NEW APPROACHES AND A SUBJECTIVE DATABASE FOR VIDEO QUALITY ASSESSMENT

By

PHONG VAN VU

Master of Science in Electrical and Computer
Engineering
Oklahoma State University
Stillwater, OK 74075
2013

Bachelor of Engineering in Electronics and
Telecommunications
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
2004

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
Doctor of Philosophy
July, 2014

NEW APPROACHES AND A SUBJECTIVE DATABASE FOR VIDEO QUALITY ASSESSMENT

Dissertation Approved:

Dr. Damon Chandler

Advisor

Dr. Martin Hagan

Dr. Guoliang Fan

Dr. Joe Cecil

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Damon M. Chandler, for his encouragement, interest, and patience. Personally, I would like to thank him for sharing his knowledge which has enriched my study and research.

I would like to thank Dr. Hagan, Dr. Fan, and Dr. Cecil for serving my committee and give me many great comments. I would like to thank Dr. Hagan and Dr. Fan for teaching me useful and important courses, and thank Dr. Cecil for giving experience with virtual reality systems and industrial engineering.

I would like to thank my partners from the Computational Perception and Image Quality laboratory, for their great supports and assistance in completing my program.

Finally, I would like to thank my wife and my parents for their encouragement, support, and love. They are always beside me to help overcome the challenges and difficulties.

The material in this dissertation is based upon work supported by, or in part by, the National Science Foundation, “Content-Based Strategies of Image and Video Quality Assessment,” Award 0917014, the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-10-1-0015.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: Phong Van Vu

Date of Degree: July, 2014

Title of Study: NEW APPROACHES AND A SUBJECTIVE DATABASE FOR VIDEO QUALITY ASSESSMENT

Major Field: Electrical and Computer Engineering

Abstract:

Video quality assessment plays an important role in multimedia systems that process digital images/videos such as video codec, video streaming server. The use of video quality assessment algorithm helps optimize system parameters, increase quality of service, and satisfy customers' demands. Traditional method that recruits human subjects to judge video quality often comes with the expense of time, money, and effort while objective method, which uses computer and built-in algorithms to judge video quality, offers a more affordable way. This dissertation report provides an efficient approach to develop objective video quality assessment algorithm.

Algorithms in video quality assessment aim to predict quality of videos in a manner that agrees with subjective ratings of quality judged by human subjects. From that, two important factors are required for the research of video quality assessment. The first factor is an algorithm that is able to predict video quality. Our approach to develop such an algorithm bases on the analyses of spatial and spatiotemporal slices in two separate stages. The first stage estimates perceived quality degradation due to spatial distortion; this stage operates by adaptively applying our previous image quality assessment algorithm on a frame basis with an extension to account for temporal masking. The second stage estimates perceived quality degradation due to joint spatial and temporal distortion; this stage operates by measuring the dissimilarity between the two-dimensional spatiotemporal slices created by taking time-based slices of the original and distorted videos. The combination of these two estimates serves as an overall estimate of perceived quality degradation.

The second important factor in the research of video quality assessment is a video-quality database with collected subjective ratings used to validate the algorithms performance. We create our own video-quality database that consists of more videos (216 videos) and more distortion types (six) comparing to the currently available video-quality databases. The experiment to collect subjective ratings of quality is conducted by 40 different subjects following the SAMVIQ methodology.

Acknowledge that in many applications, the original video is not available; we develop another video quality assessment algorithm that can predict quality of a processed video without information of the original video. This algorithm, specifically designed for videos compressed by Motion JPEG2000 compression standard, consists of two analyses of quality degradations in the edge/near-edge regions and the non-edge regions of the videos. The algorithm shows promise in the first step of developing a general no-reference algorithm for video quality assessment.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

OVERVIEW OF VIDEO QUALITY ASSESSMENT

1.1 Introduction

With the rapid growth of multimedia technologies, the digital videos play a significantly important role in the social communities today. The demands for video services, including video sharing, video streaming, video surveillance etc., increase with the development of social networks and technologies. To satisfy those demands, many applications have been developed to record videos, compress, store, and transfer from/to the servers to/from the end-user over broadband connection, wireless communications, etc. The recording devices range from a cellphone camcorder with low resolution to the super high resolution supported by the professional camcorder and therefore, the video quality varies. In most applications, the owners/producers want to maintain the videos with high quality in order to provide excellent services to the end-users/customers. To perform this task, producers need a video quality assessment (hereafter referred to as VQA) method/system that can accurately predict the quality of a specific video. This VQA method can be used to quantify the effect of a new compression standard or a custom transmission environment to the video quality before it is delivered to the end-user. VQA methods can also be implemented in a service system to maintain, control, and possibly enhance the systems QoS (quality of service), e.g. the video streaming service. Therefore, an effective and robust VQA method is crucial and required for many applications.

There are two types of VQA methods: subjective and objective. The subjective method is a reliable way to judge video quality because this method collects quality ratings from a statistically large group of human subjects, which represent the whole population of ultimate receivers in most video applications. These subjective ratings of video quality are then carefully processed using rejection criterion, score normalization, etc. to obtain the representative video quality score in terms of Mean Opinion Score (MOS) and/or Difference Mean Opinion Score (DMOS). This method has advantages of high reliability and accuracy, however, it is inconvenient, very time-consuming, and too expensive to deploy frequently because of system settings, calibrations, subject paid, etc. Therefore, it is difficult and virtually impossible to collect subjective ratings about video products and customize them to improve quality. Because of these disadvantages, the subjective assessment method is mainly necessary in final product evaluation and/or standardization processes that strictly require high assurance of service quality.

The second method that can be deployed to assess video quality is the objective method, which offers a flexible and affordable way to perform VQA. The goal of the objective VQA method is to design an algorithm that can predict perceived video quality automatically and in a manner that agrees with subjective ratings given by human subjects. Objective VQA algorithms can be classified according to the availability of the original/reference video, which is considered to be distortion-free or perfect quality. A large number of proposed objective VQA algorithms in the literature assume that the reference video is fully available. Then, a VQA method can compare that reference video with the distorted videos to quantify quality degradations based on the difference in visual perception between two videos. VQA methods that fall into this category are called full-reference methods.

On the other hand, there are many practical service applications where the reference video is not available and we need to estimate video quality without any knowledge of the reference video. Despite the unavailability of the reference video, it is known that human subjects can effectively assess quality of a particular video with high reliability and accuracy. Therefore, it is necessary to develop an algorithm that can evaluate video quality blindly. Those algorithms are called blind or no-reference VQA algorithms; and, due to the lack of reference information, developing a no-reference algorithm is more difficult and challenging than developing a full-reference algorithm. The no-reference VQA method is often associated with a specific type of video artifacts where the characteristics are homogeneous across videos.

In the last few decades, a great deal of efforts has been made to develop objective VQA algorithms. Various algorithms has been reviewed and summarized in previous work [4, 6]. Many of them incorporate perceptual quality measures by studying the response characteristics of human visual system (HVS) to video quality. Some algorithms study the impairment of the processed videos using non-visual video features. However, VQA is still far from being a mature research topic. In fact, only limited success has been reported from evaluations of models under strict testing conditions and specific distortion types.

1.2 General approaches to full-reference VQA

The ability to quantify the visual quality of an image or video is a crucial step for any system that processes digital media. Full reference VQA algorithms aim to estimate quality of a distorted video comparing to the reference video in a manner that agrees with the quality judgments reported by human observers. Currently, the most basic and widely used as a baseline full-reference objective VQA algorithms are the mean squared error (MSE) and peak signal-to-noise ratio (PSNR), which are defined as:

$$MSE = \frac{1}{N} \sum_{i,j,k} (\mathbf{I}_{i,j,k} - \hat{\mathbf{I}}_{i,j,k})^2 \tag{1.1}$$

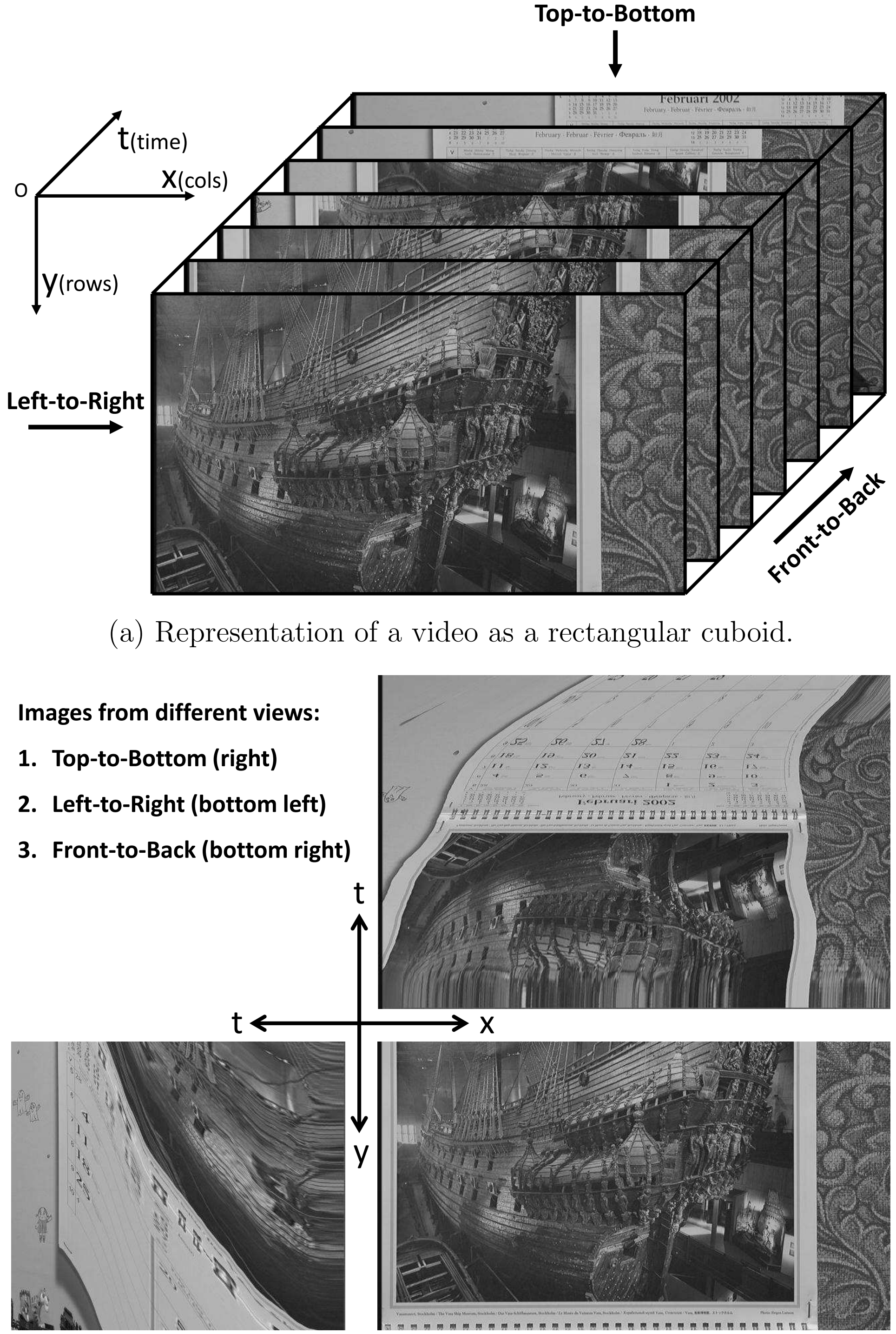
$$PSNR = 10 \log_{10} \frac{L^2}{MSE} \tag{1.2}$$

where N is the total number of digital pixels in the video; $\mathbf{I}_{i,j,k}$ and $\hat{\mathbf{I}}_{i,j,k}$ are the pixel values from the original and the distorted videos, respectively, at the spatial location $\{i, j\}$ of the k^{th} frame; L is the dynamic range of the pixel values (For an 8 bits/pixel monotonic signal, $L = 255$). The MSE and PSNR methods are widely used because of their simple calculation, clear physical meanings, and ease of use for optimization purposes. However, MSE and PSNR have been criticized for not correlating well with the perceived quality scores rated by human subjects.

Over the last few decades, numerous image-based quality assessment algorithms have been developed and shown to perform reasonably well on various image-quality databases. Therefore, a natural technique to VQA is to apply existing IQA algorithms to each frame of the video and to pool the per-frame results across time. Since each video frame is a single image, the key advantage of this frame-based approach is that it is very intuitive, easily implemented, and computationally efficient. However, such a frame-by-frame IQA approach to estimate video quality often fails to correlate with the subjective ratings of quality [7, 8].

One reason that frame-by-frame IQA performs less well for VQA is its ignorance of video temporal information, which is important for video quality due to temporal effects such as temporal masking and motion perception [9, 10]. To overcome this limitations, many researchers have incorporated temporal information into their VQA algorithms by supplementing frame-by-frame IQA with a model of temporal masking and/or temporal weighting [3, 11–13]. For example, in Refs. 11 and 12, motion-weighting and temporal derivatives have been used to extend SSIM[14] and VIF[15] for VQA.

Modern VQA algorithms often estimate video quality by extracting and comparing visual/quality features from localized space-time regions or groups of video frames. For example, in Refs. 2, 16, video quality is estimated based on spatial gradients, color information, and the interaction of contrast and motion from spatiotemporal blocks; motion-based temporal pooling is employed to yield the quality estimate. In


Top-to-Bottom
t(time)
o
x(cols)
y(rows)
Left-to-Right
Front-to-Back
Images from different views:
1. Top-to-Bottom (right)
2. Left-to-Right (bottom left)
3. Front-to-Back (bottom right)
t
t
x
y

(a) Representation of a video as a rectangular cuboid.

(b) Three different views of the video

Figure 1.1: A video can be envisaged as a rectangular cuboid in which two of the sides represent the spatial dimensions (x and y), and the third side represents the time dimension (t). If one takes slices of the cuboid from front-to-back, then the extracted slices correspond to normal video frames. Slicing the cuboid vertically and horizontally yield spatiotemporal slices images (STS images). Examples of three different slice types are presented in part (b) of the figure.

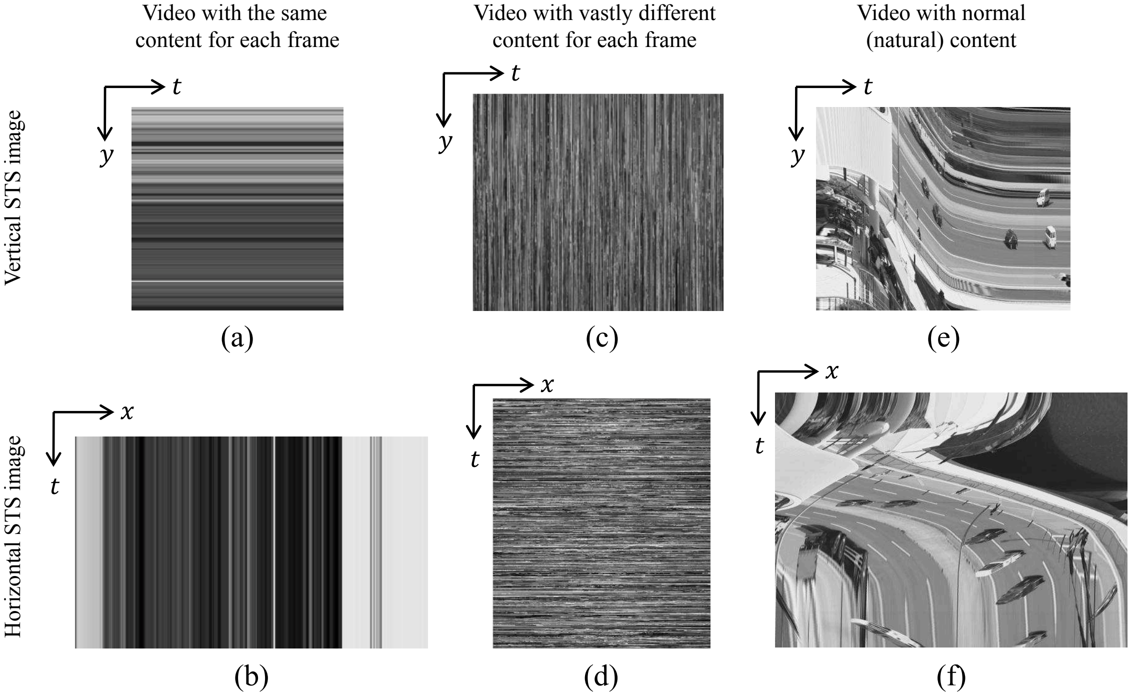

Video with the same content for each frame
Video with vastly different content for each frame
Video with normal (natural) content
Vertical STS image
Horizontal STS image
t
y
x
(a)
(c)
(e)
(b)
(d)
(f)

Figure 1.2: Demonstrative STS images extracted from a static video [(a) and (b)], from a video with a vastly different content for each frame [(c) and (d)], and from a typical normal natural video [(e) and (f)]. The STS images for the atypical videos in (a)-(d) appear similar to textures, whereas the STS images for normal videos are generally smoother and more structured due to the joint spatial and temporal (spatiotemporal) relationship.

Ref. 10, video quality is estimated via measures of spatial quality, temporal quality, and spatiotemporal quality for groups of video frames via a 3D Gabor filter-bank; the spatial and temporal components are combined into an overall estimate of quality. In Ref. 17, spatial edge features and motion characteristics in localized space-time regions are used to estimate quality.

Furthermore, it is known that the subjective score of video quality is varying [18] across time during the video display, and this temporal variation has strong influence to the overall quality ratings [19, 20]. VQA models that consider these effects have been proposed in Refs. 20–23. In Ref. 23, Ninassi *et al.* measured temporal variations of spatial visual distortions in a short-term pooling for groups of frames through a mechanism of visual attention; the global video quality score is estimated via a long-term pooling. In Ref. 20, Seshadrinathan *et al.* proposed a hysteresis temporal pooling model of spatial quality values by studying the relation between time-varying quality scores and the final quality score assigned by human subjects.

1.3 Analysis of spatiotemporal slices - A new approach for VQA

Traditional analyses of temporal variation in video quality assessment tend to formulate methods to compute spatial distortion of a standalone frame [3, 12], of local space-time regions [2, 17], or of groups of adjacent frames [10, 23], and then measure the changes of spatial distortion over time. An alternative approach, which is the technique we adopt in this report, is to use *spatiotemporal slices* (as illustrated in Figure 1.1), which allows one to analyze longer temporal variations [24, 25]. In the context of general motion analysis, Ngo *et al.* [25] stated that analyzing the visual patterns of spatiotemporal slices could characterize the changes of motion over time and describe the motion trajectories of different moving objects. Inspired by this result, we develop and propose an algorithm that estimates quality based on the differences between the spatiotemporal slices of the reference and distorted videos.

As shown in Figure 1.1(a), a video can be envisaged as a rectangular cuboid in which two of the sides represent the spatial dimensions (x and y), and the third side represents the time dimension (t). If one takes slices of the cuboid from front-to-back, then the extracted slices correspond to normal video frames. However, it is also possible to take the slices of the cuboid from other directions (e.g., from left-to-right or top-to-bottom) to extract "images" that contain spatiotemporal information, hereafter called the STS images. As shown in Figure 1.1(b), if the cuboid is sliced vertically (left-to-right or right-to-left), then the extracted slices represent time along one dimension and vertical space along the other dimension, hereafter called the *vertical STS images.* If the cuboid is sliced horizontally (top-to-bottom or bottom-to-top), then the extracted slices represent time along one dimension and horizontal space along the other dimension, hereafter called the *horizontal STS images.*

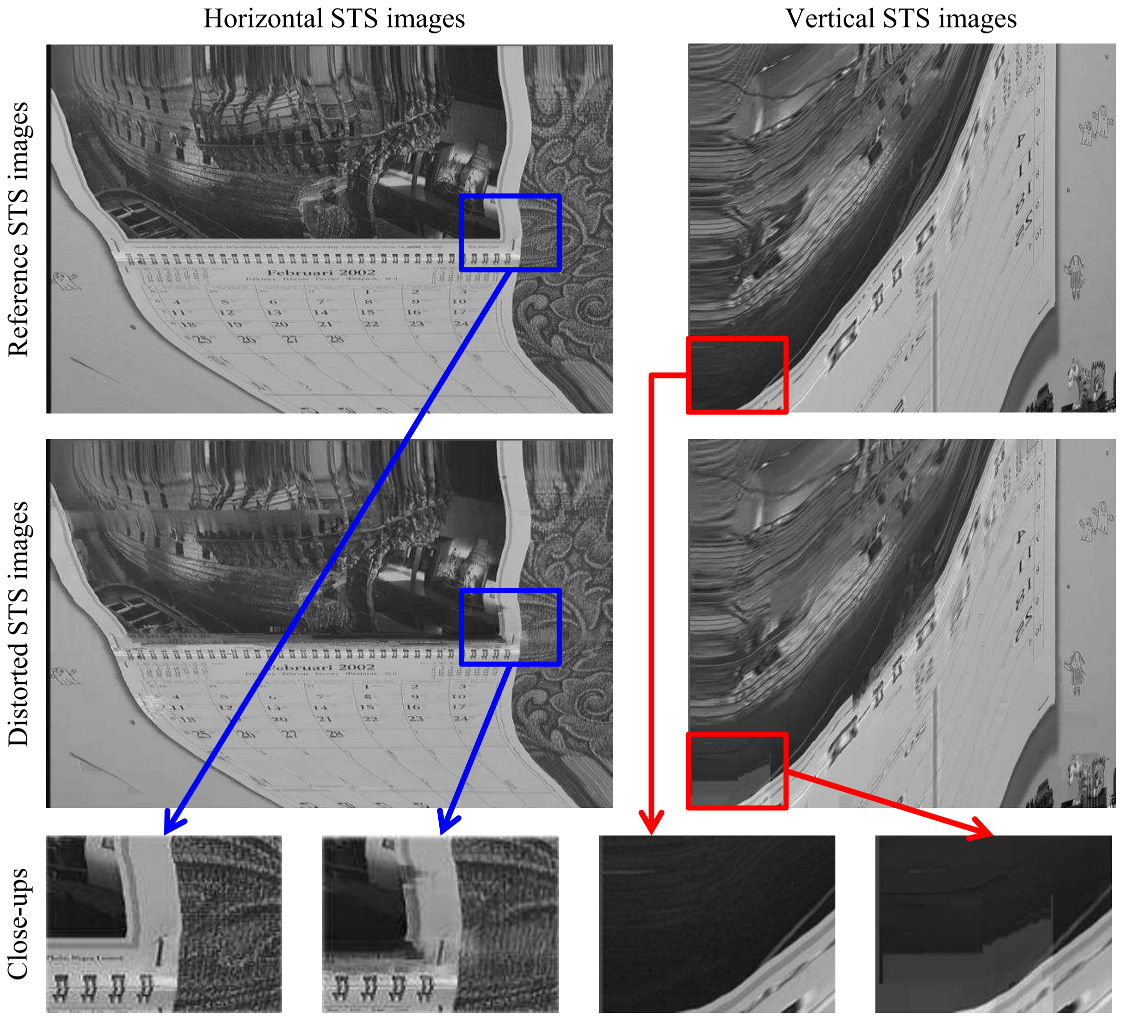
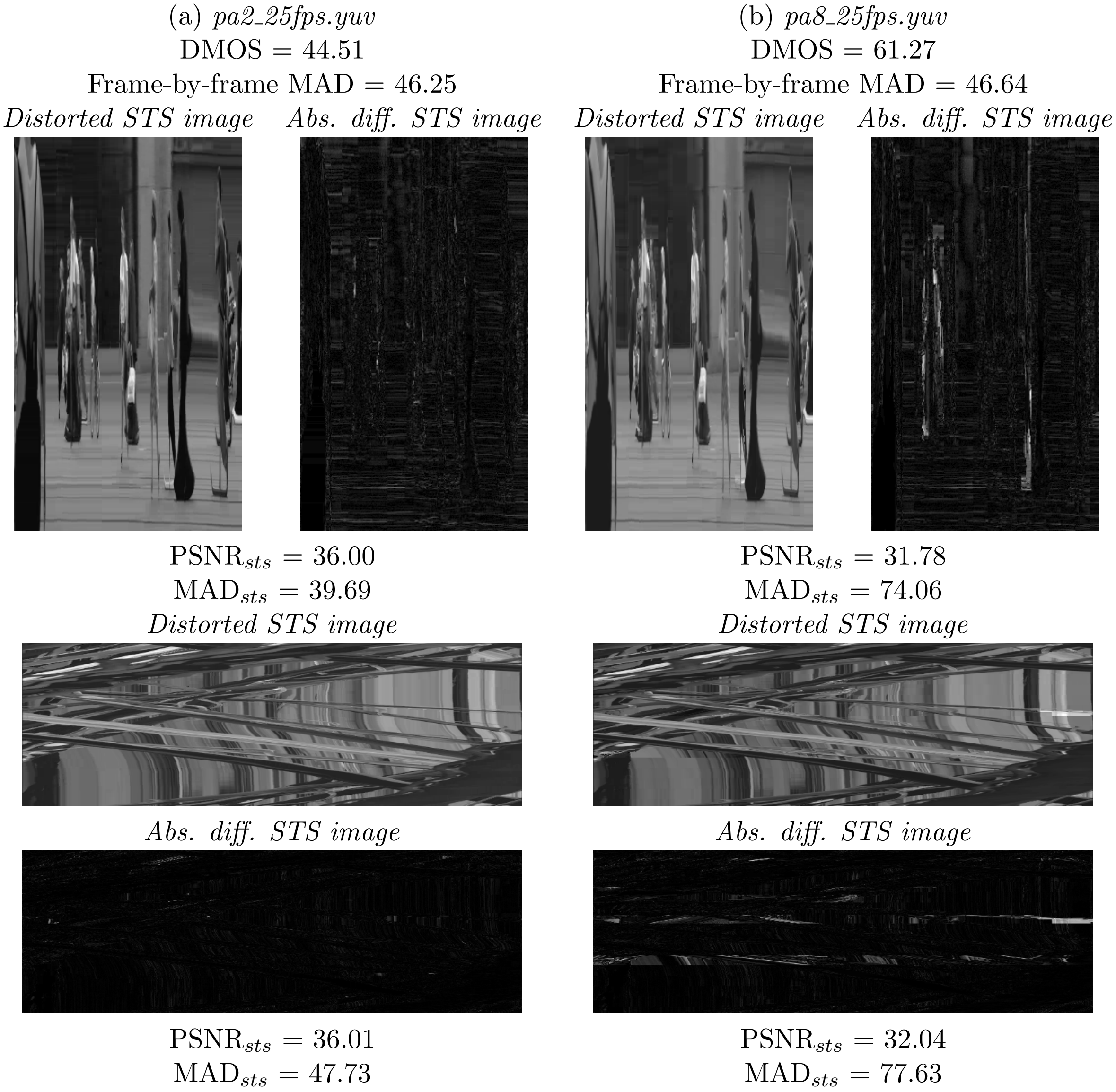


Figure 1.3: Demonstrative STS images extracted from the reference and distorted videos. The close-ups show some dissimilar regions between the STS images.

Figure 1.2 shows examples of STS images from some typical videos. At one extreme, if the video contains no temporal changes (e.g., no motion, as in a static video), then the STS images will contain only horizontal lines [see Figure 1.2(a)] or only vertical lines [see Figure 1.2(b)]. In both Figure 1.2(a) and (b), the perfect temporal relationship in the video content manifests as perfect spatial relationship along the dimension that corresponds to time in the STS images. At the other extreme, if the video is rapidly changing (e.g., each frame contains vastly different content), the STS images will appear as random patterns. In both Figure 1.2(c) and (d), the randomness of temporal content in the video manifests as spatially random pixels along the dimension that corresponds to time in the STS images. The STS images for normal videos [Figure 1.2(e) and (f)] are generally well structured due to the joint spatiotemporal relationship of neighboring pixels and the smooth frame-to-frame transition.

The STS images have been effectively used in a model of human visual-motion sensing [26], in energy models of motion perception [27], and in video motion analysis [24, 25]. Here, we argue that the temporal variation of spatial distortion is exhibited as spatiotemporal dissimilarity in the STS images and thus, these STS images can also be used to estimate video quality. To illustrate this, Figure 1.3 shows sample STS images from a reference video (reference STS image) and from a distorted video (distorted STS image) where some dissimilar regions are clearly visible in the close-ups. As we will demonstrate, by quantifying the spatiotemporal dissimilarity between the reference and distorted STS images, it is possible to estimate video quality.

Figure 1.4 shows sample STS images extracted from two distorted videos of the LIVE video database [1] and the normalized absolute difference images between the reference and distorted STS images. The associated estimates PSNR$_{sts}$ and MAD$_{sts}$ are computed by applying PSNR [28] and the Most Apparent Distortion (MAD) algorithm [29] to each pair of the reference and distorted STS images and by averaging the results across all STS images. The higher the PSNR$_{sts}$ value, the better the video quality; and the lower the MAD$_{sts}$ value, the better the video quality. As seen from Figure 1.4, the PSNR$_{sts}$ and MAD$_{sts}$ values show promise for VQA by comparing the STS images, whereas the frame-by-frame MAD fails to predict the qualities of these videos. However, it is important to note that, although PSNR and MAD show promise when applied to the STS images, neither PSNR nor MAD were designed for use with STS images. In particular, PSNR and MAD do not account for the HVS responses to temporal changes of spatial distortion. Consequently, PSNR$_{sts}$ and MAD$_{sts}$ can yield predictions which correlate poorly with MOS/DMOS. Thus, we



(a) *pa2_25fps.yuv*
DMOS = 44.51
Frame-by-frame MAD = 46.25
Distorted STS image *Abs. diff. STS image*
$\text{PSNR}_{sts} = 36.00$
$\text{MAD}_{sts} = 39.69$
Distorted STS image
Abs. diff. STS image
$\text{PSNR}_{sts} = 36.01$
$\text{MAD}_{sts} = 47.73$

(b) *pa8_25fps.yuv*
DMOS = 61.27
Frame-by-frame MAD = 46.64
Distorted STS image *Abs. diff. STS image*
$\text{PSNR}_{sts} = 31.78$
$\text{MAD}_{sts} = 74.06$
Distorted STS image
Abs. diff. STS image
$\text{PSNR}_{sts} = 32.04$
$\text{MAD}_{sts} = 77.63$

Figure 1.4: Sample STS images and their absolute difference STS images (relative to the STS images of the reference videos) extracted from videos (a) *pa2_25fps.yuv*, (b) *pa8_25fps.yuv* for vertical STS images (upper) and for horizontal STS images (lower). The videos are from the LIVE video database [1]. The values obtained by applying frame-by-frame MAD on normal (front-to-back) frames are shown for comparison. The PSNR_{sts} and MAD_{sts} values, which are computed from the STS images, show promise in estimating video quality. However, neither PSNR nor MAD account for the HVS responses to temporal changes of spatial distortion, and thus we propose an alternative method of quantifying degradation of the STS images.

propose an alternative method of quantifying degradation of the STS images via a measure of correlation and a model of motion perception.

The contents of next chapters in this dissertation are organized as follows: In Chapter 2, we provide a brief literature review of current full-reference VQA algorithms. In Chapter 3, we describe details of our full-reference VQA algorithm named ViS$_3$, which operates based on the analyses of spatial and spatiotemporal slices images. Chapter 4 presents the CSIQ video-quality database, which is developed by our Computation Perception and Image Quality lab. The database consists of more videos, more distortion types, and serves as a trusted data set to validate VQA algorithms. Experimental results on comparing performance of various algorithms on different video databases are presented in Chapter 5. In Chapter 6, we propose our no-reference VQA algorithm that is specifically designed for Motion JPEG2000 videos. General conclusions and potential future research are presented in Chapter 7.

CHAPTER 2

PREVIOUS WORK IN FULL-REFERENCE VQA

The ability to quantify quality of a video is a crucial step for any system that processes digital media. Yet, determining quality in a manner that agrees with human perception remains one of the greatest ongoing challenges in video processing. Current algorithms of image/video quality assessment still face many challenges in predicting of quality because of the presence of various distortion types in images/videos [4, 30]. In this chapter, we provide a brief review of previous work in developing full-reference VQA algorithms. Following the classification specified in Ref. 6, current VQA methods can roughly be divided into four classes: (1) those which employ IQA on a frame-by-frame basis, (2) those which estimate quality based on differences between various features of the reference and distorted videos, (3) those which estimate quality based on statsitical differences between the reference and distorted videos, and (4) those which attempt to model one or more aspects of the HVS.

2.1 Frame-by-frame IQA

As stated in Chapter 1, the most straightforward technique to estimate video quality is to apply existing IQA algorithms on a frame-by-frame basis. These per-frame quality estimates can then be collapsed across time to predict an overall quality estimate of the video. Thus, it is not uncommon to find these frame-by-frame IQA algorithms used as a baseline for comparison [1, 31], and some authors implement this technique as a part of their VQA algorithms [32, 33]. However, due to the lack of temporal information, this technique often fails to correlate with the perceived quality measurements obtained from human observers.

2.2 Algorithms based on visual features

An approach commonly used in VQA is to extract meaningful spatial and temporal visual features of the videos and then estimate quality based on the changes of these features between the reference and distorted videos [2, 16, 34–40].

One of the earliest feature-based VQA algorithm was proposed by Pessoa *et al.*[34]. Their VQA algorithm employs image segmentation and compute error measures with different weights for different types of segment category. Frames of the reference and distorted videos are first segmented into smooth, edge, and texture regions. Various pixel-based and edge-detection-based error measures are then computed between corresponding regions of the reference and distorted videos for both the luminance and chrominance components. These error measures are normalized via a logistic function, weighted based on segment category, and collapsed across all segments and all frames to yield the overall estimate of video quality.

The most popular feature-based VQA algorithms in used is the Video Quality Metric (VQM), which was developed by Pinson and Wolf [2, 16]. The block diagram of the VQM algorithm is depicted in Figure 2.1. The VQM algorithm employs "quality features" that represent spatial, temporal, and color characteristics of video, and measures the differences between those features computed from the reference and distorted videos in four sequential steps. The first step calibrates videos in terms of brightness, contrast, and spatial and temporal shifts. The second step breaks the videos into sub-regions of space and time, and then extracts a set of quality features for each sub-region. The third step compares features extracted from the reference and distorted videos to yield a set of quality indicators. The last step combines these indicators into a video quality index.

Okamoto *et al.*[35] proposed a VQA algorithm, which operates based on the three

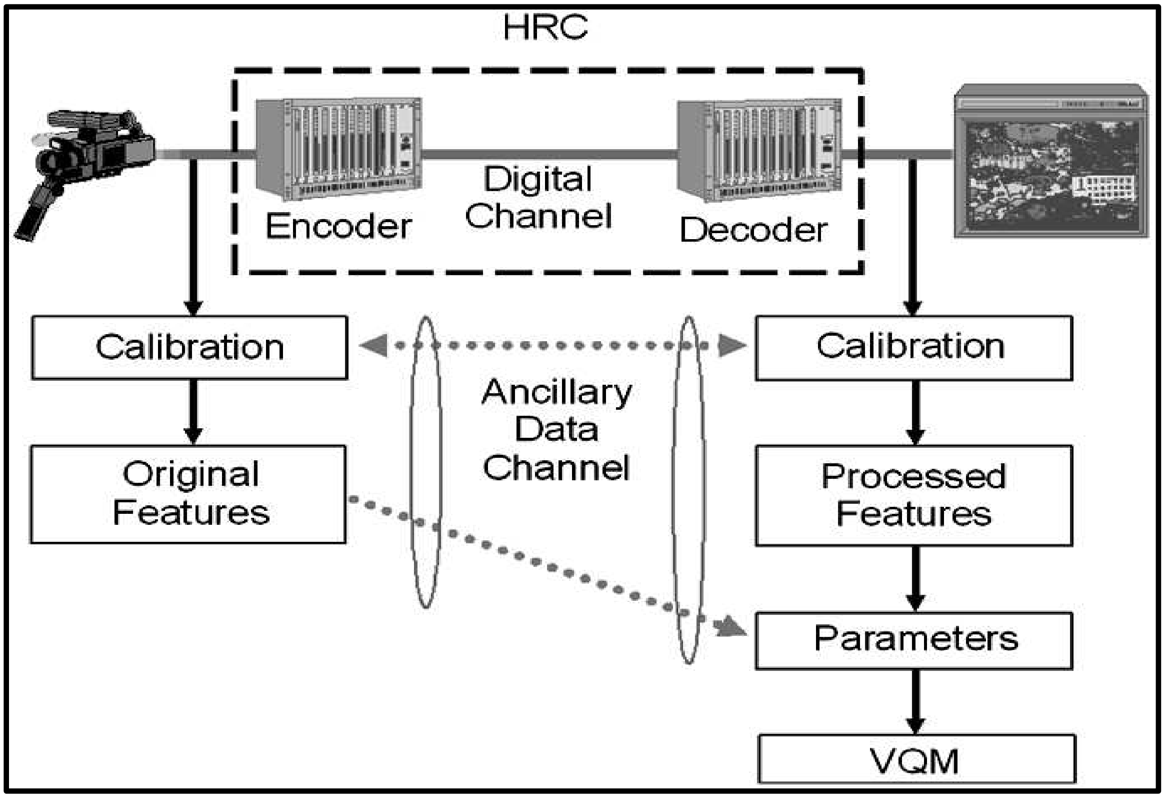


Figure 2.1: Block diagram of the Video Quality Metric (VQM) as presented in Ref. 2.

general types of artifacts appearing in videos: blurring, blocking, and motion distortion. The blurring artifacts in the edge regions are quantified via the average of edge energy difference described in ANSI T1.801.03. The blocking artifacts are quantified based on the ratio of horizontal and vertical edge distortions to other edge distortions, and the average local motion distortion is quantified based on the average difference between block-based motion measures of the reference and distorted frames. The overall video quality is estimated via a weighted average of these three features.

In Ref. 36, Lee and Sim proposed a VQA algorithm that operates under the assumption that the HVS is most sensitive near the locations of edges and block boundaries. Accordingly, their algorithm applies both edge-detection and block-boundary detection to video frames from the reference video to locate these regions. Separate measures of distortion for the edge regions and block regions are then computed between the reference and distorted frames. These two features are supplemented with a gradient-based distortion measure, and the overall estimate of quality is then obtained via a weighted sum of these three features and averaged across all frames.

In the context of packet-loss scenarios, Barkowsky *et al.* [37] designed the TetraVQM algorithm by adding a model of temporal distortion awareness to the VQM algorithm. The key idea in TetraVQM is to estimate the temporal visibility of image areas and assign weight to the degradations in these areas based on their durations. The algorithm employs block-based motion estimation to track image objects over time. The resulting motion vectors and motion-prediction errors are then used to estimate the temporal visibility, which in turn is used as a supplement to VQM algorithm for estimating overall video quality. In 39, Engelke *et al.* demonstrated that significant improvements to VQM and TetraVQM can be realized by augmenting these techniques with information regarding visual saliency.

Various features can be combined with the support of machine learning to improve VQA performance. In Ref. 13, Narwaria *et al.* proposed the Temporal Quality Variation (TQV) algorithm, a low-complexity VQA algorithm that employs a machine-learning mechanism to determine the impact of spatial and temporal factors as well as their interactions on the overall video quality. Spatial quality factors are estimated by an SVD-based algorithm [41] and the temporal variation of spatial quality factors is used as a feature to estimate video quality.

2.3 Algorithms based on statistical measurements

Another class of VQA algorithms has been proposed which estimate quality based on differences in statistical features of the reference and distorted videos [3, 11, 12].

In Ref. 3, Wang *et al.* proposed the Video Structural Similarity (VSSIM) index as depicted in Figure 2.2. VSSIM computes various structural similarity (SSIM [14]) indices at three different levels: the local region level, the frame level, and the video

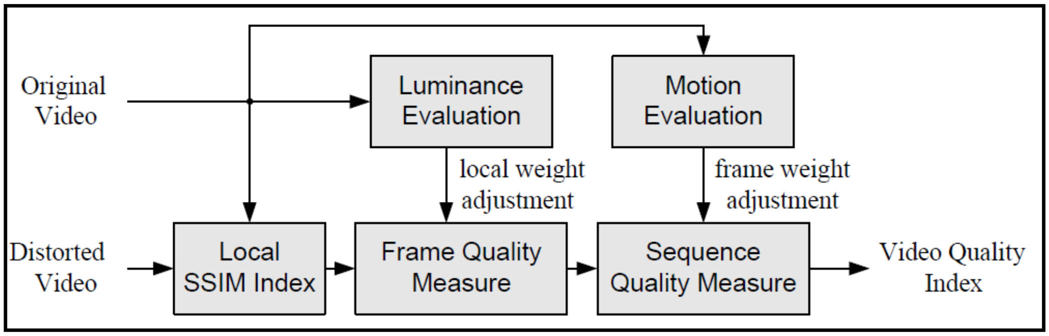


Figure 2.2: Block diagram of the Video Structural SIMilarity (VSSIM) algorithm (Figure from Ref. 3

sequence level. In the local region level, the SSIM index of each region is computed for the luminance and chrominance components, with greater weight is given to luminance component. These SSIM indices are weighted by local luminance intensity to yield the frame-level SSIM index. Finally, at the sequence level, the frame SSIM index is weighted by global motion to yield an estimate of video quality.

Another extension of SSIM to VQA, called Speed SSIM, was also proposed by Wang *et al.*[11]. The authors augmented SSIM with an additional stage that employs Stocker and Simoncelli's statistical model [42] of visual speed perception. The speed perception model is used to derive a "spatiotemporal importance weight function" which specifies a relative weighting at each spatial location and time instant. The overall estimate of video quality is obtained by using this weight function to compute a weighted average of SSIM over all space and time.

In Ref. 12, Sheikh *et al.* augmented the Visual Information Fidelity (VIF) IQA algorithm [15] for use in VQA. VIF estimates quality based on the inferred information that the distorted image provides about the reference image. VIF models images as realizations of a mixture of marginal Gaussian densities of wavelet subbands, and quality is then determined based on the mutual information between the subband coefficients of the reference and distorted images. To account for motion, the Video VIF (V-VIF) algorithm quantifies loss in motion information by measuring deviations in the spatiotemporal derivatives of the videos, which are estimated by using separable bandpass filters in space and time.

Tao and Eskicioglu[33] proposed a VQA algorithm that estimates quality based on the singular value decomposition (SVD). Each frame of the reference and distorted videos are divided into 8 × 8 blocks, and then the SVD is applied to each block. Differences in the SVDs of corresponding blocks from the reference and distorted frames, weighted by the edge-strength in each block, are used to generate a frame-level distortion estimate. Both luminance and chrominance SVD-based distortions are combined via a weighted sum. These combined frame-level estimates are then averaged across all frames to yield an overall estimate of video quality.

Peng *et al.* proposed a motion-tuned and attention-guided VQA algorithm based on a space-time statistical texture representation of motion. To construct the space-time texture representation, the reference and distorted videos are filtered via a bank of 3D Gaussian derivative filters at multiple scales and orientations. Differences in the energies within local regions of the filtered outputs between the reference and distorted videos are then computed along 13 different planes in space-time to define their temporal distortion measure. This measurement is further combined with a model of visual saliency and frame-based Multi-Scale SSIM[43] to estimate quality.

2.4 Algorithms based on HVS models

A widely adopted approach to VQA is to estimate video quality via the use of various models of the human visual system (HVS) [10, 44–55].

One of the earliest VQA algorithms based on a vision model was developed by Lukas and Budrikis [44]. Their technique employs a spatiotemporal visual filter that models visual threshold characteristics on uniform backgrounds. To account for non-uniform backgrounds, the model is supplemented with a masking function based on the spatial and temporal activities of the video.

The Digital Video Quality (DVQ) algorithm, developed by Watson *et al.* [49], also models visual thresholds to estimate video quality. The authors employ the concept of Just Noticeable Differences (JNDs), which are computed via a DCT-based model of early vision. After sampling, cropping, and color conversion, each 8×8 block of the videos is transformed to DCT coefficients, converted to local contrast, and filtered by the temporal contrast sensitivity function. JNDs values are then measured by dividing each DCT coefficient by its respective visual threshold. Contrast masking is estimated based on the differences between successive frames, and the masking-adjusted differences are pooled and mapped to a visual quality estimate.

Other HVS-based VQA algorithms include the Moving Picture Quality Metric (MPQM) [45], the Color Moving Picture Quality Metric (CMPQM) algorithm [46], the Normalization Video Fidelity Metric (NVFM) algorithm [47], wavelet-based algorithm [53, 55], and the MOtion-based Video Integrity Evaluation (MOVIE) algorithm [10]. These algorithms generally simulate HVS responses to individual spatial and temporal subbands of the reference and distorted videos, and then estimate quality based on the extent to which these responses differ. A block diagram of the general approach in these algorithms is presented in Figure 2.3.

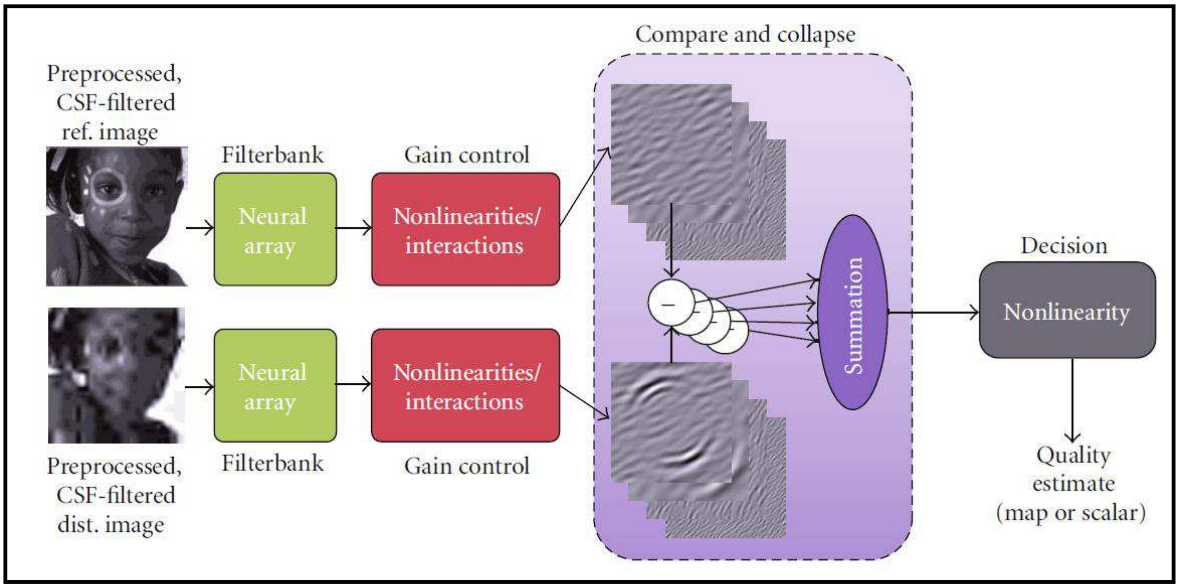

Compare and collapse
Preprocessed, CSF-filtered ref. image
Filterbank
Gain control
Neural array
Nonlinearities/ interactions
Neural array
Nonlinearities/ interactions
Filterbank
Gain control
Preprocessed, CSF-filtered dist. image
−
Summation
Decision
Nonlinearity
Quality estimate (map or scalar)

Figure 2.3: Block diagram of the HVS-based VQA algorithms (Figure from Ref. 4)

The MPQM algorithm, proposed by Basso *et al.* [45], employs a spatiotemporal model of human vision via 17 spatial Gabor filters and two temporal filters on the luminance component only. After contrast sensitivity and masking adjustments, distortion is measured within each subband and pooled to yield the quality estimate. The CMPQM algorithm [46] extends and applies the MPQM algorithm to both luminance and chrominance components with a reduced number of filters for the chrominance components (nine spatial filters and one temporal filter).

The NVFM algorithm [47] implements a visibility prediction model based on Teo-Heeger model [56]. Instead of using Gabor filters, the perceptual decomposition is performed using a steerable pyramid with four scales and four orientations. An excitatory-inhibitory stage and a pooling stage are performed to yield a map of normalized responses. The distortion is measured based on the squared error between normalized response maps generated for the reference and the distorted video.

Masry *et al.* [53] developed a VQA algorithm based on an efficient perceptually motivated multichannel decomposition via a separable wavelet transform. A visual masking model is implemented to account for HVS responses. To obtain optimal masking parameters, a training step was performed on a set of videos and their associated subjective quality scores. Later in Ref. 55, Li *et al.* utilized this algorithm as a part of their VQA algorithm, which measures and combines detail losses and additive impairments within each frame; optimal parameters were determined by training the algorithm on a subset of the LIVE video database [1].

Seshadrinathan *et al.* [10] proposed the MOtion-based Video Integrity Evaluation (MOVIE) algorithm that estimates spatial quality, temporal quality, and spatiotemporal quality via a multi-dimension subband decomposition. MOVIE decomposes both the reference and distorted videos using a 3D Gabor filter-bank with 105 spatiotemporal subbands. The spatial MOVIE component uses outputs of the spatiotemporal Gabor filters and contrast masking to capture spatial distortion. The temporal

MOVIE component employs optical flow motion estimation to determine motion information, which is combined with the outputs of the spatiotemporal Gabor filters to capture temporal distortion. These spatial and temporal components are combined into an overall estimate of video quality.

2.5 Chapter summary

In summary, although previous VQA algorithms have analyzed the effects of spatial and temporal as well as their interactions on video quality, none has estimated video quality based on spatiotemporal slices (STS images), which contain important spatiotemporal information on a longer time scale. Earlier related work was performed by Pechard *et al.* in Ref. 57, where spatiotemporal tubes rather than slices were used for VQA. Their algorithm, which was designed specifically to estimate the impact of H.264 compression artifacts on quality, employs a segmentation to create spatiotemporal tubes that are coherent in terms of motion and spatial activity. In Chapter 3, we will describe our HVS-based VQA algorithm, ViS$_3$, which employs measures of both motion-weighted spatial distortion and spatiotemporal dissimilarity of the STS images to estimate perceived video quality degradation.

CHAPTER 3

VIDEO QUALITY ASSESSMENT VIA ANALYSIS OF SPATIAL AND SPATIOTEMPORAL SLICES

3.1 Introduction

In this chapter, we describe details of our full reference VQA algorithm that estimates video quality degradation by measuring spatial distortion and spatiotemporal dissimilarity separately in two stages. To estimate perceived video quality degradation due to spatial distortion, both the detection-based strategy and the appearance-based strategy of the MAD algorithm [29] are adapted and applied to groups of normal video frames. A simple model of temporal weighting using optical flow motion estimation is employed to give greater weights to distortions in the slow-moving regions [3, 22]. To estimate spatiotemporal dissimilarity, we extend models of Watson-Ahumada [58] and Adelson-Bergen [27], which have been used to measure energy of motion in videos, to the STS images and measure differences in local variance of spatiotemporal neural responses. The spatiotemporal response is obtained by filtering the STS image via one 1D spatial filter and one 1D temporal filter [27, 58]. The overall estimate of perceived video quality degradation is given by a geometric mean of spatial distortion and spatiotemporal dissimilarity values.

We have named our algorithm ViS$_3$ according to its two main stages: the first stage estimates **Vi**deo quality degradation based on **S**patial distortion (ViS$_1$), and the second stage estimates **Vi**deo quality degradation based on the dissimilarity between **S**patiotemporal **S**lices images (ViS$_2$). The final estimate of perceived video quality degradation ViS$_3$ is a combination of ViS$_1$ and ViS$_2$. The ViS$_3$ algorithm is an upgraded version of our previous VQA algorithms presented in Refs. 59, 60. We demonstrate performance of this algorithm on various video-quality databases and compare to some recent VQA algorithms. We also analyze performance of ViS$_3$ on different types of distortion by measuring its performance on each subset of videos.

The major contributions of this algorithm are as follows. First, we provide a simple yet effective extension of our MAD algorithm for use in VQA. Specifically, we show how to apply MAD's detection- and appearance-based strategies to groups of video frames, and how to modify the combination to take into account temporal masking. This contribution is presented in the first stage of the ViS$_3$ algorithm. Second, we demonstrate that the spatiotemporal dissimilarity exhibited in the STS images can be used to effectively estimate video quality degradation. We specifically provide in the second stage of the ViS$_3$ algorithm a technique to quantify the spatiotemporal dissimilarity by measuring spatiotemporal correlation and by applying an HVS-based model to the STS images. Finally, we demonstrate that a combination of the measurements obtained from these two stages is able to estimate video quality quite accurately.

3.2 Algorithm

The ViS$_3$ algorithm estimates video quality degradation by using the luminance components of the reference and distorted videos in YUV color space. We denote $\mathbf{I}$ as the cuboid representation of the Y component of the reference video, and we denote $\hat{\mathbf{I}}$ as the cuboid representation of the Y component of the distorted video.

The ViS$_3$ algorithm employs a combination of both spatial and spatiotemporal analyses to estimate perceived video quality degradation of the distorted video $\hat{\mathbf{I}}$ in comparison to the reference video $\mathbf{I}$. Figure 3.1 shows a block diagram of the ViS$_3$ algorithm, which measures spatial distortion and spatiotemporal dissimilarity separately via two main stages:

- *Spatial Distortion*: This stage estimates average perceived video distortion that

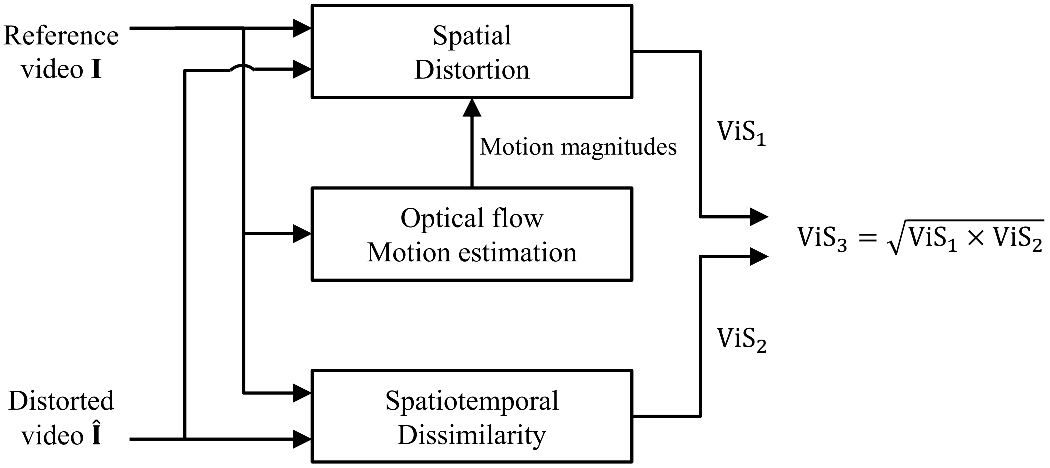

Reference video **I**
Distorted video $\hat{\mathbf{I}}$
Spatial Distortion
Motion magnitudes
Optical flow Motion estimation
Spatiotemporal Dissimilarity
ViS_1
ViS_2
$\mathrm{ViS}_3 = \sqrt{\mathrm{ViS}_1 \times \mathrm{ViS}_2}$

Figure 3.1: Block diagram of the ViS_3 algorithm. The Spatial Distortion stage is applied to groups of normal video frames extracted in a front-to-back fashion to compute spatial distortion value ViS_1. The spatiotemporal dissimilarity value ViS_2 is computed from the STS images extracted in a left-to-right fashion and a top-to-bottom fashion. The final scalar output of the ViS_3 algorithm is computed via a geometric mean of the spatial distortion and spatiotemporal dissimilarity values.

occurs spatially in every group of frames (GOF). A motion-weighting scheme is employed to model the effect of motion on the visibility of spatial distortion. These per-group distortion values are then combined into a single scalar, ViS_1, which represents an estimate of overall perceived video quality degradation due to spatial distortion.

- *Spatiotemporal Dissimilarity*: This stage estimates video quality degradation by computing the spatiotemporal dissimilarity of the STS images extracted from the reference and distorted videos via the differences of spatiotemporal neural responses. These per-STS-image spatiotemporal dissimilarity values are then combined into a single scalar, ViS_2, which represents an estimate of overall perceived video quality degradation due to spatiotemporal dissimilarity.

Finally, the spatial distortion value, ViS_1, and the spatiotemporal dissimilarity value, ViS_2, are combined via a geometric mean to yield a single scalar ViS_3 that represents the overall estimate of perceived video quality degradation. The following

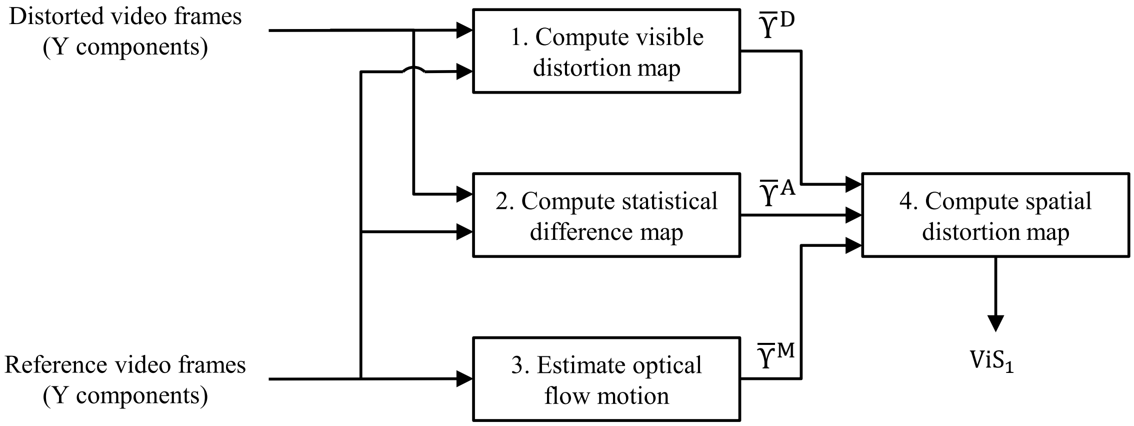

Distorted video frames
(Y components)
Reference video frames
(Y components)
1. Compute visible distortion map
2. Compute statistical difference map
3. Estimate optical flow motion
4. Compute spatial distortion map
$\overline{\Upsilon}^{D}$
$\overline{\Upsilon}^{A}$
$\overline{\Upsilon}^{M}$
ViS_1

Figure 3.2: Block diagram of the Spatial Distortion stage. The extracted frames from the reference and distorted videos are used to compute a visible distortion map and a statistical difference map of each GOF. Motion estimation is performed on the reference video frames and used to model the effect of motion on the visibility of distortion. All maps are combined and collapsed to yield a spatial distortion value ViS_1.

subsections provide details of each stage in the algorithm.

3.2.1 Spatial distortion

In the Spatial Distortion stage, we employ and extend our Most Apparent Distortion (MAD) algorithm [29], which was specifically designed to estimate distortion in the still images, to measure spatial distortion in each GOF of the video. The MAD algorithm is composed of two separate strategies: (1) a detection-based strategy, which computes the perceived distortion due to visual detection (denoted by d_{detect}); and (2) an appearance-based strategy, which computes the perceived distortion due to visual appearance changes (denoted by d_{appear}). The perceived distortion due to visual detection is measured via a masking-weighted block-based mean-squared error in the lightness domain. The perceived distortion due to visual appearance changes is measured by computing the average differences between block-based log-Gabor statistics of the reference and distorted images.

The most apparent distortion (MAD) index of the distorted image is computed via a geometric weighted mean:

$$\alpha = \frac{1}{1 + \beta_1 \times (d_{detect})^{\beta_2}}, \tag{3.1}$$

$$MAD = (d_{detect})^{\alpha} \times (d_{appear})^{1-\alpha}, \tag{3.2}$$

where the weight $\alpha \in [0, 1]$ serves to adaptively combine the two distortion indices (d_{detect} and d_{appear}) based on the overall level of distortion estimated by the detection-based strategy. As described in Ref. 29, for high-quality images, subjects tend to look for distortion and MAD should obtain its value mostly from d_{detect}; whereas for low-quality images, subjects tend to look for image content and MAD should obtain its value mostly from d_{appear}. Thus, an initial estimate of the quality level is required in order to determine the proper weighting (α) of the two strategies. In Ref. 29, the value of d_{detect} served as this initial estimate, and thus α is a function of d_{detect}. The two free parameters $\beta_1 = 0.467$, $\beta_2 = 0.130$ were obtained after training the algorithm on the A57 image database [61]; readers are recommended to see Ref. 29 for a complete description of the MAD algorithm.

To extend MAD for use in video quality assessment, we perform the following steps (shown in Figure 3.2) on each group of N consecutive frames that are taken from the video's luminance (Y) component:

1. Compute a visible distortion map for each frame by using MAD's detection-based strategy. The maps computed from all frames in each GOF are then averaged to yield a GOF-based visible distortion map.

2. Compute a statistical difference map for each frame by using MAD's appearance-based strategy. The maps computed from all frames in each GOF are then averaged to yield a GOF-based statistical difference map.

3. Compute the magnitude of motion vectors in each frame of the reference video,

the motion vectors are estimated via the Lucas-Kanade optical flow method [62]. The motion magnitude maps computed from all frames in each GOF are averaged to yield a GOF-based motion magnitude map.

4. Combine the three GOF-based maps computed from previous steps into a single spatial distortion map; the Root Mean Squared (RMS) value of this map serves as the spatial distortion value of the GOF. The estimated spatial distortion values of all GOFs are combined via an arithmetic mean to yield a single scalar that represents the perceived video quality degradation due to spatial distortion.

Explicitly, the video frames are extracted from the Y components of the reference and distorted videos. Let $I_t(x, y)$ denote the t^{th} frame of the reference video $\mathbf{I}$, and let $\hat{I}_t(x, y)$ denote the t^{th} frame of the distorted video $\hat{\mathbf{I}}$, where $t \in [1, T]$ denotes the frame (time) index, and T denotes the number of frames in video $\mathbf{I}$. These video frames are then divided into groups of N consecutive frames for both the reference and the distorted video. The following subsections describe details of each step.

A Compute visible distortion maps

We apply the detection-based strategy from Ref. 29 to all pairs of respective frames from the reference video and the distorted video. A block diagram of this detection-based strategy is provided in Figure 3.3.

A.1 Detection-based strategy As illustrated in Figure 3.3, a preprocessing step is first performed by using nonlinear luminance conversion and spatial contrast sensitivity function filtering. Then, models of luminance and contrast masking are used to compute a local distortion visibility map where distortions are present. Next, this map is weighted by local MSE to yield a visible distortion map. The specific steps are as follows (see Ref. 29 for additional details):

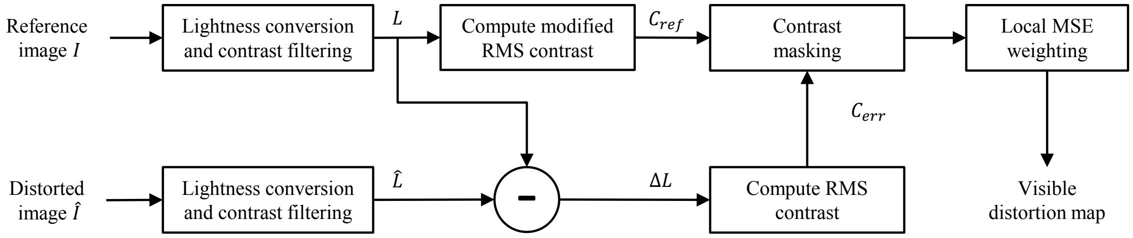

Reference image I
Lightness conversion and contrast filtering
L
Compute modified RMS contrast
C_{ref}
Contrast masking
Local MSE weighting
C_{err}
Distorted image $\hat{I}$
Lightness conversion and contrast filtering
$\hat{L}$
-
ΔL
Compute RMS contrast
Visible distortion map

Figure 3.3: Block diagram of the detection-based strategy used to compute a visible distortion map. Both the reference and the distorted frame are converted to perceived luminance and filtered by a contrast sensitivity function (CSF). By comparing the local contrast of the reference frame L and the error frame ΔL, we obtain a local distortion visibility map. This map is then weighted by local MSE to yield a visible distortion map.

First, to account for the nonlinear relationship between digital pixel values and physical luminance of typical display media, the video $\mathbf{I}$ is converted to a perceived luminance video $\mathbf{L}$ via

$$\mathbf{L} = (a + k\mathbf{I})^{\gamma/3} \tag{3.3}$$

where the parameters a, k and γ are constants specific to the device on which the video is displayed. For 8-bit pixel values and an sRGB display, these parameters are given by $a = 0$, $k = 0.02874$, and $\gamma = 2.2$. The division by 3 attempts to take into account the nonlinear HVS response to luminance by converting luminance into perceived luminance (relative lightness).

Next, the contrast sensitivity function (CSF) is applied by filtering both the reference frame L and the error frame $\Delta L = L - \hat{L}$. The filtering is performed in the frequency domain via

$$\tilde{L} = \mathbb{F}^{-1}[H(u, v) \times \mathbb{F}[L]] \tag{3.4}$$

where $\mathbb{F}$ and $\mathbb{F}^{-1}$ denote the DFT and inverse DFT, respectively; $H(u, v)$ is the DFT-based version of the CSF function defined by Equation (3) in Ref. 29.

To account for the fact that the presence of an image can reduce the detectability of distortions, MAD employs a simple spatial-domain measure of contrast masking. First, a local contrast map is computed for the reference frame in the lightness domain by dividing $\tilde{L}$ into blocks of 16 × 16 pixels (with 75% overlap between neighboring blocks), and then measuring the RMS contrast of each block. The RMS contrast of block b of $\tilde{L}$ is computed via

$$C_{ref}(b) = \tilde{\sigma}_{ref}(b)/\mu_{ref}(b), \tag{3.5}$$

where $\mu_{ref}(b)$ denotes the mean of block b of $\tilde{L}$, and where $\tilde{\sigma}_{ref}(b)$ denotes the minimum of the standard deviations of four 8×8 subblocks of b. The block size of 16×16 was chosen because it is large enough to accommodate division into reasonably sized sub-blocks (to avoid overestimating the contrast around edges), but small enough to yield decent spatial localization (see Appendix A in Ref. 29).

$C_{ref}(b)$ is a measure of local RMS contrast in the reference frame and is thus independent of the distortions. Accordingly, we next compute a local contrast map for the error frame to account for the spatial distribution of distortions in the distorted frame. The error frame ΔL is divided into blocks of 16 × 16 pixels (with 75% block overlapping), and then the RMS contrast $C_{err}(b)$ for each block b is computed via

$$C_{err}(b) = \begin{cases} \sigma_{err}(b)/\mu_{ref}(b) & \text{if } \mu_{ref}(b) > 0.5 \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where $\sigma_{err}(b)$ denotes the standard deviation of block b of ΔL. A lightness threshold value of 0.5 is employed to account for the fact that the HVS is relatively insensitive to changes in extremely dark regions.

The local contrast map is computed for both the reference frame and the error frame for every block b of size 16 × 16 with 75% overlap between neighboring blocks. The two local contrast maps $\{C_{ref}\}$ and $\{C_{err}\}$ are used to compute a local distortion visibility map denoted by $\xi(b)$ via

$$\xi(b) = \begin{cases} \ln(C_{err}(b)) - \ln(C_{ref}(b)) & \text{if } \ln(C_{err}(b)) > \ln(C_{ref}(b)) > -5 \\ \ln(C_{err}(b)) + 5 & \text{if } \ln(C_{err}(b)) > -5 \geq \ln(C_{ref}(b)) \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

The local distortion visibility map ξ is then point-by-point multiplied by the local mean squared-error (MSE) to determine a visible distortion map denoted by Υ^{D}, where the superscript $^{\text{D}}$ is used to imply that the map is computed from the *detection*-based strategy. The visible distortion at the location of block b is given by

$$\Upsilon^{\text{D}}(b) = \xi(b) \cdot \text{MSE}(b). \tag{3.8}$$

Note that in Ref. 29, the visible distortion map Υ^{D} is collapsed into a single scalar that represents the perceived distortion due to visual detection d_{detect}, which is computed via $d_{detect} = \sqrt{\sum_b [\Upsilon^{\text{D}}(b)]^2}$, where the summation is over all blocks. In this dissertation, we do not collapse Υ^{D} at this step.

A.2 Apply to groups of video frames Let Υ^{D}_t denote the visible distortion map computed from the t^{th} frame of the reference video and the t^{th} frame of the distorted video. The visible distortion maps computed from all frames in the k^{th} GOF will be denoted by $\{\Upsilon^{\text{D}}_{N(k-1)+1}, \Upsilon^{\text{D}}_{N(k-1)+2}, \cdots, \Upsilon^{\text{D}}_{Nk}\}$, where $k \in \{1, 2, \cdots, K\}$ is the GOF index and K is the number of GOFs in the video. These maps are combined via a point-by-point average across frames to yield a GOF-based visible distortion map of the k^{th} GOF, which is denoted by $\bar{\Upsilon}^{\text{D}}_k$:

$$\bar{\Upsilon}^{\text{D}}_k = \frac{1}{N} \sum_{\tau=1}^{N} \Upsilon^{\text{D}}_{N(k-1)+\tau}. \tag{3.9}$$

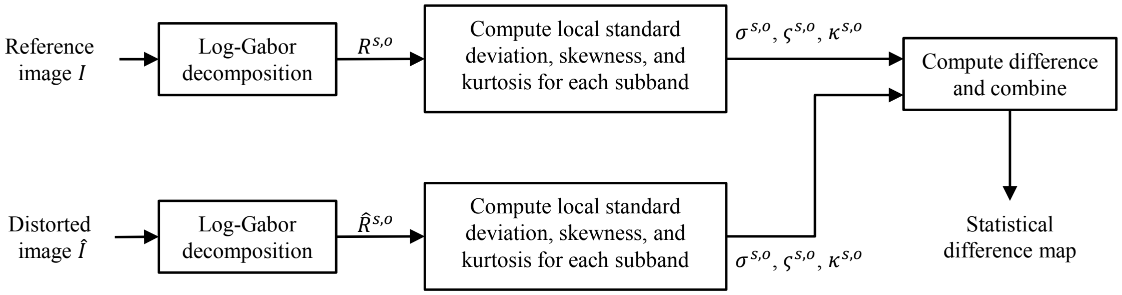

Reference image I
Log-Gabor decomposition
$R^{s,o}$
Compute local standard deviation, skewness, and kurtosis for each subband
$\sigma^{s,o}, \varsigma^{s,o}, \kappa^{s,o}$
Compute difference and combine
Distorted image $\hat{I}$
Log-Gabor decomposition
$\hat{R}^{s,o}$
Compute local standard deviation, skewness, and kurtosis for each subband
$\sigma^{s,o}, \varsigma^{s,o}, \kappa^{s,o}$
Statistical difference map

Figure 3.4: Block diagram of the appearance-based strategy used to compute a statistical difference map. The reference and the distorted frame are decomposed into different subbands using a 2D log-Gabor filter-bank. Local standard deviation, skewness, and kurtosis are computed for each subband of both the reference and the distorted frame. The differences in local standard deviation, skewness, and kurtosis between each subband of the reference frame and the respective subband of the distorted frame are combined into a statistical difference map.

B Compute statistical difference maps

As argued in Ref. 29, when the distortions in the image are highly suprathreshold, perceived distortion is better modeled by quantifying the extent to which the distortions degrade the appearance of the image's subject matter. The appearance-based strategy measures local statistics of multi-scale log-Gabor filter responses to capture changes in visual appearance. Figure 3.4 shows a block diagram of the appearance-based strategy that is employed here to compute a statistical difference map between the reference and the distorted frame.

B.1 Appearance-based strategy The appearance-based strategy employs a computational neural model using a log-Gabor filter-bank (with five scales $s \in \{1, 2, 3, 4, 5\}$ and four orientation $o \in \{1, 2, 3, 4\}$), which implements both even-symmetric (cosine-phase) and odd-symmetric (sine-phase) filters. The even and odd filter outputs are then combined to yield magnitude-only subband values. Let $\{R^{s,o}\}$ and $\{\hat{R}^{s,o}\}$ denote the sets of log-Gabor subbands computed for a reference and a distorted frame, respectively, where each subband has the same spatial size with the frames.

The standard deviation, skewness, and kurtosis are then computed for each block

b of size 16×16 (with 75% overlap between blocks) for each log-Gabor subband of the reference frame and the distorted frame. Let $\sigma^{s,o}(b)$, $\varsigma^{s,o}(b)$, and $\kappa^{s,o}(b)$ denote the standard deviation, skewness, and kurtosis computed from block b of subband $R^{s,o}$. Let $\hat{\sigma}^{s,o}(b)$, $\hat{\varsigma}^{s,o}(b)$, and $\hat{\kappa}^{s,o}(b)$ denote the standard deviation, skewness, and kurtosis computed from block b of subband $\hat{R}^{s,o}$. The statistical difference map is computed as the weighted combination of the differences in standard deviation, skewness, and kurtosis for all subbands. We denote Υ^{A} as the statistical difference map, where the superscript $^{\mathrm{A}}$ is used to imply that the map is computed from the *appearance*-based strategy. Specifically, the statistical difference at the location of block b is given by

$$\Upsilon^{\mathrm{A}}(b) = \sum_{s=1}^{5} \sum_{o=1}^{4} w_s [|\sigma^{s,o}(b) - \hat{\sigma}^{s,o}(b)| + 2|\varsigma^{s,o}(b) - \hat{\varsigma}^{s,o}(b)| + |\kappa^{s,o}(b) - \hat{\kappa}^{s,o}(b)|]. \quad (3.10)$$

where the scale-specific weights $w_s = \{0.5, 0.75, 1, 5, 6\}$ (for the finest to coarsest scales, respectively) are chosen the same as in Ref. 29 to account for the HVS's preference for coarse scales over fine scales (see Ref. 29 for more details).

Note that in Ref. 29, the statistical difference map Υ^{A} is collapsed into a single scalar that represents the perceived distortion due to visual appearance changes d_{appear}, which is computed via $d_{appear} = \sqrt{\sum_b [\Upsilon^{\mathrm{A}}(b)]^2}$, where the summation is over all blocks. In the current dissertation, we do not collapse Υ^{A} at this step.

B.2 Apply to groups of video frames Let Υ_t^{A} denote the statistical difference map computed from the t^{th} frame of the reference video and the t^{th} frame of the distorted video. The statistical difference maps computed from all frames in the k^{th} GOF will be denoted by $\{\Upsilon^{\mathrm{A}}_{N(k-1)+1}, \Upsilon^{\mathrm{A}}_{N(k-1)+2}, \cdots, \Upsilon^{\mathrm{A}}_{Nk}\}$, where $k \in \{1, 2, \cdots, K\}$ is the GOF index and K is the number of GOFs in the video. These maps are combined via a point-by-point average across frames to yield a GOF-based statistical difference map of the k^{th} GOF, which is denoted by $\bar{\Upsilon}^{\mathrm{A}}_{k}$:

$$\bar{\Upsilon}^{\mathrm{A}}_{k} = \frac{1}{N} \sum_{\tau=1}^{N} \Upsilon^{\mathrm{A}}_{N(k-1)+\tau} \tag{3.11}$$

C Optical flow motion estimation

Both the detection-based strategy and the appearance-based strategy were specifically designed for still images. They do not account for the effects of motion on the visibility of distortion. One attribute of motion that affects the visibility of distortion in video is the speed of motion (or the magnitude of motion vectors). According to Wang *et al.* [3] and Barkowsky *et al.* [22], the visibility of distortion is significantly reduced when the speed of motion is large. Alternatively, the distortion in slow-moving regions is more visible than the distortion in fast-moving regions.

To model this effect of motion, we measure the speed of motion in different regions of the video by using an optical flow algorithm. We specifically apply the optical flow method designed by Lucas and Kanade [62] to estimate motion vectors from the reference video. The Lucas-Kanade method assumes that the displacement of the frame contents between two nearby frames is small and roughly constant within a neighborhood (window) of a point under consideration. Thus, the optical-flow motion vector is assumed to be the same within a window centered at that point; and it is computed by solving the optical-flow equations using least squares criterion.

In this dissertation , we select a window of size 8 × 8. For each pair of consecutive frames, we obtain two matrices of motion vectors, M_v and M_h, which correspond to the vertical and horizontal directions. The motion magnitude matrix is then computed as $M = \sqrt{M_v^2 + M_h^2}$. Each element in this matrix represents the motion magnitude of a region defined by an 8 × 8 block in the frame.

Let M_t denote the motion magnitude matrix computed from the t^{th} video frame and its successive frame, where $t = 1, 2, \cdots, T-1$ denotes the frame index and T

is the number of frames in the video. For the k^{th} GOF of the reference video, the motion magnitude matrices computed from all N of its frames are averaged to yield an average motion magnitude matrix via:

$$\bar{M}_k = \frac{1}{N} \sum_{\tau=1}^{N} M_{N(k-1)+\tau}. \tag{3.12}$$

Note that the sizes of M_t and $\bar{M}_k$ are both 64 times smaller than a regular frame because each value in these matrices represents motion magnitude of a 8×8 window in the regular frame. We therefore rescale the $\bar{M}_k$ matrix to the size of the video frame (using nearest-neighbor interpolation) to obtain the GOF-based motion magnitude map of the k^{th} GOF. This map is denoted by $\bar{\Upsilon}_k^{\text{M}}$, where the superscript $^{\text{M}}$ is used to imply that the map is computed from the *motion* magnitudes.

D Combine GOF-based maps and compute spatial distortion value

For each GOF, we have computed the GOF-based visible distortion map $\bar{\Upsilon}^{\text{D}}$, the GOF-based statistical difference map $\bar{\Upsilon}^{\text{A}}$, and the GOF-based motion magnitude map $\bar{\Upsilon}^{\text{M}}$. Now, we extend and apply Equation (3.2) locally to respective regions of the visible distortion map and the statistical difference map to obtain the GOF-based most apparent distortion map. This map is then point-by-point weighted by the motion magnitude map $\bar{\Upsilon}_k^{\text{M}}$ to yield the spatial distortion map of the k^{th} GOF. We denote $\Delta_k(x, y)$ of size $W \times H$, which is the video frame size, as the spatial distortion map of the k^{th} GOF. Specifically, the value at point (x, y) of the spatial distortion map $\Delta_k(x, y)$ is computed via

$$\hat{\alpha}(x, y) = \frac{1}{1 + \beta_1 \times \left[\bar{\Upsilon}_k^{\text{D}}(x, y)\right]^{\beta_2}}, \tag{3.13}$$

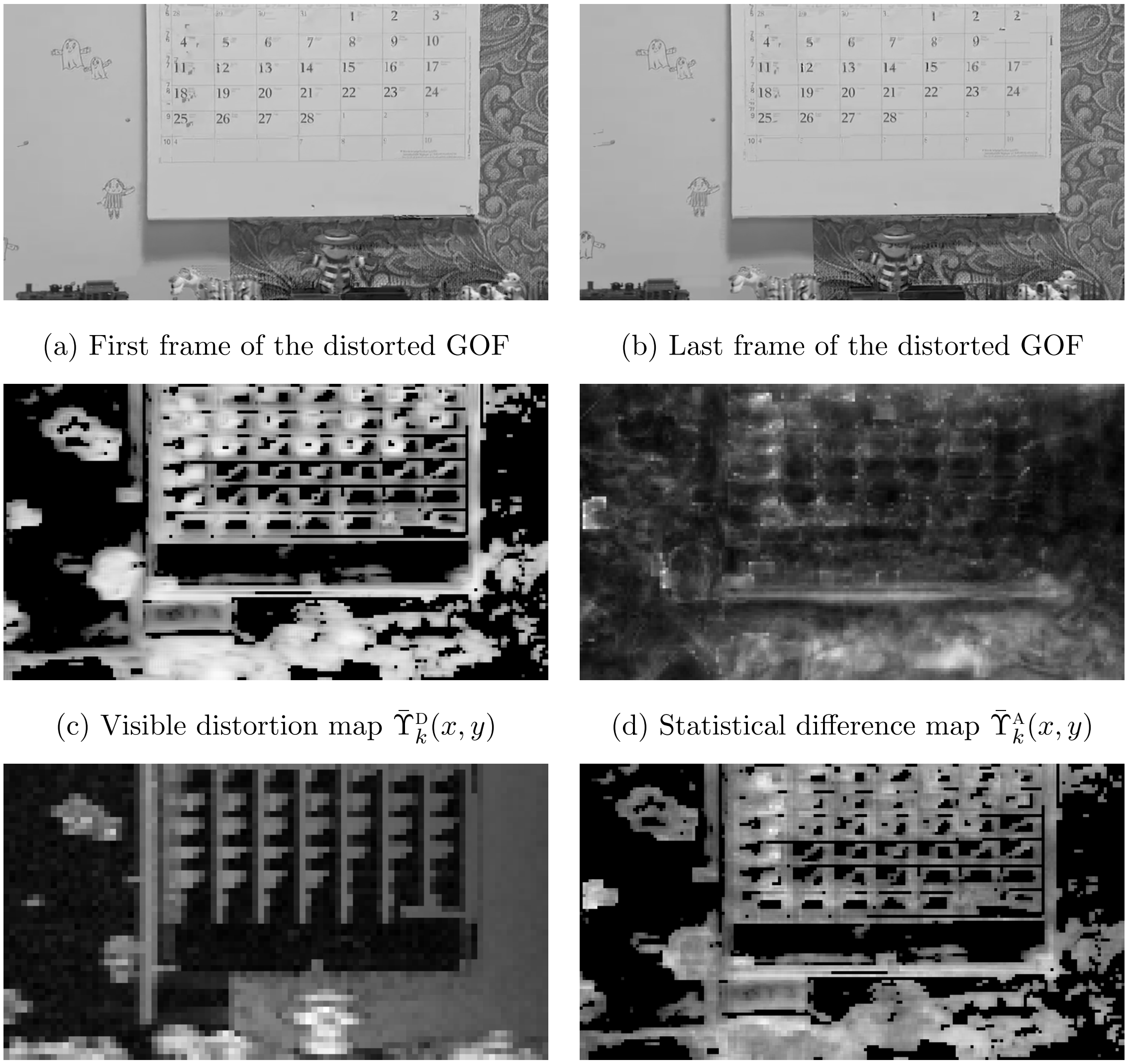
$$\Delta_k(x, y) = \frac{\left[\bar{\Upsilon}_k^{\text{D}}(x, y)\right]^{\hat{\alpha}(x,y)} \times \left[\bar{\Upsilon}_k^{\text{A}}(x, y)\right]^{1-\hat{\alpha}(x,y)}}{\sqrt{1 + \bar{\Upsilon}_k^{\text{M}}(x, y)}}. \tag{3.14}$$

The division by $\bar{\Upsilon}_k^{\text{M}}(x,y)$ accounts for the fact that the distortion in slow-moving regions is generally more visible than the distortion in fast-moving regions. When the value in the motion magnitude map $\bar{\Upsilon}_k^{\text{M}}$ is relatively large or the corresponding spatial region is fast-moving, the visible distortion value in $\Delta_k(x,y)$ is relatively small; when the value in the motion magnitude map $\bar{\Upsilon}_k^{\text{M}}$ is relatively small or the corresponding spatial region is slow-moving, the visible distortion value in $\Delta_k(x,y)$ is relatively large. When there is no motion in the region, the visible distortion is determined solely by frame-based visual detection $\bar{\Upsilon}_k^{\text{D}}$ and visual appearance change $\bar{\Upsilon}_k^{\text{A}}$.

Figure 3.5 shows examples of the first frame (a) and the last frame (b) of a specific GOF of video $mc2\_50fps.yuv$ from the LIVE video database [1]. The visible distortion map (c), the statistical difference map (d), the motion magnitude map (e), and the spatial distortion map (f) computed for this GOF are also shown. As seen from the visible distortion map (c) and the statistical difference map (d), at the regions of high visible distortion level (i.e. the train, the numbers in the calendar), the spatial distortion map is weighted more by the statistical difference map. At the regions of low visible distortion level (i.e. the wall background), the spatial distortion map is weighted more by the visible distortion map.

As also seen from Figure 3.5(c) and (d), the region corresponding to the train at the bottom of the frames is more heavily distorted than the other regions. However, due to the fast movement of the train, which is reflected in the bottom of the motion magnitude map (e), the visibility of distortion is reduced, making this region less bright in the spatial distortion map (f).

To estimate spatial distortion value of each GOF, we compute the root mean square (RMS) value of the spatial distortion map. The RMS value of the map $\Delta_k(x,y)$



(a) First frame of the distorted GOF

(b) Last frame of the distorted GOF

(c) Visible distortion map $\bar{\Upsilon}_k^{\mathrm{D}}(x, y)$

(d) Statistical difference map $\bar{\Upsilon}_k^{\mathrm{A}}(x, y)$

(e) Motion magnitude map $\bar{\Upsilon}_k^{\mathrm{M}}(x, y)$

(f) Spatial distortion map $\Delta_k(x, y)$

Figure 3.5: Examples of the first and last frames (a, b), the visible distortion map (c), the statistical difference map (d), the motion magnitude map (e), and the spatial distortion map (f) computed for a specific GOF of the video $mc2\_50fps.yuv$ from the LIVE video database [1]. All maps have been normalized in contrast to promote visibility. Note that the brighter the maps, the more distorted the corresponding spatial region of the GOF; for the motion magnitude map, the brighter the map, the faster the motion in the corresponding spatial region of the GOF.

of size $W \times H$ is given by

$$\bar{\Delta}_k^{\mathrm{XY}} = \sqrt{\frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \left[\Delta_k(x, y)\right]^2}, \tag{3.15}$$

where the superscript $^{\mathrm{XY}}$ is used to remind readers that the value is computed from the normal frames with two dimensions x and y. The overall perceived spatial distortion value, denoted by ViS_1, is computed as the arithmetic mean of all spatial distortion values $\bar{\Delta}_k^{\mathrm{XY}}$ across GOF via

$$\mathrm{ViS}_1 = \frac{1}{K} \sum_{k=1}^{K} \bar{\Delta}_k^{\mathrm{XY}}. \tag{3.16}$$

Here, ViS_1 is a single scalar that represents the overall perceived quality degradation of the distorted video due to spatial distortion in comparing to the reference video. The lower the ViS_1 value, the better the video quality. A value $\mathrm{ViS}_1 = 0$ indicates that the distorted video is equal in quality to the reference video.

3.2.2 Spatiotemporal dissimilarity

In the distorted video, the distortion does not only impact the spatial relationship between neighboring pixels within the current frame, the distortion can also affect the transition between frames, which can be captured via the use of the STS images as demonstrated in Chapter 1. The difference between STS images from the reference and distorted videos is referred to as the *spatiotemporal dissimilarity* in this dissertation. If the spatiotemporal dissimilarity between the STS images is small, the distorted video has high quality relative to the reference video; if the spatiotemporal dissimilarity between the STS images is large, the distorted video has low quality relative to the reference video. Figure 3.6 depicts a block diagram of the Spatiotemporal Dissimilarity stage, which estimates spatiotemporal dissimilarity between the reference and the distorted video via following steps:

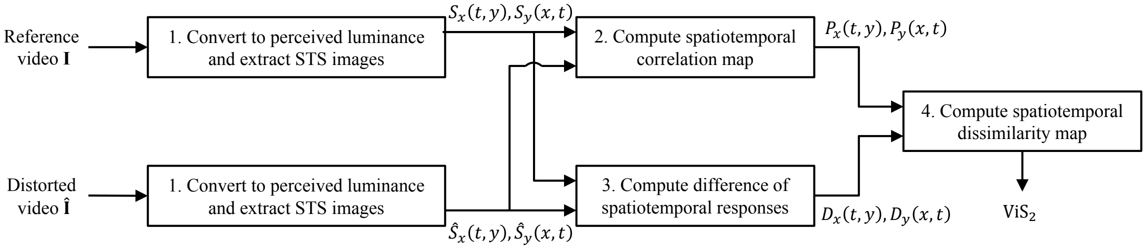

Reference video $\mathbf{I}$
Distorted video $\hat{\mathbf{I}}$
1. Convert to perceived luminance and extract STS images
1. Convert to perceived luminance and extract STS images
$S_x(t,y), S_y(x,t)$
$\hat{S}_x(t,y), \hat{S}_y(x,t)$
2. Compute spatiotemporal correlation map
3. Compute difference of spatiotemporal responses
$P_x(t,y), P_y(x,t)$
$D_x(t,y), D_y(x,t)$
4. Compute spatiotemporal dissimilarity map
ViS_2

Figure 3.6: Block diagram of the Spatiotemporal Dissimilarity stage of the ViS_3 algorithm. The STS images are extracted from the perceived luminance videos. The spatiotemporal correlation and the difference of spatiotemporal responses are computed in a block-based fashion and combined to yield a spatiotemporal dissimilarity map. All maps are then collapsed by using root mean square and combined to yield the spatiotemporal dissimilarity value ViS_2 of the distorted video.

1. Extract the vertical and horizontal STS images in the lightness domain.
2. Compute a spatiotemporal correlation map of the STS images.
3. Filter the STS images via a set of spatiotemporal filters. These filtered images are used to compute a map of spatiotemporal responses differences.
4. Combine the two above maps into a spatiotemporal dissimilarity map and collapse this map into a spatiotemporal dissimilarity value. These per-STS-image dissimilarity values are combined into a single scalar, ViS_2, which represents the overall perceived video spatiotemporal dissimilarity.

The following subsections describe details of each step.

A Extract the STS images

The reference video $\mathbf{I}$ and the distorted video $\hat{\mathbf{I}}$ are converted to perceived luminance videos $\mathbf{L}$ and $\hat{\mathbf{L}}$ respectively using Equation (3.3). Let $S_x(t, y)$ denote the vertical STS image of the video cuboid $\mathbf{L}$, where $x \in [1, W]$ denotes the vertical slice (column) index, and W denotes the spatial width of the video (measured in pixels). As shown previously in Figure 1.1, these vertical STS images contain temporal information in the horizontal direction and spatial information in the vertical direction. Thus, for a video containing T frames, $S_x(t, y)$ will be of size $T \times H$, where H denotes the spatial height of the video (measured in pixels). There are W such STS images $S_1(t, y), S_2(t, y), \cdots, S_W(t, y)$.

Similarly, let $S_y(x, t)$ denote the horizontal STS image of the video cuboid $\mathbf{L}$, where $y \in [1, H]$ denotes the horizontal slice (row) index, and H denotes the spatial height of the video. These horizontal STS images contain spatial information in the vertical direction and temporal information in the horizontal direction. Thus, for a video containing T frames, $S_y(x, t)$ will be of size $W \times T$, and there are H such STS images $S_1(x, t), S_2(x, t), \cdots, S_H(x, t)$.

The STS images extracted from the reference video $\{S_x(t, y),\ S_y(x, t)\}$ and the STS images extracted from the distorted video $\{\hat{S}_x(t, y),\ \hat{S}_y(x, t)\}$ are then used to compute the spatiotemporal dissimilarity values. This procedure consists of two main steps: (1) compute the spatiotemporal correlation maps, and (2) compute the spatiotemporal response difference maps.

B Compute spatiotemporal correlation maps

One simple way that can potentially measure the spatiotemporal dissimilarity is computing the local linear correlation coefficients of the STS images extracted from the reference and the distorted videos. If the distorted video has perfect quality relative to the reference video, these two videos should have high correlation in the STS images; if the distorted video has low quality relative to the reference video, the spatiotemporal correlation is likely to be low.

Let $\rho(b)$ denote the linear correlation coefficient computed from block b of the two STS images $S_x(t, y)$ and $\hat{S}_x(t, y)$. We define the local spatiotemporal correlation coefficient $\tilde{\rho}(b)$ of these two blocks as

$$\tilde{\rho}(b) = \begin{cases} 0 & \text{if } \rho(b) < 0 \\ 1 & \text{if } \rho(b) > 0.9 \\ \rho(b) & \text{otherwise.} \end{cases} \tag{3.17}$$

As shown in Equation (3.17), if the two blocks are highly positive correlated, we set $\tilde{\rho}(b) = 1$. The threshold value of 0.9 was chosen empirically so that a relatively high positive correlation ($\rho > 0.9$) is still considered perfect by the algorithm. On the other hand, if the two blocks are negatively correlated, we set $\tilde{\rho}(b) = 0$ to reflect the dissimilarity between the two blocks.

This process is performed on every block of size 16×16 with 75% overlap between neighboring blocks, yielding a spatiotemporal correlation map denoted by $P_x(t, y)$ between $S_x(t, y)$ and $\hat{S}_x(t, y)$. Similarly, we compute a spatiotemporal correlation map denoted by $P_y(x, t)$ between $S_y(x, t)$ and $\hat{S}_y(x, t)$. Examples of the correlation maps are shown in Figure 3.7(c). The brighter the maps, the higher the spatiotemporal correlation between corresponding regions of the two STS images.

C Compute spatiotemporal response difference maps

The spatiotemporal correlation coefficient computed in previous section does not account for the response of human visual system to the joint spatiotemporal characteristics of the video. Therefore, in addition to measuring the spatiotemporal correlation, we employ a computational HVS model that takes into account joint spatiotemporal perception based on the work of Watson and Ahumada in Ref. 58. This model applies separate 1D filters to each dimension of the STS images to measure spatiotemporal responses. In Ref. 27, Adelson *et al.* used these spatiotemporal responses to measure energy of motion in a video. Here, we apply similar model with different spatial filters to the STS images and measure the differences of spatiotemporal responses in an attempt to estimate video quality degradation.

C.1 Decompose STS images into spatiotemporally filtered images As stated by Adelson and Bergen in Ref. 27, the spatiotemporal information presented in the STS images can be captured via a set of spatiotemporally oriented filters. Watson and Ahumada [58] suggested to construct these filters via two sets of separate 1D filters (spatial and temporal) with appropriate spatiotemporal characteristics. Following this suggestion, we employ a set of log-Gabor 1D filters $\{g_s\}$, $s \in \{1, 2, 3, 4, 5\}$, as the spatial filters, where the frequency response of each filter is given by

$$G_s(\omega) = \exp\left(-\frac{(\ln|\frac{\omega}{\omega_s}|)^2}{2(\ln B_s)^2}\right), \tag{3.18}$$

where G_s, ω_s, and B_s denote the frequency response, center frequency, and bandwidth of the filter g_s respectively, $\omega \in [-\omega_s, \omega_s]$ is the 1D spatial frequency. The bandwidth B_s is held constant for all scales to obtain constant filter shape. We specifically choose five scales and a filter bandwidth of approximately two octaves ($B_s = 0.55$). Without the orientation information, these filters are the same as the log-Gabor filters used the appearance-based strategy of Ref. 29 .

Two temporal filters $\{h_z\}$, $z \in \{1, 2\}$, were selected according to Adelson-Bergen model [27]. The impulse response at time instance t of each filter is given by

$$h_z(t) = t^{n_z} \exp(-t)\left[\frac{1}{n_z!} - \frac{t^2}{(n_z+2)!}\right] \tag{3.19}$$

where $n_1 = 6$ and $n_2 = 9$,which correspond to the fast and slow motion, were chosen to approximate the temporal contrast sensitivity functions reported by Robson [63].

The STS images are filtered along the spatial dimension by each spatial filter and then along the temporal dimension by each temporal filter to yield a spatiotemporally filtered image, which represents modeled spatiotemporal neural responses. With five spatial filters and two temporal filters, each STS image yields 10 spatiotemporally filtered images. We denote $R_x^{s,z}(t,y)$ and $R_y^{s,z}(x,t)$, ($s \in \{1,2,3,4,5\}$ and $z \in \{1,2\}$), as the spatiotemporally filtered images obtained by filtering the STS images $S_x(t,y)$ and $S_y(x,t)$ from the reference video via spatial filter g_s and temporal filter h_z. These filtered images are computed via

$$R_x^{s,z}(t,y) = [S_x(t,y) *^y g_s] *^t h_z \tag{3.20}$$

$$R_y^{s,z}(x,t) = [S_y(x,t) *^x g_s] *^t h_z \tag{3.21}$$

where $*^d$, $d \in \{x,y,t\}$, denotes the convolution operator along dimension d.

Similarly, we denote $\hat{R}_x^{s,z}(t,y)$ and $\hat{R}_y^{s,z}(x,t)$ as the spatiotemporally filtered images obtained by filtering the STS images $\hat{S}_x(t,y)$ and $\hat{S}_y(x,t)$ from the distorted video via spatial filter g_s and temporal filter h_z. Then, the spatiotemporal response differences $\Delta R_x^{s,z}(t,y)$ and $\Delta R_y^{s,z}(x,t)$ are defined as the absolute difference of the spatiotemporally filtered images via

$$\Delta R_x^{s,z}(t,y) = |R_x^{s,z}(t,y) - \hat{R}_x^{s,z}(t,y)| \tag{3.22}$$

$$\Delta R_y^{s,z}(x,t) = |R_y^{s,z}(x,t) - \hat{R}_y^{s,z}(x,t)|. \tag{3.23}$$

Although the proper technique of estimating video quality based on the differences in spatiotemporal responses remains an open research question, as discussed next, we employ a simple yet effective measure based on the local standard deviation of the spatiotemporal response differences.

C.2 Compute log of response difference maps We compute the local mean and local standard deviation of the spatiotemporal response differences in a block-based fashion. Let $\mu_x^{s,z}(b)$ and $\sigma_x^{s,z}(b)$ denote the local mean and standard deviation computed from block b of the response difference $\Delta R_x^{s,z}(t,y)$. Similarly, let $\mu_y^{s,z}(b)$

and $\sigma_y^{s,z}(b)$ denote the local mean and local standard deviation computed from block b of the response difference $\Delta R_y^{s,z}(x,t)$.

The adjusted standard deviation of block b of the error-filtered image at spatial frequency index s and temporal frequency index z is given by

$$\tilde{\sigma}_x^{s,z}(b) = \begin{cases} 0, & \text{if } \mu_x^{s,z}(b) < p \\ \sigma_x^{s,z}(b) \times \sqrt{\dfrac{\mu_x^{s,z}(b)}{p + \mu_x^{s,z}(b)}}, & \text{otherwise} \end{cases} \tag{3.24}$$

and

$$\tilde{\sigma}_y^{s,z}(b) = \begin{cases} 0, & \text{if } \mu_y^{s,z}(b) < p \\ \sigma_y^{s,z}(b) \times \sqrt{\dfrac{\mu_y^{s,z}(b)}{p + \mu_y^{s,z}(b)}}, & \text{otherwise} \end{cases} \tag{3.25}$$

where $p = 0.01$ is a threshold value. When the mean value of absolute difference at the location of block b is small, there is no dissimilarity between the regions at the location of block b in the STS images; when the mean value computed from block b is large enough, the dissimilarity is approximately measured by the standard deviation of block b in the response differences.

This process is performed on every block of size 16 × 16 with 75% overlap between neighboring blocks, yielding maps of adjusted standard deviation $\tilde{\sigma}_x^{s,z}(t,y)$ and $\tilde{\sigma}_y^{s,z}(x,t)$. The log of response difference maps $D_x(t,y)$ and $D_y(x,t)$ are computed as the natural logarithm of a weighted sum of all the maps $\tilde{\sigma}_x^{s,z}(t,y)$ and $\tilde{\sigma}_y^{s,z}(x,t)$, respectively, as follows:

$$D_x(t,y) = \ln\left(1 + A \sum_{s=1}^{5} \sum_{z=1}^{2} w_s [\tilde{\sigma}_x^{s,z}(t,y)]^2\right) \tag{3.26}$$

$$D_y(x,t) = \ln\left(1 + A \sum_{s=1}^{5} \sum_{z=1}^{2} w_s [\tilde{\sigma}_y^{s,z}(x,t)]^2\right) \tag{3.27}$$

where the weights $\{w_s\} = \{0.5,\ 0.75,\ 1,\ 5,\ 6\}$ were chosen following Ref. 29 to account for the preference of human visual system for coarse scales over fine scales.

The addition of one is to prevent the logarithm of zero, and $A = 10^4$ is a scaling factor to enlarge the adjusted variance. Examples of the log of response difference maps are shown in Figure 3.7(d). The brighter the maps, the greater the difference in spatiotemporal responses between corresponding regions of the two STS images.

D Compute spatiotemporal dissimilarity value

The spatiotemporal correlation map P and the log of response difference map D are combined into a spatiotemporal dissimilarity map via a point-by-point multiplication

$$\Delta_x(t, y) = D_x(t, y) \cdot \sqrt{1 - P_x(t, y)} \tag{3.28}$$

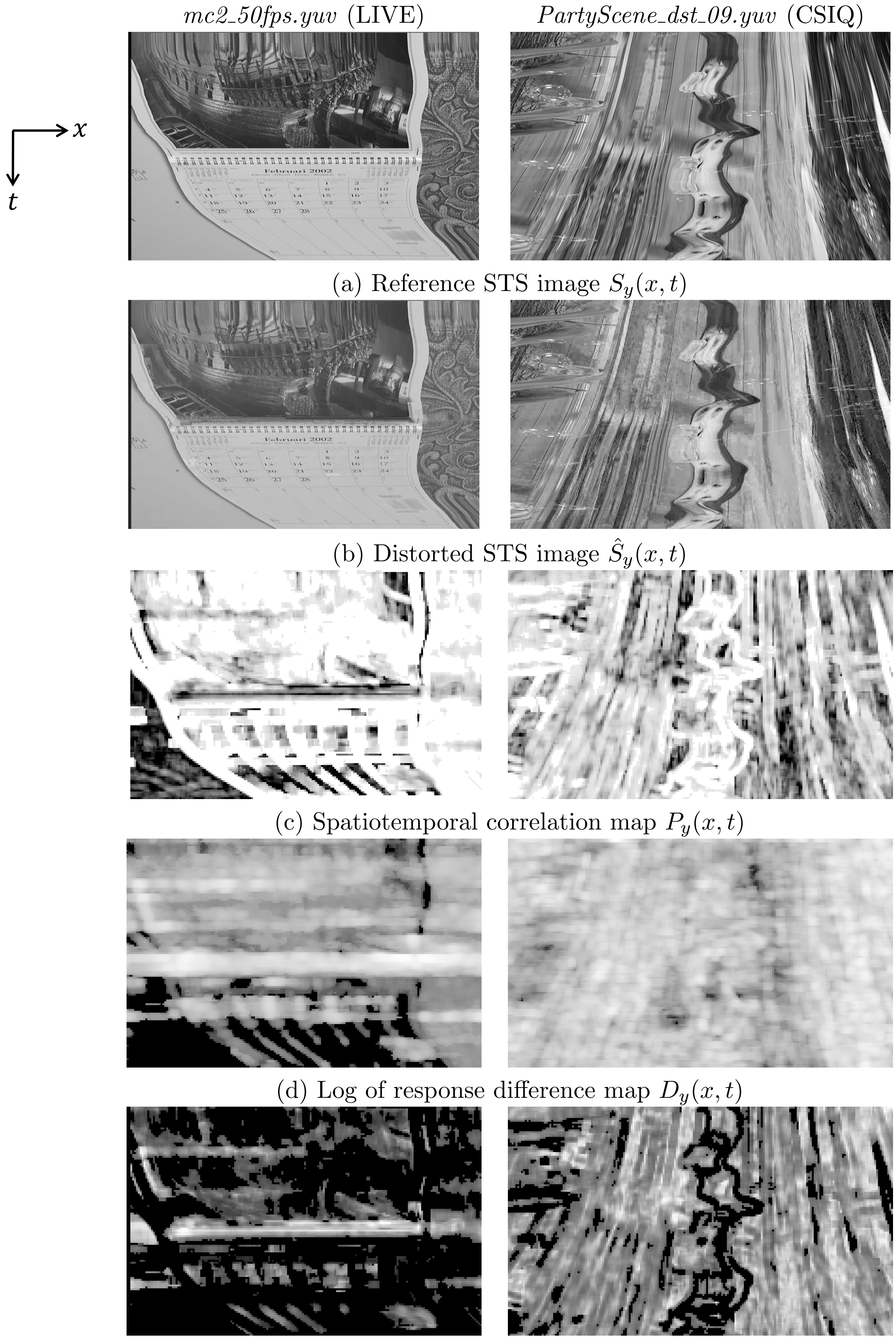
$$\Delta_y(x, t) = D_y(x, t) \cdot \sqrt{1 - P_y(x, t)}. \tag{3.29}$$

Let $\bar{\Delta}_c^{\text{TY}}$ denote the RMS value of the spatiotemporal dissimilarity map $\Delta_c(t, y)$ of size $T \times H$, where c is the column (vertical slice) index of the vertical STS images. Let $\bar{\Delta}_r^{\text{XT}}$ denote the RMS value of the spatiotemporal dissimilarity map $\Delta_r(x, t)$ of size $W \times T$, where r is the row (horizontal slice) index of the horizontal STS images. Specifically, these RMS values are computed as follows

$$\bar{\Delta}_c^{\text{TY}} = \sqrt{\frac{1}{T \times H} \sum_{t=1}^{T} \sum_{y=1}^{H} [\Delta_c(t, y)]^2}, \tag{3.30}$$

$$\bar{\Delta}_r^{\text{XT}} = \sqrt{\frac{1}{W \times T} \sum_{x=1}^{W} \sum_{t=1}^{T} [\Delta_r(x, t)]^2}, \tag{3.31}$$

where W and H are the spatial width and height of the video frame respectively, and T is number of frames in the videos. The superscripts $^{\text{TY}}$ and $^{\text{XT}}$ are used to remind readers about the two dimensions of the STS images that are used to compute the values. The spatiotemporal dissimilarity value, denoted by ViS_2, between the



(a) Reference STS image $S_y(x,t)$

(b) Distorted STS image $\hat{S}_y(x,t)$

(c) Spatiotemporal correlation map $P_y(x,t)$

(d) Log of response difference map $D_y(x,t)$

(e) Spatiotemporal dissimilarity map $\Delta_y(x,t)$

Figure 3.7: Demonstrative maps for two pairs of STS images $S_y(x,t)$ and $\hat{S}_y(x,t)$ from video *mc2_50fps.yuv* (LIVE) and *PartyScene_dst_09.yuv* (CSIQ) with the correlation maps $P_y(x,t)$, the log of response difference maps $D_y(x,t)$, and spatiotemporal dissimilarity maps $\Delta_y(x,t)$. All maps have been normalized to promote visibility. Note that the brighter the spatiotemporal dissimilarity maps $\Delta_y(x,t)$, the more dissimilar the corresponding regions in the STS images.

reference and the distorted video is given by

$$\mathrm{ViS}_2 = \sqrt{\frac{1}{W}\sum_{c=1}^{W}\left[\bar{\Delta}_c^{\mathrm{TY}}\right]^2 + \frac{1}{H}\sum_{r=1}^{H}\left[\bar{\Delta}_r^{\mathrm{XT}}\right]^2}. \tag{3.32}$$

Here, ViS_2 is a single scalar that represents the overall perceived video quality degradation due to spatiotemporal dissimilarity. The lower the ViS_2 value, the better the video quality. A value $\mathrm{ViS}_2 = 0$ indicates that the distorted video has perfect quality relative to the reference video.

Figure 3.7 shows the correlation maps $P_y(x,t)$, the log of response difference maps $D_y(x,t)$, and the spatiotemporal dissimilarity maps $\Delta_y(x,t)$ computed from two pairs of specific horizontal STS images. These maps are normalized to promote visibility. The brighter values in the spatiotemporal dissimilarity maps $\Delta_y(x,t)$ in Figure 3.7(e) denote the corresponding spatiotemporal regions of greater dissimilarity.

As observed from video *mc2_50fps.yuv* (LIVE), the spatial distortion occurs more frequently in the middle frames. These middle frames are also heavily distorted in nearly every spatial region. This fact is well-captured by the spatiotemporal dissimilarity map in Figure 3.7(e) (*left*). As observed in Figure 3.7(e) (*left*), the dissimilarity map is brighter in the middle of the map and along the entire spatial dimension. In video *PartyScene_dst_09.yuv* (CSIQ), the spatial distortion that occurs in the center of the video is smaller than the distortion in the surrounding area. This fact is also reflected in the spatiotemporal dissimilarity map in Figure 3.7(e) (*right*), where the spatiotemporal dissimilarity map shows brighter surrounding regions compared to the center regions across the temporal dimension.

3.2.3 Combine two prediction values

Finally, the overall estimate of perceived video quality degradation, denoted by ViS_3, is computed from the spatial distortion ViS_1 and the spatiotemporal dissimilarity

ViS_2. The optimal combination of ViS_1 and ViS_2 remains an area of future research. One possible solution is using an adaptive geometric weighted mean where the weight can be selected based on the video features. Here, we treat ViS_1 and ViS_2 equally since they estimate video quality using two different approaches. Moreover, the values of ViS_1 and ViS_2 are computed in different scales so a geometric combination would be more suitable than an arithmetic combination of the two indices.

Specifically, the ViS_3 value is computed as a geometric mean of ViS_1 and ViS_2, which is given by

$$\text{ViS}_3 = \sqrt{\text{ViS}_1 \times \text{ViS}_2}. \tag{3.33}$$

Here, ViS_3 is a single scalar that represents the overall perceived quality degradation of the distorted video in comparing to the reference video. The smaller the ViS_3 value, the better the video quality. A value $\text{ViS}_3 = 0$ indicates that the distorted video is equal in quality to the reference video. We will demonstrate the performance of the ViS_3 on various video-quality databases in Chapter 5.

3.3 Chapter summary

In this chapter, we proposed a full reference VQA algorithm that estimates video quality degradation by measuring spatial distortion and spatiotemporal dissimilarity separately in two stages. The spatial distortion value is computed based on two strategies from the MAD algorithm and a model of temporal weighting. The spatiotemporal dissimilarity value is computed from the analysis of the vertical and horizontal STS images. Overall estimate of perceived video quality degradation is given by a geometric mean of the spatial distortion and spatiotemporal dissimilarity values. To evaluate performance of ViS_3 algorithm, we need a practical database that contains various videos and their associative subjective ratings of video quality; such a trusted video-quality database is presented in Chapter 4.

CHAPTER 4

CSIQ - A VIDEO DATABASE FOR QUALITY ASSESSMENT

4.1 Introduction

Although the video quality assessment has drawn a lot of attention from the research communities these days, one important thing that makes the video quality assessment more difficult is the lack of a video-quality database. The video database is necessary and can be used as a reliable material to validate performance of a VQA method. A good VQA method should be able to obtain high performance in predicting quality of the videos in a selected video-quality database. The prediction performance is often measured by the correlation between the scores predicted by the VQA algorithm and the subjective ratings of quality, which are collected from a significant number of trusted human subjects who participated in the experiment.

Currently, there are few video-quality databases that are available to the research community. The most popular and widely used database for the Video Quality Assessment is the VQEG FR-TV Phase I developed by Video Quality Group Experts [31]. The videos in the VQEG Phase I database are interlaced, which potentially leads to visual artifacts when the video is displayed in increasingly common progressive scan monitors. The process of de-interlacing can create distortions associated with the particular algorithm used (juddering, combing, etc.). Moreover, the interlaced videos do not represent current trends in the video industry such as multimedia, IPTV, HDTV, and so on. Furthermore, according to the final report [31], ten proponents (include PSNR) are used to perform video quality assessment; the results and database are made publicly but they are not recommended as an ITU standard because those proponents show almost equivalent results.

The second video-quality database that is often used in the research of video quality assessment recently is the LIVE Video Database, developed by the University of Texas at Austin. It consists of ten (10) original sequences and 150 distorted videos in the progressive scanning format. The database contains four types of typical distortion including two compression distortion types (*MPEG-2*, *H.264*) and two types of distortions caused by the wireless and IP transmission environments (*Wireless* and *IPPL*) to the H.264 video streams. The diverse distortion makes this database useful in testing the consistency of VQA algorithms. However, the LIVE database does not contain the traditional distortion (white noise) and the newly released video compression standard (H.265/HEVC). The videos in the LIVE database have been also criticized for being noisy, darkened, and low contrast.

Another database that is made available for the research community is the IVPL HD video database [64], developed by the Image and Video Processing Lab at the Chinese University of Hong Kong. This database contains similar distortion types as the LIVE database but has lesser number of distorted videos. However, the IVPL database contains the distortion caused by wavelet compression (Dirac) which is missing from the LIVE database. The IVPL database can also be used to test on larger screen resolution because all of the videos is in Full HD format (1920×1088). Similar to the LIVE database, the IVPL database lacks some distortion types and the number of videos (128) is considered low comparing to the LIVE and VQEG databases. Some algorithms, such as MOVIE [10], can not be used to test on HD videos because it requires a relative large amount of memory to process these videos.

From the disadvantages and limitations of current video-quality databases, it is necessary to have a new video-quality database with more videos and more distortion types to validate the performance of objective VQA methods. In this chapter, we present a newly developed database for video quality assessment that contains more videos (12 original sequences, 216 distorted sequences), more types of distortion (six), and each distortion type has three different levels. Totally, there are 228 videos in the raw YUV420 progressive format in this database including the original sequences. To collect subjective ratings of video quality, an experiment has been performed following the SAMVIQ methodology [65] with the participation of 35 subjects.

4.2 Original video sequences

There are many sources of high quality video sequences that are available for the research community today such as the VQEG HD video database, the Technical University of Munich database [66], or the videos from Xiph Media Test website [67]. Another source of high quality videos released recently is the original sequences from the High Efficient Video Coding (HEVC) team [68], which is contributed by some well-known producers like NTT Docomo, Samsung, etc. These videos are captured by the professional, high-end equipment and stored in the YUV raw format.

4.2.1 Selected original sequences

From the available sources of high quality videos above, we carefully selected 12 original video sequences with various Spatial and Temporal Information according to ITU-R BT.50 recommendation. Nine sequences are downloaded from the HEVC project, one video (*Carving*) is selected from the Technical University of Munich (TUM) database, and two other videos are downloaded from the Xiph Media website [67]. The three latter videos are extracted from longer original sequences without changing their frame rates. All 12 video sequences are processed to have the same duration of ten seconds and the same spatial resolution of 832 × 480 whereas the frame rates span a large range from 24 to 60 frames per second. The native information of these selected videos is shown in Table 4.1 in terms of spatial resolution, frame rates, and reference sources.

Table 4.1: Native information of selected original video sequences

Videos	**Spatial resolution**	**Fps**	**Source**
BQMall	832 × 480	60	HEVC project [69]
Flowervase	832 × 480	30	HEVC project [69]
Keiba	832 × 480	30	HEVC project [69]
PartyScene	832 × 480	50	HEVC project [69]
BQTerrace	1920 × 1080	60	HEVC project [69]
BasketballDrive	1920 × 1080	50	HEVC project [69]
Cactus	1920 × 1080	50	HEVC project [69]
Kimono	1920 × 1080	24	HEVC project [69]
ParkScene	1920 × 1080	24	HEVC project [69]
Carving	1920 × 1080	25	Tech. Univ. of Munich [66]
Chipmunks	1920 × 1080	24	Xiph media [67]
Timelapse	3072 × 2304	30	Xiph media [67]

Figure 4.1 shows the representative frames of these selected sequences. A brief description of the video contents is provided as following

- *BQMall* - camera is panning from right to left showing people with various actions in the shopping mall.

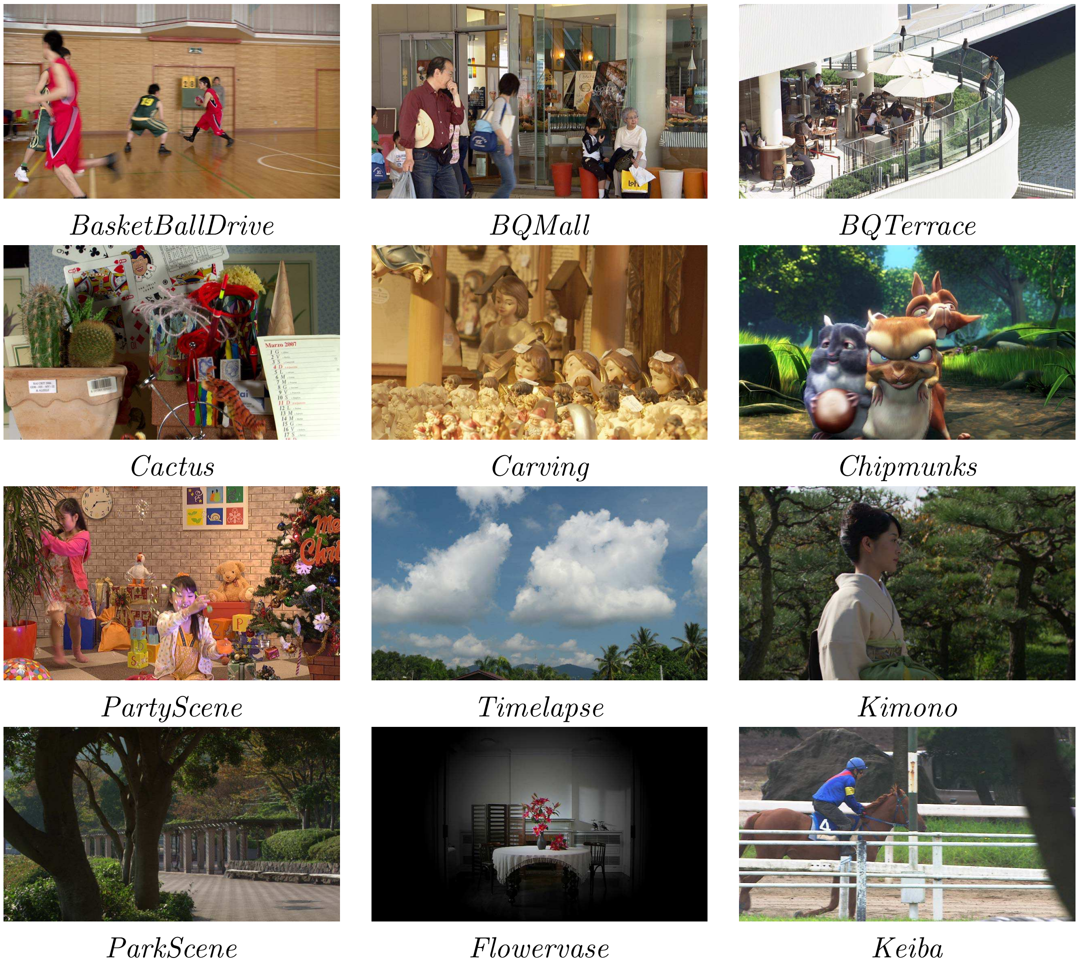
- *BQTerrace* - Camera pans and tilts, showing people in a restaurant, the cars in the highway, and the still water in the river.

- *BasketBallDrive* - A group of youngsters works in team to score a goal in a practice basketball game. Camera is chasing the ball in action.

- *Cactus* - Static camera, everything stays still except the spinning cactus bowl, the board of joker cards, and the rolling tiger toy.

- *Carving* - Two mature men visit a carving shop, camera zooming out from a doll's face in the shop.

- *Chipmunks* - Cartoon movie shows three naughty chipmunks and a log of wood coming from behind. Camera is static and tilting.



BasketBallDrive *BQMall* *BQTerrace*

Cactus *Carving* *Chipmunks*

PartyScene *Timelapse* *Kimono*

ParkScene *Flowervase* *Keiba*

Figure 4.1: Representative frames of original video sequences in the CSIQ database

- *Flowervase* - Camera zooms in a flower vase on the table in the darkroom. Lighting condition is getting brighter.

- *Keiba* - Fast moving of the horses and riders in a race track. Camera is panning from left to right with a few big trees stay in the line of sight.

- *Kimono* - A lady walks in the garden to a wooden house, camera follows the lady and captures the slow motion of human subject.

- *ParkScene* - Three riders are biking in the park from two opposite directions. Camera is panning slowly.

- *PartyScene* - Three kids are playing at a Christmas party. Camera zooms in a girl who is blowing the water bubbles in the center of the video. The other kids are running around the Christmas tree.

- *Timelapse* - Static camera captures lapse movement of clouds in the sky and the trees underneath during daytime.

4.2.2 Spatial and temporal information

According to the ITU recommendation P.901 for tested video sequences, the selected original video sequences should represent various level of spatial and temporal information complexity. These spatial and temporal information measures are computed for all frames of a complete test sequence. A maximum function is then used to remove the variability of these measures over time and yield the indices that represent the spatial and temporal information complexity of the videos.

A Spatial perceptual information measurement

The spatial perceptual information, denoted by SI, is computed using the Sobel filter. The luminance component of video frame at index n (F_n) is first filtered via the Sobel filter [$Sobel(F_n)$]. The standard deviation of the pixel values in each Sobel-filtered frame is then computed. The maximum value of these values across all frames is chosen to represent the spatial information content of the video sequence. This process can be represented in the equation form as:

$$SI = \mathrm{MAX}_{time}\{std_{space}[Sobel(F_n)]\} \tag{4.1}$$

where std_{space} is the operator to compute the standard deviation for each filtered frame, MAX_{time} is the max operator performed across time dimension of the video.

B Temporal perceptual information measurement

The temporal perceptual information, denoted by TI, is computed from the motion difference feature, $M_n(i,j)$, which is simply defined as the difference between the pixel values (of the luminance component) at the same spatial location but at successive frames. $M_n(i,j)$ is given by:

$$M_n(i,j) = F_n(i,j) - F_{n-1}(i,j) \tag{4.2}$$

where $F_n(i,j)$ is the pixel at the i^{th} row and j^{th} column of n^{th} frame in time.

The measure of temporal information complexity, TI, is computed as the maximum over time (MAX$_{time}$) of the standard deviation over space (std_{space}) of each $M_n(i,j)$ over all i and j. If two adjacent frames have more motion between them or more different, the TI will have high value.

$$TI = \mathrm{MAX}_{time}\{std_{space}[M_n(i,j)]\} \tag{4.3}$$

By using those definitions of spatial and temporal information complexity, we compute the SI and TI indices for all selected original video sequences, these indices are shown in Figure 4.2 in terms of a scatter plot between SI and TI indices. As observed from Figure 4.2, the selected video sequences span a wide range of spatial and temporal information complexity from low SI, low TI values (video *Timelapse*) to high SI, high TI values (video *Keiba*). Therefore, these selected video sequences represent different levels of spatial and temporal complexity, and can be used as candidates for visual psycho-physical experiment of video-quality.

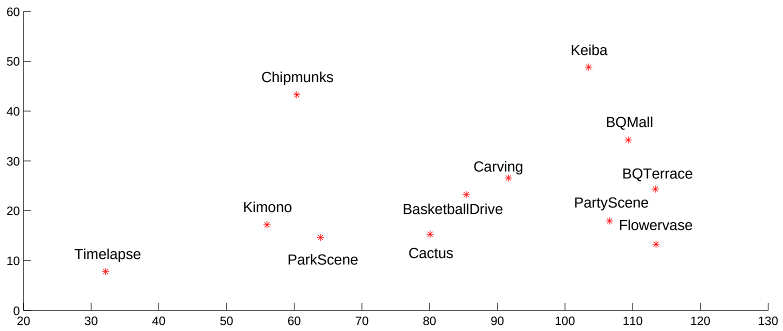


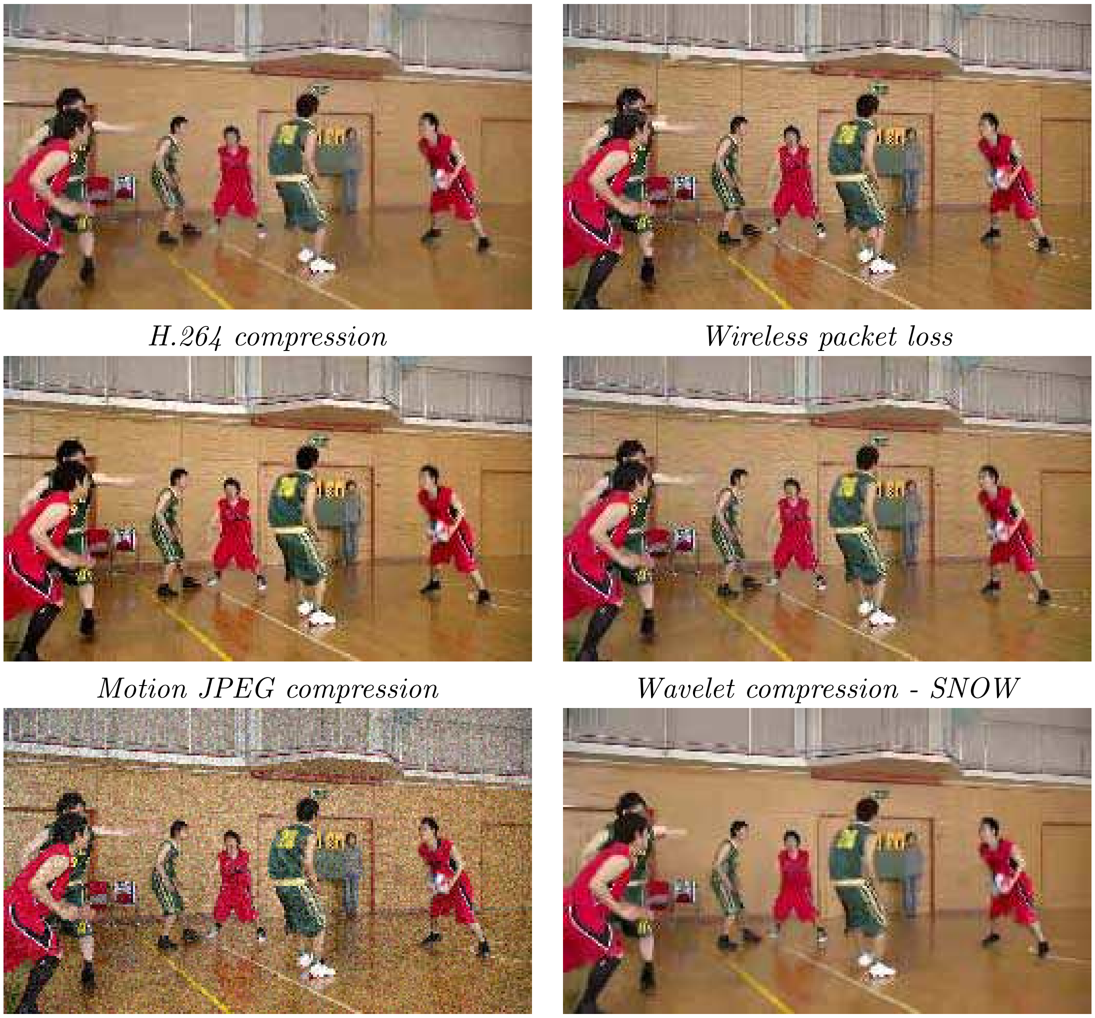
Figure 4.2: Scatter plot of spatial information (SI) and temporal information (TI) computed from selected video sequences

4.3 Test sequences and distortion types

For each selected video sequences, we created 18 test video sequences using six types of distortions, three levels of distortion is used for each distortion type. The distortion types consist of four compression-based distortion types [Motion JPEG (*MJPEG*), *H.264*, *HEVC*, and wavelet compression using *SNOW* codec [70]], and two transmission-based distortion types [packet-loss in a simulated wireless network (*WLPL*) and additive white Gaussian noise (*AWGN*)]. The distortion levels were adjusted manually and carefully selected so that the test video sequences spanned a similar range of visual quality for different original sequence and different distortion type. Figure 4.3 shows the representative frames that are generated from video *BasketBallDrive* with six different distortion types; the details for each distortion type are described as below.

4.3.1 H.264 compression with constant bitrate

The H.264 compression standard has a very broad application range that covers many forms of digital compressed video, from the low bit-rate Internet streaming applications to HDTV broadcast and Digital Cinema applications with nearly lossless coding. Therefore, it is important to include the H.264 compressed videos in the database. To generate these H.264 test sequences, we used the JM Reference software [71] version



H.264 compression *Wireless packet loss*

Motion JPEG compression *Wavelet compression - SNOW*

White noise *HEVC compression*

Figure 4.3: Representative frames of six different distortion types generated from the same video *BasketBallDrive*

15.0 to encode the original video sequences with three different constant bitrate levels in order to obtain three different quality levels. The target bitrates for compression are selected depending on the frame rate of the original sequences. To ensure that the test sequences span the range from high to low quality, for each original sequence, we generate various test sequences with different bitrates and then manually select three videos for official testing. For videos with low frame rate (24, 25, and 30), we select three birate levels of 512 Kbps, 1 Mbps, and 2 Mbps. For the high frame rate videos (50 and 60 fps), the bitrate levels are selected at 1 Mbps, 2 Mbps, and 5 Mbps. Some principal parameters for used in JM software are set as follows:

- ProfileIDC = 100 (High Profile)
- GOP Structure = IBBPBB
- Chroma format = 4:2:0
- IntraPeriod = 14
- SymbolMode = 1 (CABAC)
- SearchMode = 3 (EPZS)

4.3.2 H.264 streams with wireless packet loss

Video transmission for mobile devices in a wireless environment is a major application in high speed data systems. The superior compression efficiency and error resilience of H.264 makes it ideal for use in harsh wireless transmission environments. A packet of data transmitted over a wireless channel is susceptible to bit errors due to attenuation, fading, and multi-users interference in wireless channels. The video streaming over the wireless networks encounter with the popular packet loss and cause the distortion in the receiver's side. A typical pattern of the packet loss is the video with some totally lost areas while the other areas maintains high quality.

The H.264 compressed bitstreams were created using the JM reference software to obtain high quality (the QP parameter are set to 18). For each of the 12 high quality original H.264/AVC bitstreams, a number of corrupted bitstreams were generated by dropping packets according to a given error pattern. Except the coded slices belonging to the first frames, the other slices might be discarded from the coded bitstream to simulate loss packets. From five generated packet loss rates, three different packet loss rates were chosen (0.5%, 1.5%, and 4.5%) to span the desired range of quality in the simulated wireless environment.

4.3.3 Motion JPEG

Motion JPEG (M-JPEG or MJPEG) is a video compression standard in which each video frame or interlaced field of a digital video sequence is compressed separately as a standalone JPEG-compressed image. MJPEG is originally developed for multimedia PC applications, but MJPEG is now used by may video-capture devices and non-linear video editing systems. The MJPEG compression is simple to implement, requires minimal hardware, and can be useful for videos with rapidly changing motion where motion estimation is unable to predict correctly.

Because the JPEG compression is used for each video frame, the MJPEG compressed videos exhibit the blurring and blocky artifacts. In this CSIQ database, we generate various MJPEG videos from each original sequence using the FFMPEG software [70] with different values of quantization factor Q, which is in the range from 1 to 32. After carefully inspecting the effect of different Q values on the quality of the compressed videos, we selected three different values (8, 16, and 30) for the Q factor and applied to all the original sequences.

4.3.4 Wavelet compression - Snow codec

Comparing to more modern formats (such as JPEG2000 and H.264/MPEG-4 AVC), JPEG compression is inefficient, using more bits to deliver similar quality. The JPEG2000 compression standard has become popular in the field of image compression due to its high coding performance such as strong error resiliency, low latency, high compression performance, and good perceptual quality. Its extension to video compression, the Motion JPEG2000 (MJ2K), has been used as an efficient compression standard in various applications, which require fast, frequent, and convenient frame access; high-quality high-resolution imaging (medical and satellite); or video applications requiring real-time simple encoding.

Although the Motion JPEG200 compression for video is not available and is still under investigated in the FFMPEG software, the FFMPEG tool supports another type of video compression using wavelet transformation called *SNOW*. The *SNOW* codec offers both lossy and lossless coding, which features wavelet transform, overlapping block-based motion-compensation, and entropy coding that is not based on Huffman coding. The default wavelet used by the *SNOW* codec in FFMPEG is a symmetric biorthogonal compact 9/7 wavelet similar to the famous biorthoginal Daubechies 9/7 wavelet. The quality of the MJ2K video can be controlled by the Q factor in the FFMPEG software. We thoroughly checked and chosen three values of 4, 8, and 12 for the Q factor to generate *SNOW* test video sequences in our database.

4.3.5 White noise

White noise, a popular concept in analog video and television, is a random dot pattern of static displayed when no transmission signal is obtained by the antenna receiver of television sets and other display devices. The random pattern superimposed on the picture, visible as a random flicker of “dots” or “snow”, is the result of electronic noise and radiated electromagnetic noise accidentally picked up by the antenna. This effect is most commonly seen with analog TV sets or blank VHS tapes.

Most of the video technologies now are very effective in white noise removal but we still included this type of distortion here to make the database more diverse and reliable. The Gaussian white noise is superimposed to the original sequences by changing the values of signal to noise ratio (SNR) according to a sine function for two cycles in the duration of 10 seconds. Three different patterns of the sine function are chosen to represent the fluctuation of SNR values. The two peak amplitude values of these functions are (3, 9), (6, 18), and (12, 20) respectively with random initial phase. Figure 4.4 illustrates the fluctuation of the SNR for the duration of 10 seconds at the second level of peak amplitude values ([6, 18]).

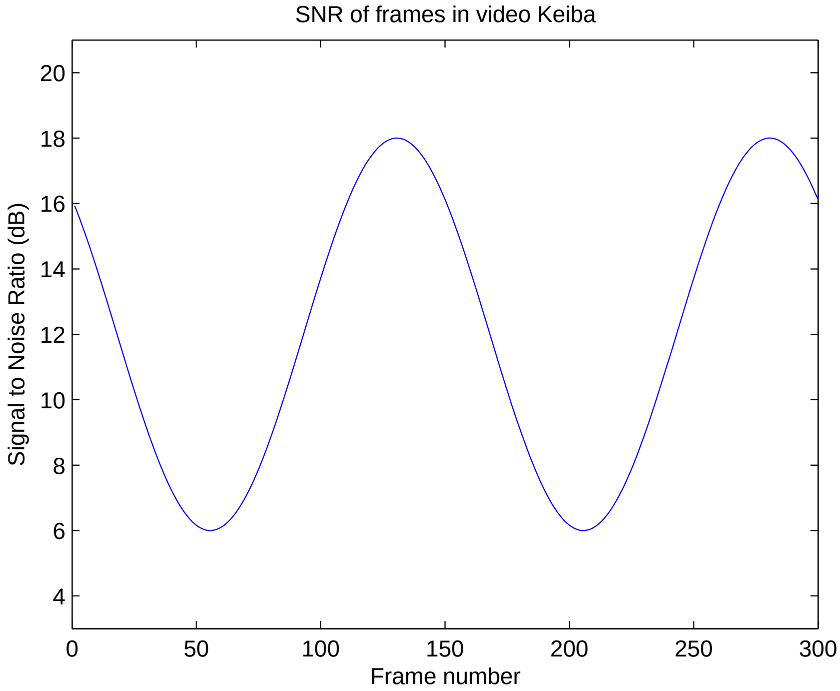


Figure 4.4: SNR varied by frames in video *Keiba*

4.3.6 HEVC compression

The newly High Efficiency Video Coding (HEVC/H.265) is a video compression standard, a successor to H.264/MPEG-4 AVC (Advanced Video Coding), that was jointly developed by the ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) [72], [68]. HEVC was designed to improve coding efficiency compared to H.264/ MPEG-4 AVC, i.e. to reduce bitrate requirements by half with comparable image quality, and at the expense of increased computational complexity. This improvement is needed for various applications that use high definition videos. Two key features that make HEVC more efficient compared to H.264/MPEG-4 AVC were the support for higher resolution video and the implementation of improved parallel processing methods. More detailed description about the HEVC coding standard can be seen at Ref. [73]. The first official version of the HEVC standard was completed and released in early 2013.

In the CSIQ database, we used the HM software [68] to generate HEVC coded sequences. The compression factor of the HEVC videos can be controlled by the quantization parameter QP, the greater the QP value, the more compressed the HEVC video. For each selected original video sequence, we manually choose three different QP values (32, 38, and 44) from various QP values to generates HEVC-compressed versions of test video sequences.

4.4 Experiment setup

The experiment to collect subjective ratings of quality for the CSIQ video database is designed following the subjective assessment methodology for video and image quality [65] (SAMVIQ), which has specifically been designed for multimedia content.

4.4.1 SAMVIQ methodology

In the procedure of SAMVIQ methodology, video sequences are shown in multi-stimulus form, so that users can choose the order of tests and correct their votes appropriately. The individual assessor can start and stop the evaluation process as he/she wishes and is allowed to determine his/her own pace for performing the grading, modifying grades, playing back when needed, etc.

The test sequences are displayed randomly and quality evaluation is carried out scene after scene. An explicit and a hidden reference are included in the test sequences. The explicit reference is an uncompressed version of the original sequence and allows the assessor to determine a near-absolute measure of video quality. A hidden reference is technically identical to the explicit reference but is not readily available to the subjects. It is actually hidden among other stimuli and the subject should be able to identify it based on their quality ratings.

Comparing to the DSCQS method, there is no continuous sequential presentation of the video sequences in SAMVIQ method, this can prevent the assessor from making errors of judgment due to a lack of concentration. The post-processing in SAMVIQ

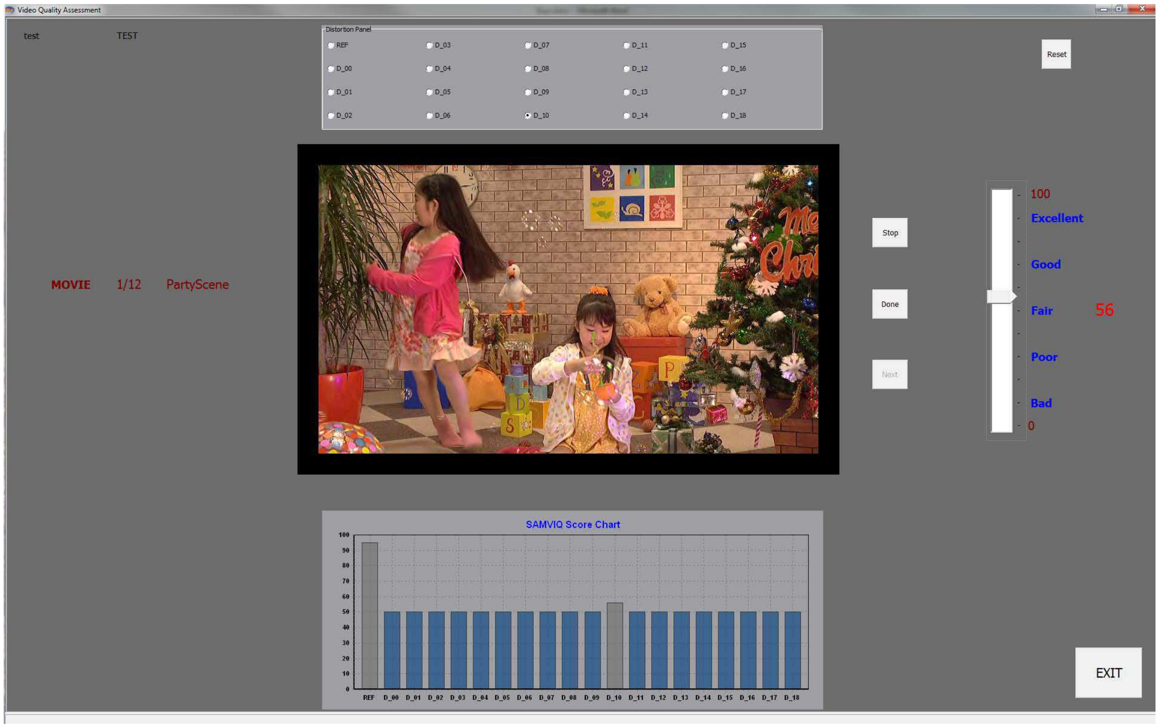


Figure 4.5: A sample screen-shot of the experiment graphical user interface

also includes some improved rejection criteria (compared with those used in BT.500). The multimedia images/videos are assessed on a multimedia screen and platforms, not on conventional TV displays, in order to avoid the artifacts due to interlacing and flickering. As a result, SAMVIQ offers higher reliability.

4.4.2 Graphical user interface design

We built the graphical user interface to perform the experiment according to the SAMVIQ methodology via Borland C++ Builder program. A screen-shot of the experiment design is shown in Figure 4.5. To prevent visual latency, the under tested video sequence is preloaded and carefully synchronized with the computer's processing performance to make sure that the video display duration is exactly 10 seconds.

The videos were displayed at their native resolution and the remaining areas of the display were set to solid gray to eliminate the effect of surrounding regions. A continuous scale for video quality was displayed on the screen to the users, with a cursor is preset at the center of the quality scale to avoid biasing subject's opinion of quality ratings. Five labels are marked on the quality scale to help subjects, the marked labels are "*Excellent*", "*Good*", "*Fair*", "*Poor*", and "*Bad*". Subjects are only able to move the cursor to change quality score after viewing the whole video sequence at least one time; they were allowed to take as much time as needed to review the video and enter the score. Subjects are allowed to go back to change their score if they feel the previous entered scores need to be modified. Once the score was assigned for every video in the subset of the same content, subject might proceed to the next video by clicking the "Next" button, and after that, he/she cannot change the score of the previous contents.

For each content, subjects are forced to watch the explicit reference video first and then watch the other 19 videos (the hidden reference video and 18 distorted ones) at a random order. Subjects need to finish all the videos of the same content before they can proceed to next content. While being in the time of one content, subjects are free to go back, replay previous videos, and change their opinion scores if needed. During the experiment, subjects can take rest at any time but a break of at least 5 minutes is mandatory after a session of 30 minutes.

The experiment is performed in a darkened room in front of the HP Lacie monitor, which is calibrated carefully. The distance from subject to the monitor are kept at least six times the height of the video (about 70 cm). Twenty-five subjects from the Digital Signal Processing class at Oklahoma State University and ten naive subjects from other fields of study are voluntary to attend and conduct the experiment. The subjects are divided into two groups so each video content is rated by either seventeen or eighteen subjects.

4.5 Data analysis

The raw subjective scores assigned by each subject to their test video sequences are automatically stored in a text file. These files are collected for post-processing after all subjects had finished the experiment. Let $\{s_{i,j,k}\} \in [0, 100]$ denote the quality assigned by subject i^{th} to video j^{th} of the k^{th} content, where $s_{i,0,k}$ denotes the score of the explicit references and $s_{i,19,k}$ denotes the score of the hidden references.

Normally, the explicit references are expected to rate at the highest quality comparing to other sequences of the same content; if any sequence is rated higher than the explicit reference, we set the score of that video equal to the score of the explicit reference. If subjects rate the hidden reference lower than some of the distorted videos, we set the score of the hidden reference equal to the score of the explicit reference. Then, the scores assigned to the explicit references are excluded from the raw scores.

$$s_{i,j,k} = s_{i,0,k} \text{ if } s_{i,j,k} > s_{i,0,k} \tag{4.4}$$

$$s_{i,19,k} = s_{i,0,k} \text{ if } s_{i,j,k} > s_{i,19,k} \tag{4.5}$$

Next, the scores rated by each subject are converted to Z-scores using

$$z_{i,j,k} = \frac{s_{i,j,k} - M_i}{S_i} \tag{4.6}$$

where M_i and S_i are the mean and standard deviation of all scores rated by i^{th} subject which are given as follows

$$M_i = \frac{1}{J \times K} \sum_{j=1}^{J} \sum_{k=1}^{K} s_{i,j,k} \tag{4.7}$$

$$S_i = \sqrt{\frac{1}{J \times K - 1} \sum_{j=1}^{J} \sum_{k=1}^{K} (s_{i,j,k} - M_i)^2} \tag{4.8}$$

where $J = 19$ is the number of test video sequences for each content and $K = 12$ is the number of original video sequences.

To ensure the reliability of the subjective rating scores, the EBU rejection method is applied to reject subjects who do not has the same opinion with the others. First, we computed the overall Z-score $\bar{z}$ as the average of the Z-scores rated by all subjects for each test video sequence.

$$\bar{z}_{j,k} = \frac{1}{J \times K} \sum_{j=1}^{J} \sum_{k=1}^{K} z_{i,j,k}. \tag{4.9}$$

The Pearson linear correlation coefficient is used as the criterion of the rejection process. If subject i has a correlation coefficient with the overall average Z-scores (Pearson$(z_i, \bar{z})$) lower than a threshold ($\alpha = 0.85$), that subject is rejected and his scores are excluded from the average scores. We repeat this step until all subjects have the correlation coefficient of at least 0.85 with the overall average Z-scores. After our rejection process, subjective scores rated by four subjects have been eliminated and each video now has been rated by either 15 or 16 subjects.

The score of each video is then subtracted from the score of the respective hidden reference video of the same content rated by the same subject to yield raw different mean opinion score (DMOS).

$$dmos_{j,k} = \bar{z}_{19,k} - \bar{z}_{j,k} \tag{4.10}$$

The scores of the hidden references are now excluded from the DMOS scores.

Again, the raw DMOS scores are normalized by converting to Z-scores using

$$\mu = mean(dmos_{j,k}) \tag{4.11}$$

$$\sigma = std(dmos_{j,k}) \tag{4.12}$$

$$\hat{z}_{j,k} = \frac{dmos_{j,k} - \mu}{\sigma} \tag{4.13}$$

The normalized DMOS scores $\hat{z}$, in the range of [-3, 3], are rescaled to the range [0, 100] to yield final DMOS scores via

$$DMOS_{j,k} = \frac{100 \times (3 - \bar{Z}_{j,k})}{6} \tag{4.14}$$

The final DMOS score of each test video sequence represents that video's subjective ratings of quality, the smaller the DMOS score, the better the video quality according to human subjects. The DMOS scores are then used to validate and compare performances of various VQA algorithms. The algorithm that predicts subjective ratings of video quality better in some strictly approved criteria is said to have better performance than the other.

4.6 Chapter summary

In this chapter, we described our work on creating a video database for video quality assessment from the beginning of choosing the test sequences to the post-processing of subjective scores. The CSIQ video database provides more videos, more distortion types, and contains more recent compression standard comparing to other publicly available video-quality databases. In next chapter, we will validate performance of our VQA algorithm proposed in Chapter 3 and compare with some other VQA algorithms in predicting quality of videos in the CSIQ database as well as in other databases.

CHAPTER 5

VQA PERFORMANCE EVALUATION

In this chapter, we analyze performances of the proposed ViS$_3$ algorithm and some other VQA algorithms in predicting subjective ratings of video quality. These algorithms are tested and validated on three publicly available video-quality databases.

5.1 Video quality databases

To evaluate performance of the proposed ViS$_3$ algorithm and other VQA algorithms, we used the following three publicly available video-quality databases that have multiple types of distortion:

1. The LIVE video database (four types of distortion) [1];

2. The IVPL video database (four types of distortion) [64];

3. The CSIQ video database (six types of distortion) [74].

5.1.1 LIVE video database

The LIVE video database [1], developed at the University of Texas at Austin, contains 10 reference videos and 150 distorted videos (15 distorted versions per each reference video). All videos are in raw YUV420 format with a resolution of 768 × 432 pixels, approximately 10 seconds in duration, and at frame rates of 25 or 50 fps.

There are four distortion types in this database: MPEG-2 compression (*MPEG-2*), H.264 compression (*H.264*), simulated transmission of H.264-compressed bitstreams through error-prone IP networks (*IPPL*), and simulated transmission of

H.264-compressed bit-streams through error-prone wireless networks (*WLPL*). Three or four levels of distortion are present for each distortion type.

5.1.2 IVPL video database

The IVPL HD video database [64], developed at the Chinese University of Hong Kong, consists of 10 reference videos and 128 distorted videos. All videos in this database are in raw YUV420 format with a resolution of 1920 × 1088 pixels, approximately 10 seconds in duration, and at a frame rate of 25 fps.

There are four types of distortion in this database: Dirac wavelet compression (*DIRAC*, three levels), H.264 compression (*H.264*, four levels), simulated transmission of H.264-compressed bit-streams through error-prone IP networks (*IPPL*, four levels), and MPEG-2 compression (*MPEG-2*, three levels). To reduce the computation time, we downsampled the videos to the size of 960 × 544 using open source FFMPEG software [70] with its default configuration.

5.1.3 CSIQ video database

The CSIQ video database [74], described in Chapter 4 of this dissertation, consists of 12 reference videos and 216 distorted videos. All videos in this database are in raw YUV420 format with a resolution of 832 × 480 pixels, a duration of 10 seconds, and span a range of various frame rates: 24, 25, 30, 50, and 60 fps.

Each reference video has 18 distorted versions with six types of distortion, three different levels for each type. The distortion types consists of four video compression distortion types [Motion JPEG (*MJPEG*) , *H.264, HEVC/H.265*, and wavelet-based compression using *SNOW* codec [70]], and two transmission-based distortion types [packet-loss in a simulated wireless network (*WLPL*) and additive white Gaussian noise (*AWGN*)]. The experiment was conducted following the SAMVIQ methodology [65] with the participation of 35 voluntary subjects.

5.2 VQA algorithms and performance measurements

We compared ViS$_3$ with traditional PSNR [28] and some recent full-reference VQA algorithms for which code is publicly available: VQM [2], MOVIE [10], and TQV [13] on the three video-quality databases as stated above. PSNR was applied on a frame-by-frame basis, VQM and MOVIE were applied using their default implementations and settings, and TQV was applied using its original training parameters. For the ViS$_3$ algorithm, we employed a GOF size of $N = 8$.

All these algorithms are applied to the videos of the three aforementioned video-quality databases to obtain the raw predicted quality scores. Before evaluating performance of each algorithm on each video database, we applied a four-parameter logistic transform to the raw predicted scores, as recommended by VQEG in Ref. 31. The four-parameter logistic transform is given by:

$$f(x) = \frac{\tau_1 - \tau_2}{1 + \exp(-\frac{x - \tau_3}{|\tau_4|})} + \tau_2, \tag{5.1}$$

where x denotes the raw predicted score and $f(x)$ denotes the logistic fitted score; τ_1, τ_2, τ_3, and τ_4 are four free parameters that are selected to provide the best fit of the predicted scores to the subjective rating scores.

Following VQEG recommendations in Ref. 31 about performance measurements, we employed the Spearman Rank-Order Correlation Coefficient (SROCC) to measure prediction monotonicity, and employed the Pearson Linear Correlation Coefficient (CC) and the Root Mean Square Error (RMSE) to measure prediction accuracy. The prediction consistency of each algorithm was measured via two additional criteria: the outlier ratio (OR [3]) and the outlier distance (OD [29]). The outlier ratio (OR) is the ratio of number of *false* scores predicted by the algorithm to the total number of predicted scores. A *false* score is defined as the transformed score lying outside the 95% confidence interval of the associated subjective score [3]. Whereas the outlier

Table 5.1: Performances of ViS$_3$ and other VQA algorithms in predicting video quality in three video databases. The best-performing result is bolded and the second best-performing result is italicized and bolded. Note that ViS$_3$ is the best-performing algorithm on all three databases.

		PSNR	VQM	MOVIE	TQV	ViS$_3$	ViS$_1$	ViS$_2$
SROCC	LIVE	0.523	0.756	0.789	***0.802***	**0.816**	0.762	0.736
	IVPL	0.728	0.845	***0.880***	0.701	**0.896**	0.872	0.817
	CSIQ	0.579	0.789	0.806	***0.814***	**0.841**	0.757	0.831
CC	LIVE	0.549	0.770	0.811	***0.815***	**0.829**	0.785	0.746
	IVPL	0.723	0.847	***0.879***	0.722	**0.896**	0.863	0.823
	CSIQ	0.565	0.769	0.788	***0.795***	**0.830**	0.739	0.830
RMSE	LIVE	9.175	7.010	6.425	***6.357***	**6.146**	6.807	7.313
	IVPL	0.730	0.561	***0.504***	0.731	**0.470**	0.534	0.601
	CSIQ	13.724	10.633	10.231	***10.090***	**9.273**	11.197	9.279
OR	LIVE	2.00%	1.33%	**0%**	**0%**	**0%**	0%	2.00%
	IVPL	7.81%	**0.78%**	1.56%	7.81%	**0.78%**	1.56%	4.69%
	CSIQ	12.96%	5.09%	***4.17%***	4.63%	**3.70%**	7.41%	3.24%
OD	LIVE	11.479	5.385	**0**	**0**	**0**	0	9.076
	IVPL	3.422	***0.411***	**0.222**	2.556	0.616	1.085	1.005
	CSIQ	169.183	56.334	44.635	***40.946***	**28.190**	59.619	30.546

distance (OD) indicates how far the outliers fall outside of the confidence interval. The OD is measured by the total distance from all outliers to their closest edge points of the corresponding 95% confidence interval [29].

5.3 Overall performance

The performance of each algorithm on each video-quality database is shown in Table 5.1 in terms of five evaluation criteria (SROCC, CC, RMSE, OR, and OD). The best-performing result is bolded, and the second best-performing result is italicized and bolded. These results indicate that ViS$_3$ is the best-performing algorithm on all three video databases in terms of all five evaluation criteria. The performances yielded by ViS$_1$ and ViS$_2$ are also noteworthy.

In terms of prediction monotonicity (SROCC), ViS$_3$ is the best-performing algorithm on all three databases. On the LIVE and CSIQ databases, ViS$_3$ and TQV are

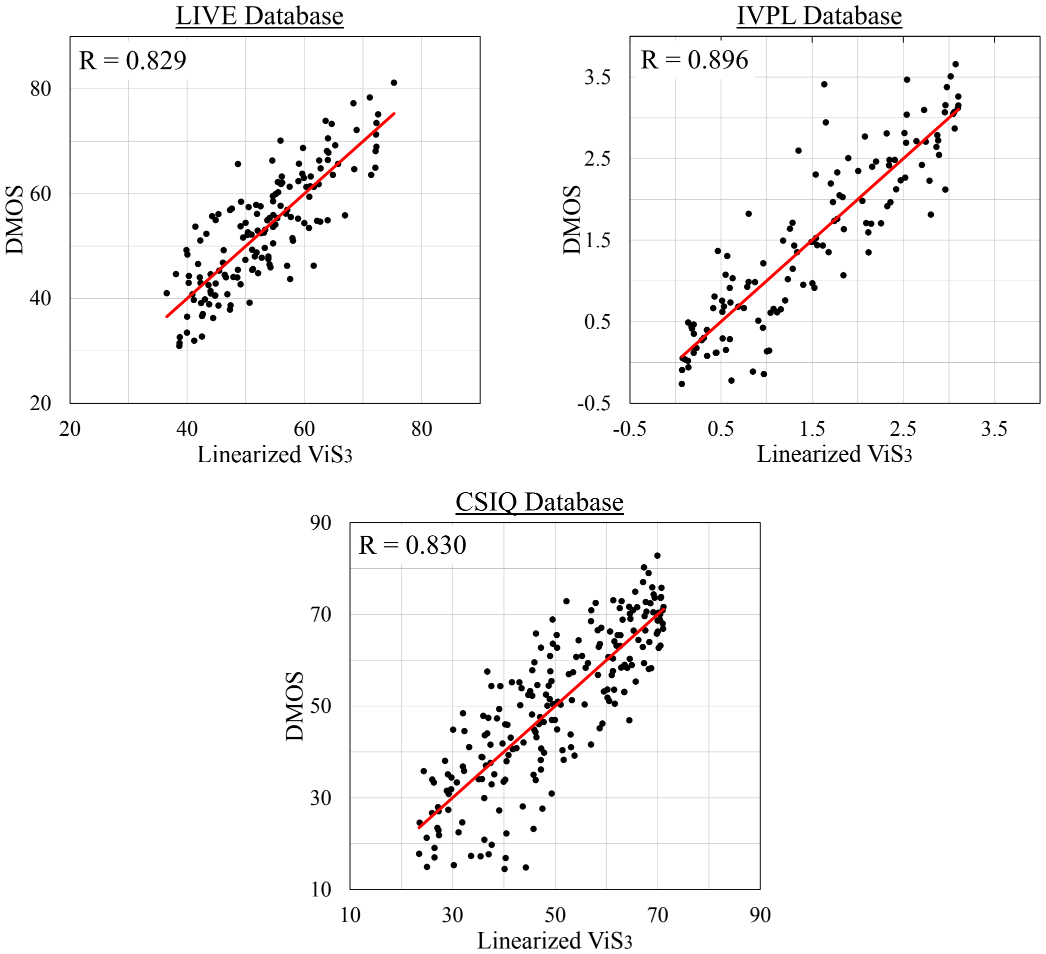


Figure 5.1: Scatter-plots of the logistic-transformed scores predicted by ViS$_3$ versus subjective scores on the three video-quality databases. Notice that all the plots are homoscedastic. The R values denote correlation coefficient (CC) between the logistic-transformed scores and subjective quality rating scores (DMOS).

the two best-performing algorithms. On the IVPL database, ViS$_3$ and MOVIE are the two best-performing algorithms. A similar trend in performance is observed in terms of prediction accuracy (CC and RMSE).

In terms of prediction consistency measured by the outlier ratio (OR), on the LIVE database, three algorithms (MOVIE, TQV, and ViS$_3$) have an OR of zero, which indicates that they do not yield any outliers. On the IVPL database, both ViS$_3$ and VQM have only one outlier. On the CSIQ database, ViS$_3$ and MOVIE are the two algorithms with the least number of outliers.

In terms of the outlier distance (OD), on the LIVE database, three algorithms

(MOVIE, TQV, and ViS$_3$) have an OD of zero because they do not have any outliers. On the IVPL database, MOVIE and VQM have the smallest OD. Although ViS$_3$ yields only one outlier on the IVPL database as well as VQM, ViS$_3$ has larger OD because this outlier lies further away from its confidence interval. This indicates that ViS$_3$ has a weakness on the *IPPL* distortion, to which the outlier belongs. Furthermore, on the CSIQ database, ViS$_3$ and TQV yield the smallest OD values.

It can be observed from Table 5.1 that ViS$_1$ and ViS$_2$ yield different relative performances depending on the database. ViS$_1$ shows better predictions than ViS$_2$ on the LIVE and IVPL databases. However, ViS$_2$ shows better predictions than ViS$_1$ on the CSIQ database. Generally, ViS$_3$ shows higher SROCC and CC and lower RMSE, OR, and OD than either ViS$_1$ or ViS$_2$ alone. Nonetheless, it may be possible to combine ViS$_1$ and ViS$_2$ in a adaptive fashion for even better prediction performance, and such an adaptive combination remains an area for future research.

The scatter-plots of logistic-transformed ViS$_3$ values vs. subjective scores (DMOS) on the three databases are shown in Figure 5.1. The plots show a highly correlated trend between the logistic-transformed ViS$_3$ values vs. DMOS values. For all the three video-quality databases, the predictions are homoscedastic; i.e., there are generally no sub-populations of videos/distortion types for which ViS$_3$ yields lesser or greater residual variance in the predictions. These residuals are used for an analysis of statistical significance in Section 5.3.4.

5.3.1 Performance on individual types of distortion

We measured performance of ViS$_3$ and other algorithms on individual types of distortion for videos from the three databases. For this analysis, we applied the logistic transform function to all predicted scores of each database, then divided the transformed scores into separate subsets according to the distortion types, and then measured the performance criteria in terms of SROCC and CC for each subset. Table 5.2

shows the SROCC and CC values resulted from this computation.

In general, VQM, MOVIE, and ViS$_3$ all perform well on the *WLPL* distortion; these three algorithms show competitive and consistent performance on the *WLPL* distortion for both the LIVE and CSIQ databases. For the *H.264* compression distortion, ViS$_3$ and MOVIE perform well and consistently across all subsets of *H.264* videos on all three databases. ViS$_3$ and MOVIE are also competitive on both the *MPEG-2* and the *IPPL* distortion types on the LIVE and IVPL databases.

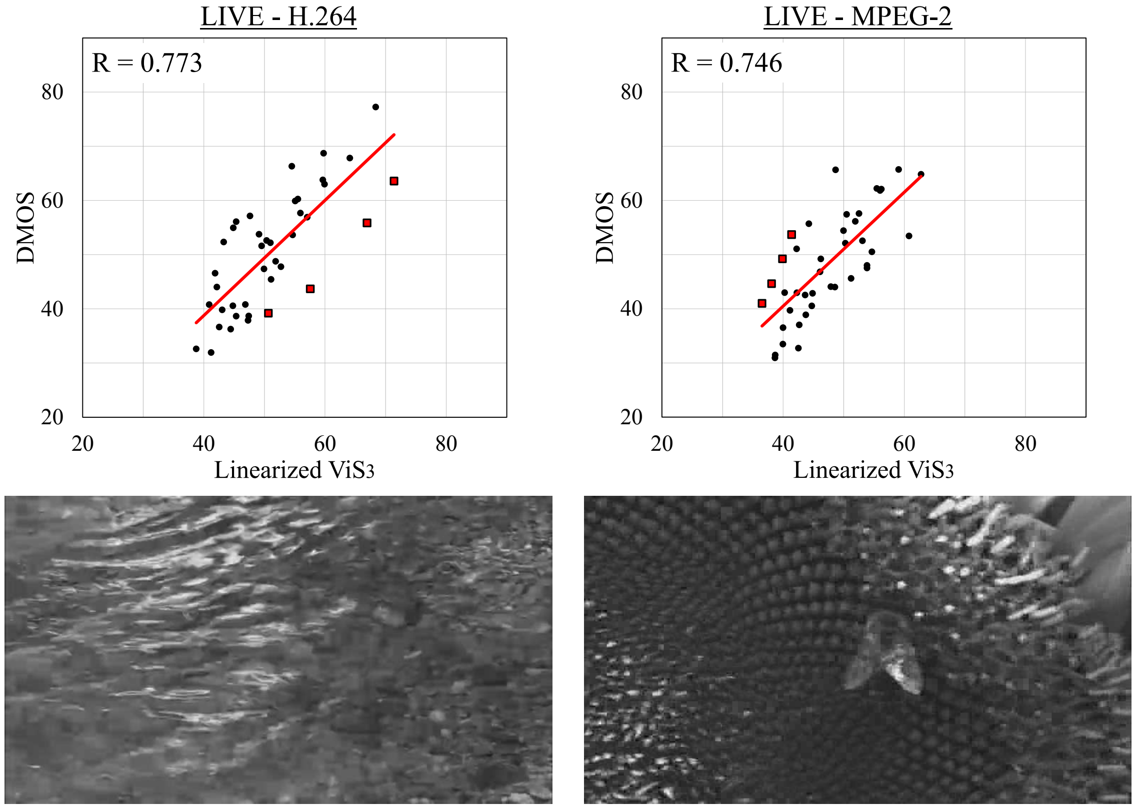
In particular, on the LIVE database, ViS$_3$ has the best performance on the *WLPL* distortion, VQM and ViS$_3$ have the best performance on the *IPPL* distortion, ViS$_3$, MOVIE and TQV are the three best-performing algorithms on the *H.264* distortion, TQV and MOVIE are the two best-performing algorithms on the *MPEG-2* distortion.

The low performance of the ViS$_3$ algorithm on *H.264* and *MPEG-2* compression types in the LIVE video database is due to the outliers corresponding to specific videos as shown in Figure 5.2; the outliers are marked by the red square markers. For *H.264*, the outliers correspond to the video *riverbed* where the water's movement significantly masks the blurring imposed by the compression. However, ViS$_3$ underestimates this masking, and thus overestimates the DMOS. For *MPEG-2*, the sunflower seeds in the video *sunflower* generally impose significant masking of the MPEG-2 blocking artifacts. However, there are select frames in this video in which the blocking artifacts become highly visible (owing perhaps to failed motion compensation), yet ViS$_3$ does not accurately capture the visibility of these artifacts, and thus underestimates the DMOS. These types of interactions between the videos and distortions are issues which certainly warrant future research.

On the IVPL database, ViS$_3$ yields the best performance on three types of distortion (*DIRAC*, *H.264*, and *MPEG-2*); ViS$_3$ yields the second best performance on the *IPPL* distortion, on which MOVIE is the best-performing algorithm. VQM and MOVIE are the second best-performing algorithms on the *MPEG-2* distortion.

Table 5.2: Performances of ViS$_3$ and other VQA algorithms measured on different types of distortion on the three video databases. The best-performing result is bolded and the second best-performing result is italicized and bolded.

Database	Distortion	PSNR	VQM	MOVIE	TQV	ViS$_3$
SROCC						
LIVE	WLPL	0.621	***0.817***	0.811	0.754	**0.845**
	IPPL	0.472	**0.802**	0.715	0.742	***0.788***
	H.264	0.473	0.686	***0.764***	**0.769**	0.757
	MPEG-2	0.383	0.718	***0.772***	**0.785**	0.730
	All data	0.523	0.756	0.789	***0.802***	**0.816**
IVPL	DIRAC	0.860	***0.891***	0.888	0.786	**0.926**
	H.264	***0.866***	0.862	0.823	0.672	**0.876**
	IPPL	0.711	0.650	**0.858**	0.629	***0.807***
	MPEG-2	0.738	0.791	***0.823***	0.557	**0.834**
	All data	0.728	0.845	***0.880***	0.701	**0.896**
CSIQ	H.264	0.802	0.919	0.897	**0.955**	***0.920***
	WLPL	0.851	0.801	**0.886**	0.842	***0.856***
	MJPEG	0.509	0.647	**0.887**	***0.870***	0.789
	SNOW	0.759	0.874	***0.900***	0.831	**0.908**
	AWGN	0.906	0.884	0.843	***0.908***	**0.928**
	HEVC	0.785	0.906	**0.933**	0.902	***0.917***
	All data	0.579	0.789	0.806	***0.814***	**0.841**
CC						
LIVE	WLPL	0.657	0.812	***0.839***	0.777	**0.846**
	IPPL	0.497	***0.800***	0.761	0.794	**0.816**
	H.264	0.571	0.703	**0.790**	***0.788***	0.773
	MPEG-2	0.395	0.737	***0.757***	**0.794**	0.746
	All data	0.549	0.770	0.811	***0.815***	**0.829**
IVPL	DIRAC	0.878	***0.898***	0.870	0.811	**0.936**
	H.264	0.855	***0.869***	0.845	0.744	**0.898**
	IPPL	0.673	0.642	**0.842**	0.735	***0.802***
	MPEG-2	0.718	***0.836***	0.824	0.533	**0.912**
	All data	0.723	0.847	***0.879***	0.722	**0.896**
CSIQ	H.264	0.835	0.916	0.904	**0.965**	***0.918***
	WLPL	0.802	0.806	**0.882**	0.784	***0.850***
	MJPEG	0.460	0.641	**0.882**	***0.871***	0.800
	SNOW	0.769	0.840	***0.898***	0.846	**0.908**
	AWGN	**0.949**	0.918	0.855	0.930	***0.916***
	HEVC	0.805	0.915	**0.937**	0.913	***0.933***
	All data	0.565	0.769	0.788	***0.795***	**0.830**



riverbed—H.264 compression

sunflower—MPEG-2 compression

Figure 5.2: Scatter-plots of logistic-transformed scores predicted by ViS$_3$ versus subjective scores on the *H.264* and *MPEG-2* distortion of the LIVE database. The second row shows representative frames of the two videos corresponding to the outliers, which correspond to the red square markers in the scatter plots.

PSNR, VQM, and MOVIE are competitive on both the *DIRAC* and *H.264* distortion.

On the CSIQ database, TQV and ViS$_3$ are the two best-performing algorithms on the *H.264* distortion; ViS$_3$ and MOVIE are the two best-performing algorithms on three types of distortion (*WLPL*, *SNOW*, and *HEVC*); MOVIE and TQV are the two best-performing algorithms on the *MJPEG*. On the *AGWN* distortion, ViS$_3$ and TQV are competitive with PSNR, which is known to perform well for additive Gaussian white noise.

Generally, ViS$_3$ excels on the *H.264* compression distortion and the wavelet-based compression distortion (*DIRAC*, *SNOW*); and ViS$_3$, VQM, and MOVIE excel on the *WLPL* distortion. ViS$_3$ also performs well on the *MPEG-2*, *HEVC*, and *AWGN* distortion. However, ViS$_3$ does not perform well on the *MJPEG* compression distortion compared to MOVIE and TQV.

5.3.2 Analysis on different types of camera motion

From the descriptions of the three tested video databases, it is acknowledged that the videos contained in these databases have different types of camera motion. The camera motion ranges from static to panning, tilting, and zooming. The STS images used in the ViS$_3$ algorithm have been used to characterized the motion information of the video in previous researches [24, 25]. Thus, a study on the performance of ViS$_3$ algorithm with respect to camera motion would be an interesting future topic.

Since the camera motion characteristic is out of the scope within this dissertation, we manually classify camera motion based on our own observation of the videos and based on the dominated motion types if the video has multiple types of camera motion. The videos from three tested video databases are assigned to three different categories of camera motion: static, zooming, and panning/tilting. The logistic transformed scores of these videos are used to evaluate the performance of the ViS$_3$ algorithm.

Table 5.3 shows the preliminary results of our study about the algorithm performance with respect to camera motion types. As seen from Table 5.3, the overall performance of ViS$_3$ algorithm does not depend on the type of camera motion. However, the spatiotemporal dissimilarity part ViS$_2$ does not perform well for videos with static camera. This is due to the fact that for static camera, human subjects tend to focus more on the moving object, therefore, the dissimilarity values of the STS images correspond to the moving objects or regions should be weighted more in the constitution of the ViS$_2$ index.

Table 5.3: Performances of ViS$_3$ on the three video databases with three different categories of camera motion. The best-performing result is bolded and the second best-performing result is italicized and bolded.

Camera motion		Static	Pan/Tilt	Zoom	All types
ViS$_2$					
SROCC	LIVE	0.585	0.755	0.837	0.731
	IVPL	0.835	0.896	0.828	0.818
	CSIQ	0.780	0.878	0.831	0.831
CC	LIVE	0.598	0.767	0.858	0.740
	IVPL	0.832	0.890	0.812	0.822
	CSIQ	0.784	0.875	0.814	0.830
ViS$_3$					
SROCC	LIVE	0.729	0.852	0.851	0.816
	IVPL	0.925	0.904	0.884	0.896
	CSIQ	0.822	0.874	0.835	0.841
CC	LIVE	0.725	0.865	0.884	0.829
	IVPL	0.932	0.895	0.886	0.896
	CSIQ	0.811	0.861	0.806	0.830

5.3.3 Performance with different GOF sizes

As we mentioned in Section 3.2.1, for ViS$_1$, the size of the GOF used in Equations (3.9), (3.11), and (3.12) is a user-selectable parameter (N). The results presented in the previous subsection were obtained with a GOF size of $N = 8$. To investigate how the prediction performance varies with different GOF sizes, we computed SROCC and CC values for ViS$_1$ and ViS$_3$ using values of N ranging from 4 to 16. The results of this analysis are listed in Table 5.4.

As shown in the upper portion of Table 5.4, the performance of ViS$_1$ tends to increase with larger values of N. This trend may partially be attributable to the fact that a larger GOF size can give rise to a more accurate estimate of the motion, and thus perhaps a more accurate account of the temporal masking. Nonetheless, as demonstrated in the lower portion of Table 5.4, ViS$_3$ is relatively robust to small changes in N. The choice of $N = 8$ generally provides good performance on all three databases. However, the optimal choice of N remains an open research question.

Table 5.4: Performances of ViS$_3$ on three video databases with different GOF size. The results show that ViS$_3$ is robust with the change of the GOF size.

GOF size		4	6	8	10	12	16
ViS$_1$							
SROCC	LIVE	0.754	0.759	0.762	0.767	0.770	0.768
	IVPL	0.868	0.871	0.872	0.871	0.873	0.874
	CSIQ	0.751	0.753	0.757	0.758	0.759	0.760
CC	LIVE	0.778	0.783	0.785	0.789	0.791	0.793
	IVPL	0.860	0.862	0.863	0.865	0.866	0.868
	CSIQ	0.733	0.736	0.739	0.740	0.742	0.743
ViS$_3$							
SROCC	LIVE	0.818	0.817	0.816	0.814	0.813	0.812
	IVPL	0.897	0.897	0.896	0.897	0.897	0.896
	CSIQ	0.840	0.840	0.841	0.841	0.841	0.841
CC	LIVE	0.833	0.831	0.829	0.828	0.827	0.825
	IVPL	0.896	0.896	0.896	0.896	0.897	0.896
	CSIQ	0.829	0.829	0.830	0.830	0.830	0.830

5.3.4 Statistical significance analysis

To assess the statistical significance of differences in performances of ViS$_3$ and other VQA algorithms, we used an F-test to compare the variances of the residuals (errors) of the algorithms' predictions [75]. If the distribution of residuals is sufficiently Gaussian, an F-test can be used to determine the probability that the residuals are drawn from different distributions and are thus statistically different.

To determine whether the residuals of an algorithm have Gaussian distributions, we performed the Jarque–Bera (JB) test (see Ref. 61) on the residuals to measure the JBSTAT value. If the JBSTAT value is smaller than a critical value, then the distribution of residuals is significantly Gaussian. If the JBSTAT value is greater than the critical value, then the distribution of residuals is not Gaussian. The JB test results show that for the LIVE database, all the algorithms pass the JB test and their residuals have Gaussian distributions. On the IVPL database, only PSNR does not pass the JB test. On the CSIQ database, only VQM and ViS$_3$ pass the JB test.

We performed an F-test with 95% confidence to compare the residual variances

Table 5.5: Statistical significance relationship between each pair of algorithms on the three video databases. A “0” value implies that variances of residuals between the algorithm indicated by the column and the algorithm indicated by row are not significantly different. A “+” sign implies that the algorithm indicated by the column has significantly smaller residual variance than the algorithm indicated by the row. A “−” sign implies that the algorithm indicated by the column has significantly larger residual variance than the algorithm indicated by the row.

		PSNR	VQM	MOVIE	TQV	ViS$_3$
LIVE	PSNR		+	+	+	+
	VQM	−		0	0	0
	MOVIE	−	0		0	0
	TQV	−	0	0		0
	ViS$_3$	−	0	0	0	
IVPL	VQM			0	−	+
	MOVIE		0		−	0
	TQV		+	+		+
	ViS$_3$		−	0	−	
CSIQ	VQM					+
	ViS$_3$		−			

of the algorithms whose distributions of residuals are significantly Gaussian. If the variances are significantly different, we conclude that the two algorithms are significantly different. The algorithm that yields smaller variance of residuals is concluded to have better prediction performance.

Table 5.5 shows the F-test results between each pair of algorithms whose distributions of residuals are significantly Gaussian. A “0” value implies that two algorithms are not significantly different in performance. A “+” sign implies that the algorithm indicated by the column has significantly smaller residual variance than the algorithm indicated by the row, and therefore, it has better performance. A "−" sign implies that the algorithm indicated by the column has significantly larger residual variance than the algorithm indicated by the row, and therefore, it has worse performance.

As seen from Table 5.5, on the LIVE database, the variance of residuals yielded by PSNR is significantly larger than the variances of residuals yielded by the other algorithms, and therefore, PSNR is significantly worse than the other algorithms. The difference in residuals of ViS$_3$ and either of VQM, MOVIE, or TQV is not statistically significant. On the IVPL database, the variance of residuals yielded by TQV is significantly larger than the variances of residuals yielded by VQM, MOVIE, and ViS$_3$, and therefore, VQM, MOVIE, and ViS$_3$ are significantly better than TQV on this database. On both IVPL and CSIQ databases, the variance of residuals yielded by VQM is significantly larger than the variance of residuals yielded by ViS$_3$, and therefore, ViS$_3$ is significantly better than VQM on these databases.

Although ViS$_3$ is not significantly better than MOVIE on any of the three databases, it should be noted that MOVIE is not significantly better than VQM on any of the three database while ViS$_3$ is significantly better than VQM on the IVPL and CSIQ databases. Moreover, MOVIE requires more computation time than ViS$_3$. Specifically, using a modern computer (Intel Quad Core at 2.66 GHz, 12 GB RAM DDR2 at 6400 MHz, Windows 7 Pro 64-bit, Matlab R2011b) to estimate the quality of a 10-second video of size 352×288 (300 frames total), MOVIE requires about 200 minutes, whereas basic Matlab implementations of VQM and ViS$_3$ require about 1 and 7 minutes, respectively. Reducing the computational complexity in the image/video processing algorithms is a new project that we are currently working on. Experimental results in Refs. 76, 77 show significant improvement in computational complexity via micro-architectural analysis of image quality assessment algorithms. These results provide a solid foundation for the performance improvement of video quality assessment algorithms.

5.4 Chapter summary

In this chapter, we analyzed prediction performance of the proposed ViS$_3$ algorithm and compared to some other VQA algorithms. The evaluation is performed on three publicly available video-quality databases with five evaluation criteria. We also performed some tests to judge the robustness of the ViS$_3$ algorithm and evaluate the significance difference in performances. Experimental results show that ViS$_3$ is better than current VQA algorithms and is robust to the change of the GOF size.

CHAPTER 6

NO REFERENCE MOTION JPEG2000 QUALITY ASSESSMENT

6.1 Introduction

The JPEG2000 standard has become popular in the field of image compression due to its high coding performance. Its extension to video compression, the Motion JPEG2000 (MJ2K), has been used as an efficient compression standard in various applications. With the superior coding performance of JPEG2000 compression standard such as strong error resiliency, low latency, high compression performance, and good perceptual quality comparing to the other compression standards [78–82], it is reasonable to expect the similar performance of MJ2K in comparing to other video compression standards. Experiments performed by various researchers have shown that the MJ2K has better compression performance than Motion JPEG and MPEG-2, and has competitive performance with Intra-coding H.264/AVC video compression in high-quality applications [81].

In general, MJ2K provides advantage features such as scalability, Region of Interest coding, rate control, error resiliency, and no blocky artifacts that beyond the ability of Intra AVC coding. Furthermore, the intra-coded only mechanism in MJ2K makes it easy to access individual frames and therefore, efficient to support large dynamic range. MJ2K is widely used in video applications, which require fast, frequent, and convenient frame access; high-quality high-resolution imaging (medical and satellite); or video applications requiring real-time simple encoding. As a result, JPEG2000 has been adopted as an archive format by the Digital Cinema Initiative (DCI), which also deals with large image formats (2K, 4K) [81, 83].

In video applications such as compression, transmission, archiving, etc., it is difficult and even unable to avoid degradation of video quality due to the effects of implementation constraints (limited bandwidth, limited storage, etc.). These effects impact the video's visual perception and cause annoyance to the observers. Therefore, it is necessary and sufficient to maintain quality of the video at the receiver/storage as high as possible with/without knowledge of the original video. An algorithm that can predict video quality in an accurate and reliable manner is highly needed, especially when the original video is not available.

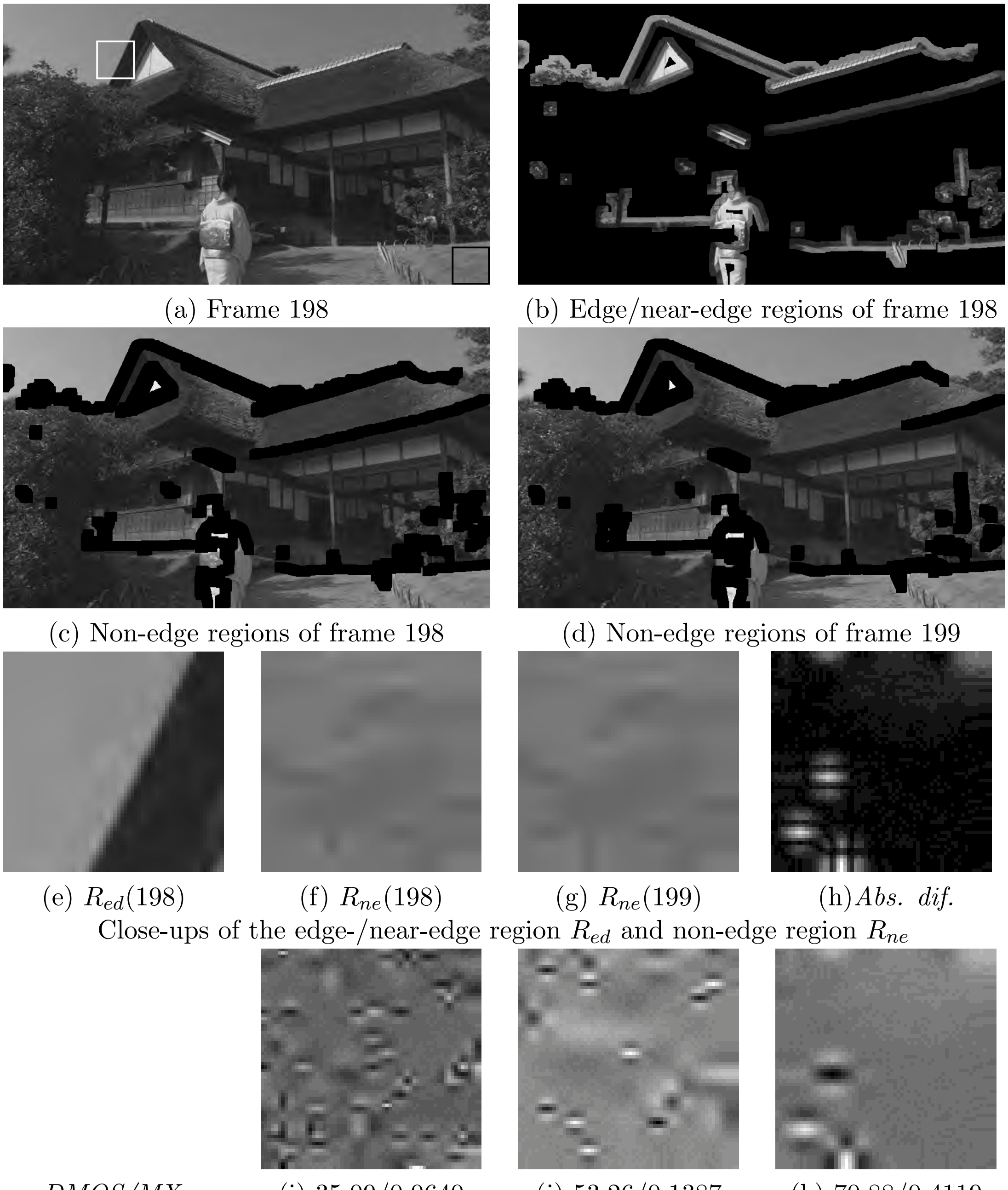
The MJ2K is intra-coded where each frame is an image compressed by JPEG2000 standard. To estimate quality of MJ2K videos, it is intuitive and straightforward to estimate quality of each frame using an image-based quality estimators and then collapse the indices over time. Many algorithms are developed to predict the quality of JPEG2000-compressed images, the general approach involves either estimating the amount of blurring artifacts or estimating the perceived ringing artifacts, or a combination of the two estimates.

It is known that the destruction of sharp regions due to JPEG2000 encoding reduces visual quality, resulting in the blurring artifacts appear in the images. The effect of blurring artifacts can be quantified by the average edge width [84], via the edge spread along the gradient and its perpendicular directions 85, or based on either 1-D or 2-D kurtosis in the discrete cosine transform domain of general image blocks [86], based on the total log-energy of the high frequency components in the wavelet domain [87]. Although these blurriness/sharpness algorithms have shown competitive performance at predicting the quality of blurred images [87, 88], they are often failed to estimate effects of blurring artifacts to the quality of JPEG2000-compressed images. This is because images compressed by the JPEG2000 compression standard also exhibit ringing artifacts, which often appear around the strong edges and locally produce haloes and/or rings in the images. To overcome this limitation, researchers have incorporated effects of both blurring and ringing artifacts in their algorithms [89–92] to predict quality of JPEG2000-compressed images.

Instead of an attempt to quantify predefined artifacts (blurring and ringing) in the JPEG2000-compressed images, some researchers extract features from the pixels/regions in the image and study the changes of these features with respect to image quality. Sheikh *et al.* [93] presented a natural scene statistics model of visual quality loss via the wavelet subband probabilities. Sazzad *et al.* [94] estimated image quality based on pixel distortions and edge information. These algorithms employ a training step to select optimal parameters. In Ref. 95, Zhang *et al.* proposed an algorithm that introduces a basic activity map of general pixels via a pixel classification into monotone-changing, zero-crossings, or inactive pixel. The activity map is then weighted by structural content and pooled to yield an estimate of image quality.

While there is an abundance of algorithms that are designed to estimate quality of JPEG2000 images, only a handful of algorithms has been proposed for MJ2K video quality assessment. The algorithm proposed by Nishikawa and Kiya [96] focuses on quality of MJ2K in a packet loss scenario. In fact, each frame in the MJ2K videos is intra-coded by the JPEG2000 compression standard. As demonstrated in Figure 6.1(a, b), the visible and popular artifacts in MJ2K videos are the blurring and ringing artifacts that are similar to artifacts in JPEG2000-compressed still images. Therefore, researchers often rely on the image-based quality algorithm to estimate quality of MJ2K videos via a pooling stage of frame quality indices over time.

As shown in Figure 6.1, a typical frame of a MJ2K videos (a) contains both the blurring and ringing artifacts, especially around the sharp edges of the image content. For example, the edges around the roof of the house are smoothed out and the ringing artifacts appear as the oscillations in the regions around the edges. The close-up regions in Figure 6.1(b) show the visibility of blurring and ringing artifacts at the selected regions. To quantify effects of these artifacts to video quality, in



(a) Frame 198

(b) Edge/near-edge regions of frame 198

(c) Non-edge regions of frame 198

(d) Non-edge regions of frame 199

(e) $R_{ed}(198)$ (f) $R_{ne}(198)$ (g) $R_{ne}(199)$ (h)*Abs. dif.*

Close-ups of the edge-/near-edge region R_{ed} and non-edge region R_{ne}

DMOS/MX (i) 35.09/0.0640 (j) 53.26/0.1387 (k) 70.88/0.4119

Temporal difference between consecutive frames from three compression levels

Figure 6.1: Representative frames and various types of distortion appear in a MJ2K video and their selected close-ups regions for better visibility.

the next sections of this chapter, we applied our image-based quality algorithm that predicts quality of JPEG2000-compressed images based on the analysis of artifacts in the edge/near-edge regions [5]. Blurring artifacts are estimated by the wavelet-based sharpness algorithm, FISH [87], and ringing artifacts are estimated by the local variance of the high frequency components measured by the Laplace filter. As we had demonstrated in Ref. 5, the proposed algorithm is competitive with current state-of-the-art algorithms in predicting quality of JPEG2000-compressed still images.

The video quality can be potentially estimated by pooling the frame quality indices over time. However, this simple technique is usually criticized for not being well-correlated with subjective ratings of quality because it ignores the role of motion information to the perception of video quality. It is known that the visibility of distortion in the videos is reduced in the fast-moving regions comparing to the slow-moving regions [3, 22]. We employ a technique proposed in Chapter 3 and in Ref. [97] to overcome this limitation of frame-based quality estimators.

The above approach estimates quality of MJ2K video by analyzing the artifacts in the edge/near-edge regions. This approach works reasonably well for predicting quality of the still images, because the edge/near-edge regions play an important role in early vision and contain image details, while the non-edge regions do not contain image details that are of interest to human subjects. However, in the MJ2K videos, due to the temporal transition between video frames, the image structures, which contain compression ringing artifacts, in the non-edge regions fluctuate over time and generate temporal flickering artifact. This temporal flickering artifact appears as "snow" distortion in the videos, and impacts video quality.

To illustrate the effect of temporal flickering, Figure 6.1(c, d) show the non-edge regions extracted from two consecutive frames of the same video. The close-ups in Figure 6.1(f, g) show the non-edge regions cropped from black bounding box of these frames. Despite the similarity of these close-ups in terms of image structure and light intensity, when the video is playing, the temporal difference of these regions, represented by the absolute difference regions in Figure 6.1(h), exhibit temporal flickering that appear as snow artifacts in the videos. We also show the temporal difference in grayscale color of the same spatial and temporal location extracted from three videos with increasing compression ratio from left to right in Figure 6.1 (i, j, k). These differences are quantified by the absolute value of the average difference between two regions (denoted by MX). Our preliminary computation for the selected regions shows that this MX value is getting bigger when the video is more compressed or the quality is getting worse. Therefore, this measurement of temporal change can potentially reflect the quality of MJ2K videos.

In this chapter, we propose a no-reference VQA algorithm for MJ2K videos called EDVQ (EDge-based Video Quality), which is based on two stages to estimate perceptual visual artifacts in the videos. The quality of a MJ2K video is estimated by quantifying the blurring and ringing artifacts that appear in the edge/near-edge regions, and the temporal change of local light intensity in the non-edge regions. The first stage of the EDVQ algorithm estimates quality of each JPEG2000-compressed frame using the EDIQ algorithm proposed in Ref. 5. In the second stage, for each group of $N = 8$ consecutive frames (denoted by GOF), we compute the absolute value of average temporal difference in the non-edge regions for each pair of consecutive frames. The representative temporal change of the GOF is computed as the maximum of these absolute values at each local block in the non-edge regions. These values are then weighted by local motion magnitude, estimated by the Lucas-Kanade optical flow method, to account for the effect of motion to the visibility of distortion. The computed values are averaged over all blocks in the common non-edge regions of all frames in the GOF to yield an overall value of temporal change in the non-edge regions. By combining the values computed from two different stages, we yield a scalar number that represents quality of the input MJ2K video.

The remain of this chapter is organized as follows: In Section 6.2, we provide details of the EDVQ algorithm. Section 6.3 presents results of EDVQ on subsets of JPEG2000-compressed images and videos from CSIQ video quality database. Chapter summary is presented in Section 6.4.

6.2 Algorithm

By recognizing the two observations above, in this section, we describe our proposed no-reference VQA algorithm to for MJ2K videos, which employs the following steps as illustrated in Figure 6.2.

1. Compute a perceived quality map for each frame by using EDIQ algorithm [5]. The maps computed from all frames in each GOF are then averaged to yield a GOF-based perceived quality map. The dilated binary edge map computed from the middle frame of each GOF serves as the masking to separate the edge/near-edge regions and non-edge regions within each GOF.

2. Compute temporal change of light intensity within each GOF in a block-based fashion. The temporal change in each 8×8 block is computed as the maximum change of average light intensity between each pair of consecutive frames in the GOF. This process yields a map of temporal change due to minor oscillation.

3. Estimate magnitude of motion vectors in each video frame via Lucas-Kanade optical flow method [62]. The motion magnitude maps computed from all frames in each GOF are averaged to yield a GOF-based motion magnitude map.

4. Weight the perceived quality map and the temporal change map by the motion magnitude map to model the effect of motion to quality perception. The RMS values computed from the weighted perceived quality map in the edge/near-edge regions and the weighted temporal change map in the non-edge regions are

combined and averaged across all GOFs to yield a single scalar that represents the perceived quality of the input video.

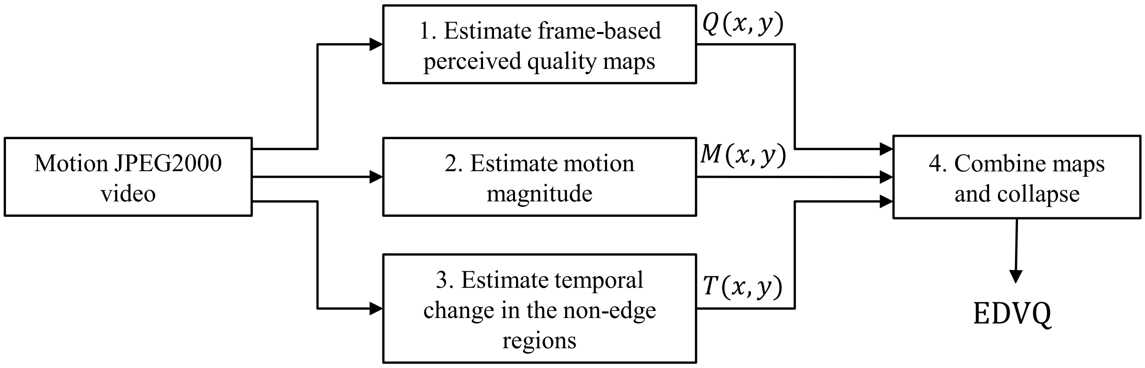

Motion JPEG2000 video
1. Estimate frame-based perceived quality maps
2. Estimate motion magnitude
3. Estimate temporal change in the non-edge regions
$Q(x,y)$
$M(x,y)$
$T(x,y)$
4. Combine maps and collapse
EDVQ

Figure 6.2: Block diagram of the EDVQ algorithm. Each group of N consecutive frames of the input MJ2K video is used to generate a local perceived quality map, $Q(x,y)$, a motion magnitude map $M(x,y)$, and a map of temporal change in light intensity $T(x,y)$. These maps are motion-weighted, region-masked, and collapsed into two scalar values, which are combined to yield a single scalar that represents the perceived quality of the input image.

The details of the algorithm are described as follows:

6.2.1 Apply EDIQ to every single frames

The quality of each frame from the MJ2K video is estimated using the EDIQ algorithm in Ref. 5, which is specifically designed for JPEG2000-compressed images. Here, we give a brief description of how the algorithm works on each standalone video frame.

A Estimate local perceived blurring artifact

The blurring artifact in a JPEG2000-compressed image is due to the attenuation of the high-frequency components in the image's frequency spectral. To estimate local perceived blurring artifact, we examine the energy in high-frequency wavelet subbands, which have been employed in our recent sharpness estimator, Fast Image SHarpness (FISH [87]).

As described in Ref. 87, FISH applies a three-level separable discrete wavelet transform (DWT) to the input image and measures the log-energy of the high-frequency DWT subbands. A global image sharpness is estimated based on a weighted geometric mean of these log-energies [see Figure 6.3(a)]. In addition, by clustering the DWT coefficients as illustrated in Figure 6.3(b), FISH can be modified to construct a map that represents the relative sharpness of each image region. The details of the FISH algorithm are described as below.

The grayscale input image is first decomposed into wavelet subbands using the Cohen-Daubechies-Fauraue 9/7 filters [98] with three levels of decomposition. Let S_{LH_k}, S_{HL_k}, S_{HH_k} denote the LH, HL, and HH subbands at DWT level $k \in \{1, 2, 3\}$ respectively. The log-energy of each subband at each decomposition level is given by

$$E_{XY_k} = \log_{10}\left(1 + \frac{1}{N_k}\sum_{i,j}[S_{XY_k}(i,j)]^2\right), \tag{6.1}$$

The total log-energy at decomposition level k is then given by

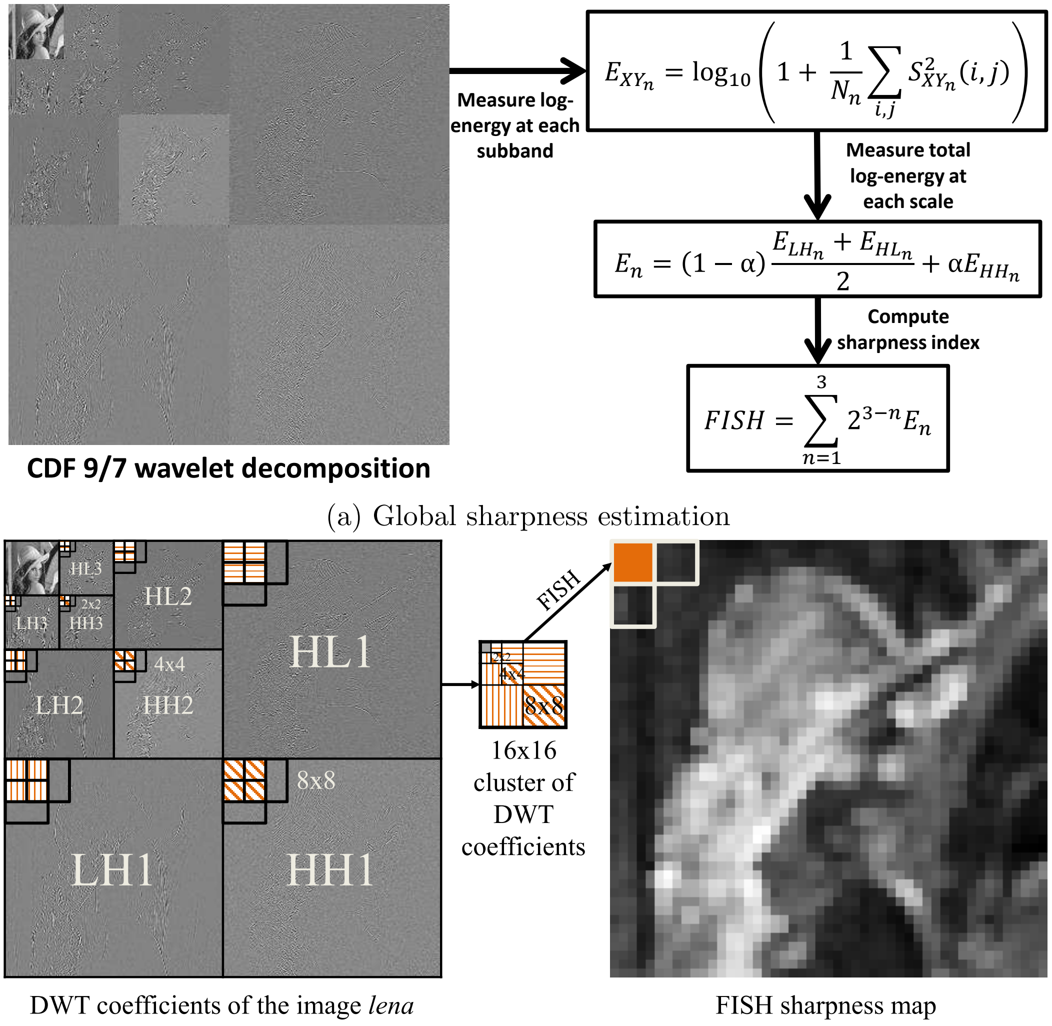
$$E_k = 0.2 \times \frac{E_{LH_k} + E_{HL_k}}{2} + 0.8 \times E_{HH_k}, \tag{6.2}$$

Finally, the three per-level log-energy values E_1, E_2, and E_3 are combined as

$$\text{FISH} = \sum_{k=1}^{3} 2^{3-k} E_k, \tag{6.3}$$

where the factors $2^{3-k} = \{4, 2, 1\}$ when $k = \{1, 2, 3\}$ are used to provide greater weight to the finer scales (higher-frequency bands). Here, FISH ≥ 0, is a single scalar that represents the overall global perceived sharpness of the image; the greater the FISH value, the greater the perceived image sharpness.

To generate a local sharpness map, we compute a collection of local FISH values in a block-based fashion by using the clusters of DWT coefficients corresponding to


$E_{XY_n} = \log_{10}\left(1 + \frac{1}{N_n}\sum_{i,j} S^2_{XY_n}(i,j)\right)$
Measure log-energy at each subband
Measure total log-energy at each scale
$E_n = (1-\alpha)\frac{E_{LH_n} + E_{HL_n}}{2} + \alpha E_{HH_n}$
Compute sharpness index
$FISH = \sum_{n=1}^{3} 2^{3-n} E_n$
CDF 9/7 wavelet decomposition
HL3
2x2
LH3
HH3
HL2
LH2
4x4
HH2
HL1
LH1
8x8
HH1
FISH
2x2
4x4
8x8
16x16 cluster of DWT coefficients
DWT coefficients of the image *lena*
FISH sharpness map

(a) Global sharpness estimation

(b) Clustering of wavelet coefficients to construct a local sharpness map

Figure 6.3: Illustration of global FISH algorithm (a) and DWT coefficients clustering into a wavelet block of size 16 × 16 to construct a local perceived sharpness map (b). The orange pixel and its two adjacent pixels in the sharpness map are shown according to the orange stripe set of DWT coefficients and two adjacent sets of DWT coefficients with 50% overlap. Note that, to promote visibility, the size of the blocks and sharpness map are not drawn to scale. (Figure from Ref. 5.)

each block b of size 16×16 in the image. As shown in Figure 6.3(b), each DWT subband is divided into small blocks of size 8×8, 4×4, and 2×2 for levels 1, 2, and 3, respectively with 50% overlap between neighboring blocks. These small blocks are assembled in clusters of size 16×16 to estimate local perceived sharpness. The FISH value is computed for every cluster of 16×16 DWT coefficients generated with 50% overlap between neighboring blocks of DWT coefficients in each subband, yielding a local FISH sharpness map.

As we observed, in the blurring regions, the high frequency components are attenuated and filtered, therefore, we can employ the FISH sharpness map as the local perceived blurring map of the input image, denoted by $B(x, y)$. The greater the $B(x, y)$ value, the smaller the perceived blurring at the corresponding spatial region.

B Estimate local perceived ringing artifact

The present of the ringing artifacts in the JPEG2000-compressed images appear in the form of oscillation in the regions, therefore the perceived ringing artifacts can be estimated by computing local variance in intensity within the detected ringing regions [90, 99, 100]. Here, we employ a simple method by applying a low-pass filter in order to suppress the image structures from the distorted image. The difference image between the input compressed image and the low-pass filtered image is assumed to contain the ringing artifacts. Local variance is computed from the difference image in a block-based fashion to estimate the perceived ringing artifacts. The details of this step are described as follows.

The difference image between the input image I and the low-pass filtered-image $\hat{I}$ is denoted by K, which is given by $K = I - \hat{I}$. In other word, this difference image is obtained by filtering the input image with a Laplace filter. The kernel of the Laplace filter used here is :

$$ker = \frac{1}{4}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Next, we compute local variance of the difference image K in a block-based fashion. Let $\mu(b)$ and $\sigma^2(b)$ denote the overall mean and variance of block b of size 16×16 in the difference image K. These values are given by

$$\mu(b) = \frac{1}{16 \times 16}\sum_{x=1}^{16}\sum_{y=1}^{16} b(x,y), \tag{6.4}$$

$$\sigma^2(b) = \frac{1}{16 \times 16 - 1}\sum_{x=1}^{16}\sum_{y=1}^{16}[b(x,y) - \mu(b)]^2. \tag{6.5}$$

The perceived ringing artifact at location of block b in the input image I is assumed to be proportional to the standard deviation $\sigma(b)$ of the corresponding region at the same location in image K. Specifically, let $R(b)$ denote the perceived ringing artifact at location of block b in the input image I, which is defined as follows

$$R(b) = \sigma(b) = \frac{1}{16}\sqrt{\sum_{x=1}^{16}\sum_{y=1}^{16}[b(x,y) - \mu(b)]^2}. \tag{6.6}$$

The greater the $R(b)$ value, the greater the perceived ringing artifact at location of block b. The process is performed on every block b of size 16×16 from the difference image K with 50% overlap between neighboring blocks, yielding a map R that represents the local perceived ringing artifacts in the image.

C Combine maps into perceived quality maps

For the given input image, we have computed the perceived blurring map $B(x,y)$, the perceived ringing map $R(x,y)$. We now combine these two maps into a perceived quality map of the input image via a point-by-point division

$$Q(x,y) = \begin{cases} \dfrac{B(x,y)}{\sqrt{R(x,y)}} & \text{if } R(x,y) > p \\ \\ \dfrac{B(x,y)}{\sqrt{p}} & \text{otherwise.} \end{cases} \tag{6.7}$$

The use of parameter $p = 1$ is to prevent the division by zero and to differentiate the ringing artifact from the remained image structure. When the perceived ringing artifact is sufficiently small, the quality of the input image will solely be estimated by the perceived blurring artifact. When the perceived ringing artifact is relative large, the perceived image quality is relative low.

D Apply to the video frames

We apply the previous steps to each frame of the MJ2K video to yield a set of perceived quality maps denote by $Q_t(x, y)$ where t is the index of the computed frame. The perceived quality maps computed from all frames in each GOF are averaged to yield the GOF-based quality map of that group. Specifically, let $\bar{Q}_k$ denote the GOF-based perceived quality map of the k^{th} GOF, the value at point (x, y) of the $\bar{Q}_k$ map is computed via :

$$\bar{Q}_k(x,y) = \frac{1}{N} \sum_{\tau=1}^{N} Q_{N(k-1)+\tau}(x,y). \tag{6.8}$$

where $N = 8$ represents the number of consecutive frames in the GOF.

For each GOF, we compute the overall dilated binary edge map from the middle frame in the group. Specifically, to determine the locations of strong edges, we apply the Canny edge detection [101] with two thresholds (low 0.1, high 0.4) to the middle frame to obtain a binary edge map. Let E_k denote the binary edge map of the k^{th} GOF, the binary value at point (x, y) of the binary edge map E_k is given by

$$E_k(x,y) = \begin{cases} 1 & \text{if } E_k(x,y) \text{ is an edge point,} \\ 0 & \text{if } E_k(x,y) \text{ is not an edge point.} \end{cases} \tag{6.9}$$

Each edge point $E_k(x, y)$ in the binary edge map is expanded to a $d \times d$ square with the center located at that point using image morphological processing (dilation) to obtain a dilated binary edge map D_k. The regions of the input image corresponding to $D_k(x, y) = 1$ are considered the edge/near-edge regions of the k^{th} GOF, denoted by R_{ed}^k; the regions of the input image corresponding to $D_k(x, y) = 0$ are considered the non-edge regions of the k^{th} GOF, denoted by R_{ne}^k.

6.2.2 Estimate effect of temporal change

As stated in previous section and in Figure 6.1, the temporal change of light intensity in the non-edge regions has important effects on the video quality. These changes appear as temporal flickering in the non-edge regions, the larger the temporal change in light intensity, the worse the video quality.

To estimate the effect of temporal change to video quality, we employ a block-based approach to each frame. For each block of 8×8 pixels, we compute the temporal change (temporal difference) of light intensity for each pair of consecutive frames within the GOF. This temporal change of block b between the t^{th} and $(t+1)^{th}$ frame is computed as the absolute difference in average light intensity. Let $\mu_t(b)$ and $\mu_{t+1}(b)$ denote the arithmetic mean of pixel values in block b of the t^{th} and $(t+1)^{th}$ frame respectively. Let $G(b)$ denotes the maximum absolute difference between $\mu_t(b)$ and $\mu_{t+1}(b)$ within the GOF. Specifically, for the k^{th} GOF, $G_k(b)$ is defined as

$$G_k(b) = \max_{\tau=N(k-1)+1}^{Nk} |\mu_\tau(b) - \mu_{\tau+1}(b)|. \tag{6.10}$$

The effect of temporal change to video quality at the position of block b in the k^{th} GOF is computed as follows

$$T_k(b) = \begin{cases} G(b) & \text{if } G(b) < 25 \\ 0 & \text{otherwise.} \end{cases} \tag{6.11}$$

A threshold of 25 is employed to account for the fact that if the temporal change is greater than a threshold, it is highly possible that the temporal change is caused by content change between frames and does not affect video quality. Otherwise, the greater the temporal change $T_k(b)$, the more degraded the video quality at the location of block b in the k^{th} GOF.

6.2.3 Lucas Kanade motion estimation

Both the perceived quality and temporal change maps do not reflect the fact that the visibility of distortion is significantly reduced when the speed of motion is large [3, 22]. Alternatively, the distortion in slow-moving regions is more visible than the distortion in fast-moving regions.

To model this effect of motion, we measure the speed of motion in different regions of the video by using the optical flow method designed by Lucas and Kanade [62] as in Chapter 3 to the input MJ2K video. The video is pre-filtered by a low-pass Gaussian filter to eliminate the effect of noise to the estimated motion vectors. The Gaussian kernel is chosen as a 11×11 window size and a standard deviation of 3. For the k^{th} GOF, the motion magnitude matrices computed from all its frames are averaged to yield an average motion magnitude matrix $\bar{V}_k$. This matrix is rescaled to the size of the video frame by using nearest-neighbor interpolation to obtain the GOF-based motion magnitude map of the k^{th} GOF, denoted by M_k.

6.2.4 Combine maps and compute perceived quality index

For each GOF, we have computed the GOF-based perceived quality map $\bar{Q}_k$, the GOF-based temporal change map T_k, the GOF-based motion magnitude map M_k. The perceived quality map $\bar{Q}_k$ and the temporal change map T_k are then point-by-point weighted by the motion magnitude map M_k to yield corrected perceived quality and temporal change maps $\tilde{Q}_k$ and $\tilde{T}_k$ respectively as follows

$$\tilde{Q}_k(x,y) = \bar{Q}_k^2(x,y) \times (1 + M_k), \tag{6.12}$$

$$\tilde{T}_k(x,y) = \frac{1}{1 + \exp\left(\frac{-T_k(x,y)}{1+M_k(x,y)}\right)}. \tag{6.13}$$

To estimate perceived quality index of each GOF, we compute the root mean square (RMS) value of the perceived quality map in the edge/near-edge regions and the RMS value of the temporal change map in the non-edge regions. These RMS values are given by

$$\alpha_k = \frac{1}{N_{ed}} \sum_{E_k(x,y)=1} \tilde{Q}_k(x,y), \tag{6.14}$$

$$\beta_k = \frac{1}{N_{ne}} \sum_{E_k(x,y)=0} \tilde{T}_k(x,y), \tag{6.15}$$

$$VQ_k = \frac{\log(1+\alpha_k)}{\beta_k} \tag{6.16}$$

where N_{ed} and N_{ne} are the number of pixels in the edge-/near-edge and non-edge regions respectively. The overall perceived video quality, denoted by $EDVQ$, is computed as the arithmetic mean of all perceived quality estimates VQ_k via

$$\text{EDVQ} = \frac{1}{K} \sum_{k=1}^{K} VQ_k. \tag{6.17}$$

where K is the number of GOF in the video. Here, EDVQ is a single scalar that represents the overall perceived quality of the given video. The greater the VQ value, the better the video quality. A value VQ = 0 indicates that the video is completely distorted and has the worst quality.

Table 6.1: Performances of EDVQ and other algorithms on the subset of MJ2K compressed videos. The best-performing result is bolded and the second best-performing result is italicized and bolded.

	SROCC	**CC**	**RMSE**	**OR**	**OD**
Full-reference VQA algorithms					
PSNR	0.759	0.794	9.205	5.56%	14.682
MS-SSIM	***0.865***	**0.895**	**6.771**	**0.00%**	**0**
VQM	**0.874**	0.878	7.267	**0.00%**	**0**
No-reference VQA algorithms					
NSS	0.509	0.488	13.227	5.56%	14.068
SAZHV	0.345	0.390	13.955	13.89%	24.072
K1FB	0.304	0.305	14.435	11.11%	21.908
EDIQ	0.682	0.689	10.987	2.78%	6.025
EDVQ	0.852	***0.890***	***6.902***	**0.00%**	**0**

6.3 Experimental results

In this section, we analyze the performance of the EDVQ algorithm in predicting subjective ratings of quality on the subset of MJ2K videos from the CSIQ video database [74, 97]. The subset consists of 36 distorted videos from 12 different contents with 3 distorted versions per each content. We compared EDVQ with some full-reference VQA algorithms PSNR [28], MS-SSIM [43], and VQM [2]. PSNR and MS-SSIM were applied on a frame-by-frame basis, VQM was applied using their default implementations and settings. We also compare performances of other JPEG2000 specific no-reference IQA algorithms on a frame-based approach. These methods are NSS (developed by Sheikh *et al.* [93]), SAZHV (proposed by Sazzad *et al.* [94]), K1FB (Zhang *et al.* [95]), and EDIQ [5]. The raw predicted scores are transformed via a four-parameter logistic transform as in Equation 5.1.

The performance of each algorithm is shown in Table 6.1 in terms of five different criteria (SROCC, CC, RMSE, OR, and OD). The best-performing result is bolded, and the second best-performing result is italicized and bolded. These data indicates that EDVQ is better than PSNR and competitive with the MS-SSIM and VQM algorithms in terms of all five evaluation criteria for video quality prediction.

It is also understandable that other frame-based no-reference JPEG2000 IQA algorithms, when applied to estimate video quality, do not yield good prediction performance due to the lack of temporal information analysis. The frame-based EDIQ performed better than the other frame-based no reference JPEG2000 algorithms due to its superior performance for still images. However, frame-based EDIQ is still far from a competitor with EDVQ algorithm, which has an advantage of temporal weighting and temporal flickering analysis comparing to the other frame-based algorithms.

6.4 Chapter summary

In this chapter, we proposed a no-reference VQA algorithm for estimating quality of MJ2K videos. The algorithm, EDVQ, is based on our previous no-reference IQA algorithm, which is specifically designed for JPEG2000-compressed images, and an additional stage of analyzing the temporal change in the non-edge regions. These measurements are weighted and modified by the motion magnitude to account for the effects of motion to video quality. Experimental result shows that the proposed algorithm has a competitive performance with popular full-reference VQA algorithms and performs much better than other no-reference VQA algorithms, which are constructed from frame-based image quality of JPEG2000-compressed images.

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH

7.1 Conclusions

Through the contents of the previous chapters in this dissertation report, we have presented the general knowledge of video quality assessment and the efforts of the research community to develop VQA algorithms to tackle this problem. We also proposed our approaches to estimate video quality with/without the reference information and created the CSIQ video-quality database.

We developed a full reference VQA algorithm, ViS_3, that analyzes various two-dimensional space-time slices of the video to estimate perceived video quality degradation via two different stages. The algorithm adaptively applies two strategies in the MAD algorithm to groups of video frames with a model of temporal weighting to estimate perceived video quality degradation due to spatial distortion. Spatiotemporal correlation and an HVS-based model of spatiotemporal responses are applied to the STS images to estimate perceived video quality degradation due to spatiotemporal dissimilarity. The overall estimate of perceived video quality degradation is given as the geometric mean of the two measurements obtained from the two stages.

Via testing on various video-quality databases, we have demonstrated that our proposed full reference VQA algorithm, ViS_3, performs well in predicting video quality. ViS_3 does not only excel at predicting video quality for the entire database with varying types of distortion and varying distortion levels, but it also performs well on videos with a specific type of distortion. Our performance evaluation demonstrates that ViS_3 is either better than or statistically tied with current state-of-the-art VQA algorithms. A statistical analysis also shows that ViS$_3$ is significantly better than PSNR, VQM, and TQV in predicting the qualities of videos from specific databases.

We also presented our newly developed video-quality database that contains more videos and more distortion types comparing to the current available video-quality databases. Our video-quality database (CSIQ video database) consists of 12 reference videos and 216 distorted videos. All videos in this database are in raw YUV420 format with a resolution of 832 × 480 pixels, a duration of 10 seconds, and span a range of various frame rates. Each reference video has 18 distorted versions with six types of distortion; each distortion type has three different levels. A psycho-physical experiment was conducted following the SAMVIQ testing protocol with 35 subjects to collect subjective ratings of quality.

A no-reference VQA algorithm has been proposed in chapter 6 of this dissertation, which is specifically designed for motion JPEG2000 videos. The algorithm, EDVQ, estimates video quality based on a frame-based JPEG2000-compressed quality assessment and an addition analysis of temporal flickering in the non-edge (smooth) regions. Experimental results show that the approach in EDVQ algorithm can predict quality of MJ2K videos in a good correlation with subjective ratings. The EDVQ algorithm is better than other no-reference frame-based JPEG2000 quality algorithms and is competitive with some popular full-reference VQA algorithms.

7.2 Limitations and potential improvements

Yet, our proposed algorithms, ViS$_3$ and EDVQ, are not without limitations. One important limitation of ViS$_3$ is in regards to the potentially large memory requirements for long videos. The STS images of a long video can require a prohibitively large width or height for the dimension corresponding to time. In this case, one solution would be to divide the video into small chunks across time, where each chunk has a length of approximately 500 to 600 frames. The final result can be estimated via the mean of the ViS_3 values computed for each chunk.

Another limitation of both ViS_3 and EDVQ is that they currently take into account only the luminance component of the video. Further improvements may be realized by also considering degradations in chrominance. Another possible improvement might be realized by employing a more accurate pooling model of the spatiotemporal responses used in the spatiotemporal dissimilarity stage.

Equation (3.33) gives the same weight to the spatial distortion and spatiotemporal dissimilarity values. However, it would seem possible to adaptively combine the two values in a way that more accurately reflects the visual contribution of each degradation to the overall quality degradation. Our preliminary attempts to select the weights based on the video motion magnitudes, the difference in motion, or the variance of spatial distortion have not yielded significant improvements. We are currently conducting a psycho-physical study to better understand if and how the spatial distortion and spatiotemporal dissimilarity values should be adaptively combined.

The incorporation of visual-attention modeling is another avenue for potential improvements. Some studies have shown that visual attention can be useful for quality assessment (e.g., Refs. 39, 102–104). One possible technique for incorporating such data into ViS_3 would be to weight the maps generated during the computation of both ViS_1 and ViS_2 based on estimates of visual gaze data or regions-of-interest in both space and time. Another interesting avenue of future research would be to compare the ViS_1 and ViS_2 maps with gaze data to identify any existing relationships, and perhaps determine techniques for predicting gaze data based on the STS images.

The approach in EDVQ algorithm is based on the observation of MJ2K videos and is specifically designed this type of videos. Other types of videos have different characteristics of distortion and requires more sophisticated study. The most potential future work can be developed from the current study of the EDVQ algorithm is to construct a general no-reference VQA algorithm that can predict video quality without the requirement of the reference videos or the prior knowledge of the distortion characteristic. It would be perfect to have a no-reference algorithm that can handle all types of distortion as the ViS_3 algorithm. The EDVQ algorithm suggests that we can develop a no-reference VQA algorithm based on studying the characteristics of separate distortion types in the videos. Once we characterized the distortion-related features of the video, a machine learning mechanism can provide a great support of combining these features into the final video quality score of the video. We have already developed some algorithms for general image quality assessment [105, 106], which show promises in prediction performance with different types of distortion. Extended versions of these algorithms for video quality assessment are potential research topics that are currently under our study.

REFERENCES

[1] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Transactions on Image Processing* **19**, 1427–1441 (2010). [doi:10.1109/TIP.2010.2042111].

[2] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting* **50**, 312–322 (2004). [doi:10.1109/TBC.2004.834028].

[3] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication* **19**(2), 121–132 (2004). [doi:10.1016/S0923-5965(03)00076-6].

[4] D. M. Chandler, "Seven challenges in image quality assessment: Past, present, and future research," *ISRN Signal Processing* **2013** (2013).

[5] P. Vu, "On the use of image sharpness to JPEG2000 no-reference image quality assessment," Master's thesis, Oklahoma State University (2013).

[6] S. Chikkerur, V. Sundaram, M. Reisslein, and L. Karam, "Objective video quality assessment methods: A classification, review, and performance comparison," *IEEE Transactions on Broadcasting* **57**, 165–182 (2011). [doi:10.1109/TBC.2011.2104671].

[7] B. Girod, *Digital images and human vision*, MIT Press, Cambridge, MA, USA (1993).

[8] A. M. Eskicioglu and P. S. Fisher, “Image quality measures and their performance,” *IEEE Transactions on Communications* **43**(12), 2959–2965 (1995). [doi: 10.1109/26.477498].

[9] B. A. Wandell, *Foundations of Vision*, Sinauer Associates, Sunderland, MA, USA (1995).

[10] K. Seshadrinathan and A. Bovik, “Motion tuned spatio-temporal quality assessment of natural videos,” *IEEE Transactions on Image Processing* **19**, 335–350 (2010). [doi:10.1109/TIP.2009.2034992].

[11] Z. Wang and Q. Li, “Video quality assessment using a statistical model of human visual speed perception,” *J. Opt. Soc. Amer. AOpt. Image Sci. Vis.* **24**, B61–B69 (2007). [doi:10.1364/JOSAA.24.000B61].

[12] H. R. Sheikh and A. C. Bovik, “A visual information fidelity approach to video quality assessment,” in *First International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, *Proc. VPQM*, 23–25 (2005).

[13] M. Narwaria, W. Lin, and A. Liu, “Low-complexity video quality assessment using temporal quality variations,” *IEEE Transactions on Multimedia* **14**, 525–535 (2012). [doi:10.1109/TMM.2012.2190589].

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing* **13**, 600–612 (2004). [doi:10.1109/TIP.2003.819861].

[15] H. R. Sheikh and A. C. Bovik, “Image information and visual quality,” *IEEE Transactions on Image Processing* **15**, 430–444 (2006). [doi:10.1109/TIP.2005.859378].

[16] S. Wolf and M. Pinson, "In-service performance metrics for MPEG-2 video systems," in *Measurement Techniques of the Digital Age Technical Seminar*, *Proc. Made to Measure*, IAB, ITU and Technical University of Braunschweig, Montreux, Switzerland (1998).

[17] Y. Wang, T. Jiang, S. Ma, and W. Gao, "Novel spatio-temporal structural information based video quality metric," *IEEE Transactions on Circuits and Systems for Video Technology* **22**(7), 989–998 (2012). [doi:10.1109/TCSVT.2012.2186745].

[18] D. E. Pearson, "Variability of performance in video coding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, *Proc. ICASSP* **1**, 5–8 (1997). [doi:10.1109/ICASSP.1997.598844].

[19] D. E. Pearson, "Viewer response to time-varying video quality," in *Human Vision and Electronic Imaging III*, B. E. Rogowitz and T. N. Pappas, Eds., *Proc. SPIE* **3299**, 16–25 (1998).

[20] K. Seshadrinathan and A. C. Bovik, "Temporal hysteresis model of time varying subjective video quality," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, *Proc. ICASSP*, 1153 – 1156 (2011). [doi:10.1109/ICASSP.2011.5946613].

[21] M. A. Masry and S. S. Hemami, "A metric for continuous quality evaluation of compressed video with severe distortions, signal processing," *Signal Processing: Image Communication* **19**(2), 133–146 (2004). [doi::10.1016/j.image.2003.08.001].

[22] M. Barkowsky, B. Eskofier, R. Bitto, J. Bialkowski, and A. Kaup, "Perceptually motivated spatial and temporal integration of pixel based video

quality measures," in *Welcome to Mobile Content Quality of Experience, Proc. MobConQoE*, 4:1–4:7, ACM, (New York, NY, USA) (2007). [doi:10.1145/1577504.1577508].

[23] A. Ninassi, O. Le Meur, P. Le Callet, and D. Barba, "Considering temporal variations of spatial visual distortions in video quality assessment," *IEEE Journal of Selected Topics in Signal Processing* **3**(2), 253–265 (2009). [doi:10.1109/JSTSP.2009.2014806].

[24] C. Ngo, T. Pong, and H. Zhang, "On clustering and retrieval of video shots through temporal slices analysis," *IEEE Transactions on Multimedia* **4**, 446–458 (2002). [doi:10.1109/TMM.2002.802022].

[25] C. Ngo, T. Pong, and H. Zhang, "Motion analysis and segmentation through spatio-temporal slices processing," *IEEE Transactions on Image Processing* **12**, 341–355 (2003). [doi:10.1109/TIP.2003.809020].

[26] A. B. Watson and J. Albert J. Ahumada, "Model of human visual-motion sensing," *J. Opt. Soc. Am. A* **2**, 322–341 (1985). [doi:10.1364/JOSAA.2.000322].

[27] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Am. A* **2**, 284–299 (1985). [doi:10.1364/JOSAA.2.000284].

[28] ANSI T1.TR.74-2001, "Objective video quality measurement using a peak-signal-to-noise-ratio (psnr) full reference technique," tech. rep., American National Standards Institute, Ad Hoc Group on Video Quality Metrics, Washington, DC, USA (2001).

[29] E. Larson and D. Chandler, "Most apparent distortion: Full-reference image quality assessment and the role of strategy," *Journal of Electronic Imaging* **19**(1), 011006 (2010). [doi:10.1117/1.3267105].

[30] D. M. Chandler, M. M. Alam, and T. D. Phan, "Seven challenges for image quality research," *Proc. SPIE* **9014**, 901402–901402–14 (2014).

[31] VQEG, "Final report from the video quality experts group on the validation of objective models of video quality assessment, phase II," tech. rep., Video Quality Expert Group (2003). http://www.vqeg.org.

[32] L. Lu, Z. Wang, A. C. Bovik, and J. Kouloheris, "Full-reference video quality assessment considering structural distortion and no-reference quality evaluation of MPEG video," in *IEEE Intl Conf. on Multimedia and Expo*, *Proc. ICME* **1**, 61–64 (2002). [doi:10.1109/ICME.2002.1035718].

[33] P. Tao and A. M. Eskicioglu, "Video quality assessment using M-SVD," in *Image Quality and System Performance IV*, *Proc. SPIE* **6494** (2007). [doi:10.1117/12.696142].

[34] A. Pessoa, A. Falco, R. Nishihara, A. Silva, and R. Lotufo, "Video quality assessment using objective parameters based on image segmentation," *SMPTE Journal* **108**(12), 865–872 (1999). [doi:10.5594/J04308].

[35] J. Okamoto, T. Hayashi, A. Takahashi, and T. Kurita, "Proposal for an objective video quality assessment method that takes temporal and spatial information into consideration," *Electronics and Communications in Japan (Part I: Communications)* **89**(12), 97–108 (2006). [doi:10.1002/ecja.20265].

[36] S. O. Lee and D. G. Sim, "New full-reference visual quality assessment based on human visual perception," in *Digest of Technical Papers. International Conference on Consumer Electronics*, *Proc. ICCE*, 1–2 (2008). [doi:10.1109/ICCE.2008.4587874].

[37] M. Barkowsky, J. Bialkowski, B. Eskofier, R. Bitto, and A. Kaup, "Temporal

trajectory aware video quality measure," *IEEE Journal of Selected Topics in Signal Processing* **3**, 266–279 (2009). [doi:10.1109/JSTSP.2009.2015375].

[38] A. Bhat, I. Richardson, and S. Kannangara, "A new perceptual quality metric for compressed video," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, *Proc. ICASSP*, 933–936 (2009). [doi:10.1109/ICASSP.2009.4959738].

[39] U. Engelke, M. Barkowsky, P. Le Callet, and H. J. Zepernick, "Modelling saliency awareness for objective video quality assessment," in *Second International Workshop on Quality of Multimedia Experience*, *Proc. QoMEx*, 212–217 (2010). [doi:10.1109/QOMEX.2010.5516159].

[40] X. Gu, G. Qiu, X. Feng, D. Liu, and Z. Chen, "Region of interest weighted pooling strategy for video quality metric," *Telecommunication Systems* **49**(1), 63–73 (2012). [doi:10.1007/s11235-010-9353-8].

[41] M. Narwaria and W. Lin, "Scalable image quality assessment based on structural vectors," in *IEEE International Workshop on Multimedia Signal Processing*, *Proc. MMSP*, 1–6 (2009). [doi:10.1109/MMSP.2009.5293244].

[42] A. A. Stocker and E. P. Simoncelli, "Noise characteristics and prior expectations in human visual speed perception," *Nature Neuroscience* **9**, 578–585 (2006).

[43] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, **2**, 1398–1402 (2003).

[44] F. Lukas and Z. Budrikis, "Picture quality prediction based on a visual model," *IEEE Transactions on Communications* **30**, 1679–1692 (1982). [doi:10.1109/TCOM.1982.1095616].

[45] A. Basso, I. Dalgic, F. A. Tobagi, and C. J. van den Branden Lambrecht, "Study of MPEG-2 coding performance based on a perceptual quality metric," in *Proceedings of PCS96*, 263–268 (1996).

[46] C. J. van den Branden Lambrecht, "Color moving pictures quality metric," in *IEEE International Conference on Image Processing*, *Proc. ICIP* **1**, 885–888 (1996). [doi:10.1109/ICIP.1996.559641].

[47] P. Lindh and C. J. van den Branden Lambrecht, "Efficient spatio-temporal decomposition for perceptual processing of video sequences," in *IEEE International Conference on Image Processing*, *Proc. ICIP* **3**, 331–334 (1996). [doi:10.1109/ICIP.1996.560498].

[48] A. Hekstra, J. Beerends, D. Ledermann, F. de Caluwe, S. Kohler, R. Koenen, S. Rihs, M. Ehrsam, and D. Schlauss, "PVQM - a perceptual video quality measure," *Signal Processing: Image Communication* **17**(10), 781–798 (2002). [doi:10.1016/S0923-5965(02)00056-5].

[49] A. B. Watson, J. Hu, and J. F. McGowan, "Digital video quality metric based on human vision," *Journal of Electronic Imaging* **10**(1), 20–29 (2001). [doi:10.1117/1.1329896].

[50] C. Lee and O. Kwon, "Objective measurements of video quality using the wavelet transform," *Optical Engineering* **42**(1), 265–272 (2003). [doi:10.1117/1.1523420].

[51] E. Ong, X. Yang, W. Lin, Z. Lu, and S. Yao, "Video quality metric for low bitrate compressed videos," in *IEEE International Conference on Image Processing*, *Proc. ICIP* **5**, 3531–3534 (2004). [doi:10.1109/ICIP.2004.1421878].

[52] E. Ong, W. Lin, Z. Lu, and S. Yao, "Colour perceptual video quality metric," in

IEEE International Conference on Image Processing, *Proc. ICIP* **3**, III–1172–5 (2005). [doi:10.1109/ICIP.2005.1530606].

[53] M. Masry, S. Hemami, and Y. Sermadevi, "A scalable wavelet-based video distortion metric and applications," *IEEE Transactions on Circuits and Systems for Video Technology* **16**, 260–273 (2006). [doi:10.1109/TCSVT.2005.861946].

[54] P. Ndjiki-Nya, M. Barrado, and T. Wiegand, "Efficient full-reference assessment of image and video quality," in *IEEE International Conference on Image Processing*, *Proc. ICIP* **2**, II–125–II–128 (2007). [doi:10.1109/ICIP.2007.4379108].

[55] S. Li, L. Ma, and K. Ngan, "Full-reference video quality assessment by decoupling detail losses and additive impairments," *IEEE Transactions on Circuits and Systems for Video Technology* **PP**(99), 1 (2012). [doi:10.1109/TCSVT.2012.2190473].

[56] P. C. Teo and D. J. Heeger, "Perceptual image distortion," in *IEEE International Conference on Image Processing*, *Proc. ICIP* **2**, 982–986 (1994). [doi:10.1109/ICIP.1994.413502].

[57] S. Pechard, P. L. Callet, M. Carnec, and D. Barba, "A new methodology to estimate the impact of h.264 artefacts on subjective video quality," in *Third International Workshop on Video Processing and Quality Metrics*, *Proc. VPQM* (2007).

[58] A. B. Watson and J. Albert J. Ahumada, "A look at motion in the frequency domain," Technical Memo TM-84352, NASA (1983).

[59] P. V. Vu, C. T. Vu, and D. M. Chandler, "A spatiotemporal most-apparent-distortion model for video quality assessment," in *IEEE International Conference on Image Processing*, *Proc. ICIP*, 2505–2508 (2011). [doi:10.1109/ICIP.2011.6116171].

[60] P. V. Vu and D. M. Chandler, "Video quality assessment based on motion dissimilarity," in *Seventh International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, *Proc. VPQM* (2013).

[61] D. Chandler and S. Hemami, "VSNR: A wavelet-based visual signal-to-noise ratio for natural images," *IEEE Transactions on Image Processing* **16**, 2284–2298 (2007). [doi:10.1109/TIP.2007.901820].

[62] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *7th International Joint Conference on Artificial Intelligence*, *Proc. IJCAI* **2**, 674–679, Vancouver, B. C., Canada (1981).

[63] J. G. Robson, "Spatial and temporal contrast-sensitivity functions of the visual system," *J. Opt. Soc. Am.* **56**, 1141–1142 (1966). [doi:10.1364/JOSA.56.001141].

[64] Image & Video Processing Laboratory, The Chinese University of Hong Kong, "IVP subjective quality video database," (2012). [Available: http://ivp.ee.cuhk.edu.hk/research/database/subjective/index.shtml].

[65] EBU Technical review, "Subjective quality of internet video codec phase II evaluations using SAMVIQ," tech. rep., European Broadcasting Union (2005).

[66] I. f. D. P. Technische Universitt Mnchen, "(2011) TUM multi format test set. [online].." Available: www.ldv.ei.tum.de/videolab.

[67] "Xiph.org test media. available online."

[68] "High efficient video coding (HEVC) project."

[69] "HEVC test sequences [online]." Available: ftp://ftp.tnt.uni-hannover.de/testsequences/.

[70] "FFMPEG tool." Available: http://www.ffmpeg.org.

[71] "Joint video coding team.."

[72] J. W. Woods, *Multidimensional Signal, Image, and Video Processing and Coding*, Academic Press, second ed. (2011).

[73] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *Circuits and Systems for Video Technology, IEEE Transactions on* **22**, 1649–1668 (2012).

[74] O. S. U. Laboratory of Computational Perception & Image Quality, "CSIQ video database," (2013). Available: http://vision.okstate.edu/csiq/.

[75] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on Image Processing* **15**, 3440–3451 (2006). [doi:10.1109/TIP.2006.881959].

[76] E. C. Larson and D. M. Chandler, "Performance-analysis-based acceleration of image quality assessment," in *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, (2012).

[77] T. D. Phan, S. K. Shah, D. M. Chandler, and S. Sohoni, "Microarchitectural analysis of image quality assessment algorithms," *Journal of Electronic Imaging* **23**(1), 013030 (2014).

[78] W. Yu, R. Qiu, and J. E. Fritts, "Advantages of motion-JPEG2000 in video processing," in *Proc. SPIE*, **4671**, 635–645 (2002).

[79] M. Ouaret, F. Dufaux, and T. Ebrahimi, "On comparing JPEG2000 and intraframe AVC," in *Proceedings of SPIE, Applications of Digital Image Processing*, **6312**, 63120U (2006).

[80] P. Topiwala, T. Tran, and W. Dai, "Performance comparison of JPEG2000 and H.264/AVC high profile intra-frame coding on HD video sequences," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **6312** (2006).

[81] A. Kouadio, "JPEG2000: A potential solution for archiving HDTV content," *SMPTE Mot. Imag. J* **118**, 33–40 (2009).

[82] W. H. Dale Stolitzka and S. Foessel3, "New JPEG2000 profiles for broadcast contribution," *SMPTE Mot. Imag. J* **120**, 36–44 (2011).

[83] A. Bilgin and M. W. Marcellin, "JPEG2000 for digital cinema," in *International Symposium on Circuits and Systems (ISCAS)*, (2006).

[84] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "A no-reference perceptual blur metric," in *IEEE International Conference on Image Processing*, *Proc. ICIP* **3**, III–57–III–60 (2002).

[85] E. Ong, W. Lin, Z. Lu, S. Yao, X. Yang, and L. Jiang, "No-reference JPEG-2000 image quality metric," in *International Conference on Multimedia and Expo*, *Proc. ICME* **1**, I–545–I–548 (2003). [doi:10.1109/ICME.2003.1220975].

[86] J. Zhang, S. H. Ong, and T. M. Le, "Kurtosis-based no-reference quality assessment of JPEG2000 images," *Signal Processing: Image Communication* **26**(1), 13–23 (2011). [doi:10.1016/j.image.2010.11.003].

[87] P. Vu and D. Chandler, "A fast wavelet-based algorithm for global and local image sharpness estimation," *IEEE Signal Processing Letters* **19**, 423–426 (2012). [doi:10.1109/LSP.2012.2199980].

[88] C. T. Vu, T. D. Phan, and D. M. Chandler, "S_3: A spectral and spatial measure of local perceived sharpness in natural images," *IEEE Transactions on Image Processing* **21**, 934–945 (2012).

[89] H. Tong, M. Li, H. J. Zhang, and C. Zhang, "No-reference quality assessment for JPEG2000 compressed images," in *IEEE International Conference on Image Processing, Proc. ICIP* **5**, 3539–3542 (2004).

[90] R. Barland and A. Saadane, "Blind quality metric using a perceptual importance map for JPEG2000-compressed images," in *IEEE International Conference on Image Processing, Proc. ICIP*, 2941–2944 (2006).

[91] L. Liang, S. Wang, J. Chen, S. Ma, D. Zhao, and W. Gao, "No-reference perceptual image quality metric using gradient profiles for JPEG2000," *Signal Processing: Image Communication. Special Issue on Image and Video Quality Assessment* **25**(7), 502–516 (2010). [doi:10.1016/j.image.2010.01.007].

[92] P. V. Vu and D. M. Chandler, "A no-reference quality assessment algorithm for jpeg2000-compressed images based on local sharpness," *Proc. SPIE* **8653**, 865302–865302–8 (2013).

[93] H. Sheikh, A. Bovik, and L. Cormack, "No-reference quality assessment using natural scene statistics: JPEG2000," *IEEE Transactions on Image Processing* **14**, 1918–1927 (2005).

[94] Z. M. P. Sazzad, Y. Kawayoke, and Y. Horita, "No reference image quality assessment for JPEG2000 based on spatial features," *Signal Processing: Image Communication* **23**(4), 257–268 (2008). [doi:10.1016/j.image.2008.03.005].

[95] J. Zhang and T. M. Le, "A new no-reference quality metric for JPEG2000 images," *IEEE Transactions on Consumer Electronics* **56**, 743–750 (2010). [doi:10.1109/TCE.2010.5505996].

[96] K. NISHIKAWA and H. KIYA, "No-reference image quality estimation for motion JPEG2000 enabling precise estimation of PSNR values," in *APSIPA Annual Summit and Conference*, 294–297 (2010).

[97] P. V. Vu and D. M. Chandler, "Vis3: an algorithm for video quality assessment via analysis of spatial and spatiotemporal slices," *Journal of Electronic Imaging* **23**(1), 013016 (2014).

[98] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics* **45**, 485–560 (1992). [doi:10.1002/cpa.3160450502].

[99] S. H. Oguz, Y.-H. Hu, and T. Q. Nguyen, "Image coding ringing artifact reduction using morphological post-filtering," in *IEEE Second Workshop on Multimedia Signal Processing*, *Proc. MMSP.1998*, 628 – 633 (1998).

[100] H. Liu, N. Klomp, and I. Heynderickx, "A no-reference metric for perceived ringing artifacts in images," *IEEE Transactions on Circuits and Systems for Video Technology* **20**(4), 529 – 539 (2010).

[101] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6), 679 – 698 (1986).

[102] U. Engelke, V. X. Nguyen, and H. Zepernick, "Regional attention to structural degradations for perceptual image quality metric design," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 869–872 (2008). [doi:10.1109/ICASSP.2008.4517748].

[103] J. You, A. Perkis, M. M. Hannuksela, and M. Gabbouj, "Perceptual quality assessment based on visual attention analysis," in *Proceedings of the 17th ACM International Conference on Multimedia*, *MM '09*, 561–564, ACM, (New York, NY, USA) (2009). [doi:10.1145/1631272.1631356].

[104] O. L. Meur, A. Ninassi, P. L. Callet, and D. Barba, "Overt visual attention for free-viewing and quality assessment tasks: Impact of the regions of interest on a video quality metric," *Signal Processing: Image Communication* **25**(7), 547–558 (2010).

[105] Y. Zhang and D. M. Chandler, "No-reference image quality assessment based on log-derivative statistics of natural scenes," *Journal of Electronic Imaging* **22**(4), 043025 (2013).

[106] Y. Zhang, A. K. Moorthy, D. M. Chandler, and A. C. Bovik, "C-diivine: No-reference image quality assessment based on local magnitude and phase statistics of natural scenes," *Signal Processing: Image Communication* **29**(7), 725 – 747 (2014).

VITA

Phong Van Vu

Candidate for the Degree of

Doctor of Philosophy

Dissertation: NEW APPROACHES AND A SUBJECTIVE DATABASE FOR VIDEO QUALITY ASSESSMENT

Major Field: Electrical and Computer Engineering

Biographical:

- Education:

  Completed the requirements for the Doctor of Philosophy in Electrical and Computer Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2014.

  Completed the requirements for the Master of Science in Electrical and Computer Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2013.

  Completed the requirements for the Bachelor of Science in Electronics and Telecommunications at Posts and Telecommunications Institute of Technology, Hanoi, Vietnam in 2004.

- Professional experience

  Research and Teaching Assistant, *Oklahoma State University*, 01/2010 - 07/2014

  Business Data Analyst, *Viettel Telecom*, 09/2008 - 12/2009

  Technical Engineer, *Beijing Telestone Telecommunications*, 08/2007 - 08/2009

  Service Manager, *Bang and Olufsen Vietnam*, 08/2006 - 07/2007

  Transmission Engineer, *TST Jsc. Co.*, 03/2005 - 07/2005