Journal of
Electronic Imaging

SPIEDigitalLibrary.org/jei

ViS$_3$: an algorithm for video quality assessment via analysis of spatial and spatiotemporal slices

Phong V. Vu
Damon M. Chandler

SPIE

IS&T
imaging.org

Journal of Electronic Imaging 23(1), 013016 (Jan–Feb 2014)

ViS$_3$: an algorithm for video quality assessment via analysis of spatial and spatiotemporal slices

Phong V. Vu* **and Damon M. Chandler**
Oklahoma State University, School of Electrical and Computer Engineering, 202 Engineering South, Stillwater, Oklahoma 74078

Abstract. Algorithms for video quality assessment (VQA) aim to estimate the qualities of videos in a manner that agrees with human judgments of quality. Modern VQA algorithms often estimate video quality by comparing localized space-time regions or groups of frames from the reference and distorted videos, using comparisons based on visual features, statistics, and/or perceptual models. We present a VQA algorithm that estimates quality via separate estimates of perceived degradation due to (1) spatial distortion and (2) joint spatial and temporal distortion. The first stage of the algorithm estimates perceived quality degradation due to spatial distortion; this stage operates by adaptively applying to groups of spatial video frames the two strategies from the most apparent distortion algorithm with an extension to account for temporal masking. The second stage of the algorithm estimates perceived quality degradation due to joint spatial and temporal distortion; this stage operates by measuring the dissimilarity between the reference and distorted videos represented in terms of two-dimensional spatiotemporal slices. Finally, the estimates obtained from the two stages are combined to yield an overall estimate of perceived quality degradation. Testing on various video-quality databases demonstrates that our algorithm performs well in predicting video quality and is competitive with current state-of-the-art VQA algorithms. *© 2014 SPIE and IS&T* [DOI: 10.1117/1.JEI.23.1.013016]

Keywords: video quality assessment; spatiotemporal dissimilarity; spatiotemporal analysis; most apparent distortion; image quality assessment.

Paper 13358 received Jun. 30, 2013; revised manuscript received Dec. 21, 2013; accepted for publication Jan. 8, 2014; published online Feb. 4, 2014.

1 Introduction

The ability to quantify the visual quality of an image or video is a crucial step for any system that processes digital media. Algorithms for image quality assessment (IQA) and video quality assessment (VQA) aim to estimate the quality of a distorted image/video in a manner that agrees with the quality judgments reported by human observers. Over the last few decades, numerous IQA 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. A key advantage of this approach is that it is very intuitive, easily implemented, and computationally efficient. However, such a frame-by-frame IQA approach often fails to correlate with the subjective ratings of quality.[1,2]

1.1 *General Approaches to VQA*

One reason frame-by-frame IQA performs less well for VQA is because it ignores temporal information, which is important for video quality due to temporal effects, such as temporal masking and motion perception.[3,4] 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.[5–8] For example, in Refs. 6 and 7, motion-weighting and temporal derivatives have been used to extend structural similarity (SSIM)[9] and visual information fidelity (VIF)[10] 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. 11 and 12, 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 Ref. 4, video quality is estimated via measures of spatial quality, temporal quality, and spatiotemporal quality for groups of video frames via a three-dimensional (3-D) Gabor filter-bank; the spatial and temporal components are combined into an overall estimate of quality. In Ref. 13, spatial edge features and motion characteristics in localized space-time regions are used to estimate quality.

Furthermore, it is known that the subjective assessment of video quality is time-varying,[14] and this temporal variation can strongly influence the overall quality ratings.[15,16] Models of VQA that consider these effects have been proposed in Refs. 16 to 19. For example, in Ref. 19, 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. 16, 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.

1.2 *Different Approach for VQA: Analysis of Spatiotemporal Slices*

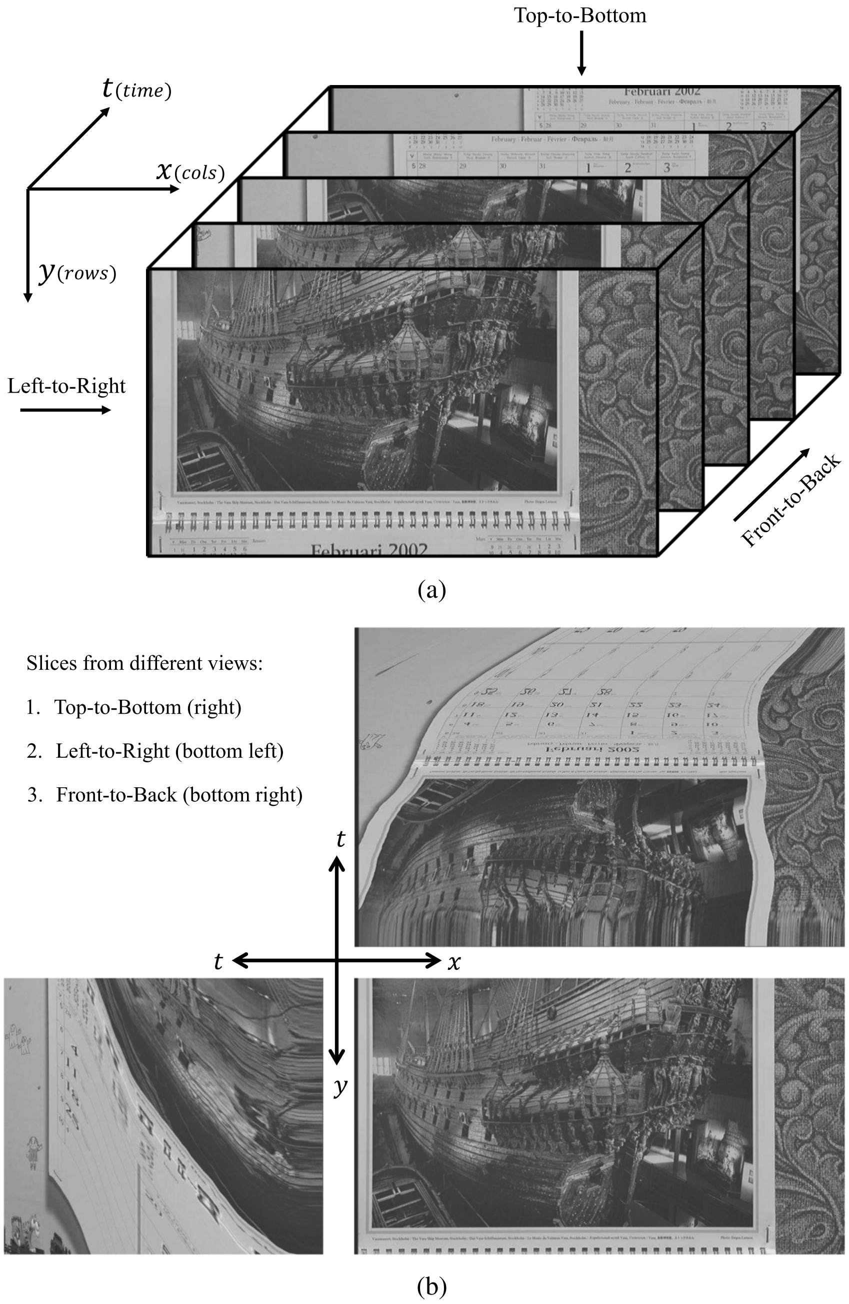
Traditional analyses of temporal variation in VQA tend to formulate methods to compute spatial distortion of a standalone frame,[5,7] of local space-time regions,[12,13] or of groups of adjacent frames[4,19] and then measure the changes

*Address all correspondence to: Phong V. Vu, E-mail: phong.vu@okstate.edu

0091-3286/2014/$25.00 © 2014 SPIE and IS&T

of spatial distortion over time. An alternative approach, which is the technique we adopt in this paper, is to use spatiotemporal slices (as illustrated in Fig. 1), which allows one to analyze longer temporal variations.[20,21] In the context of general motion analysis, Ngo et al.[21] 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, in this paper, we present an algorithm that estimates quality based on the differences between the spatiotemporal slices of the reference and distorted videos.

As shown in Fig. 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 Fig. 1(b), if the cuboid is sliced vertically (left-to-right or right-to-left), then the extracted slices represent time along



(a)

(b)

Fig. 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 yields spatiotemporal slice images (STS images). Examples of three different slice types are presented in part (b) of the figure.

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 2 shows examples of STS images from some typical videos. At one extreme, if the video contains no changes across time (e.g., no motion, as in a static video), then the STS images will contain only horizontal lines [see Fig. 2(a)] or only vertical lines [see Fig. 2(b)]. In both Figs. 2(a) and 2(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 Figs. 2(c) and 2(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 [Figs. 2(e) and 2(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,[22] in energy models of motion perception,[23] and in video motion analysis.[20,21] 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, Fig. 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 4 shows sample STS images from two distorted videos of the LIVE video database[24] 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 peak SNR (PSNR)[25] and the most apparent distortion (MAD) algorithm[26] 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 Fig. 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 responses of the human visual system (HVS) to temporal changes of spatial distortion. Consequently, PSNR$_{sts}$ and MAD$_{sts}$ can yield predictions that correlate poorly with mean opinion score (MOS)/ difference mean opinion score (DMOS). Thus, we propose an alternative method of quantifying degradation of the STS images via a measure of correlation and a model of motion perception.

1.3 *Proposal and Contributions*

In this paper, we propose a VQA algorithm that estimates video quality by measuring spatial distortion and

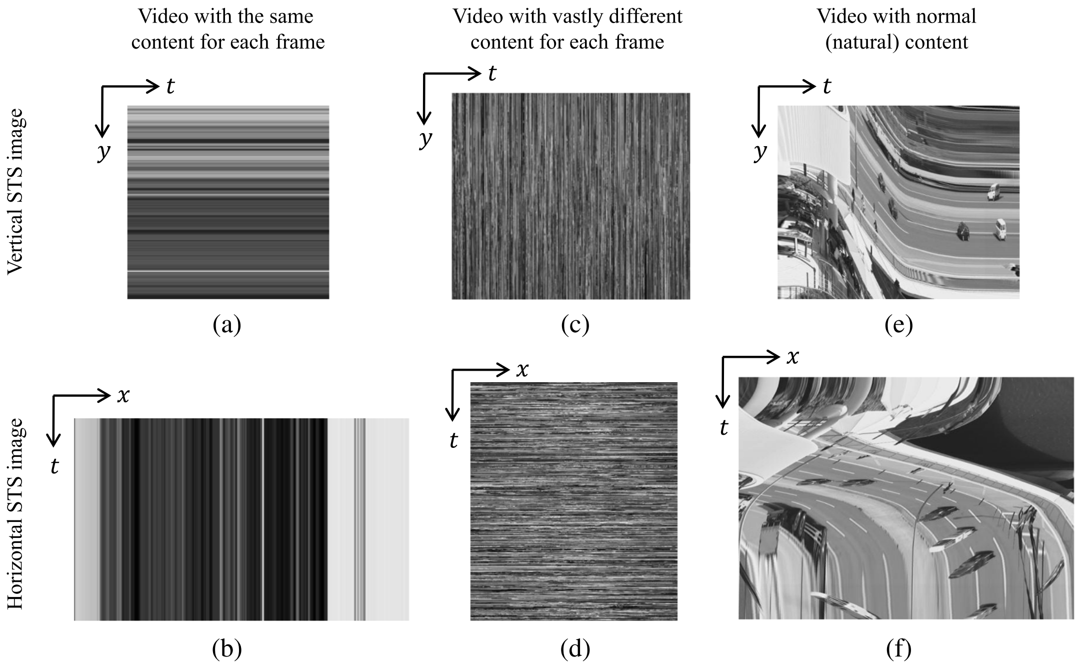

Video with the same content for each frame
Video with vastly different content for each frame
Video with normal (natural) content
Vertical STS image
Horizontal STS image
t
y
x
t
(a)
(c)
(e)
(b)
(d)
(f)

Fig. 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) to (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.

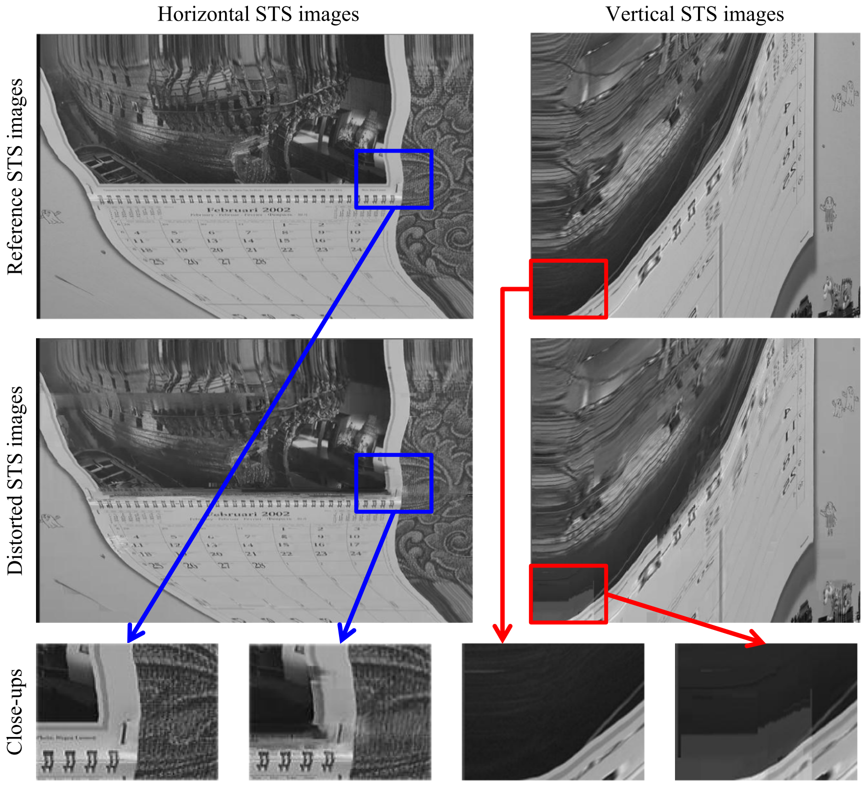


Fig. 3 Demonstrative STS images extracted from the reference and distorted videos. The close-ups show some dissimilar regions between the STS images.

spatiotemporal dissimilarity separately. To estimate perceived video quality degradation due to spatial distortion, both the detection-based strategy and the appearance-based strategy of our MAD algorithm 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.[5,18] To estimate spatiotemporal dissimilarity, we extend the models of Watson–Ahumada[27] and Adelson–Bergen,[23] which have been used to measure energy of motion in videos, to the STS images and measure the local variance of spatiotemporal neural responses. The spatiotemporal response is measured by filtering the STS image via one one-dimensional (1-D) spatial filter and one 1-D temporal filter.[23,27] The overall estimate of perceived video quality degradation is given by a geometric mean of the spatial distortion and spatiotemporal dissimilarity values.

We have named our algorithm ViS$_3$ according to its two main stages: the first stage estimates video quality degradation based on spatial distortion (ViS$_1$), and the second stage estimates video quality degradation based on the dissimilarity between spatiotemporal slice 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 improved and extended version of our previous VQA algorithms presented in Refs. 28 and 29. We demonstrate the performance of this algorithm on various video-quality databases and compare to some recent VQA algorithms. We also analyze the performance of ViS$_3$ on different types of distortion by measuring its performance on each subset of videos.

The major contributions of this paper 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.

This paper is organized as follows. In Sec. 2, we provide a brief review of current VQA algorithms. In Sec. 3, we describe details of the ViS$_3$ algorithm. In Sec. 4, we present and compare the results of applying ViS$_3$ to different video databases. General conclusions are presented in Sec. 5.

2 Brief Review of Existing VQA Algorithms

In this section, we provide a brief review of current VQA algorithms. Following the classification specified in Ref. 30, current VQA methods can roughly be divided into four classes: (1) those that employ IQA on a frame-by-frame basis, (2) those that estimate quality based on differences between visual features of the reference and distorted videos, (3) those that estimate quality based on statsitical differences

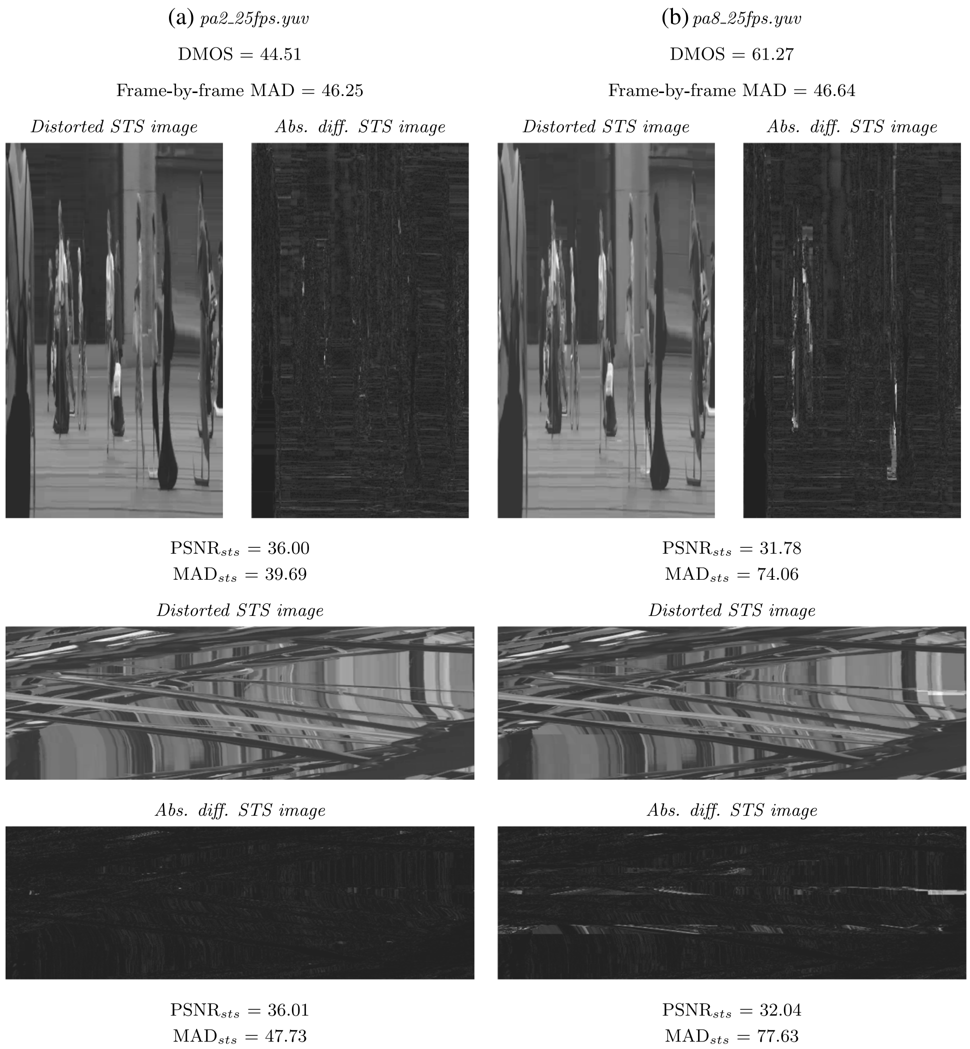


Fig. 4 Sample STS images and their absolute difference STS images (relative to the STS images of the reference videos) extracted from videos (a) *pa2_25fps.yuv* and (b) *pa8_25fps.yuv* for vertical STS images (upper row) and for horizontal STS images (lower row). The videos are from the LIVE video database.[24] The values obtained by applying frame-by-frame most apparent distortion (MAD) on normal (front-to-back) frames are shown for comparison. The PSNR$_{\text{sts}}$ and MAD$_{\text{sts}}$ values, which are computed from the STS images, show promise in estimating video quality. However, neither peak SNR nor MAD account for human visual system responses to temporal changes of spatial distortion, and thus we propose an alternative method of quantifying degradation of the STS images.

between the reference and distorted videos, and (4) those that attempt to model one or more aspects of the HVS.

2.1 *Frame-by-Frame IQA*

As stated in Sec. 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. It is common to find these frame-by-frame IQA algorithms used as a baseline for comparison,[24,31] 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 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.[11,12,34–40]

One of the earliest approaches to feature-based VQA was proposed by Pessoa et al.[34] Their VQA algorithm employs segmentation along with segment-type-specific error measures. Frames of the reference and distorted videos are first segmented into smooth, edge, and texture segments. 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. The overall estimate of quality is computed via a weighted linear combination of logistic-normalized versions of these error measures, using segment-category-specific weights, collapsed across all segments and all frames.

One of the most popular feature-based VQA algorithms, called the video quality metric (VQM), was developed by Pinson and Wolf.[11,12] The VQM algorithm employs quality features that capture spatial, temporal, and color-based differences between the reference and distorted videos. The VQM algorithm consists of 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 subregions of space and time, and then extracts a set of quality features for each subregion. 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 that operates based on the distortion of edges in both space and time. Okamoto et al. employ three general features: (1) blurring in edge regions, which is quantified by using the average edge energy difference described in ANSI T1.801.03; (2) blocking artifacts, which are quantified based on the ratio of horizontal and vertical edge distortions to other edge distortions; and (3) the average local motion distortion, which 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 propose a VQA algorithm that operates under the assumption that visual sensitivity is greatest near edges and block boundaries. Accordingly, their algorithm applies both an edge-detection stage and a block-boundary detection stage to 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 linear sum of these three features 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, therefore, weight the degradations in these areas based on their durations. TetraVQM 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, and this information is used to supplement VQM for estimating the overall quality. In Ref. 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 have also been combined via machine-learning for improved VQA. In Ref. 8, 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 the spatial and temporal factors as well as their interactions on the overall video quality. Spatial quality factors are estimated by a singular value decomposition (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 that estimate quality based on differences in statistical features of the reference and distorted videos.[5–7]

In Ref. 5, Wang et al. proposed the video structural similarity (VSSIM) index. VSSIM computes various SSIM[9] indices at three different levels: the local region level, the frame level, and the video sequence level. In the local region level, the SSIM index of each region is computed for the luminance and chrominance components, with greater weight 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 and Li.[6] There, they augmented SSIM[9] 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. 7, Sheikh and Bovik augmented the VIF IQA algorithm[10] for use in VQA. VIF estimates quality based on the amount of 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, V-VIF quantifies loss in motion information by measuring deviations in the spatiotemporal derivatives of the videos, the latter of 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 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 of 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 derive 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 spacetime texture representation, the reference and distorted videos are filtered via a bank of 3-D 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 temporal distortion measure is then combined with a model of visual saliency and multiscale SSIM[43] (averaged across frames) to estimate quality.

2.4 *Algorithms Based on Models of Human Vision*

Another widely adopted approach to VQA is to estimate video quality via the use of various models of the HVS.[4,44–55]

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 nonuniform backgrounds, the model is supplemented with a masking function based on the spatial and temporal activities of the video.

The digital video quality 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 discrete cosine transform (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 a model of the temporal contrast sensitivity function. JNDs 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 approaches to VQA have employed various subband decompositions to model the spatiotemporal response properties of populations of visual neurons, which are assumed to underlie the multichannel nature of the HVS.[4,45–47,53,55] These algorithms generally compute simulated neural responses to the reference and distorted videos and then estimate quality based on the extent to which these responses differ.

The moving picture quality metric algorithm, proposed by Basso et al.,[45] employs a spatiotemporal multichannel HVS model by using 17 spatial Gabor filters and two temporal filters on the luminance component. After contrast sensitivity and masking adjustments, distortion is measured within each subband and pooled to yield the quality estimate. The color MPQM algorithm, proposed by Lambrecht,[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 normalization video fidelity metric algorithm, proposed by Lindh and Lambrecht,[47] implements a visibility prediction model based on the Teo–Heeger gain-control model.[56] Instead of using Gabor filters, the multichannel decomposition is performed by using the 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 videos.

Masry et al.[53] developed a VQA algorith that employs a multichannel decomposition and a masking model implemented via a separable wavelet transform. A training step was performed on a set of videos and associated subjective quality scores to obtain the masking parameters. Later in Ref. 55, Li et al. utilized this algorithm as part of a VQA algorithm that 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.[24]

Seshadrinathan and Bovik[4] proposed the motion-based video integrity evaluation (MOVIE) algorithm that estimates spatial quality, temporal quality, and spatiotemporal quality via a 3-D subband decomposition. MOVIE decomposes both the reference and distorted videos by using a 3-D Gabor filter-bank with 105 spatiotemporal subbands. The spatial component of MOVIE uses the outputs of the spatiotemporal Gabor filters and a model of contrast masking to capture spatial distortion. The temporal component of MOVIE 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 *Summary*

In summary, although previous VQA algorithms have analyzed the effects of spatial and temporal interactions on video quality, none have 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 Péchard et al.,[57] where spatiotemporal tubes rather than slices were used for VQA. Their algorithm employs a segmentation to create spatiotemporal tubes, which are coherent in terms of motion and spatial activity. Similar to our STS images, the spatiotemporal tubes permit analysis of spatiotemporal information on a long time scale, and Pechard et al. demonstrated the superiority of their approach compared to other VQA algorithms on videos containing H.264 artifacts.

In the following section, we 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.

3 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 the perceived video quality degradation of the distorted video $\hat{\mathbf{I}}$ in comparison to the reference video $\mathbf{I}$. Figure 5 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 the average perceived distortion that occurs spatially in every group of frames (GOF). A motion-weighting scheme is used to model the effect of motion on the visibility of distortion. These per-group spatial distortion values are then combined into a single scalar, ViS$_1$, which denotes an estimate of overall perceived video quality degradation due to spatial distortion.

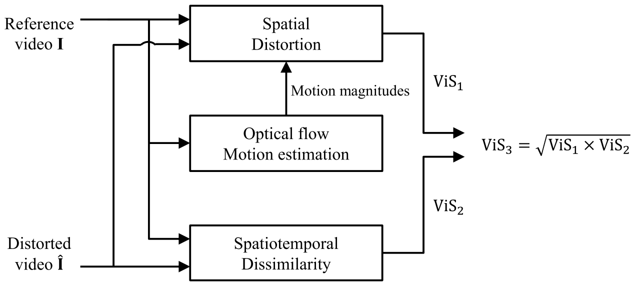

Reference video **I**
Distorted video **Î**
Spatial Distortion
Motion magnitudes
Optical flow Motion estimation
Spatiotemporal Dissimilarity
ViS$_1$
ViS$_2$
ViS$_3$ = $\sqrt{\text{ViS}_1 \times \text{ViS}_2}$

Fig. 5 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.

- Spatiotemporal dissimilarity: The spatiotemporal dissimilarity 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 responses of modeled visual neurons. These per-STS-image spatiotemporal dissimilarity values are then combined into a single scalar, ViS$_2$, which denotes 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 perceived quality degradation of the video. The following subsections provide details of each stage of the algorithm.

3.1 *Spatial Distortion*

In the spatial distortion stage, we employ and extend our MAD algorithm,[26] which was designed for 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 by using 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 the block-based log-Gabor statistics of the reference and distorted images.

The MAD index of the distorted image is computed via a geometric weighted mean.

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

$$\text{MAD} = (d_{\text{detect}})^{\alpha} \times (d_{\text{appear}})^{1-\alpha}, \tag{2}$$

where the weight $\alpha \in [0, 1]$ serves to adaptively combine the two strategies (d_{detect} and d_{appear}) based on the overall level of distortion. As described in Ref. 26, for high-quality images, MAD should obtain its value mostly from d_{detect}, whereas for low-quality images, 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. 26, the value of d_{detect} served as this initial estimate, and thus, α is a function of d_{detect}. The two free parameters $\beta_1 = 0.467$ and $\beta_2 = 0.130$ were obtained after training on the A57 image database;[58] see Ref. 26 for a complete description of the MAD algorithm.

To extend MAD for use with video, we take the Y components of the videos and perform the following steps (shown in Fig. 6) on each group of N consecutive frames:

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. Estimate the magnitude of the motion vectors in each frame of the reference video by using the Lucas–Kanade optical flow method.[59] 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 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.

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 the details of each step.

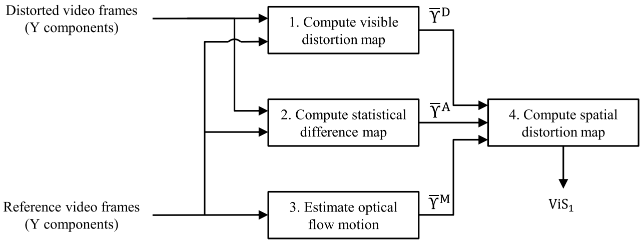

Distorted video frames (Y components)
Reference video frames (Y components)
1. Compute visible distortion map
2. Compute statistical difference map
3. Estimate optical flow motion
4. Compute spatial distortion map
$\overline{\Upsilon}^D$
$\overline{\Upsilon}^A$
$\overline{\Upsilon}^M$
ViS$_1$

Fig. 6 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 group of frames (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$_1$.

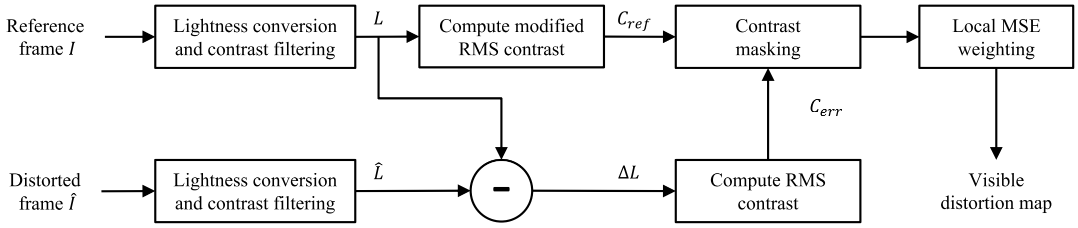

Reference frame I
Lightness conversion and contrast filtering
L
Compute modified RMS contrast
C_{ref}
Contrast masking
Local MSE weighting
C_{err}
Distorted frame $\hat{I}$
Lightness conversion and contrast filtering
$\hat{L}$
-
ΔL
Compute RMS contrast
Visible distortion map

Fig. 7 Block diagram of the detection-based strategy used to compute a visible distortion map. Both the reference and the distorted frames are converted to perceived luminance and filtered by a contrast sensitivity function. 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 mean squared error to yield a visible distortion map.

3.1.1 *Compute visible distortion map*

We apply the detection-based strategy from Ref. 26 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 Fig. 7.

Detection-based strategy. As illustrated in Fig. 7, a preprocessing step is first performed by using the 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. Next, this map is weighted by local mean squared error (MSE) to yield a visible distortion map. The specific steps are given below (see Ref. 26 for additional details).

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

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

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

$$\tilde{L} = \mathbb{F}^{-1}[H(u,v) \times \mathbb{F}[L]], \tag{4}$$

where $\mathbb{F}$ and $\mathbb{F}^{-1}$ denote the discrete fourier transform (DFT) and inverse DFT, respectively; $H(u,v)$ is the DFT-based version of the CSF function defined by Eq. (3) in Ref. 26.

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 16×16 blocks (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

$$C_{\text{ref}}(b) = \tilde{\sigma}_{\text{ref}}(b)/\mu_{\text{ref}}(b), \tag{5}$$

where $\mu_{\text{ref}}(b)$ denotes the mean of block b of $\tilde{L}$, and $\tilde{\sigma}_{\text{ref}}(b)$ denotes the minimum of the standard deviations of the 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 subblocks (to avoid overestimating the contrast around edges), but small enough to yield decent spatial localization (see Appendix A in Ref. 26).

$C_{\text{ref}}(b)$ is a measure of the 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 the distortions in the distorted frame. The error frame ΔL is divided into 16×16 blocks (with 75% overlap between blocks), and then the RMS contrast $C_{\text{err}}(b)$ for each block b is computed via

$$C_{\text{err}}(b) = \begin{cases} \sigma_{\text{err}}(b)/\mu_{\text{ref}}(b) & \text{if } \mu_{\text{ref}}(b) > 0.5 \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $\sigma_{\text{err}}(b)$ denotes the standard deviation of block b of ΔL. A lightness threshold 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 maps are 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 $\{C_{\text{ref}}\}$ and $\{C_{\text{err}}\}$ are used to compute a local distortion visibility map denoted by $\xi(b)$ via

$$\xi(b) = \begin{cases} \ln[C_{\text{err}}(b)] - \ln[C_{\text{ref}}(b)] & \text{if } \ln[C_{\text{err}}(b)] > \ln[C_{\text{ref}}(b)] > -5 \\ \ln[C_{\text{err}}(b)] + 5 & \text{if } \ln[C_{\text{err}}(b)] > -5 \geq \ln[C_{\text{ref}}(b)] \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

The local distortion visibility map ξ is then point-by-point multiplied by the local MSE to determine a visible distortion map denoted by Υ^{D}, where the superscript 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

$$\Upsilon^{\text{D}}(b) = \xi(b) \cdot \text{MSE}(b). \tag{8}$$

Note that in Ref. 26, 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 $d_{\text{detect}} = \sqrt{\sum_b [\Upsilon^{\text{D}}(b)]^2}$, where the summation is over all blocks. In the current paper, we do not collapse Υ^{D}.

Apply to groups of video frames. Let Υ_t^{D} 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 $\{\Upsilon^{\text{D}}_{N(k-1)+1}, \Upsilon^{\text{D}}_{N(k-1)+2}, \dots, \Upsilon^{\text{D}}_{Nk}\}$, where $k \in \{1, 2, \dots, K\}$ is the GOF index and K is the number of GOFs in the video. These maps are combined via a point-by-point average to yield a GOF-based visible distortion map of the k'th GOF, which is denoted by $\bar{\Upsilon}^{\text{D}}_k$.

$$\bar{\Upsilon}^{\text{D}}_k = \frac{1}{N} \sum_{\tau=1}^{N} \Upsilon^{\text{D}}_{N(k-1)+\tau}. \tag{9}$$

3.1.2 *Compute statistical difference map*

As argued in Ref. 26, 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 multiscale log-Gabor filter responses to capture changes in visual appearance. Figure 8 shows a block diagram of the appearance-based strategy used to compute a statistical difference map between the reference and the distorted frame.

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 orientations $o \in \{1, 2, 3, 4\}$), 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 is the same size as 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 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

$$\Upsilon^{\text{A}}(b) = \sum_{s=1}^{5} \sum_{o=1}^{4} w_s [|\sigma^{s,o}(b) - \hat{\sigma}^{s,o}(b)| + 2|\varsigma^{s,o}(b) - \hat{\varsigma}^{s,o}(b)| + |\kappa^{s,o}(b) - \hat{\kappa}^{s,o}(b)|], \tag{10}$$

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. 26 to account for the HVS's preference for coarse scales over fine scales (see Ref. 26 for more details).

Note that in Ref. 26, 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 $d_{\text{appear}} = \sqrt{\sum_b [\Upsilon^{\text{A}}(b)]^2}$, where the summation is over all blocks. In the current paper, we do not collapse Υ^{A}.

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 $\{\Upsilon^{\text{A}}_{N(k-1)+1}, \Upsilon^{\text{A}}_{N(k-1)+2}, \dots, \Upsilon^{\text{A}}_{Nk}\}$, where $k \in \{1, 2, \dots, K\}$ is the GOF index and K is the number of GOFs in the video. These maps are combined via a point-by-point average to yield a GOF-based statistical difference map of the k'th GOF, which is denoted by $\bar{\Upsilon}^{\text{A}}_k$.

$$\bar{\Upsilon}^{\text{A}}_k = \frac{1}{N} \sum_{\tau=1}^{N} \Upsilon^{\text{A}}_{N(k-1)+\tau}. \tag{11}$$

3.1.3 *Optical-flow motion estimation*

Both the detection-based strategy and the appearance-based strategy were designed for still images. They do not account

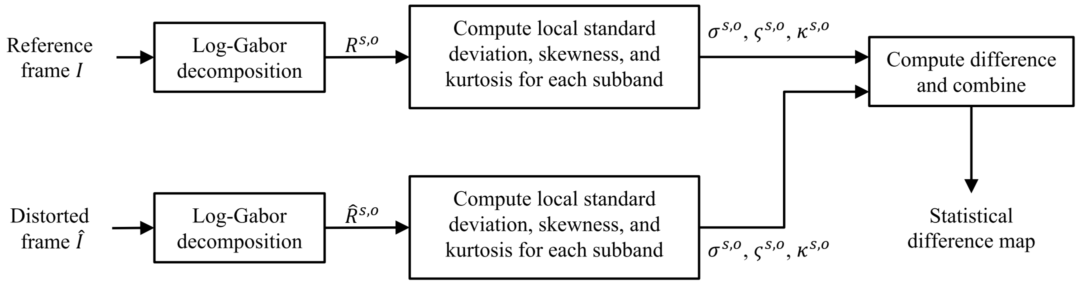

Reference frame I
Log-Gabor decomposition
$R^{s,o}$
Compute local standard deviation, skewness, and kurtosis for each subband
$\sigma^{s,o}$, $\varsigma^{s,o}$, $\kappa^{s,o}$
Compute difference and combine
Distorted frame $\hat{I}$
Log-Gabor decomposition
$\hat{R}^{s,o}$
Compute local standard deviation,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

Fig. 8 Block diagram of the appearance-based strategy used to compute a statistical difference map. The reference and the distorted frames are decomposed into different subbands using a two-dimensional log-Gabor filter-bank. Local standard deviation, skewness, and kurtosis are computed for each subband of both the reference and the distorted frames. The differences of 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.

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.[5] and Barkowsky et al.,[18] 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[59] to the reference video to estimate motion vectors. The Lucas–Kanade method assumes that the displacement of the frame contents between two nearby frames is small and approximately constant within a neighborhood (window) of a point under consideration. Thus, the optical-flow motion vector can be assumed the same within a window centered at that point, and it is computed from solving the optical-flow equations using the least squares criterion.

By using a window of size 8 × 8, for each pair of consecutive frames, we obtain two matrices of motion vectors, M_v and M_h, with respect 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

$$\bar{M}_k = \frac{1}{N}\sum_{\tau=1}^{N} M_{N(k-1)+\tau}. \tag{12}$$

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 an 8 × 8 window in the regular frame. We therefore resize the $\bar{M}_k$ matrix 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 $\bar{\Upsilon}_k^{\mathrm{M}}$, where the superscript M is used to imply that the map is computed from the motion magnitudes.

3.1.4 *Combine maps and compute spatial distortion value*

For each GOF, we have computed the GOF-based visible distortion map $\bar{\Upsilon}^{\mathrm{D}}$, the GOF-based statistical difference map $\bar{\Upsilon}^{\mathrm{A}}$, and the GOF-based motion magnitude map $\bar{\Upsilon}^{\mathrm{M}}$. Now, we extend and apply Eq. (2) 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 $\bar{\Upsilon}_k^{\mathrm{M}}$ to yield the spatial distortion map of the k'th GOF. We denote $\Delta_k(x,y)$ of size $W \times H$, the video frame size, as the spatial distortion map of the k'th GOF. Specifically, the value at (x,y) of the spatial distortion map $\Delta_k(x,y)$ is computed via

$$\hat{\alpha}(x,y) = \frac{1}{1+\beta_1 \times [\bar{\Upsilon}_k^{\mathrm{D}}(x,y)]^{\beta_2}}, \tag{13}$$

$$\Delta_k(x,y) = \frac{[\bar{\Upsilon}_k^{\mathrm{D}}(x,y)]^{\hat{\alpha}(x,y)} \times [\bar{\Upsilon}_k^{\mathrm{A}}(x,y)]^{1-\hat{\alpha}(x,y)}}{\sqrt{1+\bar{\Upsilon}_k^{\mathrm{M}}(x,y)}}. \tag{14}$$

The division by $\bar{\Upsilon}_k^{\mathrm{M}}(x,y)$ 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 $\bar{\Upsilon}_k^{\mathrm{M}}$ 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 $\bar{\Upsilon}_k^{\mathrm{M}}$ 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 $\bar{\Upsilon}_k^{\mathrm{D}}$ and $\bar{\Upsilon}_k^{\mathrm{A}}$.

Figure 9 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.[24] 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 Figs. 9(c) and 9(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).

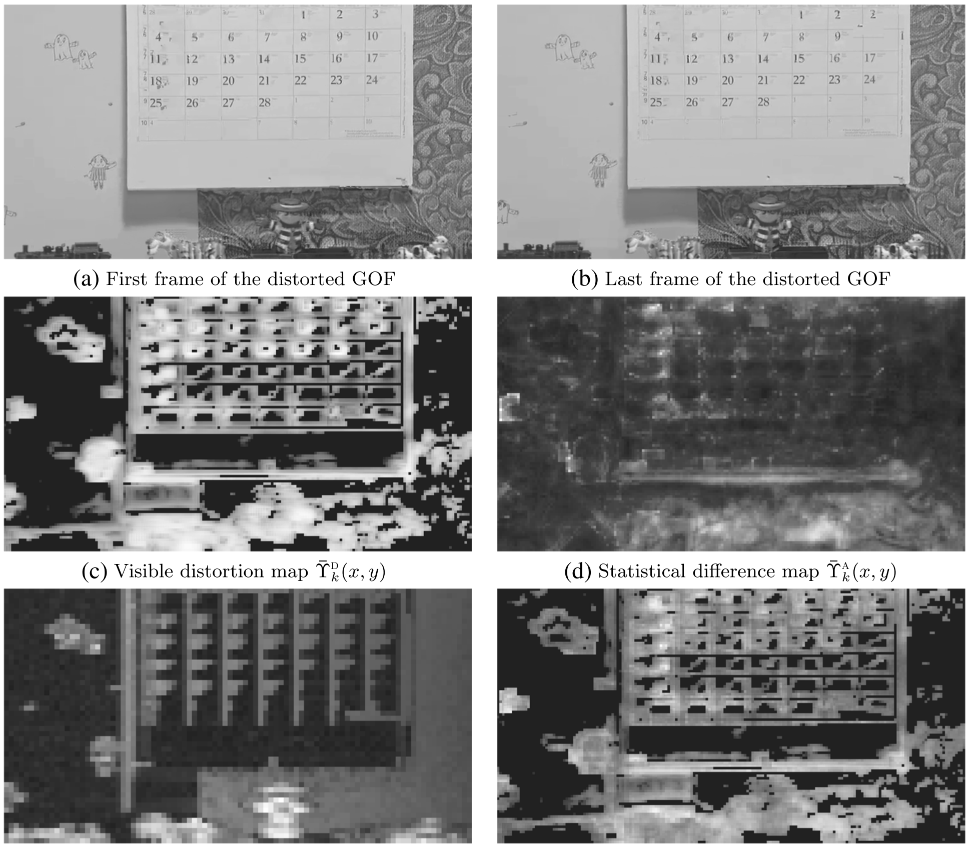
To estimate spatial distortion value of each GOF, we compute the RMS value of the spatial distortion map. The RMS value of the map $\Delta_k(x,y)$ of size $W \times H$ is given by

$$\bar{\Delta}_k^{\mathrm{XY}} = \sqrt{\frac{1}{W \times H}\sum_{x=1}^{W}\sum_{y=1}^{H}[\Delta_k(x,y)]^2}, \tag{15}$$

where the superscript 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 $\bar{\Delta}_k^{\mathrm{XY}}$ via

$$\mathrm{ViS}_1 = \frac{1}{K}\sum_{k=1}^{K}\bar{\Delta}_k^{\mathrm{XY}}. \tag{16}$$

Here, ViS$_1$ is a single scalar that represents the overall perceived quality degradation of the video due to spatial distortion. 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.



(a) First frame of the distorted GOF

(b) Last frame of the distorted GOF

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

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

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

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

Fig. 9 Examples of the first and last frames [(a) and (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.[24] 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.

3.2 *Spatiotemporal Dissimilarity*

In the distorted video, the distortion impacts not only the spatial relationship between neighboring pixels within the current frame, but also the transition between frames, which can be captured via the use of STS images. The difference between the STS images from the reference and distorted videos is referred to as the spatiotemporal dissimilarity in this paper. 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 10 depicts a block diagram of the spatiotemporal dissimilarity stage, which estimates the spatiotemporal dissimilarity between the reference and the distorted video via the following steps:

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 by using a set of spatiotemporal filters. These spatiotemporally filtered images are used

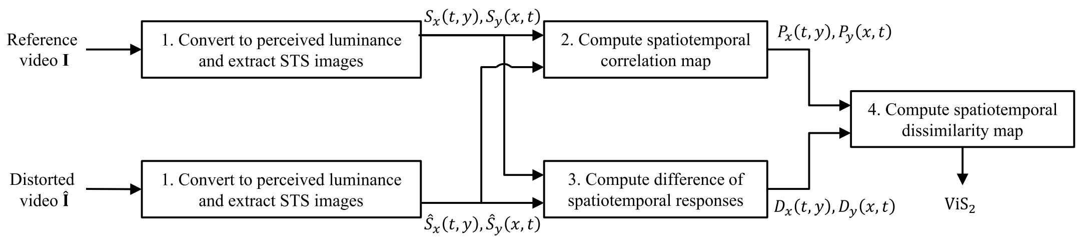

Reference video $\mathbf{I}$
1. Convert to perceived luminance and extract STS images
$S_x(t,y), S_y(x,t)$
2. Compute spatiotemporal correlation map
$P_x(t,y), P_y(x,t)$
4. Compute spatiotemporal dissimilarity map
Distorted video $\hat{\mathbf{I}}$
1. Convert to perceived luminance and extract STS images
3. Compute difference of spatiotemporal responses
$\hat{S}_x(t,y), \hat{S}_y(x,t)$
$D_x(t,y), D_y(x,t)$
ViS$_2$

Fig. 10 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.

to compute a map of spatiotemporal response differences.

4. Combine the above two 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 denotes the overall perceived video spatiotemporal dissimilarity.

The following subsections describe the details of each step.

3.2.1 *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 Eq. (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 Fig. 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.

3.2.2 *Compute spatiotemporal correlation map*

One simple way to measure the spatiotemporal dissimilarity is by using 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 will 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

$$\tilde{\rho}(b) = \begin{cases} 0 & \text{if } \rho(\mathrm{b}) < 0 \\ 1 & \text{if } \rho(\mathrm{b}) > 0.9 \, . \\ \rho(b) & \text{otherwise} \end{cases} \tag{17}$$

As shown in Eq. (17), if the two blocks are highly positively 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. As we demonstrate in the online supplement to this paper,[60] the performance of the algorithm is relatively robust to small changes in this threshold value. 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 Fig. 11(c). The brighter the maps, the higher the spatiotemporal correlation between corresponding regions of the two STS images.

3.2.3 *Compute spatiotemporal response difference map*

The spatiotemporal correlation coefficient computed in Sec. 3.2.2 does not account for the HVS's response to 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. 27. This model applies separate 1-D filters to each dimension of the STS images to measure spatiotemporal responses. In Ref. 23, Adelson and Bergen used these spatiotemporal responses to measure energy of motion in a video. Here, we apply the model to the STS images and measure the differences of spatiotemporal responses to estimate video quality.

Decompose STS images into spatiotemporally filtered images. As stated by Adelson and Bergen in Ref. 23, the spatiotemporal information presented in the STS images can be captured via a set of spatiotemporally oriented filters. As suggested by Watson and Ahumada,[27] these filters can be constructed by two sets of separate 1-D filters (spatial and temporal) with appropriate spatiotemporal characteristics. Following this suggestion, we employ a set of log-Gabor 1-D filters $\{g_s\}$, $s \in \{1,2,3,4,5\}$, as the spatial filters, where the frequency response of each filter is given by

$$G_s(\omega) = \exp\left[-\frac{\left(\ln\left|\frac{\omega}{\omega_s}\right|\right)^2}{2(\ln B_s)^2} \right], \tag{18}$$

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 1-D 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 ($B_s = 0.55$). These filters are almost the same as the log-Gabor filters used in Ref. 26 without the orientation information.

The two temporal filters $\{h_z\}$, $z \in \{1,2\}$, were selected following the Adelson–Bergen model.[23] The impulse response at time instance t of each filter is given by

$$h_z(t) = t^{n_z}\,\exp(-t)\left[\frac{1}{n_z!} - \frac{t^2}{(n_z+2)!}\right], \tag{19}$$

where $n_1 = 6$ and $n_2 = 9$ were chosen to approximate the temporal contrast sensitivity functions reported by Robson,[61] which correspond to the fast and slow motions, respectively.

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

$$R_x^{s,z}(t,y) = [S_x(t,y) *^y g_s] *^t h_z, \tag{20}$$

$$R_y^{s,z}(x,t) = [S_y(x,t) *^x g_s] *^t h_z, \tag{21}$$

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

$$\Delta R_x^{s,z}(t,y) = |R_x^{s,z}(t,y) - \hat{R}_x^{s,z}(t,y)|, \tag{22}$$

$$\Delta R_y^{s,z}(x,t) = |R_y^{s,z}(x,t) - \hat{R}_y^{s,z}(x,t)|. \tag{23}$$

Although the proper technique of estimating video quality based on the response differences 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.

Compute log of response difference map. We compute the local mean and 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 $\Delta R_x^{s,z}(t,y)$. Let $\mu_y^{s,z}(b)$ and $\sigma_y^{s,z}(b)$ denote the local mean and 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

$$\tilde{\sigma}_x^{s,z}(b) = \begin{cases} 0, & \text{if } \mu_{\mathrm{x}}^{\mathrm{s,z}}(\mathrm{b}) < p \\ \sigma_x^{s,z}(b) \times \sqrt{\frac{\mu_x^{s,z}(b)}{p+\mu_x^{s,z}(b)}}, & \text{otherwise} \end{cases}, \tag{24}$$

$$\tilde{\sigma}_y^{s,z}(b) = \begin{cases} 0, & \text{if } \mu_{\mathrm{y}}^{\mathrm{s,z}}(\mathrm{b}) < p \\ \sigma_y^{s,z}(b) \times \sqrt{\frac{\mu_y^{s,z}(b)}{p+\mu_y^{s,z}(b)}}, & \text{otherwise} \end{cases}, \tag{25}$$

where $p = 0.01$ is a threshold value. When the mean value 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 of 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 a 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:

$$D_x(t,y) = \ln\left\{1 + A\sum_{s=1}^{5}\sum_{z=1}^{2} w_s[\tilde{\sigma}_x^{s,z}(t,y)]^2\right\}, \tag{26}$$

$$D_y(x,t) = \ln\left\{1 + A\sum_{s=1}^{5}\sum_{z=1}^{2} w_s[\tilde{\sigma}_y^{s,z}(x,t)]^2\right\}, \tag{27}$$

where the weights $\{w_s\} = \{0.5, 0.75, 1, 5, 6\}$ were chosen following Ref. 26 to account for the HVS's preference for coarse scales over fine scales. The addition of 1 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 Fig. 11(d). The brighter the maps, the greater the difference in spatiotemporal responses between corresponding regions of the two STS images.

3.2.4 *Compute spatiotemporal dissimilarity value*

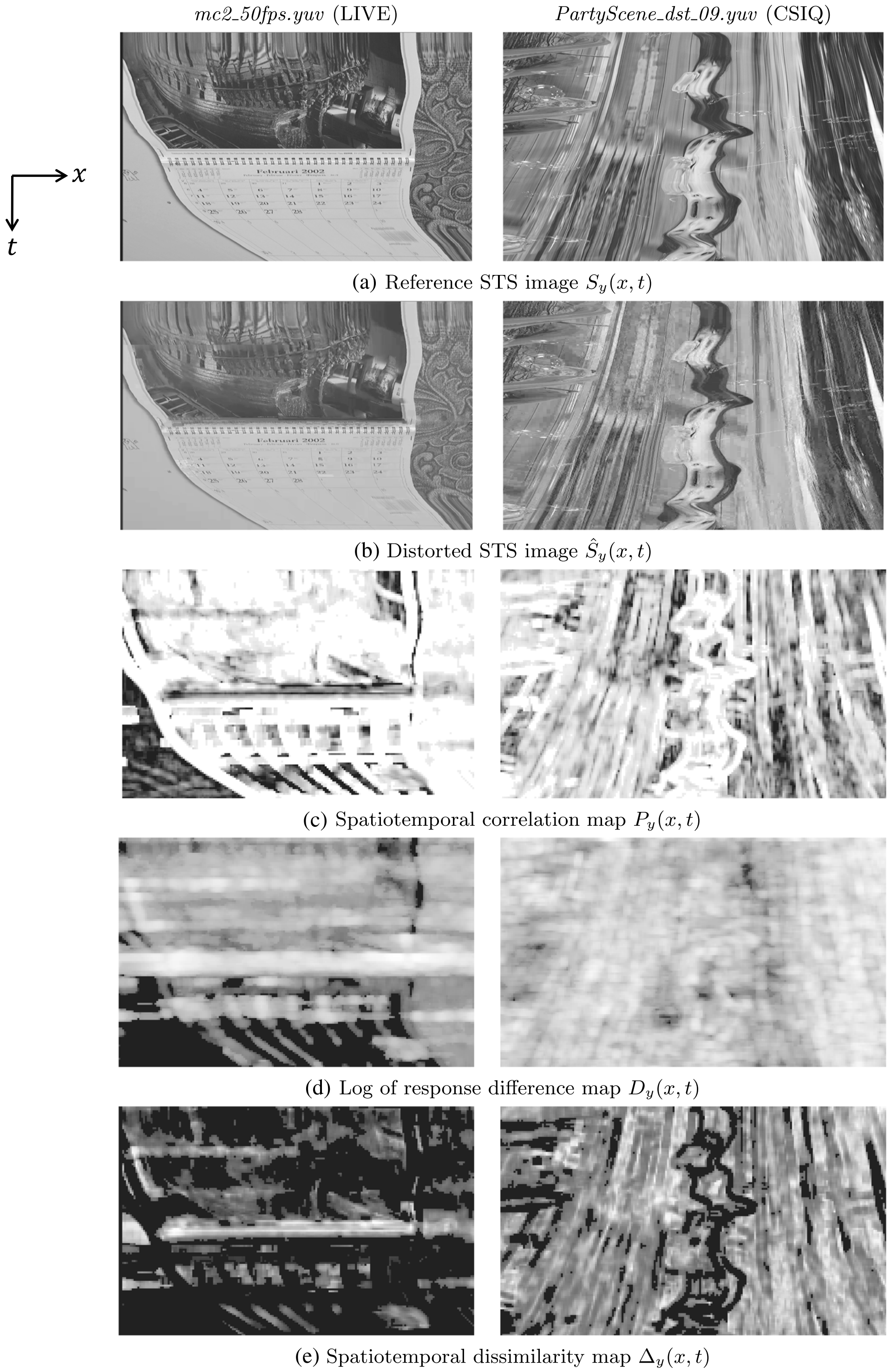
The spatiotemporal correlation map P and the log of response difference map D are combined into a spatiotemporal dissimilarity map via a point-by-point multiplication.

$$\Delta_x(t,y) = D_x(t,y) \cdot \sqrt{1 - P_x(t,y)}, \tag{28}$$

$$\Delta_y(x,t) = D_y(x,t) \cdot \sqrt{1 - P_y(x,t)}. \tag{29}$$

Let $\bar{\Delta}_c^{\mathrm{TY}}$ 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 $\bar{\Delta}_r^{\mathrm{XT}}$ 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:

$$\bar{\Delta}_c^{\mathrm{TY}} = \sqrt{\frac{1}{T \times H}\sum_{t=1}^{T}\sum_{y=1}^{H}[\Delta_c(t,y)]^2}, \tag{30}$$



mc2_50fps.yuv (LIVE) *PartyScene_dst_09.yuv* (CSIQ)

(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 $\Delta_y(x,t)$

Fig. 11 Demonstrative maps for two pairs of STS images $S_y(x,t)$ and $\hat{S}_y(x,t)$ from videos *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.

$$\bar{\Delta}_r^{\mathrm{XT}} = \sqrt{\frac{1}{W \times T} \sum_{x=1}^{W} \sum_{t=1}^{T} [\Delta_r(x,t)]^2}, \tag{31}$$

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 TY and 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 reference and the distorted video is given by

$$\mathrm{ViS}_2 = \sqrt{\frac{1}{W}\sum_{c=1}^{W}[\bar{\Delta}_c^{\mathrm{TY}}]^2 + \frac{1}{H}\sum_{r=1}^{H}[\bar{\Delta}_r^{\mathrm{XT}}]^2}. \tag{32}$$

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 of ViS$_2 = 0$ indicates that the distorted video has perfect quality relative to the reference video.

Figure 11 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. The brighter values in the spatiotemporal dissimilarity maps $\Delta_y(x,t)$ in Fig. 11(e) denote the corresponding spatiotemporal regions of greater dissimilarity.

As observed from the 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 Fig. 11(e) (left). As observed in Fig. 11(e) (left), the dissimilarity map is brighter in the middle 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 Fig. 11(e) (right), where the spatiotemporal dissimilarity map shows brighter surrounding regions compared to the center regions across the temporal dimension.

3.3 *Combine Spatial Distortion and Spatiotemporal Dissimilarity Values*

Finally, the overall estimate of perceived video quality degradation, denoted by ViS$_3$, is computed from the spatial distortion value ViS$_1$ and the spatiotemporal dissimilarity value ViS$_2$. Specifically, ViS$_3$ is computed as a geometric mean of ViS$_1$ and ViS$_2$, which is given by

$$\mathrm{ViS}_3 = \sqrt{\mathrm{ViS}_1 \times \mathrm{ViS}_2}. \tag{33}$$

Here, ViS$_3$ is a single scalar that represents the overall perceived quality degradation of the video. The smaller the ViS$_3$ value, the better the video quality. A value of ViS$_3 = 0$ indicates that the distorted video is equal in quality to the reference video.

Note that the values of ViS$_1$ and ViS$_2$ occupy different ranges. Thus, the use of a geometric mean in Eq. (33) allows us to combine these values without the need for custom weights (which would be required when using an arithmetic mean). Other combinations are also possible, e.g., using a weighted geometric mean with possibly adaptive weights. However, our preliminary attempts to select such weights have not yielded significant improvements (see also Sec. 4.3.4).

4 Results

In this section, we analyze the performance of the ViS$_3$ algorithm in predicting subjective ratings of quality on three publicly available video-quality databases. We also compare the performance of ViS$_3$ with other quality assessment algorithms.

4.1 *Video Quality Databases*

To evaluate the performance of ViS$_3$ and other quality assessment 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);[24]
2. The IVPL video database (four types of distortion);[62]
3. The CSIQ video database (six types of distortion).[63]

4.1.1 *LIVE video database*

The LIVE video database[24] 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, ~10 s 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 bit-streams 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.

4.1.2 *IVPL video database*

The IVPL HD video database[62] 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, ~10 s 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 rescaled the videos to 960 × 544 using FFMPEG software[64] with its default configuration.

4.1.3 *CSIQ video database*

The CSIQ video database[63] developed by the authors at Oklahoma State University 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 s, 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; each distortion type has three different levels. The distortion types consist of four video compression distortion types [Motion JPEG (*MJPEG*), *H.264*, *HEVC*, and wavelet compression using *SNOW* codec[64]] 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 testing protocol[65] with 35 subjects.

4.2 *Algorithms and Performance Measures*

We compared ViS$_3$ with PSNR[25] and recent full-reference video quality assessment algorithms for which code is publicly available, VQM,[12] MOVIE,[4] and TQV,[8] on the three video databases. 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 ViS$_3$, we used a GOF size of $N = 8$.

Before evaluating the performance of each algorithm on each video database, we applied a four-parameter logistic transform to the raw predicted scores, as recommended by video quality experts group (VQEG) in Ref. 31. The four-parameter logistic transform is given by

$$f(x) = \frac{\tau_1 - \tau_2}{1 + \exp\left(-\frac{x - \tau_3}{|\tau_4|}\right)} + \tau_2, \tag{34}$$

where x denotes the raw predicted score and τ_1, τ_2, τ_3, and τ_4 are 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, 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 by two additional criteria: the outlier ratio (OR[5]) and the outlier distance (OD[26]). OR is the ratio of number of false scores predicted by the algorithm to the total number of scores. A false score is defined as the transformed score lying outside the 95% confidence interval of the associated subjective score.[5] In addition, 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 confidence interval.[26]

4.3 *Overall Performance*

The performance of each algorithm on each video database is shown in Table 1 in terms of the five criteria (SROCC, CC, RMSE, OR, and OD). The best-performing algorithm is bolded, and the second best-performing algorithm is italicized and bolded. These data indicate that ViS$_3$ is the best-performing algorithm on all three video databases in terms of all five evaluation criteria. The performances of 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 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).

Table 1 Performances of ViS$_3$ and other algorithms on the three video databases. The best-performing algorithm is bolded and the second best-performing algorithm is italicized. Note that ViS$_3$ is the best-performing algorithm on all three databases.

		Peak SNR (PSNR)	Video quality metric (VQM)	Motion-based video integrity evaluation (MOVIE)	TQV	ViS$_3$	ViS$_1$	ViS$_2$
Spearman rank-order correlation coefficient (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
Root mean square error	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
Outlier ratio	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%
Outlier distance	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

In terms of prediction consistency measured by 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 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.

Observe 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 an 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 versus DMOS on the three databases are shown in Fig. 12. The plots show a highly correlated trend between the logistic-transformed ViS$_3$ values versus DMOS values. For all the three databases, the predictions are homoscedastic; i.e., there are generally no subpopulations 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 Sec. 4.3.3.

4.3.1 *Performance on individual types of distortion*

We measured the 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 2 shows the resulting SROCC and CC values.

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 the

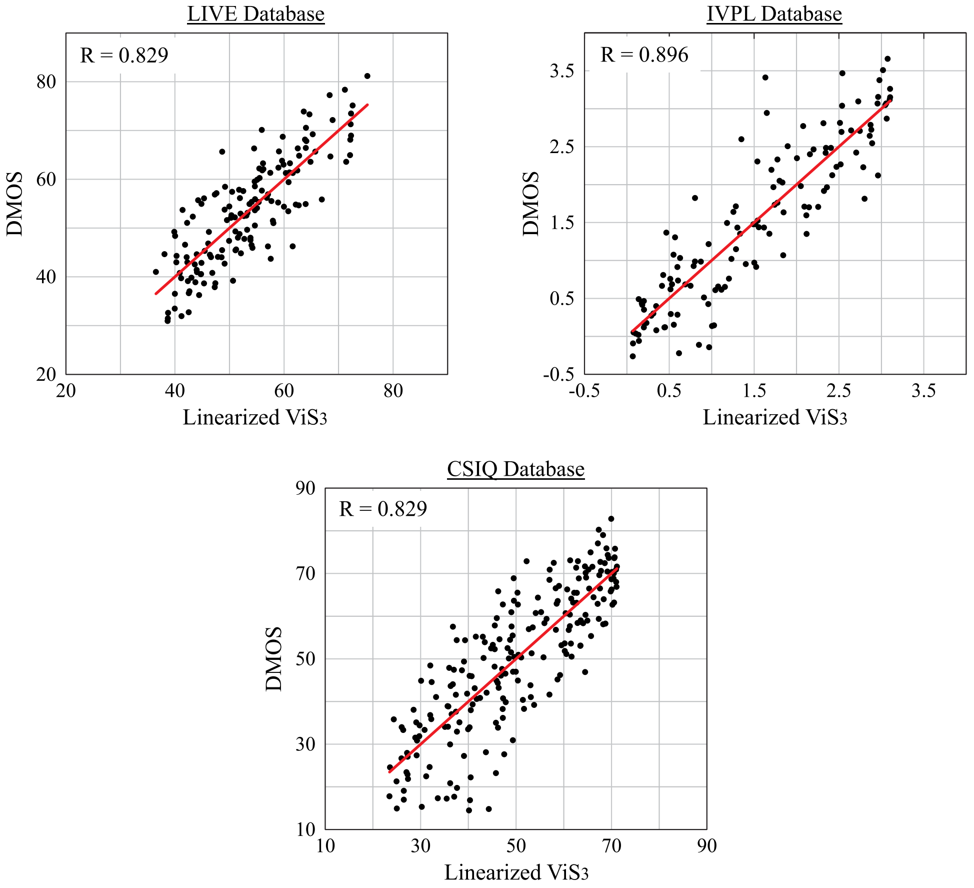


Fig. 12 Scatter-plots of logistic-transformed scores predicted by ViS$_3$ versus subjective scores on the three databases. Notice that all the plots are homoscedastic. The *R* values denote correlation coefficient between the logistic-transformed scores and subjective scores (DMOS).

Table 2 Performances of ViS$_3$ and other quality assessment algorithms measured on different types of distortion on the three video databases. The best-performing algorithm is bolded and the second best-performing algorithm is italicized.

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
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**
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*
			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
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**
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*

MPEG-2 compression distortion and the *IPPL* distortion on both 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* compression distortion; and TQV and MOVIE are the two best-performing algorithms on the *MPEG-2* compression 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 Fig. 13; 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 signficant 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 that 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. PSNR, VQM, and MOVIE are also competitive on both the *DIRAC* and *H.264* distortion.

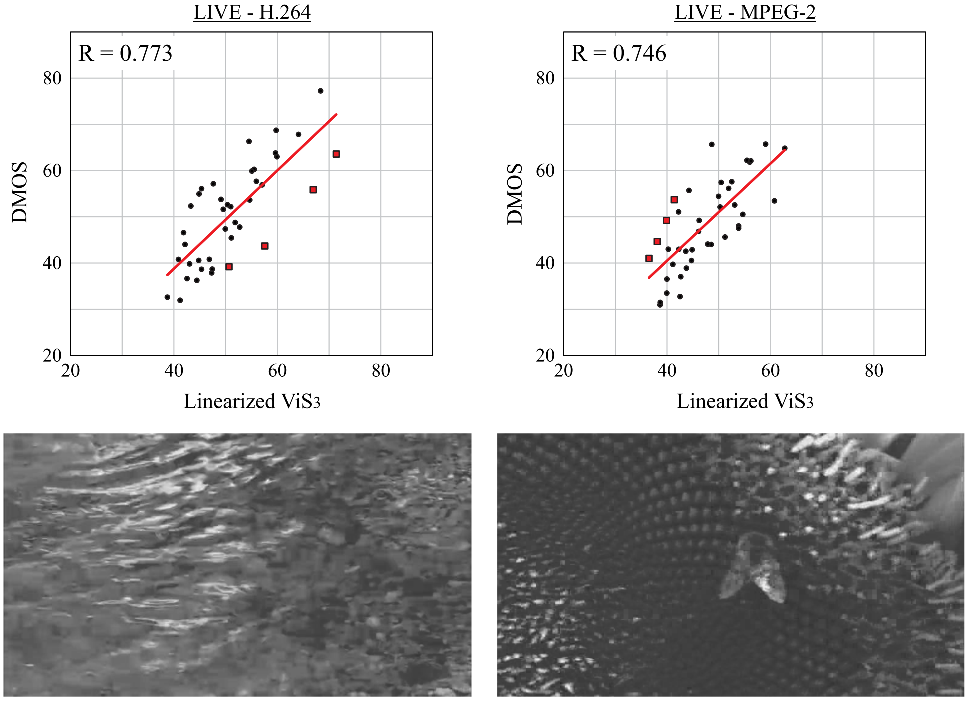
On the CSIQ database, TQV and ViS$_3$ are the two best-performing algorithms on the *H.264* compression distortion; ViS$_3$ and MOVIE are the two best-performing algorithms on three types of distortion (*WLPL*, *SNOW*, and *HEVC*); MOVIE and TQV are the two best-performing algorithms on the *MJPEG*.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

4.3.2 *Performance with different GOF sizes*

As we mentioned in Sec. 3.1, for ViS$_1$, the size of the GOF used in Eqs. (9), (11), and (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 3.

As shown in the upper portion of Table 3, 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



riverbed—H.264 compression

sunflower—MPEG-2 compression

Fig. 13 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 (red square markers in the plots).

Table 3 Performances of ViS$_3$ on the three video databases with different group of frames (GOF) size. Note that ViS$_3$ is robust with the change of the GOF size on all three databases.

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

masking. Nonetheless, as demonstrated in the lower portion of Table 3, 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.

4.3.3 *Statistical significance*

To assess the statistical significance of differences in performances of ViS$_3$ and other algorithms, we used an F-test to compare the variances of the residuals (errors) of the algorithms' predictions.[66] 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. 58) 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 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 smaller the variance of residuals, the better the prediction performance of the algorithm.

Table 4 shows the F-test results between each pair of the algorithms whose distributions of residuals are significantly Gaussian. A "0" value implies that residual variances of two algorithms 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, 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 4, 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

Table 4 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$_3$		–			

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-s video of size 352 × 288 (300 frames total), MOVIE requires ~200 min, whereas basic MATLAB® implementations of VQM and ViS$_3$ require ~1 and 7 min, respectively.

4.3.4 *Summary, limitations, and future work*

Through testing on various video-quality databases, we have demonstrated that ViS$_3$ performs well in predicting video quality. It not only excels at VQA for whole databases with varying types of distortion and varying distortion levels, but 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.

Yet, ViS$_3$ is not without its limitations. One important limitation 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 ~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 ViS$_3$ is that it currently takes 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 (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 psychophysical 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, 67, and 68; see also Ref. 69). 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.

5 Conclusions

In this paper, we have presented a 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 first stage of the algorithm adaptively applies two strategies in the MAD algorithm to groups of video frames to estimate perceived video quality degradation due to spatial distortion. An optical-flow-based weighting scheme is used to model the effect of motion on the visibility of distortion. The second stage of the algorithm measures spatiotemporal correlation and applies an HVS-based model 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. The ViS$_3$ algorithm has been shown to perform well in predicting quality of videos from the LIVE database,[24] the IVPL database,[62] and the CSIQ database.[63] Statistically significant improvements in predicting subjective ratings are achieved in comparison to a variety of existing VQA algorithms. The online supplement to this paper is available in Ref. 60.

Acknowledgments

This material is based upon work supported by the National Science Foundation Awards 0917014 and 1054612, and by the U.S. Army Research Laboratory and the U.S. Army

Research Office under contract/grant number W911NF-10-1-0015.

References

1. B. Girod, *Digital Images and Human Vision*, MIT Press, Cambridge, Massachusetts (1993).
2. A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Trans. Commun.* **43**(12), 2959–2965 (1995).
3. B. A. Wandell, *Foundations of Vision*, Sinauer Associates, Sunderland, Massachusetts (1995).
4. K. Seshadrinathan and A. Bovik, "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Trans. Image Process.* **19**(2), 335–350 (2010).
5. Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Process.: Image Commun.* **19**(2), 121–132 (2004).
6. Z. Wang and Q. Li, "Video quality assessment using a statistical model of human visual speed perception," *J. Opt. Soc. Am. A* **24**(12), B61–B69 (2007).
7. H. R. Sheikh and A. C. Bovik, "A visual information fidelity approach to video quality assessment," in *First Int. Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Scottsdale, Arizona, pp. 23–25 (2005).
8. M. Narwaria, W. Lin, and A. Liu, "Low-complexity video quality assessment using temporal quality variations," *IEEE Trans. Multimedia* **14**(3), 525–535 (2012).
9. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).
10. H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.* **15**(2), 430–444 (2006).
11. S. Wolf and M. Pinson, "In-service performance metrics for MPEG-2 video systems," in *Measurement Techniques of the Digital Age Technical Seminar*, pp. 12–13, IAB, ITU and Technical University of Braunschweig, Montreux, Switzerland (1998).
12. M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Trans. Broadcast.* **50**(3), 312–322 (2004).
13. Y. Wang et al., "Novel spatio-temporal structural information based video quality metric," *IEEE Trans. Circuits Syst. Video Technol.* **22**(7), 989–998 (2012).
14. D. E. Pearson, "Variability of performance in video coding," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Munich, Germany, Vol. 1, pp. 5–8, IEEE (1997).
15. D. E. Pearson, "Viewer response to time-varying video quality," *Proc. SPIE* **3299**, 16–25 (1998).
16. K. Seshadrinathan and A. C. Bovik, "Temporal hysteresis model of time varying subjective video quality," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Prague, Czech Republic, pp. 1153–1156, IEEE (2011).
17. M. A. Masry and S. S. Hemami, "A metric for continuous quality evaluation of compressed video with severe distortions, signal processing," *Signal Process.: Image Commun.* **19**(2), 133–146 (2004).
18. M. Barkowsky et al., "Perceptually motivated spatial and temporal integration of pixel based video quality measures," in *Welcome to Mobile Content Quality of Experience*, pp. 4:1–4:7, ACM, New York (2007).
19. A. Ninassi et al., "Considering temporal variations of spatial visual distortions in video quality assessment," *IEEE J. Sel. Topics Signal Process.* **3**(2), 253–265 (2009).
20. C. Ngo, T. Pong, and H. Zhang, "On clustering and retrieval of video shots through temporal slices analysis," *IEEE Trans. Multimedia* **4**(4), 446–458 (2002).
21. C. Ngo, T. Pong, and H. Zhang, "Motion analysis and segmentation through spatio-temporal slices processing," *IEEE Trans. Image Process.* **12**(3), 341–355 (2003).
22. A. B. Watson and A. J. Ahumada, "Model of human visual-motion sensing," *J. Opt. Soc. Am. A* **2**(2), 322–341 (1985).
23. E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Am. A* **2**(2), 284–299 (1985).
24. K. Seshadrinathan et al., "Study of subjective and objective quality assessment of video," *IEEE Trans. Image Process.* **19**(6), 1427–1441 (2010).
25. American National Standards Institute, "Objective video quality measurement using a peak-signal-to-noise-ratio (PSNR) full reference technique," Tech. Rep. T1.TR.74-2001, American National Standards Institute, Ad Hoc Group on Video Quality Metrics, Washington, DC (2001).
26. E. Larson and D. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," *J. Electron. Imaging* **19**(1), 011006 (2010).
27. A. B. Watson and A. J. Ahumada, *A look at motion in the frequency domain*, NASA Tech. Memo. TM-84352 (1983).
28. P. V. Vu, C. T. Vu, and D. M. Chandler, "A spatiotemporal most-apparent-distortion model for video quality assessment," in *IEEE Int. Conf. on Image Processing*, pp. 2505–2508 (2011).
29. P. V. Vu and D. M. Chandler, "Video quality assessment based on motion dissimilarity," in *Seventh Int. Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Scottsdale, Arizona (2013).
30. S. Chikkerur et al., "Objective video quality assessment methods: a classification, review, and performance comparison," *IEEE Trans. Broadcasting* **57**(2), 165–182 (2011).
31. Video Quality Expert Group, "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).
32. L. Lu et al., "Full-reference video quality assessment considering structural distortion and no-reference quality evaluation of MPEG video," in *IEEE Int. Conf. on Multimedia and Expo*, Lausanne, Switzerland, Vol. 1, pp. 61–64, IEEE (2002).
33. P. Tao and A. M. Eskicioglu, "Video quality assessment using M-SVD," *Proc. SPIE* **6494**, 649408 (2007).
34. A. Pessoa et al., "Video quality assessment using objective parameters based on image segmentation," *SMPTE J.* **108**(12), 865–872 (1999).
35. J. Okamoto et al., "Proposal for an objective video quality assessment method that takes temporal and spatial information into consideration," *Electron. Commun. Jpn.* **89**(12), 97–108 (2006).
36. S. O. Lee and D. G. Sim, "New full-reference visual quality assessment based on human visual perception," in *Int. Conf. on Consumer Electronics*, Las Vegas, Nevada, pp. 1–2, IEEE (2008).
37. M. Barkowsky et al., "Temporal trajectory aware video quality measure," *IEEE J. Sel. Topics Signal Process.* **3**(2), 266–279 (2009).
38. A. Bhat, I. Richardson, and S. Kannangara, "A new perceptual quality metric for compressed video," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, pp. 933–936, IEEE (2009).
39. U. Engelke et al., "Modelling saliency awareness for objective video quality assessment," in *Second Int. Workshop on Quality of Multimedia Experience*, Scottsdale, Arizona, pp. 212–217 (2010).
40. X. Gu et al., "Region of interest weighted pooling strategy for video quality metric," *Telecommun. Syst.* **49**(1), 63–73 (2012).
41. M. Narwaria and W. Lin, "Scalable image quality assessment based on structural vectors," in *IEEE Int. Workshop on Multimedia Signal Processing*, Rio De Janeiro, Brazil, pp. 1–6, IEEE (2009).
42. A. A. Stocker and E. P. Simoncelli, "Noise characteristics and prior expectations in human visual speed perception," *Nat. Neurosci.* **9**(4), 578–585 (2006).
43. Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *Conf. Record of the Thirty-Seventh Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, California, Vol. 2, pp. 1398–1402, IEEE (2003).
44. F. Lukas and Z. Budrikis, "Picture quality prediction based on a visual model," *IEEE Trans. Commun.* **30**(7), 1679–1692 (1982).
45. A. Basso et al., "Study of MPEG-2 coding performance based on a perceptual quality metric," in *Proc. of Picture Coding Symp. 1996*, Melbourne, Australia, pp. 263–268, IEEE (1996).
46. C. J. van den Branden Lambrecht, "Color moving pictures quality metric," in *IEEE Int. Conf. on Image Process.*, Lausanne, Switzerland, Vol. 1, pp. 885–888, IEEE (1996).
47. P. Lindh and C. J. van den Branden Lambrecht, "Efficient spatio-temporal decomposition for perceptual processing of video sequences," in *IEEE Int. Conf. on Image Processing*, Lausanne, Switzerland, Vol. 3, pp. 331–334, IEEE(1996).
48. A. Hekstra et al., "PVQM—a perceptual video quality measure," *Signal Process.: Image Commun.* **17**(10), 781–798 (2002).
49. A. B. Watson, J. Hu, and J. F. McGowan, "Digital video quality metric based on human vision," *J. Electron. Imaging* **10**(1), 20–29 (2001).
50. C. Lee and O. Kwon, "Objective measurements of video quality using the wavelet transform," *Opt. Eng.* **42**(1), 265–272 (2003).
51. E. Ong et al., "Video quality metric for low bitrate compressed videos," in *IEEE Int. Conf. on Image Processing*, Singapore, Vol. 5, pp. 3531–3534, IEEE (2004).
52. E. Ong et al., "Colour perceptual video quality metric," in *IEEE Int. Conf. on Image Processing*, Genoa, Italy, Vol. 3, pp. III-1172-5, IEEE (2005).
53. M. Masry, S. Hemami, and Y. Sermadevi, "A scalable wavelet-based video distortion metric and applications," *IEEE Trans. Circuits Syst. Video Technol.* **16**(2), 260–273 (2006).
54. P. Ndjiki-Nya, M. Barrado, and T. Wiegand, "Efficient full-reference assessment of image and video quality," in *IEEE Int. Conf. on Image Processing*, San Antonio, Texas, Vol. 2, pp. II-125–II-128, IEEE (2007).
55. S. Li, L. Ma, and K. N. Ngan, "Full-reference video quality assessment by decoupling detail losses and additive impairments," *IEEE Trans. Circuits Syst. Video Technol.* **22**(7), 1100–1112 (2012).
56. P. C. Teo and D. J. Heeger, "Perceptual image distortion," in *IEEE Int. Conf. on Image Processing*, Austin, Texas, Vol. 2, pp. 982–986, IEEE (1994).
57. S. Péchard et al., "A new methodology to estimate the impact of H.264 artefacts on subjective video quality," in *Third Int. Workshop on Video Processing and Quality Metrics*, Scottsdale, Arizona (2007).

58. D. Chandler and S. Hemami, "VSNR: a wavelet-based visual signal-to-noise ratio for natural images," *IEEE Trans. Image Process.* **16**(9), 2284–2298 (2007).
59. B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *7th Int. Joint Conf. on Artificial Intelligence*, Vol. 2, pp. 674–679, Morgan Kaufmann Publishers Inc., San Francisco, California (1981).
60. P. Vu and D. Chandler, "Online supplement: ViS$_3$: an algorithm for video quality assessment via analysis of spatial and spatiotemporal slices," 2013, http://vision.okstate.edu/vis3/ (25 December 2013).
61. J. G. Robson, "Spatial and temporal contrast-sensitivity functions of the visual system," *J. Opt. Soc. Am.* **56**(8), 1141–1142 (1966).
62. Image & Video Processing Laboratory, The Chinese University of Hong Kong, "IVP subjective quality video database," http://ivp.ee.cuhk.edu.hk/research/database/subjective/index.shtml (20 April 2012).
63. Laboratory of Computational Perception & Image Quality, Oklahoma State University, "CSIQ video database," 2013, http://vision.okstate.edu/csiq/ (15 November 2012).
64. F. Bellard et al., "FFMPEG tool," http://www.ffmpeg.org (15 November 2012).
65. EBU Technical Review, "Subjective quality of internet video codec phase II evaluations using SAMVIQ," Tech. Rep., European Broadcasting Union (2005).
66. H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Trans. Image Process.* **15**(11), 3440–3451 (2006).
67. U. Engelke, V. X. Nguyen, and H. Zepernick, "Regional attention to structural degradations for perceptual image quality metric design," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Las Vegas, Nevada, pp. 869–872, IEEE (2008).
68. J. You et al., "Perceptual quality assessment based on visual attention analysis," in *Proc. of the 17th ACM Int. Conf. on Multimedia*, pp. 561–564, ACM, New York, NY (2009).
69. O. L. Meur et al., "Overt visual attention for free-viewing and quality assessment tasks: impact of the regions of interest on a video quality metric," *Signal Process.: Image Commun.* **25**(7), 547–558 (2010).

Phong V. Vu received his BE in telecommunications engineering from the Posts and Telecommunications Institute of Technologies, Hanoi, Vietnam, in 2004. He is currently working toward his PhD degree in the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, Oklahoma. His research interests include image and video processing, image and video quality assessment, and computational modeling of visual perception.

Damon M. Chandler received his BS degree in biomedical engineering from Johns Hopkins University, Baltimore, Maryland, in 1998, and his MEng, MS, and PhD degrees in electrical engineering from Cornell University, Ithaca, New York, in 2000, 2004, and 2005, respectively. He is currently an associate professor in the School of Electrical and Computer Engineering at Oklahoma State University, Stillwater, Oklahoma, where he heads the Laboratory of Computational Perception and Image Quality.