Automatic Mesh Animation Preview With User Voting-Based Refinement

Zhong Zhou, *Member, IEEE*, Na Jiang, Ke Chen, and Jingchang Zhang

***Abstract*—With the rapid growth in the number and quality of mesh animations, browsing mesh animations wastes considerable bandwidth and requires tremendous rendering resources. Accordingly, the preview technique, which provides users a rapid understanding of a mesh animation before downloading, has received increasing attention. In this paper, we propose an automatic mesh animation preview method that incorporates the interframe motion saliency, intraframe surface saliency, user preference, and camera smoothness constraint to formulate the viewpoint selection as a minimization problem. Then, the minimization is solved by finding the shortest path, and the viewpoints for mesh animation preview are generated accordingly. A voting mechanism is introduced into this process to collect user feedbacks and periodically use user voting feedbacks to refine the preview camera path. The experiment results show that our mesh preview method helps users acquire a good understanding of animation contents. A user study demonstrates that our preview results are superior to those generated by typical preview methods in terms of the subjective visual quality.**

***Index Terms*—Automatic preview, mesh animation, motion saliency, user voting.**

I. Introduction

WITH the development of 3D reconstruction techniques [1], 3D mesh animations [2] and point cloud sequences [3], which are obtained from multiple synchronized cameras or online 3D scanners, have emerged as important resources on the Internet. Today there are a variety of 3D data online publishers that allow users to download models or animations for commercial or academic purposes, such as TurboSquid, The 3D Studio and Berkeley MHAD. As the number and quality of 3D models increase, browsing mesh animations wastes considerable bandwidth and requires tremendous rendering resources. Accordingly, there is an increasing demand for an effective and economical preview to provide users with a rapid understanding of mesh animations before downloading. The preview technique automatically controls the camera and generates some typical view images or videos.

However, most existing preview methods only work on static models and are not suitable for 3D mesh animations. Unlike a static 3D model, a mesh animation is more complex, because it has geometric shape features and inter-frame motion characteristics that bring difficulties in selecting preview viewpoints. Moreover, user subjectivities also significantly affect the visual quality of viewpoints, but unfortunately, few works other than [4], [5] take user preferences into account due to tedious statistics or inefficient feedback.

In this paper, we propose an automatic mesh animation preview method that incorporates the inter-frame motion saliency, intra-frame surface saliency, user preferences and camera smoothness constraint to formulate the viewpoint selection as a minimization problem. Then, the proposed method constructs a weighted directed graph of key frame sampling viewpoints and casts the minimization problem into the problem of finding the shortest path. An initial path comprising the viewpoints of key frame meshes is found using the Dijkstra's algorithm, and then the viewpoints of non-key frame meshes are added to the path with an interpolation between key frame viewpoints. Finally, the viewpoints of key and non-key frames are interleaved to form a mesh animation preview. To collect user preferences, we introduce a voting mechanism, in which a user can vote for good and bad views by simply clicking when watching an animation preview. The proposed method periodically refines the camera path and updates the mesh animation preview video according to user voting feedbacks. The main contributions of this paper are as follows:

1) Propose a novel mesh animation preview method that incorporates the inter-frame motion characteristics, geometric shape features, user preferences and camera smoothness constraint for key frame viewpoint selection, and periodically refines the preview camera path according to user voting feedbacks.
2) Define the inter-frame motion saliency and user preference metrics, in which the inter-frame motion saliency metric measures the significance of movement between adjacent key frames and the user preference metric evaluates the subjective visual quality of a viewpoint.
3) Introduce a voting mechanism that allows users to vote for good and bad views when watching a preview and collects user voting feedbacks for periodical camera path refinement.

Manuscript received February 5, 2016; revised June 19, 2016; accepted September 14, 2016. Date of publication September 21, 2016; date of current version January 17, 2017. This work was supported by the National 863 Program of China under Grant 2015AA016403, and by the Natural Science Foundation of China under Grant 61572061 and Grant 61472020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Maria Martini.

The authors are with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China (e-mail: zz@buaa.edu.cn; jiangna@buaa.edu.cn; chenke19850113@gmail.com; jczhang@buaa.edu.cn).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. This includes a demo video in which the framework of the proposed preview approach and examples of the authors' preview results are displayed. This material is 17.7 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2016.2612124

The remainder of this paper is organized as follows. Section II reviews previous work. Section III presents an overview of the proposed mesh animation preview method. Section IV describes the details of the viewpoint quality metrics. Section V presents the camera path generation of the mesh animation preview. The experiment results are described in Section VI. Finally, Section VII presents the conclusions.

II. Related Work

The technology of 3D model preview is relatively abundant and pivotal in the field of virtual reality, model published online and so on. Recently, most of feasible methods are mainly focused on static 3D model and mesh animation. According to the difference of implementation rationale, the current model preview methods can be classified into three categories: image-based preview, video-based preview and mesh-streaming-based preview.

A. Mesh Animation Preview

Image-based preview method [6], [7] realizes preview through a series of representative thumbnail images, which are 2D view generated by multiple key viewpoints observing model. The method is more commonly adopted when considering low requirements for network bandwidth, nevertheless it has a limitation in information transmission capacity. For mesh animation, it is difficult to reveal the geometric features and motion information vividly. Video-based preview method [8]–[10] needs a predefined camera path, then generated a detailed and juicy preview video of observing model along the camera path. The previews generated by this method usually contain more information than image-based preview. A major drawback of the method is dependent on high bandwidth network which will lead to bad user experience under poor network condition. Mesh-streaming-based preview method [11], [12] also needs set a camera path, whereas the difference between with video-based preview is that it transmits geometric meshes other than video frames through network. Its algorithm design and implementation are very complicated. In order to meet a distortion constraint of 3D mesh streaming, Al-Regib and Altunbasak [13] proposed a new application-layer protocol for 3D mesh streaming. To realize error control techniques at the application layer, Park *et al.* [14] segmented a progressive mesh into smaller partitions that are loss resilient. At the same time, the approach need client with certain graphics computing and long preprocessing. Due to the simplified model and the interpolation reconstruction, the details may be missed.

In conclusion, video-based preview is more simplicity and usability in practical application, considering the bandwidth, information transmission capacity and complexity. Therefore, in this paper we proposed an automatic mesh animation preview on the basic of video-based preview.

B. Viewpoint Selection

Viewpoint selection, which selects typical viewpoints to capture the most salient views of a 3D model, provides some preview images and videos for a user to acquire a general understanding of the mesh model. Most viewpoint selection methods focus on static 3D models and select preview viewpoints based on some viewpoint quality metrics. Kamada and Kawai [6] use an orthographic projection to project the faces of a 3D model according to their normal vectors and then select viewpoints to maximize the projected area. Fleishman *et al.* [7] propose an automatic viewpoint selection method that uses the visible pixels of a 3D model as the viewpoint quality metric and selects viewpoints by solving the art gallery problem. Stoev and Straber [15] incorporate both the projected area and visible scene depth into a viewpoint quality function and then select viewpoints by maximizing the quality function. Their preview approach is more suitable for viewing terrains.

In addition to the projected area and visible scene depth, viewpoint entropy, silhouette length, surface curvature and many other attributes can also be used as viewpoint quality metrics. Based on Shannon's information theory, Vázquez *et al.* [9] propose the viewpoint entropy metric to determine the amount of information seen from a viewpoint and use a greedy algorithm to choose the minimal set of views that captures the maximum information from the scene. Feixas *et al.* [16] define an information channel between a set of viewpoints and the set of polygons of an object, and propose a unified framework for viewpoint selection and mesh saliency. Tao *et al.* [17] consider the information channels between a pool of candidate streamlines and a set of sample viewpoints and use an information based method to select best viewpoints for 3D flow visualization. Hisada *et al.* [18] adopt a three-dimensional Voronoi diagram to extract salient shape regions of a mesh model, such as creases, ridges, and ravines. Lee *et al.* [8] introduce the concept of mesh saliency, which is defined by the Gaussian-weighted average of the mean curvature, and then use an iterative gradient-descent method to find the best views of a mesh model. Saleem *et al.* [10] use the mesh saliency to select representative viewpoints and generate an automatic camera path by solving the TSP problem (Traveling Salesman Problem). Shilane and Funkhouser [19], [20] attempt to find distinctive regions of a mesh model by using a shape-based search, which uses each region as a query into a database containing various shapes of objects. Viewpoint entropy considers the probability of information appearing, but neglects the difference of importance of various parts in the same model. Mesh saliency takes into account perceptibility of 3D objects and distance relationship between human and objects, whereas implementation is very complicated. Silhouette length is suitable for model with sharp silhouette, however the silhouette extraction is complex and controversial problem. To sum up, the evaluation methodology of viewpoint quality has certain one-sidedness and subjectivity, so that using single attribute is not enough to select high quality viewpoint. Therefore, Secord *et al.* [5] propose a perceptual model for viewpoint selection that is a linear combination of area attributes, silhouette attributes, depth attributes, surface curvature attributes and semantic attributes. Except for the perceptual information and geometric analysis, good viewpoints are also affected by subjective factors which depends on user preferences. The works in [4], [5] introduce user preferences to generate the "best" viewpoint for the static models. Vieira *et al.* [4] proposes a machine learning approach for camera placement. Experiments demonstrate that the learning of users' choices is efficient for

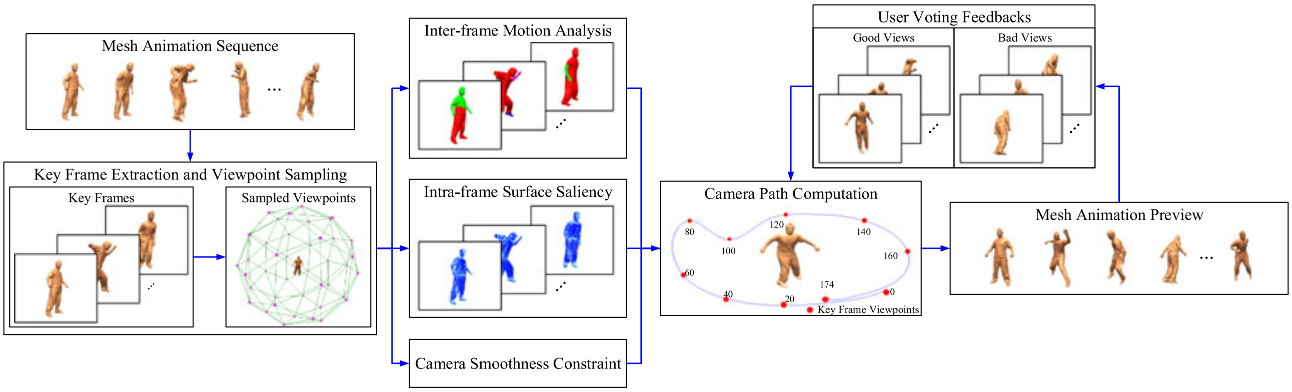


Fig. 1. Overview of the proposed preview approach.

such scenes with occlusion and high depth complexity. However, it requires user interaction on a small set of models with similar objectives. In contrast, [5] designs a "goodness" measure for broad range of models and applications. Firstly, amounts of pairs of views are selected for different model in the form of pictures. Secondly, similar pairs are constantly required to be chosen manually according of literal instructions. This approach can provide appropriate viewpoint selection by a large user study for static models, but not fit for mesh animations. The collection of human preferences is also complicated and the generation of similar pairs needs massive calculation. Though, the introduction of user preferences to the viewpoint metric brings inspiration to our work. For mesh animations, camera path generated by geometric energies don't contain enough semantic view information, sometimes even odd. In view of these above, a voting mechanism is introduced into this paper to collect user feedbacks and periodically use user voting feedbacks to refine the preview camera path.

Compared to a static 3D model, a mesh animation is a sequence of mesh models arranged in time order. Therefore, the viewpoint selection of a mesh animation must consider both the geometric shape features and inter-frame motion characteristics, which is a more complex task than static model viewpoint selection. To date, only a few methods address the problem of automatic camera path generation for 3D animations. Kwon and Lee [21] introduce the concept of motion area, which is the area swept by the motion of a 3D model, and then propose a camera path generation algorithm to select viewpoints by maximizing the motion area. Han *et al.* [22] formulate the camera path generation as a shortest-path problem and solve it by using Dijkstra's algorithm to generate automatic preview videos of mesh sequences. However the proposed algorithm has two severe flaws. The first one is that it only considered the every frame surface information of model and ignored the peculiar motion information of mesh animation. The other one is that preview video is unsmooth along camera path which connected fixed viewpoints.

Existing researches on mesh animations provide worthy references, such as motion exploration techniques [23] and animation synthesis techniques [24], [25]. The idea of using a clustering method to derive near-rigid parts from given mesh model provides suggestions for motion saliency measurement. Inter-frame motion analysis has been used in the automatic camera control of motion capture data visualization [26]–[28]. Inspired by them, we propose an automatic mesh animation preview method that considers the inter-frame motion characteristics, geometry shape features, user preferences and camera smoothness constraint. Meanwhile, our method collects user feedbacks. With increasing user feedbacks, the updated preview will meet the public preference of the vast majority of users.

III. Overview of Our Approach

The preview of a mesh animation is a camera path (a set of viewpoints) that describes the camera location, looking-at vector and field of view for each frame. A mesh model M is a triangular mesh that consists of a set of vertices $\mathbb{V} = \{v_1, v_2 \ldots v_N\}$ and a set of faces $\mathbb{F} = \{f_1, f_2 \ldots f_Q\}$. A mesh animation $\mathbb{M}$ is a sequence of mesh models, i.e., $\mathbb{M} = \{M_1, M_2 \ldots M_L\}$, where L is the length of $\mathbb{M}$. Given a mesh animation $\mathbb{M}$ and a set of sampling viewpoints $\mathbb{P}$, the goal of the preview is to automatically generate a camera path that is a series of camera positions $\mathbb{C} = \{c_1, c_2 \ldots c_L\}$.

As stated in [29], [30], the construction of a camera path is treated as an optimization problem that maximizes or minimizes a viewpoint quality function composed of several scene attributes. Therefore, when the key frame mesh model K_r is given, we spare no effort to find the viewpoint c_r that captures the most expressive part of K_r while keeping the trajectory of the camera path smooth. Fig. 1 shows the overview of the proposed preview approach. Table I summarizes the notations used in this paper.

The input of our preview approach is a mesh animation sequence $\mathbb{M}$ that consists of a series of mesh models. We first extract key frames $\mathbb{K} = \{K_1, K_2 \ldots K_R\}$ from the mesh sequence $\mathbb{M}$ and distribute sampling viewpoints for each key frame. The sampling viewpoints $\mathbb{P}$ are a finite number of viewpoints on a given viewing sphere. Next, we conduct an inter-frame motion analysis on the extracted key frames and define a motion saliency metric to evaluate the motions between adjacent key frames. We also define the intra-frame surface saliency (static surface saliency) metric, which is based on the mesh geometry of a single frame instead of multiple frames. The initial viewpoint energy function incorporates the inter-frame motion

TABLE I
SUMMARY OF NOTATIONS

Symbol	Description
$\mathbb{M}$	Sequence of Mesh Models
$\mathbb{V}$	Set of Vertices
$\mathbb{F}$	Set of Faces
Q	Number of Faces
N	Number of Vertices
L	Length of Mesh Sequence
$\mathbb{P}$	Set of Sampling Viewpoints
$\mathbb{C}$	Series of Camera Positions
$\mathbb{K}$	Set of Key Frames
K_r	*r*-th Key Frame of Mesh Model
$\sum_v$	Variable Segmentation of Key Frames
T_i^j	Best Rigid Transformation of Segment
d_i^j	Displacement Distance
φ_i^j	Rotation Angle
$w(K_i^j)$	Motion Weight of Segment K_i^j
η	Contribution Factor (set to 0.5)
Ω_p	Set of Visible Faces from *p*
E_M	Inter-frame Motion Saliency Energy
R	Number of Key Frames
p_r^i	*i*-th Sampling Viewpoint of Key Frame *r*
$\theta_{i\,,} A_i$	Apex Angle, Area of *i*-th Triangle Incident on *v*
ψ_p	Set of Visible Vertices from *p*
E_S	Intra-frame Surface Saliency Energy
u_k	User Vote
$\vec{v}_{u_k}, f_{u_k}$	Viewing Direction, Frame Number of Voted Viewpoint
w_{u_k}	User Personal Preference
θ_0	Spatial Impact Range of User Vote (set to $\pi/6$)
$3\sigma_0$	Time Impact Range of User Vote ($\sigma_0 = 15$)
E_P	User Preference Energy
r_v	Radius of Viewing Sphere
α	Field of View for Sampling Viewpoints
E_C	Camera Smooth Constraint Energy
$\mathbb{E}$, $\mathbb{E}'_R$	Viewpoint Quality Energy
$\mathbb{E}_I$	Initial Viewpoint Quality Energy
c_1, c_2, c_3, c_4	Weighting Coefficients of Various Energy Terms
S	Number of Nodes (Sampling Viewpoints)

saliency and intra-frame surface saliency with a camera smoothness constraint, which is related to the distance between two adjacent key frames' viewpoints. The smoothness constraint prevents abruptly large camera movements in a short time. We formulate the camera path construction as a minimization problem and then find an optimal camera path by minimizing the energy function. Then, we interpolate viewpoints of non-key frames and finally generate a preview for the mesh animation sequence $\mathbb{M}$.

The mesh preview is then provided to a variety of users for watching. When watching the preview, a user can mark good and bad views by performing a simple click. The user is not required to vote for each frame, and he/she only marks certain frames according to personal preference. We record voting feedbacks for each user and periodically use user feedbacks to refine the mesh preview. To evaluate the history of user feedbacks, a user voting metric is defined and added to the previous viewpoint energy function. We compute the viewpoints of key frames by minimizing the new energy function, which includes four energy terms: the inter-frame motion saliency, intra-frame surface saliency, user voting metric, and camera smoothness constraint. Then, the viewpoints of non-key frames are interpolated, and the user voting refined preview is generated. The refined preview is also released to collect new user feedbacks for further improvement. The refinement process is iteratively invoked after a period of user voting collection. As the amount of user voting feedbacks increases, the user voting refined preview not only captures the significantly salient part of the mesh sequence but also satisfies the general preference of various users.

IV. Viewpoint Quality Metrics

The visual effects of automatic mesh animation preview highly depends on the quality of viewpoint selection. High-quality viewpoint is supposed to reflect abundant information of model animation. Motion information between frames and geometric features are very crucial from the spatial-temporal view. Therefore, we proposed metric of motion saliency and surface saliency to choose decent viewpoints. Moreover, considering that automatic preview is subjectivity, the impact of user preferences is supposed to be introduced. Therefore, we use voting metric to optimize the obtained result through the users voting results.

A. *Inter-frame Motion Saliency Metric*

The inter-frame motion saliency measures the significance of vertex movement between adjacent key frames. Before computing the motion saliency for a key frame, motion analysis and segmentation are the most critical procedures. For these purposes, we employ the segmentation algorithm proposed by Arcila *et al.* [31], [32] to decompose a key frame mesh into several rigidly moving components. A variable segmentation $\sum_v$ of key frames $\mathbb{K} = \{K_1, K_2 \ldots K_R\}$ is a set of segmentations $\sum_v^i = \{K_i^1, K_i^2 \ldots K_i^{q_i}\}$ of each mesh K_i, in which the number q_i of sub-meshes may vary with all segmentations $\sum_v^i$. The variable segmentation allows two neighboring parts of a mesh with different rigid motions to be firstly put into different sub-meshes and later merged into a single sub-mesh when they have the same motion. It is helpful to describe different motion information at each key frame.

The variable segmentation algorithm decomposes key frames into components that move rigidly over time without any prior knowledge. The algorithm consists of two phases: matching and segmentation. Given key frame meshes K_i and K_{i+1}, the matching phase builds a bidirectional mapping from vertices v_i^j in K_i to vertices v_{i+1}^k in K_{i+1}. It deforms mesh K_i to fit mesh K_{i+1} by using the mesh registration method of Cagniart *et al.* [33]. The mapping from K_i to K_{i+1} is created by finding the closest vertex in K_{i+1} for each vertex in the deformed mesh of K_i. The mapping from K_{i+1} to K_i can be constructed in a similar manner. According to the bidirectional mapping between K_{i-1} and K_i, the segmentation $\sum_v^{i-1}$ of K_{i-1} provides an initial segmentation for K_i. Then, based on Horn's method [34], the segmentation phase estimates the motion of each vertex v_i^j in K_i as well as each segment in the initial segmentation. Next, neighboring segments that have similar motions are merged. After that, the segmentation phase uses a motion-based spectral clustering to refine the segments and yields the final segmentation $\sum_v^i$ of K_i. We use the Bouncing sequence as an example to show the segmentation results. As Fig. 2 shows, each key frame mesh is decomposed into several moving segments denoted by different colors.

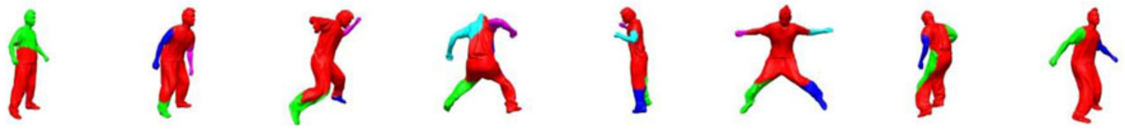


Fig. 2. Segmentation results for the Bouncing sequence.

Each segment K_i. Then, based on Horn's method [29], the segmentation phase estimates the motion of each vertex v_i^j in K_i as well as each segment in the initial segmentation. Next, neighboring segments that have similar motions are merged. After that, the segmentation phase uses a motion-based spectral clustering to refine the segments and yields the final segmentation $\sum_v^i$ of K_i. We use the Bouncing sequence as an example to show the segmentation results. As Fig. 2 shows, each key frame mesh is decomposed into several moving segments denoted by different colors.

Each segment K_i^j is associated with a 4 × 4 matrix T_i^j that represents the best rigid transformation of the segment. To evaluate the movement of segment K_i^j, we compute the displacement distance d_i^j ($d_i^j \geq 0$) and the rotation angle φ_i^j ($0 \leq \varphi_i^j \leq \pi$) of matrix T_i^j according to the relation between the transformation matrix and quaternion. The φ_i^j is a scalar value. Then, the motion weight of segment K_i^j is defined as

$$w\left(K_i^j\right) = \eta \cdot \frac{d_i^j}{d_i^{\max}} + (1-\eta) \cdot \frac{\varphi_i^j}{\pi} \tag{1}$$

where $d_i^{\max} = \{d_i^1,\, d_i^2 \dots d_i^{q_i}\}$ is the maximum displacement distance of segments in $\sum_v^i$ and η indicates which factor contributes more to motion weight between displacement distance and rotation angle. η is typically set to 0.5 since the displacement distance and rotation angle are considered equally important to segment motion.

Based on the motion weight, the inter-frame motion saliency *MS(*K_r*,* O_M*)* is defined as the weighted area of a key frame mesh. *MS(*K_r*,* O_M*)* is calculated by

$$MS\left(K_r,\, O_M\right) \;= \int_{f\in\, O_M} w\left(K_r^f\right)dA \tag{2}$$

where O_M denotes a set of faces and K_r^f denotes the motion segment to which face *f* belongs. When calculating the weighted area of a key frame mesh, $O_M = F_r$. F_r denotes the face set of mesh K_r. When evaluating the motion information captured by a single view *p* in key frame K_r, $O_M = \Omega_p$ $(\Omega_p \subset F_r)$. $MS(K_r, \Omega_p)$ is computed by

$$MS\left(K_r, \Omega_p\right) = \int_{f\in\Omega_p\subset F_r} w\left(K_r^f\right)dA \tag{3}$$

where Ω_p denotes the set of visible faces from *p*. Incorporating $MS(K_r, F_r)$ and $MS(K_r, \Omega_p)$, we define the inter-frame motion saliency energy E_M of a mesh preview as

$$\begin{aligned} E_M \;&=\; \sum_{\mathrm{r}=1}^{R}\left(MS\left(K_r, F_r\right) - MS\left(K_r, \Omega_{p_r}\right)\right) \\ &=\; \Sigma_{\mathrm{r}=1}^{R} D_M\left(K_r, p_r\right) \end{aligned} \tag{4}$$

where *R* is the key frame number, p_r denotes the viewpoint of the *r*-th key frame, and $D_M(K_r, p_r)$ denotes the difference between $MS(K_r, F_r)$ and $MS(K_r, \Omega_{p_r})$, i.e., the weighted area of invisible faces from viewpoint p_{r}. A higher motion salient viewpoint results in a smaller weighted area of the invisible faces and thus lowers $D_M(K_r, p_r)$. To gain more motion information from a mesh sequence, we must find a set of preview viewpoints to make E_M as low as possible.

B. Intra-frame Surface Saliency Metric

According to the research of Hoffman and Singh [35], human visual systems are sensitive to regions of high curvature, which also makes sense for 3D mesh models. High curvature parts of a mesh are typically considered as salient surface regions [36], which significantly attract a user's visual attention. Curvature is a primary geometric property of a mesh model and provides a measure of geometric complexity. The Gaussian curvature at a vertex *v* is calculated by using the Gaussian curvature discrete operator proposed by Meyer *et al.* [37] as follows:

$$C\left(v\right) = \frac{2\pi - \Sigma_i\, \theta_i}{\Sigma_i\, A_i} \tag{5}$$

where θ_i and A_i respectively denote the apex angle and area of the *i*-th triangle incident on *v*, i.e., the *i*-th triangle in *v*'s one-ring neighborhood. Fig. 3 visualizes the magnitude of the Gaussian curvature for each key frame mesh in the Bouncing sequence.

As Fig. 3 shows, the highlighted regions are the regions of high curvature, i.e., the geometric salient regions, which reflect the shape features of the key frame mesh models. Considering the geometric salience, a good view must capture as much of the geometric complexity of a mesh as possible. Based on the Gaussian curvature, we define the function of intra-frame surface saliency $SS(K_r, O_S)$ as

$$SS\left(K_r, O_S\right) = \int_{v\in O_S} C\left(v\right)dv \tag{6}$$

where O_S denotes a set of vertices. When calculating the Gaussian curvature of a key frame mesh, $O_S = V_r$. V_r denotes the vertex set of mesh K_r. When measuring the geometric saliency information captured from a given sampling viewpoint *p* in key frame K_r, $O_s = \Psi_p$ $(\Psi_p \subset V_r)$. $SS(K_r, \Psi_p)$ is computed by

$$SS\left(K_r, \Psi_p\right) = \int_{v\in\Psi_p\subset V_r} C\left(v\right)dv \tag{7}$$

where Ψ_p denotes the set of visible vertices from the p. We incorporate $SS(K_r, V_r)$ and $SS(K_r, \Psi_p)$ to construct the intra-frame surface saliency energy E_S of a mesh preview, which is

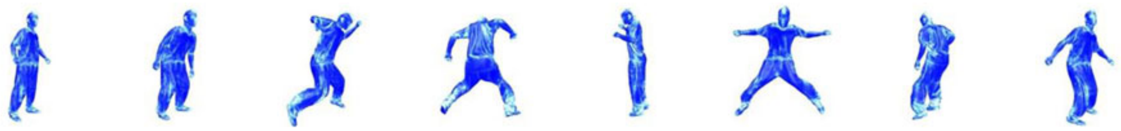


Fig. 3. Gaussian curvature distribution for the key frame meshes in the Bouncing sequence.

calculated by

$$\begin{aligned} E_S &= \sum_{r=1}^{R} \left(SS\left(K_r, V_r\right) - SS\left(K_r, \Psi_{p_r}\right) \right) \\ &= \Sigma_{\mathrm{r}=1}^{R} D_S\left(K_r, p_r\right) \end{aligned} \tag{8}$$

where *R* is the key frame number, p_r denotes the viewpoint of the *r*-th key frame, and $D_S(K_r, p_r)$ denotes the difference between $SS(K_r, V_r)$ and $SS(K_r, \Psi_{p_r})$. $D_S(K_r, p_r)$ represents the sum of the Gaussian curvatures of invisible vertices from viewpoint p_r. A higher surface salient viewpoint captures more geometric information of a mesh and therefore leads to a lower value of $D_S(K_r, p_r)$. Considering the geometric salience of a mesh sequence, the preview viewpoints must keep E_S at the lowest possible value.

C. User Voting Metric

Beyond motion characteristics and shape features, user feedbacks, which are collected from people of diverse attributes, also have a significant impact on preview viewpoint selection. The user feedbacks reflect the visual aesthetic quality of a mesh preview, because different views are not equally effective in revealing shapes and people express clear preferences for certain views over others. We introduce a voting mechanism to collect user feedbacks and leverage user preference as a complementary attribute to guide key frame viewpoint selection. In our voting mechanism, when watching a mesh preview, a user can freely cast positive or negative votes for any frames (including key frames and non-key frames) depending on his/her personal preference. A user vote u_k is defined by a triple $\{\vec{v}_{u_k}, f_{u_k}, w_{u_k}\}$, in which $\vec{v}_{u_k}$ and f_{u_k} denote the viewing direction and frame number of the user voted viewpoint, respectively, and w_{u_k} relies on user personal preference. For simplicity, the value of w_{u_k} is calculated by

$$w_{u_k} = \begin{cases} 1, & \text{for a positive vote} \\ -1, & \text{for a negative vote} \end{cases} \tag{9}$$

User votes significantly affect the preference importance of sampling viewpoints. The importance of each sampling viewpoint in each key frame can be distinguished by considering the spatial and time impacts of each user vote. For simplicity of illustration, Fig. 4 uses a 2D example to show the spatial impact $I_s(u_k, p)$ of user vote u_k on a sampling viewpoint *p*. The value of $I_s(u_k, p)$ strongly relies on the angle between $\vec{v}_{u_k}$ and $\vec{v}_p$, and $I_s(u_k, p)$ is defined by

$$I_s\left(u_k, p\right) = \begin{cases} \cos\left(\frac{\arccos\left(\vec{v}_{u_k} \cdot \vec{v}_p\right)}{\theta_0} \cdot \frac{\pi}{2} \right), & \vec{v}_{u_k} \cdot \vec{v}_p > \cos\theta_0 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

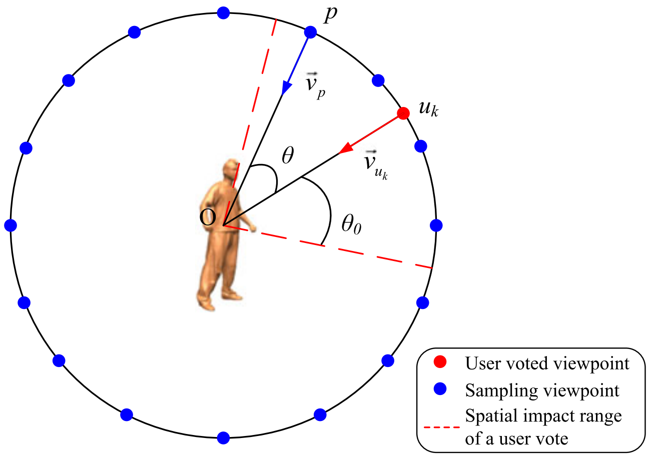


Fig. 4. Spatial impact of a user vote.

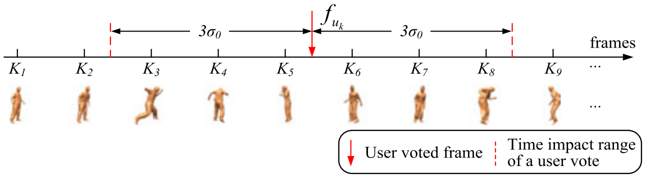


Fig. 5. Time impact of a user vote.

where $\vec{v}_{u_k}$ and $\vec{v}_p$ denote the viewing directions of user vote u_k and sampling viewpoint *p*, respectively, and θ_0 denotes the spatial impact range of a user vote. According to the 30-degree rule proposed by Corrigan *et al.* [38], if the camera position changes by less than 30 degrees, the difference between the two shots will not be sufficiently substantial, which also makes sense for determining the spatial impact range. By considering the view difference, θ_0 is typically set to $\pi/6$ (i.e., 30 degrees).

A user vote also has a time impact range and only affects the preference importance of key frames inside the range. As Fig. 5 shows, the time impact range of u_k is the period from frame $f_{u_k} - 3\sigma_0$ to frame $f_{u_k} + 3\sigma_0$, where f_{u_k} denotes the frame number of the user voted viewpoint. Inspired by the Gaussian function, which is centered at the origin with a decay length set to a specific radius, we define the time impact $I_T(u_k, K_r)$ of u_k on key frame K_r as

$$I_T\left(u_k, K_r\right) = \begin{cases} \exp\left(-\frac{\left(f_{K_r} - f_{u_k}\right)^2}{2\sigma_0^2} \right), & \left|f_{K_r} - f_{u_k}\right| \leq 3\sigma_0 \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where f_{K_r} and f_{u_k} denote the frame numbers of key frame K_r and user vote u_k, respectively, and $3\sigma_0$ limits the time range of a user vote impact. A frame interval σ_0 mesh is set to 15, because apparent shape changes of mesh sequence can typically be observed after nearly 15 frames. Therefore, we empirically set $3\sigma_0$ as three times the frame interval in order to undergo significant changes.

Based on the spatial and time impacts of user votes, we collect and count the feedbacks of every frame, no matter key frames or non-key frames. Let O_P be a set of sampling viewpoints. We define the preference importance of key frame K_r as $PI(K_r, O_P)$ as follows:

$$PI\left(K_r, O_P\right) = \Sigma_{p \in O_P} PI\left(K_r, p\right). \tag{12}$$

When calculating the total importance of all its sampling viewpoints in key frame K_r, $O_P = \mathbb{P}$. $\mathbb{P}$ is the sampling viewpoint set of key frame K_r. When $O_P = p$, the preference importance *PI*(K_r, *p*) of sampling viewpoint *p* in key frame K_r denotes the accumulative impact of user votes. *PI*(K_r, *p*) is calculated by

$$PI\left(K_r, p\right) = \Sigma_{u_k \in U} I_T\left(u_k, K_r\right) \cdot I_s\left(u_k, p\right) \cdot w_{u_k} \tag{13}$$

where $U = \{u_1, u_2, \ldots, u_{i,} \ldots\}$ denotes the set of all user votes and w_{u_k} is the preference value of user vote u_k. By integrating *PI*(K_r, *p*) and $PI(K_r, \mathbb{P})$, we construct the user preference energy E_P of a mesh preview, which is defined as

$$\begin{aligned} E_P &= \sum_{\mathrm{r}=1}^{R} \left(PI\left(K_r, \mathbb{P}\right) - PI\left(K_r, p_r\right)\right) \\ &= \Sigma_{\mathrm{r}=1}^{R} \left(D_P\left(K_r, p_r\right)\right) \end{aligned} \tag{14}$$

where *R* is the key frame number, p_r denotes the viewpoint of the *r*-th key frame, and $D_P(K_r, p_r)$ denotes the difference between $PI(K_r, \mathbb{P})$ and *PI*(K_r, p_r). The key frame preview viewpoint selection must take care of the preferences of the majority of users. A key frame preview viewpoint must receive the impact of positive user votes as much as possible, whereas it must receive the impact of negative user votes as little as possible. Thus, the mesh preview prefers the sampling viewpoints that lead to lower values of $D_P(K_r, p_r)$, i.e., a set of preview viewpoints with lower E_P.

V. Camera Path Generation of Mesh Animation Preview

A. Key Frame Extraction and Viewpoint Sampling

For a mesh animation $\mathbb{M}$, which consists of a sequence of mesh models, a set of mesh models $\mathbb{K} = \{K_1, K_2 \ldots K_R\}$ ($\mathbb{K} \subset \mathbb{M}$) need to be selected as key frames in the preparation stage. In earlier work, key frames are selected by sampling video frames randomly or uniformly at certain time intervals [39]. This method is fast and simple but ignores the video contents. As a consequence, representative frames maybe replaced by some redundant frames. In order to deal with the problem, three categories of approaches have been put forward. There are curve simplification based algorithms, clustering based methods and matrix factorization, respectively. Considering that only using any method alone will bring inevitable shortcomings, we use the composite way proposed by Halit *et al.* [40], which combines curve simplification with clustering, to select expressive frames. According to statistics, about 7% of the frames can be selected to be key frames for every mesh sequence.

The sampling viewpoints of a key frame mesh are distributed onto a viewing sphere around the mesh model. As Fig. 6(a) shows, the radius r_v of the viewing sphere is determined by the viewing frustum of a sampling viewpoint and the bounding sphere of the mesh model. r_v is computed by

$$r_v = \frac{r_b}{\cos\left(\alpha/2\right)} \tag{15}$$

where r_b denotes the radius of the bounding sphere and α is the field of view for sampling viewpoints.

To place sampling viewpoints on a sphere in a nearly uniform manner, regular icosahedron subdivision is applied, which is the closest approximation for a uniform sphere discretization. A regular icosahedron consists of 12 vertices and 20 equilateral spherical triangles. For further triangulation, we subdivide each triangle into multiple smaller spherical triangles, that is, project the midpoints of the three sides onto the sphere and connect the projection points and the three sides' endpoints to yield four smaller triangles. This subdivision process may repeat several times until the desired surface resolution is obtained. Fig. 6(b) illustrates the discrete spherical surface by a level-1 icosahedron subdivision. The vertices (a total of 42 vertices) on the subdivision surface are chosen as the sampling viewpoints on the viewing sphere. In the subsequent process, these candidate viewpoint will be used to chosen the optimal path under the constraints of energy function.

B. Camera Path Computation and Refinement

A sampling viewpoint at a key frame mesh K_r is considered to be more important if it has larger inter-frame motion saliency, larger intra-frame surface saliency or high user preference. Therefore, we define the initial viewpoint quality energy $\mathbb{E}_I$ as a linear combination of the inter-frame motion saliency energy E_M, intra-frame surface saliency energy E_S, user preference energy E_P and camera smooth constraint energy E_C. More formally, the initial viewpoint quality energy $\mathbb{E}_I$ is described as

$$\mathbb{E}_{\mathbb{I}} = c_1 E_M + c_2 E_S + c_3 E_P + c_4 E_C \tag{16}$$

where c_1, c_2, c_3 and c_4 are the weighting coefficients of the various energy terms. They represent the importance of four types of energies. Increasing c_1 would increase the importance of the sampling viewpoints, which capture more motion information regarding a mesh sequence. Increasing c_2 would cause the camera to focus on the mesh parts with more geometric complexities. Increasing c_3 would make the preview path be closer to the user's preferences. The user preference energy E_P can be computed by (14). The initial camera path is a special case, since there are no user voting feedbacks, $E_P = 0$. c_4 is the weighting of camera smooth energy E_C, and E_C is computed

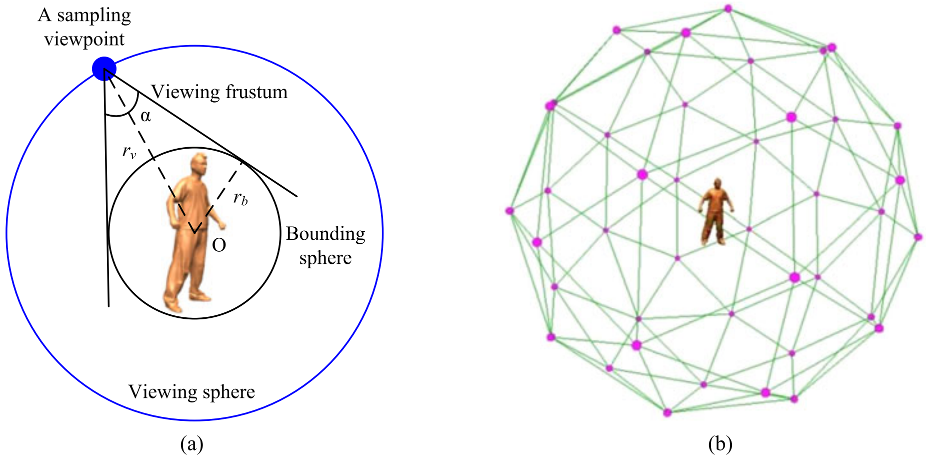


Fig. 6. Sampling viewpoints of a key frame mesh: (a) viewing sphere of a key frame mesh and (b) sampling viewpoints on the subdivision surface of the viewing sphere.

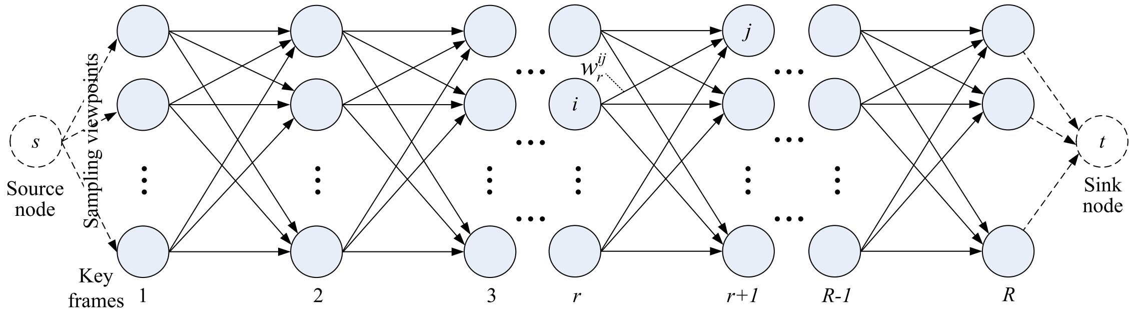


Fig. 7. Sampling viewpoint graph of key frame meshes.

by

$$E_C = \Sigma_{\mathrm{r}=1}^{R-1} \|p_r - p_{r+1}\|_2 = \Sigma_{\mathrm{r}=1}^{R-1} D_C\left(p_r, p_{r+1}\right) + \theta \qquad (17)$$

$$\theta = \arccos\left(\frac{\overrightarrow{p_{\mathrm{r}-1}p_r} \cdot \overrightarrow{p_r p_{\mathrm{r}+1}}}{|\overrightarrow{p_{\mathrm{r}-1}p_r}| \cdot |\overrightarrow{p_r p_{\mathrm{r}+1}}|}\right) \qquad (18)$$

where p_r and p_{r+1} denote the viewpoints of the r-th and $r+1$-th key frames, respectively. The distance $D_C(p_r,\, p_{r+1})$ between two consecutive key frame viewpoints is computed by the L2-norm. θ is angle between $\overrightarrow{p_{r-1}p_r}$ and $\overrightarrow{p_r p_{r+1}}$. E_C constrains camera movements, so that the position changes of consecutive key frame viewpoints are visually smooth (i.e., not too large).

We normalize each energy term E in $\mathbb{E}_I$ to [0, 1], and the normalized energy term E' (except E'_C) is calculated by

$$E' = \begin{cases} \frac{E - E^{\min}}{E^{\max} - E^{\min}}, & E^{\max} \neq E^{\min} \\ 0, & E^{\max} = E^{\min} \end{cases} \qquad (19)$$

where $E^{\max}$ and $E^{\min}$ denote the maximum and minimum values of E, respectively. E'_C can be calculated by

$$E'_C = \Sigma_{\mathrm{r}=1}^{R-1} D_C\left(p_r, p_{r+1}\right) + \frac{\theta}{\pi}. \qquad (20)$$

Based on the normalized energy terms E'_M, E'_S, E'_P and E'_C, we formulate the camera path computation as a minimization problem defined by

$$\min_{p_1, p_2 \cdots p_R \in \mathbb{P}} \left(\mathbb{E}'_R = c'_1 E'_M + c'_2 E'_S + c'_3 E'_P + c'_4 E'_C\right) \qquad (21)$$

where $\mathbb{P}$ is the set of sampling viewpoints on the viewing sphere and $p_1, p_2\ \ldots\ p_R$ denote the viewpoint of each key frame mesh K_r ($r = 1, 2\ \ldots\ R$), respectively.

Inspired by the solution of Han *et al.* [22], we cast the minimization problem into the problem of finding the shortest path. We organize the sampling viewpoints of key frames into a weighted directed graph G. As Fig. 7 shows, each sampling viewpoint of a key frame is represented by a node in G, and each node connects to its temporal neighbors, that is, a node at key frame r connects to all nodes at key frame $r + 1$. We add a virtual source node s and a virtual sink node t into G. s connects to all nodes at the first key frame, and t is connected by all nodes at the last key frame. Thus, G is a graph with $R{\cdot}S + 2$ nodes and $(R{-}1){\cdot}S^2 + 2R$ edges, where R is the number of key frames and S is the number of nodes (sampling viewpoints) in each key frame. The weight of each edge can be derived from the energy function $\mathbb{E}'_I$ in (19). Specifically, the weight w_r^{ij} of edge (p_r^i, p_{r+1}^j) is calculated by

$$w_r'^{\mathrm{ij}} = \begin{cases} c'_1 \frac{D_M\left(K_{\mathrm{r}+1}, p_{\mathrm{r}+1}^j\right)}{E_M^{\max} - E_M^{\min}} + c'_2 \frac{D_S\left(K_{\mathrm{r}+1}, p_{\mathrm{r}+1}^j\right)}{E_S^{\max} - E_S^{\min}} \\ + c'_3 \frac{D_P\left(K_{\mathrm{r}+1}, p_{\mathrm{r}+1}^j\right)}{E_P^{\max} - E_P^{\min}} + c'_4 \frac{D_C\left(p_r^i, p_{\mathrm{r}+1}^j\right)}{E_C^{\max} - E_C^{\min}}, 0 < r < R \\ c'_1 \frac{D_M\left(K_{\mathrm{r}+1}, p_{\mathrm{r}+1}^j\right)}{E_M^{\max} - E_M^{\min}} + c'_2 \frac{D_S\left(K_{\mathrm{r}+1}, p_{\mathrm{r}+1}^j\right)}{E_S^{\max} - E_S^{\min}} \\ + c'_3 \frac{D_P\left(K_{\mathrm{r}+1}, p_{\mathrm{r}+1}^j\right)}{E_P^{\max} - E_P^{\min}}, \quad r = 0 \ \text{(for source node)} \\ 0, \quad r = R \ \text{(for sink node)} \end{cases} \qquad (22)$$

TABLE II
DETAILS OF THE TEST MESH ANIMATIONS

Mesh Animations	Vertices	Faces	Frames	Key Frames
Handstand	10002	20000	175	12
March	10002	20000	180	13
Bouncing	10002	20000	175	12
Samba	9971	19938	175	12
Swing	9971	19938	150	11

where p_r^i denotes the *i*-th sampling viewpoint of key frame *r* and p_{r+1}^j denotes the *j*-th sampling viewpoint of key frame $r + 1$.

With the help of the sampling viewpoint graph *G*, we solve the minimization problem in (21) by using Dijkstra's algorithm, which finds the shortest path from source node *s* to sink node *t*. The nodes on the shortest path correspond to the sampling viewpoints selected for key frames. Based on these viewpoints, the viewpoints of non-key frames are interpolated through the spherical spline quaternion method proposed by Shoemake *et al.* [41]. Then, the viewpoints of key frames and non-key frames are interleaved to form a series of viewpoints for mesh animation preview. In order to refine the preview, an iterative process will follow it. In the process, the preview will regularly updated according to the users voting feedbacks. After a collection period, the new feedbacks and previous feedbacks will be combined to realize the mesh sequence's preview refinement.

VI. RESULTS AND DISCUSSION

To evaluate the effectiveness of our preview approach, several experiments are conducted on a computer with an Intel Core i7 2.4 GHz CPU, 4 GB of RAM and an NVIDIA GeForce GTX 760 graphics card. We test the proposed approach on five mesh animations, namely, Handstand, March, Bouncing, Samba and Swing [42]. Details of these animations, including the numbers of vertices, faces, frames and key frames, are provided in Table II.

In the experiment, we use our preview approach to generate the initial preview for each test animation with weighting parameters $c_1' = 0.3$, $c_2' = 0.3$, $c_3' = 0.3$ and $c_4' = 0.1$. Then, the initial previews are provided for 50 volunteers to watch and each volunteer watches each of the test animations 5 times in a random order. When watching a mesh preview, each user can freely make positive votes for its preferred views and negative votes for his/her disliked views by performing simple mouse clicks. These voting feedbacks help to refine the viewpoint series of a mesh animation preview. Because that the user voting based preview refinement is an iterative process, the preview of a mesh sequence will be updated periodically. The updated preview is then provided to volunteers for watching, and new voting feedbacks for "good views" or "bad views" are collected. During the refinement process, the increase of c_3' weight will optimize the preview path toward user's preference quickly, while it also brings trouble to camera smoothness. The reason of such change is that user evaluation reflects in part the personal preference of viewpoints in the voted frames and ignores the stability and movement speed of the complete preview path. If the positions of voted viewpoints in neighboring frames change distinctively, the generated path will contain the unstable segments. By the way, some good sampling points with abundant geometric information and motion information will not be preferred with the decrease of weights c_1' and weights c_2', if increasing the weight of c_3'. Therefore, the weighting parameters are set as $c_1' = 0.3$, $c_2' = 0.3$, $c_3' = 0.3$ and $c_4' = 0.1$ since the inter-frame motion saliency, intra-frame surface saliency and user preference are considered equally important to the viewpoint quality. With the increase of the times of watching, the influence of voting will be accumulated and the preview path will be constantly optimized. Fig. 8 shows some typical frames of the refined previews of the test animations.

Fig. 8 illustrates that the proposed approach can successfully capture the salient motions and shape features of a mesh animation. As a result, our mesh preview method helps users acquire a good understanding of animation contents. We also visualize the preview camera path of each test animation, and Fig. 9 takes the Handstand and Swing animations as an example to show the visualization results. In Fig. 9, a blue curve denotes a preview camera path, a red dot denotes the viewpoint of a key frame and the corresponding number denotes the frame number of the key frame. As Fig. 9 shows, the preview camera paths are visually smooth, and the camera position does not have sudden changes in a short time. In order to verify the robustness of the proposed method, we compare the preview paths between an original sequence and some slightly modified sequences. Taking Handstand sequence as an example, the first sequence is the original sequence, the second sequence changes frames from 31th to 44th and frames from 121th to 134th, and the third sequence directly deletes frames from 166th to the last frame. To exclude the effect of user voting, we conduct experiment on the three sequences without voting. The results are rendered in Fig. 10. The three paths are similar in the most parts except the segments of changed frames with different mesh models. The comparison illustrates that our method is stable and robust, and local variation doesn't affect the whole path.

We test the time cost of our preview approach, which depends on the number of key frames. Table III lists the time cost results for each mesh animation. The total time cost of our approach ranges from 192.1 to 256.7 sec, which consists of the time costs of inter-frame motion saliency computation, intra-frame surface saliency computation, user preference computation and camera path computation. The most time consuming part is inter-frame motion saliency computation, which applies the matching and segmentation phases to decompose key frame meshes into several rigidly moving components. Both the two phases are computationally complex, and thus, more computational time is required. The camera path computation is another time consuming part, using Dijkstra's algorithm to find the shortest path in the sampling viewpoint graph. Due to the complexity of Dijkstra's algorithm, the camera path computation takes more time than inter-frame motion saliency computation and user preference computation.

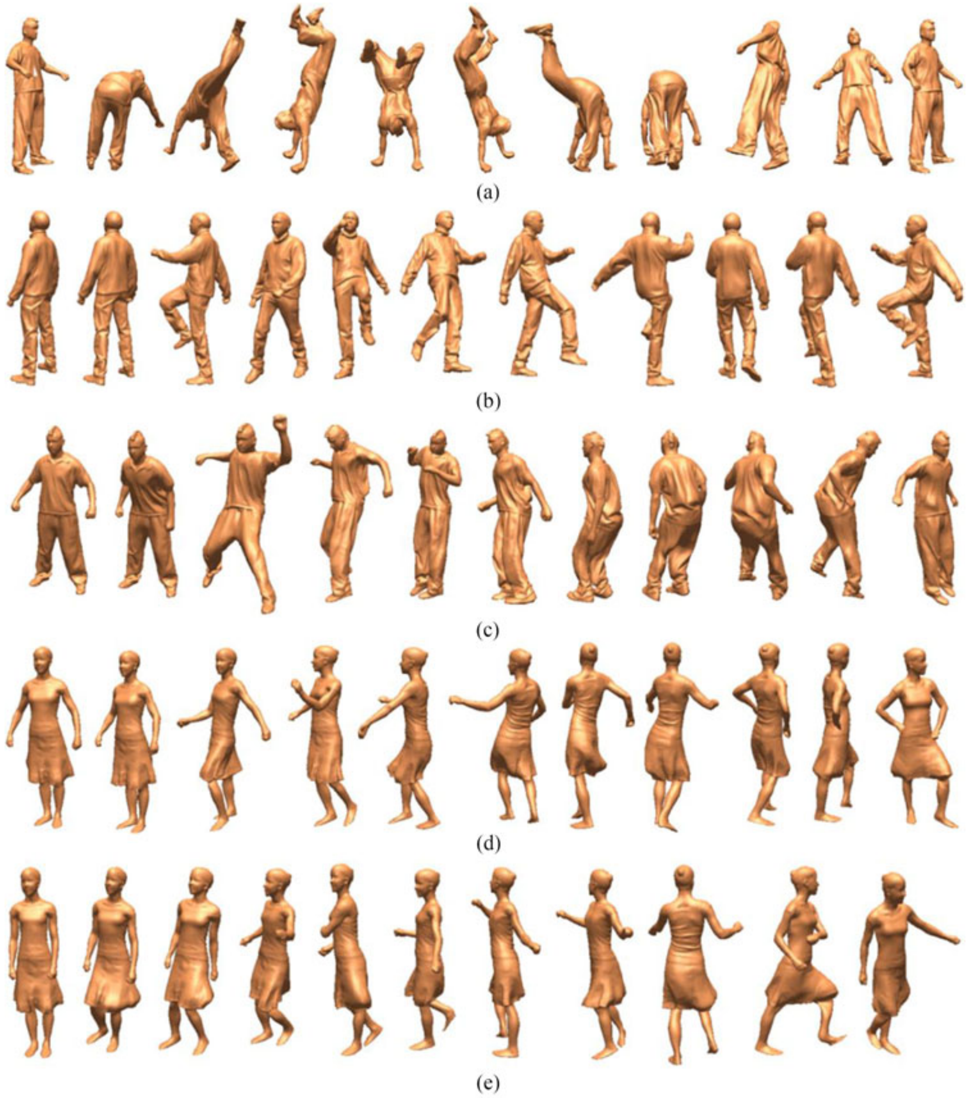


Fig. 8. Typical frames of the refined previews of the test animations: (a) Handstand, (b) March, (c) Bouncing, (d) Samba, and (e) Swing.

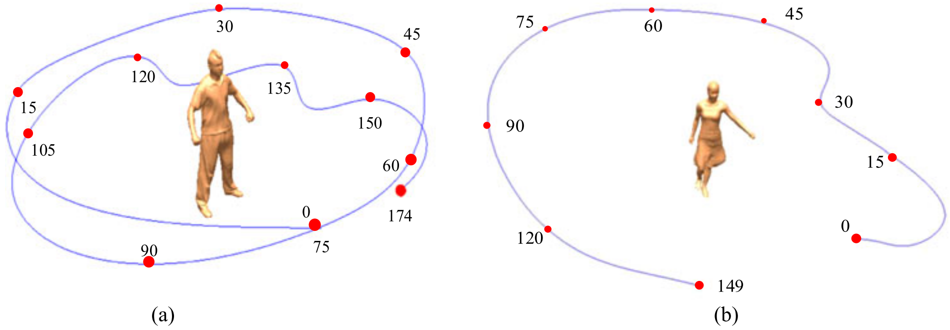


Fig. 9. Visualization results of the preview camera path: (a) Handstand and (b) Swing.

Considering that neighbors of voted frame will change with optimization, the updated sequences are different from the initial previews. The static picture cannot show these differences. We provide a demo to display comparison of previews before and after optimization. And we also conduct a user study to evaluate the subjective visual quality of our preview results with/without voting. Meanwhile, our preview approach is compared with the method proposed by Han *et al.* [22] and the

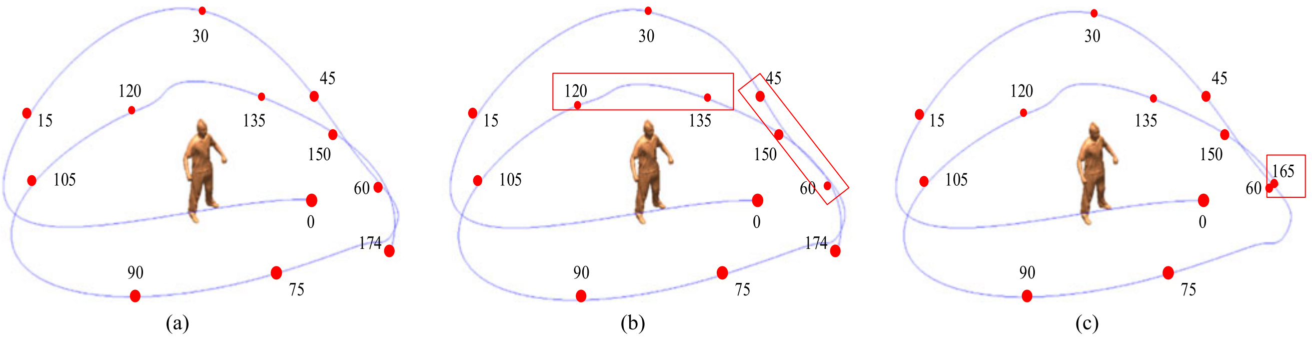


Fig. 10. Comparison results of the preview camera path without voting: (a) original Handstand, (b) second sequence of Handstand, and (c) third sequence of Handstand.

TABLE III
TIME COST OF OUR METHOD FOR EACH ANIMATION (S)

Mesh Animations	Inter-frame motion saliency computation	Intra-frame surface saliency computation	User preference computation	Camera path computation	Total
Handstand	75.3	45.8	29.5	69.2	219.8
March	87.4	54.2	36.5	78.6	256.7
Bouncing	77.5	42.2	23.7	70.8	214.2
Samba	73.2	46.3	30.1	71.4	221
Swing	61.8	35.4	29.6	65.3	192.1

TABLE IV
PAIRWISE VOTING RESULTS FOR EACH ASPECT

Methods	Informative	Expressive	Stable	Total (4500)
Our method with voting	569	518	542	1629
Our method without voting	420	445	469	1334
Han *et al.*'s method	347	291	273	911
Mesh saliency based method	174	238	214	626

TABLE V
PREFERENCES FOR THE THREE PREVIEW METHODS

Methods	Method 1	Method 2	Method 3	Method 4	Total (4500)
1.Our method with voting	-	517	549	563	1629
2.Our method without voting	233	-	547	554	1334
3.Han *et al.*'s method	201	203	-	507	911
4.Mesh saliency based method	187	196	243	-	626

mesh saliency based method. The mesh saliency based method selects preview view-points for key frames according to the mesh saliency proposed by Lee *et al.* [8]. We test five animations, and thus, there are a total of 30 previews. Then, we perform our user study with the paired comparison method proposed by David [43], which is widely used for the statistical evaluation of subjective preferences. Following this method, previews are displayed side-by-side in pairs to a user, who then selects a preferred one in each pair. In our experiment, for each pair of previews, a user is required to vote which one is more informative, expressive and stable, respectively (i.e., 3 votes for each pair). We invite another 50 volunteers with no prior knowledge of the test animations. Each volunteer watches $5 \times 6 = 30$ pairs of previews; thus, there are a total of 1500 pairwise comparisons. To reduce the bias in our experiment, the playing order of preview pairs are randomized for each volunteer, and we do not provide any information about the test preview methods. Table IV lists the pairwise voting results in each aspect. From the pairwise voting results of the preview without voting, it has weakness in informative aspect. The optimized results through a series of voting get the approval from the volunteers. And expressive aspect is also gradually improved. Thus, the algorithm of automatic mesh animation preview with user voting proposed is effective. Compared with the other two methods, our method is more informative and stable, which means that our method can better describe the animation contents and keep the camera motion visually smooth. Table IV also shows that the four methods are close in expressiveness and able to capture the salient shape features of a mesh model.

Table V summarizes the votes received by each preview method and shows the overall preference across the 50 volunteers. As Table V shows, the total number of votes is 4,500, in which each pair of preview methods is compared 250 times and receives $250 \times 3 = 750$ votes, i.e., 750 votes for each aspect. Our method with voting is accepted by 68.9 percent (517 of 750) of volunteers, achieving higher preference than the one without voting. Meanwhile, our approach is clearly preferred over the mesh saliency based method since our preview results win 75.07 percent (563 of 750) of volunteer votes. Compared to Han *et al.*'s method, our preview results are preferred in

73.2 percent (549 of 750), whereas the results of the mesh saliency based method are only favored 32.4 in percent (243 of 750). In general, the volunteers prefer our method in 36.2 percent (1629 of 4,500) of the pairwise comparisons, whereas the preferences for Han *et al.*'s method and the mesh saliency based method are 20.2 percent (911 of 4,500) and 13.9 percent (626 of 4,500), respectively.

To verify the consistency of volunteer votes, we calculate Kendall's coefficient [44] of concordance W, which is defined by

$$W = \frac{12\,\Sigma^{n}_{\mathrm{i}=1}\left(R_i - \bar{R}\right)^2}{m^2\left(n^3 - n\right)} \tag{23}$$

where m denotes the number of pairwise comparisons, n denotes the number of methods, R_i denotes the number of votes won by the *i-th* method and $\bar{R}$ is the average value of $R_i (i = 1, 2 \ldots n)$. Based on the research of Kendall and Smith [41], *m·(n − 1)*·W follows a $\chi2$ distribution with *n − 1* degrees of freedom. The value of W is 0.095 and the value of *m·(n − 1)*·W is 427.5. According to the theory of hypothesis testing, a p-value = 0.01 corresponds to $\chi^2_{0.01}(3) = 11.34$, which is considerably smaller than 427.5; therefore, there is a statistically significant agreement among the volunteers regarding the three preview methods. The volunteers agree that optimization based on voting is effective and our approach outperforms Han *et al.*'s method and the mesh saliency based method in terms of the subjective visual quality of preview results.

Except the statistical analysis of voting data above, we also analyze the cases where our method is not regarded as the preferred in user study. The evaluation of visual quality of the preview paths is a subjective problem. Different users have their personalized criterions to evaluate quality of preview path, and any preview path is rare to meet each user's preferred. Taking informative indicator as an example, some people think that the preview paths with abundant local geometric information are more informative, some others are willing to vote for the camera paths which show amounts of global semantic information. Correspondingly, volunteers in favor of the front criterion may vote for the Han *et al.*'s method or mesh saliency based method in our user study, and the proposed method will not be preferred. The cases are similar when it comes to the evaluation of expressive indicator and stable indicator. In view of above analysis, we can see that the viewpoints not conforming to the user preferences are unpopularity. Hence, the voting mechanism is introduced into the proposed method to improve such cases with help of the accumulative influence of user votes.

VII. Conclusion

Unlike a static 3D model, a mesh animation is a sequence of mesh models arranged in time order, which have geometry shape features as well as inter-frame motion characteristics. Therefore, the viewpoint selection of mesh animations is more complex than that of static models. To date, most existing viewpoint selection methods still focus on static 3D models and are not suitable for mesh animations.

In this paper, we propose an automatic mesh animation preview method with user voting refinement. The proposed method selects preview viewpoints of a mesh animation by using the inter-frame motion saliency, intra-frame surface saliency and user preference as viewpoint quality metrics. We introduce a voting mechanism that allows users to vote for good and bad views when watching a preview. After a period of user voting feedback collection, the proposed method periodically uses user voting data to update the mesh preview.

To evaluate the effectiveness of our method, we conduct a user study participated by 50 volunteers to compare our preview results with those generated by Han *et al.*'s method and the mesh saliency based method. The experiment results demonstrate that our method can make a better description of animation contents. And the method with voting can further improve the preview results. In addition, it is also shown that the volunteers prefer our method with/without voting in 36.2/29.6 percent of the pairwise comparisons, whereas the preferences for Han *et al.*'s method and the mesh saliency based method are 20.2 and 13.9 percent, respectively.

References

[1] H. Jiang, G. Zhang, H Wang, and H. Bao, "Spatio-temporal video segmentation of static scenes and its applications," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 3–15, Jan. 2015.

[2] J. K. Ahn, Y. J. Koh, and C. S. Kim, "Efficient fine-granular scalable coding of 3D mesh sequences," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 485–497, Apr. 2013.

[3] J. Fu *et al.*, "Kinect-like depth data compression," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1340–1352, Oct. 2013.

[4] T. Vieira *et al.*, "Learning good views through intelligent galleries," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 717–726, 2009.

[5] A. Secord *et al.*, "Perceptual models of viewpoint preference," *ACM Trans. Graph.*, vol. 30, no. 5, 2011, Art. no. 109.

[6] T. Kamada and S. Kawai, "A simple method for computing general position in displaying three-dimensional objects," *Comput. Vis., Graph. Image Process.*, vol. 41, no. 1, pp. 43–56, 1988.

[7] S. Fleishman, D. Cohen-Or, and D. Lischinski, "Automatic camera placement for image-based modeling," *Comput. Graph. Forum*, vol. 19, no. 2, pp. 101–110, 2000.

[8] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," *ACM Trans. Graph.*, vol. 24, no. 3. pp. 659–666, 2005.

[9] P. P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, "Automatic view selection using viewpoint entropy and its application to image-based modelling," *Comput. Graph. Forum*, vol. 22, no. 4. pp. 689–700, 2003.

[10] W. Saleem, W. Song, A. Belyaev, and H. P. Seidel, "On computing best fly," in *Proc. ACM Spring Conf. Comput. Graph.*, 2007, pp. 115–121.

[11] S. Zhao, W. T. Ooi, A. Carlier, G. Morin, and V. Charvillat, "3D mesh preview streaming," in *Proc. 4th ACM Multimedia Syst. Conf.*, 2013, pp. 178–189.

[12] S. Zhao, W. T. Ooi, A. Carlier, G. Morin, and V. Charvillat, "Bandwidth adaptation for 3D mesh preview streaming," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 10, no. 1s, 2014, Art. no. 13.

[13] G. Al-Regib and Y. Altunbasak, "3TP: An application-layer protocol for streaming 3D models," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1149–1156, Dec. 2005.

[14] S. B. Park, C. S. Kim, and S. U. Lee, "Error resilient 3-D mesh compression," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 885–895, Oct. 2006.

[15] S. L. Stoev and W. Strasser, "A case study on automatic camera placement and motion for visualizing historical data," in *Proc. IEEE Vis.*, Oct.-Nov. 2002, pp. 545–548.

[16] M. Feixas, M. Sbert, and F. González, "A unified information-theoretic framework for viewpoint selection and mesh saliency," *ACM Trans. Appl. Perception*, vol. 6, no. 1. pp. 1–23, 2009.

[17] J. Tao, J. Ma, C. Wang, and C. K. Shene, "A unified approach to streamline selection and viewpoint selection for 3D flow visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 3, pp. 393–406, Mar. 2013.

[18] M. Hisada, A. G. Belyaev, and T. L. Kunii, "A skeleton-based approach for detection of perceptually salient features on polygonal surfaces," *Comput. Graph. Forum*, vol. 21, no. 4, pp. 689–700, 2002.

[19] P. Shilane and T. Funkhouser, "Distinctive regions of 3D surfaces," *ACM Trans. Graph.*, vol. 26, no. 2, pp. 1–15, 2007.

[20] P. Shilane and T. Funkhouser, "Selecting distinctive 3D shape descriptors for similarity retrieval," in *Proc. IEEE Int. Conf. Shape Model. Appl.*, Jun. 2006, pp. 18–27.

[21] J. Y. Kwon and I. K. Lee, "Determination of camera parameters for character motions using motion area," *Vis. Comput.*, vol. 24, nos. 7–9, pp. 475–483, 2008.

[22] S. R. Han, T. Yamasaki, and K. Aizawa, "Automatic preview video generation for mesh sequences," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 2945–2948.

[23] J. Bernard *et al.*, "MotionExplorer: exploratory search in human motion capture data based on hierarchical aggregation," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2257–66, Dec. 2013.

[24] A. Vögele, M. Hermann, B. Krüger, and R. Klein, "Interactive steering of mesh animations," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2012, pp. 53–58.

[25] H. Rhodin *et al.*, "Generalizing wave gestures from sparse examples for real-time character control," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–12, 2015.

[26] J. Assa, D. Cohen-Or, I. C. Yeh, and T. Y. Lee, "Motion overview of human actions," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 115, 2008.

[27] J. Assa, L. Wolf, and D. Cohen-Or, "The virtual director: A correlation-based online viewing of human motion," *Comput. Graph. Forum*, vol. 29, no. 2, pp. 595–604, 2010.

[28] I. C. Yeh *et al.*, "Social-event-driven camera control for multicharacter animations," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 9, pp. 1496–1510, Sep. 2012.

[29] M. Christie, R. Machap, J. M. Normand, P. Oliver, and J. Pickering, "Virtual camera planning: A survey," *Smart Graph*., vol. 3638, pp. 40–52, 2005.

[30] M. Christie, P. Olivier, and J. M. Normand, "Camera control in computer graphics," *Comput. Graph. Forum*, vol. 27, no. 8, pp. 2197–2218, 2008.

[31] R. Arcila, K. Buddha, F. Hétroy, F. Denis, and F. Dupont, "A framework for motion-based mesh sequence segmentation," in *Proc. Int. Conf. Comput. Graph., Vis. Comput. Vis.*, 2010, pp. 33–40.

[32] R. Arcila, C. Cagniart, F. Hétroy, E. Boyer, and F. Dupont, "Segmentation of temporal mesh sequences into rigidly moving components," *Graph. Models*, vol. 75, no. 1, pp. 10–22, 2013.

[33] C. Cagniart, E. Boyer, and S. Ilic, "Iterative deformable surface tracking in multi-view setups," in *Proc. 5th Int. Symp. 3D Data Process., Vis. Transmiss.*, 2010, pp. 1–8.

[34] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A*, vol. 4, no. 4, pp. 629–642, 1987.

[35] D. D. Hoffman, and M. Singh, "Salience of visual parts," *Cognition*, vol. 63, no. 1, pp. 29–78, 1997.

[36] L. Dong, Y. Fang, W Lin, and H. S. Seah, "Perceptual quality assessment for 3D triangle mesh based on curvature," *IEEE Trans Multimedia*, vol. 17, no. 12, pp. 2174–2184, Dec. 2015.

[37] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and Mathematics III*. New York, NY, USA: Springer-Verlag, 2003, pp. 35–57.

[38] T. Corrigan and P. White, *The Film Experience*. New York, NY, USA: Bedford/St. Martin's, 2004, p. 140.

[39] Y. Taniguchi, A. Akutsu, Y. Tonomura, and H. Hamada, "An intuitive and efficient access interface to real-time incoming video based on automatic indexing," in *Proc. 3rd ACM Int. Conf. Multimedia*, 1995, pp. 25–33.

[40] C. Halit and T. Capin, "Multiscale motion saliency for keyframe extraction from motion capture sequences," *Comput. Animation Virtual Worlds*, vol. 22, no. 1, pp. 3–14, 2011.

[41] K. Shoemake, "Animating rotation with quaternion curves," *ACM SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 245–254, 1985.

[42] D. Vlasic, I. Baran, W. Matusik, and J. Popović. (2008). [Online]. Available: http://people.csail.mit.edu/drdaniel/mesh_animation/index.html#data

[43] H. A. David, *The Method of Paired Comparisons*. London, U.K.: Charles Griffin Company*,* 1963.

[44] M. G. Kendall and B. B. Smith, "The problem of m rankings," *Ann. Math. Statist.*, vol. 10, no. 3, pp. 275–287, 1939.

Zhong Zhou (M'14) received the B.S. degree from Nanjing University, Nanjing, China, in 1999, and the Ph.D. degree from Beihang University, Beijing, China, in 2005.

He is currently a Professor and a Ph.D. Adviser with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing. His main research interests include augmented virtual environment, natural phenomena simulation, distributed virtual environment, and Internet-based VR technologies.

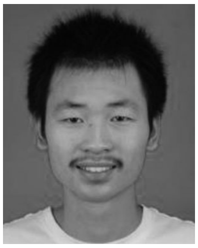
Dr. Zhou is a Member of ACM and CCF.

Na Jiang received the M.S. degree in computer science and technology from Jilin Agricultural University, Jilin, China, in 2014, and is currently working toward the Ph.D. degree in computer science at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China.

Her research interests include remote rendering, three-dimensional visualization, and motion segmentation.

Ke Chen received the M.S. degree in computer science and technology from Beihang University, Beijing, China, in 2010, and the Ph.D. degree in computer science from the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, in 2015.

His research interests include remote rendering, three-dimensional visualization, and video coding.



Jingchang Zhang received the M.S. degree in computer science and technology from Beihang University, Beijing, China, in 2016.

His research interests include remote rendering and three-dimensional visualization.