QoE Estimation Models for Tele-immersive Applications

Narasimha Raghavan Veeraragavan*, Hein Meling†, and Roman Vitenberg*
*Department of Informatics, University of Oslo, Norway
†Department of Electrical Engineering and Computer Science, University of Stavanger, Norway
Email: raghavan@ifi.uio.no, hein.meling@uis.no, romanvi@ifi.uio.no

***Abstract*—Tele-immersive applications, which are regarded as the next generation distributed multimedia applications, are highly interactive and aims to offer an immersive experience to its users. A challenge with such applications is to provide the best possible quality of experience (QoE) under changing conditions. In particular, it would be highly desirable to be able to predict the QoE perceived by users in response to adaptation, prior to actual deployment. However, there are no QoE prediction models for tele-immersive applications. Instead QoE is evaluated after deployment using either objective or subjective assessment techniques. Unfortunately, objective assessment lacks the accuracy of human perception. At the same time, subjective assessment requires human-provided ratings of the applications, which is time consuming and thus not cost-effective.**

In this paper, we propose QoE prediction models that will accurately predict the user-perceived QoE of a tele-immersive conferencing application. The proposed models are cost-effective and lend themselves to fast evaluation cycles, because the models does not involve human-provided ratings. We validate our models using results from subjective assessment experiments. The models can be used for real-time monitoring of user-perceived QoE, in addition to designing QoE-driven adaptation for tele-immersive applications.

I. Introduction

Next generation multimedia applications, also called tele-immersive applications, will provide new media through which users at various locations can interact with each other in 3D virtual environments. Examples of these applications include World Opera [1] and collaborative gaming [2]. These 3D virtual environments aim to make geographically distributed users feel as if they are physically co-located by enabling tight interaction among users. A key requirement for these applications is to provide the best possible QoE to their users.

In order to meet this requirement, it is important to develop effective methods for assessing QoE. Typically, QoE is considered as a subjective measure to describe a real user experience. In [3] it was shown that an effective way to measure the QoE for tele-immersive applications is by means of a subjective assessment method called Comparative Mean Opinion Score (CMOS). In this method, a group of participants are asked to compare two media samples with different configurations. Each participant needs to provide a rating that indicates how the two samples compare in terms of quality.

Measuring QoE by means of subjective assessment has several problems. First of all, applications must be deployed prior to conducting a subjective assessment study. Thus, if the results of the study reveal that significant changes to the application architecture are necessary, significant cost and time overruns can ensue. Second, conducting subjective assessment typically involves a group of paid participants who must be trained to correctly rate the application. It also requires collecting and computing the final rating for the application. Besides, every time the application is updated, the subjective assessment must be repeated. Above all, the QoE is typically affected by numerous QoS metrics. The number of possible combinations of values for these metrics is subject to a combinatorial explosion. Therefore, complete coverage cannot realistically be accomplished via subjective assessment.

We propose to solve the above problems by complementing subjective assessment with an online QoE estimation method. In the proposed method, the subjective assessment needs to be performed only once to produce a partial, yet representative set of QoE ratings. From this point on, a much more efficient QoE estimation method is invoked. The goal of this method is to compute the estimation of perceived QoE for the combination of QoS values not covered by the subjective assessment. The computation can be done in real time and without human in the loop. The estimated value for the QoE is very close to the value that would have been produced by the average human observer if a subjective assessment experiment had been invoked for the new combination of values.

To this end, we propose QoE estimation models, which capture the relationship between the important QoS metrics and the QoE of the application through complex mathematical functions. The QoE estimation models that we propose in this paper are based on Neural Networks [4]. Neural networks were chosen for their ability to emulate non-trivial unknown functions. Our proposed estimation model captures the complex relationship between 4-dimensional objective quality metrics (video frame rate, audio signal quality, synchronization quality, and interactivity) and the QoE of tele-immersive applications expressed in terms of CMOS.

In addition to solving the problems of subjective assessment methods, QoE estimation models provide many benefits to designers of tele-immersive applications. One important benefit is to help designers in effective resource planning and allocation by endowing them with additional knowledge. For example, whether or not new hardware components such as additional cameras or microphones needs to be introduced to the existing architecture to meet the target QoE can be decided

by estimating the QoE of the current architecture. Similarly, the models can aid with estimating the amount of bandwidth that needs to be allocated to meet the expected QoE.

Another important benefit is that QoE estimation models can be used as a basis for real-time QoE monitoring of tele-immersive applications. For example, with the help of QoE real-time monitoring, system developers can detect a change in the QoE value and thus identify the responsible QoS metric. Accordingly, developers can use adaptation and dependability strategies to handle the situation. Consider a scenario, where the QoE value drops below a certain threshold due to a change in the video frame rate. Then dynamic strategies can be designed to adapt the video frame rate in order to improve the QoE value. Moreover, if a drop in QoE value can be traced to a network or component failure, fault tolerant backup mechanisms can be invoked.

In order to validate our models, we compared the results of our models with the results of the subjective assessment experiments conducted in [3] for a conferencing application. The results show that our models can mimic the subjective assessment results with good accuracy. Furthermore, we produce the unknown equations that capture the relationship between the QoS parameters and QoE parameters through our models.

II. Tele-immersive Applications

Next generation distributed interactive multimedia applications [1], [2], [3], [5] have started to emerge. In order to provide high interactivity and an immersive experience to the users, these applications deploy a large sets of hardware components such as camera, microphone, and sensor arrays, high bandwidth networks, etc. at multiple locations.

With the help of these components, multiple streams such as high quality video, audio, and sensor streams must be synchronized and sent to remote sites. From the sender's viewpoint, these generated streams collectively represent the real-world data. At the receiving sites, the sender's real-world data are viewed as virtual data. The received virtual data along with the local real-world data create a mixed-reality medium.

The system operation consists of the following five phases shown in Figure 1: initialization, capturing, processing, streaming, and rendering. During the *initialization phase*, all client-side technical components such as cameras and microphone arrays receive activation signals. In the *capturing phase*, components such as cameras and microphones start generating the streams. In the *processing phase*, all generated streams are processed to remove noise. Additionally, video streams are encoded to reduce their size, timestamped, and processed using computer vision techniques for aesthetic reasons. However, in many tele-immersive applications, the processing phase is not used for latency reasons. In the *streaming phase*, the streams are sent and received by the remote sites. In the *rendering phase*, the received streams are processed (e.g., decoded), synchronized based on their timestamps and then rendered to the virtual-world. For a detailed explanation of these phases, see [1].

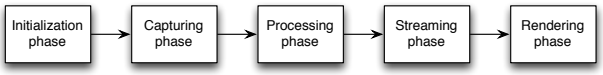


Fig. 1: Phases of operation in tele-immersive applications.

These phases help tele-immersive applications offer an experience similar to a face-to-face interaction. Researchers argue that these applications should be evaluated from the human-centric perspectives, rather than system-centric perspectives [6]. Human-centric perspectives mean that the entire life-cycle of the system should bear human focus. Accordingly, subjective assessment techniques are being devised by the researchers to evaluate the QoE perceived by users [3].

As we mentioned in Section I, these techniques suffer from relatively high cost and long completion duration. Thus, they can only be used for offline evaluation. Yet, it is beneficial for these applications to perform an online QoE evaluation, e.g., for the purpose of proactive identification and troubleshooting of performance bottlenecks. However, to reduce time and costs of repeated QoE evaluation, we need online evaluation models that can mimic the results of subjective assessment techniques.

We consider a *conferencing* tele-immersive application, for which we propose QoE estimation models. An example deployment of this application is described in [3], in which the authors focus on a social conversational scenario (CONV). The goal of the conferencing application is to provide geographically distributed users, with an experience that makes them feel as if they are talking to people in the same room. In this application scenario, audio intelligibility of the conversation is considered more important than video quality. For video, typically, only the lips would move, whereas the body remains stationary. For details about the experiments, see [3].

III. Background

In this section, we provide a short introduction to the neural network modeling technique and existing methods for subjective QoE assessment.

A. Neural Network Modeling

Artificial neural networks are computational models that help in understanding the complex relationships between the numerical data inputs and outputs. For example, given a set of input data $x = \{x_1, x_2, \cdots, x_n\}$ and a set of output data $y = \{y_1, y_2, \cdots, y_n\}$ that can be produced by an unknown function $y = f(x)$, the computational models help in obtaining a reasonable estimation of the function $f(x)$. To derive the estimation, the following steps must be performed. First, an appropriate computational model suitable for the given input and output data sets must be identified and its structure defined. Second, the computational model needs to be trained by using sufficiently representative data. Finally, the computational model must be tested with the same and also relevant sources of input data, to validate whether the derived estimation is of acceptable quality.

The specific type of neural networks utilized in the proposed models in Section IV is called Feed Forward Neural Networks

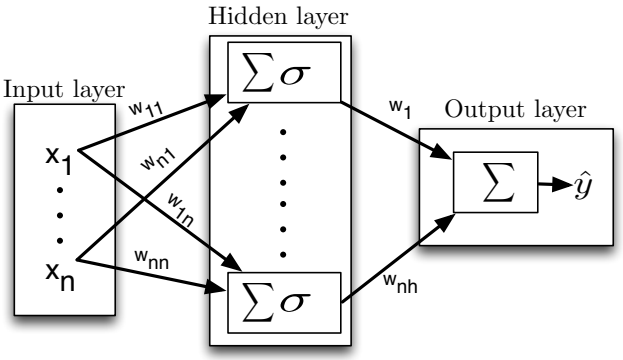


Fig. 2: Feed Forward Neural Network Architecture.

(FF-NN) [4]. The general structure of FF-NN includes three layers: input, hidden, and output layers as shown the Figure 2. Its working principle is as follows. The input layer consists of inputs $(x_1, \cdots, x_n)$ to the model. Each input (x_j) to the model is transferred to each computational unit in the hidden layer, along with an adjustable weight parameter (w_j). After receiving the inputs with weights, each computational unit calculates a weighed sum, which then passes a non-linear activation function (σ) called a neuron or computational function. In short, each computational unit in the hidden layer performs the following function: $\sigma(\sum_{j=1}^{n} w_j x_j)$. Furthermore, the output of each computational unit in the hidden layer is passed to the computational unit in the output layer, which computes a weighed sum and produces the network output $(\hat{y})$, as shown in Figure 2.

Defining the structure of the FF-NN refers to the specification of these three layers. That is, defining the number of inputs and outputs, as well as their values, and the number of computational units in the hidden layer. The appropriate size of the hidden layer for a given input-output pair will only be known after training network. Until the result of the training session matches the expected output, the size of the hidden layer is changed by trial-and-error.

The training process refers to tuning the weights, so that the FF-NN model approximates the unknown function producing the input-output pairs. The training process will continue until the output produced by the model is close enough to the target output. After the model is trained, it is validated by providing the inputs from the same source that is used to train the model. If the model produces the expected output, then we obtain an approximation function that describes the input-output pair relationships.

B. Existing Methods for Subjective QoE Assessment

The procedure for subjective QoE assessment starts with a number of preparatory steps. First, the set of parameters that have impact on the perceived QoE of the application is identified. For each identified parameter, a discrete set of reasonable values is chosen. Each combination of the values, one value for each parameter, corresponds to a configuration of the system. The set of all combinations can be represented as a Cartesian product. Since the number of possible combinations is typically too high, a reasonable subset of configurations is selected either randomly or by representing the extremes on the range of values.

Then, the actual subjective assessment experiment is performed. In this experiment, the selected configurations are used as the settings for the application. For each setting, a trained group of people is asked to provide ratings based on the perceived QoE. The average of the ratings is taken to represent the QoE perceived by the users for those particular settings.

In [3], the authors conducted the above steps including subjective assessment for two distinct tele-immersive applications, conferencing and collaborative gaming. They identified the following parameters as highly influential for the perceived QoE of the users in tele-immersive applications: *video quality, audio quality, synchronization quality*, and *interactivity quality*. Although, there are many factors that determine the video quality, such as video frame, encoding rate, and spatial resolution, only the video frame rate is considered in [3]. A larger value for the video frame rate results in better smoothness in the video motion. The *Perceptual Evaluation of Speech Quality* (PESQ) metric defined in ITU-T P.862 [7], is used to represent the audio quality. Accordingly, a larger value of PESQ corresponds to greater audio intelligibility. The synchronization quality is represented by the difference between the end-to-end video and end-to-end audio delays. If the difference is less than zero, then the video transfer is ahead of the audio transfer, and vice versa. The interactivity quality for conferencing application is the summation of the round-trip end-to-end delays between the users with human response delays. For collaborative gaming application, the interactivity quality is only affected by the bidirectional end-to-end delay of the media streams.

The combined impact of these parameters influence the QoE perceived by the users in tele-immersive applications. The discrete values chosen for each of these parameters in [3] are shown in Table I. In [3] various combinations of these parameter values were used in several different configurations for conducting subjective assessment experiments in conferencing and collaborative gaming applications. In this paper, we restrict our scope to only one application due to the page limits.

In [3], 19 participants of average age 26 were used to provide ratings for each configuration of the experiments based on the comparative category rating scales [8]. Accordingly, the participants are first shown the optimal configuration of the applications (bold parameter values in Table I) that provides the best possible quality and then shown the configurations with degraded quality. The participants are asked to provide a rating from the score set {3, 2, 1, 0, -1, -2, -3}. The rating indicates that the quality of the first sample is {*much better, better, slightly better, same, slightly worse, worse, much worse*}, than that of second sample. From this, the average of the users voting for each configuration is computed, referred to as CMOS, as defined in ITU-T P.910 [8]. For more details about the experiments, see [3].

TABLE I: Parameter values used in [3]. Optimal value in **bold**.

Metrics	Discrete values
Video frame rate (x_v) [fps]	2.5, 5, 7.5, 10, 12.5, 15, 17.5, **20**
Audio (x_a) [PESQ]	2.0, **4.0**
Synchronization (x_s) [ms]	**0**, $\pm$75, $\pm$150, $\pm$225
CONV Interactivity (x_d) [s]	**0.8**, 1, 1.2, 1.4, 1.6, 1.8, 2, 2.2, 2.4, 2.6

IV. QoE Estimation based on Neural Networks

In this section, we propose a procedure for constructing a QoE estimation model for a tele-immersive conferencing application. The QoE estimation model is based on FF-NN as described in Section III-A.

We assume that a subjective assessment experiment, as described in Section III-B, has initially been conducted to produce a suitable dataset. The subjective assessment is typically incapable of producing ratings for all possible combinations of input parameter values. This is because of the combinatorial explosion in the number of such combinations. The goal of the proposed models is therefore to compute an estimate of the perceived QoE for those combinations not covered by the subjective assessment. The requirement is that the computation should be performed in real time and without a human in the loop. The estimated QoE value should be very close to the value that would have been produced by the average human observer if a subjective assessment experiment had been invoked for the new combination of values.

The procedure for constructing a QoE estimation model for a tele-immersive application is as follows:

1) The results of the subjective assessment experiments are used to create the input-output pairs. Each setting of the experiment is referred to as an input configuration, and the corresponding QoE value of the experiment is referred to as the output for this configuration.
2) Some of the created input-output pairs are used to train the FF-NN model as described in Section III-A.
3) Additional created input-output pairs are used to validate the model.
4) The trained and successfully validated FF-NN model is used as a QoE estimation model.

In the model we propose for the conferencing application, we set the size of the input layer to 4, representing the four objective quality metrics and the size of the output layer to 1, representing the CMOS. We found that 4 works well as the size of the hidden layer. The resulting architecture is shown in Figure 3.

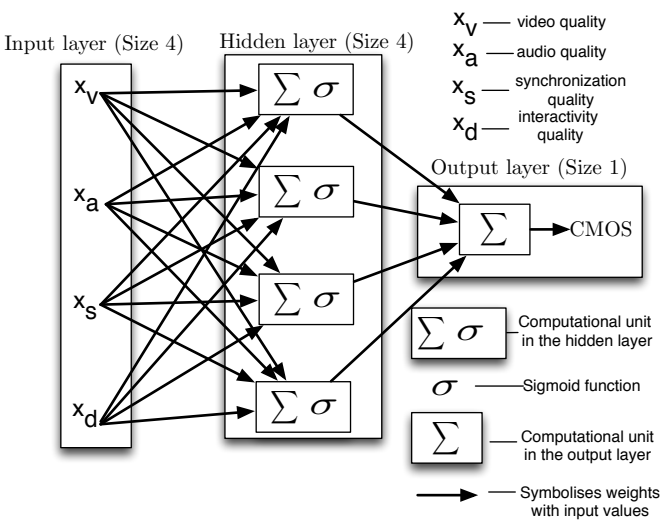


Fig. 3: FF-NN architecture for the conferencing application.

V. Evaluation

We use Mathematica [9] to construct and validate our QoE models discussed in Section IV. We have four objectives for our experiments. First, we *validate* the constructed model by verifying that it could mimic the results of the subjective assessment experiments. To this end, we plot the error distribution graphs of the constructed model against the trained and validation datasets. The error distribution graphs produce the differences between the model produced output, and the output of the trained and validation datasets. The training performance was measured against the validation dataset, in order to make sure that the model was generalized properly, and that no over training was affecting the results. Second, we aim to produce *extended output* that includes results not available in the subjective assessment datasets to show the strength of the developed models. Third, since the dataset derived from the subjective assessment is limited, the model is not able to compute the CMOS for all possible combinations of values of the four input parameters. To this end, we describe the *limits of the model*, namely, the input configurations for which the proposed model works. Finally, we want to produce *QoS-to-QoE mapping*, i.e., equations that map the QoS values to QoE values for the developed model. These equations can be used as utility functions for real-time QoE monitoring as discussed in Section VI.

A. Description of the Datasets

The model is trained and validated with the dataset derived from the subjective assessment results in [3] that describe the CMOS values (representing the user-perceived quality of experience) for various input configurations of a tele-conferencing application. As mentioned above, the four input QoS parameters are video quality, audio quality, synchronization, and delay.

The dataset in [3] contains a total 41 input configurations. Our first task is to analyze these configurations. We classify them into six categories shown in Table II. In the first category, the audio quality, synchronization, and delay are set to the optimal values, while we let the video quality vary over the values shown in Table II, resulting in a total of 8 different configurations. In the third category, the delay is set to the optimal value, the synchronization is set to a fixed realistic suboptimal value (150ms), while the video and audio quality varies with the values shown in Table II. Likewise the characterization for the other four categories is presented.

TABLE II: Experiment configurations derived from [3]. The table shows the set of values for x_v, x_a, x_s, x_d for the different configurations. The optimal value for each parameter is shown in **bold**. Units are shown in square brackets [].

# of configurations	Video (x_v) [fps]	Audio (x_a) [PESQ]	Sync (x_s) [ms]	Delay (x_d) [s]
8	{2.5, 5, 7.5, 10, 12.5, 15, 17.5, **20**}	**4**	**0**	**0.8**
7	**20**	**4**	{**0**, ±75, ±150, ±225}	**0.8**
8	{5, 10, 15, **20**}	{2, **4**}	150	**0.8**
9	**20**	**4**	**0**	{1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6}
6	5	{2, **4**}	**0**	{1.2, 1.6, 2.0}
3	**20**	2	**0**	{1.2, 1.6, 2.0}

TABLE III: Combined behavior of input parameters for which the FF-NN model does not produce meaningful output because of insufficient training dataset.

Video (x_v)	Audio (x_a)	Sync (x_s)	Delay (x_d)
Varying	Varying	Varying	Varying
Varying	Varying	Varying	Optimal
Varying	Varying	Optimal	Varying
Varying	Optimal	Varying	Varying
Optimal	Varying	Varying	Varying
Optimal	Optimal	Varying	Varying

TABLE IV: Input behavior for which the optimal delay model provides an accurate estimation of the output.

Video (x_v)	Audio (x_a)	Sync (x_s)	Delay (x_d)
Varying	Varying	Varying	Optimal
Varying	Varying	Optimal	Optimal
Varying	Optimal	Optimal	Optimal
Optimal	Varying	Optimal	Optimal
Optimal	Optimal	Varying	Optimal
Optimal	Varying	Varying	Optimal
Varying	Optimal	Varying	Optimal

We make an immediate observation that the dataset only partially explores the mapping from the four QoS parameters to the user-perceived QoE. For example, it does not consider what happens if we set the video and audio quality to optimal while varying the synchronization and delay. Additional trends of potential interest not covered in the dataset are presented in Table III. Non-surprisingly, it turns out that we cannot come up with a model that provides a complete QoS-to-QoE mapping, whether for the architecture presented in Figure 3 or for alternative architectures that we have considered. Furthermore, external smoothening functions are of limited use in this situation as there is not sufficient information in the dataset to enable these functions to work for all configurations. Consequently, one of the goals for our evaluation is to describe the limits of the model, namely, the input configurations for which the proposed model does work.

In fact, the dataset contains no single configuration in which both synchronization and delay are suboptimal. The dataset can be divided into two groups: configurations with the optimal delay and those with the optimal synchronization. The *optimal delay group* contains 25 (8+7+8) configurations representing the first three categories in Table II. The *optimal synchronization group* contains 26 (8+9+6+3) configurations representing categories 1, 4, 5, 6 from Table II. The idea behind this division is that it allows us to provide a separate QoS-to-QoE mapping for the case when the synchronization is optimal and the other three parameters vary and another separate mapping for the case when the delay is optimal while the other parameters vary. We construct a separate instance of the model for each of these two goals and train this instance using the configurations in the corresponding group.

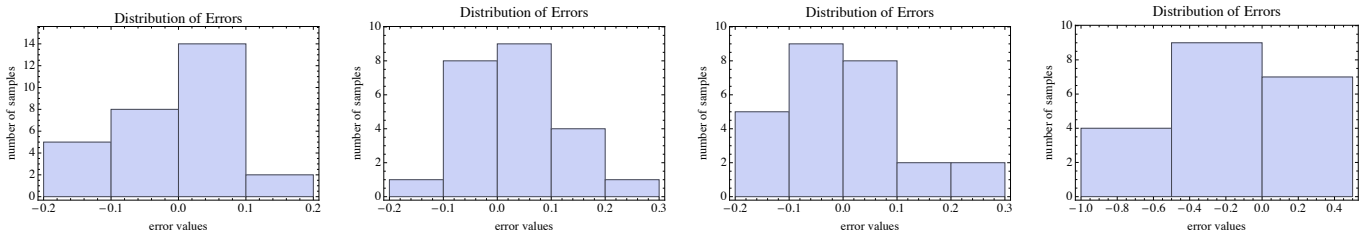
To make sure that the number of configurations in each group is sufficient for training and validation of the corresponding model instance, we extrapolate the given 25 configurations in the optimal delay group to 52 and the 26 configurations in the optimal synchronization group to 46. Following the methodology used in [10] for a different VoIP-based application, we split each group into the training and validation subsets. Out of the 52 configurations in the optimal delay group, we use 29 for training and 23 for validation. The 46 configurations in the optimal synchronization group are split into 26 for training and 20 for validation.

B. Optimal Delay Group

1) Validation objective: Figures 4a and 4b show a histogram of the error distribution values for the training and validation datasets of the optimal delay group. Each bar in Figures 4a and 4b shows the number of configurations produced by the model with an estimation error within the corresponding range. It can be seen that for a majority of the training and validation configurations, the estimation error lies within ± 0.1, which indicates the good accuracy capability of this model.

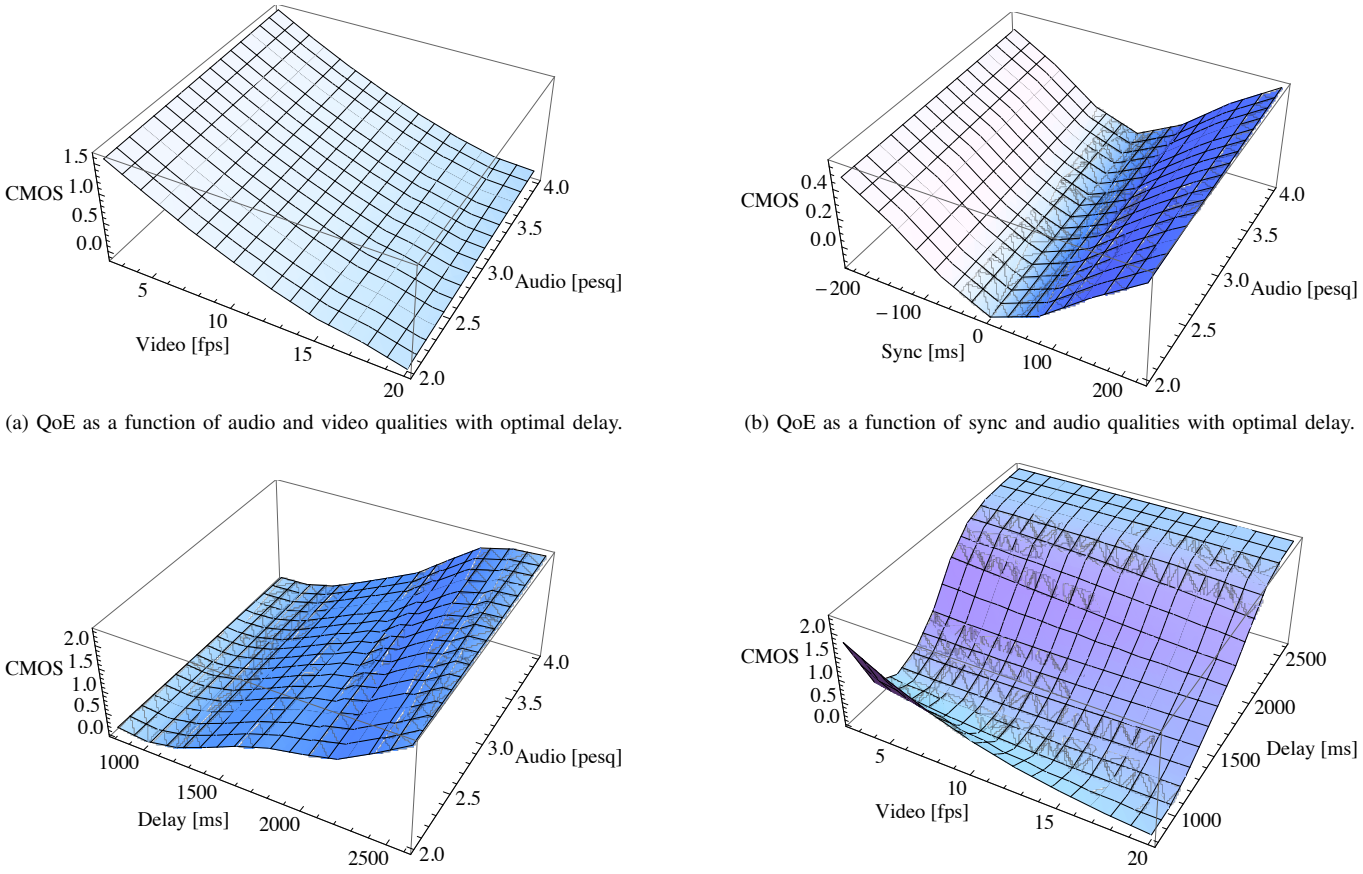
2) Extended output objective: With the trained optimal delay model, we are able to produce QoE results for a large range of values of the input parameters that are not available in the dataset. Figures 5a and 5b show the variation of perceived QoE as a function of two parameters. In Figure 5a, it can be clearly seen that when the video quality is high, the model predicts that the audience does not perceive significant difference between degraded audio quality ($x_a = 2.0$) and high audio quality ($x_a = 4.0$). This in turn shows the poor audio intelligibility of the audience, which is demonstrated by Figure 5b where the impact of varying the synchronization quality shows that the perceived QoE remains almost the same regardless of the audio quality.

3) Limits of the model: The optimal delay model is able to estimate the QoE values for the combined behavior of input parameters shown in Table IV.



(a) Optimal delay, training dataset. (b) Optimal delay, validation dataset. (c) Optimal sync, training dataset. (d) Optimal sync, validation dataset.

Fig. 4: Error distribution of the trained and validated datasets for the optimal delay and optimal synchronization models.



(a) QoE as a function of audio and video qualities with optimal delay.

(b) QoE as a function of sync and audio qualities with optimal delay.

(c) QoE as a function of audio and delay qualities with optimal sync.

(d) QoE as a function of video and delay qualities with optimal sync.

Fig. 5: Perceived quality (QoE) as a function of two input parameters. The other two parameters are set to their optimal values.

4) QoS-to-QoE mapping: The equation that maps the QoS parameters to QoE in the optimal delay model is as follows:

$$QoE_{delay} = 52.9993 + \frac{2.22606}{1+e^{a}} - \frac{55.1635}{1+e^{b}} - \frac{54.3458}{1+e^{c}} + \frac{19.4004}{1+e^{d}} - \frac{94.7714}{1+e^{e}} + \frac{77.4208}{1+e^{f}},$$

$$\begin{aligned} a &= 2.0054 - 0.0257x_v + 0.0219x_a - 0.0147x_s \\ b &= 10.693 + 0.0054x_v + 0.0066x_a - 0.002x_s \\ c &= -11.073 + 0.0052x_v - 0.0061x_a + 0.011x_s \\ d &= 1.7064 - 0.0202x_v - 0.0167x_a + 0.021x_s \\ e &= 1.7215 + 0.0146x_v - 0.0011x_a + 0.031x_s \\ f &= 1.458 + 0.0360x_v + 0.0022x_a + 0.033x_s \end{aligned}$$

C. Optimal Synchronization Group

1) Validation objective: Figures 4c and 4d show the error distribution values for the training and validation datasets of the optimal synchronization group. It can be seen from these figures that for a majority of the configurations, the estimation error lies within ± 0.1 for the training dataset and ± 0.5 for the validation dataset, which indicates the good accuracy capability also for this model.

2) Extended output objective: With the trained optimal synchronization model, we are able to produce QoE results for a large range of values of the input parameters that are not available in the dataset. Figures 5c and 5d show the variation of perceived QoE as a function of two parameters. Figure 5c shows that the impact of varying delay is almost the same

TABLE V: Input behavior for which the optimal synchronization model provides an accurate estimation of the output.

Video (x_v)	**Audio** (x_a)	**Sync** (x_s)	**Delay** (x_d)
Varying	Varying	Optimal	Varying
Varying	Varying	Optimal	Optimal
Varying	Optimal	Optimal	Varying
Optimal	Varying	Optimal	Varying
Optimal	Optimal	Optimal	Varying
Optimal	Varying	Optimal	Optimal
Varying	Optimal	Optimal	Optimal

for the degraded and high audio quality. Figure 5d shows that the impact of degraded delay ($x_d = 2.6$s) under the optimal video quality ($x_v = 20$) is greater than the impact of degraded video quality ($x_v = 2.5$) under the optimal delay ($x_d = 0.8$s). This is because in a conferencing application, the audience is more concerned about the interactivity compared to the video quality.

3) Limits of the model: The optimal synchronization model is able to estimate the QoE values for the combined behavior of input parameters shown in Table V.

4) QoS-to-QoE mapping: The equation that maps the QoS parameters to QoE in the optimal synchronization model is as follows:

$$QoE_{sync} = 1.613 + \frac{0.715}{1+e^{a}} - \frac{1.367}{1+e^{b}} - \frac{0.684}{1+e^{c}} + \frac{1.056 \times 10^{13}}{1+e^{d}},$$

where

$$\begin{aligned} a &= 21.404 - 0.1685x_v + 0.3291x_a - 0.0129x_d \\ b &= -18.363 - 0.1161x_v + 1.5298x_a + 0.0073x_d \\ c &= -10.595 - 0.0063x_v + 0.6772x_a + 0.0094x_d \\ d &= -11.131 + 0.1198x_v - 0.0350x_a + 0.0503x_d \end{aligned}$$

VI. Discussion

Our modeling approach requires to conduct extensive subjective assessment technique of tele-immersive applications, in order to construct the datasets sufficient to train the models. After the models are successfully trained, they can be used to replace the future evaluation of these applications during various scenarios such as updating the system architecture or policies of the system. Furthermore, the equations obtained from the models can be useful for designing real-time QoE monitoring.

To illustrate potential usefulness of our models, consider three geographically distributed clients interacting using a tele-immersive conferencing application offered by a service provider. The service provider takes responsibility for all client-side hardware, software and network services on behalf of its clients. The objective of the service provider is to guarantee maximum QoE to all three clients. Figure 6 depicts the assumed architecture. The following is the sequence of events that the service provider can perform in order to achieve the objective:

1) The service provider collects key performance metrics for video, audio, synchronization, and interactivity qualities (as discussed in Section II) from its clients.

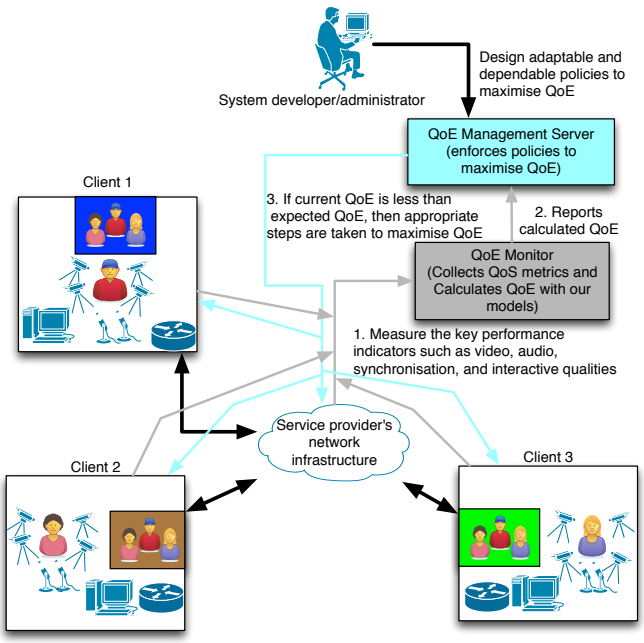


Fig. 6: An example use case of QoE real-time monitoring.

2) The collected metrics are given as an input to our proposed model/function, which calculates the overall QoE of the three users, when they are using the service. The probes along with a proposed model form a QoE monitor.
3) The calculated QoE along with QoS metrics should be passed to the QoE management server, which is used to enforce the system administrator's adaptable and dependable policies and maximize the QoE. If the current QoE value is smaller than the target QoE value, the QoE management server identifies the QoS metrics responsible for the difference and invokes the appropriate mechanisms to maximize the QoE. For example, if the QoE management server detects that the interactivity quality becomes suboptimal and the end-to-end delay between the users gets increasing because of network congestion, then the server can use mechanisms to enforce streaming via an alternative network path.

VII. Related Work

Most of the works in the area of assessing the QoE for tele-immersive applications pursue the direction of either objective or subjective evaluation. To the best of our knowledge, there are no QoE prediction models available for tele-immersive applications as the area is still developing. However, there exist QoE prediction models in the literature for interactive multimedia applications with somewhat less strict demands for immersiveness and interactivity, such as VoIP [11], [12] or VVoIP [13].

The objective QoE evaluation is essentially system-centric, where the QoS metrics of the system is used to calculate the QoE of the users. Studies have shown that systems providing the best QoS do not necessarily provide the best QoE to the

users [14]. This is due to the difference between the human-centric and system-centric evaluation methods. Nevertheless, objective evaluations are used in practice in the scenarios where conducting subjective assessment is difficult. For VoIP, ITU-T recommends the PESQ model [7] and the E-model [15] to compute the QoE of the users.

In the subjective evaluation, the users are requested to provide ratings based on their experience of using the system. ITU-T G.1070 [16] describes the standards for conducting subjective assessment for real-time interactive video-conferencing applications. [17] proposed a subject assessment technique, in which the users are requested to click a dedicated key, when they are dissatisfied with the quality of the applications they use. [3] has shown the effectiveness of comparative category rating-based subjective assessment techniques in tele-immersive applications.

[11] and [12] use models for assessing the QoE of VoIP and similar services based on random neural networks. Similarly, [13] proposes a QoE estimation model for VVoIP, which is also based on neural networks. It should be emphasized, however, that the QoS parameters that affect the user-perceived QoE substantially vary across different multimedia applications. In particular, tele-immersive applications require a different set of parameters, as recently shown by [3]. To illustrate this point, consider that network problems such as jitter or packet loss are uncommon in tele-immersive applications that are typically deployed over dedicated high reliable networks. On the other hand, hardware and software problems on the client side can significantly contribute to the application delay. The specific set of QoS parameters is so important because it is this set that should be used to train the neural network models to closely mimic the results of subjective assessment techniques. As a result, the models in [11], [12], and [13] are not applicable for evaluating the QoE of tele-immersive applications.

VIII. Conclusion

We have provided QoE estimation models for a conferencing type of tele-immersive applications. The model can be useful for real-time QoE monitoring for these applications. Thus, it can be helpful in designing QoE-driven adaptable and dependable solutions that provide acceptable quality of experience to the users. Furthermore, they can be helpful in the context of resource management for these applications.

In the future, we aim to compare the performance of various architectures of neural network models and also apply our modeling approach to other tele-immersive applications such as collaborative gaming. Additionally, we aim to design QoE models for other tele-immersive applications such as distributed opera performance [1].

References

[1] N. R. Veeraragavan, R. Vitenberg, and H. Meling, "Reliability modeling and analysis of modern distributed interactive multimedia applications: a case study of a distributed opera performance," in *Proceedings of the 12th IFIP WG 6.1 international conference on Distributed Applications and Interoperable Systems*, ser. DAIS'12, 2012, pp. 185–193.

[2] W. Wu, M. A. Arefin, Z. Huang, P. Agarwal, S. Shi, R. Rivas, and K. Nahrstedt, "I'm the Jedi! - A Case Study of User Experience in 3D Tele-immersive Gaming," in *ISM*, 2010, pp. 220–227.

[3] Z. Huang, A. Arefin, P. Agarwal, K. Nahrstedt, and W. Wu, "Towards the understanding of human perceptual quality in tele-immersive shared activity," in *Proceedings of the 3rd Multimedia Systems Conference*, ser. MMSys '12. New York, NY, USA: ACM, 2012, pp. 29–34.

[4] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1994.

[5] D. E. Ott and K. Mayer-Patel, "Coordinated multi-streaming for 3d tele-immersion," in *Proceedings of the 12th annual ACM international conference on Multimedia*. ACM, 2004, pp. 596–603.

[6] A. Jaimes, N. Sebe, and D. Gatica-Perez, "Human-centered computing: a multimedia perspective," in *Proceedings of the 14th annual ACM international conference on Multimedia*, 2006, pp. 855–864.

[7] "ITU-P. 862," 2001, perceptual evaluation of Speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.

[8] "ITU-P. 910," 2008, subjective video quality assessment methods for multimedia applications.

[9] W. Research, *Mathematica Edition Version 9.0*. Champaign, Illinois: Wolfram Research, Inc, 2012.

[10] A. P. C. da Silva, M. Varela, E. de Souza e Silva, R. M. M. Leão, and G. Rubino, "Quality assessment of interactive voice applications," *Computer Networks*, vol. 52, no. 6, pp. 1179–1192, 2008.

[11] G. Rubino, M. Varela, and J.-M. Bonnin, "Controlling multimedia qos in the future home network using the psqa metric," *Comput. J.*, vol. 49, no. 2, pp. 137–155, 2006.

[12] M. Varela, "Pseudo-subjective quality assessment of multimedia streams and its applications in control," Ph.D. dissertation, Université de Rennes 1, November 2005.

[13] P. Calyam, E. Ekici, C.-G. Lee, M. Haffner, and N. Howes, "A GAP-model based framework for online VVoIP QoE measurement," *Communications and Networks, Journal of*, vol. 9, no. 4, pp. 446 –456, dec. 2007.

[14] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, "User acceptance of computer technology: a comparison of two theoretical models," *Manage. Sci.*, vol. 35, no. 8, pp. 982–1003, Aug. 1989.

[15] "ITU-G. 107," 2011, the E-model, a computational model for use in transmission planning.

[16] "ITU-G. 1070," 2007, opinion model for video-telephony applications.

[17] K.-T. Chen, C.-C. Tu, and W.-C. Xiao, "Oneclick: A framework for measuring network quality of experience," in *INFOCOM*, 2009, pp. 702–710.