Towards Understanding User Tolerance to Network Latency and Data Rate in Remote Viewing of Progressive Meshes

Ransi Nilaksha De Silva, Wei Cheng, Wei Tsang Ooi, and Shengdong Zhao
Department of Computer Science
National University of Singapore
Singapore 117417

ABSTRACT

We conducted experiments with 38 users who interacted with 3 progressively streamed and rendered 3D meshes in order to study their tolerance levels for network data rate and delay. Our study shows that over 90% of users can tolerate a data rate of 80 KBps and above (when the delay is 400ms) and over 95% of users can tolerate delay up to 1 second (when the data rate is 100 KBps). Our study shows that data rate and delay tolerance levels do not vary significantly among the three meshes we used.

Categories and Subject Descriptors

I.3.2a [**Graphics Systems**]: Distributed/Network Graphics; C.2.4b [**Distributed Systems**]: Distributed Applications

General Terms

Human Factors, Experimentation

Keywords

Streaming, Progressive Meshes, Quality of Experience

1. INTRODUCTION

With the availability of sophisticated 3D scanning and sculpting techniques, 3D mesh is emerging as an important new media type, with applications in virtual worlds, virtual art galleries, and on-line antique shops. For instance, efforts have been put into digitizing the sculptures of Michelangelo [16] and Rodin [19]; The Digital Sculpture and SAVE projects from the Virtual World Heritage Laboratory aim to digitize and archive 3D models of thousands of cultural artifacts [8].

While the availability of such high quality 3D meshes has increased, sharing these 3D meshes over the Internet remains a challenge due to the bandwidth requirement. The abundance of portable devices adds to the challenge as they typically do not have the computational power to render high quality meshes. For example, a regular mesh such as the *Thai Statue* (Figure 1, left) has 5 million vertices, occupies 17 MB after compression, and would more than 2 minutes to download at 1 Mbps.

Due to the large size of high quality 3D meshes, users who wishes to view the 3D meshes remotely would have to experience long waiting time before the meshes are fully downloaded. A solution to reduce the waiting time is to encode and transmit the 3D meshes progressively [13]. The user initially receives a coarse base mesh, followed by a series of refinements (called *vertex splits*) that add granularity to the base mesh. The user can thus view a low-quality version of the mesh first, interact with it, while waiting for the mesh quality to improve over time.

Research into building large-scale distributed systems to disseminate high-quality progressive meshes have focused on issues such as designing of stateless server [6], improving scalability through peer-assisted schemes [5], choices of transport protocols [17], and error protection schemes [2]. Existing research, however, has a glaring gap between the quality of service (QoS) provided and the quality of experience (QoE) perceived by the users. Among the factors that affect users' QoE of streaming progressive meshes, network delay and rate of refinement are arguably the two most important ones. Yet, the user tolerance of these factors is poorly understood.

Network delay can affect viewing experience in the following way. When a user changes his or her viewpoint to view the 3D mesh from a different (and never seen before) angle, the request for new refinements to improve the quality of the newly-visible part of the 3D mesh is sent. Network delay causes a delay between the time the viewpoint starts to change to the moment new refinements arrive, which initiates improvement in the mesh quality. Note that, for the case of client/server architecture, the delay introduced is typically dominated by the round trip time between the user and the server. For peer-to-peer architecture, however, the delay introduced consists of multiple delay components, depending on the design of the peer-to-peer protocol.

The rate of refinements is determined by the data rate at which the refinements are sent. A slower sending rate causes slower improvement of mesh quality, lengthening user's waiting time.

Understanding user acceptance level for high network delay and slow data rate while viewing progressive meshes over the network can help us design better systems for streaming progressive meshes in several ways. First, we can better provision the server based on the minimum data rate required to stream a progressive mesh without degrading the user experience. Second, we can design a better overlay network for disseminating progressive meshes, knowing the bounds on network delay that the user can tolerate. Third, we can design better error-resilient protocols by choosing the appropriate forward error correction rate, without violating the acceptable delay and data rate of the users.

Designing a study to measure the user tolerance level to network delay and data rate when viewing a progressively streamed mesh is challenging, since such tolerance levels can depend on many factors, such as the quality of the mesh, progressive coding structure of the mesh, the users' intention when viewing the mesh, and the camera angle and viewing distance between users' eyes and the mesh (which can be affected by zoom level, display size, etc). Even the tolerance level to network delay can depend on the data rate, and vice versa. A full-scale user study would have to vary each of these parameters, leading to a combinatorial explosion of the number of configurations.

While we think that such full scale study is necessary, we opt to start small and focus our study using three carefully chosen meshes with fixed zoom-level in a typical office environment. In our study, we vary the network delay while keeping the data rate constant, and vary the data rate while keeping the network delay constant. While we cannot generalize the results of our study to other scenarios, we believe our study is the first step towards better understanding of the user tolerance to network delay and data rate when viewing a progressively streamed mesh.

Our study of 38 users shows that over 90% of users can tolerate a data rate of 80 KBps or above when the delay is negligibly small (400 ms). Surprisingly, over 95% of users can tolerate delay up to 1 second when the data rate is sufficiently high (100 KBps). Our results also show that tolerance levels do not vary significantly among the tested meshes.

We structure the rest of the paper as follows. We discuss related user studies in Section 2. Section 3 describes our user study. Sections 4 and 5 presents the results of our experiments. We briefly discuss the findings in Section 6 and conclude in Section 7.

2. RELATED WORK

We are not aware of any previous study on user tolerance level to delay and data rate in progressive mesh streaming. There are, however, many studies into user tolerance level to network delay of other multimedia data types.

The user acceptance level of networked delay to voice communication is well known. ITU-T G.114 standard suggests a maximum one way mouth-to-ear delay of 400ms, but a delay of at most 150ms is recommended [1]. For video conferencing, end-to-end delay of 250ms is widely accepted as the acceptable threshold [11]. A landmark study on video and audio synchronization by Ralf Steinmetz evaluates user tolerance level to skews in lip synchronization under various scenarios [22]. A skew between -80ms and +80ms is found to be acceptable to most users.

User tolerance level to network delay in multi-player networked games have also been studied. For first person shooting game, a study on Unreal Tournament 2003 have shown that a network latency of 100ms becomes noticeable to users, while a latency of 200ms becomes annoying [4]. For real-time strategy game, users are found to accept latency up to 800ms, although the gaming experience starts to degrade with latency beyond 500ms [21]. Similar studies have been conducted for sports games, where playability is found to degrade with latency of 500ms in Online Madden NFL Football) [20] and role playing games, where playability is found to degrade with latency of 1.25s in EverQuest 2 [12].

User acceptance level of data rate for watching video clips has also been studied. The data rate of a video is mainly affected by three encoding parameters – resolution, frame rate, and quantization. Apteker et. al studied the acceptability of video at different frame rate for a variety of video clips and found that in general a frame rate of 15 fps is barely acceptable and a frame rate of 5 fps is very unacceptable [3]. McCarthy et. al investigated the trade off between the encoding quality each frame with the frame rate of the video [18]. The study, focusing on sports video, found that users prefer higher quality than higher frame rate. Knoche et. al found that video resolution smaller than 168×126 is unacceptable, especially with the encoding quality is high [15]. In contrast, for first person shooting games, Claypool et. al have found that frame rate has significant effect on player performance, compared to the resolution of the game. A game rendered at the frame rate of 7 fps is almost un-playable, but players still do well at a resolution of 320×240 [7].

3. APPROACH

To determine the user tolerance level to delay and data rate, we conduct the following experiment. We ask users to view a progressive mesh that is streamed from a server under a given delay and data rate configuration. The user is then asked to rate if he or she is satisfied with the streaming quality. We vary the mesh, delay, and data rate for each user. The details of the experiment is described in the rest of this section.

3.1 Meshes

Three 3D meshes are chosen from the freely available Stanford 3D Scanning Repository[1]: *Happy Buddha*, *Dragon*, and *Thai Statue* (see Figure 1). These meshes vary in complexity (amount of vertices), orientation, and symmetry in space from the default viewing direction. *Happy Buddha* is the simplest mesh, is vertically oriented, and has a default viewing direction orthogonal from the face of the Buddha. From that direction, the mesh is asymmetric between front and back. The geometric shape of *Happy Buddha* is somewhat representative of human-like statues. *Dragon* is more complex and is horizontally oriented. The default viewing point is from the left side of the body. The *Dragon* is front-back symmetric relative to the default viewing direction. The geometric shape of *Dragon* is somewhat representative of most mammals. *Thai Statue* is the most complex among the three chosen meshes, composing of three identical sides, each with three different objects: a Goddess, an elephant, and a dragon, stacking vertically from top to bottom. These three sides connect to form a triangular cylinder. The default viewing direction is from a corner of the triangular cylinder. The *Thai Statue* is included as an example of complex compound mesh.

Figure 1: Meshes used in our experiments. Left to right: Thai Statue, Dragon, and Happy Buddha.

Each mesh is converted into a progressive mesh using OpenMesh [2] and encoded using the algorithm described by Cheng and Ooi [6], which is based on the work of Kim and Lee [14]. Each vertex split is encoded with 26 to 40 bits, depending on its level in the mesh. Thus, 1 KB of data roughly corresponds to 200 to 300 vertex splits. Details about each mesh is given in Table 1.

[1]http://graphics.stanford.edu/data/3Dscanrep/
[2]http://www.openmesh.org

Table 1: Properties of meshes used in experiments

Name	Size of Base Mesh (Bytes)	Size of Base Mesh (Vertices)	Size of Refinements (Bytes)	Size of Refinements (Vertex Splits)
Happy Buddha	221 KB	13,436	2.0 MB	530,216
Dragon	247 KB	15,992	12.1 MB	3,593,608
Thai Statue	253 KB	16,384	16.8 MB	4,983,616

3.2 Experimental Setup

The participants view the meshes on a PC with an Intel Core 2 CPU 6600 at 2.4GHz and 4 GB of RAM, equipped with an NVIDIA Quadro FX 3500 graphics card. The experiments are conducted in our research lab, which is similar to a typical cubicle-style office environment. The viewing client connects to a mesh streaming server through a simulated network. We use the network emulation (`netem`) tool to control the network delay and Linux traffic control (`tc`) to cap the data rate.

Each mesh is streamed using UDP over the simulated network using a view-dependent, receiver-driven protocol we previously presented [6]. To make sure that user tolerance levels are not effected by differences in base mesh transmission time, we load all base meshes from the local hard disk. Thereafter, progressive refinements are streamed under controlled network conditions.

3.3 Procedure

Before the user study begins, the participants were briefed about the keyboard commands to view and interact with the 3D meshes and were introduced to the user interface. The users were also given the following instruction:

> For each of the test cases that is given to you, if the streaming quality is sufficient, please rate it as "accept", otherwise, please rate it as "reject."

Each participant is then presented with the experiments in random order (i.e., either the delay experiment first or the data rate experiment first). Within each experiments, the test cases are presented in random order as well.

For each test case, when a participant clicks the "view" button, a new viewing window of size 800×800 pixels is opened and the corresponding 3D mesh is progressively streamed and rendered in the window. The participants can translate or rotate the mesh with six-degrees of freedom until they close the window. They can view the mesh multiple times if needed.

Since distance of the user's viewpoint to the mesh (zoom level) affects the tolerance level to delay and data rate, we fixed the zoom level in our experiments. The user's viewpoint is positioned such that the rendered mesh occupies about 3/4 of the viewing window.

When the participant is ready to rate his or her viewing experience, he or she gives a binary response by clicking on either the "accept" or the "reject" button. The participant is then presented with the next random test case.

The delay and data rate parameters are not revealed to the users.

3.4 Pilot Study

We conducted a set of pilot studies consisting of 5 to 10 users before the user study to find the proper experimentation range for delay and data rate.

We found that most users: (i) rejected all test cases that had data rates below 20 KBps, (ii) cannot distinguish between data rate changes of less than 20 KBps, (iii) cannot distinguish among data rates above 100 KBps, and (iv) rejected all test cases with delay over 4 seconds, (v) cannot distinguish among delays lower than 1 seconds.

Table 2: Delay and data rate values for data rate tolerance experiment.

Test Case	Delay (ms)	Data rate (KBps)
1	400	20
2	400	40
3	400	60
4	400	80
5	400	100

Table 3: Delay and data rate values used for delay tolerance experiment.

Test Case	Delay (seconds)	Data rate (KBps)
1	1	100
2	2	100
3	3	100
4	4	100
5	5	100
6	6	100

3.5 Experiment Parameters

Based on our pilot study, we have identified the range of data rate between 20 to 100 KBps and the delay of 1 second to 6 seconds for our user study. To eliminate the effects of data rate on the tolerance level of delay, and vice versa, we fixed the delay to an acceptable constant of 400 ms (well below the tolerance range based on the pilot study) when we vary the data rate. Similarly, we fixed the data rate at 100 KBps when we vary the network delay.

Since viewing of 3D meshes has a short session length (103 seconds on average for our experiments, comparable with the values reported previously [9]), we do not use the Method of Limits [10] for our experiments. In other words, we do not change the streaming parameters continuously until the user's tolerance level has been reached. Instead, we used multiple streaming sessions (i.e., test cases), and, in each session, the control variable was changed.

The data rate range and delay range chosen are summarized in Tables 2 and 3.

3.6 The Experiments

Participants A total of 22 male and 16 female paid participants, aged 20 to 31, mostly from the university community participated in the experiment. Each participants is compensated monetarily for their time. None had any visual handicaps. Five of the participants have participated in our previous experiment [9] and have some previous experience with streaming mesh.

Normalization The procedure for the experiment is similar to that explained in Section 3.3, except that at the start of the user study, the user is asked to view the three meshes, streamed with a data rate of 120 KBps and delay of 400ms. These network parameters are far better than the test cases, and they were only provided for normalizing differences between user expectations. As mesh

streaming systems are not yet common, it is vital that all users have the same experience of what a good streaming experience should be. Only after viewing the three meshes for normalization, the user is given a random sequence of test cases.

Data Sanitization We assume that if a participant tolerates a test case with data rate of R for a given mesh, then the same participant would tolerate a data rate larger than R when viewing the same mesh. Similarly, if a participant tolerates a delay of d for a given mesh, the same participant would tolerate delay smaller than d when viewing the same mesh. To assure fidelity of the data analysis, if we encounter a response that is inconsistent with the assumption above from a participant for a particular mesh, we simply remove all responses for that mesh by that participant. Sanitization removes 32% of data from both experiments. Since this is not a small number, we also show the pre-sanitized results in this paper.

4. MAIN RESULTS

Figure 2 (a) shows the user acceptance percentage graphs for the data rate experiment. The user acceptance percentage for a specific data rate is calculated by finding the percentage of users who mark that data rate as *acceptable*.

Figure 2 (b) is constructed using the minimum user acceptance percentage of each mesh for each data rate from Figure 2 (a). The figure shows that irrespective of the type of mesh, more than 90% of users can tolerate 80 KBps or above and that more than 80% of users cannot tolerate data rates below 20 KBps. Note that the rate of 80 KBps translates to roughly 16,000 - 24,000 vertex splits per second while the rate of 20 KBps is roughly 4,000 - 6,000 vertex splits per second for our meshes.

Results for the delay experiment are shown in Figures 3 (a) and (b). Figure 3 (a) shows that less than 5% of users can tolerate up to 6 seconds of delay for *Happy Buddha* and no user can tolerate 6 seconds of delay for *Dragon* and *Thai Statue*. A summary graph is shown in Figure 3 (b), which is constructed using the minimum user acceptance percentage for each delay from Figure 3 (a). The plot shows that irrespective of the type of mesh, more than 95% of the users can tolerate delay of 1 second and that about 90% of users cannot tolerate delays more than 5 seconds.

Another observation we had is that user data rate and delay tolerance levels have very little variability over the three different meshes tested, despite their diversity in terms of shapes and complexity. This observation can be seen in trend curves in Figure 2 (a) and Figure 3 (a). More user studies are needed with different meshes before a solid conclusion is reached.

5. ADDITIONAL EXPERIMENTS AND RE-SULTS

5.1 Pre-sanitization

To verify that sanitization of data (Section 3.6) does not affect our main conclusion, we plotted the corresponding graph of Figures 2 and 3 before data sanitization in Figure 4. The figure shows that the general trend of tolerance level is still similar. An interesting observation is that more users rate the data rate of 60 KBps as acceptable than 80 KBps, indicating that perhaps the users cannot significantly distinguish between data rate above 60 KBps.

5.2 Trading off between delay and data rate

We included an additional set of test cases (shown in Table 4), where we varied both the data rate and delay in the study. This set of test cases, called the *trade-off experiment*, is presented with

the other two tolerance level experiments in random order to the participants.

Table 4: Delay and data rate value used for the trade-off experiment

Test Case	Configuration 1 (delay, data rate)	Configuration 2 (delay, data rate)
1	(1s, 20 KBps)	(2s, 40 KBps)
2	(2s, 20 KBps)	(3s, 40 KBps)
3	(1s, 40 KBps)	(2s, 60 KBps)
4	(2s, 40 KBps)	(3s, 60 KBps)

We designed this experiment to find out if participants had a preference for better data rate and worst delay values or lower data rate and smaller delay values, when neither delay nor data rate tolerance levels could be met. In this experiment, participants are asked explicitly to compare, for each test case, two configurations and choose which configuration is preferable. The test cases are presented to the participant in random order. For each test case, the participants are presented with two buttons, a left button, which triggers the streaming and rendering of a mesh using Configuration 1, and the right button, which triggers the streaming and rendering of the same mesh, using Configuration 2. Only one configuration can be active at one time, but the participants can go back-and-forth between the two configurations as many times as they like. Once the participant is ready to decide, he or she can select which configuration they prefer, and they will be presented with the next test case in the user interface. Again, the delay and data rate parameters of the experiments are not revealed to the participants.

The results are plotted in Figures 5 (a)-(c). User preference percentage is calculated by finding the percentage of users who prefer the higher data rate and higher delay values (i.e., Configuration 2 in Table 4) over the lower ones. Our results show that for Test Cases 1 and 2, there is a preference of 50% to 70% in favor of higher valued tuples, where as for Test Cases 3 and 4, the preference is towards lower valued tuples. The preference of one choice over the other is not overwhelming, and shows a more significant dependence on the 3D meshes being viewed compared to the tolerance-level experiments.

6. DISCUSSION

We find that, when the delay is negligibly small, the acceptable data rate is quite high. Even if we conservatively take the rate of 60 KBps, or 480 kbps, as the acceptable threshold, the data rate required is comparable to the current broadband upload bandwidth available in most household using cable modem (512 kbps). This finding points to considerable challenge for deploying peer-to-peer techniques to stream the mesh from another peer, until the upload bandwidth from homes improves. Streaming from multiple peers would be needed to support the high data rate requirement for progressive mesh.

The other finding is more encouraging. We find that when the data rate is sufficiently high, the delay tolerated by the user is about 1 second. This delay is much higher than the tolerable delay in other interactive applications, such as video conferencing and multiplayer games. The reason for such high tolerance, we believe, is the progressive nature of the application. Whenever a user changes the viewpoint, the display would change immediately and render the mesh at the new viewpoint. Any part of the mesh that is never seen before is still rendered, albeit at a lower quality. Thus, even if the refinement starts to appear one second later, we find that most users still find this acceptable. This finding points to more leeway in de-
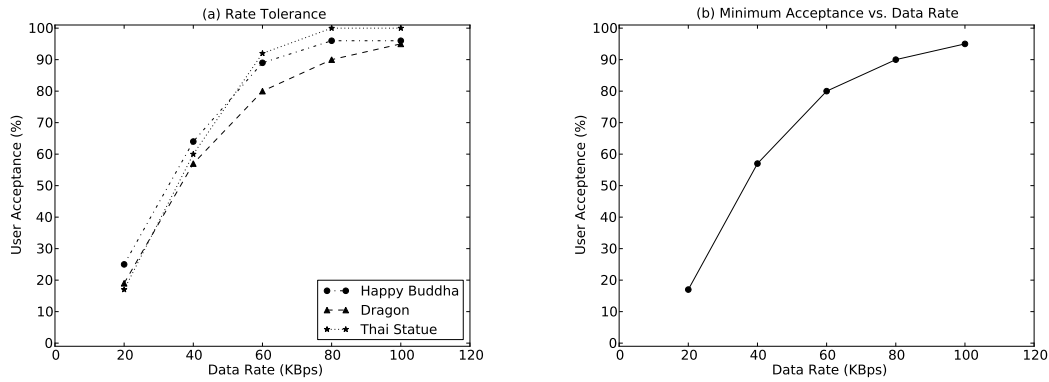
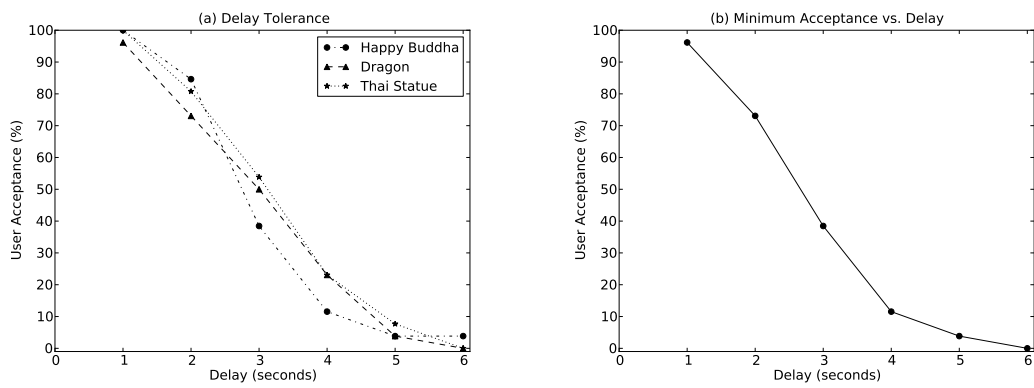**Figure 2: User acceptance for data rate tolerance experiment (with sanitization)**

Figure 3: User acceptance for delay tolerance sub-experiment (with sanitization)

signing of peer-to-peer mesh streaming system, where a requesting peer can spend more time to discover other peers that can supply the mesh data needed, or allow the data to be forwarded multiple hops in an overlay network before reaching the requesting peer.

7. CONCLUSION

Our study is preliminary in nature. Focusing on three carefully chosen meshes, with a fixed zoom-level, we have found out how high can the network delay be in a progressive mesh streaming system before it is deemed unacceptable by users, but assuming that the data rate is acceptably high. Similarly, we have found out how slow can the mesh streaming rate be before it is deem unsatisfactory by users, but assuming that the delay is acceptably low. Our work presented in this paper, while does not explore the full set of parameters, has identified a region in the design space that is acceptable to the user.

More research is certainly needed to understand user tolerance level in other scenarios. The correlation between zoom level, display size, and tolerance level needs to be more carefully explored. User role and intention when viewing the meshes (e.g., a casual museum visitor, an expert archaeologist, or auction bidder) may also affect the expectation of viewing quality and thus their tolerance level. We plan to continue working towards this direction and hope that research community will join us as well.

8. REFERENCES

[1] ITU-T Recommendation G.114 - one-way transmission time.

[2] G. Al-regib, Y. Altunbasak, and J. Rossignac. Error-resilient transmission of 3D models. *ACM Trans. Graph.*, 24(2):182–208, 2005.

[3] R. Apteker, J. Fisher, V. Kisimov, and H. Neishlos. Video acceptability and frame rate. *Multimedia, IEEE*, 2(3):32 –40, Autumn/Fall 1995.

[4] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool. The effects of loss and latency on user performance in Unreal Tournament 2003. In *Proceedings of NetGames '04*, pages 144–151, Portland, OR, Aug. 2004.

[5] W. Cheng, D. Liu, and W. T. Ooi. Peer-assisted view-dependent progressive mesh streaming. In *Proceedings of ACM MULTIMEDIA '09*, pages 441–450, Beijing, China, Oct. 2009.

[6] W. Cheng and W. T. Ooi. Receiver-driven view-dependent streaming of progressive mesh. In *Proceedings of NOSSDAV '08*, pages 9–14, Braunschweig, Germany, June 2008.

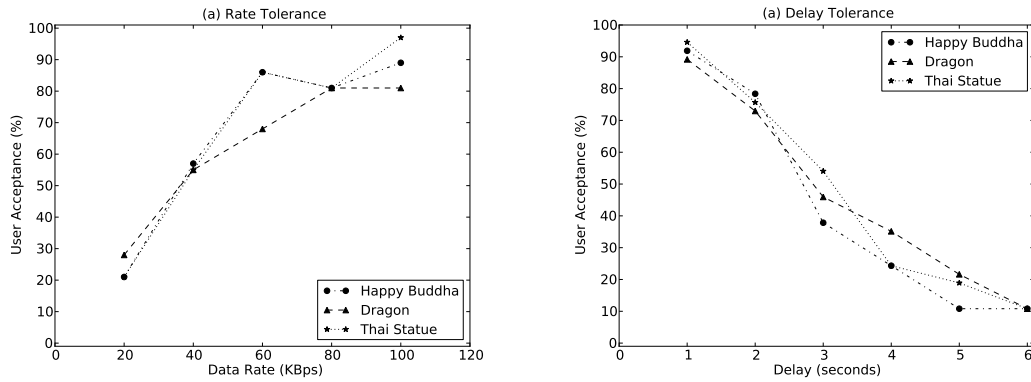[7] M. Claypool, K. Claypool, and F. Damaa. The effects of frame rate and resolution on users playing first person

Figure 4: User acceptance for (a) data rate and (b) delay tolerance experiment (before sanitization)
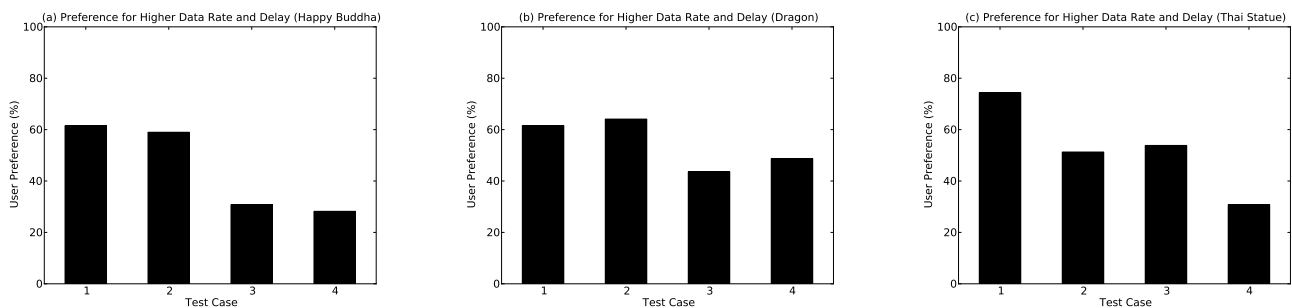

Figure 5: User acceptance when trading-off delay with data rate.

shooter games. In *Proceedings of Multimedia Computing and Networking*, pages 1–11, San Jose, CA, Jan. 2006.

[8] H. de Roos. The digital sculpture project. *Computer and Information Science*, 9(2), 2004.

[9] R. N. De Silva, W. Cheng, D. Liu, W. T. Ooi, and S. Zhao. Towards characterizing user interaction with progressively transmitted 3D meshes. In *Proceedings of ACM MULTIMEDIA '09*, pages 881–884, Beijing, China, Oct. 2009.

[10] G. Fechner. *Elements of Psychophysics: Volume 1*. Holt, Rinehart and Winston New York, 1966.

[11] D. Ferrari. Client requirements for real-time communication services. *Communications Magazine, IEEE*, 28(11):65 –72, Nov. 1990.

[12] T. Fritsch, H. Ritter, and J. Schiller. The effect of latency and network limitations on MMORPGs: a field study of EverQuest2. In *Proceedings of NetGames '05*, pages 1–9, Hawthorne, NY, Oct. 2005.

[13] H. Hoppe. Progressive meshes. In *Proceedings of SIGGRAPH '96*, pages 99–108, New Orleans, LA, USA, Aug. 1996.

[14] J. Kim and S. Lee. Truly selective refinement of progressive meshes. In *Proceedings of the Graphics Interface 2001*, pages 101–110, Ottawa, Ontario, Canada, June 2001.

[15] H. Knoche, J. D. McCarthy, and M. A. Sasse. Can small be beautiful?: Assessing image resolution requirements for

mobile TV. In *Proceedings of ACM MULTIMEDIA '05*, pages 829–838, Hilton, Singapore, Nov. 2005.

[16] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D scanning of large statues. In *Proceedings of SIGGRAPH '00*, pages 131–144, New Orleans, LA, USA, July 2000.

[17] H. Li, M. Li, and B. Prabhakaran. Middleware for streaming 3D progressive meshes over lossy networks. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(4):282–317, 2006.

[18] J. D. McCarthy, M. A. Sasse, and D. Miras. Sharp or smooth?: Comparing the effects of quantization vs. frame rate for streamed video. In *Proceedings of CHI '04*, pages 535–542, Vienna, Austria, Apr. 2004.

[19] D. Miyazaki, M. Kamakura, T. Higo, Y. Okamoto, R. Kawakami, T. Shiratori, A. Ikari, S. Ono, Y. Sato, M. Oya, et al. 3D digital archive of the Burghers of Calais. *Lecture Notes in Computer Science*, 4270:399, 2006.

[20] J. Nichols and M. Claypool. The effects of latency on Online Madden NFL Football. In *Proceedings of NOSSDAV '04*, pages 146–151, Cork, Ireland, June 2004.

[21] N. Sheldon, E. Girard, S. Borg, M. Claypool, and E. Agu. The effect of latency on user performance in Warcraft III. In *Proceedings of NetGames '03*, pages 3–14, Redwood City, CA, May 2003.

[22] R. Steinmetz. Human perception of jitter and media synchronization. *IEEE Journal on Selected Areas in Communications*, 14(1):61–72, 1996.