Temporal Quality Assessment for Mobile Videos

An (Jack) Chan, Amit Pande, Eilwoo Baik and Prasant Mohapatra
University of California, Davis, CA 95616, USA
{anch, pande, ebaik, pmohapatra}@ucdavis.edu

ABSTRACT

Video quality assessment in mobile devices, for instances smart phones and tablets, raises unique challenges such as unavailability of original videos, the limited computation power of mobile devices and inherent characteristics of wireless networks (packet loss and delay). In this paper, we present a metric, Temporal Variation Metric (TVM), to measure the temporal information of videos. Despite its simplicity, it shows a high correlation coefficient of 0.875 to optical flow which captures all motion information in a video. We use the TVM values to derive a reduced-reference temporal quality assessment metric, Temporal Variation Index (TVI), which quantifies the quality degradation incurred in network transmission. Subjective assessments demonstrate that TVI is a very good predictor of users' Quality of Experience (QoE). Its prediction shows a 92.5% of correlation to subjective Mean Opinion Score (MOS) ratings. Through video streaming experiments, we show that TVI can also estimate the network conditions such as packet loss and delay. It depicts an accuracy of almost 95% in extensive tests on 183 video traces.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Video

General Terms

Performance

Keywords

Video quality, Quality of Experience, mobile devices

1. INTRODUCTION

In 2011, video traffic has accounted for more than 50% of the total traffic in mobile networks [4]. Quality of Experience (QoE) of watching videos over mobile devices, such as smart phones and tablets, has been attracting interest from content providers and network service providers.

MobiCom'12, August 22–26, 2012, Istanbul, Turkey.

Effective and efficient video quality metrics are highly desirable for service providers. The metric can help them to extract quick feedback from end-users and can enable them to "turn knob" at their end (if possible) to enhance the quality of service. Many tools have evolved for evaluating video quality delivered to end users. Most of them re-use the techniques in image quality measurements which evaluate the spatial quality, such as Peak Signal to Noise Ratio (PSNR [10]) and Structural SIMilarity index (SSIM [25]) of each video frame. However, video quality comprises of both spatial and temporal quality.

Unfortunately, unlike its counterpart, the temporal quality assessment cannot re-use the existing static image quality measurement techniques. Videos over mobile devices (e.g. smart phones & tablets) are usually delivered through wireless such as Wi-Fi or cellular networks. This includes a host of applications including interactive multimedia applications, video-on-demand (VOD), video chat & HDTV streaming. We call these videos as ***"mobile videos"***. Temporal quality degradation due to packet loss and delay is more pervasive in wireless networks than in wired scenarios. Mobile devices also pose unique constraints such as limited computation power, small memory, short battery life and low video resolution. Existing temporal quality assessment schemes tend to depend on the availability of the original video (i.e. full-reference metrics) [16] or incur heavy computational cost in computing motion information [23], neither of which is possible in mobile context.

In this paper, we present a novel scheme for temporal quality assessment. We first measure the motion information of a video by comparing the consecutive frames of a video. Considering both the accuracy and the low computational cost, we design a temporal information metric, we call it Temporal Variation Metric (TVM). TVM evaluates the difference of the corresponding pixel values in the two neighboring frames to estimate the motion of objects in the video. Compared to other frame comparison techniques proposed in video quality research community, TVM has the shortest running time and a relatively low memory usage. Such properties make it suitable for using in mobile videos. Despite its simplicity, TVM shows an average Pearson correlation coefficient of 0.875 with the optical flow, while the optical flow is widely used to measure the actual temporal motion in a video.

From TVM, we also design a new reduced-reference temporal quality metric, Temporal Variation Index (TVI). TVI measures the temporal quality degradation between the source (original) and the received videos. From subjective video

quality tests, we find that TVI strongly correlates with Mean Opinion Score (MOS) of a pool of human observers watching the video. It is worth noting that TVI evaluates the temporal quality of the received video without the original copy of the video. TVI can quantify the perceived video quality degradation caused by coding or communication impairments.

In addition to the correlation between TVI and MOS, through extensive experiments, we also find that the correlation between TVI and network impairments, in particular packet loss rate and end-to-end delay. With such correlation, by measuring TVI in the video application layer at the user end, we can estimate the packet loss rate and delay in the wireless networks. This information will be extremely useful to network service providers to improve the video streaming quality (and thus Quality of Experience (QoE)) by adjusting video coding and allocating network resources.

The main contributions of this paper are as follows:

1. We propose Temporal Variation Metric (TVM) to measure the temporal information of mobile videos. TVM has low computation and memory requirements to suit the mobile devices and it closely relates to motion information in videos.

2. From TVM, we derive a new temporal quality metric, Temporal Video Index (TVI). TVI quantifies the perceived quality degradation between the source and received videos.

3. Subjective tests demonstrate the strong correlation between TVI and end-user QoE.

4. We also show that TVI can estimate the network impairments, in particular packet loss rate and end-to-end delay.

Although TVM and TVI are video quality metrics, they can help facilitate many other technologies and applications. For example, TVM and TVI are very useful for content distribution in multicast groups to maximize users' QoE. They are also useful in scalable video delivery for quality-oriented bandwidth allocation in wireless networks. Mobile videos in surveillance applications can also be monitored to guard against network impairments. TVI and TVM can be used to monitor video quality, to benchmark video processing systems and to be embedded in communication systems to optimize algorithms and parameter settings for content delivery.

The rest of paper is organized as follows: Section 2 gives the motivation for developing a new temporal metric. We propose TVM in Section 3. To show the advantages of TVM, we also discusses other potential candidate metrics for estimating the motion information of a video. In Section 4, we derive TVI from TVM. We show that TVI demonstrates a strong correlation to the MOS values obtained from subjective tests. In Section 5, we further explore the relationship between TVI and network metrics in particular packet loss rate and end-to-end delay. Section 6 concludes the paper.

2. MOTIVATION

The most accurate way to obtain the end-user Quality of Experience (QoE), such as Mean Opinion Score (MOS) ratings [12] or Crowdsourcing [3], is to conduct subjective tests involving human subjects. Such experiments have to be conducted offline and incurs huge cost in terms of labor and time. More importantly, we cannot use the subjective measurement for real-time monitoring to guarantee the quality of video delivery. Therefore, an objective measurement of QoE is necessary to feedback and maintain the quality of delivered video over mobile devices.

2.1 Getting Video Temporal Information

A video is composed of both spatial and temporal information. The spatial information comprises of appearance of objects, resolution, smoothness, etc. The degradation of spatial quality in videos causes blocking and blurring. A number of metrics derived from static image quality assessment can be re-used to measure spatial quality across a video. Such examples include Peak Signal-to-Noise Ratio (PSNR), Structural SIMilarity index (SSIM [25]), VQM [29], and Blocking[14].

Temporal information is the measure of the motion of objects in a video or movement of background including scene changes. Temporal quality measurement needs motion information in a video, so existing static image quality evaluation metrics cannot be re-used for this purpose. Motion information is formally obtained by calculating optical flow. It is based on the movement of an object in a video frame caused by a relative motion between an observer and the scene [27]. Optical flow field is represented by vectors of the object points. The length of the vectors represents the magnitude of the motion while the direction of the vectors represents the direction of the motion [9]. Motion vectors, which are widely used for motion estimation in video coding, are a special case of optical flow when vectors are computed in a macro block basis [13]. Calculating the optical flow or motion vectors is a computationally intensive task which is unsuitable for embedded devices. Pauwels and Hulle [20] reported that a GPU (GeForce FX 5800) and CPU (Pentium-4 2.8 GHz) implementation of optical flow calculation for low resolution 256x256 images take around 13 ms and 35 ms of dedicated computation time respectively.

Unfortunately, most existing temporal quality assessments [23, 30] still depend on measuring temporal information by optical flows and motion vectors. To reduce the computation cost of extracting optical flow or motion vectors, the authors in [16] assume that the motion vectors during video encoding can be re-used in temporal quality assessment. However, such assumption is not always valid. In many cases, the codecs may be closed source (consider for example Skype video). Furthermore, it may be very difficult to account for optical flow information from the codec. Take Scalable Video Coding (SVC) codec as an example. There are tiers of encoding/decoding done to achieve scalability, so it is very difficult to assess how motion vectors in different layers are to be assimilated and compared. Thus, it is desirable to build a temporal quality metric for videos which can provide measurements without using motion vectors nor being dependent on the underlying video codecs.

2.2 Desired Properties of a Video Quality Metric

The degradation of temporal quality is primarily observed as delay, freezing(stop-motion), blockiness and blackout. A desired temporal quality metric should be able to capture these effects alongside with motion information.

There is a gap between the temporal quality evaluation

and end-user QoE which is how people actually perceive the video quality. There has been little effort to quantify the correlation between end-user QoE and temporal quality metric measurements. Therefore, a desired temporal quality metric, in addition of capturing the video temporal quality degradation, should also have a great correlation to the end-user QoE.

Although the optical flow gives the accurate temporal information of a video, mobile devices are computational power limited, making them impossible to re-compute the optical flow from reconstructed frames. Therefore, a desired video temporal quality metric should have low computational cost.

To make the video temporal quality metric be readily applied to the end-user mobile devices, the metric should be video codec independent. Of course, video content providers can always customize the metric to their own codecs for the optimized use.

In some wireless networks, for example Wi-Fi, the network impairments such as the packet loss rate and the delay are easily measured. (These network impairments are only the measurement of Quality of Service (QoS) of the network not the QoE of the end users.) In service providers' point of view, it is also interesting to know the network impairments so that they can "tune" the parameters, for example network bandwidth allocation, to increase QoS and QoE. The bad news is in mobile networks these network impairments are not always easily available. For instance, to measure the packet loss rate in video streaming over cellular networks, we require packet transmission information from the base station. Comparing how many packets sent from the base station, and how many packets received in the mobile device, we can calculate the packet loss rate. More importantly, to access the network layer metrics in mobile devices, e.g. smart phones, we need the root access privilege which is usually not open to general public, such as application developers.

Therefore, it is desirable if an application-layer video quality metric can also be an indication of network impairments which lead to quality degradation. The network metrics such as packet loss and delay can be estimated using the video quality metric obtained at the end user, allowing the service providers to modify the channel resources allocation or video coding to guarantee the quality of videos delivered.

In addition, existing temporal quality metric tend to depend on the availability of the original video (i.e. full-reference metrics) [16]. This is not feasible for on-line computations where it is not possible to have original video for referencing.

In summary, a video temporal quality metric should have the following desired features:

- Captures the degradation of the temporal quality;
- Correlates with user QoE;
- Low computational cost;
- Codec independent;
- Estimates the network impairments such as packet loss and delay; and
- Does not require the original copy of the video.

2.3 Related Work

There is a large amount of existing efforts on video spatial quality assessments [5, 26]. However, there are few works about video temporal quality which becomes important when video applications move to wireless context. MOVIE index [23] integrates both spatial and temporal aspects of distortion assessment using a full-reference technique which requires the original copy of the video. The metric is designed based on modeling of human visual system. It shows a good performance but also incurs a high computational cost.

Moorthy and Bovik [16] present a video quality assessment algorithm and enable the motion vector re-use in the decoding stage to reduce the computational complexity. But the drawback is that not all video codecs have the motion vectors reusable during the decoding process. The metric is also full-reference.

Yang et al. [30] consider various factors such as frame dropping, scene boundary, motion activity and motion mapping in the video to estimate the temporal quality. It reuses motion activity and motion mapping information from video codec and packet loss information from network. Again, it depends on the reusability of motion information during the decoding process. It also requires a cross-layer design to get the packet loss information from the network layer.

Vidal and Gicquel [19] presents a no-reference quality metric for detecting fluidity impairments due to frame losses in video. This mainly accounts for frame dropping in the video. The subjective experiments by [11] suggest that content, motion magnitude and frame rate are some factors affecting perception of temporal artefacts. This motivates us to develop a new metric, which quantifies the end user perception of the video and accounts for these factors.

2.4 Our Approach

Considering the large number of video codecs and service providers, we look for a codec independent solution for temporal quality assessment. We compare the similarity of two neighboring (consecutive) frames of video to estimate the motion in video. A large similarity between two frames indicates a slow motion of objects during the time period of these two frames. Instead of recording the difference of each pixel, we propose only record the average similarity score of a whole frame. This decreases the complexity compared to those in optical flow or motion vector calculations. Furthermore, consecutive frames comparison can be done in a no-reference manner without a copy of source video at the receiver.

We call our metric measuring the motion of objects in the video as Temporal Variation Metric (TVM). TVM is suitable for mobile devices because of low computational power requirement and small memory overhead. Based on TVM values between the source and the received videos, we design a reduced-reference temporal quality metric, Temporal Variation Index (TVI). TVI measures the temporal quality degradation in mobile video streaming. It is worth noting TVI only needs the TVM of the original video but not the whole copy of the original video. This makes it easy to be deployed in mobile devices and services.

We conduct extensive subjective tests to demonstrate the correlation between TVI and the subjective user experience. Based on the subjective tests, we related the TVI and the Mean Opinion Score (MOS) which is the end-user QoE.

With the linear regression technique, we derive the equations to estimate MOS from TVI. In addition, we find a strong correlation between TVI and network packet loss. We also derive equations to estimate the packet loss rate from TVI. By feeding back TVI to the source, the content providers and network services providers can adjust the video coding or network resources to improve the end-user experience of video streaming.

3. MEASURING TEMPORAL INFORMATION

To develop a temporal quality metric, accurately measuring the temporal information of a video is important. We design Temporal Variation Metric (TVM) to measure such information.

3.1 Temporal Variation Metric (TVM)

Temporal Variation Metric (TVM) measures temporal information of a video by comparing the content amongst consecutive frames in a video. Two successive frames are compared to obtain a single score by calculating the Peak Signal to Error Ratio in the consecutive frames. This is analogous to Peak Signal-to-Noise Ratio (PSNR) measurement used in comparing reconstruction quality of images and videos. Different from PSNR, we do not use any original video frame, but instead use previous frame of the video as a reference and calculate temporal motion in the video. Unlike PSNR which is an indicator of noise introduced in the image due to communication, TVM is an indicator of motion or temporal variation in the current frame as compared with previous frame.

TVM only uses the two neighboring frames for measuring temporal variation, so if any temporal distortion happens in the current frame will only be recorded by the current and the next TVM calculations. This effect in the current and the next TVM values will not propagate to the subsequent TVM calculations.

Numerically, TVM is calculated as log of mean square value of difference between two consecutive frames (F_{p-1} and F_p) of the video (measured in dB).

$$TVM_p = 10 \log_{10} \left(\frac{k^2}{d} \right) \quad (1)$$

where k is a constant for a video with a particular color depth. It is the maximum pixel value of the video frame. For example, if the color depth of the video is 8-bit, $k = 255$. d is the mean square difference of the corresponding pixels in two frames, F_{p-1} and F_p. Logarithm is used to compensate the non-linearity of human visual system.

$$d = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(F_p(i,j) - F_{p-1}(i,j) \right)^2$$

where $F_p(i,j)$ is the value of the pixel in i^{th} row and in j^{th} column in the p^{th} frame. Each frame is of size $M \times N$ pixels.

3.2 Consecutive Frames Comparison

Besides the peak signal-to-error ratio which we used in TVM, there are other spatial quality metrics which can be modified to measure the difference between two frames. Thus, they can also be potential candidates for measuring temporal information. In this section, we compare other candidates with TVM to evaluate their appropriateness to be a measurement of the temporal information in mobile videos. We select three representative candidates from the research literature: Structural Similarity Index (SSIM), Edge Similarity Score (ESS) and Visual Signal-to-Noise Ratio (VSNR). These three metrics represent a wide choice amongst existing research. Instead of using them directly, we apply these metrics on *consecutive* frames of a same video to measure the motion in the video. Thus, the derived three potential temporal metrics are respectively called TSSIM, TESS and TVSNR.

TSSIM (Temporal Structure Similarity Index) - [24] proposes SSIM to measure the structural similarity of two frames. SSIM first measure the luminance which is the intensity of the pixel values of the frames. Then, the luminance is removed from the frames and the contrast which is the standard deviation between two frames is obtained. Then the contrast measure is also removed to measure the structural similarity between two frames. We derive TSSIM to measure the SSIM of the two consecutive frames. It is robust to common spatial artifacts such as mean-shifts, contrast-stretch, compression (codec) losses, blurring and salt-pepper noise.

TESS (Temporal Edge Similarity Score) - [28] proposes ESS to reflect the way that human perceives visual information. The important information extracted by human visual system includes spatial-luminance information, edge and contour information [7]. TESS measures the edge and contour change information of the consecutive frames to evaluate the motion. The frames are first partitioned into macro blocks and then Sobel's mask is applied to get the edge information along each pixel along both horizontal and vertical directions. Edge direction information is obtained from these values and quantizing it into one of the eight representative directions that are equally spaced by $\pi/8$ radians (from $-\pi/2$ to $\pi/2$). The dominant edge direction for each macroblock is computed and the average cosine of the difference in edge directions measures the consecutive edge similarity score of the consecutive frames.

TVSNR (Temporal Visual Signal to Noise Ratio) - VSNR metric is proposed in [2] to capture the human visual system properties. It operates in two stages. (1) It computes the contrast thresholds for the detection of distortions via wavelet-based models of visual masking and visual summation. Then it determines the visibility of distortions in an image. If the distortions are below the threshold, the distorted image is deemed to be of perfect visual fidelity (VSNR = infinite) and no further analysis is required. (2) To measure supra-threshold distortions, the low-level visual property of the perceived contrast and the mid-level visual property of global precedence are used. These two properties are modeled as Euclidean distances in the distortion-contrast space using a multi-scale wavelet decomposition. Linear sum of these distances is computed for consecutive frames in our case to get TVSNR value.

3.3 Testing with Videos

To evaluate whether the proposed TVM is a good metric to measure the temporal information of videos, we test it with six different videos and compared the result with TSSIM, TESS and TVSNR. The videos we used are listed in Table 1. They have different resolutions, lengths and degrees of motion. A long video will constitute a combination of scenes of all degrees of motion. Therefore, by separately

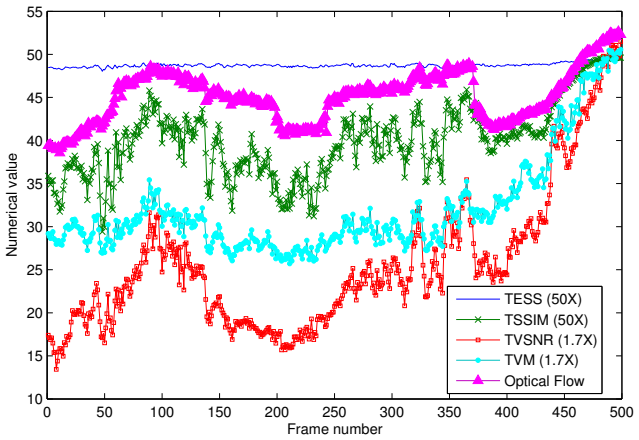


Figure 1: Plot showing the correlation of TVM, TESS, TSSIM and TVSNR with Optical Flow (dB)

Table 1: Details of sample videos

Name	Length	Resolution	Motion
Old Woman	30 sec	1920 × 1080	1(relatively slow)
Cows	60 sec	1920 × 1080	2
Foreman	10 sec	352 × 288	3
Duck	10 sec	1920 × 1080	4
Park	10 sec	1920 × 1080	5
Intersection	60 sec	352 × 288	6(relatively fast)

considering these scenes of different motion degrees, we have actually considered the variety in existence. These few video clips are the representative video sequences chosen. In the experiments presented later, we have used hundreds of traces with varying network conditions.

For each video, we calculate TVM, TSSIM, TESS TVSNR and the optical flow. Figure 1 shows the variation of the optical flow and different temporal information measurements for the video sequence *Duck*. In this video, TESS has low correlation while the other metrics show high correlation. TSSIM curve shows many uncorrelated high frequency components. Table 2 shows the Pearson Correlation Coefficients of the different temporal information measurements with the optical flow. We find that both TVSNR and TVM have the highest correlation coefficients with the optical flow.

Then we evaluate the computation time and memory overhead required by different temporal information measurements. Table 3 gives the details of computation time required by those metrics on a 12GB 12-core Intel Xeon (dual-core) processor running Matlab release 2011. TVM gives the shortest running time which shows its low computational cost. As expected, due to its huge computational cost, the optical flow calculation requires the longest time.

In memory overhead measurements, we exclude the optical flow and only focus on the four different temporal information measurements. We find that different video content or degree of motion has no obvious effect on the memory requirements. Instead, the resolution of the video greatly influences the memory overhead of temporal information measurements. Table 4 shows the memory overhead of temporal information measurements. We scale the video sequence *Park* into three different resolutions for comparison. We find that TVM and TESS have the smallest memory overhead in operation.

From the above tests and comparisons, we have shown that TVM is the most suitable candidate for measuring temporal information considering its accuracy and suitability for mobile devices.

Table 2: Pearson Correlation Coefficients of Temporal Information Measurements with Optical Flow

Video	TVM	TSSIM	TESS	TVSNR
Old Woman	0.7719	**0.8033**	0.7500	**0.8054**
Cows	**0.8679**	0.7476	0.7408	**0.9153**
Foreman	**0.8598**	0.7720	0.7390	**0.9156**
Duck	**0.9106**	**0.9183**	-0.2461	0.7477
Park	0.6072	**0.7828**	0.7200	**0.9178**
Intersection	**0.9241**	0.5706	0.6107	**0.9230**

Table 3: Running Time (in seconds in 3 significant figures) of Different Temporal Information Measurements in Matlab

Video	OP	TVM	TSSIM	TESS	TVSNR
Old Woman	46600	**9.65**	99.9	6230	45.8
Cows	83100	**11.6**	129	11700	84.5
Foreman	16700	**0.983**	8.38	1980	13.6
Duck	29700	**3.70**	21.5	3280	23.9
Park	29600	**2.97**	20.0	3290	24.6
Intersection	84000	**12.7**	117	11500	84.5

OP - optical flow.

Table 4: Memory Overhead (in KB in 3 significant figures) of Different Temporal Information Measurements

Resolution	TVM	TSSIM	TESS	TVSNR
Low (352x288)	**428**	23300	**315**	3240
Medium (960x640)	**1930**	129000	**1120**	19000
High (1920x1080)	**6200**	428000	**3440**	67800

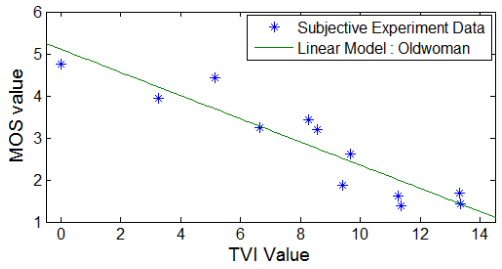
4. ESTIMATING VIDEO TEMPORAL QUALITY

With the help of TVM, we derive a new reduced-reference temporal quality metric, Temporal Variation Index (TVI). We then show that TVI has a strong linear relationship with end-users' quality of experience (QoE). We can use TVI to accurately predict QoE.

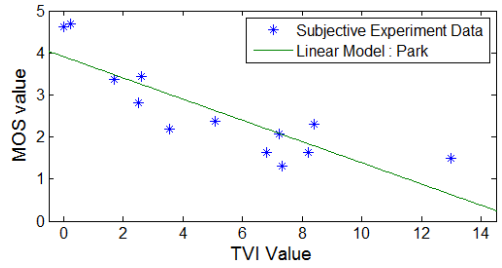
4.1 Temporal Variation Index (TVI)

TVM is a measure of temporal information between two neighboring frames in a video. TVM of the original video measures the motion of the objects in the video. In addition to the motion, TVM of the received video in the end user also indicates the temporal quality degradation. For example, a large TVM value can imply either the objects in the video scene moves very fast or there are some losses in the video frame sequence. But by comparing the temporal information of the received video with that of the original video, we can measure the temporal information change. Such temporal information change is due to a degradation of temporal quality in video processing, such as wireless video streaming. Therefore, we design a reduced-reference temporal quality metric, Temporal Variation Index (TVI), to estimate the temporal quality degradation. TVI removes the "motion" part which is in the TVM values.

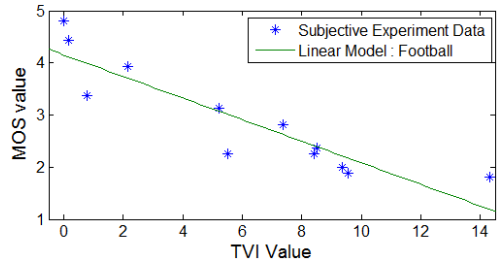
TVI of a video at time t, $TVI(t)$, is defined as follows.



(a) *Slow Motion* (Old Woman)



(b) *Moderate Motion* (Park).



(c) *Fast Motion* (Football).

Figure 2: Linear fits of all three video categories for MOS prediction.

$$TVI(t) = \frac{|TVM_s(t) - TVM_r(t)|}{TVM_s(t)} \quad (2)$$

where $TVM_r(t)$ is the TVM value for the received frame at current time, t. For every received video frame, the mobile device records its time stamp, t. It then calculates TVM, $TVM_r(t)$, of the received video at time t by comparing frame at that time, f_t, with the preceding frame $f_{t-\delta}$. $\delta = \frac{1}{fps}$, where fps is the frame per second of the video. The receiver will search the corresponding TVM value $TVM_s(t)$ of the source video from the control data, and calculate $TVI(t)$.

In case the original video itself has some temporal noise or distortion, TVI would not indicate such temporal distortion. Therefore, TVI merely measures the temporal distortion or noise incurred in the transmission, such as streaming, process. Generally speaking, TVI can measure the temporal distortion incurred in any video processing procedure, for example, video transcoding. In this paper, we focus on wireless video transmission.

It is worth noting that TVI does not require the copy of the original video, but TVM of the original video. We assumed that the sender computes the TVM values of the original video and sends them to the receiver through the control channel during the streaming. This assumption is reasonable as TVM calculation has low computation and storage requirement. For each pair of video frames, a TVM value is represented by a 32-bit float number. So, the size of the temporal information of a 2-hour movie with 25 frame-per-second (fps) is only around 0.7MB. If we compress it (zip), the size is merely a few KB. Therefore, the overhead for adding the TVM values into the video during streaming is very small. In a typical video streaming, this temporal information can be sent through out-of-band control channel, such as Real-Time Control Protocol (RTCP) sender report (SR) [22], so that the receiver can get this information for every pair of neighboring frames.

Although TVI only captures the temporal quality degradation of the received video, it is a very good predictor for the end-users' QoE. In the next subsections, we carry out subjective tests and show that TVI accurately predict the Mean Opinion Score (MOS) of different videos.

4.2 Subjective Experiments

To test the TVI with videos having different degrees of motion, we add another video sequence, *Football*, into our video samples. *Football* has scenes with very fast motion. We select three videos with different degrees of motion for the experiments. They are *Old Woman*, *Park* and *Football*, where *Old Woman* has the slowest motion while *Football* has the fastest motion.

We first stream these three videos over our wireless testbeds with different channel conditions. We collect a total of 50 samples of these videos with different quality. There are 17 samples of *Old Woman*, 17 samples of *Park* and 16 samples of *Football*. We then write an Android application on Nexus S smartphone and load the collected sample videos on the phone. We engaged 17 volunteers as subjects to watch the videos on the smartphone [1].

The Android app asked each subject to score the watched video on a standard five-grade scale [12]. Score 1 is for a video with the worst quality and it means the impairment in the video is very obvious and very annoying. Score 5 is for a video with the best quality and it means the video is perfect.

When a subject rates video quality while viewing a video, his/her rating changes as the video plays. In a long video, the subject would give several different quality scores to the different portions of the video. Basically, it is same as watching several short clips of video, and the subject gives one score for each clip. Therefore, in these subjective tests and the subsequent experiments in Section 5, we use short video clips, each plays around 10 seconds, which respectively contain slow, moderate and fast motion scenes. It simplifies our experimental settings and procedures.

Our test was performed according to the ITU single-stimulus (SS) method [12]. The standard videos with the five different scores were shown to the viewer at the beginning of the test. During the test, only the videos to be scored were shown without any display of the standard videos. For each video, the android app records the quality scores given by the subjects and obtain a mean score that is the Mean Opinion Score (MOS).

4.3 Estimating QoE

After the tests, we group the video samples according to their content. i.e. there are three groups, *Old Woman*, *Park* and *Football*. They respectively correspond to three motion

[1]According to ITU-R BT.500-11 subjective assessment standard [12], 15 subjects would be enough for subjective quality evaluation

Table 5: Estimation of the linear model for each video category.

Predicted MOS	Linear Model $(\hat{\beta}_0, \hat{\beta}_1)$	95% Confidence Interval of $\hat{\beta}_1$	ρ_{MOS}
$\widehat{TMOS}_{slow}$	(5.1, -0.28)	[-0.346 -0.205]	0.8708
$\widehat{TMOS}_{moderate}$	(3.9, -0.25)	[-0.345 -0.161]	0.9396
$\widehat{TMOS}_{fast}$	(4.2, -0.21)	[-0.268 -0.146]	0.9655

categories, *slow*, *moderate* and *fast*. We also measure TVI for each video sample. For each category, we will predict the MOS with TVI measurement.

For each category, we randomly select four samples and separate them from the others. These four samples will serve as a validation set. The remaining samples are used to derive an equation to predict the MOS with TVI. We use the linear model in the derivation. We propose a two-parameter linear model to predict MOS of each video category.

$$TMOS_{md} = \beta_0 + \beta_1 \overline{TVI} \tag{3}$$

for some constants β_0 and β_1. In this linear model, we use the average TVI, $\overline{TVI}$ of the video as the predictor variable. $TMOS$ is the predicted MOS, not the actual MOS that is evaluated from the human subjects. Hence, $TMOS$ is an objective video quality metric based on $\overline{TVI}$. We use a subscript md to indicate the motion degree of the video. In our categorization, md can be *slow*, *moderate* or *fast*.

It is interesting to note that during the TVI measurement, TVI values for some frames in the video are infinite. It is because that the TVM values of the corresponding frames in the received video are infinite. When there are frames lost and delay, the receiver playback mechanism will duplicate the last received frame. Therefore, the TVM value evaluated goes to infinite. We will explain this phenomenon in more detail in Section 5. For MOS prediction, we take infinite TVI value as 1. In fact, most TVI values are smaller than 1.

If we predict a quality score, Y, given by a particular user, we have

$$Y = \beta_0 + \beta_1 X + \epsilon \qquad E[\epsilon] = 0 \tag{4}$$

where X can be any predictor variable. There is also an error term, ϵ, added in the regression analysis. But, we are only interested in predicting the mean value of Y that is $TMOS$, hence we ignore the error term [15]. The same principle is also applied to the prediction of network impairment that we will discuss in Section 5.

Figure 2 shows the linear fits of the estimated $TMOS$, $\widehat{TMOS}$, for each video category. We use the *linear model* package of the statistics tool, ***R*** [21], to derive $\hat{\beta}_0$ and $\hat{\beta}_1$, that are respectively the estimates of β_0 and β_1 in Equation (3). We have the following general equation for the linear model estimation.

$$\widehat{TMOS}_{md} = \hat{\beta}_0 + \hat{\beta}_1 \overline{TVI} \tag{5}$$

The resultant $\widehat{TMOS}_{md}$, $\hat{\beta}_0$ and $\hat{\beta}_1$ for each video category are shown in Table 5.

The results shows that TVI has a very strong linear relation with MOS which is the users' QoE. Take *slow*-motion video as an example, its 95% confidence interval for $\hat{\beta}_1$ is (-0.346, -0.205). The small interval indicates that the sample size (number of videos) in the training set is large enough for a good estimation. Mean of $\hat{\beta}_1$ (-0.28) is significant to $TMOS$ prediction. This justifies our decision of including $\overline{TVI}$ in our linear model for predicting MOS.

With $\hat{\beta}_0$ and $\hat{\beta}_1$ in Table 5, we then use the TVI of the video samples in the validation set to estimate TMOS of each sample. Comparing the estimated TMOS and the actual MOS we measured from the subjective experiments, we calculate the Pearson Correlation Coefficient between TMOS and MOS. Table 5 also shows such correlation coefficients (ρ_{MOS}). We find that across the different video categories, TMOS is highly correlated with MOS with an average correlation coefficient of 0.925. It shows TVI is a very good predictor of QoE.

5. NETWORK EVALUATION

A good video quality metric for mobile videos should not only estimate QoE accurately but also predict the network impairments, such as packet loss and delay. With extensive video streaming experiments, we demonstrate how TVI computations can be used to accurately estimate delay and packet loss.

5.1 Experimental Setup

We set up a single-hop wireless testbed. The WLAN configuration for the testbed is IEEE802.11n and the average minimum end-to-end delay is 0.604 milliseconds. Like in Section 4, we use three representative videos, one for each motion degree (slow, moderate and fast), for this set of experiments. We use FFmpeg [6] for video coding and VideoLan [18] for video streaming.

During the streaming, we use IPFireWall tool [8] to deliberately introduce packet loss and delay into the wireless link. Packet loss is injected as uniform distributed or as bursty pattern or a combination of both. Delays are injected randomly. Details of packet loss rate and delay are specified in each subsection below. Packet loss and delay in the network lead to the loss of temporal quality in the video. We see different extent of blocking, blurring and freezing in the receiver end when the packet loss rate and delay change.

For each streaming, both the sender and receiver calculate the TVM values of the video it sends and receives. The receiver also calculates the TVI values by matching the corresponding TVM values in received video with source video. TVI, TVM, packet loss rate and delay values are collected for each video streaming session. We then analyse the collected data sets with Matlab and R software.

5.2 Detecting Packet Loss

We first conduct experiments to observe the variation in TVM and TVI values with packet losses in the network. We vary the packet loss rate (PLR) from 0.1% to 3% for high-resolution videos and from 5% to 50% for low-resolution videos.

A low uniform PLR causes more significant distortions in high-resolution videos than in low-resolution ones. We can see an example in Figure 3. In Figure 3, *Old Woman* is an high-resolution video while *Foreman* is a low-resolution video. *Old Woman* with 3% PLR seems to have the same distortion as *Foreman* with 50% PLR. This is because high-resolution videos have larger spatial coding interdependencies than low-resolution videos. The frame size is large in a high-resolution video. If the frame size is larger than Maxi-

(a) 0.5% Loss; Old Woman (b) 3% Loss; Old Woman

(c) 5% Loss; Foreman (d) 5% Loss; Foreman

Figure 3: Comparing low and high resolution videos with different network packet loss effects

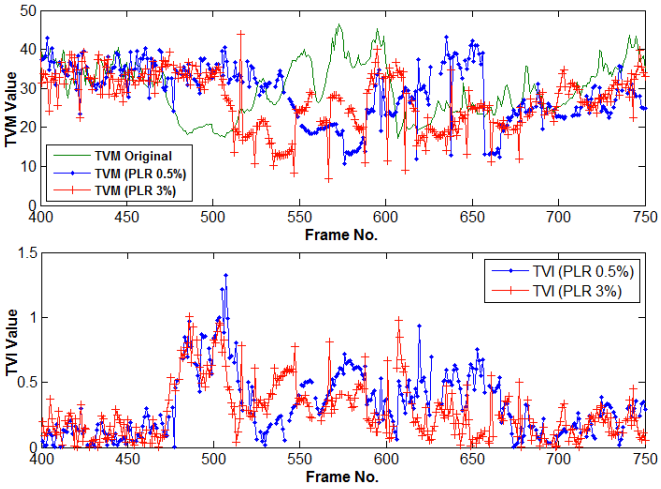


Figure 4: TVM and TVI computations for Video *Old Woman*

mum Transmission Unit (MTU), the frame is split into several packets during streaming. A packet lost in the network will cause incomplete reconstruction of the frame in the receiver, leading to display distortion. However, if the video resolution is low, a frame can be represented by a single packet during streaming. A packet loss results in a whole frame loss. In the perspective of human visual system, an occasional single frame loss is not as obvious to visual artifacts. Therefore, low-resolution videos are more "immune" to packet loss.

From TVI measurements, we find that increasing PLR leads to higher TVI values, indicating larger temporal quality degradation. Figure 4 shows the variation of TVM and TVI values for a high resolution video (*Old Woman*) when PLR is 0.1% and 3%. Figure 5 shows those values for a low resolution video (*Foreman*) when PLR is 5% and 50%. The variation of TVI of *Old Woman* with 3% PLR looks even greater than the variation of TVI of *Foreman* with 50% PLR due to visual distortions introduced by packet loss.

If we look at Figure 4 and Figure 5 more closely, we find there are many unconnected dots in Figure 5. This is due the infinite (INF) values of TVM and TVI. When we compute TVM, the current frame f_t becomes exactly the same as the preceding frame, $f_{t-\delta}$, where $\delta = \frac{1}{fps}$, so the value of $TVM_r(t)$ goes to INF according to Equation (1). When $TVM_r(t)$ is INF, according to Equation (2), TVI also becomes INF. These "INF" values are represented by breaks or discontinuity in Figure 5.

If the original video has some stationary scenes, $TVM_s(t)$ during those scenes will be recorded as INF. If there is no impairment in the streaming process, the received video will also have the stationary scenes, so $TVM_r(t)$ will also have corresponding INF values. It creates subtlety in arithmetic in Equation (2). To avoid that, we treat INF as a symbolic arithmetic variable in Equation (2). If there are two INFs in the numerator, they will cancel each other. In this way, any INF values in TVI indicate freezing due to packet losses.

Uniform and burst losses

Uniform and burst losses have different effect on temporal video quality. We demonstrate such effect by introducing uniform packet loss and burst packet loss in the same video streaming session but in different times. The uniform packet loss was applied from 1 second to 3.5 second of the video at PLR of 30%. In this uniform loss period, the packets are dropped randomly with a probability of 30%. The dropped

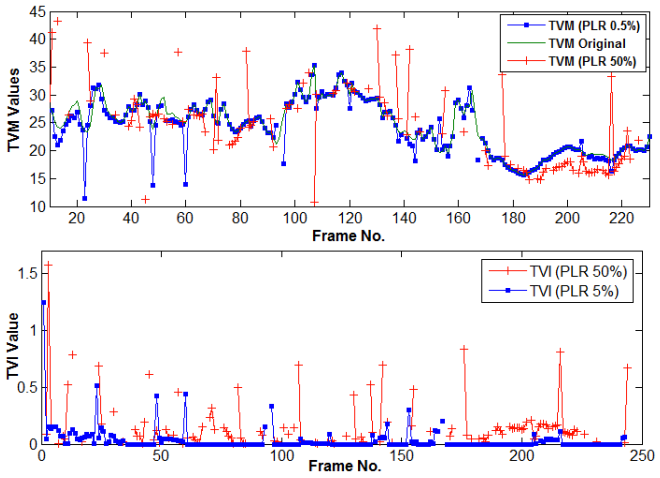


Figure 5: TVM and TVI computations for Video *Foreman*

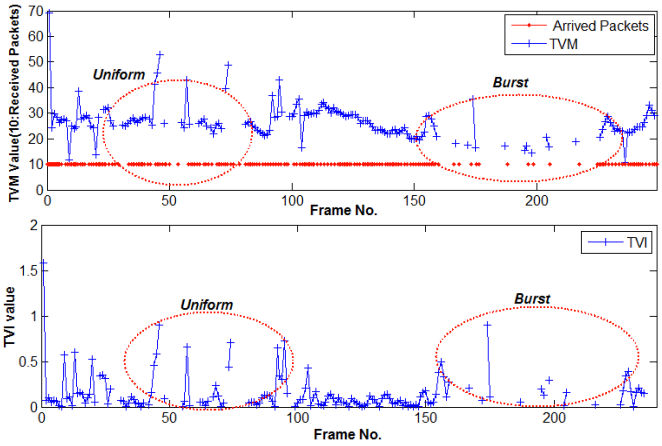


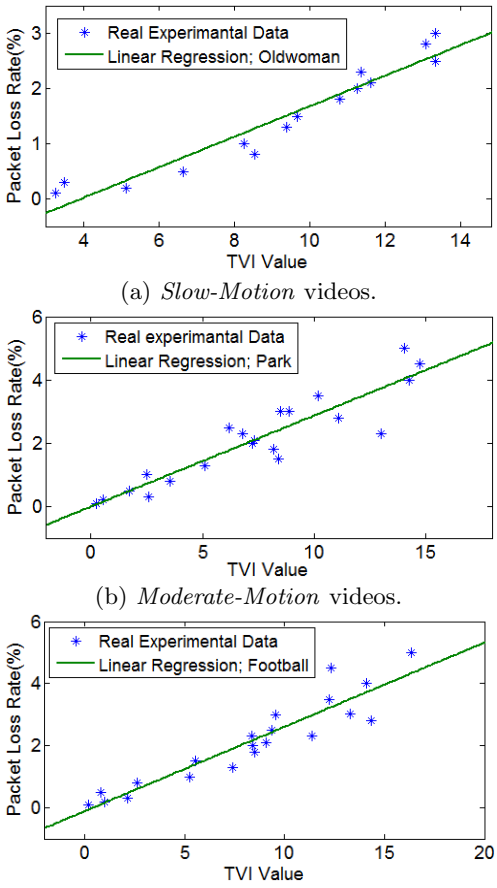
Figure 6: TVM and TVI values at PLR(30%) Types:Uniform and Burst(dense);

packets are not necessary adjacent to each other. Burst packet loss happens from 5 second to 7 second. Within the burst loss period, the system drops the packets with an average probability of 30%. But once a packet is selected to be dropped, its neighboring packets will also be dropped with a probability of 90% to 95%.

Figure 6 shows the TVM and TVI values for a sample video. It can be clearly observed that there are more INF values (discontinuities) for burst loss period than that in uniform loss period. This is attributed to the fact that burst losses cause more frame losses while a uniform loss may only impair the quality of frame without frame loss. That is for burst loss period, there will be more freezing of the video. An end-user gets a very annoying experience with subsequent freezing of video frames (number of INF values in our case).

5.3 Predicting Packet Loss Rate

In Section 5.2, we discuss how to use TVI to detect packet loss in a video streaming. Generally, when TVI value increases (including the number of INF increases), the packet loss rate also increases. Therefore, we expect there is a linear relationship between TVI and packet loss rate (PLR). To confirm this relationship, we carried out an extensive video streaming experiments. In the experiment, different videos are streamed over our wireless testbed (refer to Section 5.1) and different PLR are introduced during streaming. The received videos are collected and their TVI are measured.



(a) *Slow-Motion* videos.

(b) *Moderate-Motion* videos.

(c) *Fast-Motion* videos.

Figure 7: Linear fits of all three video categories for PLR prediction.

We choose three videos for the experiment: the videos *old woman* has slow-motion, *park* has moderate-motion, and *football* is in the fast-motion category. For slow-motion video, we collect 60 samples from the experiment, among them 40 are for linear regression model while 20 are for validation. For moderate-motioned video, we collect 78 samples: 42 are for regression model and 36 are for validation. We collect 45 samples for fast motion video: 30 are for regression model while 15 are for validation.

We propose to have have following linear model for TVI to predict PLR.

$$TPLR_{md} = \beta_0 + \beta_1 \overline{TVI} \quad (6)$$

for some constants β_0 and β_1. In this linear model, we use the average TVI, $\overline{TVI}$ of the video as the predictor variable. *TPLR* is the predicted PLR, not the actual PRL that is measured from the network metrics [2]. Similarly to the derivation of *TMOS*, we use a subscript *md* to indicate the

[2]In our case, PLR is predefined as a system input

Table 6: Reliablilty to expect actual packet loss; Validation with linear regression models.

Predicted PLR	Linear Model $(\hat{\beta}_0, \hat{\beta}_1)$	95% Confidence Interval of $\hat{\beta}_1$	ρ_{PLR}
$\widehat{TPLR}_{slow}$	(-1.091, 0.277)	[0.234 0.319]	0.9655
$\widehat{TPLR}_{moderate}$	(-0.002, 0.287)	[0.236 0.343]	0.9498
$\widehat{TPLR}_{fast}$	(-0.104, 0.271)	[0.224, 0.318]	0.9046

motion degree of the video. Again, *md* can be *slow*, *moderate* or *fast*. Like the *TMOS* derivation, we take TVI INF value as 1 if there is any.

Figure 7 shows the linear fits of the estimated $TPLR$, $\widehat{TPLR}$, for each video category. We use the *linear model* package of the statistics tool, ***R*** [21], to derive $\hat{\beta}_0$ and $\hat{\beta}_1$, that are respectively the estimates of β_0 and β_1 in Equation (6). We have the following general equation for the linear model estimation.

$$\widehat{TPLR}_{md} = \hat{\beta}_0 + \hat{\beta}_1 \overline{TVI} \tag{7}$$

The resultant $\widehat{TPLR}_{md}$, $\hat{\beta}_0$ and $\hat{\beta}_1$ for each video category are shown in Table 6.

The results shows that TVI has a very strong linear relation with PLR. Take *slow*-motion video as an example, its 95% confidence interval for $\hat{\beta}_1$ is (0.234, 0.319). The small interval indicates that the sample size (number of videos) in the training set is large enough for a good estimation. Mean of $\hat{\beta}_1$ (0.28) is significant to $TPLR$ prediction. This justifies our decision of including $\overline{TVI}$ in our linear model for predicting PLR.

With $\hat{\beta}_0$ and $\hat{\beta}_1$ in Table 6, we then use the TVI of the video samples in the validation set to estimate TPLR of each sample. Comparing the estimated TPLR and the actual PLR we measured from the streaming experiments, we calculate the Pearson Correlation Coefficient between TPLR and PLR. In Table 6, ρ_{PLR} indicates the Pearson's Correlation Coefficient from the validation data. We find that across the different video categories, TPLR is highly correlated with PLR with an average correlation coefficient of 0.940. It shows TVI is a very good predictor of PLR.

5.4 Detecting and Predicting Delay

Delay causes freezing in streaming video services. In this section, we discuss the relationship between end-to-end delay and TVI. We first describe the impairment of the video quality caused by delay.

The frames transmitted by the source are split into packets and transmitted into the network. The packets received in the client are reconstructed into frames and played by decoder. Any delay introduced in the network will lead to delay in packets being received by the receiver and subsequently in freezing of frames. Figure 8 shows this effect. The delay in network causes modifications in TVI computations and INF values are generated in the TVI measurements at receiver. Unlike the INF values generated by packet losses, these INF values are followed by a lagged version of TVM values. The receiver estimates the best match for TVM values at receiver to the values at source (using a sequence alignment like algorithm, see [1, 17] for details).

In order to predict the delay effect, TVM values are calculated first in the receiver end. In the bottom of figure 8,

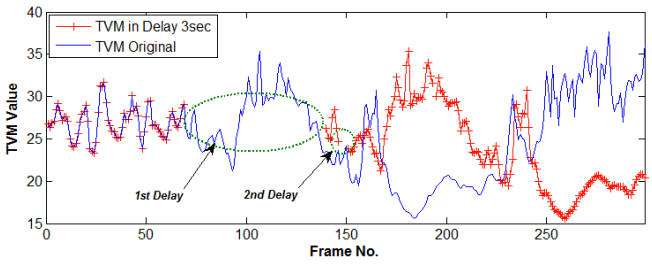


Figure 9: TVM values of Video *Foreman* with a delay of 3 seconds

Table 7: Estimation of Network Delay using TVI computations

Video	fps	D(actual)	# INF	$\hat{D}$	Error(%)
Duck	50	2.00	99	1.98	1
Foreman	25	3.00	75	3.00	0
Park	50	2.00	99	1.98	1
Oldwoman	30	4.50	155	5.17	15

after the first two frames, there is delay between frame $f2$ and frame $f3$. After $f2$ arrived well, decoder cannot play the very next frame on time so it has to wait until it receives next frame. This brings freezing or stop-motion in the display. Therefore, INF are produced during the delay because there is no frame to be calculated so it processes with the previous one. This does not necessary mean that another $f2$ is made to calculate the corresponding TVM.

Based on the TVM values of the receiver side, The TVM values are compared with TVM of source video to get TVI values. There are two delays in the figure 8. The first delay causes 3 INFs and the second one brings 2 INFs in this example.

Figure 9 shows TVM values of original and delayed video *foreman*. The dotted circle indicates vacant part of the graph, which are INFs in TVM computations of the delayed video. The TVM values of received video are shifted to right and INFs appear in between. An prediction of end-to-end delay (D) can be obtained by counting these INFs.

$$\hat{D} = \frac{I}{fps} \tag{8}$$

where fps denotes the frame rate of the video. I is the number of INFs in between two non-INF TVMs at the receiver end.

Table 7 shows average results for the four videos we selected to test the accuracy of end-to-end delay prediction. As only number of INFs in TVM is counted, unlike MOS and PLR prediction, the delay prediction is video motion independent. Therefore, we just randomly choose the four different video contents from our video pool. The inaccuracy of the delay prediction mainly come from the inaccuracy of TVM matching (the application of sequence alignment algorithm).

In Table 7, # INF means the total number of INFs (pertaining to delay) in the whole playing time. The estimation has the accuracy of over 95%.

In practical scenarios, we calculate TVI as the video streaming is ongoing. We will first detect the delay by applying existing sequence alignment methodologies [1, 17]. After

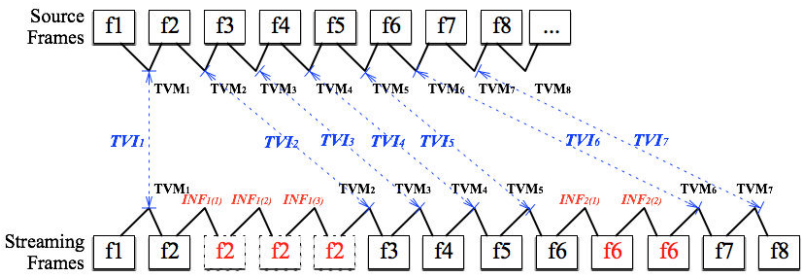


Figure 8: TVM/ TVI computations in presence of network delay

excluding those delay-caused INFs, we detect and predict the packet loss rate based on TVI values.

6. CONCLUSIONS

Accurately evaluating video streaming quality over networks always attracts great attention from content providers and network service providers. However, video quality evaluation itself is a challenging topic as users' QoE cannot be measured with low cost. When the video applications move to mobile devices, evaluating video quality becomes even more challenging. Wireless networks cause packet loss and delay to the video streaming while mobile devices have limited computation power and memory. In this paper, we have proposed a novel reduced-reference temporal quality metric, Temporal Variation Index (TVI) for mobile videos. Despite its simplicity, it shows strong linear relationship with users' Mean Opinion Score (MOS). Therefore, it is a very good predictor for users' QoE on mobile videos. Its prediction of MOS has a correlation coefficient of 0.925 with the actual MOS measured from subjective tests. TVI also predicts network impairments, such as packet loss and end-to-end delay. From our extensive experiments of video streaming over a wireless network testbed, we find that the packet loss rate predicted by TVI has a correlation coefficient of 0.94 with the measured loss rate. During the computation of TVI, we can also predict the end-to-end delay with an accuracy of around 95%. These appealing features make TVI a very useful tool for content and network service providers to offer the future high quality video streaming services.

Although temporal quality is a unique and important aspect for video, it is also very interesting to integrate TVI with other spatial quality metrics. We believe that such integration will provide a comprehensive video quality evaluation metric for the future use.

7. ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation through the Grant No. CNS-0831914.

8. REFERENCES

[1] A. Chan, K. Zeng, P. Mohapatra, S. Lee, and S. Banerjee. Metrics for evaluating video streaming quality in lossy ieee 802.11 wireless networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[2] D. Chandler and S. Hemami. Vsnr: A wavelet-based visual signal-to-noise ratio for natural images. *Image Processing, IEEE Transactions on*, 16(9):2284–2298, 2007.

[3] K. Chen, C. Chang, C. Wu, Y. Chang, and C. Lei. Quadrant of euphoria: a crowdsourcing platform for qoe assessment. *Network, IEEE*, 24(2):28–35, 2010.

[4] CiscoSystem. Cisco visual networking index: Global mobile data traffic forecast update, 2010 - 2015, 2011.

[5] U. Engelke and H.-J. Zepernick. Perceptual-based quality metrics for image and video services: A survey. In *Next Generation Internet Networks, 3rd EuroNGI Conference on*, pages 190 –197, may 2007.

[6] FFMPEG. Ffmpeg. `http://www.ffmpeg.org`, 2012.

[7] D. Field, A. Hayes, and R. Hess. Contour integration by the human visual system: Evidence for a local. *Vision Research*, 33(2):173–193, 1993.

[8] F. Group. Freebsd ipfirewall. `http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/`, 2012.

[9] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

[10] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13):800 –801, 19 2008.

[11] Q. Huynh-Thu and M. Ghanbari. Temporal aspect of perceived quality in mobile video broadcasting. *Broadcasting, IEEE Transactions on*, 54(3):641 –651, sept. 2008.

[12] ITU. *Methodology for the subjective assessment of the quality of television pictures*. International Telecommunication Union: Recommendation ITU-R BT.500-11, 2002.

[13] D. Kim, S. Cho, and J. Jeong. A motion vector recovery algorithm for temporal error concealment using optical flow in h.264 video coding. *Multimedia and Expo, IEEE International Conference on*, 0:1713–1716, 2006.

[14] A. Leontaris and A. Reibman. Comparison of blocking and blurring metrics for video compression. In *Proceedings ICASSP*, 2005.

[15] N. Matloff. *From Algorithm to Z-Scores: Probabilistic and Statistical Modeling in Computer Science*. Creative Commons Lincense, `http://heather.cs.ucdavis.edu/matloff/probstatbook.html`, 2009.

[16] A. K. Moorthy and A. C. Bovik. Efficient video quality assessment along temporal trajectories. *IEEE Trans. Circuits Syst. Video Techn.*, 20(11):1653–1658, 2010.

[17] D. W. Mount. *Bioinformatics: sequence and genome analysis*. CSHL press, 2004.

[18] V. Organization. Videolan. `http://www.videolan.org`, 2012.

[19] R. Pastrana-Vidal and J. Gicquel. Automatic quality

assessment of video fluidity impairments using a no-reference metric. In *Proc. of Int. Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2006.

[20] K. Pauwels and M. V. Hulle. Realtime phase-based optical flow on the gpu. In *Computer Vision and Pattern Recognition Workshops*. IEEE Computer Society, 2008.

[21] R. The r project for statistical computer. `http://www.r-project.org`, 2009.

[22] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Real time control protocol (rtcp) attribute in session description protocol (sdp). RFC 3550, The Internet Society, 2003.

[23] K. Seshadrinathan and A. C. Bovik. Motion tuned spatio-temporal quality assessment of natural videos. *IEEE Trans. on Image Processing.*, 2010.

[24] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[25] Z. Wang, L. Lu, and A. Bovik. Video quality assessemnt based on structural distortion measurement. *Signal Processing: Image Communication*, 19(2):121–132, February 2004.

[26] Z. Wang, H. Sheikh, and A. Bovik. Objective video quality assessment, 2003.

[27] D. H. Warren and E. R. Strelow. *Electronic Spatial Sensing for the blind: contributions from perception*. Springer, 1985.

[28] C. Wu and C. Kuo. Design of integrated multimedia compression and encryption systems. *Multimedia, IEEE Transactions on*, 7(5):828–839, 2005.

[29] F. Xiao et al. Dct-based video quality evaluation. *Final Project for EE392J*, page 769, 2000.

[30] K.-C. Yang, C. Guest, K. El-Maleh, and P. Das. Perceptual temporal quality metric for compressed video. *Multimedia, IEEE Transactions on*, 9(7):1528 –1535, nov. 2007.