

УДК 681.515.8

Zdolbitska N.V., Zdolbitsky A.P., Chmil O.M.  
Lutsk national technical university

## LIBRARY DEVELOPMENT FOR CHARACTER REAL-TIME ANIMATION

**Zdolbitska N.V., Zdolbitsky A.P., Chmil O.M. Library development for character real-time animation.** Increasing quantity and quality of games and other applications made popular using the detailed characters. The article describes skeleton and library for character that has realistic looking and skeleton structure. Most of the elements that are included to library are found in modern character animation libraries. To make unique characteristic of the character additionally was made new methods of animation and control. Moreover, animation technique of importing one model format to another is described.

**Key words:** skeleton, library, animation, effect, rigid bones, model.

**Figures 5, Bibliography 11**

**Здолбіцька Н.В., Здолбіцький А.П., Чміль О.М. Розробка бібліотеки реального часу для анімації персонажа.** Збільшення кількості та якості ігор та інших програм популяризувало використання деталізованих персонажів. Стаття описує скелет та бібліотеку, що створені для персонажу, який має реалістичний вигляд та скелетну структуру. Більшість елементів бібліотеки знайдено у сучасних анімаційних бібліотеках. Додатково були розроблені нові методи контролю та анімації для додавання унікальних рис персонажу. Також описано техніку імпортування одного формату моделі в інший.

**Ключові слова:** скелет, бібліотека, анімація, ефект, жорсткі кістки, модель.

**Здолбицкая Н.В., Здолбицкий А.П., Чмиль О.М. Разработка библиотеки реального времени для анимации персонажа.** Увеличение количества и качества игр и других программ популяризировало использование детализированных персонажей. Статья описывает скелет и библиотеку, созданных для персонажа, который имеет реалистичный вид и скелетную структуру. Большинство элементов библиотеки найдено в современных анимационных библиотеках. Дополнительно были разработаны новые методы контроля и анимации для добавления уникальных черт персонажа. Также описана техника импортирования одного формата модели в другой.

**Ключевые слова:** скелет, библиотека, анимация, эффект, жесткие кости, модель.

**Research Analysis/ previous works.** There are open and commercial systems that implement some form of character animation. Open systems include few libraries that have functions close to the developed one. Cal3d is only one stand-alone library and Nebula Device 2 and Irrlich are most popular through game developers.

Cal3d is an open source character animation library that is written in C++. It is skeletal based 3D character animation library that built in platform-graphic API-independent way. The library does not belong to a certain engine. Character basic concepts of animation are included in this library, but advanced features are not available. It uses an original file format and runs others format that are converted.

The Irricht engine is fast and supports skeleton and mesh based animation. There is no possibility to use advanced effects in it. The main focus is on character features, but not animation.

Advanced graphic engine Nebula Device 2 has good animation system and a high performance. It supports most character animation features. To perform skinning it uses palette-skinning vertex shader.

The main focus of new systems is on skeleton animation. Unfortunately, they do not put attention to mesh animation. Flexible library need to have methods that can control and extend skeletal animation, these include inverse kinematics and rigid bone attachment. Exactly this direction is chosen to make Skeleton Library.

**Development of the Skeleton Library.** Developed library includes features that have to be in the modern animation library, there are also some improvements in it. Furthermore this library is independent, so it can be added to any engine. It includes the following features:

- Model's control skeleton animation
- Support of existing and new formats
- Weighted skinning
- Seamless animation blending
- Advanced function controlled effects for any part of the model
- Skeleton fitting gives a possibility to use animation from other model formats
- Possibility to add library in any project

**Approach to design.** Library design is originated from BVH format. This format lacks a mesh structure. But library that has been developed took it to account. This is resulted in library including structures for a skeleton, mesh, keyframe animation and blender. Here the example of the listing:

```

HIERARCHY
ROOT Hips
{
  OFFSET -0.347901 96.7718 1.79791
  CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
  JOINT LeftUpLeg
  {
    OFFSET 8.91 -6.27 -2.145
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT LeftUpLegRoll
    {
      OFFSET 0 -22.7323 0
      CHANNELS 3 Zrotation Xrotation Yrotation
    }
  }
  ...
}
    
```

Supporting of new formats is gained by parsers that are integrated to developed library. So adding a new format became really easy. Popular feature extensions like inverse kinematics and rigid bone attachment can be done through using the function controlled effects.

The BVH is the format that stores motion data. It is modeling format that has hierarchical structure of joints and effectors.

**Library features**

*Model* contains skeleton structure and mesh that defines the character's look and poses. It includes a number of animations and effects based on function. They determine movements that character is capable of. Another important thing is that model must have skeleton structure; the reason for it mesh animation is not supported. The model structure is depicted in Figure 1. All improvements are presented in Figure 2.

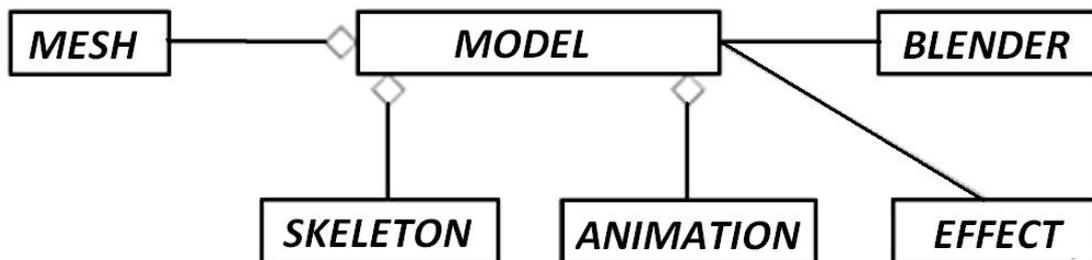


Fig. 1 – Model structure

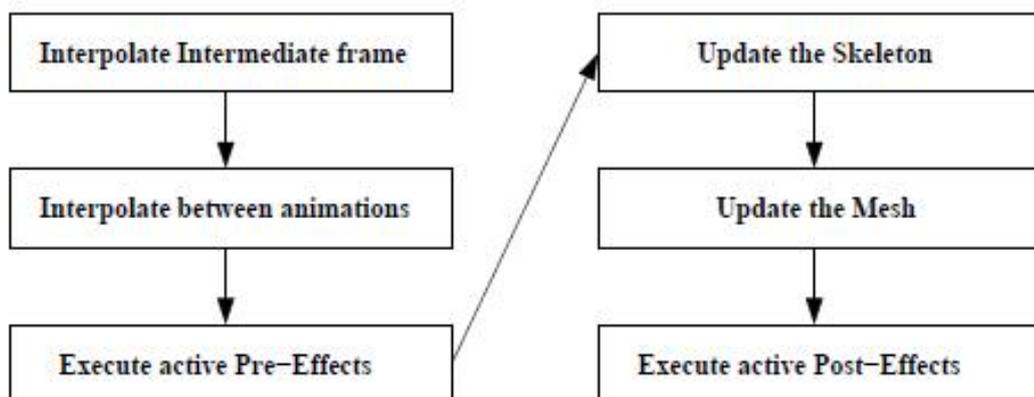


Fig. 2 – Module improvements

*Skeleton structure* consists of a number of joints, each one connected with parent joints to lengths offset. Also joints have a rotation that is used for character movements. In addition the end effectors are used. An end effector is a rigid extension from the joint which cannot rotate. This gives an opportunity to use external tools to the body. The basic relative rotation to its parent is kept by the end effectors. The main root joint is the base of structure. Its offset is pointing the translation of the model. Next Figure describes the structure of the skeleton.

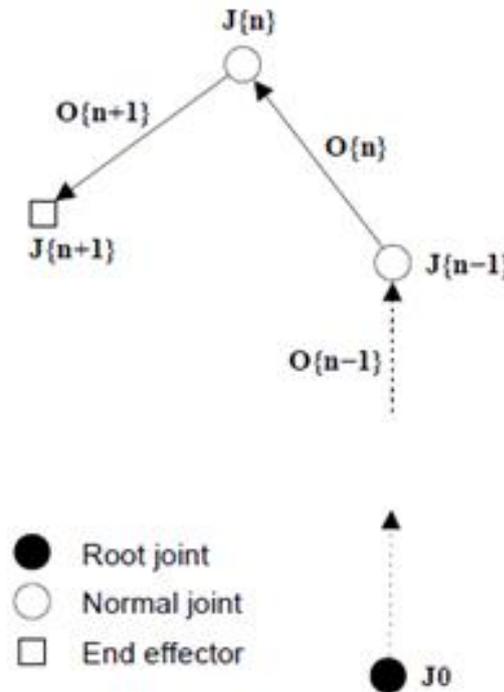


Fig. 3 – Skeleton structure

**Mesh.** The mesh possesses all main properties that include vertex normals, material properties, faces and texture coordinates. Weighted vertices are also supported. Each vertex connects to several bones and that is used to make the transitions between submeshes. There can be any number of weights per vertex, but in some cases more than four weights are demanded. The mesh is divided to submeshes.

**Animation.** Here is used keyframe animation that means an animation includes few keyframes each consist of one rotation matrix and one offset vector for one bone. Every animation has a timer that helps to track current keyframe position. The animation also can be looped that means that character repeats one movement. There is also a possibility to apply many animations at the same time (Blender can be used for it).

**Blender and Progressive Skeleton Fitting.** Blender takes care of two main steps in the animation process: interpolation between keyframes of one animation and interpolation of all active animations by interpolating these together to one picture. To do so Blender uses quaternions and use following to spherical interpolation. Quaternions are weighted by user. It need to be remembered that only whole model can be blended.

OSL\_PARSER is performing skeleton fitting. It takes two models, source and target, and joint map to define which joints have to be matched. Thanks to this function fitted animation can be added. Nowadays, each set of model format demands separate map.

**Effects are controlled by functions.** All the effects implemented in subclass OSL\_Effect. Inside the subclass there is a function execute (OSL\_Model\*) that adds all updates of the model. This function is integrated with a model and has access to all private members. Pre-effect and Post-effect can be added to the model. They are made in the same way, but using Post-Effects demands extra care. The reason is that the Skeleton and Mesh are already done and all updates should be done correctly.

The effects will typically execute every update cycle of the model either prior to update or as a post-effect. The use of the post effects has the advantage in calculating the positions for multiple models only needing to do one update run through the entire model and adding the individual effects afterwards, skipping a lot of unnecessary recalculations. Of course it must be taken into consideration that changing a bone in a skeleton structure after its update means that a recalculation of its children must take place. The Pre-Effects are simply easier to create since no concern for specific update issues must be addressed.

In the case of Post-Effects, care must be taken to make sure that everything is updated correctly. For instance if the Effect influences a bone, it must make the effect of this alternation propagate to its children (if this is desired).

Additional improvements:

- Inverse Kinematics
- Animation bits - blending control
- Level-on-detail
- Vertex shader driven skinning
- Integrated physics

**Skeleton Fitting.** The following example illustrates the use of skeleton fitted animations. The application shows the typical usage where a very good animation for a skeleton with few bones exists but the animation is to be used on an advanced model with a detailed skeleton with a mesh. In this demo, the MD5 model fatty from Doom 3 is chosen as target model for animation. The source animation is taken from the the BVH model format, used for motion capture data. Target and source model in this scenario consists of 80 and 19 joints, respectively. The two models can be seen in Figure 4.

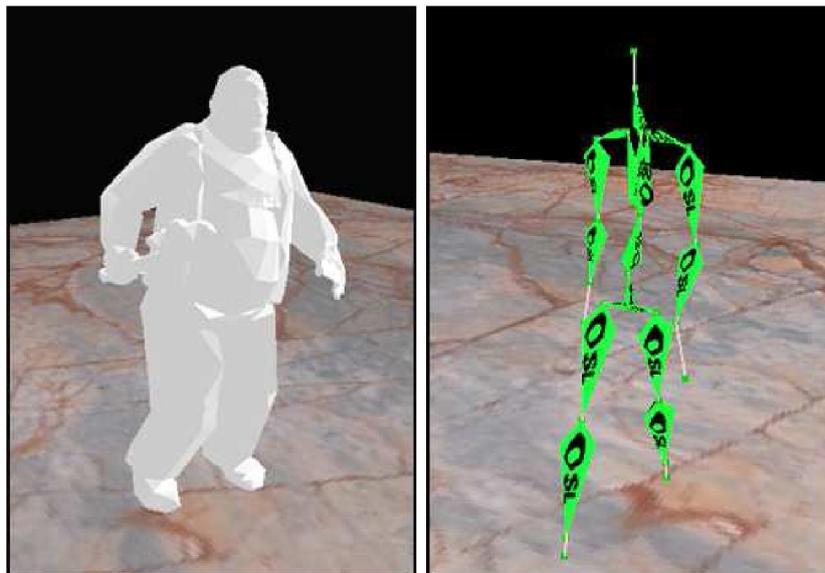


Fig. 4 – Target and source model

The process of transforming the animation from the source to the target starts with loading the different models with their corresponding models:

```
OSL_Parser_MD5 md5_p;  
OSL_Parser_BVH bvh_p;  
OSL_Model md5Model = md5_p.parseModel("zfat/ zfat.md5mesh");  
OSL_Model bvhModel = bvh_p.parseModel("bvh/ ballet.bvh");
```

To match the bones correctly, a map must be required. It contains an index at every position pointing to the matching joint index of the source model. If no matching joint is available, the default value will be -1.

```
std::vector<int> map;  
for (int m = 0; m<md5.getSkeleton() ->getNbrOfBones(); m++)  
{  
std::string name =  
md5model . getSkeleton() ->getBone (m) ->getName());
```

```
int bvhbone;  
if (name == "Body") bvhbone = BVH_HIPS;  
else if (name == "Lupleg") bvhbone = BVH_LEFTHIP;  
else if (name == "Rupleg") bvhbone = BVH_RIGHTHIP  
.br/>.br/>.br/>else bvhbone = -1;  
}  
map.pus_back(bvhbone);
```

This map is then used to apply the animation to the target model.

```
OSL_Parser::fitAnimation(md5model, bvhModel, 0, map);
```

This line of code adds the animation starting at index 0 from the bvhModel to md5model. The fitting of the bindpose is made by matching the target's bindpose to the source's bindpose which can be seen in Figure 5, from left to right, the default bindpose of target, the source's bindpose and finally the target's fitted bindpose. The fitting is now complete and the animation is loaded and usable by the md5model.

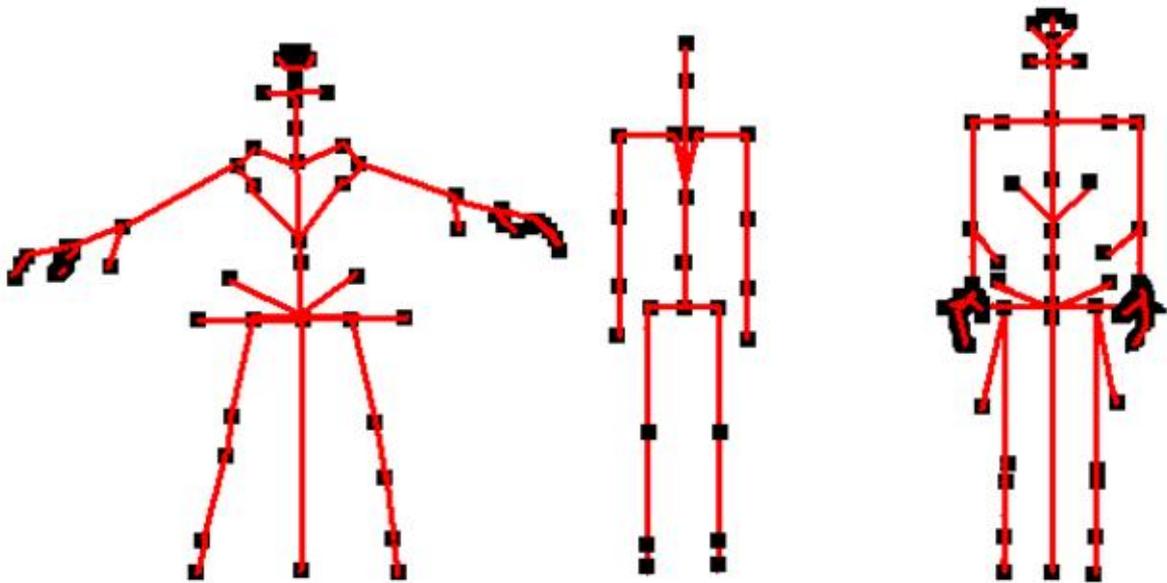


Fig. 5 – The steps of skeleton fitting

**Rigid Bone Attachment.** To illustrate how effective the function controlled effects are, they have been used to implement rigid bone attachments for OSL. The attachment is implemented through a Post-Effect, taking an OSL\_Model as the attachment, and connecting it to a designated bone. It is then added to a target model where its execute function replaces the attached model's root offset and rotation found at the target's designated bone. Since it is a Post-Effect it will execute after the target model has performed its updates, thus, putting it in the right place. The trick is to communicate with the model update through the use of its matrix and vector stores.

**Conclusions and Results.** The Open Library has a sound design capable of including many features. All implemented animation techniques in the library are smoothly integrated with each other and can be used in a smooth way. The performance of the library can be improved by implementing vertex shader driven skinning.

The new features, skeleton fitting and function controlled effects, were successfully implemented. Through the Effects the same model can be used repeatedly with individual characteristics, preventing the appearance of the cloning effect often seen in games. The Effects can also be used to easily add new arbitrary methods of control. The skeleton fitting is an excellent tool for applying animations from one model to another. This is especially useful when motion capture data, which is obviously done for a human skeleton, should be applied to a non-human model.

Although not complete the Open Skeleton library has the potential to be one of the leading open source character animation libraries. A clear design and extensible structure makes it ideal as a modular extension in, for instance, a game engine.

1. Здолбіцька Н.В., Здолбіцький А.П., Яручик В.Л. Крокуюча система на Bluetooth управлінні // Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" – Луцьк: Видавництво ЛНТУ. – Вип. 16-17. – 2014. – С. 27–31.
2. Здолбіцька Н.В., Савлук А.В. Моделювання поведінки живих організмів за допомогою аніматів на базі роботизованих платформ // Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" – Луцьк: Видавництво ЛНТУ. – Вип. 12. – 2013. – С. 27-30.
3. Шолом П.С., Здолбіцька Н.В. Пропорційно-інтегрально-диференціальне управління стабільністю руху мобільного агента Зрі по заданій траєкторії / П.С. Шолом, Н.В. Здолбіцька // Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" – Луцьк: Видавництво ЛНТУ. – Вип. 8. – 2012. – С. 125–129.
4. Bio Vision Incorporated <http://www.biovision.com>
5. Blundell, Barry G. & Schwarz, Adam J. "Creative 3-D display and interaction interfaces: a transdisciplinary approach". Hoboken. Wiley-Interscience, 2006
6. Cheng, Irene... [et al.], (eds.). "3D online multimedia & games: processing, transmission and visualization". Hackensack, N.J. World Scientific Publishing, 2008
7. Mikio L. Braun. Accurate error bounds for the eigenvalues of the kernel matrix. J. Mach. Learn. Res., 7:2303–2328, 2006.
8. Pablo de Heras Циєчомські Parametric Dynamics - A method for addition of dynamic motion to computer animated human characters Lund University of Technology 2001.
9. Pauli, David B., Programming Dynamic Character Animation, 2002 Charles River Media, inc, pages 164-165.
10. Sebastien Domine Mesh Skinning nVidia 2005.
11. Watt, Alan and Policarpo Fabio, 3D Games - Animation and Advanced Real-time Rendering Vol. 2, 2003 Pearson Educational Lint, pages 358-362.