Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/072L39

Perceptually Driven 3D Progressive Model Selective Transmission over Wireless Network

Bailin Yang[1], *Xun Wang*[2] *and Zhaoyi Jiang*[3]

[1]School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, 310018, China
[2]School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, 310018, China
[3]School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, 310018, China

Received: 12 Oct. 2012, Revised: 26 Dec. 2012, Accepted: 28 Dec. 2012
Published online: 1 Jun. 2013

Abstract: Realtime streaming 3D model over wireless network is a challenging proposition due to the characteristic of 3D progressive geometry data and wireless network. In this paper, we propose a perceptual-driven 3D progressive model selective transmission method. For the progressive model, the data including the base mesh and a series of vertex splits will be packed into several packets. A novel perceptual quality measure for each packet is presented that takes both the saliency information of the 3D model and the rendering dependency among the packets into accounts. For the dropped packets, a selective transmission method is designed in this paper that will resend or ignore these packets according to their perceptual quality. For retransmitting the dropped packets with higher perceptual quality, a priority-based ARQ, together with a scheduling algorithm which considers not only the perceptual importance of the packets but also their enduring time in the retransmitted buffer is applied at the application layer. Simulation result shows that the 3D model received at the client can achieve better rendering effects with lower transmission delay by our transmission method.

Keywords: Perceptually-driven, 3D progressive model, selective transmission

1 Introduction

Recently, 3D models have been widely used in many Internet applications, such as online shopping, online entertainment and distributed simulation. A 3D model can be transmitted and visualized by users properly when the applications are running on reliable network connections. However, when sending the model over lossy network environments, such as the wireless network, some model information may be lost and makes a user fail to visualize the model.

Many methods have been developed to address above problems. They are mainly categorized into model-based methods and network-based methods. Model-based methods usually fix the visual degradation problem by reconstructing the lost data of a model at the client side using certain compensation procedure. Based on this idea, several error-concealment techniques [1,2,3] have been proposed. Although these methods give a better performance by avoiding the transmission of additional error protection bits alongside the model packets or model information retransmission, they still suffer when some of the dependent model information of the received one has been lost or cannot be arrived on schedule. To our knowledge, there is not an effective error-concealment technique providing exact reconstruction of the lost model information. One feasible solution to improve the situation is by minimizing the dependencies among model packets.

Regarding the network-based methods, the Equal Error Protection (EEP) and Unequal Error Protection (UEP) [4] have been proposed to enable the error recovery. Unfortunately, since the above error recovery methods introduce extra error protection information to the model packets, a substantial degradation of transmission performance would probably be resulted. This is not favorable for wireless network. Whats more, there are many TCP-based protocols such as Indirect-TCP(I-TCP)[5], Mobile TCP(M-TCP), TCP with delayed Congestion Response(TCP-DCR)[6], and Wireless TCP(WTCP)[7] have been presented. Unfortunately, all of above methods may cause undesired delay in model rendering, as data retransmission is required when data lost occurs. Automatic repeat request (ARQ) is an efficient method among wireless network

* Corresponding author e-mail: ybl@mail.zjgsu.edu.cn

since it is fast and simple to implement. For the traditional ARQ, it does not take the perceptual importance of the delivered data into account. References [8,9] have proposed the improved ARQ methods that take advantage of the different perceptual importance of each packet contained in a compressed multimedia bitstream. Unfortunately, these methods are targeted for video data, and they are not suitable for 3D model.

In this paper, we propose a perceptual-driven 3D progressive model selective transmission. Before the model transmission, we will build a perceptual quality measurement, which will take the rendering dependencies among the model packets and the visual saliency of the model data into accounts. Guided by this quality measurement, a Priority-based ARQ Transmission mechanism is proposed for optimizing the usage of network bandwidth and decreasing the transmission delay. With this Priority-based ARQ, only parts of the dropped packets will be retransmitted. Also, a priority function is designed for resending the dropped packets in the buffer.

The rest of the paper is organized as follows. Section 2 presents the Perceptual Quality Measure. Section 3 describes the Priority-based ARQ Transmission method. Section 4 shows the experiment results. Finally, section 5 concludes the paper and discusses the future work.

2 Perceptual Quality Measure

The quality of the 3D model transmission over wireless packet network may be impaired in case of packet loss. The amount of quality degradation strongly varies depending on the perceptual importance of the lost data. In this paper, we present a perceptual quality measure for each packet that takes both the saliency information of the 3D model and the rendering dependency among the packets into accounts.

2.1 Saliency Estimation

For 3D geometry model, we can consider some vertices as salient vertices which are distinct from their surroundings in terms of model geometry. Actually, these salient vertices are visually important from the aspect of perceptual quality. While the adhered model is transmitted progressively from the server to client for rendering, these salient vertices should be appeared in the progressive model at the client.

For each vertex of a model, we can estimate its saliency value from a saliency map, which is created by center-surround difference mechanism. Usually, center-surround differences are calculated as an across-scale difference between coarse and fine scales. Specifically, the method for calculating the saliency of each vertex is implemented as follows:

First, the mean curvature *MC* at each vertex *vi* *(i=1...n, n* is the number of vertices of the mesh) is calculated. Whats more, we adopt the reference [10]′s method to choose the neighboring vertex set *NS(vi)* for acquiring enough vertices.

Second, the Gaussian weighted average *GW(Vi)* at different scales is calculated as follows.

$$GW(V_i,r) = \frac{\sum_{x\in W_r} MC(x)\exp\left[-\|x-V_i\|^2/(2r^2)\right]}{\sum_{x\in W_r}\exp\left[-\|x-V_i\|^2/(2r^2)\right]} \quad (1)$$

where r is the radius of sphere *s* whose center is vertex *vi*.

Then, compute the geometry feature map $G_l(m,n)$ at scale *l* between the two scales *m* and *n* for *vi* as

$$G_l(m,n) = |GW(V_i,m) - GW(V_i,n)| \quad (2)$$

Finally, we make use of the non-linear suppression operator to combine the feature maps $G_l(m,n)$ into the final geometry saliency map GF. In our case, we take the mean curvature of each vertex into consideration while combining the above four scales into the final salient region. We also adjust the weight α and β_{l} to get the final geometry feature map using the following formula.

$$GF(V_i) = \alpha\mathrm{N}(MC) + \sum_{l=1}^{4}\beta_l\mathrm{N}(G_l) \quad (3)$$

According to saliency map GF, we can obtain the saliency S of each vertex. Fig.1 depicts the resulted saliency map generated by our method on two different geometry models: Armadillo and Laurana. From the result, it can be seen that our method can obtain not only the block salient areas but clear boundary of these exact salient regions. For instance, we do not only capture the knee on the leg for Armadillo model, but also obtain clear boundary for eyes, mouth and nose of the Laurana model.

2.2 Rendering Dependency Estimation

Progressive Mesh [11], as a good solution to the transmission of 3D models over network, is represented by a base mesh M_0 followed by an ordered list of vertex split $(VSplit)$, which is in the form of $\{M_0,\{VSplit_1,VSplit_2,...,\ VSplit_n\}\}$. There exists dependency relationship among these $VSplit$ operations. In practice, these $VSplits$ will be packed into packets for transmission over network. Hence, these packets also have dependency relationship. Consequently, $VSplits$ could not be rendered unless their dependent $VSplits$ arrived at the client. If some of the received packets are dependent on the lost packet, the client will endure a rendering delay since the lost packet retransmission will be invoked. On the contrary, if no or just a small number of the received packets are dependent on the lost packets,

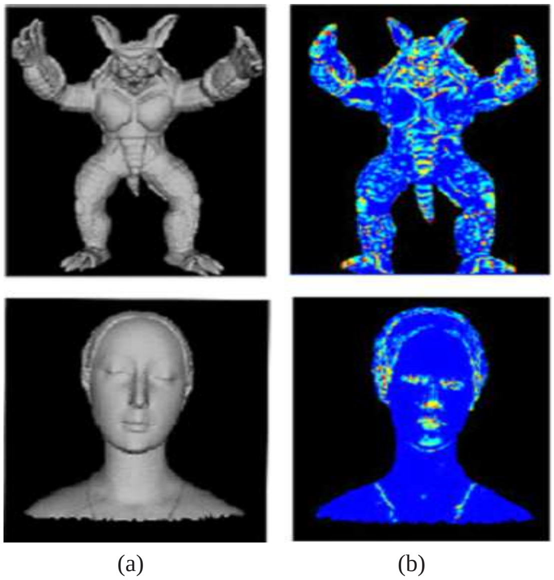


Fig. 1 Saliency Map of the two models:(a) Source model ; (b) Our method.

the client can render more vertices at a given period of time and the delay will be reduced. Therefore, while more packets are dependent on a given packet P, we regard this packet P has higher Rendering Dependency Value (*RDV*).

To obtain the *RDV* of each packet, we will carry out the *counting procedure* that counts the number of packets dependent on packet *P*. Before this *counting procedure*, we should pack the 3D model data into packets with a packetization method we proposed in reference [12]. In this packetization method, two steps will be performed. A Non-Redundant Directed Acyclic Graph (NR-DAG) will be constructed to encode all the necessary dependencies among the *VSplit* operations. Second, a Global Graph Equipartition Packing Algorithm (GGEPA) is applied to divide the NR-DAG into different partitions which are regarded as packets. Fig. 2 shows the dependency graph among packets after the packetization.

In [12]′s method, it just counts the number of edge between partitions (packets) that are connected directly for representing the *dependency value* (*DV*) as shown in Fig. 2(a). For example, the *DV* of packet I is just 2 since there are only packets II and III connected with it. Whats more, the DV of packet IV is 5. From Fig. 2(a), it can be seen that packet I is the root packet, which must include the base mesh of the 3D model. If it is lost during the transmission, the rest model data will not be rendered. For the packet IV, however, while it is lost, only the model data within the four packets including VII, VIII, XI, X will not be rendered. Therefore, the conclusion can be drawn that the *DV* appeared in Fig. 2(a) produced by [12]′s method is not proportional to the importance of packet.

In this paper, we will carry out a *counting procedure* to compute the *RDV* of each packet. Actually, the *RDV* indicates how many packets are dependent on a given packet *P*. In the following, the method to compute the *RDV* is depicted in detail.

For the dependency graph, each node (packet) except the root node has parent nodes. Also, each node except the leaf nodes has children nodes. We will carry out two steps to finish the *RDV*′s counting. In first step, we travel the dependency graph from the root node with the $breadth - first$ search method and count the number of children of each node. Fig.1 (b) shows the result for the example dependency graph. For example, node I which is the root node has two children and nodes VII, VIII, IX, X and VI which are leaf nodes have zero child. In the second step, we travel the graph again to count each nodes offspring. In the implementation, we can travel this graph from the leaf nodes. While we find one nodes parent nodes, the *RDV* of all the corresponding parent nodes will be added one. It can be seen from Fig. 1(c) that nodes pertaining to the upper levels usually have large number. However, in some situations, some nodes pertaining to the lower level still have more dependent nodes than the upper level nodes do. For instance, node IV has 4 offspring but node III just has 3 offspring. It means that more nodes depend on node IV than node III. From the aspect of rendering dependency, we regard node IV is more important than node III.

Based on the above two estimations, one packets saliency and the corresponding *RDV* can be obtained. We can formulate a final perceptual quality (*PQ*) value for

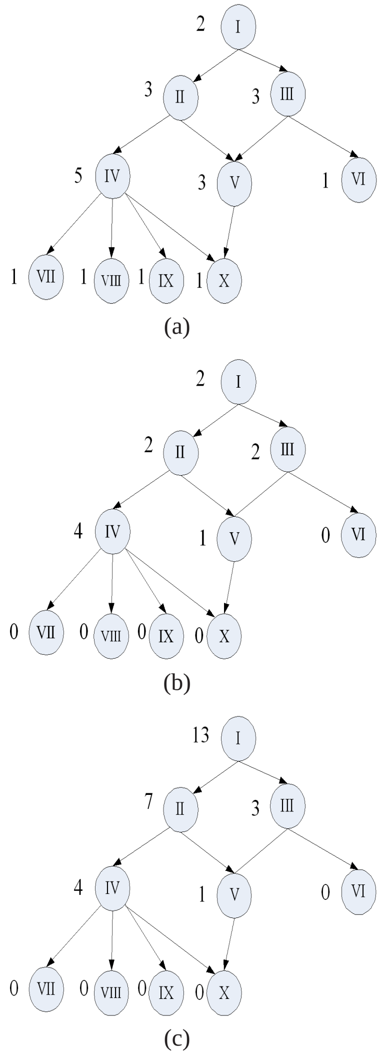


Fig. 2 (a) shows the *DV* for each packet; (b) shows the number of each packets children;(c) shows the *RDV* of each packet.

each packet as

$$I = \kappa_1 N(S) + \kappa_2 N(RDV) \tag{4}$$

where N is a nonlinear normalization operation, and κ_1, κ_2 are the corresponding cost for S and RDV, respectively.

3 Priority-based ARQ Transmission

After the packetization, each packet is assigned with a perceptual quality (*PQ*) value. In order to optimize the usage of network bandwidth and decrease the transmission delay, only parts of the dropped packets will be retransmitted according to their *PQ* values in our method. We apply a priority-based ARQ protocol to retransmit these packets. However, the rest dropped packets with lower *PQ* such as the leaf nodes (packets) as shown in Fig.1(c) will be not processed any more. The technique has been designed to work with the IP/UDP/RTP(RTCP) protocol stack.

The Priority-based ARQ scheme is illustrated as Fig. 3. In this method, each packet is transmitted once from the S_buf, and then it is stored in a retransmission buffer R_buf while its PQ is higher than a certain threshold T. The receiver will generate the ACK or NACK for each packet that is stored in the R_buf . If the packet receives the ACK message, it will be marked as OK since it has been received successfully. Besides, if the remaining number of retransmissions for this packet is less than zero, it will be also marked OK since it will not be retransmitted again. For the packet marked as OK, it will be discarded from the R_buf.

If a NACK is received and the η for its corresponding packet is also larger than zero, this packet will be marked RES in this buffer, which means it should be resent again. Thus, there will be many packets marked RES to be resent in the R_buf. A priority function is computed for each packet marked RES in the R_buf each time a new retransmission opportunity approaches. In our method, we compute this priority function for each packet i as

$$P_i = \lambda_1 N(PQE) + \lambda_2 N(\Delta T_i) \tag{5}$$

$$\Delta T_i = T_c - T_b \tag{6}$$

where ΔT_i is the enduring time the packet in the R_buf. In ΔT_i, T_c is current time and T_b is the time packet entering into the R_buf. In the implementation, the η is initially set 3 and is decremented whenever this packet is retransmitted. From this priority function, it can be seen that the packets that are perceptual important or have been stayed in the R_buf longer will be set high priority value. Whats more, we can adjust theλ_1 and λ_2 to control relative importance of the perceptual and the enduring time terms of the formula. The packets in the R_buf will be sorted by P_i and the one with high priority will be inserted into the S_buf for sending again.

4 Experimental Results

In this section, we compare the end-to-end transmission time and visual quality of the experimental models for the traditional application ARQ (TARQ) method and ours. In the simulations, we used the ns-2[13] network simulator. The simulation layout and the corresponding network parameter are depicted in Fig. 4. In this simulation, an 802.11e MAC layer has been configured between the BS and mobile hosts. In order to set the different loss rate in the PHY layer, the Gilbert Error model is added into NS2. The bit errors generated by this model are introduced to MAC frame.

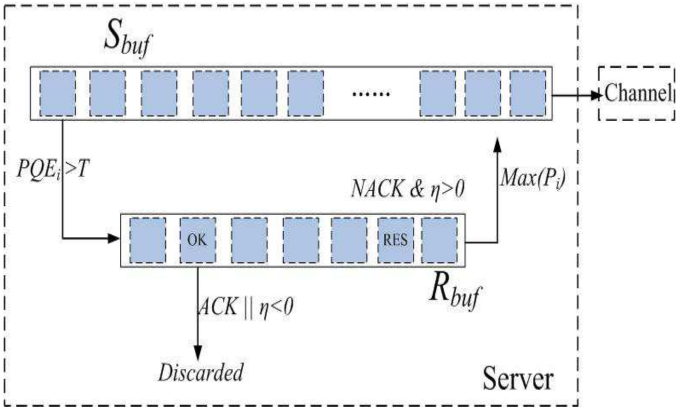


Fig. 3 Scheduling procedure for Priority-based ARQ.

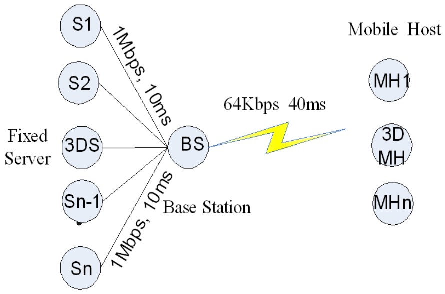


Fig. 4 Simulation layout.

Before the transmission of the model, we adopt the packetization presented above to encode the 3D model with base mesh and a sequence $VSplits$. Each $VSplit$ including topology and geometry information is 30-bytes quantity. Thus, one packet whose max size is 512-byte defined in this paper contains roughly 17 $Vsplits$. In this experiment, we report the experimental result for models Armadillo and Laurana as shown in Fig. 1(a), which will be divided into 172974 $VSplits$ and 23500 $VSplits$, respectively. The number of vertices for the base mesh of both models is roughly occupied 10% percent of the full model.

4.1 End-to-End Delay

Fig. 5 shows the total transmission-rendering time with the simulated loss rate from 0% to 6% for the TARQ and Ours. Comparing with the TARQ, ours takes less time to finish the transmission for the full model. Especially, our method can achieve good result while the loss rate is high. The main reason is that our method just retransmits the dropped packets that have high PQ value and the rest packets will be not resent again.

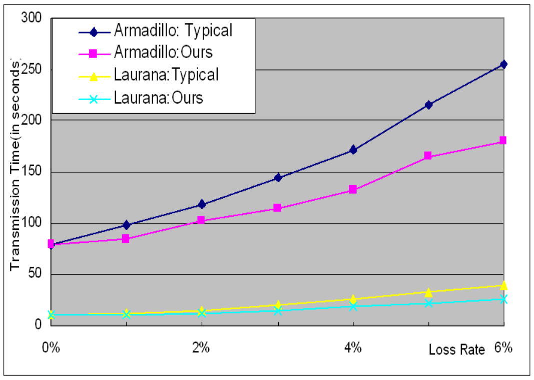


Fig. 5 Total transmission time of the test models when the simulator loss rate is from 0% to 6% for TARQ and ours.

4.2 Visual Appearance

In this section, we firstly present the visual quality of the Laurana progressive model at certain time during the transmission for TARQ and ours. In this case, only parts of 3D model have been transmitted from the server to client. Factually, the received parts at the client for both methods are different. Since only the dropped packets with high PQ value will be retransmitted in our method, more perceptually important vertices will be gathered at the client at certain time in contrast to the TARQ. Therefore, we can obtain good appearance at the client during the transmission. Fig. 10 shows the rendered result for the Lanuara and part of the Armadillo models (the mouth) while about 5% percent model data has been received at the client at 3% loss ratio. Clearly, good rendering result for the perceptual important areas such as the eyes, nose and mouth of the Lanuara model can be obtained by our method. Also, the visual important parts such as the mouth of Armadillo model can obtain more triangles by our method than the TARQ do. Furthermore, we will measure the geometry distortion for the models produced by different transmission mechanism at certain time and loss ratio. In our experiment, we will calculate the mean distance ε_{mn}, root-mean-square distance ε_{rms} and Hausdorff distance H, which works successfully in measuring the quality of the meshes. Because the ε_{mn} and ε_{rms} are not symmetric, i.e., ε_{mn} $(s_0, s_1) \neq \varepsilon_{mn}$ (s_1, s_0) and ε_{rms} $(s_0, s_1) \neq \varepsilon_{rms}$ (s_1, s_0) in general, we will show these values ε_{mn} (s_0, s_1), ε_{mn} (s_1, s_0), ε_{rms} (s_0, s_1), and ε_{rms} (s_1, s_0) respectively.

Table 1 shows above metrics for the Lanuara model at the 9th second while the loss rate is 3%. S1 is the original model, S0 is the model produced by TARQ and ours. The total transmission time of the Lanuara model for TARQ and ours are 20.52 and 15.01 seconds, respectively. From Table 1, it can be seen that the model produced by our method has the smaller ε_{mn}, ε_{rms} , and H values. The conclusion is that the quality degradation of the model

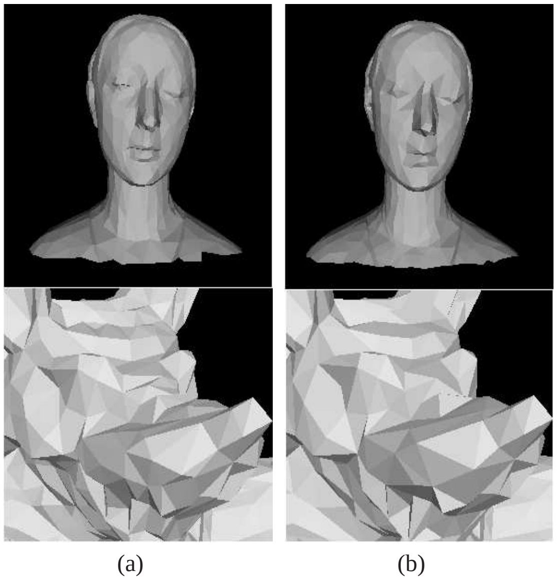


Fig. 6 Rendered result while about 5% percent model data has been received at the client at loss ratio 3%.(a) Ours ; (b) TARQ.

Table 1 Measured ε_{mn}, ε_{rms} , and H between original reference model S_0 and test model S_1

	$\varepsilon_{mn}(S_0,S_1)$	$\varepsilon_{rms}(S_0,S_1)$	$\varepsilon_{mn}(S_1,S_0)$	$\varepsilon_{mn}(S_1,S_0)$	$\varepsilon_{mn}(S_1,S_0)$
TARQ	0.016598	0.08340	0.00085	0.0154	0.97316
Ours	0.00992	0.05236	0.00036	0.0116	0.82709

created by our method comparing with the reference model is the lower than the TARQ transmission mechanisms.

5 Conclusion and future work

In this paper, we have presented a selective transmission method for the 3D progressive model. In order to optimize the usage of network bandwidth and decrease the transmission delay, only parts of the dropped packets will be retransmitted by a priority-based ARQ. To decide the priority of each packet, both the saliency information of the 3D model and the rendering dependency among the packets are considered. From the experimental result, we can see that our method achieve fewer end-to-end delay and higher rendering quality than traditional method. In the future, we will employ this method to process the transmission of textured model, which includes the geometry and the texture image transmission. Evidently, how to design a proper perceptual criteria for the textured model is still a key problem.

Acknowledgement

This work is supported by in part Zhejiang Province Natural Science Fundation for Distinguished Young Scientists (Grant No. LR12F02001), National Natural Science Foundation of China(Grant No.61170214), the Scientific Research Foundation of Zhejiang Provincial Education Department(Grant No.Z201018041), Zhejiang Province Natural Science Foundation (Grant No. Z1101340).

References

[1] S.B. Park and C. S. Kim and S. UK Lee , Error resilient coding of 3-D meshes, Proceedings of International Conference on Image Processing,Volume 1,Barcelona, Spanish, 14-17 (2003).

[2] Z. Yan and S. Kumar and C. C. Jay Kuo, Mesh segmentation schemes for error resilient coding of 3-D graphic models.IEEE Trans, Circuits Syst, Video Techn**15**, 38-144 (2005).

[3] G. Al-regib and Y. Altunbasak and J. Rossignac, Error-resilient transmission of 3D models,ACM Transactions on Graphics **24**, 182-208 (2005)

[4] G. Al-Regib and Y. Altunbasak, An unequal error protection method for packet loss resilient 3-D mesh transmission, Proceedings of IEEE INFOCOM,Volume 2, New York, USA, 23-27 (2002)

[5] A. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts, Proceedings of 15th Int. Conf. Distributed Computing Systems (ICDCS), Vancouver, BC, Canada, 136-143 (1995)

[6] S. Bhandarkar and N. Sadry and A. L. N. Reddy and N. Vaidya,TCP-DCR: a novel protocol for tolerating wireless channel errors, IEEE Trans.Mobile Comput **4**, 517-529 (2004)

[7] P. Sinha and T. Nandagopal and N. Venkitaraman and R. Sivakumar and V. Bharghavan, WTCP: a reliable transport protocol for wireless wide-area networks,Wireless Networks, Volume 8, 231-241 (1999)

[8] Y. Shan. Cross-layer techniques for adaptive video streaming over wireless networks, EURASIP Journal on applied signal processing **2**, 220-228 (2005)

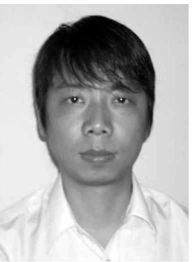
[9] P. Bucciol and E. Masala and E.Filippi and J. C. De Martin, Cross-Layer Perceptual ARQ for Video Communications over 802.11e Wireless Networks, Advances in Multimedia, Volume 2007, 1-1 (2007)

[10] Bailin Yang, Xun Wang, Frederick W.B. Li, Salient Region of Textured 3D Model. Proceedings of 18th pacific conference on computer graphics and application, Hangzhou, China, 25-27 (2010)

[11] Hoppe H. Progressive meshes. In Proceedings of SIGGRAPH **96**, 99-108 (1996)

[12] Bailin Yang, Frederick W.B. Li, Zhigen Pan, Xun Wang, Journal of Computer Science and Technology **6**, 23, (2008)

[13] UCB/LBNL/VINT. Network simulator ns (Version 2). Available on URL: http://www.isi.edu/nsnam/ns/.



Bailin Yang received the Doctor’s degree in department of computer science from Zhejiang Unive-rsity in 2007. He is a associate professor in the department of computer and electronic engineering of Zhejiang Gong-shang University. His researchinterests are in moblie graphics, realtime rendering and mbile game.

Xun Wang received the Doctor’s degree in department of computer science from Zhejiang University. He is a professor in the department of computer and elect-ronic engineering of Zhejiang Gongshang University. His research interests are in multimedia information retrieval, pattern recognition, and mobile networks. and statistical machine learning.

Zhaoyi Jiang graduated from Xian Jiaotong University in 1998, and received the PhD degree in applied mathematics from Zhejiang University in 2007. He is currently an associate professor in the Department of Computer Science, Zhejiang Gongshang University. His current research interests include geometry processing, shape analysis, and mobile graphics.