AUTOMATIC MESH ANIMATION PREVIEW

*Yi Li, Qiaodong Cui, Fei Dou, Lin Zhang, Zhong Zhou**

State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
Beijing, China
*e-mail: zz@vrlab.buaa.edu.cn

ABSTRACT

With growing number of high quality 3D models published online, the technique of generating efficient and economical 3D model previews has raised increasing concerns. Although several previous work has been done on the preview of static mesh, that of animated mesh is different and more complex due to the difficulty in describing the animation. In this paper, we present a novel method of automatic preview generation for 3D mesh animation. A new measure named inter-frame surface saliency, which evaluates both inter-frame motions and surface saliency in each frame, is introduced. Given an animated mesh, an energy function combining this measure and camera smoothness is constructed for the representative viewpoints selection in key frames, and then an optimal camera path is generated. Finally, a brief but informative preview could be created by moving the camera along this path with frame rate control.

Index Terms— mesh animation preview, inter-frame surface saliency, camera path generation

1. INTRODUCTION

With the development of 3D modeling and animation techniques, as well as the computer hardware, the amount of high quality 3D static/animated models is increasing at a relatively high rate. Today there have been more and more online data publishers for 3D models, such as TurboSquid, The3DStudio and Berkeley MHAD, which allow users to download models or animation datas for academic or commercial purposes. An efficient and economical 3D static/animated models preview technique is usually needed to give the user a general understanding before download, due to the growing data size of 3D models. Many works have been done on preview of 3D static model, however that of animated model is different and more complex due to the difficulty of creating a brief but informative description of the animation.

This work is supported by the National 863 Program of China under Grant No.2012AA011801, the Natural Science Foundation of China under Grant No.61170188, and the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No.20121102130004.

Similar to the preview of static model, there are mainly two kinds of methods to preview models, image-based and video-based. The idea of image-based method is using a set of images taken from different representative viewpoints around a 3D model to help users understand it [1, 2, 3, 4]. The video-based method is to preview the model through a video generated from a selected camera path [5, 6, 7, 8, 9]. Comparing to the image-based method, the video-based one gives a better view of the continuous dynamic changes of the animated models, resulting a better understanding of the animation.

To the best of our knowledge, most of the image-based and video-based methods work on static models, few works have been done on mesh animation preview before. Han et al. [5] proposed a method to generate automatic preview video of mesh sequences by adopting Dijkstra's algorithm which considering both of information quantity and camera travel distance. However, Han et al.'s method considered only the information quality from the static models' point of view without taking the significance of motion into account. Moreover, this method is actually an image-based strategy that only selects a fixed viewpoint for each frame and therefore cannot generate a real smooth preview video.

In this paper, we present a novel method of automatic preview video generation for 3D mesh animations. A new measure, namely inter-frame surface saliency, which contains both of motion information between frames and surface saliency in each frame is introduced. To generate a brief but informative description of the animation sequences, some key frames of the animated mesh are firstly selected. An energy function of both inter-frame surface saliency and camera smoothness is constructed for computing the representative viewpoints, and an optimal camera path is then generated. Finally, a brief preview video is created by moving the camera along this path with frame rate control.

The rest of this paper is organized as follows. In Section 2, we present the preprocessing steps of viewpoint selection called motion analysis and the definition of our new measure. The algorithm of optimal camera path generation is presented in Section 3. The evaluation of experiment results on several models are presented in Section 4 and followed by a conclusion and expectation in Section 5.

Fig. 1. Segmentation using the transformation matrices. First row: on successive frames before key frame extraction. Second row: on extracted successive key frames.

2. VIEWPOINT SELECTION METRIC

Previous researches about preview are mainly aimed at static meshes. Only a few method could directly work on mesh animations [5] but still does not take into account the motion information generating between frames. In this paper we propose a new measure called inter-frame surface saliency to evaluate the viewpoints taking into account both motion information and static surface information.

2.1. Motion analysis

In order to generate the preview of a mesh animation, analyzing and evaluating its motions is the foremost procedure. For this purpose, we employ a method inspired by Arcila et al.'s research [10, 11] to segment the mesh in each frame of an animation into several parts according to the motions. The inter-frame surface saliency computation are implemented based on the segmentation.

According to Arcila et al.'s research [10], the variable segmentation can display the different motion information at each time step, and the variable segmentation is defined below: Given a mesh sequence $MS = \{M^t, t \in [1, T]\}$ that consists of T frames, a variable segmentation is Σ_v , which is a set of segmentations $\Sigma^t = \{M_1^t, ..., M_{k_t}^t\}$ of each mesh M^t of the sequence MS such that the number k of sub-meshes is possibly different for all segmentations.

There are three steps to obtain the variable segmentation. First, the motion of each vertex v_j^t of M^t is estimated by using Horn's method [12]. This method can estimate the best rigid transformations of the segments between two frames. Several 4×4 matrices which describe the transformations are also obtained. Then neighboring segments that present similar motions are merged. After that a spectral clustering approach is used to refine the segmentation to the final segmentation Σ^t.

After the transformation matrices are obtained, we can use them to divide the mesh sequence into many shots and the key frame of each shot is selected as the most representative one(see Section 3.1). After the key frame sequence is generated, the transformation matrices between two key frames are recalculated. Our estimation of motion importance is mainly based on these matrices because they are more robust than those we get between two successive frames(see Fig. 1).

In order to evaluate the weight of a motion for a certain motion segment, we can compute the angle φ rotated and displacement v from the matrix of the segment obtained above. Then it can be represented by the weight function w, which is defined as:

$$w = A\varphi + Bv \tag{1}$$

where A and B are two tunable positive factors to weight between rotation and translation. Fig. 2 provides visualizations of segment weight for one of models used in this paper.

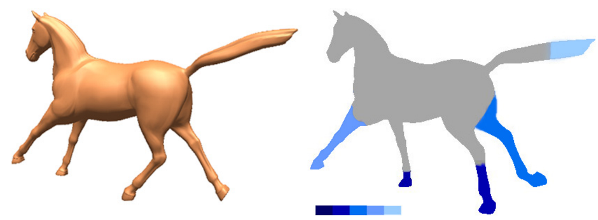


Fig. 2. Two successive frames of a mesh animation. The segment weights are mapped on the latter(right) frame. Deeper color implies heavier weight.

2.2. Inter-frame surface saliency

Given a mesh animation MS to preview, we propose to find a series of corresponding viewpoints $P = \{p^t, t \in [1, T]\}$ at each of which the camera can acquire more information than any others per frame. To find the best viewpoint set P, the crucial step is to structure the measure of viewpoint quality. Unlike surface saliency of static meshes, in this paper we take into account both of inter-frame motions and surface saliency in each frame to define a new measure for preview of mesh animations called inter-frame surface saliency.

As mentioned in Section 2.1, each segment can be weighted according to its motion between two consecutive frames in the course of segmentation. We define the inter-frame motion information as the total weighted visible area of the motion segments from a certain viewpoint. Thus we can measure the inter-frame motions using the function as:

$$E_1 = \sum_{t=1}^{T} D(M^t, p^t) \tag{2}$$

$$D(M,p) = \int_{f \in \Phi \subset F} w(f) dA - \int_{f \in \Phi_p \subset \Omega_p \subset F} w(f) dA \tag{3}$$

where $w(f)$ denotes the weight of face f in accordance with the motion weight w of the segment it belongs to. Other notation is defined as follows: Ω_p is the set of visible faces from p, Φ_p the set of visible faces belong to motion part and F the set of faces of the whole mesh M. It's obvious that $D(M,p)$ actually represents the weighted invisible area of the motion segment. To gain more motion information along a mesh animation, we should find a set of viewpoints to make Function (2) reach its minimum.

As motion is just one of the properties of mesh animation, static surface saliency is also important for preview. The majority methods for surface saliency of 3D meshes based only on geometry but not texture or lighting. Among the many view descriptors(e.g., viewpoint entropy [8], visibility ratio [13], mesh saliency [14]) assumed to measure the geometric complexity of the visible surface as viewed from a point, we employ mean curvature and visual area for the computation of static surface information since they are the primary geometric properties of a mesh model.

For each mesh model M^t of a mesh animation MS, we demand to find the viewpoint p^t at which the sum of mean curvature of the visible vertices and the total visual area both reach the maximum as we simply consider that the larger value gives the better understanding. Then the energy function about static surface information is defined as follows:

$$E_2 = \sum_{t=1}^{T} (C(M^t, p^t) + S(M^t, p^t)) \tag{4}$$

For the mesh M in each frame, $C(M,p)$ is the measure of static surface information about mean curvature and $S(M,p)$ is about visual area:

$$C(M,p) = \int_{v \in V} c(v) dv - \int_{v \in \Psi_p \subset V} c(v) dv, \tag{5}$$

$$S(M,p) = \int_{f \in F} dA - \int_{f \in \Omega_p \subset F} dA \tag{6}$$

where $c(v)$ is the mean curvature of vertex v, Ψ_p the set of visible vertices from p, V the set of vertices of the whole mesh M. Therefore $C(M,p)$ and $S(M,p)$ respectively denotes the sum of mean curvature of the invisible vertices and the total area of the invisible faces of mesh M viewed at point p in fact. We want to minimize Function (4) to acquire the maximal static surface information.

3. CAMERA PATH GENERATION

With the measure of viewpoint above, an optimal path can be found along which the camera can generate a preview meeting the requirements mentioned below:

- ***Conciseness***: the preview should be brief in form but comprehensive in scope;
- ***Abundant***: the preview must convey maximal information to viewers as far as possible;
- ***Stability***: the changes of scene must be visually acceptable.

3.1. Viewpoint sampling and Key frame extraction

The number of points in space is infinite. On account of that, we want to sample a representative finite set $\mathcal{P}$ in the space where an animated mesh lies to be candidate viewpoints. Therefore we adopt subdivision surfaces algorithm to make the bounding sphere of the animated mesh into several polygons and let the vertices of them be the candidate viewpoints.

As we utilize subdivision surfaces algorithm to sample viewpoints, there is a certain distance between each pair of them. If we choose a fixed viewpoint for each frame, the camera may stop (several successive frames have the same optimal viewpoint) or swing (each frame has a quite different optimal viewpoint) in all probability. Thus key frame extraction has been employed to reduce the frequency of view changes and ensure no missing important information at the same time. For mesh animations, key frame implies the more abundant motion information than others'. We utilize a motion curve to depict how the sum of the product of each segment's area and weight varies over time, then the mesh animation is divided into several shots using k-means clustering where $k = \frac{T}{15}$. The first frame is considered as the key frame of the first cluster. For each cluster except the first one, the frame with highest value on the motion curve stands out as the key frame(see Fig. 3).

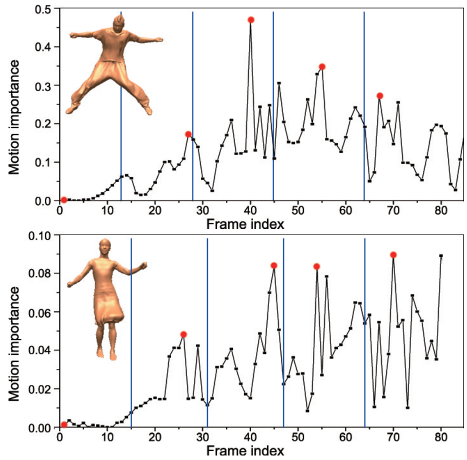


Fig. 3. In two different animations, motion importance varies over time with key frames marked by red dots. Motion importance is calculated as the sum of the product of each segment's area and weight.

In addition, after the key frame sequence $MK = \{M^{k_i}, i \in [1, \widetilde{T}]\}$, k_i denotes the number of i^{th} key frame, is generated from an original mesh animation, the motion matrices between two key frames are recalculated to obtain a more robust estimation of motion weight and the computation time for inter-frame surface saliency will decrease as the number of viewpoints be reduced. As a result our goal can be considered as finding a path covering a set of viewpoints $\widetilde{P} = \{p^{k_i}, i \in [1, \widetilde{T}]\}$. The first requirement of conciseness is satisfied.

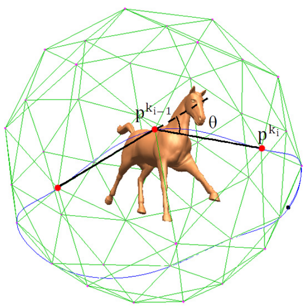


Fig. 4. Camera smoothness constraint. The camera moves from p^{k_i} to $p^{k_{i-1}}$.

3.2. Camera smoothness constraint

Given a candidate viewpoint set $\mathcal{P}$, except for measuring their quality by inter-frame surface saliency, there is another constraint to choose a representative viewpoint for each key frame from them: the camera movements along these representative viewpoint must be smooth so that the changes of scene is visually acceptable. We consider this constraint as the Euclidean distance and rotation angle (see Fig. 4) of the camera moving between two successive viewpoints:

$$\widetilde{E_3} = \sum_{i=1}^{\widetilde{T}} P(p^{k_i}, p^{k_{i-1}}) \tag{7}$$

$$P(p^{k_i}, p^{k_{i-1}}) = \frac{\|p^{k_i} - p^{k_{i-1}}\|_2}{R} + \frac{\theta}{\pi} \tag{8}$$

The distance is used to avoid excessive transition of the camera and the rotation angle θ to avoid swing. Dividing the distance by the diameters R of the viewpoint sphere is to normalize it and so does the rotation angle. The current view location has significant impact on the choice of the next one. We use Function (7) to meet the last requirement because the smaller the value, the more stable the preview.

3.3. Camera path computation

Since inter-frame surface saliency and camera smoothness constraint are discussed, the final objective is to find an optimal camera path by minimizing the total energy function:

$$E_{total} = \alpha\widetilde{E_1} + \beta\widetilde{E_2} + \gamma\widetilde{E_3} \tag{9}$$

We use the same strategy as [5] to solve this problem as building a weighted graph G whose nodes donate the candidate viewpoints and edges the quality of them and then find the shortest path using Dijkstra's algorithm. In Function (9), α, β, γ are three tunable positive impact factors.

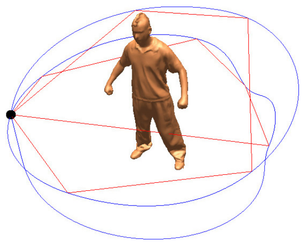


Fig. 5. Camera path of a mesh animation. Red lines: viewpoints are connected with straight lines. Blue lines: viewpoints are connected using spherical splines interpolation.

On account of that we only sample viewpoints in key frames, those in intermediate frames are interpolated by Shoemake's method [15]. This method is for spherical interpolation using quaternion, cubic spline and spherical Bezier curves. Camera move along spherical splines on the bounding sphere of a mesh sequence performs much better than connecting the viewpoints in key frames with straight lines in terms of stability(see Fig. 5).

While camera smoothness constraint and viewpoint interpolation satisfy the stability requirement together, inter-frame surface saliency represented as $\widetilde{E_1}$ and $\widetilde{E_2}$ meet the abundant requirement. With this, it implies that a optimal camera path has been generated.

3.4. Frame rate control

Camera speed during two key frames, is determined by the distance between the viewpoints of the frames and the frame rate of the mesh animation:

$$\mathcal{V}(i) = \frac{p^{k_i} - p^{k_{i-1}}}{k_i - k_{i-1}} \cdot FPS \tag{10}$$

The camera should slow down when passing over visually important shots of the animation, and speed up for uninteresting views. Such a similar work has already been attempted in preview for static meshes [6], which is derived from Two-Thirds Power Law [16]. In our case, we assume the frame rate of the animation depending on the distance between two key frames, which is:

$$f(i) = FPS(\frac{1}{(k_i - k_{i-1}) + r})^{\frac{1}{3}} \tag{11}$$

where r is a constant offset to compensate for the 0 to 1 normalization of $k_i - k_{i-1}$. Putting $r = 1$ makes f between FPS and $\frac{FPS}{\sqrt[3]{2}} \approx 0.79FPS$. The value of FPS is appropriate when the changes in frame rate is not only steady but also discernible .

4. EVALUATION

In this section, we introduce the experiments results to evaluate our method using several mesh animations. We run the program on a Windows 7 PC with Intel Core2 Duo 3.00GHz CPU and 3GB RAM. For each animation, 42 viewpoints were sampled on the sphere per frame and then a graph G composed of $T \times 42$ nodes and $T \times 42 \times 42$ edges is built, where T denotes the length of the animation. The first and the last viewpoints is selected as the one with largest inter-frame surface saliency. The data including the number of vertices and the faces of the experimental models and the frame number of their animations is shown in Table 1.

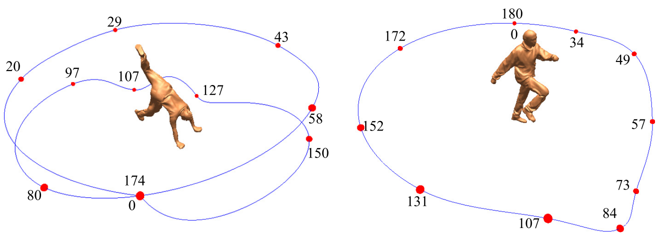


Fig. 6. Camera path of two different mesh animations. Left: "Handstand"; Right: "March".

Because of the subjectivity of visual evaluation, it's hard to give fixed values to the parameters in Function (9) for every model. Thus we just provide default values($\alpha = 0.4, \beta = 0.4, \gamma = 0.2$) which were determined according to 20 users' feedback after they used our system to make preview videos

Table 1. Data of several mesh animations for experiments

Index	Models	Frames	Vertices	Faces
1	Bouncing	175	10002	20000
2	Handstand	175	10002	20000
3	March	181	10002	20000
4	Samba	174	9971	19938
5	Horse	49	8431	16843

Table 2. Experiment results

Index	Key frames	Computation time	Preview time
1	11	220s	10s
2	11	204s	9s
3	12	272s	11s
4	11	225s	8s
5	3	73s	4s

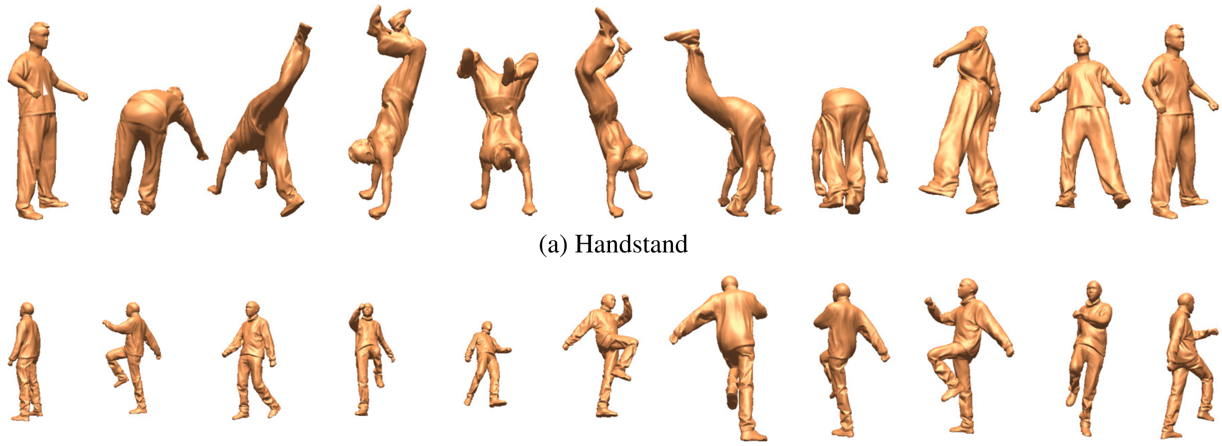
of 5 models. The experimental results in the form of data which were obtained by fine-tuning(± 0.2) the default values specifically for different models are shown in Table 2. The user satisfaction survey revealed that more than 90% users consider that our system is practical and easy-to-use.

We display the optimal camera path of two animations called "Handstand" and "March" in Fig. 6, which also locates the viewpoints in the key frames. In order to present our previews, in Fig. 7, we use the pictures extracted at the key frames from the previews of these two animations according to the path shown in Fig. 6. Actually we have tested all the meshes that appear in this paper, but do not display all of them due to the space limit. A demo as the supplemental material can be acquired with this paper for detailed view.

5. CONCLUSION AND FUTURE WORK

In this paper, we present a novel method of automatic preview generation for 3D mesh animation. A new measure, namely inter-frame surface saliency, which contains both of motion information between frames and surface saliency in each frame is introduced. An optimal camera path is generated by interpolating the representative viewpoints which are selected in key frames following our new measure and the camera smoothness constraint. Finally, a brief but informative preview is created by moving the camera along this path with frame rate control. In fact, using the inter-frame motions to produce a more logical preview for the mesh animation have not been considered before.

There are two aspects we want to improve in the future. To begin with, segmentation in this paper only apply to temporally coherent mesh sequence that has no topology changes. It is obvious that preview for temporally incoherent mesh sequence, which implies varying connectivities as well as topol-



(a) Handstand

(b) March

Fig. 7. Scenes extracted at key frames from the previews of ”Handstand” and ”March”.

ogy, is very meaningful. Moreover, it is great if our method can be extended to handle scenes with more than one object.

6. REFERENCES

[1] P. P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, “Automatic view selection using viewpoint entropy and its application to image-based modelling,” in *Computer Graphics Forum*. 2003, vol. 22, pp. 689–700, Blackwell Publishing, Inc.

[2] V. Hlaváč, A. Leonardis, and T. Werner, *Automatic selection of reference views for image-based scene representations*, Springer, 1996.

[3] T. Denton, M. F. Demirci, J. Abrahamson, A. Shokoufandeh, and S. Dickinson, “Selecting canonical views for view-based 3-d object recognition,” in *Pattern Recognition(ICPR), IEEE International Conference on*, 2004, vol. 2, pp. 273–276.

[4] D. R. Roberts and A. D. Marshall, “Viewpoint selection for complete surface coverage of three dimensional objects.,” in *BMVC*. Citeseer, 1998, pp. 1–11.

[5] S. R. Han, T. Yamasaki, and K. Aizawa, “Automatic preview video generation for mesh sequences,” in *Image Processing (ICIP), IEEE International Conference on*, 2010, pp. 2945–2948.

[6] W. Saleem, W. Song, A. Belyaev, and H. P. Seidel, “On computing best fly,” in *Proceedings of the 23rd spring conference on computer graphics*, 2007, pp. 143–149.

[7] N. Burtnyk, A. Khan, G. Fitzmaurice, and G. Kurtenbach, “Showmotion: camera motion based 3d design review,” in *Proceedings of symposium on Interactive 3D graphics and games*. ACM, 2006, pp. 167–174.

[8] P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, “Viewpoint selection using viewpoint entropy,” in *VMV*, 2001, vol. 1, pp. 273–280.

[9] S. Zhao, W. T. Ooi, A. Carlier, G. Morin, and V. Charvillat, “3d mesh preview streaming,” in *Proceedings of the 4th ACM Multimedia Systems Conference*. ACM, 2013, pp. 178–189.

[10] R. Arcila, C. Cagniart, F. Hétroy, E. Boyer, and F. Dupont, “Segmentation of temporal mesh sequences into rigidly moving components,” *Graphical Models*, 2012.

[11] R. Arcila, K. S. Buddha, F. Hétroy, F. Denis, and F. Dupont, “A framework for motion-based mesh sequence segmentation,” in *Int. Conf. on Comp. Graphics, Visual. and Comp. Vision, WSCG*, 2010.

[12] B. KP. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[13] E. Dichter, “What’s in an image,” *Journal of consumer marketing*, vol. 2, no. 1, pp. 75–81, 1985.

[14] C. H. Lee, A. Varshney, and D. W. Jacobs, “Mesh saliency,” in *ACM Transactions on Graphics (TOG)*. ACM, 2005, vol. 24, pp. 659–666.

[15] K. Shoemake, “Animating rotation with quaternion curves,” *ACM SIGGRAPH computer graphics*, vol. 19, no. 3, pp. 245–254, 1985.

[16] C. deSperati and P. Viviani, “The relationship between curvature and velocity in two-dimensional smooth pursuit eye movements,” *The Journal of Neuroscience*, vol. 17, no. 10, pp. 3932–3945, 1997.