OneClick: A Framework for Measuring Network Quality of Experience

Kuan-Ta Chen[†], Cheng-Chun Tu[‡], and Wei-Cheng Xiao[†]

[†]Institute of Information Science, Academia Sinica
[‡]Department of Computer Science, Stony Brook University

***Abstract*—As the service requirements of network applications shift from high throughput to high media quality, interactivity, and responsiveness, the definition of QoE (Quality of Experience) has become *multidimensional*. Although it may not be difficult to measure individual dimensions of the QoE, how to capture users' overall perceptions when they are using network applications remains an open question.**

In this paper, we propose a framework called `OneClick` to capture users' perceptions when they are using network applications. The framework only requires a subject to click a dedicated key whenever he/she feels dissatisfied with the quality of the application in use. `OneClick` is particularly effective because it is *intuitive, lightweight, efficient, time-aware, and application-independent*. We use two objective quality assessment methods, PESQ and VQM, to validate `OneClick`'s ability to evaluate the quality of audio and video clips. To demonstrate the proposed framework's efficiency and effectiveness in assessing user experiences, we implement it on two applications, one for instant messaging applications, and the other for first-person shooter games. A Flash implementation of the proposed framework is also presented.

***Index Terms*—Human Satisfaction, MOS, Online Gaming, Poisson Regression, QoE (Quality of Experience), User Perception, VoIP**

I. INTRODUCTION

Providing high QoE (Quality of Experience) for users of network applications is one of the most challenging tasks that network researchers have to deal with because communication bandwidth is not always infinite. As the service requirements of network applications shift from high throughput to high media quality, interactivity, and responsiveness, *the definition of QoE has become multidimensional*. For example, the QoE requirements of VoIP conversations should at least include criteria for sound fidelity, voice loudness, noise levels, echo levels, and conversational delay; and the QoE requirements of online gaming should at least include criteria of interactivity, responsiveness, and consistency. Although it may not be difficult to measure individual dimensions of the QoE, *how to capture users' overall perceptions when they are using network applications remains an open question*. For instance, suppose one network provides a QoE setting of $(10, 15, 20)$, which represents the consistency level, interactivity level, and responsiveness level respectively, and another network provides a QoE setting of $(20, 15, 10)$. The issue of which network configuration is "better" from the user's perspective has yet to be investigated.

The most common way to obtain feedback is via opinion rating, i.e., users are asked to complete a questionnaire about their network experiences after they use an application. For example, MOS (Mean Opinion Score) [8] is a common metric employed in user surveys of VoIP quality. The MOS score range is 1 to 5; 3 or above is considered acceptable. However, the survey method has the following weaknesses.

1) A survey cannot capture users' perceptions *over time* as it is performed afterwards. Specifically, in a 3-minute test, we cannot determine whether a signal in the first minute and a signal in the last minute have an identical impact on a subject with the survey method.
2) People are limited by their finite memory; for example, only seven items can be remembered by a human being at one time according to [13]. Consequently, people are subject to the recency effect [16], i.e., the most recent experience dominates a user's ratings.
3) With the survey method, subjects are expected to give *absolute ratings*; thus, they need to score the quality levels with text descriptions. For example, the MOS scores 1, 2, 3, 4, and 5 correspond to Bad, Poor, Fair, Good, and Excellent experiences, respectively. It is not surprising that people may have different interpretations of the term "Poor" and give different ratings even though they have the same experience in a test.
4) Little information is captured in each test because a subject only contributes one score. This increases the cost of user surveys, as many subjects are needed before sufficient information can be acquired. Even if the test if repeatable, usually more than several dozen subjects are required because a subject cannot take the same test too many times due to reasons like: 1) the test requires full concentration, and subjects can get tired easily; and 2) a subject may become an "expert" from taking repeated tests and give scores with biased ratings.

In this paper, we propose a framework to capture users' perceptions when they are actually using network applications. *We call the framework `OneClick`, because users are asked to click a dedicated button whenever they feel dissatisfied with the quality of the application in use*. `OneClick` is particularly effective because it is *intuitive, lightweight, efficient, time-aware, and application-independent*. We explain each of its advantages below.

1) It is intuitive because users do not need to differentiate between "Good" and "Fair" or between "Bad" and "Poor." In an `OneClick` test, users only need to click the dedicated button if they are not happy with the current application's quality. In other words, a simple dichotomous decision rather than a multiple-choice decision is required. Furthermore, users do not need to remember previous experiences, as feedback can

be given immediately. Therefore, as subjects do not need to pay much attention to the scoring procedure, their flow experience in using the tested application can be maintained.

2) It is lightweight because it does not need large-scale or expensive deployments to perform experiments. In fact, the only extra equipment needed is the hardware required to run the tested application. For example, to evaluate users' perceptions about video quality, any PC that is capable of playing video clips can be used to perform the experiment.
3) It is efficient because it captures a large number of click events from a subject in each test. Our experiment results show that a single 5-minute test with one subject is normally sufficient to reveal his/her perceptions about audio or video quality over a wide spectrum of network conditions.
4) It is time-aware because users can provide feedback any time during a test. Thus, the measurement results reveal how a subject's perception changes over time. For example, our method can be used in video quality tests to evaluate if a subject tends to adapt to the jerky screen updates due to network lags over time.
5) It is application-independent because it relies purely on users' feedback rather than any application-specific design. Thus, it can be used to measure users' perceptions of many types of network applications. In Section IV and Section V, we show that `OneClick` is capable of evaluating the quality of audio streaming tools, compressed video clips, and interactive online games.

To demonstrate the efficacy of our approach, we select the three most well-known instant messaging (IM) applications, Skype, MSN Messenger, and AIM Messenger, and use `OneClick` to evaluate their audio and video quality in various network configurations. The results show that, from the user's perspective, Skype provides the best quality over a wide range of network loss rates and bandwidth. We also evaluate user experience in two first-person shooter games, Unreal Tournament and Halo. The results show that Unreal Tournament can provide better game play experience than Halo under conditions of moderate network delay and delay variations.

The contribution of this paper is three-fold. 1) To quantify users' perceptions of network applications, we propose the `OneClick` framework which is intuitive, lightweight, efficient, time-aware, and application-independent. 2) We apply the proposed framework to evaluate the quality of three popular IM applications and two first-person shooter games at very low cost. This manifests that `OneClick` is an application-independent and effective way of evaluating the quality of network applications. 3) We provide an online implementation of `OneClick` at *http://mmnet.iis.sinica.edu.tw/oneclick* for collaborative studies. Any one can contribute his/her perceptions of the quality of the provided audio clips. The contributions will be merged to form a composite perception for future QoE studies.

The remainder of this paper is organized as follows. Section II contains a review of related work. In Section III, we present the basic methodology of the `OneClick` framework and a revised methodology based on the implications of pilot studies. In Section IV, we validate the effectiveness of the `OneClick` framework using well-known objective evaluation methods of audio and video quality. Section V details an implementation of the proposed framework. Specifically, we use it to compare the quality of three IM applications and two first-person shooter games. Then, in Section VII, we summarize our conclusions.

II. Related Work

Methods for assessing the quality of network applications can be classified as either *objective* or *subjective*. Subjective methods require users' input in the evaluation process. Objective methods do not require such input, but they are often based on well-validated data derived from subjective evaluations.

Objective evaluations rely on passive monitoring of an application's or a user's behavior. For example, they may take network-level measurements [2, 4, 7] or media-level measurements [11, 15] as the input for quality assessment. Many techniques and standards based on this approach exist for evaluating the quality of audio content [10, 11], video content [9], and online games [3]. Objective evaluation methods are effective when user input is not required; however, they cannot assess all the QoE dimensions that may affect users' experiences in quality summarization. For example, to ensure that a model remains tractable, the loss of volume is not considered by the PESQ (Perceptual Evaluation of Speech Quality) model [11]; the distance between viewers and the video is not considered by the VQM (Video Quality Measurement) model [9]; and the higher-order variations, i.e., the burstiness, of end-to-end delay and loss are not considered by the E-model, which is designed for assessing VoIP quality [10].

Subjective quality evaluations, on the other hand, require users to indicate their feelings in a survey. However, considering the limited efficiency and high resource overhead of such schemes, and the fact that people's feelings are subject to various effects like personal biases and the memory effect, it is not always easy to obtain reliable evaluation results. The Mean Opinion Score (MOS) [8] is probably the most widely used subjective quality assessment method. Although it is often used to evaluate user experience in applications, it has a number of limitations, as described in [12].

In [5], Dinda et al. proposed a scheme that employs a similar concept to that of the `OneClick` framework to evaluate user satisfaction in response to different scheduling policies and clock frequencies of computer systems. However, the goals of the two applications are totally different. Our work expands the idea of "click whenever dissatisfied" into a complete framework, which includes experiment design, modeling, and interpretation. In addition, we provide a validation of the proposed framework based on two objective evaluation methods, PESQ and VQM, and prove its usefulness by applying it to popular IM applications and online games.

III. Methodology Development

In this section, we describe the design of the `OneClick` framework and the rationale behind it. We begin with a more intuitive design and then gradually revise the design based on the findings in pilot studies. Finally, we summarize the framework with the data flow chart shown in Fig. 6.

A. Basic Methodology

The OneClick framework comprises two phases. The first phase involves setting up experiments to collect users' feedback, and the second analyzes the collected raw data to determine users' perceptions under different network settings.

The experiments can be performed on any computer equipped with a (software) key logger and a traffic monitor, which can be omitted if pre-recorded test materials are used. We ask a subject who is using a real-time network application, e.g., conferencing or gaming, to click a dedicated button whenever he/she feels dissatisfied with the quality of the application in use. Here, the quality of the application refers to all the QoE dimensions that affect user's satisfaction. For example, it could be poor voice quality or conversation smoothness in conferencing, or screen freezing, status inconsistency, or slow responsiveness in online gaming. Users do not need to be well-trained to participate in the experiments as only an intuitive click action is required.

Technically speaking, we consider a subject as a stochastic system whose transfer function is unknown. During the experiments, we record both the input and the output of the system, where the input is *the process of network conditions*, and the output is *the subject's click process*. After analyzing the input and output signals, we use a generalized linear model to describe the relationship between user satisfaction and network conditions. Although the subjects' click actions are obtained via passive measurement, the network conditions can be actively controlled or passively observed, depending on the experiment's goal. Generally, it is easier to achieve reliable quality assessment by actively controlling network quality, which is appropriate for more systematic evaluations. Experiments are conducted on uncontrolled networks to understand user experience in realistic scenarios.

1) Test Material Compilation: OneClick can be used to evaluate the quality of an application currently in use, or the application's prerecorded output, which we call *test materials*. When an application has complex interactions with network factors, we suggest using test materials to ensure the accuracy of the assessment. For example, most VoIP applications adopt a playout buffer to absorb the variations in network delays. With this design, if we change the network condition at time t, the audio quality may be changed at time $t + t_\delta$, where $t_\delta \neq 0$ is unobservable and unpredictable. In this case, using test materials would prevent the misalignment between the network conditions and users' feedback; however, test materials can only be used in non-interactive scenarios.

We use an example to illustrate the compilation of test materials. To compile test materials for evaluating users' perceptions of VoIP listening quality with k network settings $N_1, N_2, ..., N_k$. We first make k recordings, $R_1, R_2, ..., R_k$, at a VoIP receiver under network conditions $N_1, N_2, ..., N_k$, respectively. We then compile a set of test materials by extracting *random non-overlapping segments* from $R_1, R_2, ..., R_k$, as shown in Fig. 1. By so doing, the resulting clip contains exactly the same content as the original clip, except that each portion of the resulting clip is a degraded version because of network transmission. In addition, we purposely insert segments of R_0, which is actually from the original clip, into the resulting clip at regular intervals. The purpose of this design is to help users recall the audio quality in a perfect scenario, as their perceptions of quality may become biased due to long-term imperfect and varying scenarios.

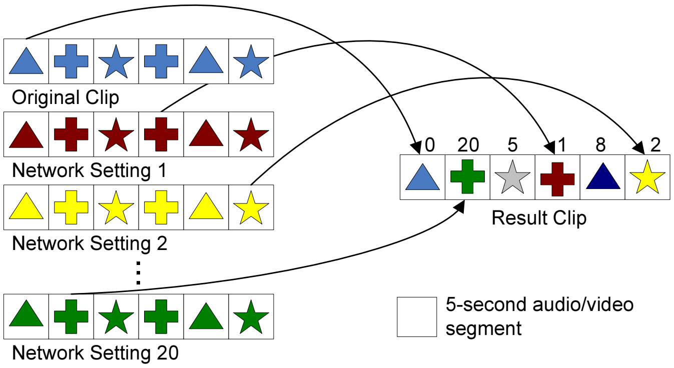


Fig. 1. A test material is composed by random non-overlapping segments from k clips recorded under k network conditions respectively.

2) Click Rate Modeling: We apply regression to resolve the relationship between network conditions and the occurence of click events for three reasons: 1) it is a well-established statistical methodology with considerable resources and applications; 2) it is flexible because we can easily switch from a simple linear regression to a more complicated non-linear regression when a more accurate model is required; and 3) it provides ready-to-use tools, such as hypothesis tests, which are helpful for assessing whether a certain network factor has a significant impact on users' perceptions.

We treat a subject's click events over time as a *counting process*; hence, the Poisson regression is a natural choice [6]. Our model treats the network factors as predictors and the click rate as a dependent variable, which is computed as the average number of times the subject clicks the button in one second. Assume the click rate is $C(t)$ and the network factors are $N_1(t), N_2(t), \ldots, N_k(t)$ at time t. Then, the Poisson regression equation is

$$\log(C(t)) = \alpha_0 + \alpha_1 N_1(t) + \ldots + \alpha_k N_k(t), \qquad (1)$$

where α_i denotes the Poisson regression coefficients estimated using the maximum likelihood method.

By treating the click rate as a linear indicator of a user's level of satisfaction with the tested application, we can estimate the relative impact of any combination of network factors with the fitted model. For example, suppose we include two factors, network delay and loss rate, and we wish to know whether the network setting (100 ms delay, 5% loss rate) is better than (200 ms delay, 10% loss rate) from the user's perspective. By computing $C_a = \alpha_0 + \alpha_1 100 + \alpha_2 0.05$ and $C_b = \alpha_0 + \alpha_1 200 + \alpha_2 0.1$, we can justify the above conjecture if C_a is lower than C_b; otherwise, we can reject it.

B. Pilot Study

In the following, we describe a series of pilot studies conducted to evaluate whether our basic methodology is effective and reliable. Each study evaluates the audio quality of AOL Instant Messenger (AIM) with a different network loss rate. Test materials are used to avoid the misalignment problem between predictors and dependent variables due to playout buffers. The test audio clip is a popular English song of 300 seconds, and each test scenario lasts for 5 seconds.

In the experiment, subjects are asked to press the space key whenever they feel unhappy about the music quality, without

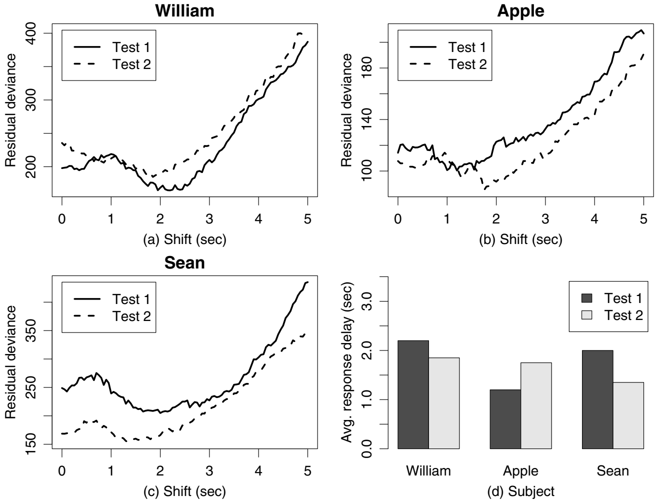


Fig. 2. (a)(b)(c) The relationship between the time lag added to the click event process and the residual deviance of the fitted Poisson regression model for three subjects. (d) The average response delay time in each of the 6 tests made by the three subjects.

knowing the current network setting. We asked three computer science students to take the tests so that we could study their response delays, the consistency and bias of user feedback, and the effect of test materials.

1) Response Delays: We observe that users may not always respond quickly; that is, they may unintentionally delay click actions after they become aware of the degraded application quality. To compensate for this effect, we assume that the response delays in the same test are generally close to the average response delays; a subject may have different response delays from time to time. Our solution is to shift the click event process and determine how much delay should be added to yield the best model fit. We search for the average delay time d_{avg} by fitting the Poisson regression model for network factors and click event processes with different time lags, where d_{avg} is computed as

$$argmin_d\{\text{deviance of (1) by replacing C(t) with C(t+d)}\}.$$

We conducted a number of experiments to validate the proposed solution. Figure 2 shows the residual deviance of the fitted model versus the average delay time for the tests taken by the three subjects. We can see that the residual deviance with different average delays is generally concave upwards with a local minimum around 1–2 seconds. This implies that 1) the Poisson regression model fits our data; 2) the variable response delays can be accounted for by an average response delay; and 3) our subjects normally delay their click actions after hearing music with unsatisfactory quality[1]. From the graphs, we observe that a subject's average response delays may change in different tests and, not surprisingly, the average response delays are different for the three subjects.

As we have demonstrated the existence of response delays, we use the shifted click process (by d_{avg}) to replace the original click process in the rest of analysis and modeling tasks.

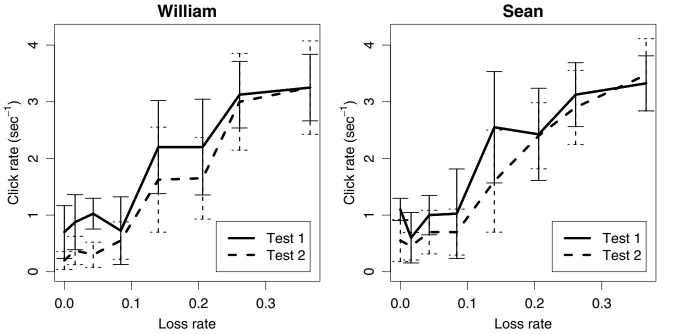


Fig. 3. A subject tends to produce similar click rate curves in different tests.

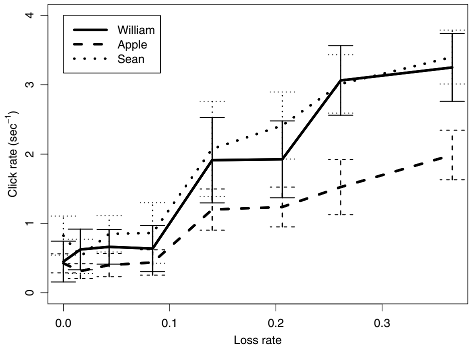


Fig. 4. Different subjects may have very different click behavior due to personal characteristics or preferences even if they share similar perceptions of an application's quality of experience.

2) Consistency of User Feedback: One common concern about subjective quality evaluation methods is the consistency of users' feedback, i.e., if the experiment design tends to make a subject give similar ratings in repeated experiments. Because of this concern, we asked two subjects to take the same test twice. We then plotted the relationship between their click rates and network loss rates, as shown in Fig. 3. The vertical bars denote the 95% confidence band of the average click rates with a certain loss rate[2]. The figure shows that each subject tended to give similar ratings in two tests because the average click rates in both tests in most of the scenarios were statistically equivalent. For this reason, we merge the results in different tests taken by the same user by averaging the click rates in each test scenario. This result also suggests that a 5-minute `OneClick` test is sufficient to reliably measure a subject's perceptions of audio quality under a wide spectrum of network loss rates.

3) Bias of User Feedback: `OneClick`'s design reduces the difficulty of decision-making by offering dichotomous choices; that is, a subject only needs to decide whether or not to click the button. However, users' preferences may still have a significant effect on their click behavior. For example, some subjects may click the button when the perceived quality is slightly worse than expected, while others may only click it when the quality is really unacceptable. The effects of a user's preferences and personal characteristics are obvious, especially

[1] The response delays may also be due to the application's processing delay, such as the time taken in audio decoding and playback; however, it is unrelated to our processing here.

[2] The unequal horizontal spaces between samples are intentional because the impact of loss rates on audio quality is nonlinear, as shown by PESQ curve shown in Fig. 7. Thus, we select the loss rates that are expected to yield the largest discrepancy in click rates. The purpose of this design is simply to reduce the number of test scenarios and maintain the efficiency of the experiments. It does not affect the effectiveness of the method or the accuracy of the assessment results.

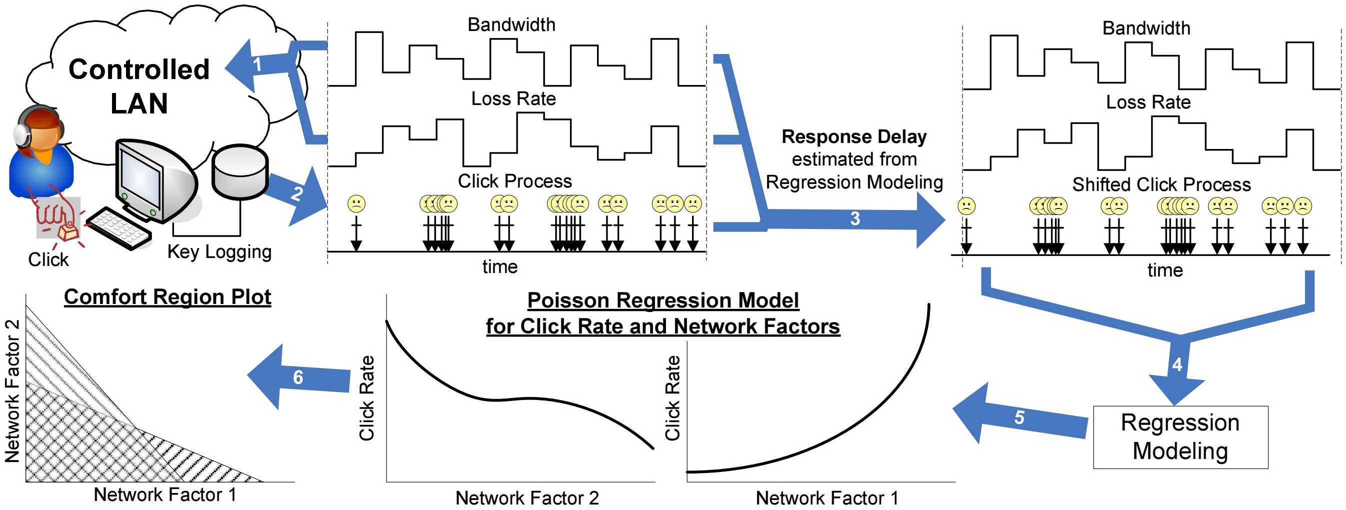


Fig. 6. The flow of a complete OneClick assessment procedure. 1) Preparing test materials (optional); 2) asking subjects to do experiments; 3) inferring average response delays; 4) modeling the relationship between network factors and click rates; 5) predicting the click rate given each network factor; 6) summarizing an application's QoE over various network conditions by comfort regions (will be introduced in Section V.

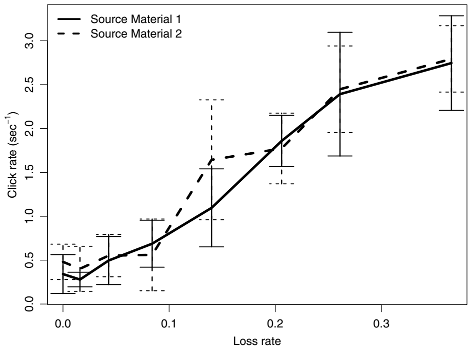


Fig. 5. The choice of source materials made limited effects to users' click behavior (if the materials share similar characteristics).

when an application's quality is unsatisfactory for a lengthy period. In such cases, some subjects may click the button repeatedly and as fast as possible to reflect their high degree of dissatisfaction; others may be more patient and only click the button spontaneously and casually, even though they are also frustrated by the continuous bad quality. Personal differences like those in the above example certainly introduce biases into users' click decisions.

To determine how user bias impacts on the evaluation results, we plot the relationships between the average click rates and the loss rates for three subjects, as shown in Fig. 4. The three curves clearly indicate that while two of the subjects, William and Sean, exhibit similar click behavior, Apple was much more conservative in deciding whether to click. Even so, we can observe that the difference between Apple's click rate curve and that of the other two subjects is approximately linear to the loss rate. In other words, if we multiply Apple's click rates by an appropriate constant, her resulting click rate curve will be similar to those of William and Sean. This implies that Apple's perception of audio quality with network loss is similar to the perceptions of the other two subjects, but her more conservative feedback strategy leads to the discrepancy in click rates.

For the above reasons, to obtain the overall average perceptions of many subjects, we need a way to combine the click rate curves of the subjects with consideration of the subjects' biases. Our solution is to normalize each subject's click rate curves before merging all the curves. Let the average click rate of a subject i in the test scenario j be $cr_i(j)$. We align the medians of $cr_i(\cdot), \forall i$, by making all of them equal to $med(med(cr_i(\cdot)))$, where $med(\cdot)$ denotes the median function. More specifically, we define $cr'_i(j) = cr_i(j) - med(cr_i(j)) + med(med(cr_i(j)))$, and compute the overall click rate curve $cr_{all}(j)$ by taking an average of $cr'_i(j)$ for all i.

4) Effect of Source Material: It is also possible that different source materials, e.g., audio or video clips, will result in different conclusions about the impact of network conditions on application quality. To resolve this concern, we selected two popular songs, one in English, and the other in Chinese, and asked three subjects to evaluate the music quality under different loss rates. As shown in Fig. 5, the overall experience of the three subjects under different network scenarios for both songs was surprisingly similar, in that the click intensity in response to both types of source material was statistically equivalent. Even so, we cannot conclude that users' perceptions are not affected by the source materials because such a strong conclusion would need to be supported by more experiments. However, we believe that the quality of audio clips with comparable acoustic characteristics should exhibit similar levels of degradation due to network impairment; therefore users' experience should be similar, as demonstrated by the experiment we performed.

C. Revised Methodology

Our changes to the basic methodology are as follows:

1) We shift the click process by d_{avg}, which is computed by Poisson regression modeling, before any other analysis and modeling tasks.
2) We normalize the click rate curves of different subjects before merging them into an overall curve.

The flow chart in Fig. 6 illustrates the steps in evaluating an application's QoE using the OneClick framework. In the next section, we validate the OneClick framework using

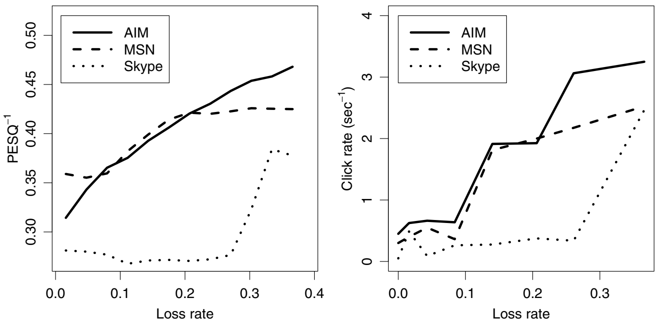


Fig. 7. Comparison of PESQ scores and `OneClick` evaluation results for the audio quality of IM applications over different loss rates.

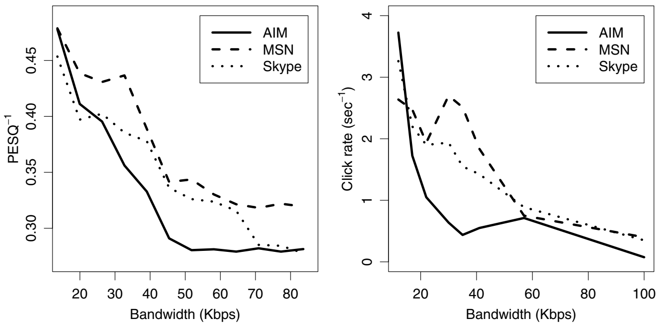


Fig. 8. Comparison of PESQ scores and `OneClick` evaluation results for the audio quality of IM applications over different network bandwidths.

objective quality assessment methods, and use it to evaluate the quality of several real-life applications.

IV. Framework Validation

Thus far, we have shown that users' satisfaction with an application under different network conditions can be quantified using the proposed framework (Section III-B). However, we need to demonstrate that the framework's evaluation results are trustworthy and reliable. Therefore, in this section, we validate the `OneClick` framework by comparing its assessment results with two objective quality evaluation methods, namely Perceptual Evaluation of Speech Quality (PESQ) and Video Quality Measurement (VQM).

We design two experiments for validation purposes. One is based on PESQ, an objective audio quality assessment tool; and the other is based on VQM, an objective video quality assessment tool. Our results show that users' perceptions measured using the `OneClick` framework are consistent with those derived by the objective methods.

A. PESQ-based Validation

In the first experiment, we use `OneClick` and PESQ to evaluate the relative audio quality of three popular instant messaging (IM) applications: AIM, MSN Messenger, and Skype. The experiment setup is exactly the same as that used in the pilot study. However, instead of comparing the audio quality with different network settings, we focus on the difference in the audio quality provided by different IM applications.

Figure 7 compares the results derived by PESQ and `OneClick`. Because better audio quality leads to a higher PESQ score and a lower click rate, the reciprocals of the PESQ scores are used for comparison. From the graph, we observe that the relative quality of the three applications is similar from the perspective of both evaluation methods. Specifically, Skype's audio quality is the best under different loss rates, and MSN Messenger performs slightly better than AIM when the loss rate is high. In addition, the difference between Skype's quality and that of the other two applications is significant.

In the second experiment, we change the network bandwidth in different test scenarios and keep the loss rate at zero. As shown in Fig. 8, the evaluation results obtained using PESQ and `OneClick` are consistent. From the results, we observe that AIM performs the best in the non-loss and bandwidth-limited scenarios, while Skype and MSN Messenger are ranked second and third respectively. The consistency in evaluation results demonstrate the efficiency of the `OneClick` framework, as a subject only required 15 minutes (5 minutes for each application) to obtain all the results shown in the right pane of Fig. 8.

Given that PESQ is effective in assessing the audio quality of IM applications, one may wonder why we have developed a framework that relies on the evaluations of human subjects. Our motivation is that, although PESQ normally provides accurate assessments of user satisfactions with audio signals, it can only capture a limited number of QoE dimensions. Moreover, there are many factors that it cannot measure or quantify. According to ITU-T P.862 [11], PESQ yields inaccurate predictions when used in conjunction with factors like listening levels, loudness loss, effect of delays in conversational tests, talker echo, and sidetone. Taking the listening level as an example, PESQ assumes a standard level of 79 dB SPL, so it requires the volume of both the original and degraded audio recordings to be normalized before the quality estimation procedures. Thus, if different applications incur different degree of loudness loss in their audio transmissions, PESQ will not be able to take account of such discrepancies. In addition, many factors that may affect listeners' perceptions are present in the playback stage, but information about them is not accessible through the audio recordings. For example, the effect of playback devices (speakers or headphones), environmental noise, and the distance between the listener and the speakers (if used) are not considered by PESQ because they are not part of the audio recordings. An objective quality evaluation scheme like PESQ or VQM, no matter how sophisticated it is, cannot capture all factors that may affect users' perceptions as some of the factors are unmeasurable. In contrast, subjective evaluation methods can capture all the QoE dimensions that humans perceive, as the ratings are given by the subjects directly. Therefore, we use PESQ for validation purposes and the `OneClick` framework in more complex scenarios, where unmeasurable factors are present. For example, the framework can be used to evaluate the audio quality of IM applications in which the environmental noise levels at the talker side may vary over time.

B. VQM-based Validation

Our second validation experiment is based on VQM, which is designed to evaluate the quality of a degraded video clip by comparing it with the original version before transmission or compression. In this experiment, we evaluate the impact of

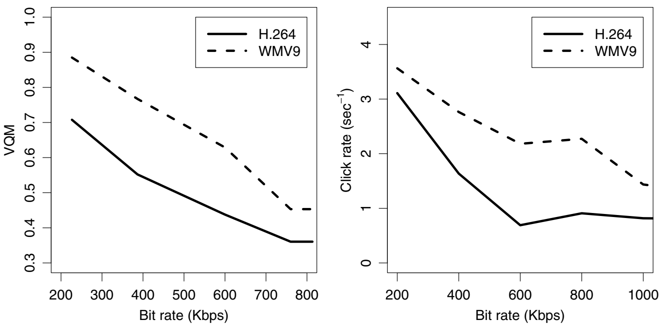


Fig. 9. Comparison of VQM scores and OneClick evaluation results for the video quality of two codec over different compression levels.

low-bit-rate compression, rather than that of network impairment, on video quality. The purpose of this design is to prove that the OneClick framework can be extended to assess the impact of non-network factors on an application's quality.

We take a 5-minute video clip from the movie "Transformer," and compress it with two video codecs, H.264 and WMV9, which are chosen because their compression levels can be specified arbitrarily. In this experiment, we use five compression levels from 200 Kbps to 1000 Kbps. First, we compiled a set of test material with random segments of different compression levels following the procedures described in Section III-A1, and asked the subjects to evaluate the result clip with OneClick. As shown in Fig. 9, both the objective and the subjective results indicate that users' satisfaction levels are roughly linear to the bit rate used. The VQM score is consistently in the range 0 to 1; a lower score reflects a higher quality. In addition, both evaluation methods demonstrate that the perceptual quality of the video clips compressed using H.264 is better than that derived by WMV9. The results of VQM-based validation support our assertion that the OneClick framework is reliable and efficient. Each subject only required 10 minutes to take the tests (i.e., 5 minutes for each codec), as shown in Fig. 9.

V. Case Studies: Evaluation of Application Quality

In this section, we demonstrate the efficacy of the OneClick framework for evaluating users' experience when using instant messaging applications and playing online games under different network conditions. We present two case studies to show that the proposed framework is application-independent and can be used to assess quality of experience in either non-interactive or interactive scenarios.

A. Instant Messaging Applications

In the first case study, we evaluate the listening quality of AIM, MSN Messenger, and Skype under various combinations of network loss rates and network bandwidth to observe the interaction between the two factors. Figure 10 shows the expected click rates with different loss rates and bandwidth. We observe that in most of the scenarios, MSN Messenger performs the worst among the three applications. Skype's quality is the best when the loss rate is less than 80 Kbps, but AIM performs better than Skype when the bandwidth is greater than 80 Kbps.

Interestingly, the applications have different degrees of robustness to network loss and bandwidth. We use the contour plot shown in Fig. 11 to gauge their robustness to different network conditions by examining how the click rate changes with a particular condition. On all the graphs, the contour lines run from the lower left-hand corner to the upper right-hand corner, which conforms to the intuition that higher loss rates and lower bandwidth have a semantically equivalent effect in terms of degrading the sound quality. However, the slopes of each application's contour lines for each application are different, which suggests that the applications have different degrees of sensitivity to network conditions. Specifically, *the slope of a contour line indicates the ratio between an application's sensitivity to network loss and that of bandwidth shortage.* Based on this criterion, we find that AIM is more sensitive to network loss than to bandwidth shortage, but Skype is much more sensitive to bandwidth shortage than to network loss. The two observations are consistent with PESQ's ratings, which we discussed in Section IV-A. Recall that Skype performs best in lossy scenarios, and AIM performs best in bandwidth-restricted scenarios. This phenomenon suggests that *applications' QoS requirements are often multi-dimensional; thus, simply focusing on one dimension and ignoring other dimensions may lead to the wrong conclusions.*

To compare the three applications' QoS requirements with a more global view, we define a "comfort region" as the set of network scenarios that leads to an expected click rate lower than a certain threshold. In other words, a comfort region comprises the set of network scenarios that yield an average click rate lower than the specified threshold. Figure 12 shows that the comfort regions of the three applications have different shapes, regardless of the click rate threshold. When the threshold is set to 1.5 times per second, the smallest comfort regions of MSN Messenger indicate that it is more difficult to provide the same QoE using MSN Messenger than using the other two applications. On the other hand, AIM and Skype have much larger comfort regions. This fact indicates that they can still offer acceptable audio quality under sub-optimal network conditions. We make the following observations from the graphs. 1) Skype is more effective in scenarios with low bandwidth, e.g., less than 60 Kbps, and a moderate loss rate, e.g., lower than 10%. 2) When the network bandwidth is sufficiently large, e.g., higher than 80 Kbps, AIM yields better sound quality than Skype. 3) In most scenarios, MSN Messenger's sound quality is the least optimal among the three applications. There may be a number of reasons why the three applications have such different levels of robustness to network impairment, for example, the design of the applications, the transport protocols, and the codecs they use. Discussion of these issues is beyond the scope of this work, but we will address them in future studies.

B. First-Person Shooter Games

Real-time online games are usually considered QoS-sensitive, i.e., sensitive to bad network quality. Online gamers often complain about network lags on forums, and comments like "ping times longer than 150 ms are harmful" are frequently heard in the gaming community. Network and game researchers are interested in understanding the effect of poor network quality on network game playing; however, there is

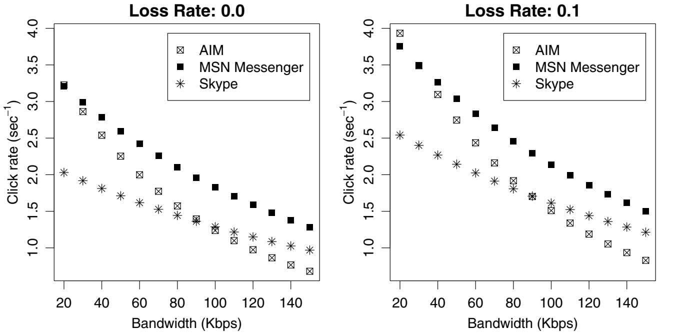


Fig. 10. The expected click rates under different combinations of network loss rates and bandwidth.

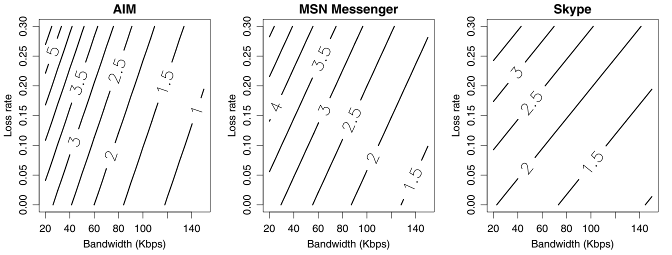


Fig. 11. The contour plot of expected click rates under different network settings.

currently no standard method for evaluating gamers' network experiences [3]. Some researchers use in-game achievements as objective indices [1, 14], for example, the number of kills in shooting games, or the time taken to complete each lap in racing games. However, game scores are highly dependent on the players' skills, the game's design, and the content, so the results are not comparable and generalizable across different games. Most of the remaining studies are based on traditional subjective evaluation methods, such as MOS. Because of the cost of such survey methods, the scale of user studies is often small; hence it is difficult to validate and generalize the results.

Here, we apply the `OneClick` framework to evaluate the impact of network conditions on two popular first-person shooter (FPS) games, namely Unreal Tournament and Halo. In FPS games, players must make sub-second reactions because of rapidly changing environmental factors, such as the actions of companions and opponents; therefore, network delays play an important role in users' perceptions of the quality of game play. In addition, the variations in network delay may affect the game play if the status updates from other parties are jerky and the response time of players' commands constantly vary. As a result, players may have difficulty to making a turn or firing on an opponent at an appropriate time. Thus, we change the network delay and its variation in the test scenarios.

Because game playing is interactive, we cannot use pre-recorded test materials. Instead, we put a router with dummynet in front of a game client and use it to add an intentional delay to every packet sent from the client to the game server. Assuming every game packet spends a relatively constant time traversing the Internet, we can evaluate the impact of different combinations of network delay and delay variations on the game's quality with `OneClick`. We define "delay jitter" as the standard deviation of network delays; and, without loss of generality, we assume delay jitters follow a Gamma distribution. The router is configured to automatically change

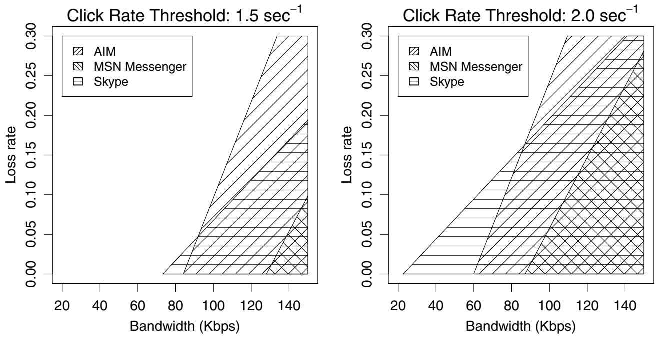


Fig. 12. The comfort regions of the three IM applications based on the Poisson regression model.

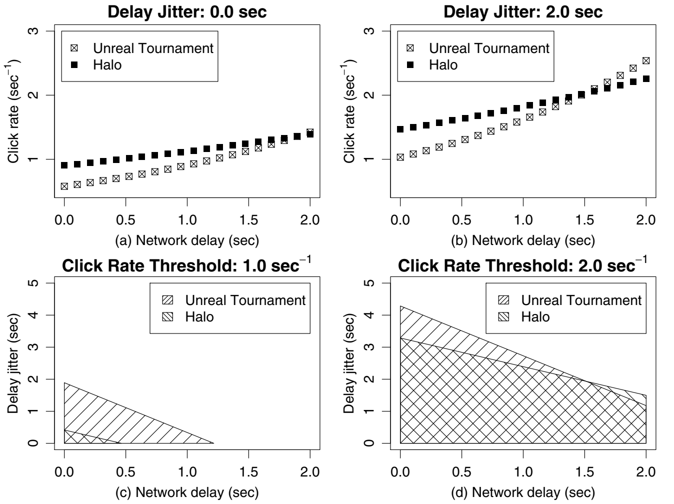
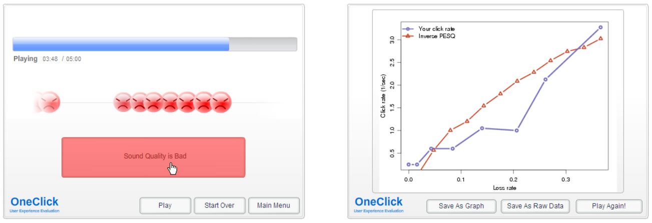


Fig. 13. (a)(b) The expected click rates under different combinations of network delay and delay jitter. (c)(d) The comfort regions of UT and Halo based on the Poisson regression model.

the network setting every 5 seconds. For each reconfiguration, it randomly sets the network delay and delay jitter within 2 seconds.

We asked three subjects to participate in the experiments, each of which lasted for 5 minutes, and then combined the subjects' evaluation results. Figures 13(a) and (b) show the expected click rates with different network delays and delay jitter. Generally, Unreal Tournament (UT) provides better quality than Halo under the same network conditions. UT is slightly less effective than Halo when both the delay and delay jitter are fairly high (nearly 2 seconds). The comfort region plots in Fig. 13(c) and (d) also reveal that UT provides better overall quality than Halo unless the network delay is extremely large. We cannot understand why UT is more robust to poor network quality than Halo unless we can analyze the designs of the game engines, especially the network handling and screen updating components, of both games. However, based on the comments of the experiment's participants, we suspect that UT implements certain delay-concealment and dead reckoning mechanisms to deal with network lags. Therefore, it provides players with better gaming experiences, even when the network quality deteriorates. As there is no standard method for comparing players' network experiences across games, we plan to further validate our evaluation results in future work.



(a) Evaluation stage

(b) Result review stage

Fig. 14. The Flash implementation of `OneClick`.

VI. Discussion

A. Multimodal QoE Assessment

Although there are a number of well-developed objective quality evaluation methods for audio and video quality, we believe that the `OneClick` framework is especially helpful for *multimodal QoE assessment and management*. Real-time multimodal communications have become more popular in recent years. Video conferencing, which combines audio and video real-time streaming, is a notable example. Besides, many online games now have a built-in audio chatting facility, so that the game data and voice data can be transmitted between game peers simultaneously. While many studies have focused on optimizing voice data delivery and game data delivery separately, how to optimize a network *simultaneously for both game data and voice data transmission* from the perspective of users has yet to be addressed. From this perspective, `OneClick` can be useful in the multimodal QoE assessment and management studies because of its application-independent nature.

B. OneClick Online

To demonstrate efficacy of `OneClick`, we provide a Flash implementation on the project website at *http://mmnet.iis.sinica.edu.tw/oneclick*. Upon entering the webpage, a user can choose which application's audio quality he/she wants to evaluate. Currently we provide three IM applications for evaluation, namely, AIM, MSN Messenger, and Skype. The user interface in the assessment stage is shown in Fig. 14(a). Each test lasts for 3 minutes, and the elapsed time is indicated by a progress bar. Whenever the user feels dissatisfied with the sound quality, he/she needs to click the big button with a mouse or the SPACE key. We use an unhappy face to symbolize a click. It will look more angry and bigger if the clicks occur frequently. Once the evaluation finishes, a plot is provided to let users know whether their evaluation results are similar to that computed by PESQ, as shown in Fig. 14(b). The user interface also provides users with an option to download the raw data they produced. The data contains the timestamp of every click action they made.

We believe that the Flash implementation would be helpful for collecting more user traces because, with this implementation, it is easier to ask the masses to join our study. Currently we plan to hold some competition events with certain rewards (physical or virtual) to attract Internet users to take tests in `OneClick`. The results produced by each participant are recorded. Eventually, the results will be used to produce an overall average result that incorporates the efforts of many subjects. We believe that this type of online quality assessment tool can be a key to accumulate a wealth of user measurement results in QoE studies.

VII. Conclusion and Future Work

In this paper, we present a framework called `OneClick` for capturing users' network experiences when using applications. As we have summarized our contribution in the Introduction, here, we focus on how we will extend this work in the future. First, we are currently investigating how to quantify the efficiency and reliability of a subjective quality evaluation method, so that we can estimate the benefit of using the `OneClick` framework instead of a traditional evaluation method like MOS. Second, we are studying how to validate the evaluation results of `OneClick`. We plan to devise a hypothesis-test-like method to decide whether the evaluation results of a certain experiment are *trustworthy*. Third, we will use `OneClick` to evaluate users' perceptions of application quality over time, as we believe the results would shed some light on more sophisticated QoE management. Finally, multimodal QoE studies, such as how to optimize simultaneous game and audio data delivery, will form part of our future work.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 97-2219-E-001-001 and NSC 97-2219-E-011-006. It was also supported in part by the National Science Council of Taiwan under the grant NSC 96-2628-E-001-027-MY3.

References

[1] G. Armitage, "An experimental estimation of latency sensitivity in multiplayer Quake 3," in *11th IEEE International Conference on Networks (ICON)*, 2003.

[2] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "Quantifying Skype user satisfaction," in *Proceedings of ACM SIGCOMM 2006*, Pisa, Itlay, Sep 2006.

[3] K.-T. Chen, P. Huang, and C.-L. Lei, "How sensitive are online gamers to network quality?" *Communications of the ACM*, vol. 49, no. 11, pp. 34–38, Nov 2006.

[4] ——, "Effect of network quality on player departure behavior in online games," *IEEE Transactions on Parallel and Distributed Systems*, 2009.

[5] P. A. Dinda, G. Memik, R. P. Dick, B. Lin, A. Mallik, A. Gupta, and S. Rossoff, "The user in experimental computer systems research," in *ExpCS '07: Proceedings of the 2007 workshop on Experimental computer science*. ACM, 2007, p. 10.

[6] F. Harrell, *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Springer, 2001.

[7] T.-Y. Huang, K.-T. Chen, and P. Huang, "Tuning the redundancy control algorithm of skype for user satisfaction," in *Proceedings of IEEE INFOCOM 2009*, April 2009.

[8] ITU-T Recommandation, "P. 800. Methods for subjective determination of transmission quality," International Telecommunication Union, 1996.

[9] ——, "J. 144. Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference," International Telecommunication Union, 2001.

[10] ——, "G. 107. The E-Model, a Computational Model for Use in Transmission Planning," International Telecommunication Union, 2002.

[11] ITU-T Recommendation P.862, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," Feb 2001.

[12] H. Knoche, H. De Meer, and D. Kirsh, "Utility curves: mean opinion scores considered biased," in *Proceedings of International Workshop on Quality of Service*, 1999, pp. 12–14.

[13] G. Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information," 1956.

[14] J. Nichols and M. Claypool, "The effects of latency on online madden NFL football," in *Proceedings of NOSSDAV'04*, 2004, pp. 146–151.

[15] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2001, pp. 73–76.

[16] A. Watson and M. Sasse, "Measuring perceived quality of speech and video in multimedia conferencing applications," *Proceedings of the sixth ACM international conference on Multimedia*, pp. 55–60, 1998.