A Study of Blending Methods for Generating Multiple Skeletal Animations

멀티플 골격 애니메이션을 위한 블렌딩 방법의 연구

장동삼, 허기택
동신대학교, 동신대학교

Zhang Dongsen, Gi-Taek Hur
Dongshin University, Dongshin University

요약

*.x 확장자는 3D 모델개발자와 3D프로그래머들이 프로그램을 보다 쉽게 개발하기 위해서 주로 사용된다. 오늘날에는 3D프로그래밍을 위한 하드웨어들의 성능이 많이 개선되었지만, 아직도 효율적인 애니메이션 생성을위한 방법들에 관한 연구는 계속 진행되고 있다. 특히 3D 모델개발자들은 *.x 파일을 이용하여 골격애니메이션을 만들 때 한 부분의에러가 다른 부분으로 전이되어 찌그러지는 현상이 나타날 수 있기 때문에 에러를 수정하는데 많은 시간이 필요로 하는 문제점을 내포하고 있다. 그래서 이와 같은 문제점을 해결하기 위해서 본 논문에서는 여러 개의 골격애니메이션을 블랜딩 할 수 있는 방법을 제안하였다. 이를 위해서 3D Max, Maya와 같은 애니메이션 블랜딩을 위한 프로그램이나 *.x 파일을 코딩하여 사용하고 있고, 골격 애니메이션을 합성하는 특수효과를 3D Max8.0을 사용하여 *.x 파일로 구현하였다.

■ 중심어 : | *.x 파일 | 골격애니메이션 | 애니메이션 블렌딩 |

Abstract

We all know *.x file is a bridge of the 3D model developer and the game developer. In today's 3D game programming, more and more advanced hardware support it to run, but we still need to consider the methods to generate the animations. 3D model maker is tired of rectifying the skeletal animations. If he mistakes one point in some one animation, this will lead to distortion in *.x file. Then the modification consumes long time. So finding a good blending method is the best choice for generating multiple skeletal animations. There were some methods for animations blending. In this paper, we could use 3D max or Maya to blend animations; and we could use *.x file and blend animations in coding. And we will use 3D max 8.0 to export the *.x file and present a better way to combine skeletal animations.

■ keyword : | *.x file | skeletal animations | animations blending |

I. Introduction

One of the greatest, and most confusing, additions to Direct3D is that of skinned meshes. These miraculous mesh marvels allow the user to dynamically deform a mesh in order to produce animation. As you can tell from the title of this article, we are not here to talk about the subject of skinned meshes, but rather to study the methods of blending multiple skeletal animations on the base of understanding of the file format that you're most likely to use when dealing with skinned meshes: XFile(with the extension .x). To be more precise, we are only going to show you some blending methods for generating multiple skeletal animations through *.xfile. Now don't throw your hands up in the air asking what good this does you! In fact, understanding how to parse an *.x file template is the first big step you'll take to loading and using skinned meshes.

Though knowing the Microsoft *.x file structure is the important step, many books have introduced detailedly. You could read the site [1] and know it exactly.

II. Multiple Skeletal Animations Blending Methods

A Direct3D developer could create a simple geometric object easily through DirectX function library such as box、sphere、cylinder、teapot、polygon、torus, etc. If

you have attempted to construct your own 3D object by manually specifying the vertices, you will found it quite tedious. Thus we should find a best method to alleviate this tiresome task of constructing the data of 3D objects.

2.1 Modeler Animations Blending Methods

Article[2] dedicated to game programmers that explain the basic skeletal animation. A 3D model maker should be good at some a useful 3D application. Special applications called 3D modelers have been developed. These modelers allow the user to build complex and realistic meshes in a visual and interactive environment with a rich tool set, making the entire modeling process much easier. Examples of popular modelers used for 3D game development are 3DS Max、Maya、Light Wave 3D、Motion Builder and Motion Capture.

2.1.1 Base Manual Animations Blending Method

We know, the 3D model animations is the model skeletal animations. If we need some animations, he has to modify the key frame to the 3D model skeleton. After finishing the cockamamie modification, export the created mesh data (geometry coordinate, materials, animations, and other possible useful data) to a file. Thus, we could write a file reader to extract the mesh data and use it in our 3D applications. This is certainly a viable solution. The cockamamie work is just the base manual animations blending method.

There are so many tutorials to tell you how to blend the model animations by manual modification. Because of three-dimensional model, the 3D model maker will spend much time on modifying animations on key frame and the animations are not real-life. If he mistakes one mesh point, the file that exported will appears coordinate error. This method is not fit for large-scale 3D game development.

2.1.2 3D Model Makers Animations Blending Method

A different approach to real-time human body modeling, which is based on special software called Body Builder, is presented in [3]. 3D model maker is different from 3D model developer. 3D model developer should be good at more 3D applications than 3D model maker. Today's 3D applications are more and more advantaged for developing animated 3D model. Such as Autodesk Motion Builder、Motion Capture and so on. These applications are used for generating living animations.

In general, the advanced toolset consists of rapid skeleton creation for bipeds and quadrupeds, support for inverted joints for characters like birds and dogs, as well as animation mapping skeletons brought in from 3D programs that support the FBX format. Wide motion-capture support is well integrated and substantial target between complex characters is made easier and more automatic.

The Motion Builder solver is a hybrid forward and inverse kinematic system, so you can animate in both modes at the same time. If you are animating the hand with IK, the rest of the arm follows and tries to resolve the position, extending the end effector beyond the reach of the character which causes the arm to straighten as it tries to reach the goal. If you pull the effector beyond the character's reach with the body solver on, the entire biped skeleton will adjust the spine, hips and legs as well. This gives a lot of control for rapidly setting up performances because there is no need to manually adjust each body part.

A Motion Capture records are only the movements of the actor, not his visual appearance. These movements are recorded as animation data which are mapped to a 3D model (human, giant robot, etc.) created by a computer artist, to move the model the same way. Data output of high performance real-time 3D motion capture is incredibly clean and foot data is very accurate. The export file is in *.bvh format allowing easy integration into all industry standard software including motion builder, Maya and 3DS Max.

Motion Capture is expensive equipment the better one could capture motion for more than 30 minutes, so it's generally used for 3D movie motion capture. Motion Builder is popular for 3D model developers. Actually, the 3D Tool Animations Blending Method is same to the Base Manual Animations Blending Method, they all generate the animations *.x file that the 3D game developer need. But the 3D Tool Animations Blending Method is just to use the equipments to blend the animations. It's a better method for us to save time.

Although the method served our purpose quite nicely, its generated animation sequences did lack uniqueness. Once an animation generated, the animation are always the same, regardless of how many times you play them.

2.2 Coding Animations Blending Methods

These methods are the work of the 3D game developers in coding.

2.2.1 Behavior Simulation Blending Method

This animations blending method belong to the advanced mathematics scope. It bases on the motion dynamics, takes the model mesh vertex as the orientation points and carries out the virtual animations. It's different from the other skeletal blending methods. Actually, it's a skin animations blending method. It tells us a theory that the skin animations are instead of skeletal animations. The method is presented in [5]. It's hard for us to be in charge of the method, because we should know the virtual humanoids motion dynamics better first.

2.2.2 Skinned Mesh Matrices Transform Animations Blending Method

The skeleton is a hierarchical set of bones. Each bone has a matrix which translates and rotates that bone relative to its parent (this is just the same process as the frame hierarchy discussed above but the mesh is normally separate to the hierarchy). Since the matrix is relative to the parent rather than the character itself we say they are defined in bone space. We want to place the bones relative to the character itself so must convert these matrices into character space by combining them with their parent matrix. This method is presented better in [6]. Through reading it, you could know mesh matrices transform define and transform process deeply.

III. Mix Blending Method for Generating Multiple Skeletal Animations

Our project is base on the skinned mesh matrices transform animations blending method. But we have improved the mesh matrices transform method. First of all, we scan the list of bones still in the skeletal structure of the *.x file. For each bone, we are going to track the combined transformations from the appropriate key frames. For each key frame found in the animation, we add the transformation to the skeletal structure's transformations.

3.1 Mesh Matrices Transform

We add a variable to our extended mesh container structure for this combined matrix(called exFrame CombinedMatrixPointer - one entry per bone).Since we already maintain a combined matrix for each frame in the hierarchy (exCombinedTransformationMatrix), we do not need to duplicate it therefore our exFrameCombined MatrixPointeris an array of pointers to the correct frames' exCombinedTransformationMatrix. Finding the correct frame is carried out during SetupBoneMatrices using the Direct3D D3DXFrameFind function.

Combined Bone Matrix = Local Bone Matrix * Parent Combined Bone Matrix

There is one further matrix required when doing skinning and that is a matrix to connect the bones to the mesh, this is known as a bone offset matrix. We add this to our structure as exBoneOffsets. These are assigned during the CMeshHierarchy::CreateMeshContainer function - one per bone. This allows us to create our final matrix:

Final Bone Matrix = Bone Offset Matrix * Combined Bone Matrix

This final matrix transforms the mesh into bone space then applies the bone matrix to the mesh. This is calculated during the frame move function and fed into the UpdateSkinnedMesh function.

3.2 Main Coding with the Object

There are a number of methods of carrying out skinning with Direct3D. The demo code uses software skinning but there are other methods that use hardware acceleration for greater speed. The Skinned Mesh sample that comes with the DirectX SDK describes three other methods: Fixed Function Non Indexed Skinning, Fixed

Function Indexed Skinning and Shade Based Skinning.

The software skinning method used by the demo is carried out in FrameMove. The process is:

```
for (UINT i = 0; i < Bones; ++i)
D3DXMatrixMultiply(&m_boneMatrices[i],&pMesh->exBoneOff
sets[i],pMesh->exFrameCombinedMatrixPointer[i]);
void *srcPtr;
pMesh->MeshData.pMesh->LockVertexBuffer(D3DLOCK_R
EADONLY, (void**)&srcPtr);
void *destPtr;
pMesh->exSkinMesh->LockVertexBuffer(0, (void**)&destPtr);
// Update the skinned mesh
pMesh->pSkinInfo->UpdateSkinnedMesh(m_boneMatrices,
NULL, srcPtr, destPtr);
// Unlock the meshes vertex
bufferspMesh->exSkinMesh->UnlockVertexBuffer();
pMesh->MeshData.pMesh->UnlockVertexBuffer();
```

We lock the vertex buffer of our original unaltered mesh for read. We also lock the vertex buffer of the destination mesh vertices to write into (this is why we stored a copy of the mesh in our D3DXMESHCONTAIN-

ER_EXTENDED structure. It was copied during SetupBoneMatrices). We then call the Direct3D UpdateSk-

innedMesh function which takes the transformation matrices and our locked vertex buffers and carries out the skinning. It does this by combining the matrix and a set of bone weights to alter the position of the original vertices and write them into the output buffer.

Locking vertex buffers is to be avoided as much as possible if we want our graphic card to work at maximum speed. A lock can cause a stall in the graphics processing leading to slow down. To avoid this there are a number of methods that use the hardware to carry out the skinning itself. If you are interested I suggest you look at the skinned mesh sample that comes with the DirectX SDK.

3.3 Art Creation

The creation actor.x file contains a number of animations (Figure 1).

▶▶ Figure 1. All the animations of the model is contained in the left combox

As already mentioned, if the *.x file has a hierarchical set of bones, we can carry out a new animation simply by altering a frame matrix. It's worth viewing the following figure (Figure 2) while understanding the blending method.

▶▶ Figure 2. the blending result with the selected animations

IV. Conclusion

The animations blending method is useful for us to model real-time virtual humanoids animations and be applied in the 3D game programming. The modeler animations blending methods are just fit for 3D model developers to study. The coding animations blending methods are best fit for 3D game developers. The animations blending method that we proposed improve the animations blending speed. The animations could be smoothly and easily included in the simple and short-time multiple skeletal animation system based on the animations blending methods. Especially, the skinned mesh matrices transform animations blending method is a good method of choice for generating real-time character modeling multiple skeletal animations. But it exist some scarcity still. Two or more different local bone animations couldn't be blending at the same time. Future

work, we will find another way to solve the problem and improve to blend the flowing multiple skeletal animations included in high-level animation systems.

▌참 고 문 헌▐

[1] http://www.xmission.com/~legalize/book/download/index.html
[2] Lander, J., "Game Programming for the Web Generation", Game Developer Magazine, pp. 11-16, May 1998
[3] Kalra, P., Magnenat Thalmann, N., Moccozet, L., Sannier, G., Aubel, A., Thalmann, D., "Real-time Animation of Realistic Virtual Humans". IEEE Computer Graphics and Applications,Vol.18, No.5, pp.42-55, 1998
[4] http://en.wikipedia.org/wiki/Rotoscope
[5] Kingsley Stephens, Binh Pham, Aster Wardhani, "Modelling Fish Behaviour", Queensland University of Technology
[6] Jim Adams, "Advanced Animation with DirectX", a division of Thomson Learning, pp.91-106, 2003