

PUBLICATIONS

Kai-Chieh Yang, Gokce Dane, and Khaled El-Maleh, “Temporal Quality Evaluation for Enhancing Compressed Video”, to appear at *Proc. IEEE 16th International Conference on Computer Communications and Network*, pp. 1160 - 1165, Aug., 2007

Kai-Chieh Yang, Clark C. Guest, Khaled El-Maleh and Pankaj K. Das, “Perceptual Temporal Quality Metric for Compressed Video”, *IEEE Transactions on Multimedia*, Volume 9, Issue 7, pp. 1528 - 1535, Nov. 2007

Kai-Chieh Yang, Clark C. Guest, Khaled El-Maleh and Pankaj K. Das, “Perceptual Temporal Quality Metric for Compressed Video”, 3rd International Workshop on *Video Processing and Quality Metric for Consumer Electronics*, Jan. 2007

6

Perceptual Frame Interpolation Quality Metric

6.1 Introduction

In many video applications such as broadcasting and high definition TV, motion compensated frame interpolation (MCFI) is often adopted at the decoder to improve temporal video quality by increasing the frame rate. By doing this, motion jerkiness and jitter induced by low frame rates can be effectively removed. In MCFI, the missing frames are interpolated using the received motion vector field (MVF) between two temporally adjacent reconstructed frames, denoted by f_{n-1} and f_{n+1} respectively. Based on the assumption of smooth motion trajectory, the (i, j)th pixel in the missing frame f_n can be represented as follows:

$$f_n(i,j) \;\; = \;\; \frac{1}{2} \cdot f_{n-1}(i + \frac{1}{2}v_h, j + \frac{1}{2}v_v) + \frac{1}{2} \cdot f_{n+1}(i - \frac{1}{2}v_h, j - \frac{1}{2}v_v), \qquad (6.1)$$

where $(\mathrm{v}_h, \mathrm{v}_v)$ is the received MVF in the bitstream used to reconstruct the frame f_{n+1}. Instead of using forward and backward predictions on the motion trajectory, this interpolation scheme takes bidirectional predictions using the received MVF divided by 2 to avoid missing part and overlap problems during frame interpolation. This method is also called the *direct* MCFI as it assumes that the received motion vectors (MVs) represent true motion and can be used directly. However, the received MVF is often estimated using a block matching algorithm to maximize coding efficiency, rather than finding true motion. As a result, by averaging bidirectional predictions for the interpolated frame in Equation (6.1), visual artifacts such as blocking and ghost artifacts [71] can be easily observed when unreliable MVs are used.

To solve this problem, several MV processing techniques at the decoder have been proposed to obtain a better MVF for MCFI. Using the assumption of a smooth MVF, a vector median filter is generally employed to remove MV outliers to obtain a smoother MVF for the interpolated frame. In [81], an adaptively weighted vector median filter exploiting prediction residues is presented. The work in [82] proposed using a finer MVF for frame interpolation to eliminate blocking artifacts. That is, each received MV is resampled into four MVs with smaller block sizes using a smoothness measurement. In order to reduce ghost artifacts caused by mismatched bidirectional predictions, the method in [83] adopts bidirectional MV processing for MCFI. Instead of using high complexity motion re-estimation at the decoder, the authors in [83] proposed selecting the best MV for each merged group from the neighboring MVs based on minimizing the difference between forward and backward motion compensations. They further proposed a multi-stage MV processing algorithm in [2], which corrects unreliable MVs by gradually reducing block sizes until all ghost artifacts and blocking artifacts can be removed effectively.

All MCFI techniques assume that temporal quality of the compressed video sequences improved by MCFI can be perfectly restored to before-compression levels. However, spatial quality of interpolated images could be severely degraded. Improving this has became a major challenge to recent MCFI research. Therefore, an accurate and appropriate quality assessment scheme for interpolated video data is essential to understand the performance of different MCFI techniques.

6.2 Related Works

Subjective evaluation [81, 82] is the most convincing approach because it collects direct responses from end users. However, it is inconvenient, expensive, and time consuming. Objective methods provide an alternate feasible solution. Most research evaluates interpolated frame quality using fidelity metrics. Reference [83] uses Peak-Signal-to-Noise-Ratio (PSNR), a normalized Mean-Square-Error (MSE) between original and processed images, to measure the interpolation quality. Reference [89] measures the quality using a Structure Similarity (SSIM) metric from [30]. SSIM uses a combination of luminance, contrast, and pixel value correlation comparisons as the quality index. However, fidelity-only measurements could fail for the following reasons:

1 *Low resolution to supra-threshold distortion*: Video quality degradation can be separated into sub-, near-, and supra-threshold distortions according to its perceptibility to human vision [51]. The sub- and near-threshold classes refer to the types of distortion that are below or slightly above just-noticeable-difference (JND) respectively. Supra-threshold distortion generally appears in a structured form and is known as *artifacts*. Blockiness and ghost artifacts shown in Fig. 6.1(a) and (b) are known to be two major artifacts in interpolated frames. These types of distortion are very irritating to human perception and dominate subjective quality judgment. One of the main challenges to supra-threshold distortion measurement is that its appearance varies with video content, and hence, human perceived annoyance is different even with the same error energy [70]. Fidelity metrics are good for estimating the near-threshold quality distortion, but are not sufficient to cover supra-threshold distortion.

2 *Various sensitivities to different spatial-temporal locations*: Visual attention-guided quality measurement has become an important direction for video quality research [51, 90, 91]. This type of approach improves quality prediction accuracy by considering sensitivities to different spatial-temporal regions according to human visual attention. This phenomena is especially important in interpolated frames because the regions with high motion usually suffer severe quality degradation, and humans also tend to pay more attention to moving regions. Hence, rather than evenly distributed weights, higher weights should be assigned to these regions when pooling the local spatial quality measurement.

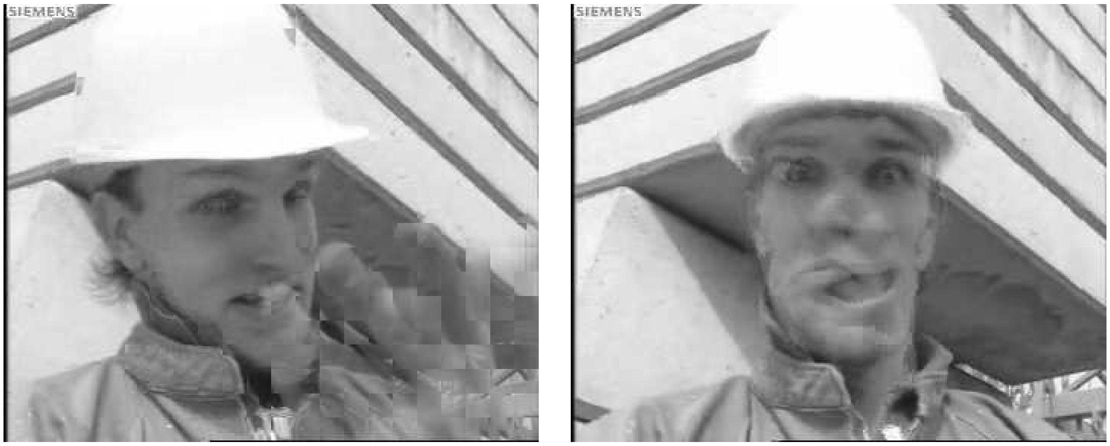
3 *Pixel shift*: Some MCFI schemes excel in producing good quality for moving regions, but the side-effect is that some pixels in the static regions may be

slightly changed. This pixel shift can be considered as sub-threshold distortion. It is barely noticed by human eyes and the perceived quality is good. However, fidelity metrics usually yield a low quality score since they still count it as part of the distortion. Visual observation shows that Fig. 6.2(a) and (b) are the same quality, but PSNR gives 3dB higher score to Fig. 6.2(a).

4 *Conspicuous local distortion*: In general, quality degradation caused by compression usually evenly distributes through entire frame since the compression ratios of all compression units (i.e. macroblocks) are similar. However, the quality degradation introduced by frame interpolation may highly concentrate in a small region from using unreliable MVs. Areas with this type of distortion might be small and conventional distortion quantification is low. But its large difference in quality to neighboring regions makes it very conspicuous and dramatically enlarges its annoyance to human perception. Most existing quality evaluation methods consider distortion perceptibility only from psychovisual (i.e. texture or temporal masking) or visual attention aspects. None of them take into account the salience of high spatial quality contrast; hence, the severity of local aggregated distortion can be incorrectly determined. In this case, the impact from regions with low quality scores is smeared out by frame level evenly weighted averaging. Figure 6.3(a) and Fig. 6.3(b) have very similar quality except a noticeable distortion on the edge of the helmet in Fig. 6.3(a). However, PSNR contradictorily assigns a lower quality value to Fig. 6.3(b). This provides strong evidence that annoyance caused by local high quality contrast must be considered.

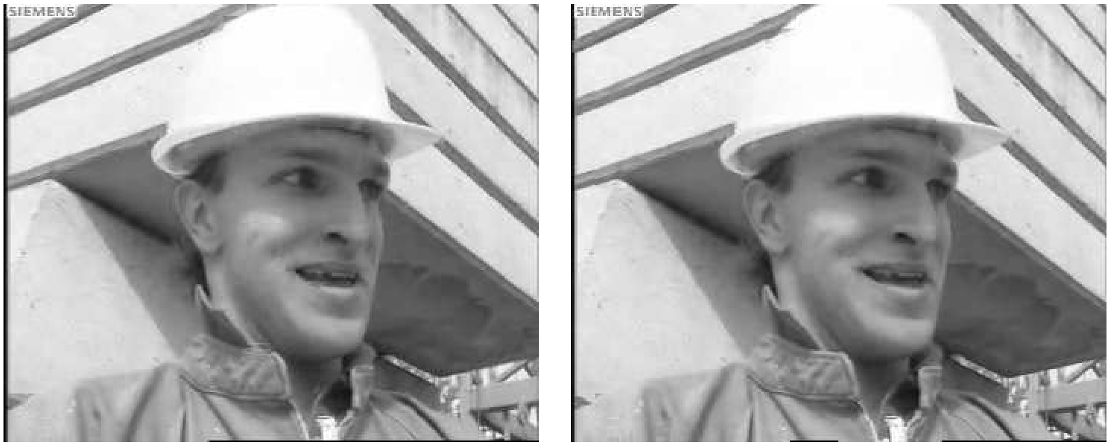
In order to overcome the problems described above, a novel metric - *Perceptual Frame Interpolation Quality Metric (PFIQM)* is proposed to evaluate the spatial quality degradation introduced by frame interpolation. The focus of this metric is to assess the spatial quality of interpolated frame, while temporal quality and motion smoothness of interpolated content are assumed perfect and are not considered here.

This chapter is organized as follows. Section 6.3 describes the PFIQM in detail. Section 6.4 evaluates the performance of PFIQM and other metrics by comparing the objective scores to subjective rating. A summary of this work is given in Section 6.5.



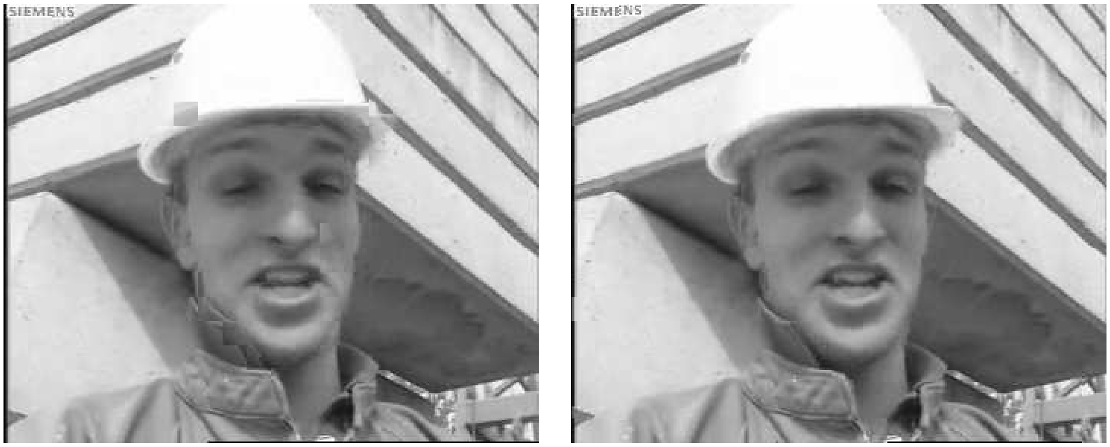
(a) (b)

Figure 6.1: Example of (a) blocking and (b) ghost artifacts introduced by frame interpolation.



(a) (b)

Figure 6.2: Examples of interpolated frames with pixel shift, which are produced by (a) direct MCFI approach and (b) multi-stage method, with the same subjective quality but PSNR = 31.42dB and 28.38dB respectively.



(a) (b)

Figure 6.3: Examples of strong local distortion that are produced by (a) direct MCFI approach with PSNR = 29.63dB and (b) multi-stage with PSNR = 28.06dB.

6.3 Proposed Metric

Figure 6.4 presents the system diagram of PFIQM. Denote B, and C as reconstructed, and interpolated frames respectively. Assume that Total Frames = $B \cup C$, and $B \cap C = \emptyset$. Original, reconstructed, and interpolated frames are used as inputs. This metric can be separated into two principle parts, Global Quality Estimator (GQE) and Local Distortion Estimator (LDE). The former emulates visual attention effects and estimates the frame-wide distributed distortion, and the later covers locally aggregated quality distortion while considering spatial quality contrast. A blockiness metric is used to estimate the amount of blocking artifacts - $\mathcal{B}$, whereas SSIM is used for estimating the severity of ghost artifact denoted as $\mathcal{S}$. In the GQM, both metrics' outputs are adjusted by a Conspicuousness Map ($\mathcal{CM}$) based on motion and gaze centering effects. Adjusted metrics' outputs, $Q_{B,G}$ and $Q_{S,G}$ for blockiness and ghost artifacts, are normalized by neighboring reconstructed frames' quality values and form the normalized quality scores - $Q_{B,G,norm}$ and $Q_{S,G,norm}$. Subsequently, these normalized quality scores are integrated into a Global Quality (GQ) value. For LDE, the metrics' outputs are adjusted by a Distortion Salience Map (DSM) and produce the quality score of blockiness and ghost aspects for local quality distortion measurement - $D_{B,G}$ and $D_{S,G}$. Afterward, these distortion measurements are normalized into $D_{B,G,norm}$ and $D_{S,G,norm}$. Subsequently, these normalized local quality

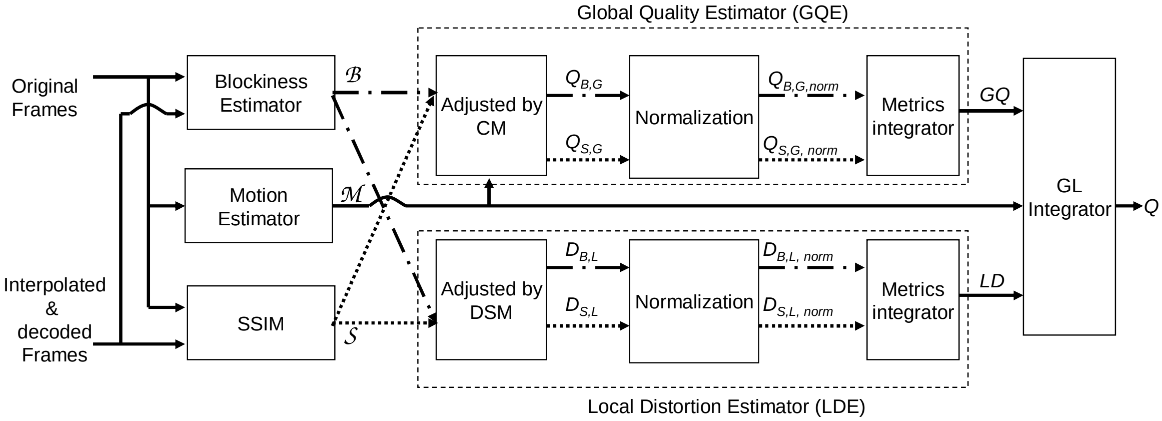


Figure 6.4: System diagram of PFIQM.

distortion measurements are combined into a Local Distortion (LD) value. Finally, both GQ and LD values are integrated with the guidance of motion to form a final quality score: Q.

6.3.1 Blockiness Estimator

A well known blockiness metric from [44] is adopted to measure the amount of blocking artifacts. Quality metrics can be categorized into Full-, Reduced-, and Non-Reference (FR, RR, and NR respectively) based on the accessibility of original video data [21, 22]. This blockiness metric originally was a NR metric, but it has been modified to be FR, in order to ensure high blockiness estimation accuracy. This metric calculates the pixel value discontinuity at each 8×8 boundary of both original and processed frames. Then the difference of the boundary discontinuity between original and processed frames is used as a boundary discontinuity measurement. Because blocking artifacts cannot be recognized in very dark or bright lighting conditions [21], the discontinuity is weighted by a luminance masking function. The adjusted pixel discontinuity is normalized by the inter-block pixel difference to emulate the texture masking phenomenon [22].

Let the original and processed images be f_x and f_y respectively, (i, j) is the index of a pixel and $i = 1, 2 \cdots, N_R$ and $j = 1, 2 \cdots, N_C$ respectively, where N_R and N_C represent the height and width of a frame, and (i', j') is the index of an 8×8 block, where $i' = 1, 2, \cdots, N_R/8-1$ and $j' = 1, 2, \cdots, N_C/8-1$. The pixel discontinuity on the vertical block boundary, B_v, is estimated by

$$B_v(i, 8 \times j') = \|w(i, 8 \times j') \cdot \Delta f(i, 8 \times j')\|^2, \tag{6.2}$$

where $\Delta f(i, 8\times j') = |f_y(i, , 8\times j') - f_y(i, , 8\times j'+1)| - |f_x(i, , 8\times j') - f_x(i, , 8\times j'+1)|$, $\|\cdot\|$ is the l_2 norm, and $w(i, 8\times j')$ is the output of a luminance masking function used to adjust the perceptual importance of each boundary discontinuity. The luminance masking function is defined as

$$w(i, 8 \times j') = \begin{cases} \tau ln(1 + \frac{\sqrt{\mu(i,8\times j')}}{1+\sigma(i,8\times j')}) & \text{if } \mu(i, 8 \times j') \leq \zeta \\ \tau ln(1 + \frac{\sqrt{255-\mu(i,8\times j')}}{1+\sigma(i,8\times j')}) & \text{otherwise} \end{cases} \tag{6.3}$$

where

$$\tau = \frac{ln(1 + \sqrt{255 - \zeta})}{ln(1 + \sqrt{\zeta})}, \tag{6.4}$$

and ζ represents the most suitable lighting condition in an 8-bit scale for human visual perception, which is set as 81. Parameters $\mu(i, 8 \times j')$ and $\sigma(i, 8 \times j')$ are the mean value and standard deviation of the pixels on the same row within two adjacent blocks respectively, and are given by

$$\mu(i, 8 \times j') = \frac{1}{16} \sum_{q=-7}^{8} f_x(i, 8 \times j' + q), \tag{6.5}$$

and

$$\sigma(i, 8 \times j') = \sqrt{\frac{1}{16} \sum_{q=-7}^{8} [f_x(i, 8 \times j' + q) - \mu(i, 8 \times j')]^2}. \tag{6.6}$$

The final vertical blockiness map, B'_v, is obtained after normalizing the discontinuity with the average inter-pixel difference of the non-boundary pixels as

$$B'_v(i, 8 \times j') = \frac{7 \cdot B_v(i, 8 \times j')}{\sum_{q=1}^{7} [\sum_{j'=1}^{N_C/8-1} \Psi(i, 8 \times j' + q)]^{0.5}}, \tag{6.7}$$

where

$$\Psi(i, 8 \times j' + q) = \|w(i, 8 \times j') \cdot [f_x(i, 8 \times j + q) - f_x(i, 8 \times j' + q + 1)]\|^2. \tag{6.8}$$

The horizontal blockiness map, B'_h, can be obtained with the same process as for the vertical blockiness map.

The vertical and horizontal boundary based blockiness map is transformed to a block basis blockiness map by

$$\mathcal{B}(i',j') = \frac{1}{16}\sum_{q=1}^{8} B'_h(8i', 8j'-8+q) + B'_v(8i'-8+q, 8j'). \tag{6.9}$$

Higher $\mathcal{B}$ value implies more blocking artifacts.

6.3.2 Similarity Estimator

SSIM is used to estimate the severity of ghost artifacts by measuring the deviation of pixel values and the distribution of original and processed images. First, both the original and processed images are low-pass filtered by a Gaussian filter with a 11 × 11 window and variance σ_W^2. The intention of the low-pass filtering is to remove any distortion imperceivable to human eyes. The structural similarity,ss , of (i,j)th pixel is estimated by

$$ss(i,j) = \frac{[2\mu_x(i,j)\mu_y(i,j) + c_{ss}(1)]\cdot[2\sigma_{xy}(i,j) + c_{ss}(2)]}{[\mu_x^2(i,j) + \mu_y^2(i,j) + c_{ss}(1)]\cdot[\sigma_x^2(i,j) + \sigma_y^2(i,j) + c_{ss}(2)]}, \tag{6.10}$$

where $\sigma_{xy}(i,j)$, $\mu_x(i,j)$ and $\mu_y(i,j)$, and $\sigma_x(i,j)$ and $\sigma_y(i,j)$ are the correlation, mean, and standard deviations of each 11 × 11 window around the (i,j)th pixel in f_x and f_y respectively, and $c_{ss}(1)$ and $c_{ss}(2)$ are determined experimentally by [30], which are given by $c_{ss}(1) = 6.5$ and $c_{ss}(2) = 58.52$ respectively. The similarity map is further processed into a block basis by

$$\mathcal{S}(i',j') = \frac{1}{64}\sum_{q=1}^{8}\sum_{q'=1}^{8} ss[8(i'-1)+q, 8(j'-1)+q']. \tag{6.11}$$

Higher S value implies fewer ghost artifact and better quality.

6.3.3 Global Quality Estimator (GQE)

To implement the GQE, the blockiness and structure distortion values are first weighted by a conspicuousness map (CM). Next, the weighted metrics' outputs are normalized and pooled together to yield frame level blockiness and ghost artifact measurements. Finally, these two quality scores are integrated to form a Global Quality (GQ) score.

Conspicuousness Map

Humans are interested in moving objects, and in addition, quality of moving regions is usually fragile during frame interpolation. Moreover, because of human eyes' biological structure, the central part of an image usually draws most of the human attention [64]. Hence, visual sensitivity to distortion decreases as spatial location moves away from central area toward the boundary of an image. This is applicable across different content types and camera view. Based on these reasons, a conspicuousness map is determined by both the motion and gaze centering maps.

Assuming that the bitstream information is not accessible, the MVs are obtained by processing a block-matching motion estimation [92] on the original video data. Let $(\mathrm{v}_{i',j',h}, \mathrm{v}_{i',j',v})$ be the motion vectors of (i', j')th block in the horizontal and vertical directions. The corresponding motion activity is

$$MA(i', j') = \sqrt{\mathrm{v}^2_{i',j',h} + \mathrm{v}^2_{i',j',v}}, \tag{6.12}$$

which is normalized by the max and min motion activity value within each frame by

$$MA'(i', j') = \frac{MA(i', j') - \min(MA)}{\max(MA) - \min(MA)}. \tag{6.13}$$

Finally, the motion map, $\mathcal{M}$, is obtained after $MA'(i', j')$ is post-processed by a spatial median filter as

$$\mathcal{M}(i', j') = \mathrm{median}_1[MA'(i'_w, j'_w)], \tag{6.14}$$

where median_1 denotes one iteration median filtering, $(i'_w, j'_w) \in w$ and w is a 3×3 block mask centered at (i', j'). On the other hand, for frames with very few motion such that $\sum_{i',j'} MA(i', j') = 0$, the quality score of all blocks are counted, and all $\mathcal{M}(i', j')$ are assigned to 1.

A two-dimensional anisotropic Gaussian kernel, P_2, is implemented to emulate the gaze centering phenomenon as

$$P_2(i', j') = \frac{1}{\sqrt{2\pi}(\sigma^2_h + \sigma^2_v)/2} e^{-\frac{1}{2}(\frac{(i'-i'_c)^2}{\sigma^2_h} + \frac{(j'-j'_c)^2}{\sigma^2_v})}, \tag{6.15}$$

where, (i'_c, j'_c) represents the index of the spatial central point of an image, and σ_h^2, σ_v^2 are the width of Gaussian distribution in horizontal and vertical directions, which are set to 800 and 500 respectively suggested by Ref. [64]. After normalization, the final central focus map, $\mathcal{C}$, is given by

$$\mathcal{C}(i', j') = \frac{P_2(i', j') - \min(P_2)}{\max(P_2) - \min(P_2)}. \tag{6.16}$$

Figure 6.5 shows the outputs of (6.16). Central regions are assigned higher values, which correspond to higher visual sensitivity.

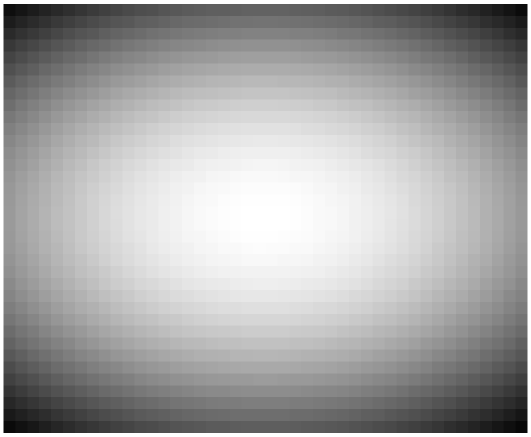


Figure 6.5: Value of the gaze centering map. Higher value represents stronger visual sensitivity and decreases as spatial location moves toward the boundary.

The final conspicuousness map, $\mathcal{CM}^+$, in (i', j')th block is given by

$$\mathcal{CM}^+(i', j') = \mathcal{C}(i', j') \cdot \mathcal{M}(i', j'). \tag{6.17}$$

Weighted by $\mathcal{CM}^+$, quality scores from both blockiness and SSIM of all blocks within a frame are averaged as

$$Q_{B,G} = \frac{1}{(\frac{N_C}{8} - 1)(\frac{N_R}{8} - 1)} [\sum_{i',j'} \mathcal{CM}^+(i', j') \cdot \mathcal{B}(i', j')], \tag{6.18}$$

and

$$Q_{S,G} = \frac{1}{(\frac{N_C}{8} - 1)(\frac{N_R}{8} - 1)} [\sum_{i',j'} \mathcal{CM}^+(i', j') \cdot \mathcal{S}(i', j')]. \tag{6.19}$$

Normalization

Appearance of supra-threshold distortion varies with video content, and hence, outputs from the blockiness metric and SSIM may occur in different ranges as video content changes. This will cause many ambiguities when integrating multiple metrics' outputs, and therefore, outputs from (6.18) and (6.19) must be normalized into a certain range. Another purpose of normalization is to align the semantic interpretation of metrics' output. Higher values of the blockiness metric suggests lower quality, but output from SSIM is interpreted the other way around. Hence, after normalization, higher values from both metrics means better quality.

Most frame interpolation techniques construct the interpolated frame by fetching part of the data from neighboring reconstructed frames. This implies that (a) the content of the interpolated frames is similar to its neighboring reconstructed frames, and (b) quality degradation due to compression in reconstructed frames may propagate to interpolated frames. Therefore, metrics' outputs from the nearest backward and forward reconstructed frames are employed as normalization baselines for interpolated frames. Consider X_{G} as $Q_{B,G}$ or $Q_{S,G}$ of the interpolated frames, the metrics' outputs are normalized by

$$X_{\mathrm{G,norm}} = \beta_{\mathrm{G,norm}}[X_{\mathrm{G}} - \frac{(\bar{X}_{\mathrm{dec}-1} + \bar{X}_{\mathrm{dec}+1})}{2}] + \Phi_{\mathrm{G,norm}}, \tag{6.20}$$

where $\bar{X}_{\mathrm{dec}-1}$ and $\bar{X}_{\mathrm{dec}+1}$ are the outputs from Equation (6.18) or (6.19) of the closest backward and forward reconstructed frames respectively, and $\beta_{\mathrm{G,norm}}$ and $\Phi_{\mathrm{G,norm}}$ are the normalization constants for global quality, which are obtained experimentally using several interpolated frames that contain global quality distortion only. Finally, the normalized value - $X_{\mathrm{G,norm}}$ is bounded by

$$X_{\mathrm{G,norm}} = \begin{cases} 1 & \text{if } X_{\mathrm{G,norm}} > 1 \\ 0 & \text{if } X_{\mathrm{G,norm}} < 0 \\ X_{\mathrm{G,norm}} & \text{Otherwise} \end{cases}, \tag{6.21}$$

to emulate the perceptual saturation phenomenon. Notations $Q_{B,G,norm}$ and $Q_{S,G,norm}$ represent the normalized values from both Equation (6.18) and (6.19) for global quality assessment, and a higher value means better quality.

Metrics Integrator

The final GQ is produced by averaging $Q_{B,G,norm}$ and $Q_{S,G,norm}$ as

$$GQ = \frac{Q_{B,G,norm} + Q_{S,G,norm}}{2}, \tag{6.22}$$

where higher GQ value implies better quality.

6.3.4 Local Distortion Estimator (LDE)

Similar to the GQE, both blockiness and structural distortions are weighted according to human visual sensitivity to estimate the quality level of a frame. Different to the GQE, LDE determines visual sensitivity from the local distortion contrast perspective instead of motion.

Distortion Salience Map

Since the appearance of strong local distortion varies, blockiness measurement might not be generic enough to cover all different local distortion cases. Therefore, outputs from Equation (6.11) are used to determine the noticeability of aggregated quality distortion by considering local distortion contrast as

$$\mathrm{DM}(i', j') = \max(|\mathcal{S}(i', j') - \mathcal{S}(i'_w, j'_w)|). \tag{6.23}$$

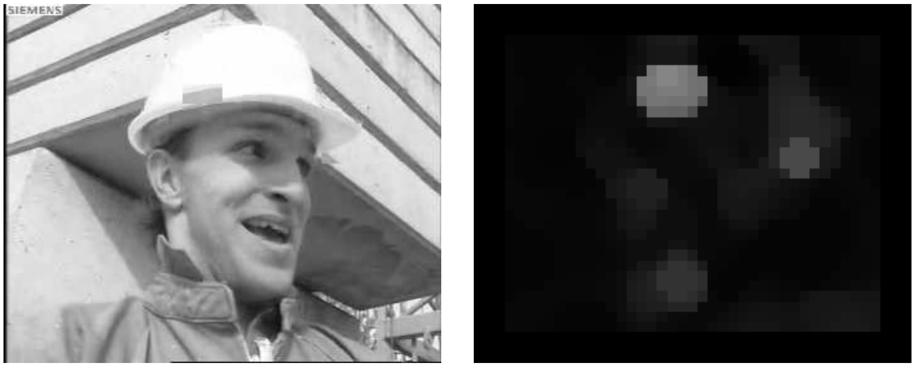
Post-processed by a three iteration median filtering and with the gaze centering map from Equation (6.16), the final DSM of (i', j')th block is

$$\mathcal{DSM}^+(i', j') = \mathrm{median}_3[\mathrm{DM}(i'_w, j'_w)] \cdot \mathcal{C}(i', j'). \tag{6.24}$$

Figure 6.6 shows an DSM extraction example. A conspicuous distortion is seen at the edges of the helmet, DSM successfully detects it and assigns high values to those regions.

Only the quality scores with strong DSM are considered in local distortion estimation. The local quality distortion from blockiness and structural similarity aspects, $D_{B,L}$ and $D_{S,L}$, are estimated as

$$D_{B,L} = \frac{1}{N_L}[\sum_{i'_L, j'_L} \mathcal{B}(i'_L, j'_L)], \tag{6.25}$$



(a) (b)

Figure 6.6: An example of DSM determination, where (a) a noticeable local distortion appears around the edge of the helmet, and (b) the corresponding DSM indicates a high value on those regions.

and

$$D_{S,L} = \frac{1}{N_L}[\sum_{i'_L, j'_L} \mathcal{S}(i'_L, j'_L)]. \tag{6.26}$$

where (i'_L, j'_L) and N_L are indices and total number of the blocks with $\mathcal{DSM}^+(i'_L, j'_L) \geq 0.5$.

Normalization

Consider X_L as either $D_{B,L}$ or $D_{S,L}$, outputs from Equation (6.25) and (6.26) are normalized by

$$X_\mathrm{L,norm} = \begin{cases} 0 & \text{if } N_L \leq 11 \\ \min(\beta_\mathrm{L,norm} X_\mathrm{L} + \Phi_\mathrm{L,norm}, 1), & \text{Otherwise} \end{cases}, \tag{6.27}$$

where $\beta_\mathrm{L,norm}$, $\Phi_\mathrm{L,norm}$ are the normalization constants for local distortion, and $X_\mathrm{L,norm}$ is bounded by 1. The N_L can be thought of as the area of strong local distortion. If N_L is less than 11, then the local distortion is considered as not-noticeable and $X_\mathrm{L,norm}$ is assigned to 0. Unlike GQ, higher normalized quality scores, $D_{B,L,norm}$ and $D_{S,L,norm}$, suggest more local distortion and *worse* quality.

Metrics Integrator

The final local distortion score is given by

$$LD = \frac{D_{B,L,norm} + D_{S,L,norm}}{2}. \tag{6.28}$$

6.3.5 Global and local quality integrator

Motion is used to determine the dominance between GQ and LD when forming a final quality score. High motion results in more artifacts, and in this case, GQ is more dominant than LD. Moreover, high motion introduces temporal masking and spatial detail will be filtered out; thus, LD can be discarded. Therefore, the final quality of nth interpolated frame, Q_n, is given by

$$Q_n = \begin{cases} GQ_n & \text{if } ma_n \geq th \\ GQ_n - \frac{LD_n}{ma_n}, & \text{if } ma_n < th \end{cases}, \tag{6.29}$$

where ma_n is the average motion activity of nth frame given by

$$ma_n = \frac{1}{(\frac{N_C}{8} - 1)(\frac{N_R}{8} - 1)} \sum_{i',j'} MA_n(i', j'),$$

and th is a threshold to trigger the influence of LD. In the case that motion activity is lower than th, LD is perceivable but its dominance is inversely proportioned to motion, which indicates that LD is less important as motion increases. Also, the negative sign of the LD term indicates the fact that local distortion is an additional distortion to global quality, so GQ is decreased by a weighted LD.

6.4 Experimental Confirmation

6.4.1 Experimental Setup

A subjective test has been carried out to collect viewers' perceived quality of frames interpolated by different MCFI schemes. Two video clips, FOREMAN [76] and WALK [93], were selected as test sequences because of their wide range of different motion activity and texture. The original sequences are CIF (352 × 288) frame resolution with an original frame rate of 30 fps. They are encoded using H.263, but even numbered frames are skipped to generate video bitstreams of 15 fps. The rate control function is disabled by fixing quantization parameter (QP) values at 10. The averaged bit rates of these two test sequences are 395.77 Kbps and 430.39 Kbps for FOREMAN and WALK, respectively. Fourteen interpolated frames are chosen as test materials based on the following criteria: Each selected frame has five different images produced by five different MCFI techniques - direct MCFI, vector median filter [81], MV smoothing method as described in [82], MV selection similar to [83] but with fixed block size, and the multi-stage MV processing method in [2] respectively. Therefore, a total of seventy interpolated images are included in test data set.

Table 6.1: Test material selective criteria for PFIQM performance evaluation

	Global Quality Degradation	Local Distortion
i	Strong	Weak
ii	Strong	Strong
iii	Weak	Weak
vi	Weak	Strong

The Double Stimulus Continuous Quality Scale (DSCQS) method [18] as described in Chapter 2.3 is adopted as the subjective test method. Original frames serve as reference data and the interpolated frames are used as test data. One reference and one test datum form a test case. The procedure for a test case is illustrated in Fig. 6.7, where T1 = 3 sec, showing either reference or test image data, T2 = 2 sec, showing a gray image for buffering. The DSCQS method first presents the reference, then the test data to participants. Subsequently, this image pair is repeated but in random order and participants vote on quality score. Total duration of a test case is 20 sec, so the entire test session lasts $20 \times 70 = 1400$ sec = 23.3 min. Half an hour has been proved as the most appropriate experimental length [18] for video subjective test to avoid tiring assessors and thus producing unreliable data.

The assessors are asked to grade their perceived quality on a continuous linear scale that ranges from 1 to 5 with semantic meaning of ”Bad”, ”Poor”, ”Fair”, ”Good”, and ”Excellent” to perceived quality. Viewers are allowed to vote with 0.1 increment. A total of ten viewers participated in the experiment composed of five expert and five non-expert viewers. The raw scores of each test case are converted to difference scores

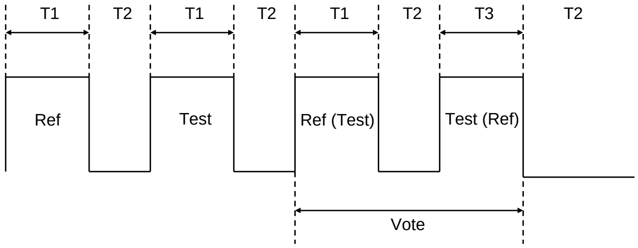


Figure 6.7: Subjective test procedure for a test case

(between the test and reference image) to obtain a subjective Difference Mean Opinion Score (DMOS) value for each interpolated image, which is denoted as $DMOS_s$ and higher value represents better quality.

Objective metrics involved in performance comparison are the PFIQM, PSNR, and SSIM. In this experiment, two outputs of PFIQM, Q from Equation (6.29) and GQ from Equation (6.22), are used as two different metrics to evaluate the quality prediction accuracy with and without considering local quality distortion. Since objective outputs are content dependent and also for the sake of data interpretation convenience, metrics' outputs are normalized by the max and min value of each sequence by

$$\mathcal{VQR}_n = 1 + 4\frac{VQR_n - \min(VQR)}{\max(VQR) - \min(VQR)} \tag{6.30}$$

where VQR_n denotes a metric's output of nth frame, and VQR is a set of a metric's outputs for a sequence. According to the Phase II Final Report from Video Quality Experts Group (VQEG) [56], the relationship between the metrics' outputs and the $DMOS_s$ may not be linear, as subjective testing can have nonlinear quality testing compression at the extremes of the test range. In order to remove any nonlinearity caused by subjective rating process and to facilitate comparison of metrics in a common analysis space, normalized metrics' outputs are mapped by a nonlinear regression function as

$$DMOS_{o,n} = \frac{b(1)}{1 + exp[-b(2) \times (\mathcal{VQR}_n - b(3))]} \tag{6.31}$$

where $DMOS_{o,n}$ denotes the mapped objective score for nth frame, and b is a set of parameters obtained by fitting the $\mathcal{VQR}$ of each metric against $DMOS_s$. As a result, each metric has a b parameters set and the corresponding $DMOS_o$ represents the objective scores that are closest to subjective ratings. The best performance of each metric can be obtained by this mapping process.

After normalization and nonlinear transformation, outputs of all metrics range from 1 to 5 and higher value means better quality. The $DMOS_o$ were compared with the $DMOS_s$ values by computing the correspondences using the following metrics:

(a) Pearson correlation coefficient (C_P): This metric is used to estimate the model prediction accuracy, which is the ability of the objective metric to predict subjective ratings with minimum average error,

$$C_P = \frac{\sum \Delta DMOS_{o,n} \cdot \Delta DMOS_{s,n}}{\sqrt{\sum \Delta DMOS_{o,n}^2 \cdot \Delta DMOS_{s,n}^2}}, \tag{6.32}$$

where $\Delta DMOS_{o,n} = DMOS_{o,n} - \overline{DMOS_o}$ and $\Delta DMOS_{s,n} = DMOS_{s,n} - \overline{DMOS_s}$, which $\overline{DMOS_o}$, $\overline{DMOS_s}$ are the mean values of mapped objective and subjective scores respectively. Larger C_P means higher prediction accuracy.

(b) Spearman rank order correlation coefficient (C_S): This coefficient is designed to determine the level of monotonicity by measuring the correlation of decreasing(increasing) trend of both variables independently of the magnitude. The equation is

$$C_S = 1 - 6\sum \frac{(DMOS_{o,n} - DMOS_{s,n})^2}{N(N^2 - 1)}, \tag{6.33}$$

where N is the number of data point. Larger C_S means better prediction performance.

(c) Root-Mean-Square-Error (C_R): Root-Mean-Square-Error (RMSE) is the square root of the mean squared difference between objective and subjective values, which is

$$C_R = \sqrt{\sum_n (DMOS_{o,n} - DMOS_{s,n})^2}. \tag{6.34}$$

Lower C_R means less deviation between subjective and objective data and better prediction performance.

6.4.2 Experimental Results

The data for the $DMOS_o$ vs. $DMOS_s$ comparison are arranged into two groups, where the first group contains all test cases and the second group only includes the test cases with local distortion (i.e. test cases ii and vi in Table 6.1).

Figures 6.8, and 6.9, and Table 6.2 present scatter plots and quantitative correspondence measurement of these two groups of data. In the analysis of all test cases, both Q and GQ have much higher correlation and lower RMSE than PSNR and SSIM. Hence, the PFIQM significantly outperforms both PSNR and SSIM, the two commonly used metrics in MCFI research. SSIM is better than PSNR since it has better capability to detect structural distortion. However, it is still worse than the PFIQM since it does not consider blocking artifacts. Detailed comparison shows that Q has slightly better performance than GQ, since GQ does not include local distortion.

In the performance analysis of test cases ii and vi only, all metrics' performance drop dramatically but Q decreases least and still maintains very good performance compared to other metrics. Since GQ focuses on detecting globally spread distortion, its performance is much worse than Q. However, since GQ takes blockiness and visual attention factors into account, it still performs better than PSNR and SSIM. It is worth noting that the performance decrease rate of both PSNR and SSIM is large and similar. According to Fig. 6.9(c) and (d), both PSNR and SSIM are very insensitive to quality degradation due to local distortion. It is fair to say that these two metrics fail to handle this type of quality impairment.

Table 6.2: Quantitative performance comparison for PFIQM, PSNR, and SSIM

	All cases			Only ii and vi		
	C_P	C_S	C_R	C_P	C_S	C_R
PFIQM (Q)	0.87	0.88	5.21	0.80	0.83	2.72
PFIQM (GQ)	0.85	0.82	5.68	0.70	0.61	3.28
PSNR	0.63	0.60	8.46	0.44	0.45	4.15
SSIM	0.69	0.70	7.86	0.44	0.43	4.15

Figures 6.10 through 6.13 show some interpolated frames produced by different MCFI methods, and Table 6.3 shows all metrics' outputs processed by Equation (6.30) and (6.31) for each image. In Fig. 6.10, visual observation gives a consensus that Fig.

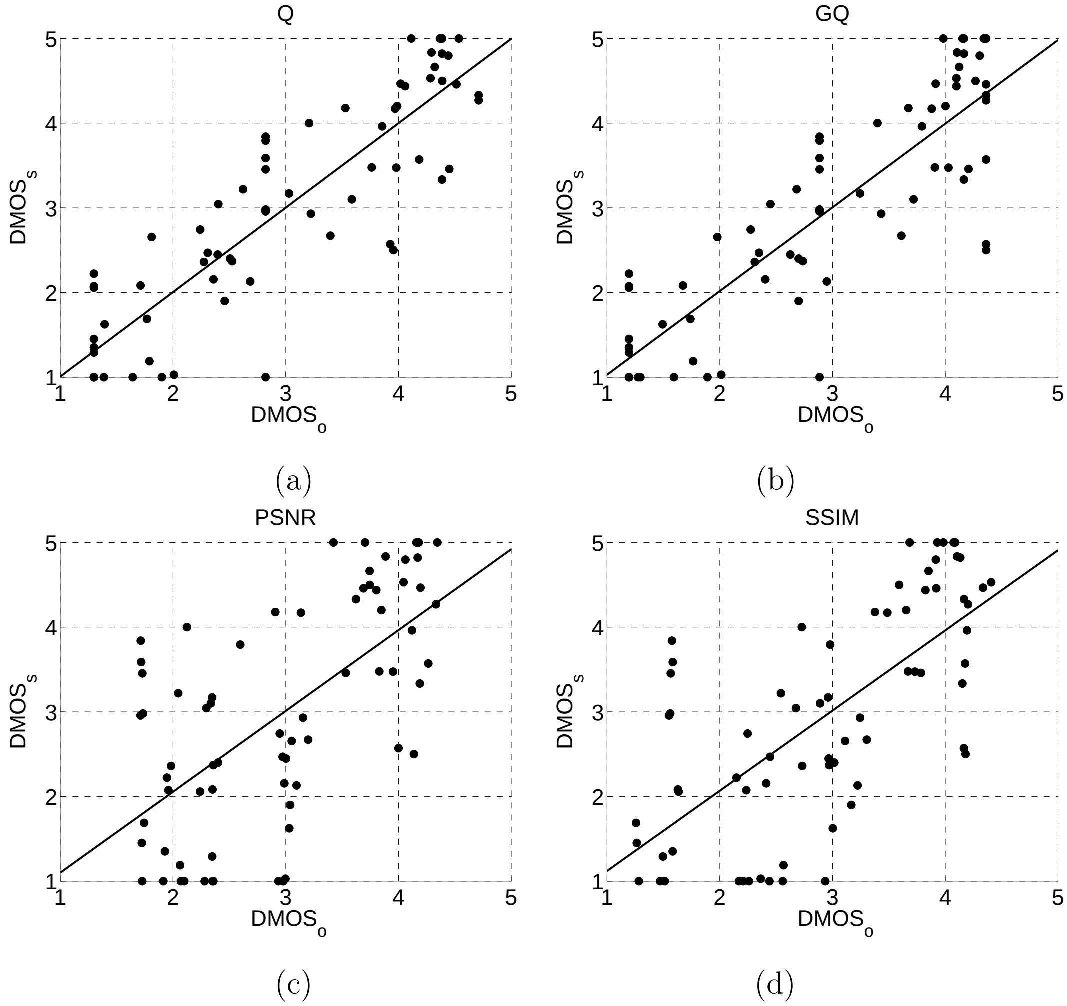
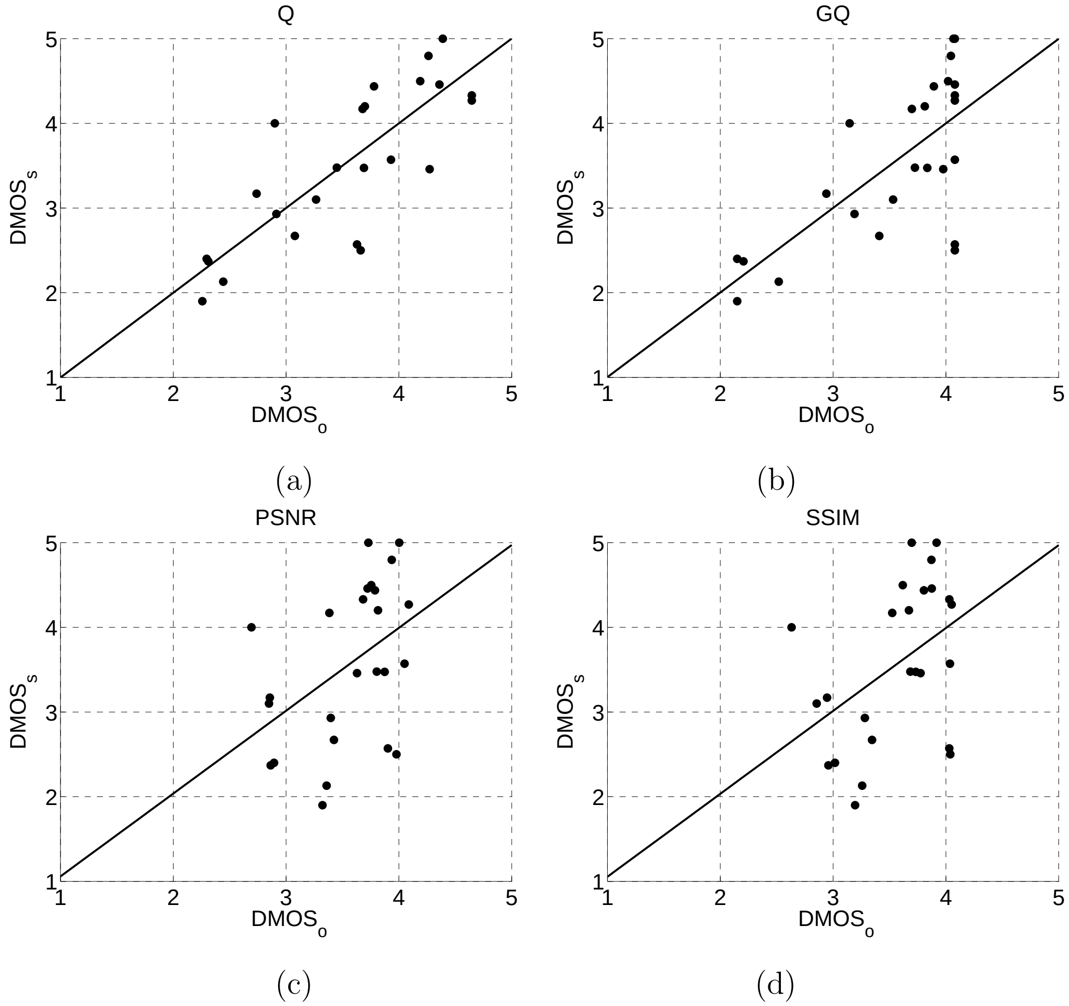


Figure 6.8: Subjective rating of all test cases vs. the corresponding objective scores from (a) PFIQM(Q), (b) PFIQM(GQ), (c) PSNR, and (d) SSIM

6.10(a) fails in preserving mouth and nose structure and also suffers severe blockiness, but Fig. 6.10(b) has better quality. The PFIQM and SSIM successfully reflect this difference, but PSNR fails in this case. Figure 6.11 shows examples of a mixture of global and local quality impairment. Both samples are highly degraded in the moving regions, but Fig. 6.11(a) contains more blockiness than Fig. 6.11(b). Also, Fig. 6.11(a) has a very noticeable artifact on foreman's helmet and face. Among the three metrics, both PSNR and SSIM tend to assign higher quality values to 6.11(a), only PFIQM's results are consistent with visual observation. Figure 6.12 provides an example where both Fig. 6.12(a) and Fig. 6.12(b) have very similar visual quality.



(a)

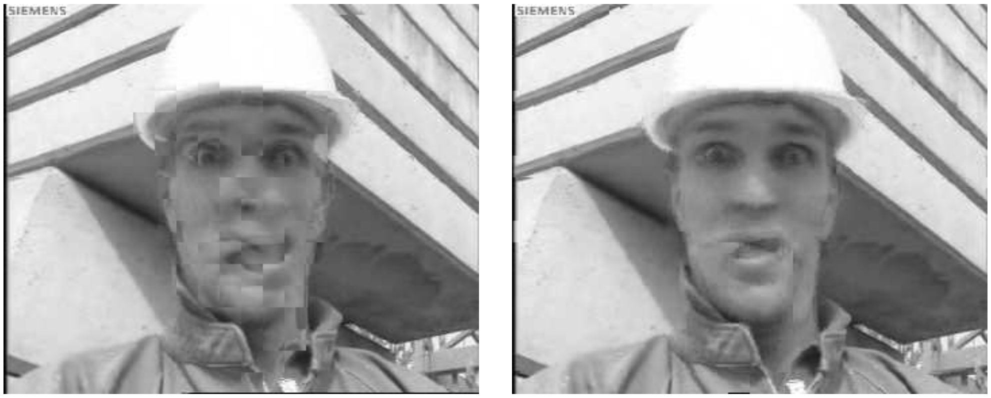
(b)

(c)

(d)

Figure 6.9: Subjective rating of test cases ii and vi only vs. the corresponding objective scores from (a) PFIQM(Q), (b) PFIQM(GQ), (c) PSNR, and (d) SSIM

The PFIQM assigns a similar score to these two images, but PSNR gives a much higher score to Fig. 6.12(a). In the following, an example with weak global distortion but salient local quality impairment is shown in Fig. 6.13. Ignoring the conspicuous blocking artifacts on the helmet edge, the quality of these two images is very close. However, if local distortion is counted, then Fig. 6.13(a) is worse than Fig. 6.13(b). The PFIQM's outputs agree with the visual judgment, but PSNR scores show a strong contradiction. Not surprisingly, SSIM gives similar scores for both images since it does not consider local distortion. Overall, high correlation of the PFIQM with subjective evaluation has been proven through this exercise. The second best metric is SSIM,



(a) (b)

Figure 6.10: Examples of severe global distortion that (a) is interpolated by the direct approach, and (b) is produced by the multi-stage method

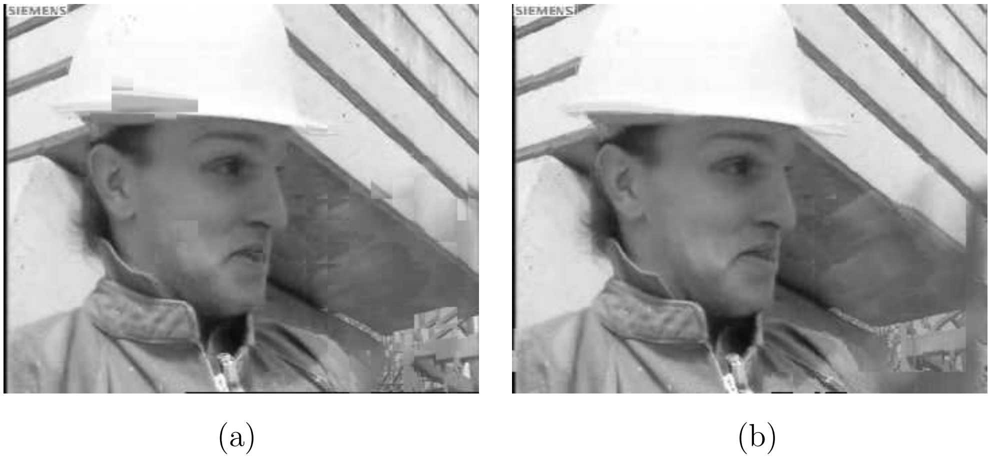
and PSNR performs worst in measuring the quality degradation caused by frame interpolation.

Table 6.3: Quality scores from PFIQM, PSNR, and SSIM of several sample images

	$DMOS_o(Q)$	$DMOS_o(PSNR)$	$DMOS_o(SSIM)$
Fig. 6.10(a)	1	3.12	3.9
Fig. 6.10(b)	3.74	3.09	4.25
Fig. 6.11(a)	2.67	2.45	3.92
Fig. 6.11(b)	3.41	2.12	3.71
Fig. 6.12(a)	4.64	4.22	4.76
Fig. 6.12(b)	4.62	3.6	4.63
Fig. 6.13(a)	4.18	4.19	4.78
Fig. 6.13(b)	5	3.77	4.77

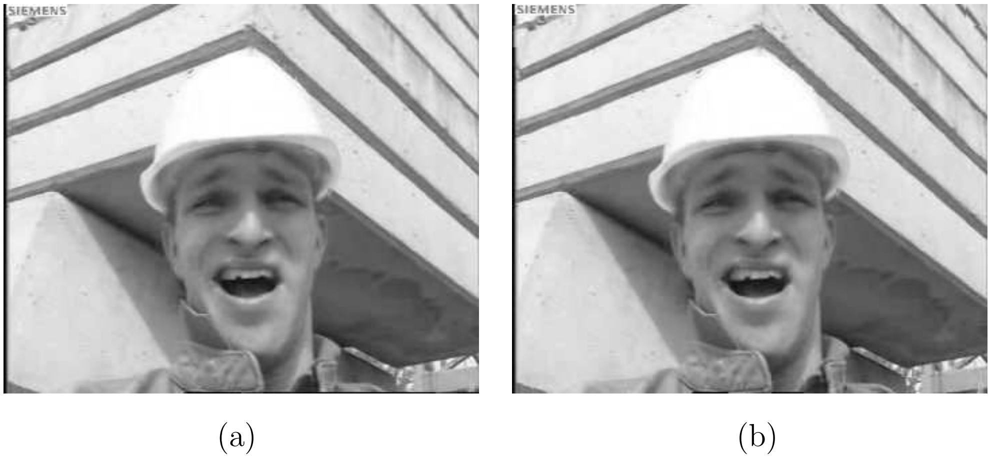
6.5 Summary

This chapter has investigated the quality prediction accuracy of two widely used spatial quality metrics for frame interpolation. Several disadvantages of the metrics have been presented, and a new metric, PFIQM, that overcomes these issues is demonstrated. This metric is designed based on prior knowledge about frame interpolation, such as type of artifacts, possible regions of quality degradation, and the occurrence of highly conspicuous local distortion. Performance evaluation shows that



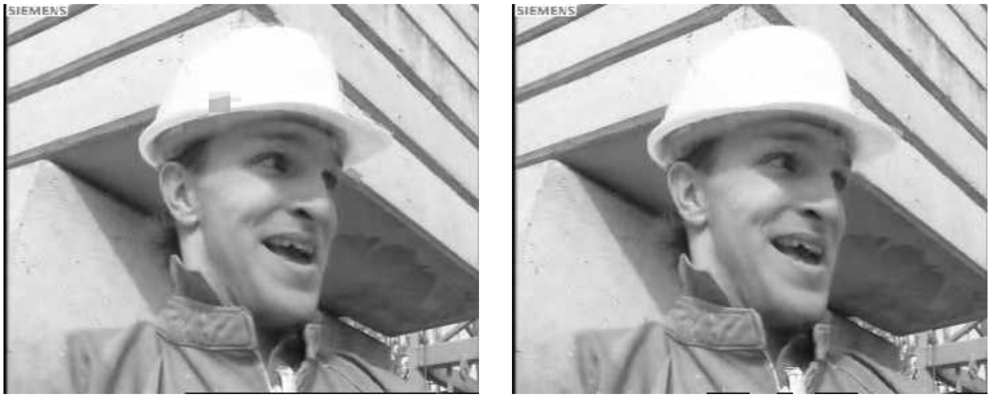
(a) (b)

Figure 6.11: Examples of mixtures of global and local quality degradation that (a) is interpolated by direct approach, and (b) is produced by the multi-stage method



(a) (b)

Figure 6.12: Examples of same visual quality but different PSNR value, where (a) is interpolated by direct approach, and (b) is produced by multi-stage method [2]



(a) (b)

Figure 6.13: Examples of low global quality degradation but strong local distortion, where (a) is interpolated by vector median, and (b) is produced by the multi-stage method

the PFIQM significantly outperforms the other metrics and is highly consistent with subjective ratings.

PUBLICATIONS

Kai-Chieh Yang, Ai-Mei Huang, Truong Nguyen, Clark C. Guest, and Pankaj K. Das, "A New Objective Quality Metric for Frame Interpolation Used in Video Compression ", submitted to *IEEE Transaction on Broadcasting*

Kai-Chieh Yang, Ai-Mei Huang, Truong Nguyen, Clark C. Guest, and Pankaj K. Das, "New Objective Quality Metric for Frame Interpolation Using in Video Compression", in *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. II - 177 - II - 180, Oct. 2007

7

Conclusion

Video related applications are prominent in our daily life. Because of the popularity of multimedia services, viewers are no longer satisfied with just the availability of multimedia applications, but also demand better quality multimedia service. Therefore, optimizing the compression performance of digital video systems with respect to viewer perceived quality has become an important issue in the field of video processing. Thorough understanding and accurate quantitative analysis of the video quality distortion caused by video compression is crucial for this task.

Most video quality measurement techniques focus on developing quality metrics for a general purpose, which outputs a single quality score for a compressed video sequence. Very often, this type of metric uses the fidelity between compressed and original video data, along with several human visual system (HVS) factors, to predict viewer perceived quality. Human vision is a very complicated process; it is affected by many high-level cognitive and low level visual biological factors. Emulating such a system requires an elaborate implementation and results in costly computation. Also, since this type of metric only produces a single quality score, it lacks the ability to provide deeper and more detailed analysis of the root causes of quality degradation from different sources; it does not help codec designer to enhance compressed video quality in the most efficient fashion. Because of its dependency on fidelity information, requiring the original video data greatly limits the application of this type of metric. Aside from general-purpose metrics, some researchers have developed metrics for certain types of artifacts introduced by video compression. However, the development of these type of metrics is still in its infancy. Thus, for the purpose of enhancing the visual quality of compressed video, accurate and detailed quality measurements for several pronounced artifacts are essential.

7.1 Achievements

Following an overview of video compression, types of video quality degradation have been described, and the latest developments in video quality assessment techniques have been reviewed. Then, a human visual module, the Visual Blurriness Sensitivity Map (VBSM) designed for quality assessment and enhancement related applications, is presented. Unlike conventional visual attention models, the VBSM not only includes positive stimuli affecting the visual aspect, but also considers suppression effects from a HVS perspective. It works in the spatial frequency domain and on a block basis. These characteristics permit a greater variety of applications for VBSM.

Because of current codec design, blurring artifacts have become the most pronounced impairments. A thorough review of several representative blurriness metrics has been presented, and a novel perceptual blurriness metric (PBM) for compressed video is implemented. The PBM works without accessing original video data, and it uses a robust estimation basis for blurriness assessment. The VBSM is employed to emulate human visual attention and masking effects. Several cognitive factors are considered. In contrast to many existing blurriness metrics, PBM is insensitive to the type of video content. A subjective experiment was carried out to examine the significance of several modules of PBM and compare the overall blurriness estimation accuracy of PBM to other metrics. Experimental results show that PBM not only has higher blurriness estimation accuracy than other metrics for video sequences compressed by the MPEG-4 codec, but also exhibits remarkable performance for video sequences compressed by H.264/AVC.

Temporal quality degradation is often caused by either frame skipping for the purpose of video data size reduction or transmission errors. An overview of existing temporal quality metrics has been given in this thesis and a new temporal quality metric, the Perceptual Temporal Quality Metric (PTQM), was demonstrated. Most existing temporal quality metrics can only estimate the temporal quality degradation caused by uniformly distributed frame loss. However, the PTQM can accurately estimate the temporal quality distortion caused by any frame loss distribution. It also employs a robust motion mapping model to adjust the temporal quality score for sequences with the same amount of frame loss but different motion levels. Outputs from PTQM are arranged in a hierarchical format; they not only contain a sequence level temporal quality score, but also those of segment and frame levels. Subjective experiments show that PTQM's outputs highly correspond to human perceived temporal quality.

Frame interpolation technique is a common way to enhance temporal quality by reproducing missing frames. However, it also introduces spatial artifacts that are very annoying and differ from compression artifacts. Conventional quality assessment approaches can not be applied to these artifacts directly. Thus, a detailed investigation of spatial quality impairment introduced by frame interpolation is reported. Based on this investigation, a perceptual frame interpolation quality metric (PFIQM) has been implemented. This metric considers the following artifact characteristics: appearance, possible occurrence location, and spatial distribution. A subjective test was carried out to compare the performance of PFIQM to two other quality metrics that are widely used in frame interpolation quality assessment. Comparison shows that the PFIQM is the metric that provides quality scores closest to subjective ratings.

7.2 Future work

Several artifacts that occur most often and are most pronounced in modern multimedia applications have been analyzed in this thesis. The conclusions address a strong need for an accurate measurement of each individual artifact. Based on this foundation, several novel metrics have been developed, and experimental results show superior performance for the proposed metrics compared to existing metrics. However, only a small number could be investigated within the scope of this thesis, and many extensions and improvements can be contemplated.

Several direct improvements can be considered:

- In VBSM, output of individual visual features in VBSM that belong to the same object are grouped together in the spatial post-processing stage by a median filter. It is a low computation solution, but lacks a sense of object orientation. Thus, including object segmentation one might enhance the accuracy of visually significant region determination.

- Shifting camera motion is the only type of camera motion considered in VBSM since it is the major type of camera motion that causes temporal masking effects. Other types of camera motion may not directly relate to temporal masking phenomenon, but they can induce other impacts in determining human visually significant regions. Hence, including other types of camera motion can increase the generality of VBSM and permit wider application.

- The normalization and cognitive modules in PBM are trained by a limited amount of training data. In order to increase the generality of their application, these modules should be trained using a larger data set with wider coverage of different subjects. However, as the size of training data increases, the over-training should be cautiously avoided by measuring the correlation for each training datum.

- Finally, the pooling method used in PFIQM is an evenly weighted average. This part still requires more research effort for optimization, since the dominance of each artifact may change along with the amount of artifacts, content type, and many other subjective factors. The evenly weighted multi-metrics pooling method can only produce a fairly good, but not the most accurate, quality prediction performance. Therefore, a deeper and more detailed study of pooling methods for multiple metrics is a challenging but helpful task.

7.3 Closing remarks

Video quality assessment is a complicated topic. It requires a knowledge of video and image processing, compression, the human vision system, and psychological effects. The mainstream of this research field is to establish mathematical models to measure human perceived quality without carrying out subjective quality test experiment by collecting viewers' feedbacks. More than one decade of research effort has been spent on this topic and a perfect solution for establishing good objective quality metrics is still not in existence. As the size of video display devices gets larger, any small quality impairment will be enlarged and become very annoying. As a result, viewers' expectation for displayed video quality will significantly increase as well. Thus, producing high quality compressed video data will be more challenging than in the past, and the role of good video quality assessment methods will become more important. Moreover, combining low computational cost while staying close to human perception quality metric using compression algorithms to achieve perceptual coding is another interesting research direction. With guidance from a good metric, the resources (i.e. bits) can be allocated in a more adaptive fashion to ensure minimal quality distortion of compressed video data. As two-dimensional video service achieves its maturity, video service displayed in three-dimensions will be the next focus in the multimedia industry. Visual quality issues in two-dimensional video data is not entirely applicable to video data displayed in the three-dimensional domain. Some additional effects, such as object depth and specific artifacts caused by stereo display, must be addressed.

Bibliography

[1] X. Marichal, W.-Y. Ma, and H. Zhang, "Blur determination in the compressed domain using DCT information," in *Proc. IEEE International Conference on Image Processing*, vol. 2, 1999, pp. 386–390.

[2] A.-M. Huang and T. Q. Nguyen, "A novel multi-stage motion vector processing method for motion compensated frame interpolation," in *Proc. IEEE International Conference on Image Processing*, vol. 5, 2007, pp. V – 389–V – 392.

[3] I. E. Richardson, Ed., *H.264 and MPEG-4 Video Compression.* John Wiley and Sons, 2003.

[4] M. Yuen, "Coding artifacts and visual distortions," in *Digital Video Image Quality and Perceptual Coding*, H. Wu and K. Rao, Eds., November 2005, pp. 87–119.

[5] I. T. ISO/IEC JTCI 10918-1. ITU-T Rec. T.81, "Digital compression and coding of continuous-tone still images: Requirements and guidelines," ISO/IEC, Tech. Rep., 1993.

[6] I. T. ISO/IEC 14495-1, "Lossless and near-lossless compression of continuous-tone still images: Baseline," ISO/IEC, Tech. Rep., 2000.

[7] I. T. ISO/IEC 14496-2, "Coding of audio-visual objects - part2: Visual," ISO/IEC, Tech. Rep., 2001.

[8] I. T. ISO/IEC 14496-10 ITU-T Rec. H.264, "Advanced video coding," ISO/IEC, Tech. Rep., 2003.

[9] I. T. ISO/IEC 15938, "Multimedia content description interface (mpeg-7)," ISO/IEC, Tech. Rep., 2002.

[10] I. T. ISO/IEC 11172, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5mbit/s (mpeg-1)," ISO/IEC, Tech. Rep., 1993.

[11] I. T. ISO/IEC 13818, "Generic coding of moving pictures and associated audio information (mpeg-2)," ISO/IEC, Tech. Rep., 1995.

[12] W. Li, “Overview of fine granular scalability in MPEG-4 video standard,” *IEEE Transaction on Circuites System Video Technology*, vol. 3, pp. 301–318, March 2001.

[13] I.-T. R. H.261, “Video codec for audiovisual services at px64kbit/s,” ITU, Tech. Rep., 1993.

[14] I.-T. R. H.263, “Video coding for low bit rate communication, version 2,” ITU, Tech. Rep., 1998.

[15] G. Sullivan and T. Wiegand, “Video compression - from concepts to the h.264/avc standard,” *Proceedings of IEEE*, vol. 93, pp. 18–31, Jan 2005.

[16] Z. Lu, W. Lin, X. Yang, E. Ong, and S. Yao, “Modeling visual attention’s modulatory aftereffects on visual sensitivity and quality evaluation,” *IEEE Transaction Image Processing*, vol. 14, pp. 1928–1942, Nov. 2005.

[17] K. Yang, C. Guest, and P. Das, “Motion blur detection by support vector machine,” in *Proc. SPIE nternational Symposium on Multimedia*, vol. 5916, Aug. 2005, pp. 261–273.

[18] I.-R. R. BT.500-11, “Methodology for the subjective assessment of the quality of television pictures,” 2002.

[19] G. Rubino and M. Varela, “A new approach for the prediction of end-to-end performance of multimedia streams,” in *Proc. in the first International Conference on the Quantitative Evaluation of Systems*, Sep. 2004, pp. 110– 119.

[20] S. Mohamed and G. Rubino, “A study of real-time packet video quality using random neural networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1071– 1083, Dec. 2002.

[21] S. Winkler, Ed., *Digital Video Quality: Vision Models and Metrics.* John Wiley and Sons, March 2005.

[22] S. Winkler, “Preceptual video quality metrics - a review,” in *Digital Video Image Quality and Perceptual Coding*, 2005, pp. 155–172.

[23] ——, “A perceptual distortion metric for digital color video,” in *Proc. of the SPIE*, vol. 3644, January 1999, pp. 175–184.

[24] M. A. Marsy and S. S. Hemami, “A metric for continuous quality evaluation of compressed video with severe disortions,” *Signal Processing: Image Communication*, vol. 19, pp. 133–146, 2004.

[25] J. Lubin, “A visual discrimination model for imaging system design and evaluation,” *Vision Models for Target Detection and Recognition*, pp. 245–283, 1995.

[26] S. Daly, “The visible differences predictor: An algorithm for the assessment of image fidelity,” in *Proc. of the SPIE Human Vision, Visual Processing, and Digital Display III*, vol. 1666, January 1993, pp. 2–15.

[27] A. B. Watson, “Toward a perceptual video quality metric,” in *Proc. of the SPIE Human Vision and Electronic Imaging III*, vol. 3299, July 1998, pp. 139–147.

[28] K. Tan and M. Ghanbari, “A multi-metric objective picture-quality measurement model for MPEG video,” *IEEE Transactions in Circuits and Systems for Video Technology*, vol. 10, pp. 1208–1213, Oct. 2000.

[29] S. Wolf and M. H. Pinson, “Spatial-temporal distortion metrics for in-service quality monitoring of any digital video system,” in *Proc. SPIE*, vol. 3845, September 1999, pp. 266–277.

[30] Z. Wang, L. Lu, and A. C. Bovik, “Video quality assessment based on structural distortion measurement,” *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, February 2004.

[31] P. G. J. Barten, Ed., *Contrast Sensitivity of the Human Eye and Its Effects on Image Quality.* SPIE Optical Engineering Press, 1999.

[32] G. van der Horst and M. A. Bouman, “Spatiaotempoal chromaticity discrimination,” *Journal of the Optical Society of America*, pp. 1482–1488, 1969.

[33] E. M. Granger and J. C. Heurtley, “Visual chromaticity-modulation transfer function,” *Journal of the Optical Society of America*, pp. 1173–1174, 1973.

[34] S. A. Klein, T. Carney, L. Barghout-Stein, and C. W. Tyler, “Seven models of masking,” in *Proc. SPIE*, Feb 1997, pp. 13–24.

[35] M. J. Nadenau, J. Reichel, and M. Kunt, “Performance comparison of masking models based on a new psychovisual test method with natural scenery stimuli,” *Signal Processing: Image Communication*, pp. 807–823, November 2002.

[36] A. B. Watson, R. Borthwick, and M. Taylor, “Image quality and entropy masking,” in *Proc. SPIE*, February 1997, pp. 2–12.

[37] J. Caviedes and F. Oberti, “No-reference quality metric for degraded and enhanced video,” in *Proc. SPIE*, vol. 5150, July 2003, pp. 621–632.

[38] Z. Wang, H. Sheikh, and A. C. Bovik, “No-reference perceptual quality assessment of JPEG compressed images,” in *Proc. IEEE International Conference on Image Processing*, Seq. 2002, pp. I–477 – I–480.

[39] R. Chan and P. Goldsmith, “A psychovisually-based image quality evaluator for JPEG images,” in *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, Oct. 2000, pp. 1541–1546.

[40] J. Caviedes and F. Oberti, “A new sharpness metric based on local kurtosis, edge and energy information,” *Signal Processing: Image Communication*, vol. 19, pp. 147–161, 2004.

[41] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahini, “Perceptual blur and ringing metrics: Applications to JPEG2000,” *Signal Proccessing: Image Communication*, pp. 163–172, 2004.

[42] K. C. Yang, C. C. Guest, and P. K. Das, “Perceptual sharpness metric for compressed video,” in *Proc. IEEE international Conference on Multimedia and Expo*, July 2006, pp. 777 – 780.

[43] H. R. Wu and M. Yuen, “Quantitative quality metrics for video coding blocking artifacts,” in *Proc. of Picture Coding Symposium 1*, 1996, pp. 23–26.

[44] H. Wu and M. Yuen, “A generalized block-edge impairment metric for video coding,” *IEEE Signal Processing Letters*, vol. 4, pp. 317–320, Nov. 1997.

[45] S. A. Karunasekera and N. G. Kingsbury, “A distortion measure for blocking artifacts in images based on human visual sensitivity,” *IEEE Transaction Image Processing*, vol. 4, no. 6, pp. 713–724, 1995.

[46] W. Gao, C. Mermer, and Y. Kim, “A de-blocking algorithm and a blockiness metric for highly compressed images,” *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 3, pp. 1150 – 1159, 2002.

[47] S. Oguz, Y. Hu, and T. Nguyen, “Image coding ringing artifact reduction using morphological post-filtering,” in *Proc. of IEEE Second Workshop on Multimedia Signal Processing*, 1998, pp. 628 – 633.

[48] N. Zhang, A. E. Vladar, M. T. Postek, and B. Larrabee, “A kurtosis-based statistitcal measure for two-dimensional processes and its application to image sharpness,” in *Proc. of Section of Physical and Engineering Sciences of American Statistical Society*, 2003, pp. 4730–4736.

[49] R. Feghali, D. Wang, F. Speranza, and A. Vincent, “Quality metric for video sequences with temporal scalability,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, pp. 137–40.

[50] G. Iacovoni, S. Morsa, and R. Felice, “Quality-temporal transcoder driven by the jerkiness,” in *Proc. IEEE International Conference on Multimedia and Expo*, 2005, pp. 1452–1455.

[51] Z. Lu, W. Lin, B. Seng, S. Kato, S. Yao, E. Ong, and X. Yang, “Measuring the negative impact of frame dropping on perceptual visual quality,” in *Proc. of the SPIE Human Vision and Electronic Imaging X*, vol. 5666, March 2005, pp. 554–562.

[52] M. Montenovo, A. Perot, M. Carli, P. Cicchetti, and A. Neri, "Objective quality evaluation of video services," in *Proc. of the Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, January 2006.

[53] R. R. Pastrana-Vidal and J. C. Gicquel, "Automative quality assessment of video fludity impairments using a no-reference metric," in *Proc. of the Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, January 2006.

[54] R. R. Pastrana-Vidal, J. C. Gicquel, C. Colomes, and C. Hocine, "Sporadic frame dropping impact on quality perception," in *Proc. SPIE Electronic Imaging, Human Vision and Electronic Imaging IX*, 2004, pp. 182–193.

[55] K. Watanabe, J. Okamoto, and T. Kurita, "Objective video quality assessment method for freeze distortion based on freeze aggregation," in *Proc. SPIE Electronic Imaging, Image Quality and System Performance*, 2006, pp. 60 590–60 598.

[56] ITU-R Document 6Q/14, "Final report from the video quality experts group on the validation of objective models of video quality assessment, Phase II (FR-TV2)," 2003. [Online]. Available: http://www.vqeg.org

[57] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transaction Patten Analysis Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[58] X. Yang, W. Lin, Z. Lu, X. Lin, S. Rahardja, E. Ong, and S. Yao, "Rate control for videophone using local perceptual cues," *IEEE Transaction Circuits Systems Video Technology*, vol. 15, no. 4, pp. 496–507, April. 2005.

[59] Z. Lu, W. Lin, X. Yang, E. Ong, and S. Yao, "Modeling visual attention's modulatory aftereffects on visual sensitivity and quality evaluation," *IEEE Transaction Image Processing*, vol. 14, no. 11, pp. 1928 – 1942, Nov. 2005.

[60] K. Yang, C. Guest, and P. Das, "Human visual attention for compressed video," in *Proc. IEEE international Symposium on Multimedia*, Dec. 2006, pp. 525–532.

[61] ITU-R Recommendation BT.601-4, "Encoding parameters of digital television for studios." International Telecommunication Union, 1994.

[62] J. Caviedes and S. Gurbuz, "A perceptual model for JPEG applications based on block classification, texture masking, and luminance masking," in *Proc. IEEE International Conference on Image Processing*, vol. 3, 1998, pp. 428–432.

[63] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using non-parametric kernel density estimation for visual surveillance," in *Proc. IEEE*, vol. 90, 2002, pp. 1151 – 1163.

[64] J. McCarthy, M. Sasse, and D. Miras, “Sharp or smooth? comparing the effects of quantization vs. frame rate for streamed video,” in *Proc. SIGCHI conference on Human factors in computing systems*, 2004, pp. 535 – 542.

[65] Y. Jia, W. Lin, and A. Kassim, “Estimating just-noticeable distortion for video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 820–829, July 2006.

[66] B. Birod, “The information theoretical significance of spatial and temporal masking in video signals,” in *Proc. SPIE Conference of Human Vision, Visual Processing and Digital Display*, vol. 1077, 1989, pp. 178–187.

[67] A. Akutsu, Y. Tonomura, H. Hashimoto, and Y. Ohba, “Video indexing using motion vectors,” in *Proc. SPIE: Visual Communication, Image Process*, 1992, pp. 1522–1530.

[68] Z. Lu, W. Lin, X. Yang, E. Ong, and S. Yao, “Modeling visual attention’s modulatory aftereffects on visual sensitivity and quality evaluation,” *IEEE Transac. Image Processing*, pp. 1928–1942, 2005.

[69] R. F. Hess and R. J. Snowden, “Temporal properties of human visual filters: Numbers, shapes and spatial covariation,” *Visual Res.*, vol. 32, no. 1, pp. 47–59, 1992.

[70] M. S. Moore, J. M. Foley, and S. K. Mitra, “Detectability and annoyance value of mpeg-2 artifacts inserted in uncompressed video sequences,” in *Proc. SPIE: Human Vision and Electronic Imaging V*, vol. 3959, June 2000, pp. 99–110.

[71] M. Yuen and H. R. Wu, “A survey of hybrid MV/DPCM/DCT video coding artifacts,” *Signal Processing*, pp. 247–278, 1998.

[72] B. Zhang, J. P. Allebach, and Z. Pizlo, “An investigation of perceived sharpness and sharpness metrics,” in *Proc. SPIE The International Society for Optical Engineering*, vol. 5668, 2005, pp. 98–110.

[73] E. Ong, W. Lin, X. Yang, S. Yao, F. Pan, L. Jiang, and F. Moschetti, “A no-reference quality metric for measuring image blur,” in *Proc. IEEE Seventh International Symposium on Signal Processing and Its Applications*, vol. 1, July 2003, pp. 469–472.

[74] J. A. Movshon and L. Kiorpes, “Analysis of the development of spatial contrast sensitivity in monkey and human infants,” *Journal of the Optical Society of America A*, no. 12, pp. 2166–2172, 1988.

[75] P. G. J. Barten, “Evaluation of subjective image quality with the square-root internal method,” *Journal of the Optical Society of America A*, no. 10, pp. 2024–2031, 1990.

[76] media.xiph.org. [Online]. Available: http://media.xiph.org/video/derf/

[77] [Online]. Available: http://trace.eas.asu.edu/yuv/index.html

[78] JM12. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[79] A. Leontaris, P. C. Cosman, and A. R. Reibman, "Quality evaluation of motion-compensated edge artifacts in compressed video," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 943–956, Apr. 2007.

[80] M. Claypool and J. Tanner, "The effects of jitter on the perceptual quality of video," in *Proc. of the seventh ACM international conference on Multimedia (Part 2)*, October 1999, pp. 115–118.

[81] L. Alparone, M. Barni, F. Bartolini, and V. Cappellini, "Adaptively weighted vector-median filters for motion-fields smoothing," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 1996, pp. 2267–2270.

[82] G. Dane and T. Q. Nguyen, "Smooth motion vector resampling for standard compatible video post-processing," in *Proc. Asilomar Conf. Signals, Systems and Computers*, vol. 2, Nov. 2004, pp. 1731– 1735.

[83] A.-M. Huang and T. Q. Nguyen, "A novel motion compensated frame interpolation based on block-merging and residual energy," in *Proc. International Workshop on Multimedia Signal Processing*, 2006, pp. 395–398.

[84] J. Meng, Y. Juan, and S. F. Chang, "Scene change detection in a MPEG compressed video sequence," in *Proc. SPIE Symposium Proceedings on Electronic Imaging: Science & Technology*, vol. 2419, February 1995, pp. 14–25.

[85] S. Kato, C. Boon, A. Fujibayashi, S. Hangai, and T. Hamamoto, "Perceptual quality of motion of video sequences on mobile terminals," in *Proc. of the Seventh IASTED International Conference on Signal and Image Processing*, August 2005, pp. 442–447.

[86] S. Kato, C. Boon, T. Horikoshi, and S. Hangai, "Time-displacement based perceptual model for motion quality of video sequences and its application on non-constant motion," in *Proc. of the First International Workshop on Image Media Quality and its Applications*, August 2005, pp. 103–107.

[87] M. Camperi and X. Wang, "A model of visualspatial working memory in prefrontal cortex: Recurrent network and celluar bistability," *Journal of Computational Neuronscience*, no. 5, pp. 383–405, Jan. 1998.

[88] H. Eng, D. Chen, and Y. Jiang, "Visual working memory for simple and complex visual stimuli," *Journal of psychonomic bulletin and review*, vol. 12, no. 6, pp. 1127–1133, Dec. 2005.

[89] J. Wang, N. Patel, and W. Grosky, “A fast block-based motion compensation video frame interpolation approach,” in *Proc. Asilomar Conf. Signals, Systems and Computers*, 2004, pp. 1740–1743.

[90] R. Barland and A. Saadane, “Reference free quality metric using a region-based attention model for jpeg-2000 compressed images,” in *Proc. SPIE: Image Quality and System Performance III*, vol. 6059, Jan. 2006, pp. 605 905–1–10.

[91] S. Lee, M. Pattichis, and A. Bovik, “Foveated video quality assessment,” *IEEE Transaction Multimedia*, vol. 4, pp. 129–132, March 2002.

[92] R. Li, B. Zeng, and M. L. Liou, “A new three-step search algorithm for block motion estimation,” *IEEE Transaction on Circuit System and Video Technology*, vol. 4, no. 4, pp. 438–442, Aug. 1994.

[93] Video Processing Lab. in UCSD. [Online]. Available: http://videoprocessing.ucsd.edu/ aihuang/DownloadPage2.htm