ROBUST WATERMARKING OF 3D SKINNING MESH ANIMATIONS

Rakhi Motwani, Ameya Ambardekar+ , Mukesh Motwani*, Frederick C. Harris, Jr.**

University of Nevada, Reno USA & Vishwakarma Institute of Technology, Pune India+*

ABSTRACT

This paper presents a novel robust watermarking algorithm for 3D skinning mesh animations by embedding the watermark in skin weights in addition to key frames. This method can be used for copyright protection, tamper proofing or content annotation purposes. The proposed watermark is immune to noise attacks on key frames and skin weights, key frame dropping and frame modification and is perceptible invisible as well. Experimental results verified that the proposed algorithm has good robustness against attacks and maintains invisibility by preprocessing the animation data sets by key frame decimation.

Index Terms — Animation, Copyright protection, Robustness, Bones

1. INTRODUCTION

3D computer animations are now widely used and accepted in gaming and entertainment industry. Typically, the graphics artist would create the mesh used for animation or use an existing mesh produced by another artist. 3D watermarking algorithms provide content protection for the mesh vertices. However, that does not protect the creator of the animation, who produces the animation using the 3D model. There have been several papers focusing on protecting the mesh but animation protection is still a wide open research field. This paper addresses this need by proposing a novel algorithm for skinned mesh animations. Skin meshes or skeletal meshes are one of the most important subjects in a 3D world. Skinned animations [1] stretch the mesh around the bones and assign a skin weight to each vertex in the model so that the skin can move over the bones especially at the joints. Skinned animations enable efficient hardware rendering, rest pose editing, and deformable collision detection. The process of watermarking mesh animation poses problems related to mesh watermarking scheme properties, such as imperceptibility of the watermark, and robustness against data modification attacks. Mesh animation can be subject to motion editing operations done intentionally (adversary) or unintentionally (user – an artist) by using motion editing software such as 3ds Max and Maya [2]. The additional requirements of the mesh animation watermarking technique are not to destroy or modify the existing watermark which may exist in the mesh.

2. RELATED WORK

There has been considerable amount of research on watermarking techniques for 3D meshes. In [3], Ohbuchi et al. proposed TSQ (Triangle Similarity Quadruple), TVR (Tetrahedral Volume Ratio), MDP (Mesh Density Pattern) watermarking based on geometrical or topological modifications. But these watermarking algorithms can't be simply extended to 3D animations since the data to be watermarked are transformation matrices and skin weights. The Human visual system is more sensitive to slight changes in the animation path affected by modifying values in the transformation indices. Thus, the strength of watermark inserted into animation is lot less as compared to insertion in meshes. Also, unlike in meshes where data transformation into spectral domain gives more robustness, such transformation cannot be applied to transformation matrix or skin weights. There has been very limited research work related with watermarking [4] motion data. In [5], a watermark bit is embedded into the change rate of gradient of neighborhood key values in reference key value that has large orientation angle among quatemion key values.

For real-time animation, key frame animation is widely used in 3D graphics. In key framing, the animated key values in several important frames among the entire animation frames are used to generate the rest of the frames by spherical or linear interpolation. 3D editing software such as 3ds Max [2] also allow these key frames to be set by the artist or allow automatic insertion of these key frames. A function curve is typically used as in [2] to decide the path of the moving object and several key points are stored on this function curve. So a typical animation would have several hundred key frames to achieve smooth interpolation. This poses as a challenge for robust watermarking of the animation since key frame decimation would eliminate the watermark. In [6], key frames are ordered according to a cost function and the error is watermarked. Hartung et al in [7], inserted watermark in the semantics of the way the animated head moves. There has been no published research yet specifically focusing on protection of skinned animations. This paper addresses this issue and presents watermarking technique for the widely used key frame based animations. Experimental results verified that the proposed algorithm has good robustness against geometrical attacks and timeline attacks.

3. OVERVIEW OF SKINNED MESH ANIMATIONS

Doug et al in [1] provide a comprehensive overview of skinned mesh animations. According to [1], a vertex skinning transform, T^t_i approximates a sequence of input meshes. The mesh sequences have predefined bone transformations, T^t_b, for skeleton-free meshes and the skin's vertex-bone dependencies, Bi, and corresponding vertex bone weights, w_{ib} specified in the animation file format.

Let the sequence of S meshes have deformed vertex positions $\mathbf{P} = (p^1, p^2, \ldots, p^S)$, where, $p^t \quad R^{3N}$, for N vertices. For convenience, we will refer to the index t as "time," even though the sequence may not have a time interpretation. Given these meshes, a skinning transformation T^t at each sequence step, transforms the undeformed rest pose points, Þ, to approximate p^t:

$$p^t \quad T^t Þ, \qquad t = 1 \ldots S, \qquad (1)$$

with similar equations possible for vertex normals. Assuming linear blend skinning, the transformation matrix for vertex i is

$$T^t_i = \quad (w_{ib}T^t_b) \qquad (2)$$

but where the transforms T^t_b need not be rigid.

The transformation matrix defines a local transform for a frame matrix. It transforms the mesh vertices to the space of the bone. When concatenated to the bone's transform, this provides the world space coordinates of the mesh as affected by the bone.

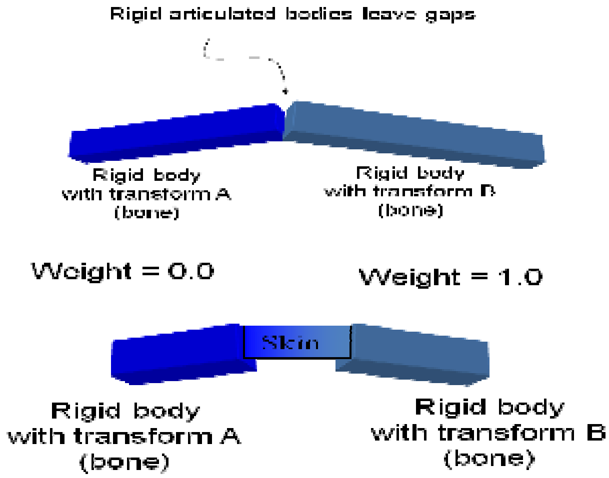


Fig 3.1 Skinning to join 2 bones

3.1 Skin Weights and Vertex Blending

While selecting key frame for transformation all joints in motion data are not equally important. It is the parent joints which are more important. A way is needed to introduce 'flexible' triangles that can straddle one or more hierarchy levels. So for finding the joints which are more important skin weight is used. Fig 3.1 taken from [8] illustrates the purpose of skinning. While transforming the parent joint it has effect on its child also. So parent joint is of greater importance. Vertex Blending allows us to create flexible, non-rigid skins. Numerical weights from 0 to 1 are specified for each vertex in the mesh. A value of 0 corresponds to no movement of the skinned mesh over the bone whereas 1 corresponds to having maximum fluidity for the mesh to move over the bone. Typically, vertices at the joints have skin weights close to 1. The weights represent a level of influence between the various hierarchy levels. Traditionally, model animation is performed by transforming separate hierarchical rigid meshes. For characters and other organic shapes, a single mesh solution is preferred, to avoid cracks and self-intersection artifacts.

4. ALGORITHM

To overcome the challenge posed by statistically redundant key frames, frame decimation is required to select important key frames. Shiyu Li et al in [9] have proposed a key decimation algorithm to automate the process. If we have an animation with only important key frames then watermarking will be more robust. Using frame decimation prior to watermarking enables entire animation data to be represented by few key frames that are perceptually important.

Fig. 4 gives a system overview of the proposed algorithm. The number of bones varies according to the animation. The bone is covered with mesh vertices having a skin weight attached to each one of them. The animation is carried out by key frames which are made up of transformation matrices. Each animation set contains number of key frames. The watermark is randomly added in the key frames and the skin weights. Thus, the watermark is spread in all the dominant key frames of the animation.

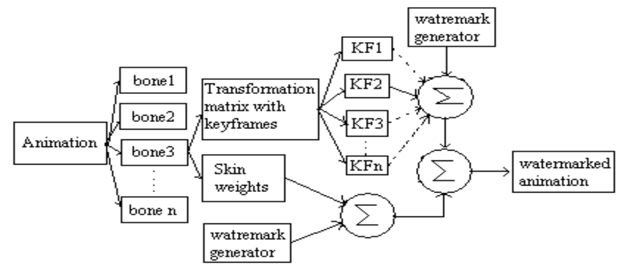


Fig. 4 Block diagram of the system

4.1 Algorithm for key frame watermarking

Step 1: Each bone has a different animation set consisting of key frames. Select the animation set of the bone to insert the watermark.
Step 2: Reduce key frames using frame decimation algorithm. [9]
Step 3: Randomly select transformation matrix from the new decimated lot.
Step 4: Modify the elements of the transformation matrix randomly by a threshold less than the permissible limit. The permissible threshold was determined to be 1.7 experimentally.

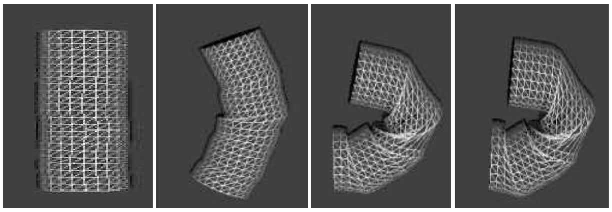
4.2 Algorithm for skin weight watermarking

Step 1: Select the bone first. The bone is covered with mesh vertices with different skin weights.
Step 2: Select vertexes randomly associated with the bone.
Step 3: If selected vertex has skin weight which is already 1.00 then skip that as the maximum permissible range is 0.00 to 1.00 and check for step 2. Otherwise proceed for step 4.
Step 4: Add the watermark content in it. Add any random value which is less than the permissible limit.
Step 5: Move on the next vertex and again check for step 2.

4.3 Algorithm for Watermark extraction

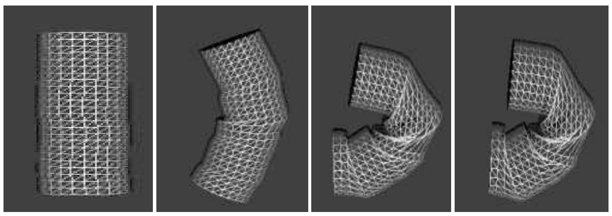
As our system uses non-blind watermark, original animation and original watermark are needed to extract the watermark from the attacked model by subtraction. Correlation is computed from the original and attacked watermark and a threshold is chosen to determine if the animation is attacked.

Fig. 4.1 shows some of the predominant key frames in the original animation while Fig. 4.2 shows the corresponding key frames which are watermarked. Comparison of the two figures illustrates the imperceptibility of the watermark.



(a) (b) (c) (d)

Fig 4.1 Predominant key frames of original animation



(a) (b) (c) (d)

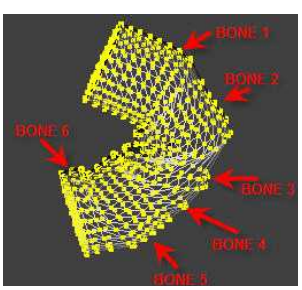
Fig 4.2 Watermarked key frames

5. EXPERIMENTS

For the purpose of testing the attacks on the watermark, we tested the algorithm on several animations. The animation shown in figure has 6 bones. This animation is provided as a sample in DirectX SDK. The mesh contains 977 vertices having 1680 faces. Each face is made up of combination of three vertices. The animation is carried out by interpolating the animation key frames. The key frames are associated with the animation sets. Each bone has its own animation set. Depending on the movement of bones required in the animation, number of key frames in the animation set changes. This is because the all bones are moving simultaneously. In our case each animation set contains 81 key frames. After frame decimation, the key frames were reduced to 52. These key frames are made up of 4x4 transformation matrix.

While doing the experiments, we observed that, the maximum permissible limit for the modification of the elements of the transformation matrix in a key frame is 1.7. The maximum permissible limit for modifying the skin weights is 0.02. Hence, during watermarking process, the data embedded is less than these limits in either of the case.

Bone Name	No. of Vertices	Bone Name	No. of Vertices
1 At top	426	4	301
2	387	5	301
3	263	6 At bottom	204



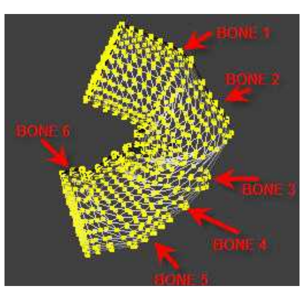
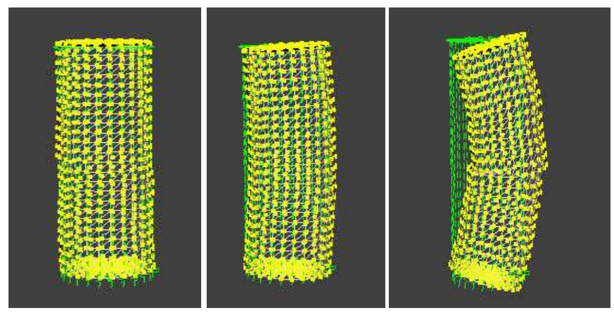


Fig 5.a Original animation Fig 5.b Watermarked animation

5.1 Key Frame dropping

In this experiment, 2 sets of animations were used. In the first set, no frame decimation was done to justify the importance of frame decimation as part of watermarking process. In the undecimated set, due to redundant key frames, if we randomly delete 6 consecutive key frames we don't see any change in animation. Only when 7th is deleted then we can see noticeable change in the animation. The dropped key frames affect the animations because when the animation has all the key frames it is modeled to follow a certain path. By deleting the key frames we remove the path that was set for the animation to follow. Hence it goes straight in the path from where the last key frame is. This is illustrated in the figure 5.1. The cylinder is shown following the boundary in (a) which is the original animation. In (b), when key frames are dropped then the animation starts leaving the boundary due to the reason discussed above. Finally (c) illustrates the condition when large numbers of key frames are deleted showing more bending in the boundary region. In the second set, where decimation was part of the watermarking stage, further key decimation caused severe distortion in the animation path, thereby destroying the watermark as well.



(a) (b) (c)

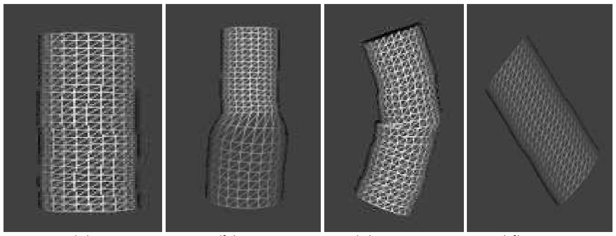
Fig.5.1. Effect of key frame dropping on the animation

5.2 Random noise in key frames

In this experiment, animation is attacked by normally distributed random numbers noise with variance 1 and mean 0. As shown in Table 5.2.1 below, noise affects watermark the most in key frame 72 of bone 4. The watermark is less variant to the noise in case of the key frame 7 of bone 1 and key frame 22 of bone 4. In spite of the noise attack the extracted watermark is still recoverable in all the cases.

Table 5.2.1 Correlation result with extracted watermark

Bone No.	Key-frame No.	Correlation Coefficient	Bone No.	Key-frame No.	Correlation Coefficient
1	0	0.9119	4	72	0.9099
1	7	0.9998	4	22	0.9991
2	14	0.9901	5	40	0.9895
2	23	0.9120	5	55	0.9919
3	0	0.9420	6	68	0.9397
3	7	0.9110	6	81	0.9091



(a) (b) (c) (d)

Fig 5.2 Original animation shown in comparison with the attacked for excessive noise

Figure 5.2 shows the effect of excessive noise on the animation. Original animation in compared with the attacked one in Fig 5.2. While animating the cylinder is straight at the rest position (a), but when attacked at the rest position it bulges out, as shown in (b). While moving down from the initial position it bends as shown in (c). When attacked instead of going down in bent position it goes slanting rightwards vertically, which is shown in (d). As the coefficients of transformation matrix are affected, the key frame is modified completely thereby making the next position of animation uncertain. This effect is clearer in the video which can be seen on [10].

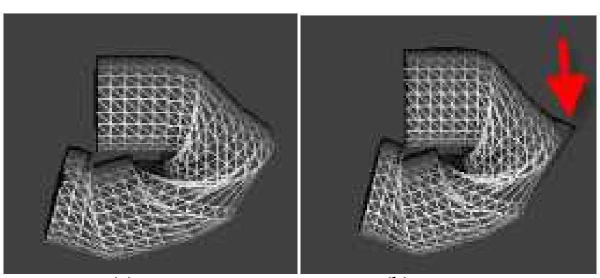
5.3 Random noise in skin weights

Here also we use the same type of noise to attack the animation, i.e., distributed random numbers noise with variance 1 and mean 0.

Table 5.3.1 Correlation results with extracted watermark

Bone Used	Correlation Coefficient	Bone Used	Correlation Coefficient
1 At top	0.9119	2	0.9132
3	0.9998	4	0.9099
5	0.9210	6 At bottom	0.9982

As shown in Table 5.3.1, watermark is affected by the noise mostly in bone 4. It shows least variations to the noise in case of bone 3. In spite of the noise attack the extracted watermark is still recognizable in all the cases. Fig 5.3 shows the skin "flying" off. If subtractive watermark was used, vertex bleeding would be observed.



(a) (b)

(a) is the original frame while (b) shows the bulging by an arrow.

Fig.5.3 Effect of noise attack on skin weights

5.4 Key frame modification

This attack is modification of the keyframes, i.e., if any of the data from the transformation matrix changes then it should be easily perceptible in the correlation result. There may or may not be any change in the animation. This is because if the modified value is within the permissible limits, the change in animation is imperceptible to the human eye, but if the change violates the limit then the same can be perceived in animation itself.

Table 5.4.1 shows us that the watermark is robust to the changes that take place in the animation due to modifications in the key frames.

Table 5.4.1. Correlation result with extracted watermark

Bone No.	Keyframe No.	Correlation Coefficient
1	0	0.9688
1	22	0.9997
4	72	0.2256

6. CONCLUSION

This paper presented a new watermarking algorithm of embedding data into the skin weights of 3D animations in addition to the key frames. Pre-processing of the skinned mesh by frame decimation improved the robustness of the algorithm significantly against key frame dropping and modification attacks. We intend to increase the density of the inserted watermark by finding ways to reduce the perceptible threshold to insert watermark. Future work needs to be directed on algorithmically estimating the influence of skin weights on neighboring vertices so that watermark can be inserted based on the influence of the skin weight instead of randomly selecting the skin weights. We would also like to extend our work to pre-defined animation sets such as animated sign language sequences.

7. REFERENCES

[1] Doug L. James and Christopher D. Twigg, "Skinning Mesh Animations", ACM Transactions on Graphics (ACM SIGGRAPH 2005), 2005

[2] AutoDesk, http://www.autodesk.com

[3] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modification," IEEE Journal on Selected Areas in Communications, vol. 16, no. 4, pp. 551-560, May 1998.

[4] Shuntaro Yamazaki, "Watermarking Motion Data", In Proc. Pacific Rim Workshop on Digital Steganography (STEG04),pp. 177--185, Nov 2004

[5] E.S. Jang, James D.K.Kim , S.Y. Jung, M.J. Han, S.O. Woo, and S.J. Lee, "Interpolator Data Compression for MPEG-4 Animation," IEEE Trans. On Circuits and Systems for Video Technology, vol. 14, no. 7, pp. 989-1008, July 2004.

[6] Shiyu Li, Okuda, Masahiro,,"Watermarking for Progressive Human Motion Animation", IEEE International Conference on Multimedia and Expo, 2007

[7] F. Hartung, P. Eisert, and B. Girod, "Digital Watermarking of MPEG-4 Facial Animation Parameters", Computers & Graphics, Vol. 22, No. 4, pp. 425-435, August 1998.

[8] nVIDIA presentation at Summer Camp '99

[9] Shiyu Li, Masahiro Okuda, Shin-ichi Takahashi,"Embedded Key-Frame Extraction for CG Animation by Frame Decimation", IEEE International Conference on Multimedia and Expo, 2005

[10] http://www.cse.unr.edu/~mukesh