The Designation and Implementation for Human Skinned Mesh Animation Based on XNA

Zhang Shi-nan
Armored Force Institute of the Army, Bengbu 233050, China

Abstract. The technology of human skeletal mesh animation is one of the most important research focuses, and remains problematic in the researching field of computer animation. Firstly, this paper gives a comprehensive analysis on the theory and implementation of skeletal mesh animation, and then an engine framework was design and completed by using Visual Studio 2012 and XNA4.0 as development tools. In order to improve the executing efficiency and reality of animation, Effective measures such as optimizing rotating algorithm, animation blending and using GPU to update vertexs are used here.

1. Introduction
Computer animation technology has been widely used in film, game, military, aviation and other fields. Its research objects include cloud, fog, rain, rivers, various plants, animals and so on in real life or virtual world, but one of the most difficult and challenging objects is human’s animation[1]. Human’s movement is very complex, shape is irregular, and muscle changes with movement. Moreover, hair, skin and clothing simulation is also difficult. Human’s behavior usually includes subtle behaviors (such as expression changes), local behaviors (such as eye closure, gestures, etc.), and systemic behaviors (such as walking, jumping, running, etc.). Because humans are so familiar with their own movements, incongruent movements are easily detected by observers. Based on the analysis of the principle of skeleton skin animation, this paper realize the human skeleton skin animation by using XNA4.0, and adopts a variety of key technologies to optimize the animation.

2. The principle of skeleton animation
At present, real-time human animation technology can be divided into three categories: key frame animation, joint animation and skeleton animation.

The key frame animation needs to first store the mesh model of each frame animation, which is essentially an animation technology that exchanges space for time. It has the advantage of less computation, faster speed and more realistic characters. This is useful if there are hundreds of animation models in a scene that share the same animation. However, its limitations are also obvious. Firstly, it takes up more space, and secondly, it is difficult to interact with users due to its poor flexibility. Therefore, key frame animation is generally used for the simulation of simple continuous actions. Using a key frame animation model in XNA is easy because XNA has classes that deal with static models.

Joint animation is one of the modeling methods based on the surface model, which rigidly connects the Skeleton layer (left in FIG. 1) with the Skin layer (right in FIG. 1). In the animation process, the skeleton motion drives the corresponding skin vertex motion[2]. In this method, there is no movement of the skin relative to the bone connected with it. So only the movement of each bone relative to the parent bone is needed to be described using matrix, regardless of the elastic deformation of the skin.
This method is simple to realize, and the calculation speed is slower than the key frame animation. However, because of the rigid connection, it is easy to produce cracks at the junction of two bones, which will affect the effect. Therefore, joint animation is suitable for situations with low requirements for skin precision, such as simulating robot behavior.

Figure 1 bone structure diagram (left) and skin model diagram (right)

Skeletal skin animation can be regarded as a combination of key frame animation and joint animation, which is a popular animation technology at present. On the basis of joint animation, it uses Vertex Blend technology to change the data structure of skin layer vertices so that it can store the skeletal information and weight of influence on the position of these vertices, and then obtain the final position of vertices through "skin" algorithm. Using the "skin" algorithm is equivalent to the dynamic skin on the bone joint, effectively solving the crack problem. However, as the weight designation in the "skin" algorithm is done manually, and it is difficult to specify the weight in the larger joint parts (such as the shoulder), deformation problems such as "collapse" and "candy wrapper" effect are prone to occur. In literature [4], Xia Kaijian proposed an improved "skin" algorithm, which effectively improved the deformation problem by adding auxiliary nodes to the skin variable region. This paper will calculate the location of skin vertex according to this improved "skin" algorithm.

3. Skeleton skin animation designation base on XNA

3.1. The creation of animation model
The creation of skeletal skin animation is generally divided into model creation and animation creation. The realistic animation model needs to be realized by experienced modelers and animators through professional 3D software such as 3DMAX and Maya. Relevant data information needs to be stored in the model:

(1) model data
The static model consists of a skeleton layer and a skin layer surrounding it. The model needs to store relevant information for skeleton skin animation, mainly including skeleton hierarchy and skin vertex information. The hierarchy of bones refers to the formation of a hierarchy according to the physical characteristics of a character. Figure 2 shows the hierarchical structure of human skeleton. The arrow points from the parent skeleton to the child. From the child skeleton, matrix multiplication is used to accumulate to the top-level root skeleton, and the transformation matrix of each child skeleton relative to the world coordinate system can be obtained. Skin vertex information includes vertex index, vertex associated skeleton index and weight, vertex color, vertex normal vector, etc.
Figure 2 hierarchical structure of human skeleton

From a programmer's point of view, animation data is the skeletal keyframes and transformation matrices that determine the motion of the model. In XNA, the transformation matrix of animation can be stored inside the model, but for models with more complex animation, it is generally stored in a separate animation file, so as to facilitate model driving, human-computer interaction and animation mixing.

3.2. data loading

When using XNA for game programming, the animation model files created by 3D software (in X format or FBX format) must first be converted to XNB format before they can be read by XNA's Content Reader. In XNA, programmers use a Content Pipeline to load resources on the hard disk [5] (such as 3D models, 2D images, sounds, etc.). This is a two-step process: first, when the project is compiled, the content pipeline reads the resource from the hard disk and processes it into binaries and stores it on the hard disk; When the game is running, the data is read directly from the binary. The advantage of this is that it allows a lot of computation to be done before the game runs, making the game run faster. At the same time, the binary data stored in the first step can be read by PC, Xbox 360, Zune and Windows Phone, thus realizing cross-platform.

While the XNA content pipeline is convenient for resource loading, it does not fully support skeleton animation because it can import X files or FBX files with skeleton animation data, but does not handle all skeleton animation data when output. To add support for skeletal skin animation to XNA, the programmer needs to extend the default model processor to create a new process and store the skeletal and animation data for the model. Figure 3 shows the classes you need to create to extend the content pipeline, adding support for skeleton skin animation models, and adding classes that have been identified in gray fonts and black boxes. In addition, you need to create new classes to store, write, and read this data.

Figure 3 extended content pipeline (part)

3.3. bone skin animation update

In the running process of the game, the character needs to be ready to interact with the user at all times to complete the corresponding animation, which is also a necessary function of real-time animation technology. From a programmer's perspective, this means updating the status of a role in real time based on user input.

(1) update of bone information
First according to the animation playback time determine the animation at the time of two key frame interpolation, and then according to the timing of these two key frame interpolation, to determine the moment a piece of bone in the local coordinate system transformation matrix (translation and rotation), with the father finally bones in the world of coordinates transformation matrix multiplication, and got a piece of bone in the world coordinates transform information [6]. In the interpolation process, the translation process can be completed by ordinary linear interpolation, while the rotation process can be achieved by spherical linear interpolation of quaternions.

(2) update skin vertex information

For skin vertices, the "skin" algorithm is used to calculate the location of skin vertices based on the bone associated with each vertex and the corresponding weight. The "skin" algorithm is essentially an interpolation algorithm, the basic idea of which is to make the skin vertex at the joint affected by several adjacent bones, the size of which is determined by the weight. The formula is expressed as follows:

\[ v = \sum_{i=1}^{n} \omega_i M_i D_i^{-1} v_d \quad \text{among} \quad \sum_{i=1}^{n} \omega_i = 1 \]  (1)

Among them, \( v_d \) is the initial state of the skin vertex coordinates in the world coordinate system, said in the initial state of the first \( D_i \) bone from the local coordinate system to world coordinate system transformation matrix, \( D_i^{-1} v_d \) said skin vertices in the coordinates of the ith segment bone local coordinate system, \( M_i \) is the location of the ith segment bone in the current transformation matrix from local coordinate system to world coordinate system. \( \omega_i \) represents the weight of the ith bone for the current vertex, and \( v \) represents the coordinates of the deformed skin vertex in the world coordinate system.

3.4. optimization of skeletal skin animation

(1) optimization of rotation algorithm

Rotation is the most basic operation of skeleton animation, so the advantages and disadvantages of rotation algorithm are closely related to the final effect of animation. Traditional XNA programming typically USES euler angles to represent rotations, requiring three floating-point Numbers to represent rotations around three axes. In this paper, quaternions are used to represent rotation. Quaternions require four floating-point Numbers. Although using one more floating-point number than euler Angle leads to increased memory consumption, quaternions can avoid the jitter and universal joint lock defects caused by euler Angle. In addition, the quaternion can also interpolate the two key frames before and after by spherical linear interpolation according to the elapse Time and anim Length of the animation, so as to obtain the transformation matrix at that time. The spherical linear interpolation formula is as

\[ slerp(p, q, t) = \frac{p \sin(1-t)\theta + q \sin(t\theta)}{\sin\theta} \]  (2)

Where \( p \) and \( q \) are source and target quaternions, \( t \) is the interpolation parameter between 0 and 1, and \( \theta \) is the actual Angle between the two quaternions.

(2) animation mixing

Animation blending is the blending of two or more existing animation sequences to produce a new animation. For example, walking and waving are combined to form an animation of walking and waving [7]. While generating new animations, this technique also saves animation loading time and memory overhead of animation storage, and reduces the workload of animators.

In this paper, animation has been divided into two parts in the animation design process: upper body animation and lower body animation. In the actual movement of the character, different animation effects can be combined according to the interaction with the user or the environment. The upper body animation is mainly hand animation, including: walk, run, squat, wave, draw a gun, shoot,
death, etc. The lower body animation is mainly leg animation, including: walk, run, squat, death, etc. The most basic animation mixing is to combine the upper and lower body animation to generate a variety of new animations. The driving mode of this kind of hybrid animation is similar to that of the skeleton skin animation, but the transformation matrix of the upper and lower body skeleton should be calculated at the same time during the update. Another kind of animation mixture is the transition between two kinds of movements, such as the drawing and shooting of a gun or the death of a bullet in the running process, which need to be seamless between the various animations. In the implementation, you need to freeze the previous animation at the instant the animation changes, get the current keyframe, and then make the difference with the first keyframe of the next animation. To achieve a smooth transition.

(3) GPU programming updates vertices
Before the advent of programmable graphics processors, cpus did most of the calculations in skeletal animation. However, with the improvement of graphics hardware performance, especially the use of vertex renderers and pixel renderers in the GPU, as well as the current flow processor, the comprehensive computing capacity of the GPU has been greatly improved, and its unique vector and matrix computing capabilities are beyond the CPU [8]. Therefore, it can give full play to the computing characteristics of GPU and share the pressure of CPU in computing.

GPU is mainly designed for the characteristics of graphics rendering. There are pipeline such as vertex rendering, rasterization and pixel rendering inside the GPU, but there is no correlation or dependence between vertices and pixels. Therefore, GPU is suitable for large-scale parallel computing, but not for logical and complicated algorithm operations, and the results will not be saved for the next operation. Therefore, to leave some of the operations of skeletal animation to the GPU to perform, these operations must be kept simple and suitable for large amounts of data. By analyzing the operations involved in the skeleton skin animation in section 2.3, it can be found that: the updating of skeleton information involves interpolation operation, recursive operation and matrix operation, and the results need to be retained for the updating of vertex information, so the updating of skeleton information cannot be calculated by GPU. The update of vertex information is for each vertex, and there is no dependency between vertex and vertex, only the related skeleton and skeleton influence weight are used, and the global transformation matrix of skeleton is used to update the vertex data, and the updated result is not required to be kept but directly displayed on the screen. Therefore, this process is very suitable for processing in the GPU. In this way, the CPU can be used to update the skeleton information, and the GPU can update the vertex information, and the two can coordinate to complete the update of the skeleton animation.

4. Realization of skeleton skin animation based on XNA
In this paper, Visual Studio 2012+XNA4.0 was used to realize the skeleton skin animation. Based on the above analysis and design of the skeleton skin animation, the following will be implemented. The program structure design drawing is shown in figure 4. The program implementation is divided into four parts: model and animation creation, data loading, data readout and data update. In the data update part, the update of the skeleton information is carried out in the CPU. The transformation matrix of the skeleton in the world coordinate system and the correlation information between the vertices and the skeleton calculated in the CPU are passed into the GPU. In the GPU part, HLSL is used to update the vertices and output to the screen. Figure 5 is the main class diagram of the program: the Key Frame Sequence class is used to get the key frame of the animation sequence and perform interpolation operation; Game Animate Model class for matrix conversion, animation mixing, animation playback, etc. The Game Player class is used to interact with the user; The Game1 class is the main class of the program in XNA and is the entry point for the program to run.
5. conclusion
Based on the principle of skeleton skin animation, this paper designs and implements skeleton skin animation on XNA platform. In the process of implementation, the rotation algorithm of the animation is optimized; the animation blending technology is used to realize the seamless connection of each action of the characters; the vertex update and rendering are completed by GPU. These measures improve the efficiency of animation execution and enhance the realism of animation. With the development of 3D technology, skeletal skin animation technology has become one of the most popular technologies in the field of real-time animation because of its excellent execution efficiency and realistic animation effect [9]. How to achieve more realistic animation mixing, how to coordinate the utilization of CPU and GPU, will become the next research hotspot.
Reference

[1] AUBEL A. Anatomically-based human body deformations [D]. Université de Marne-la-Vallée, France, 2002.

[2] Wu Xiao, Ma Lizhuang, Gu Baojun. Research status and Prospect of human body modeling and skin deformation technology in computer animation [J]. Chinese Journal of image graphics, 2008, 12 (4): 65-73

[3] ZHOU X J, ZHAO Z X. The skin deformation of a 3D virtual human[J]. International Journal of Automation and Computing, 2009, 6(4): 344-50.

[4] Xia Kaijian, Wang Shitong. Simulation of skin deformation by improved bone skin algorithm [J]. Computer application and software, 2009, 26 (12): 174-176

[5] Alexandre Santos Lobão A. Beginning XNA 3.0 Game Programming From Novice to Professional [M]. Apress, 2009: 305-319.

[6] Method in Real-time Skeletal Character Animation[J]. International Journal of Virtual Reality, 2011, 10(3): 25.

[7] Chen Jun. data processing and skeleton animation implementation of complex 3D scene [D]. Zhejiang University, 2008

[8] Lei Tao, Luo Weiwei, Wang Yi. A method of virtual human animation generation with realistic effect [J]. Computer application research, 2012, 29 (10):