International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:3, 2013

An Efficient 3D Animation Data Reduction Using Frame Removal

Jinsuk Yang, Choongjae Joo, and Kyoungsu Oh

***Abstract*—**Existing methods in which the animation data of all frames are stored and reproduced as with vertex animation cannot be used in mobile device environments because these methods use large amounts of the memory. So 3D animation data reduction methods aimed at solving this problem have been extensively studied thus far and we propose a new method as follows. First, we find and remove frames in which motion changes are small out of all animation frames and store only the animation data of remaining frames (involving large motion changes). When playing the animation, the removed frame areas are reconstructed using the interpolation of the remaining frames. Our key contribution is to calculate the accelerations of the joints of individual frames and the standard deviations of the accelerations using the information of joint locations in the relevant 3D model in order to find and delete frames in which motion changes are small. Our methods can reduce data sizes by approximately 50% or more while providing quality which is not much lower compared to original animations. Therefore, our method is expected to be usefully used in mobile device environments or other environments in which memory sizes are limited.

***Keywords*—**Data Reduction, Interpolation, Vertex Animation, 3D Animation.

I. Introduction

ONE of difficulties in developing 3D applications such as games to be used in mobile environments in which memory sizes are limited is that the sizes of 3D animation data are too large. Since data sizes become larger, larger memory capacities are required and data loading time is lengthened so that hardware of better specifications is required. To improve these conditions, reducing the sizes of 3D animation data is essential and related methods have been extensively studied thus far [1]-[4].

We introduce a method to find and remove frames in which motion changes are small using the information of joint locations in the 3D model in order to reduce the size of 3D animation data and reproduce the removed frames using the interpolation of remaining frames.

A. Our Method

Among all animation frames, frames in which motion changes are small are less easily perceived when they have been removed compared to frames in which motion changes are large. Therefore, errors are relatively smaller when the removed frames have been reconstructed using remaining frames. Our method uses the foregoing characteristics and uses the following two ways in order to find and remove frames in which motion changes are small.

1. Use the sum of the accelerations of joints

We obtain the accelerations of individual joints in each frame. Frames in which the sum of the accelerations of joints is smaller than threshold T_a are removed from the entire frames

2. Use the standard deviations of the accelerations of joints

Using the accelerations of joints obtained as set forth under first, we obtain the standard deviations of the accelerations of individual joints. Frames in which the standard deviations of the accelerations of all joints are smaller than threshold T_d are removed from the entire frames.

When the animation is played, the frames removed using the above two ways are reconstructed using the interpolation of remaining frames.

This paper is composed as follows. In chapter II, related studies are described and in chapter III, methods to remove frames in which motion changes are small and reconstruct the removed frames using interpolation. In chapter IV, experimental results are explained and in chapter V, conclusions are described.

II. Related Work

In this chapter, related studies are described which can be largely divided into two types as follows.

A. Vertex-Connected Relationship Compression

The first method is to reduce data sizes by compressing vertices' connected relationships. Vertices' connected relationships are made by listing three of the numbers of vertices that form triangles in consecutive order. These vertex-connected relationships can be compressed when many vertices are connected to one point or when there are certain rules. [5], [6] indicated the connected relationship of the vertex coming next using a certain sign in order to simply indicate vertices' connections thereby compressing 3D data. This method provides high rendering speeds since it does not require costs for interpolation. However, this method has a shortcoming of showing limited compressibility.

B. Vertex Compression

The second method is to reduce data sizes by storing the locations of vertices in a method different from methods used

Jinsuk Yang is with the Department of Global Media, Soongsil University, Seoul, Korea (phone: +82-2-82-3924; fax: +82-2-822-3622; e-mail: ispio@ssu.ac.kr).

Choongjae Joo is with the Department of Global Media, Soongsil University, Seoul, Korea (e-mail: goodbye302@gmail.com).

Kyoungsu Oh is with the Department of Global Media, Soongsil University, Seoul, Korea (e-mail: oks@ssu.ac.kr).

previously. [7] uses the property of vertices that can be expressed using arbitrary values since they are generally at locations close to the law of parallelograms. He compresses 3D data by making a virtual vertex from three vertices of a triangle and storing differences from the vertices actually connected to the triangle. Although this method has high compressibility, it has a shortcoming that it is a little complicated and difficult to implement this method.

III. Algorithm

In this chapter, 3D animation data reduction method using frames removal and interpolation. In section *A*, a method to find and remove frames in which motion changes are small is described and in section *B*, a method to reconstruct the removed frames using remaining frames when the animation is played is described. Fig. 1 shows an overview of the method proposed by us (The source of images is [8]).

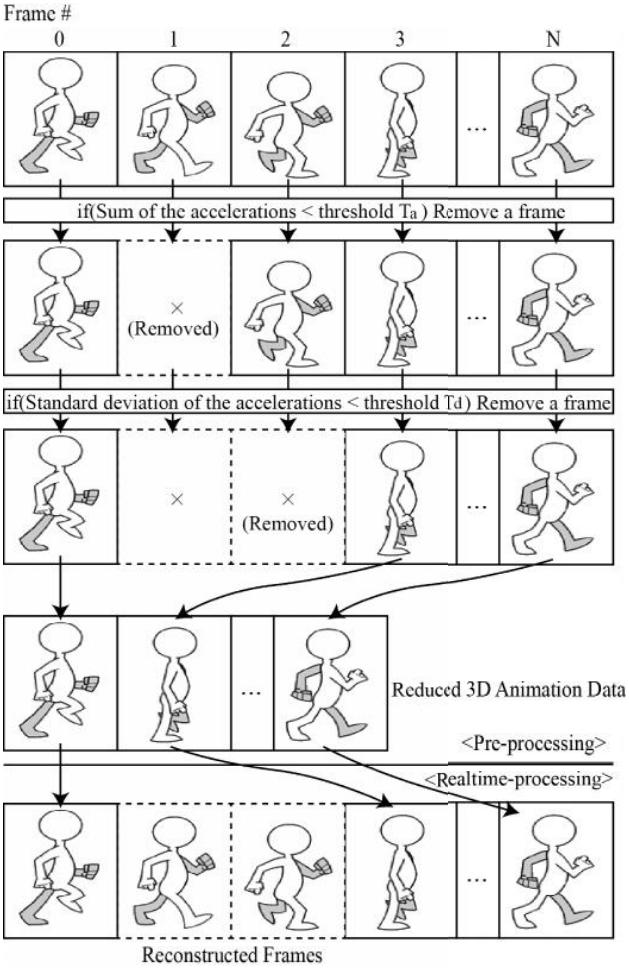

Fig. 1 An overview of our method

A. Frame Removal

In section *A*, a method to find and remove frames in which motion changes are small is described. Frames in which motion changes are small are less easily perceived when they have been removed compared to frames in which motion changes are large and errors are relatively smaller when the removed frames have been reconstructed using remaining frames. For frames in which motion changes are small, the amounts of movements of 3D model joints are small. The amounts of movements increase as the accelerations of the joint increase. Using these properties, we obtain the accelerations of individual joints and remove frames in which the sums of accelerations are large. This process consists of the following three steps.

1) Calculation of the Accelerations of Individual Joints in Each Frame

In Fig. 2, the speed of p^{-1} is the value obtained by deducting the location of p^{-1} from the location of p and the speed of p is the value obtained by deducting the location of p from the location of p^{+1}. Therefore, the current acceleration of joint p of frame t is calculated using (1) as follows.

$$\begin{aligned} v.x &= \left|(p.x - p^{-1}.x) - (p^{+1}.x - p.x)\right| \\ v.y &= \left|(p.y - p^{-1}.y) - (p^{+1}.y - p.y)\right| \\ v.z &= \left|(p.z - p^{-1}.z) - (p^{+1}.z - p.z)\right| \end{aligned} \tag{1}$$

where, $v(x, y, z)$ is the acceleration vector of p.

Using this method, the accelerations of individual joints in each frame are calculated.

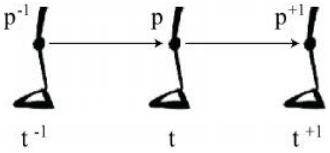

Fig. 2 The movement of joint

The p is position of joint and t is frame number

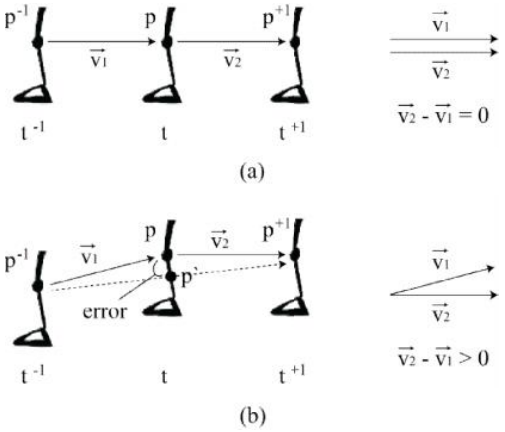

Fig. 3 The changes in the location of joint p in three successive frames

2) Frame Removal Using the Sums of the Accelerations of Joints in Each Frame

Fig. 3 shows changes in the location of joint p in three successive frames. In the case of (a), the joint moves in uniform motion. In this case, errors barely occur even if frame t is removed and later reconstructed by interpolating it between frames t^{-1} and t^{+1}. On the other hand, (b) is a case where accelerations have been changed. In this case, if frame t is removed and later reconstructed by interpolating it between p^{-1}, the joint location in frame t^{-1} and p^{+1} , the joint location in frame t^{+1}, an error as large as the difference between the locations of p and p' will occur.

Therefore, using the accelerations of individual joints calculated as set forth above, the sums of accelerations by frame are obtained and if differences in the sums of accelerations between frames are smaller than threshold T_a, the relevant frames are removed. If T_a becomes larger, the number of frames removed will be increased so that data sizes can be reduced further but errors will become larger.

3) Frame Removal Using the Standard Deviations of the Accelerations of Joints by Frame

After removing frames using the sums of the accelerations of joints, frames in which motion changes can be found and removed once more among remaining frames using the standard deviations of the accelerations of individual joints. Fig. 4 shows a case where the sums of the accelerations of joints are the same but the standard deviations of the accelerations are different. (a) has a smaller standard deviation of the accelerations compared to (b). In this case, if frame t is removed and later reconstructed by interpolation, (a) should have a smaller error compared to (b). Therefore, frames in which the standard deviations of the accelerations of individual joints are smaller than threshold Td are removed. The standard deviations of the accelerations are calculated using (2) as follows.

$$\sigma = \frac{\sum_{k=1}^{n} \sqrt{(x_k - m)^2}}{n} \tag{2}$$

where, n is the number of joints, x_k is the acceleration of the k-th joint, and m is the average of the accelerations of joints.

B. Reconstruction of Removed Frames

When animations are played, removed frames should be reconstructed using remaining frames to make natural animations. We reconstruct removed frames using linear interpolation. p', the location of the a reconstructed joint is calculated using (3) as follows.

$$\begin{aligned} p'.x &= p^{-1}.x + (p^{+1}.x - p^{-1}.x) \times (t - t^{-1})/(t^{+1} - t^{-1}) \\ p'.y &= p^{-1}.y + (p^{+1}.y - p^{-1}.y) \times (t - t^{-1})/(t^{+1} - t^{-1}) \\ p'.z &= p^{-1}.z + (p^{+1}.z - p^{-1}.z) \times (t - t^{-1})/(t^{+1} - t^{-1}) \end{aligned} \tag{3}$$

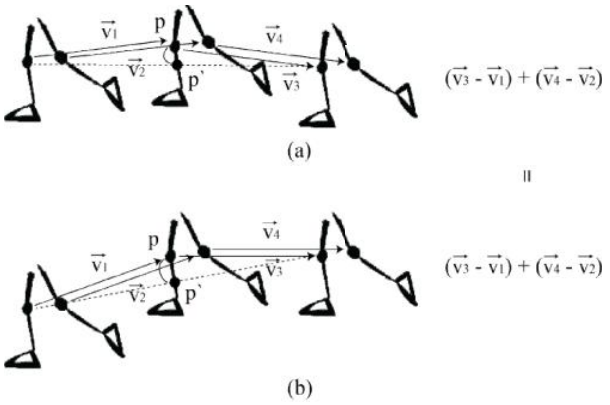

Fig. 4 A case where the sums of the accelerations are same, but the standard deviations of the accelerations are different.

IV. Results

We exported a 3D model made using Autodesk 3ds MAX to an md5 and loaded the file in an OpenGL environment to implement the 3D model. The 3D model used in the experiment had a total of 120 animation frames and 34 joints.

A. Comparison of the Results of Frame Removal Using the Sums of the Accelerations of Joints by Frame

Fig. 5 shows a comparison of the quality of an animation reconstructed after removing the frames using the sums of the accelerations of joints with the quality of the original animation. (a) is the original animation, and (b) and (c) are images reconstructed after removing the frames using Ta values of 0.5 and 0.8 respectively. Among the removed frames, the 75th frame which had the largest error was captured. (b) involves a reduction ratio of approximately 28% since only 87 frames were stored out of a total of 120 frames and its image quality is slightly different compared to (a) in the hand area but there is no big difference in general. On the other hand, (c) involves a reduction ratio of approximately 53% since only 56 frames out of a total of 120 frames and it has remarkable differences in image quality in the hand and glasses. Therefore, in this case, removing approximately 30% of frames is efficient.

B. Comparison of the Results of Frame Removal Using the Standard Deviations of the Accelerations of Joints by Frame

Fig. 6(b), 6(c) show images reconstructed after removing the frames using Ta values of 0.4 and 0.7, and Td values of 0.003 and 0.0045 respectively. As with section A, the 75th frame was captured. As shown in Table 1, whereas there is no big difference in quality when the reduction ratio was 28% (Fig. 6(b), Fig. 7(b)), the case where the standard deviations of the accelerations were used shows better quality when the

reduction ratio was 53% (Fig. 6(c), Fig. 7(c)).

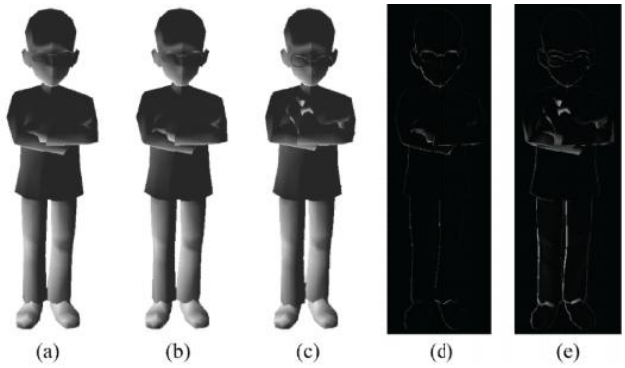

Fig. 5 A comparison of the quality of an animation reconstructed after removing the frames using the sums of the accelerations with the quality of the original animation. (a) is original animation image, (b) and (c) are reconstructed images, (d) and (e) are error images of (d), (e) respectively. In (d) and (e), bright area is error

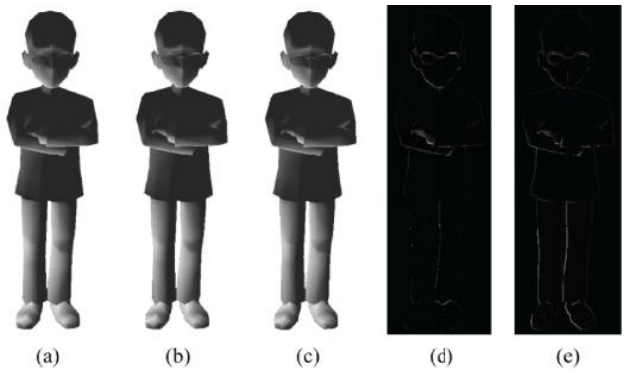

Fig. 6 A comparison of the quality of an animation reconstructed after removing the frames using the standard deviation of the accelerations and sums of the accelerations with the quality of the original animation. In common with Fig. 5, (a) is original animation image, (b) and (c) are reconstructed images, (d) and (e) are error images of (d), (e) respectively. In (d) and (e), bright area is error

V. Conclusion

We proposed a method to find and remove frames in which motion changes are small using the accelerations of joint locations in the 3D model and the standard deviations of the accelerations in order to efficiently remove 3D animation data and reproduce the removed frames through linear interpolation when the animation is played. Our method has high efficiency since it reconstructs removed frames without any great decline in quality when the reduction ratio is not higher than approximately 50%. When reconstructing removed frames, linear interpolation using quadratic equations were applied in the present study. If multi-dimensional linear interpolation using cubic equations or higher order equations is applied, quality declines will be reduced further. It is expected that, if our method is used in mobile device environments, more than two times larger animation data can be stored than those that can be stored now.

Acknowledgment

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency-(KOCCA) in the Culture Technology(CT) Research & Development.

References

[1] M. M. Chow, "Optimized geometry compression for real-time rendering," in *IEEE Visualization"97 Conference Proceedings*, pp. 347-354, 1997.
[2] J. Rossignac, "EdgeBreaker : Connectivity Compression for Triangle Meshes," *IEEE Transactions on Visualization and Computer Graphics*, pp. 47-67, 1999.
[3] M. Isenburg and J. Snoeyink, "Face Fixer: Compressing Polygon MeshesWith Properties," *in ACM SIGGRAPH 2000 Conference Proceedings*, pp. 263-270, 2000.
[4] M. Isenburg and P. Alliez, "Compressing polygon mesh geometry withparallelogram prediction," *in Visualization'02 Conference Proceedings*, pp. 141-146, 2002.
[5] P. Alliez and M. Desbrun, "Valence-Driven Connectivity Encoding of 3DMeshes," in *Eurographics 2001 Conference Proceedings*, pp. 480-489, 2001.
[6] M. Eck, T. Derose, T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle, "Multiresolution analysis of arbitary meshes," in R. Cook, editor, *SIGGRAPH 95 Conference Proceedings*, pp. 173-182. 1995.
[7] K. Mamow, T.Zaharia and F.Preteux, "Frame-based Compression of Animated Meshes in MPEG-4," in *ICME 2008*, pp. 1121-1124, 2008.
[8] http://www.idleworm.com