Noname manuscript No.
(will be inserted by the editor)

QoE Prediction Model for Mobile Video Telephony

Shraboni Jana and An Chan and Amit Pande and Prasant Mohapatra

Received: date / Accepted: date

Abstract Interactive online video applications, such as video telephony, are known for their vulnerability to network condition. With the increasing usage of hand-held wireless mobile devices, which are capable of capturing and processing good quality videos, combined with the flexibility in an end-user movements have added new challenging factors for application providers and network operators. These factors affect the perceived video quality of mobile video telephony applications, unlike conventional video telephony over desktop computers. We investigate this impact on video quality of mobile video telephony in varying network conditions and end-users movement scenarios. Based on 312 live traces, we quantitatively derive the correlation between the perceived video quality and the network Quality of Service (QoS) and user mobility. With the results, we develop a Quality of Experience (QoE) prediction model for mobile video telephony using Support Vector Regression techniques. The prediction models display ≈ 0.8 pearson correlation with experimental data. Our methodology and findings can be used to guide the video telephony application providers and network operators to work towards satisfying end-user experience.

Keywords Mobile Video Telephony, Perceptual Video Quality, Quality of Experience, Support Vector Regression

Shraboni Jana
Department of Electrical and Computer Engineering,
University of California, Davis, USA.
E-mail: sjana@ucdavis.edu

An Chan
Department of Computer Science,
University of California, Davis, USA.
E-mail: anch@ucdavis.edu

Amit Pande
Department of Computer Science,
University of California, Davis, USA.
E-mail: pande@ucdavis.edu

Prasant Mohapatra
Department of Computer Science,
University of California, Davis, USA.
E-mail: pmohapatra@ucdavis.edu

1 Introduction

With the increase in smartphone users and the rapid growth of video applications (apps) [5], mobile video traffic keeps gaining major portion of the mobile traffic. The smartphone video apps, being the major contributor of mobile video traffic, can be categorized into interactive and non-interactive video apps. Among interactive apps, gaming and video telephony are the two most sought-after smartphone video apps. Particularly, the use of mobile video telephony is on a rapid rise in both enterprise and consumer worlds [13]. Many video telephony applications for mobile phones have been rolled-out in the market. Fring, Tango, Skype, FaceTime, Vtok, ooVoo are just a few examples of them. With improvement in access network bandwidths (LTE 4G networks, 802.11ac), video coding (HEVC or H.265 codec) and device technology (quad-core processors), we expect a big leap in the user expectations of the quality of experience (QoE) in video telephony in coming years.

It is obvious that the state of the network impacts the quality of video transmitted through it. If the network path suffers from significant jitter or loss, the perceived video quality at the smartphone gets deteriorated. Buffering techniques, which are useful for non-interactive videos, such as streaming applications, cannot be of help for delay-intolerant interactive videos, in particular, video telephony applications. These stringent network Quality of Service (QoS) requirements and demand for anywhere any-time connectivity for smartphone video telephony apps have posed tremendous challenges to the network operators. It, therefore, becomes quintessential to examine the impact of the underlying network on the performance of such video apps.

However, for end-users, they only care about the quality of experience (QoE) of the delivered multimedia services. Most of the mobile video telephony apps are proprietary software with undisclosed communication protocols and often using encrypted channels. Thus, it is insufficient for network operators to evaluate the end-user QoE by simply inspecting video packets in the network. In addition, application providers need to know how their apps performed in given networking conditions. Hence, both network operators and application providers can benefit from QoE prediction models that map end-user perception of delivered video services to the network QoS parameters.

Furthermore, mobile devices make the treatment of video quality of video telephony apps different than what has been perceived conventionally [14]. The smartphones are not stationary as laptops or desktops during video call conversation. Hence, unlike other end-devices, smartphone end-users have the flexibility to hold the end-device and move around. This leads to the following questions:

1. How a given networking condition impacts the perceptual video quality of smartphone video telephony apps?
2. Does end-user mobility impact end-user perceptual video quality? In other words, for video quality assessment, should the brisk movements by the end-user with his/her smartphone be treated differently as compared to the scenario when the end-device is stationary like a laptop or a desktop?
3. Can prediction of perceptual video quality of the apps be based on the networking conditions? If so, what guarantees can be made and what not?

Although many works have characterized the performance of video telephony apps [19, 26, 31, 16, 25], to the best of our knowledge, very limited work has been done in terms of video telephony apps' perceptual quality. Also, to the best of our knowledge, no work has been done in this aspect with end-devices being mobile (for example, smart-

phones). For concreteness, we choose two representative video telephony applications, Skype and Vtok (based on Google Talk API) for studying their perceptual video quality in various network states and end-user mobility scenarios. The main contributions of this work can be summarized below:

- In this work, we characterize the performance of two major video telephony applications (Skype and Vtok) in terms of perceptual video quality. To the best of our knowledge, this is the first effort in this direction;
- Our study provides application providers and network operators with a deeper understanding of how video telephony works, the impact of network impairments such as packet losses, bandwidth and delay as well as the influence of user mobility;
- Based on objective and subjective evaluations, we build QoE prediction models for both Skype and Vtok.

Note, we have used the terms QoE prediction model and Mean Opinion Score (MOS) prediction model interchangeably. We substantiate our claims using 312 live traces, each trace lasting for around 4 minutes of video conversations over smartphones. Although some findings we draw are specific to these two applications, most of them are generic to video telephony applications. The insights gained from our work can be used by the application providers of mobile video telephony to properly handle motion in videos without sacrificing video quality. The network operators can also use these prediction models to plan ahead to optimize network resources and take appropriate steps to troubleshoot any issues arising in the network that may deteriorate the video quality. For example, in a cellular network, the base station can intelligently schedule the scarce bandwidth resource according to the video applications running on the user equipment based on the QoE prediction model.

The rest of this paper is organized as follows. Section 2 details experiment setup and evaluation metrics. In Section 3 we present our experimental results and discuss its implications in Section 4. In Section 5 we map our objective metrics to MOS scores using subjective evaluations as per the ITU Standards [12] and in Section 6, we develop a video telephony QoE prediction model, VTQoE. Section 7 discusses the related work. In Section 8, we conclude our findings.

2 Methodology

We develop QoE models from live video telephony sessions of both Skype and Vtok mobile apps. In this section we present the layout of experimental setup used to capture live traces of these apps and also discuss evaluation metrics employed to arrive at the required QoE models.

2.1 Experimental Setup

We use an experimental approach to examine the relationship between the video quality of video telephony apps and the network-layer parameters. To examine these relationships, we set up a controlled testbed between two Android SAMSUNG Exhibit II smartphones. Smartphone 1 and Smartphone 2 are in different network domains.

Figure 1 shows the experimental setup. The data collection is done for following two scenarios:

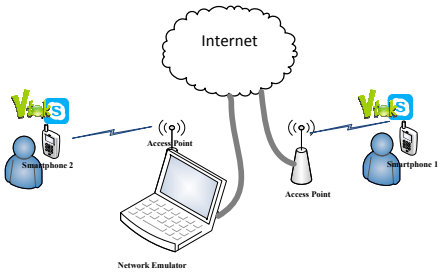


Fig. 1 Setup and layout of experimental environment. Two smartphones are configured on two non-interfering WiFi access points connected with each other via a network emulator and backbone Internet connection.

1. *Stationary*– In the *Stationary* scenario, both Smartphone 1 and Smartphone 2 are stationary during video telephony. But the end-users are free to have body movements.
2. *Mobile*– In the Mobile scenario, Smartphone 2 is recording Smartphone 1. End-user does brisk movements in the surroundings with Smartphone 1.

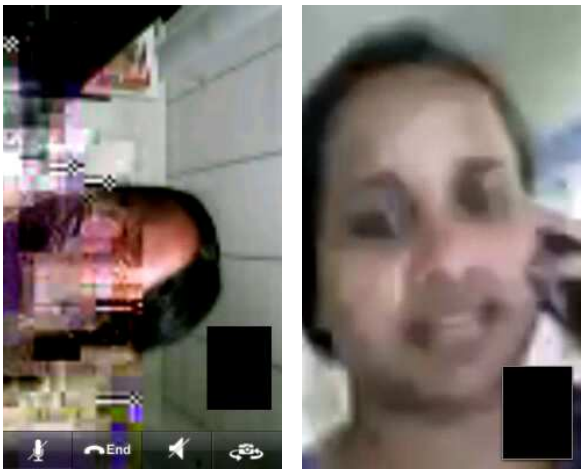
In both the scenarios, smartphone cameras are focused to capture the facial part body movements only. Smartphone 2 is connected to a Campus Access Point. A HP compaq nc6000 laptop is also configured as an access point to which Smartphone 1 connects. The laptop is equipped with Athores 802.11abg wireless cards operating using Madwifi driver. Each nc6000 has two Wifi antennas. The packets from and to the laptop access point are forwarded using Ethernet. The network emulator is also installed on the laptop. We use NETEM [18] for network emulation functionality. NETEM is used for testing protocols by emulating the properties of wide area networks. The network emulator is connected to the Internet via Ethernet. It emulates variable delay, loss, incoming and outgoing bandwidth by the command line tool 'tc' which is a part of the iproute2 package of tools. The packets are captured at both the smartphones using tcpdump.

We took 312 live video telephony traces. Each session is more than 4 minutes long. We capture the video of the chat session using Screencast Video Recorder. Screencast captures smartphone screen at 21-22 fps and saves it into a MPEG4 video with resolution 240x400. We use FFMPEG [8] to convert the captured video into sequence of bitmap image files loaded later into Matlab for further analysis.

We did our experiments in a dedicated environment and at dedicated wireless frequency channels. We did repeated experiments to measure the last access-link losses due to the end-user brisk movements with smartphones at the same location and found it to be negligible. This indicates that mobility or brisk walk doesn't induce packet losses or delays in the network and ensures that all 'effective' packet losses are occurring at NETEM and not at wireless hop.

2.2 Evaluation Metrics

Subjective and objective assessments are two major techniques to evaluate QoE. In subjective assessment, Mean Opinion Score (MOS) are collected using human subjective evaluation. Objective measurements do not capture the video quality perceived by



(a) Blocking snapshot (b) Blurring snapshot

Fig. 2 Sample of (a) Blocking and (b) Blurring (zoomed) snapshots from video call sessions.

HVS (Human Visual System), but nonetheless, provide a good approximation of video artifacts. Moreover, the cost of subjective assessment is very high in terms of time and required man-power.

Given that our dataset has 312 video recordings of each being at least 4 minutes long, it will be highly tedious (both in terms of time and resources) to perform subjective evaluations of all these videos. Hence, we first analyze the experimental data using objective metrics. In later Section 5, we perform subjective evaluations with lesser number of videos to evaluate the efficacy of the objective metrics used.

There are three basic models for video objective assessment. Full-reference and reduced-reference, two of the three models, require full or partial knowledge of original video signal. Mobile video telephony being real-time proprietary applications, it is not possible to obtain the original video transmitted by the sender. We therefore, use no-reference model, which is oblivious of the original video signal, to evaluate the quality of video telephony. In our model we use no-reference spatial metrics - Blocking & Blurring [27] to evaluate the spatial component, and use no-reference temporal metric - Temporal Variation Metric (TVM) [2] to evaluate the temporal component of the video.

Blocking - Existing video compression standards such as MPEG-x and H.26x, have adopted block-based methods. In block-based methods, the image of a video is partitioned into 8×8 blocks. Discrete Cosine Transform (DCT) is applied to the pixels in each block and then each block is independently quantized prior to encoding.

The blocking artifact is induced by two main reasons - compression in efficiency and network packet losses. During compression, each block being quantized independently may cause blocking artifact where as packet losses lead to full or partial loss of the block information. In such cases, the reconstruction of the video at the decoder will be erroneous, in turn causing visually apparent discontinuities across block boundaries (Figure 2(a)). We implement the technique proposed in [28] for measuring the amount of blocking artifact in video traces captured during video telephony.

Blurring - Blurring happens due to the loss of high frequency information. Natural images have typically much lower energy at high frequencies. Therefore, the high-frequency DCT coefficients have lower magnitudes. During the process of quantization, these high-frequency coefficients tends to be zero. Consequently, the decoded image will be blurred (Figure 2(b)).

The reasons for blurring varies from image acquisition to packet loss in blocked images. The packet loss can lead to loss of partial information which may cause blurring. We implement the model proposed in [17] to evaluate the amount of blurring in the captured videos. The images in video are divided into 8×8 blocks. Based on the histogram computations of the DCT coefficients of the entire image, and filtering out the ones with zero DCT values, blurring metric is calculated. Finally, the metric is normalized to remove dependency on the image size.

Temporal Smoothness - Blocking and blurring evaluate the video quality in the spatial dimension. But, in addition to the spatial dimension, video also has the temporal dimension. Temporal information is the measure of the motion of objects in a video or movement of background including scene changes. We use the recently proposed metric TVM [2], to measure the temporal information of the video conversations. Due to the specific scenarios and video content, i.e. video telephony, are studied in this paper, we modified TVM so that it can measure the temporal impairment. According to [2], TVM is for temporal information measurement due to its high correlation with optical information. 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).

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

where k is the color depth of a video. It depends on the number of bits used to represent a pixel in a video frame. $k = 255$ if 8 bits are used. d is the mean square difference of the corresponding pixels in two frames, F_{p-1} and F_p.

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

Logarithm is used to compensate the non-linearity of human visual system.

In a particular video content, for example, the scenarios we studied video telephony has typical head-and-shoulder scenes, higher TVM represents the better smoothness of the video. If the network condition is poor, packet loss can cause jerkiness, which is captured by TVM and leads to lowering of TVM values. However, if the loss is so severe that the whole frame(s) is/are lost, video will have freezing scenes. Freezing scenes cause higher TVM value which definitely does not indicate the good smoothness in this case. To make TVM always reflect the smoothness of a video, we propose to modify TVM to incorporate video freezing artifact, as follows:

$$\widehat{TVM} = TVM_p - \varepsilon \times \frac{F_{fl}}{F} \tag{2}$$

where (F_{fl}) is the number of full frame loss leading to video freezing. F is total number of frames in video and ε is a constant which weights the negative impact caused by video freezing. It changes for videos with different content and length. In our video

chat clips, we find that setting ε to 20 can effectively reflect the video freezing artifact in different scenarios.

It is arguable that lower TVM does not necessary mean the poor smoothness as lower TVM can be due to the fast moving scenes or scene changes. While we agree that fast moving scenes and scene changes can lead to smaller TVM value, in our video telephony scenarios, the network capacity does not change much during a particular session (which is also the case in our experiments). The fast-moving scenes and scenes changes that usually lead to higher sending rate will typically increase the chance of packet loss which in turn impact the smoothness in a negative way. Therefore, in a particular network scenario, i.e. within a session of the experiments, our modified TVM effectively indicates the smoothness of a video. We refer this modified TVM as Temporal Smoothness (TS) or simply Smoothness.

The prediction models in Section 5 (mapping objective metrics to subjective evaluations) and Section 6 (Skype and Vtok QoE prediction models) are developed in WEKA [30]. Using WEKA toolbox, we divide the data for the model into n = 10 folds, where, n - 1 folds are for supervised learning and one fold is used to test the model for errors. The errors obtained in a fold is added to the weights of nodes of next fold in the training set. This 10-fold cross validation is used to build a robust model.

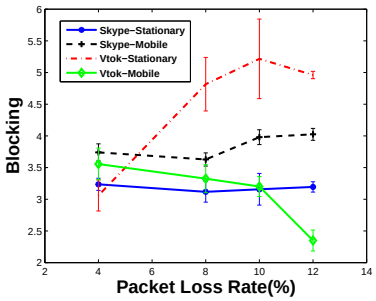
3 Experimental Results

With the mobile video telephony, end-users have the option to hold the device and move around during video conversation. This highly likely (*Mobile*) scenario of video telephony with end-user movement is compared to *Stationary* scenario, where end-users are only free to do any body movements with device being static. Our goal is to investigate all such networking conditions arising due to "mobile" video telephony and examine if they contribute to decrease/increase in end-user QoE.

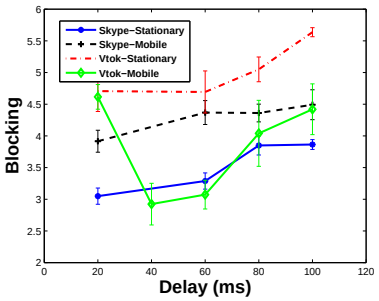
Video telephony application, being delay intolerant, primarily uses UDP protocol for data transmission. It is therefore, not possible to deduce actual loss or delay incurred by the video telephony packets from packet traces. However, each of these video telephony quality metrics performance is more-or-less intertwined with network-layer parameters. We carefully design controlled experiments (Section 2) to study the impact of each factor at a time while keeping others constant. We vary these network parameters at the network emulator, and take multiple traces in controlled experiments to study the impact on video quality. The *Delay*, *Packet Loss* and *Available Bandwidth* mentioned in the figures hereon, are the settings of network emulator. All the figures, henceforth, plot mean values of the y-axis metrics with 90% confidence interval with x-axis representing network QoS.

3.1 Spatial Impairments

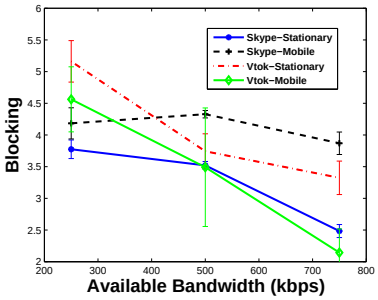
We repeat the experiments many times for a particular network setting and take arithmetic mean of blocking and blurring values over captured video frames as average blocking and blurring experienced by the end-user for that network setting. We, thus,



(a) Blocking vs Packet Loss



(b) Blocking vs Delay



(c) Blocking vs Available Bandwidth

Fig. 3 Blocking & Network Impairments

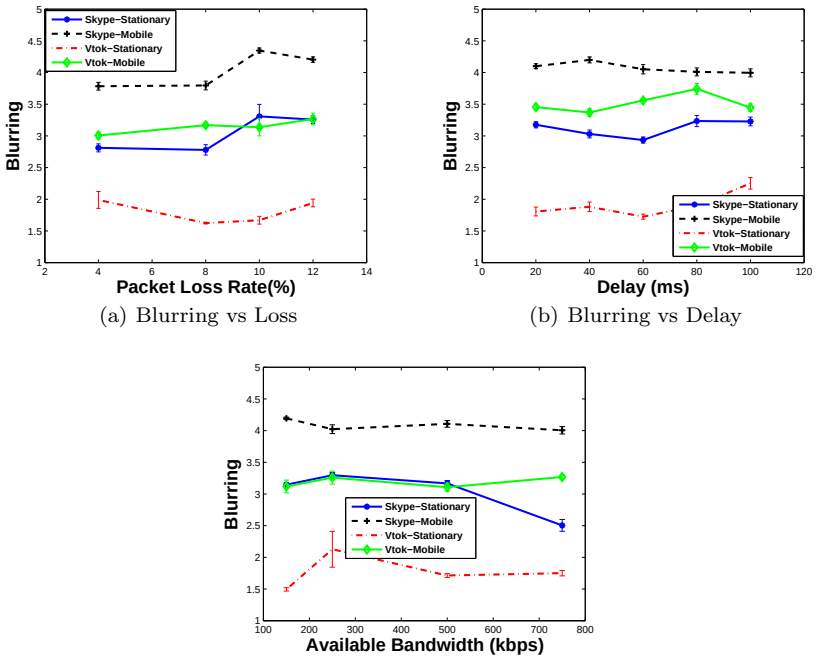
characterize the spatial quality of a video captured for a fixed network setting as -

$$Y_{avg}^{block} = \frac{\sum_{i=1}^{N} \frac{\sum_{k \in f} Y_i^{block}(k)}{|f|}}{N}, \tag{3}$$

$$Y_{avg}^{blur} = \frac{\sum_{i=1}^{N} \frac{\sum_{k \in f} Y_i^{blur}(k)}{(|f|)}}{N} \tag{4}$$

where $Y_i^{block}(k)$ and $Y_i^{blur}(k)$, are the blocking and blurring experienced by the k^{th} frame of the i^{th} video conversation trace respectively. N is the number of video conversation traces taken for a network setting. f indicates the set of such frame numbers of a video captured in each experiment. Y_{avg}^{block} and Y_{avg}^{blur}, the average blocking and average blurring respectively, are referred simply as *Blocking* and *Blurring* in the subsequent sections. Intuitively, since the packet loss in the network is bursty, the spatial impairments do not occur uniformly across all video frames. Moreover, videos captured have variable number of video frames. Hence, we consider 1000 frames of each recorded video having worst blocking and blurring artifacts to evaluate Y_{avg}^{block} and Y_{avg}^{blur}.

Packet Loss : We vary the packet loss at the emulator, from 4% to 12%. We observe from our experiments that with higher packet loss rates ($\geq$ 12%), Skype refuses to establish end-to-end video conversation session. The two-way delay in the network emulator is fixed to 20ms. The link bandwidth of the emulator is $80Mbps$, way above



(a) Blurring vs Loss

(b) Blurring vs Delay

(c) Blurring vs Available Bandwidth

Fig. 4 Blurring & Network Impairments

than the required bandwidth for the smooth functioning of mobile telephony application. The effect of network losses on spatial video quality are shown in Figure 3(a) and Figure 4(a).

Skype - The average blocking in Skype video remains unaffected in Stationary scenario where as for Mobile scenario, the average blocking artifacts increase at the loss rate of 8% and then remains constant. Unlike average blocking, average blurring increases at loss rate of 8% for both mobile and stationary scenarios. Thus, spatial impairments increase when the loss-rate is 8% or more. The blocking and blurring experienced by the *Mobile* cases is $18\% - 25\%$ more than the *Stationary* cases.

Vtok - Unlike Skype, blocking for *Mobile* Vtok is better than *Stationary* Vtok. The decrease in blocking artifacts at a high-loss rate $> 12\%$ for both cases may be due to the video frame loss. Increased frame loss implies the video has stalled and therefore no loss of full block information. Blurring in *Mobile* Vtok is more than *Stationary* cases. Mobile Vtok traces has unexpectedly lower blocking and we discuss the reasons later in Section 4.

Network Delay : Video telephony packets need to adhere to strict delay constraints of less than 150ms [10]. Increasing the bandwidth usage can be a straightforward solution for the applications that require low network delay jitter for their video data. But this can be very costly and inefficient in terms of network resource-provisioning and usage. In addition, if the delay is more than the required limitation, the video telephony application fails to initiate video conversation or the video chat application freezes.

The two-way propagation delay at the emulator is varied from $20ms$ to $100ms$. The impact of network delay on spatial metrics is depicted in Figure 3(b) & 4(b). The network emulator link bandwidth is unconstrained and the packet loss rate is set to 1%.

Skype - Blocking increases with the delay of $60ms$ and higher whereas blurring is negligibly effected by the increase in delay at the network emulator. It is worth noting that the *Mobile* and *Stationary* scenarios greatly differ in perceptual video quality. Blocking and blurring artifacts are 23% and 33% higher respectively, in the *Mobile* scenario compared to the *Stationary* scenario for the same network delay.

Vtok - As seen with increasing loss-rates, *Mobile* Vtok in comparison to *Stationary* Vtok performs better with respect to blocking artifacts but performs poorly in terms of blurring experienced. We examine Vtok's sending rate in the next section to understand these differences in spatial metrics. Similar to Skype, Vtok blurring do not change significantly with increased network delay.

Network Bandwidth Constraints : We change the two-way link bandwidth at the network emulator to $150kbps$, $250kbps$, $500kbps$ and $750kbps$. The packet loss rate is 1% and delay is $0ms$ at the network emulator. For link bandwidths lower than $150kbps$, Skype and Vtok fails to establish connection. Intuitively, this is because packet loss will occur due to link constraints.

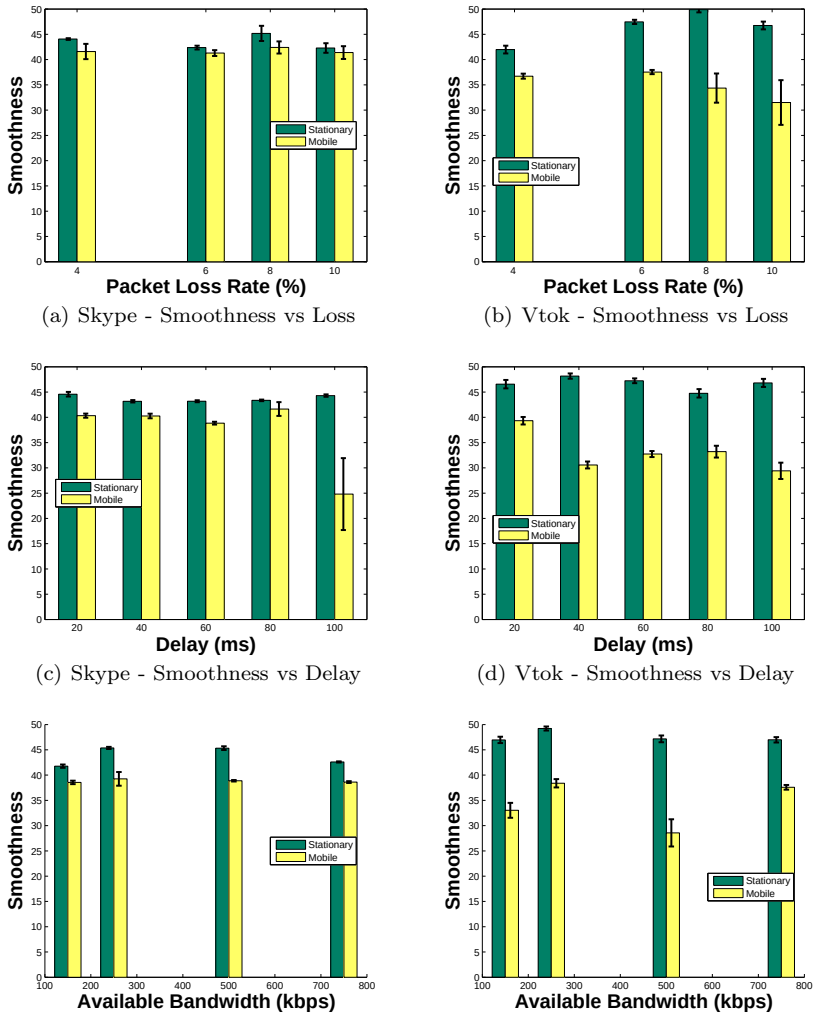
Skype - As the available bandwidth is increased at the emulator, blocking artifacts for *Stationary* scenario reduces by $\approx 17\%$ although there is only a marginal decrease in blurring impairments (Figure 3(c) & 4(c)). For *Mobile* scenario, the decrease in blocking and blurring are very less with increasing available bandwidth. The *Stationary* scenario has better perceptual video quality both in terms of blocking and blurring as the bandwidth constraints are relieved. For a network bandwidth of $750kbps$, *Mobile* Skype spatial metrics perform $\approx 38\%$ poorly compared to *Stationary* Skype.

Vtok - *Mobile* and *Stationary* Vtok show similar characteristics in terms of blocking and blurring artifacts as seen in previous two cases of increasing network loss-rate and delay. *Mobile* Vtok experiences $\approx 43\%$ decrease in blocking artifacts in-comparison to *Stationary* Vtok at link rate of 750kbps (Figure 3(c)). *Stationary* and *Mobile* Vtok have similar blurring performances for bandwidth constraint till $500kbps$. But, the *Stationary* case experiences 33% decrease in blurring artifacts than *Mobile* Vtok at $750kbps$ link capacity. (Figure 4(c)).

3.2 Temporal Smoothness

Video stalling or freezing creates very bad experience for end-users. In our model such impairments are captured using the modified TVM metric, Smoothness. The Smoothness values, as mentioned in Section 2.2, decreases when the temporal smoothness of a video deteriorates. In other words, Smoothness is negatively correlated to temporal impairments. Although all the network QoS metrics have impacts on the video smoothness, a Smoothness value of larger than 40 is generally regarded as a good smoothness.

We observe from Figure 5, that Smoothness for Skype and Vtok consistently gives larger values in *Stationary* scenarios than for *Mobile* scenarios. The Smoothness values for Skype decrease substantially by $\approx 45\%$ (Figure 5(c)) and Vtok decreases by 33% (Figure 5(d)) for *Mobile* scenarios when the network delay is as large as $100ms$. Skype & Vtok receiver may have dropped a lot of late packets in such a large network delay



(a) Skype - Smoothness vs Loss

(b) Vtok - Smoothness vs Loss

(c) Skype - Smoothness vs Delay

(d) Vtok - Smoothness vs Delay

(e) Skype - Smoothness vs Available Bandwidth

(f) Vtok - Smoothness vs Available Bandwidth

Fig. 5 Impact on Temporal Variation Metric due to network impairments/constraints - Loss, Delay and Bandwidth

scenario. This causes whole frames loss, resulting in large variation of Smoothness values. Hence, network delay under end-user mobility leads to considerable reduction in temporal video quality for both Skype and Vtok.

With the threshold Smoothness being 40, the temporal quality deterioration on *Mobile* Skype and Vtok is 0% and 25% respectively, for loss rate as high as 10% (Figures 5(a) & 5(b)). Whereas, for least available bandwidth of 150kbps *Mobile* Vtok's and *Mobile* Skype's temporal smoothness is reduced by $\approx$ 20% and 5% respectively (Figures 5(f) & 5(e)). We therefore conclude, temporal smoothness in Skype videos is better than Vtok videos.

Table 1 Correlation Coefficient Quality Metrics and Movement.

	Cor_m	p-values	95% confidence interval
Skype Blurring	0.7624	0.0000	[0.6976 0.8148]
Vtok Blurring	0.7521	0.0000	[0.6650 0.8190]
Skype Blocking	0.3633	0.0000	[0.2322 0.4814]
Vtok Blocking	-0.3494	0.0001	[-0.4952 -0.1845]
Skype Smoothness	- 0.3184	0.0000	[-0.4361 -0.1899]
Vtok Smoothness	-0.7534	0.0000	[-0.8200 -0.6666]

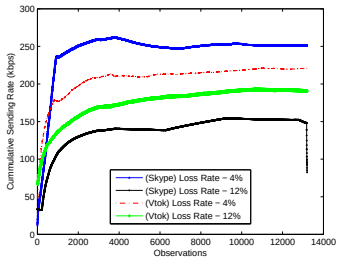


Fig. 6 A representative sending rate of Apps at different loss rates

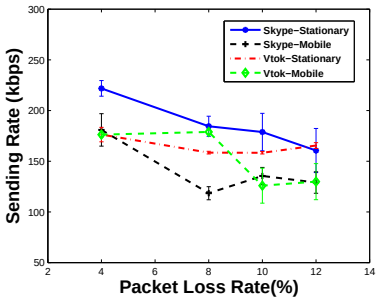
4 Analysis

Network QoS impacts perceived video quality as observed in experimental results. In addition, the perceived video quality differs consistently with end user being *Mobile* or *Stationary*. Table 1 gives significant correlation coefficients (Cor_m) of different video quality metrics with end-user movements. We consider, $Movement = 1$, for *Mobile* scenarios and $Movement = 0$ otherwise. i.e. the "Movement" is either ON or OFF, there is no middle case. From Table 1, we find that p-values are zero. The small p-values show that their correlation with the end-user brisk movements which we observed from the experiments is very reliable. The observations from Table 1 support our experiment results in Section 3. The perceived video blurring and temporal smoothness across both applications degrades consistently in video telephony due to end-user movements. Vtok blocking shows opposite trend in comparison to Skype blocking. We explain in details, the reasons for these observations later in this section.

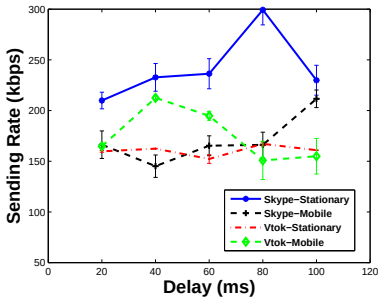
The applications being closed source, we examine the available application-level information to understand their behaviors. A video telephony application session has variable bit rates, and adapts to the actual video content and network conditions. Video telephony applications employs rate control algorithms ([9], [16], [31], [23] etc.) depending on the congestion in the network. Figure 6 shows sending rate of applications at different loss rates for arbitrary traces validating the use of rate control algorithm by these applications. To the best of our knowledge, none of the work studies these applications sending rate under *Mobile* scenario.

4.1 Skype Sending Rate

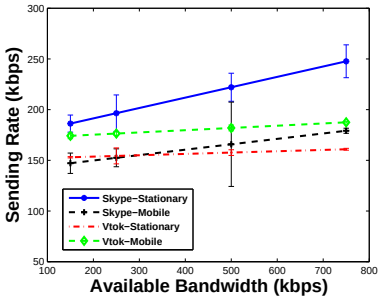
We observe from Figure 7(a), that *Stationary* Skype changes its sending rate as the loss rate in the network increases. When the loss rate is 12%, *Stationary* Skype's



(a) Loss vs Sending Rate(kbps)

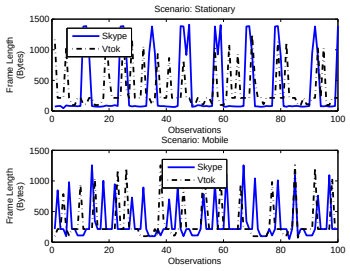


(b) Delay vs Sending Rate(kbps)

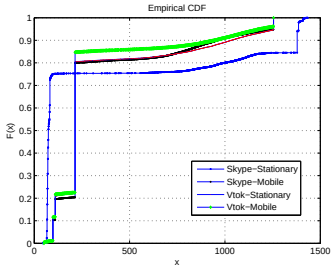


(c) Available Bandwidth vs Sending Rate(kbps)

Fig. 7 Skype Sending Rate with network impairments/constraints - Loss, Delay and Bandwidth



(a) A representative trace of applications frame length



(b) Cumulative Distribution Function of frame length

Fig. 8 Application MAC Layer sending rates

sending rate is more or less constant around 120*kbps*, the minimal data-rate which is required to carry out a smartphone video telephony conversation. Intuitively, we conclude, *Stationary* Skype adapts its sending rate not only depending upon the network losses but also depending on the end-device requirements. We observe from Figure 7(b), *Stationary* Skype does not adapt its sending rate with increasing network delay whereas, it increases its sending rate with increased network bandwidth (Figure

7(c)). The inconsistent result at 80ms delay, can be due to Skype or other confounding factors.

We find the rate-control behavior of Skype under *Mobile* conditions have similar trends as in *Stationary* conditions (Figure 7), i.e., Skype's sending rate decreases with increased network losses and with decreased network bandwidth, but the average sending rate in *Stationary* cases is always 50 kbps more than the *Mobile* cases for same networking conditions.

OBSERVATION (1) : Stationary Skype performs better than Mobile Skype for all cases

Skype incurs more losses in *Mobile* scenario than *Stationary* scenario. The data losses can be due to the congestion losses, random losses and video coding losses. Congestion losses are due to the packets lost in end-to-end transit over the Internet and will impact all the scenarios. Random losses are caused by the wireless access-link. We did repeated measurements of the access-link losses with an end-user doing brisk movements in an area and an end-user being static. In both the cases, we found that the access-link packet losses in both scenarios differ negligibly. Moreover, all the experiments were carried out in a dedicated, unsaturated indoor wireless environment. The reason for such negligible difference in packet losses at access-link can be due to Medium Access Control (MAC) level re-transmissions.

Investigating MAC frames - Skype uses lower bit-rates for *Mobile* scenarios where it sends more MAC frames of smaller length (Figures 8(a) & 8(b)) compared to larger frame sizes in *Stationary* cases. The smaller MAC frames are likely because of small sized application packets. A smaller Maximum Transmission Unit (MTU) reduces latency by reducing MAC-level re-transmissions but adversely effects data transmission rate due to the increased header overhead in the application packets.

Moreover, *Mobile* video data is changing faster. When user walks, there is a relative motion between camera and background and some jerky motion between camera and user. This leads to high temporal content being created and may require reduced latency in end-to-end transmission. Skype, though, reduces the MTU per packet, but it does not increase its sending rate to encounter the increased overhead for *Mobile* scenarios, therefore, impacting the performance.

Video coding losses can also lead to such large disparity in performance for both cases. The *Mobile* video data are highly correlated in consecutive video frames [15] in comparison to *Stationary* cases. Loss of such correlated data would lead to poor video decoding at the receiver's end. This can also lead to probable increase in video artifacts in case of *Mobile* scenarios.

4.2 Vtok Sending Rate

Vtok's sending rate with increasing loss-rate for both mobile and stationary cases (Figure 7(a)) are similar till loss-rate of 8%. But sending rate of *Mobile* Vtok drops 35% for loss rate $\geq$ 10%. As a result we see increase in blocking in Figure 3(a) with loss rate 8% - 10%. The spatial impairments are unable to capture the video quality at loss rate of 12% because of frequent video freeze. The video freeze is also observed in Figure 5(b) where Smoothness values of *Mobile* Vtok is 25% less than required Smoothness threshold.

With the network delay setting of 40ms, Vtok increases its sending rate by 50kbps (Figure 7(b)) for mobile scenarios to cater the motion component in the videos. But for network delay > 40ms, Vtok reduces its sending rate to avoid increase in network congestion. At 80 ms and 100 ms network delays, the sending rate of both *Mobile* and *Stationary* scenario cases are almost same. The bandwidth constraints have little impact on Vtok's sending rate. This, explains similarity in blocking and blurring characteristics exhibited in both scenarios (Figures 3(c) & 4(c)).

OBSERVATION (2) : Mobile Vtok blocking is negatively correlated and blurring is positively correlated to end-user movement
Blocking and blurring artifacts in videos are highly correlated when caused due to packet losses. However, in Vtok we observe that blocking and blurring artifacts show different behavior with end-user movement (Table 1). This is possible by changes in rate-distortion algorithm of codec. Moreover, we observe that sending rate of Vtok unusually increases in *Mobile* cases (Figure 7). This implies that Vtok codec has an aggressive rate-distortion algorithm, which leads to unusual response with changing networking conditions.

Another reason can be due to the video frame loss caused by increasing loss-rates. Though frame loss does not lead to increased block information but results in video freezing.

4.3 Skype vs. Vtok

OBSERVATION (3) : Blurring artifacts are lower in Vtok than Skype in general
In comparison to Skype, blurring artifacts in Vtok are less. Vtok's video-coding is more robust in terms of avoiding loss of high frequency components.

OBSERVATION (4) : Skype exhibits better Temporal Smoothness than Vtok
In a video, frame loss leads to video freezing. The better temporal smoothness in Skype can be a result of its rate-control algorithm and video coding techniques. Skype avoids video freezing in *Mobile* cases by reducing the sending rate. But in doing so, experiences spatial impairments. Whereas, Vtok increases its rate in *Mobile* cases to increase its spatial quality, and in doing so, results in video frame losses.

Hence, there is a trade-off between spatial and temporal impairments experienced and application sending rate and video coding. We next examine the subjective assessments of mobile telephony videos.

5 MOS Prediction Model

As discussed in Section 2.2, subjective video assessment is required to capture video quality perceived by HVS. In this section, we derive an MOS prediction model based on evaluated objective quality measurements of experimental data.

Complex objective measurements, such as PEVQ [20], SSIM [29], NTIA VQM [21] etc. have been proposed to approximate MOS. But all these metrics require original transmitted videos. i.e. they are full-reference (FR) metrics. The work in [2] proposed TVI to estimate MOS of videos. However, it still requires the sender to provide certain information of the original video, such as TVM. i.e. it is a reduced-reference (RR)

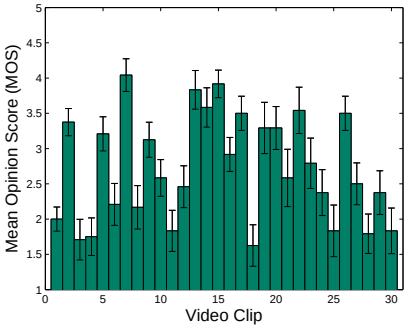


Fig. 9 Average MOS scores for selected video clips.

Table 2 Correlation Coefficient Quality Metrics and average MOS.

	Cor_{mos}	p-values	95% confidence interval
Blocking	0.1444	0.4464	[-0.2277 0.4797]
Blurring	-0.7184	0.0000	[-0.8569 -0.4831]
Smoothness	0.5231	0.0030	[0.2007 0.7433]

metric. Neither FR nor RR metric is suited for proprietary commercial video chat applications, such as Skype and Vtok, as the end-user to end-user nature does not allow us to control the sender to provide the information of the original video. We, therefore, need a metric that does not rely on the original video at all, i.e. a non-reference (NF) metric. In this section, we carry out subjective evaluation of videos of these apps to develop a NF QoE prediction model.

We conducted subjective evaluation on 30 selected test video clips from the pool of 312 recorded video conversations which include both static and mobile scenarios. The selected videos had different average blocking, blurring and Smoothness values so as to capture wide range of objective values for subjective evaluations. Each video has a duration of $\approx$ 30 seconds. Each video was presented one at a time and rated individually by 24 viewers conforming to the minimum number of viewers specified in [11]. The participants ages range from 22 to 40. During the test, an 11-inch LCD Monitor (Intel HD Graphics 4000) of resolution 1366x768 and 32-bit color pixel-depth was used. The video shown on the monitor were in the size which showing in the smartphone screen. i.e. Samsung Exhibit II's 400x800 3.7-inch screen. The videos are rated independently on 1-5 discrete point quality scale following ITU-T Recommendations [11]. The ratings of each video was averaged over all viewers to obtain a mean opinion score (MOS) [12].

Figure 9 shows average MOS for each video clip. It shows that we selected a set of videos with a variety in quality. MOS of the videos ranges from 1.5 to 4. The small 95% confidence intervals (the line segments above the bars in Figure 9) show that the individual subjects scores are in good agreement.

Next, we develop a model to map objective metrics to subjective evaluations to arrive at the video quality prediction models. The standard procedure to transform objective measures obtained from image quality metrics to predicted MOS scores is done via a nonlinear regression (for eg. [6]). We henceforth, use Support Vector Regression (SVR) to solve the problem of regression prediction. SVR has been used in wide range of applications - from prediction of air temperature to target localization in

wireless sensor networks. Unlike ANN (Artificial Neural Network), SVR is much less susceptible to over-fitting as it is based on the principle of structural risk minimization. Also, SVR always converges to a solution which is globally unique and optimal as it is framed as convex optimization problem. The basic idea of SVR is based on computation of a linear regression where input data are first mapped into an high-dimensional feature space using a fixed non-linear function and then a linear model is constructed in this feature space. The theory of SVR is given in Appendix A and the detailed information can be found in [3] & [24]. The function SMOReg in WEKA [30] implements SVR. We use the default settings of function SMOReg.

Thus, the predicted MOS, $\widehat{MOS}$, is an objective metric derived as

$$\widehat{MOS} = -0.686 * Y_{blur} + 0.0941 * Y_{tvm} + 3.7705 \tag{5}$$

where Y_{blur} and Y_{tvm} are the Blurring and Video Smoothness values of the test video clips.

The model has a Pearson correlation of 0.728 with low mean absolute error (MAE) 0.3803 and root mean square error (RMSE) 0.5233 values. We use this model (Equation 5) to obtain a single video quality metric, $\widehat{MOS}$, for each video telephony session.

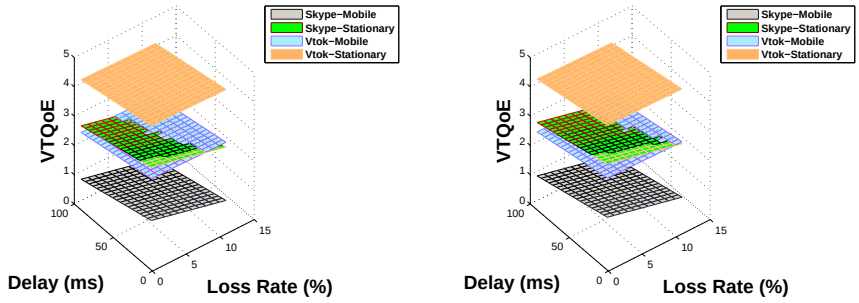
The model (Equation 5) for MOS prediction shows no dependency on blocking metric. On evaluating the Pearson Correlation, Cor_{mos} of Blocking, Blurring and Smoothness values with respect to average MOS scores of the test videos (Table 2), we find that blocking artifacts are unable to capture the subjective video quality (i.e. MOS) with high correlation. On the contrary, the blurring artifacts correlates negatively to MOS scores significantly with zero p-value. As higher Temporal Smoothness indicates better temporal video quality, the positive correlation with average MOS values with very low p-values can nonetheless be related to the subjective quality.

6 VTQoE: QoE for Video Telephony

As seen in Section 4, application's performance is currently impacted by video coding losses employed to cater the end-user motion in a video telephony session. Application sending rate is thus, dependent on Network QoS and end-user movements. Discarding the dependent variables, we obtain for each application, the relationship of $\widehat{MOS}$ with Network QoS. Such models give insight to network operators and application providers to maintain end-user QoE. We use SVR to derive these models.

The MOS prediction for Skype with correlation coefficient of 0.7691, MAE = 0.2305 and RMSE = 0.29 is given as

$$\begin{aligned} \widehat{SMOS}_n = & -0.3643 * M \\ & -0.1447 * \tilde{D} \\ & -0.132 * \tilde{L} \\ & +0.0843 * \tilde{B} \\ & +0.6493 \end{aligned} \tag{6}$$



(a) VTQoE for fixed Bandwidth = 150 kbps (b) VTQoE for fixed Bandwidth = 350 kbps

Fig. 10 VTQoE of Skype and Vtok

where as, MOS prediction for Vtok with correlation coefficient of 0.8784, MAE = 0.3772 and RMSE = 0.608 is given as

$$\begin{aligned}\widehat{VMOS}_n = &- 0.3606 * M \\ &- 0.1085 * \tilde{D} \\ &- 0.011 * \tilde{L} \\ &+ 0.0142 * \tilde{B} \\ &+ 0.9321 \end{aligned} \tag{7}$$

The normalized predicted MOS for Skype is represented as $\widehat{SMOS}_n$ and the normalized predicted MOS for Vtok is denoted as $\widehat{VMOS}_n$. M indicates end-user movement with $M \in \{1, 0\}$. $\tilde{D}$, $\tilde{L}$ and $\tilde{B}$ indicate normalized values of network delay, loss rate and bandwidth, respectively. We observe, from Equations 6 and 7, that MOS prediction model for mobile telephony applications depend on the end-user movement and network QoS. The end-users movement have significant impact on the perceptual video quality and should be considered in prediction models for accuracy. The video coding losses due to change in video content in *Mobile* cases in comparison to *Stationary* cases needs to be examined by application providers to maintain end-users QoE.

Figure 10 depicts the performance of both Skype and Vtok with increasing network loss rate and delay. The bandwidth for Figure 10(a) and Figure 10(b), are fixed to 150kbps and 350kbps respectively as the application providers as the current applications are required to operate within approximately, 128kbps - 350kbps. *Stationary* Vtok's performance is close to predicted $\widehat{MOS}$ of 4 whereas, *Mobile* Skype performs poorly in comparison to all other cases. This confirms our observations in previous sections.

7 Related Work

Smartphone apps have become very popular with billions of downloads from various app stores. There have been quite a few interesting works ([9], [22] and [7]) on characterization of smartphone apps. To the best of our knowledge, smartphone video telephony,

the new data hog in mobile apps, has not been addressed so far in terms of it's video quality in the literature.

There have been many attempts on estimating an application's video quality using deep inspection of video packets, without actually considering the perceived quality of received video. Moreover, this is not feasible for video telephony apps as the video packets are encrypted specific to application provider. Real-time video quality in IP networks studied in [26] introduces a new video quality metric, which is evaluated using only network statistics and basic codec configuration parameters obtained offline. The work mainly relies on underlying full-reference assessment, PSNR. However, it is impossible to have the original video for assessment at the receiver's side for smartphone video telephony apps.

There has been some work reported on studying Skype. In [4], authors evaluate the QoS level provided by Skype voice calls. Authors in [25] investigate QoS parameters and measure the QoE in terms of subjective assessment, but it also considers only the voice-part of the Skype application. Another work [16] investigates Skype video in order to study the rate control of Skype to match the unpredictable Internet bandwidth. The performance of four popular Instant Messenger (IM) clients - Skype, Windows Live Messenger, Eyebeam and X-Lite, focusing mainly on video-telephony part is analyzed in [19]. The authors in [23], compare perceptual voice quality of Skype and Google Talk.

The authors in [31] used ITU-T Recommendation G.1070 standardized opinion model for stationary Skype to study its video quality. The work uses a synthetic video with only head to shoulder newsreader movements for profiling Skype's behavior. The head-to-shoulder movements of end-users may not be the case with smartphones where the end-devices and end-users both may be in motion causing frequent background (video content) changes.

However, to the best of our knowledge, none of these works, study perceptual video quality of mobile video telephony apps and determine the impacting factors.

8 Conclusions

This work provides network characterization and perceptual evaluation of "mobile" video telephony applications. Our extensive experiments give us insight on applications performance and transmission mechanisms, and also helps us to draw meaningful conclusions. We point out the impact of sending rate on video quality. For an application provider, before tuning the sending rate of video telephony, the trade-off between spatial and temporal impairments are required to be thoroughly investigated. With a Pearson correlation of 0.728, we obtain MOS prediction model ($\widehat{MOS}$) for video telephony based on video blurring and temporal smoothness. We presented SVR-based models to estimate perceptual video quality from network parameters with maximum MAE = 0.3772 for objective $\widehat{MOS}$ ranging from 1 to 5.

As a future work, it is possible to track the exact motion information of end users using smartphone accelerometer sensor. By including this important piece of information, QoE prediction models based on Network QoS can be more accurate. This may enable network operators to get accurate feedback of end-user perceived video quality. Moreover, it will also be interesting to study other quality issues in video telephony such as audio quality assessment and audio-video synchronization.

References

1. Basak D, Pal S, Patranabis DC (2007) Support vector regression. Neural Information Processing, Letters and Reviews 11(3)
2. Chan AJ, Pande A, Baik E, Mohapatra P (2012) Temporal quality assessment for mobile videos. In: Proceedings of the ACM Mobicom '12
3. Chapelle O, Vapnik V (1999) Model selection for support vector machines. In Advances in Neural Information Processing Systems 12
4. Chen KT, Huang CY, Huang P, Lei CL (2006) Quantifying skype user satisfaction. In: Proceedings of the ACM SIGCOMM '06
5. CISCO (2011) Cisco visual networking index: Global mobile data traffic forecast update, 2011 - 2016. www.cisco.com
6. Cui L, Allen A (2008) An image quality metric based on corner, edge and symmetry maps. In: Proc. British Machine Vision Conf., Leeds, UK
7. Dobrian F, Sekar V, Awan A, Stoica I, Joseph D, Ganjam A, Zhan J, Zhang H (2011) Understanding the impact of video quality on user engagement. In: Proceedings of the ACM SIGCOMM 2011 conference
8. FFMPEG (2013) http://ffmpeg.org/. FFMPEG
9. Huang J, Xu Q, Tiwana B, Mao Z, Morley Z, Zhang M, Bahl P (2010) Anatomizing application performance differences on smartphones. In: Proceedings of Mobisys '10
10. ITU (1993) -T recommendation G.114. Tech. rep., International Telecommunication Union
11. ITU (2002) Bt-500-11: Methodology for subjective assessment of the quality of television picture, international telecommunication union. Tech. rep., International Telecommunication Union
12. ITU (2008) Subjective video quality assessment methods for multimedia applications, international telecommunications union, itu-t. rec. p.910. Tech. rep., International Telecommunication Union
13. Jana S, Pande A, Chan A, Mohapatra P (2013) Mobile video chat: Issues and challenges. IEEE Communications Magazine
14. Jana S, Pande A, Chan A, Mohapatra P (2013) Network characterization and perceptual evaluation of skype mobile videos. In: Proc. of ICCCN
15. Jana S, Baik E, Pande A, Mohapatra P (2014) Improving mobile video telephony. In: Proc. of IEEE SECON
16. L D Cicco VP S Mascolo (2008) Skype video responsiveness to bandwidth variations. In: NOSSDAV
17. Marichal X, Ma WY, Zhang H (1999) Blur determination in the compressed domain using DCT information. In: Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, vol 2
18. NETEM (2013) www.linuxfoundation.org. Netem
19. O Boyaci HS A G Forte (2009) Performance of video-chat applications under congestion. In: IEEE International Symposium on Multimedia
20. OPTICOM (2005) Pevq advanced perceptual evaluation of video quality. OPTICOM GmbH, Germany: PEVQ Whitepaper
21. Pinson M, Wolf S (2004) A new standardized method for objectively measuring video quality. Broadcasting, IEEE Transactions on 50(3):312–322

22. Qian F, Wang Z, Gerber A, Mao Z, Sen S, Spatscheck O (2011) Profiling resource usage for mobile applications: a cross-layer approach. In: Proceedings of MobiSys '11
23. Sat B, Wah B (2006) Analysis and Evaluation of the Skype and Google-Talk VoIP Systems. In: Multimedia and Expo, 2006 IEEE International Conference on
24. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Statistics and Computing 14(3)
25. T Hofeld AB (2007) Analysis of Skype VoIP traffic in UMTS: End-to-end QoS and QoE measurements. Computer Networks
26. Tao S, Apostolopoulos J, Guérin R (2008) Real-time monitoring of video quality in ip networks. IEEE/ACM Trans Netw 16(5)
27. Wang Z, Bovik AC (2006) Modern Image Quality Assessment. Synthesis Lectures on Image, Video, and Multimedia Processing, Morgan & Claypool Publishers
28. Wang Z, Bovik A, Evan B (2000) Blind measurement of blocking artifacts in images. In: Image Processing, 2000. Proceedings. 2000 International Conference on, vol 3
29. Wang Z, Lu L, Bovik AC (2004) Video quality assessment based on structural distortion measurement. Signal Processing: Image Communication 19(2):121 – 132
30. WEKA (2013) http://www.cs.waikato.ac.nz/ml/weka/. WEKA
31. Zhang X, Xu Y, Hu H, Liu Y, Guo Z, Wang Y (2012) Profiling skype video calls: Rate control and video quality. In: INFOCOM

A APPENDIX : Support Vector Regression

Support Vector Regression (SVR), is a machine learning tool proposed in [3]. We discuss the linear case followed by non-linear SVR algorithm.

Suppose the training data set is as following

$$S = \{(\mathbf{x_i}, y_i) | i = 1, 2, 3, ..., m\} \tag{8}$$

where real-valued inputs $\mathbf{x_i} \in \Re^n$, and target $y \in \Re$. The objective function is to find function f that returns the best fit i.e. $f : \Re^n \longrightarrow \Re$ The linear regression function f is given as

$$f(\mathbf{x}) =< \omega, \mathbf{x} > +b \tag{9}$$

where $b \in \Re$ and $\omega \in \Re^n$. To avoid over-fitting, a regularization term is introduced to have small ω and can be formulated as the convex optimization problem minimizing the euclidean norm i.e.

$$\begin{aligned} &\text{minimize} && \frac{1}{2} \parallel \omega \parallel^2 \\ &\text{subject to} && y_i - < \omega, \mathbf{x} > -b \leq \varepsilon \\ & && -y_i + < \omega, \mathbf{x} > +b \leq \varepsilon \end{aligned}$$

The convex optimization problem may not be feasible if the errors are more than ε. Thus, any point outside the ε region contributes to the cost of the function.

SVR introduces slack variables, ξ_i and ξ_i^* to cope up with not feasible constraints. The convex optimization can be defined as

$$\text{minimize} \quad \frac{1}{2} \parallel \omega \parallel^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) \tag{10}$$

$$\text{subject to} \quad \xi_i, \xi_i^* \geq 0 \tag{11}$$

$$y_i - < \omega, \mathbf{x} > -b \leq \varepsilon \tag{12}$$

$$-y_i + < \omega, \mathbf{x} > +b \leq \varepsilon \tag{13}$$

where C is a constant known as penalty factor. It consists the trade-off between smaller ω values and ε-insensitive loss function given as

$$|\xi|_\varepsilon = \begin{cases} 0 & : \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & : \text{otherwise} \end{cases}$$

The Equation 13 is the Primal form and handles inequality constraints directly. The dual form obtained by constructing a Lagrange function and taking partial derivative w.r.t. primal variables is given as-

$$\text{maximize} \quad \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < \mathbf{x}_i, \mathbf{x}_j > \tag{14}$$

$$-\varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} y_i(\alpha_i - \alpha_i^*) \tag{15}$$

$$\text{subject to} \quad \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0 \tag{16}$$

$$\alpha_i, \alpha_i^* \in [0, C] \tag{17}$$

where α_i, α_i^* are Lagrange multipliers. The steps for partial derivative can be looked upon in [1]. Interestingly, the partial derivative also gives following equation :

$$\omega = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\mathbf{x_i}. \tag{18}$$

Hence, the SVR linear regression reduces to

$$f(\mathbf{x}) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) < \mathbf{x_i}, \mathbf{x} > +b \tag{19}$$

The Equations 9-19 discussed so far are simple cases when SVR algorithm is used for a linear function. SV algorithm can be made non-linear by simply substituting every instance of $\mathbf{x}$ with $\Phi(\mathbf{x})$. The approach becomes infeasible when $\mathbf{x}$ is mapped to higher-dimensions. Using kernel method, explicit substitution of $\mathbf{x}$ with $\Phi(\mathbf{x})$ is avoided. The kernel function is given as -

$$< \Phi(\mathbf{x_i}), \Phi(\mathbf{x}) >= K(\mathbf{x_i}, \mathbf{x}) \tag{20}$$

The commonly used kernel functions are polynomial kernels: $K(x, y) = (x^T y + 1)^d$ and radial basis function (RBF) kernels: $K(x, y) = exp(\frac{-||x-y||^2}{2\sigma^2})$. Hence, from Equations 19 and 20, the solution function can be written as

$$f(\mathbf{x}) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(\mathbf{x_i}, \mathbf{x}) + b \tag{21}$$