I-035

Frame Rate Up-conversion in Time-Varying Mesh by Motion Information

Jianfeng Xu　　Toshihiko Yamasaki　　Kiyoharu Aizawa
The University of Tokyo

1. Introduction

Time-Varying Mesh (TVM), which is composed of a sequence of 3D mesh models, is attracting more attention in the past decade. Several generation systems are reported such as [1, 2], where multiple cameras are installed in a studio and dynamic scene is captured by these synchronized videos. Our TVM sequences are generated in NHK's studio [2], where stereo matching was used to efficiently refine the 3D model obtained by the volume intersection method. It is necessary to up-convert the frame rate for those with a low frame rate or large motion, which is our purpose in this paper.

The mesh models in TVM are in low level, which have neither structural information in spatial domain nor explicit corresponding information in temporal domain. Only three types of information are available in each frame (or mesh model): the positions of the vertices, represented by (x, y, z) in a Cartesian coordinate system, the connection information for each triangle that provides topological information of the vertices, and the color information attached to each vertex.

Our task is to interpolate two intermediate frames between every two neighboring frames in the original sequence. The mid-frames should be smoothly transited from one to another. In this paper, we propose an approach based on a semantic human model. We assume that a human body can be regarded as a piecewise-rigid articulated object with a tree structure. It is much more convenient to search the motion vectors in volumetric models instead of mesh models. After converting the mesh models to volumetric models frame by frame, a fast motion estimation method is proposed to reduce greatly the computational cost while keeping the performance. By the motion vectors for each body segment, immediate mesh models are generated by linear interpolation.

2. Semantic human model

It is popular to regard human body as an articulated object with a tree structure composed of segments, where the kinematic parameters include the center of torso, nine joint positions between the segment and its parent node (there are totally ten segments in our human model).

We also assume that each segment has the near-rigid motions. Due to the tree structure, only the root (torso) has full six freedoms, and other segments have three freedoms in rotation motion round their joints respectively. Therefore, our semantic human model is an articulated ten-segment human model with piecewise-rigid motions, which has totally 33 freedoms. As Fig. 1 shows, a point $P(t1)$ in segment 2 at the $t1$-th frame (called reference frame) will move to $P(t2)$ at the $t2$-th frame (called current frame) as Eq. (1). Those points in other segments have similar formulae. Note that $P(t1)$ and $P(t2)$ may be a vertex in mesh model, a voxel in volumetric model or a joint in semantic human model.

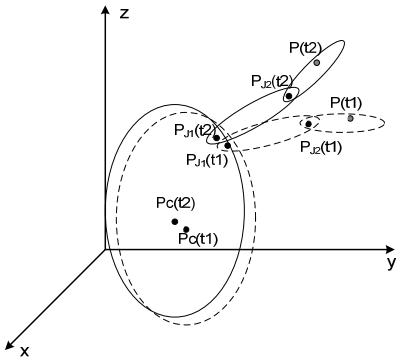


Fig. 1. Motion model in TVM sequence.

Fig.2.Mesh model, volumetric model, and segmentation result.

$$P(t2) = P_c(t2) + R_0(t1,t2)(P_{J1}(t1) - P_c(t1)) \\ + R_1(t1,t2)(P_{J2}(t1) - P_{J1}(t1)) + R_2(t1,t2)(P(t1) - P_{J2}(t1)) \quad (1)$$

where $R_i(t1,t2)$ is the rotation matrices of segment i, which can be expressed by the three Euler angles. P_c denotes the torso center. P_{Ji} is the joint position of segment i. $P(t2)$ is related to all the motions from the root to the current segment, which requires our search should be from the root to the leaf in each branch. Our basic idea for deformation is to interpolate mid-frames by the motion vectors between the $t1$-th and $t2$-th frames. Therefore, the key issue is to estimate the motions between the two frames.

In our implementation, we segment the volumetric model manually to define the ten segments in the first frame. We use the parity counting method as [3] to convert the mesh model to volumetric model. Fig. 2 gives one example.

3. Motion estimation

Motion estimation is to calculate the rotation matrices of the ten body segments between the reference volumetric model and the current volumetric model. For a candidate of rotation matrix, the voxels in the reference frame are moved to the current frame as Eq. (1) shows. Then, a cost function is calculated based on the distance field in the current frame. The cost function is optimized by a search strategy to extract the best rotation matrix. Then, the reference volumetric model is deformed by the estimated rotation motions, which will be used as the reference model for the next frame. The semantic human model in the current frame is also

calculated by Eq. (1), where a joint can be regarded as a point in its parent node. Therefore, motion estimation can continue in successive frames. In this section, we will introduce how to define the cost function and search the rotation matrices.

The distance of a voxel is defined as the length of the shortest path (using 6-neighbor definition) to the subject surface for the outside of the volumetric model, and 0 for the surface and inside of the model. Since the motion cannot enter the inside of the model, it does not need to calculate the (negative) distances inside the model so that computational cost can be decreased. A calculation method is given in our previous work [4].

To measure how different the current volumetric model is from the deformed reference volumetric model, a cost function is defined as the sum of distance values of all the voxels which are deformed from the reference volumetric model.

$$cost(R_i(t1,t2) = \sum_i cost_i$$
$$cost_i = \sum_{v(t1)\in segment(i)} D(v'(t2)) \qquad (2)$$

where $v(t1)$ is a voxel in the $t1$-th frame (the reference frame), $v'(t2)$ is the position where $v(t1)$ moves to in the $t2$-th frame (the current frame), calculated by Eq. (1). $D(v'(t2))$ is the distance value of $v'(t2)$. Therefore, motion estimation can be described as an optimization problem of the cost function.

Due to the high freedoms in the semantic human model, the computational cost is very heavy. Therefore, we just separate it into each segment and minimize the cost of a segment $cost_i$. Since the torso is the root whose motion will be used in all other segments, it is the first segment for motion estimation. Then, in each branch, motion estimation is done from parent node to child node. Therefore, only the rotation motion for current segment is unknown in Eq. (1).

To search the motion of a segment, a fast search algorithm is still necessary due to the large motion in our TVM sequences. We propose a gradient search method to reduce the computation, which is a trade-off between computational cost and performance. It should be mentioned that the cost function based on distance field makes the gradient search method work well, which is the essential reason to construct the distance field. The procedure for a segment is:

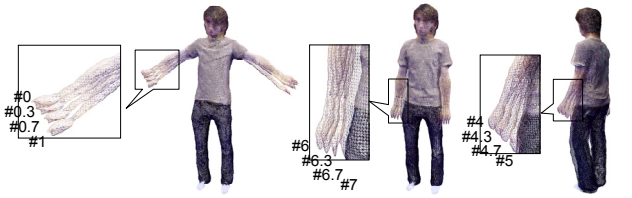
(a) Perform full search of rotation matrix for a segment in a small range (5x5x5 in our experiments).

(b) Modify the search center to the position with minimal cost in step (a). Iterate step (a) and (b) until no new center is found.

4. Mesh Interpolation

Motion vectors are searched in the volumetric models. To generate mid-frames, we firstly deform the mesh model at $t1$ to $t2$ by the estimated motion vectors. Thus, the original mesh model at $t1$ and the deformed mesh model at $t2$ share the same topology with explicit vertex correspondence. A mid-frame is interpolated linearly by the two mesh models.

To utilize the motion vectors in mesh models, it is necessary to map the vertices in the reference mesh model to the ten-segment human model. Only the volumetric model in the first frame is segmented manually, and those volumetric models in other frames are not segmented. However, the deformed volumetric



(a) Frame #0-#1 in TVM1 (b) Frame #6-#7 in TVM1 (c) Frame #4-#5 in TVM2

Fig. 3 Interpolated mid-frames in TVM sequences.

model from the first frame is available, which has the segment information. A vertex is attached to a segment where there is a nearest voxel in the reference volumetric model. Then, we calculate the vertex position in the deformed mesh model from $t1$ to $t2$ by Eq. (1). Lastly, we interpolate the vertex position in the deformed mesh model at tm ($t1<tm<t2$) by Eq. (3).

$$P(tm) = P(t1) + \frac{tm - t1}{t2 - t1}(P(t2) - P(t1)) \qquad (3)$$

where $P(t1)$ is the vertex position in the reference mesh model at $t1$, $P(t2)$ is the vertex position in the deformed mesh model at $t2$, calculated by Eq. (1). The topology and color data are the same as those in the reference mesh model at $t1$.

Two sequences are tested including a 9-frame sequence with large arm motion and an 11-frame sequence with walking motion. Fig. 3 shows some examples, where (a) and (b) come from the arm motion sequence and (c) comes from the walking motion sequence. The experiments demonstrate the deformed mesh models can smoothly transform between two original frames.

5. Conclusions

In this paper, a deformation method for frame rate up-conversion is presented. A semantic human model is defined with assumption of the articulated object with piecewise-rigid motions. Then, the motion vectors for ten segments are extracted by optimizing a cost function, which is based on the distance field in the volumetric model. The proposed motion estimation method reduces greatly the computational cost while keeping the performance. Lastly, a linear interpolation method is employed to deform the mesh model in the reference frame using the extracted motion vectors. Our experimental results demonstrate the effectiveness of our algorithm.

6. Reference

[1] T. Kanade, P. Rander, and P. Narayanan, “Virtualized reality: constructing virtual worlds from real scenes,” IEEE Multimedia, Vol. 4, No. 1, pp. 34-47, Jan./Mar. 1997.

[2] K. Tomiyama, Y. Orihara, and et al., “Algorithm for dynamic 3D object generation from multi-viewpoint images,” in Proc. of SPIE, Vol. 5599, pp. 153-161, 2004.

[3] F. S. Nooruddin, and G. Turk, “Simplification and repair of polygonal models using volumetric techniques,” IEEE Trans. on Visualization and Computer Graphics, Vol. 9, No. 2, pp. 191-205, April-June 2003.

[4] J. Xu, T. Yamasaki, and K. Aizawa, “Mutual information in 3D video,” 3DTV-CON 2007, paper #24, Greece, May 2007.