Hierarchical Evolution for Animating 3D Virtual Creatures

Tailin Chen, Daosheng Mu, and Tainchi Lu*

Dept. of Computer Science and Information Engineering

*National Chiayi University

No. 300, University Rd., Chiayi, Taiwan 600, R.O.C.

*E-mail: tclu@mail.ncyu.edu.tw

Fax: +886-5-2717741

Abstract

In this paper, we present an efficient approach to performing motion retargetting on a new creature's skeleton that is generated by taking advantage of hierarchical evaluation. A meta-skeleton is created by means of crossing over two source skeletons. We use the mutation and survival principles to reproduce the next-generation skeleton to accomplish motion transitions in a seamless and smooth manner. In addition to the skeleton construction of a new creature by using the hierarchical evaluation, motion analysis is a required process to preserve the original motion attributes of the source creatures. Spacetime constraints, degree of freedom, and principal component analysis are taken into account in the proposed method in order to prevent unexpected artifacts. The experimental results show that the proposed method enables motion behaviors of the target creature to be shown in between the two source creatures. As a result, the proposed technique can effectively produce a new virtual creature's skeleton with plausible motion data, and is applicable to working for 3D digital games and computer animations.

Keywords: 3D Computer Animation, Motion Retargetting, Motion Editing, Creature Skeleton, Evolutionary algorithms

1. Introduction

With the high growth of modern entertainment, three-dimensional computer graphics technologies play an important role in digital games and computer animations in recent years. People are addicted to enjoy realistic 3D computer animations, such as Shrek produced by DreamWorks; therefore, the 3D usages in computer animations have been more and more popular than before. Conventional 2D image animations make use of many pictures to continuously display at a time. In contrast, 3D graphics animations take advantage of keyframes and interpolations to display image frames in order. In the field of 3D computer animations, a creature's motion clip can be regarded as a series of frames animating by using continuous interpolations in between keyframes. It is not very easy to directly control the body's action of a creature's object without any support of rigid structure; this is because the surface is simply composed of hundreds of thousands of triangular polygons. As a result, a creature skeleton is taken into account to facilitate to the creature's motions and actions. Previous researches make use of joints and bones, and take advantage of keyframes and interpolations to animate creature's motions [1, 2]. The advantages of the above implicit approaches [3, 4] are easy to use without taking time-consuming geometric calculations. Nowadays, most game companies and academic agencies purchase motion capture systems for capturing creature's real motions. About the human's articulated motions, we can capture many realistic continuous postures of humans, including dancing, running, jumping, and so on. In contrast to humans' motions [5], it is obviously that we could not easily manipulate animals' motions by using any motion capture systems. Nevertheless, different motion capture systems have diverse skeleton structures in distinct file formats. An art designer cannot merely assign the existing motion data to another different skeleton of a creature in case of different number of joints or bones between two skeletons. On the other hand, expense and time required to capture new motion data are considerably high; hence, we desire to reuse these motions as much as possible for a new creature, especially a virtual creature (Figure 1). Motion retargetting is a well-known technique to edit the existing motions to gain the desired results on another target creature.

The proposed technique aims to produce a new virtual creature by means of employing hierarchical evaluation to cross over two different animal creatures, such as a horse and a pteranodon, as shown in Figure 2. After crossing over the

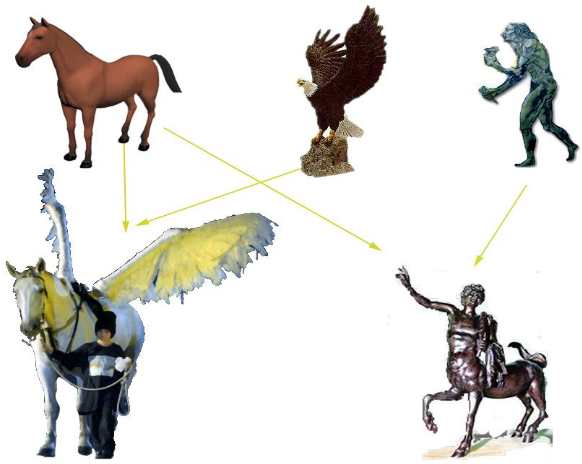


Figure 1: Diverse virtual creatures existed in a fantasy world.

source creatures, a meta skeleton is constructed according to its parents' skeletons, and it comprises joints and bones of its parents. We define a fitness evaluation function for mutating the meta skeleton and construct a next-generation skeleton, namely a target skeleton. Finally, we align the target skeleton to the parents' skeletons and accomplish motion transition [6, 7] in order to animate the new virtual creature.

2. Related Works

In the past few years, people are used to accept high diversities of 3D animated creatures because of the rapid growth of computer games and animations. We give the Shrek 3 that was produced by the DreamWorks as an example. The art designers have created two virtual animal creatures, namely donkey and dragon, and assigned flexible creature motions to them. However, it is a time-consuming and high cost task to produce a new 3D animated creature by art designers. Therefore, reuses of creatures' motions after performing some specific operations are required in the field of computer animations. We describe the technology of motion retargeting and the flow of hierarchical revolution in the following subsections.

We look forward to viewing more and more diverse creatures with realistic motions nowadays. However, it is a time-consuming and high-cost job to capture new creature's motions by means of using a motion capture system. In this paper, we devise a novel approach to combining motion retargetting technology [8, 9, 10] and genetic programming for the sake of motion reusability. The genetic programming is described in the following.

In 1992, Koza [11] proposed genetic programming that is similar to genetic algorithm. Genetic algorithm can be classified as a kind of artificial evolution, and the common purpose is to find out an optimal solution for a high complex problem. Its operation is similar to a Darwinian "survival of the fittest" approach. Therefore, the finer coming generations can survive after many evolutions. When we start to employ

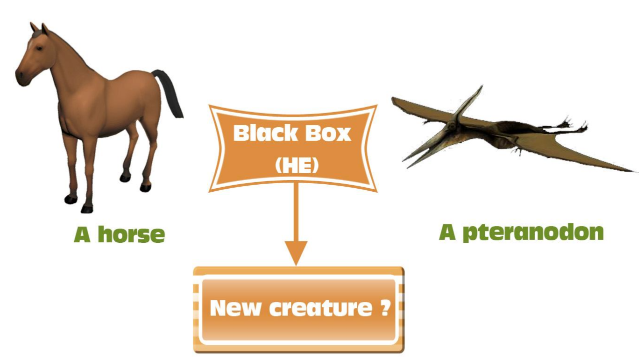


Figure 2: A new virtual creature generated by employing the hierarchical evaluation.

genetic algorithm, we need to define a problem's complexity in advance. In contrast, we do not require an understanding of the procedures or the complexity of a problem as long as we adopt genetic programming to perform a given computational task. In 1995, L. Gritz and J. K. Hahn apply genetic programming to motion control for articulated figures [12, 13]. They present an approach to articulated figure motion in which motion tasks are defined in terms of goals and ratings.

The purpose of employing genetic algorithm is to efficiently find out an optimal solution and obtain a more stable result. Its operation is similar to the natural selection phrase; that is survival of the fittest. We expect to see an adaptive generation after performing the survival competition for several times. Figure 3 shows the evolution process of performing the hierarchical evaluation in our proposed technique. By means of employing selection, crossover, mutation, and survivor operations, we can pick up more than one creature to be source objects, and make good use of the hierarchical evaluation to construct a new skeleton as the target object. Note that no final solution can be obtained if we neglect any one step of the evolution process shown in Figure 3.

3. An Approach to Generating Virtual Creatures

In this section, we propose an approach to generating virtual creatures by means of motion retargeting and hierarchical evaluation.

3.1 Steps for Accomplishing Motion Retargetting

A skeleton C^s is regarded as a collection of joints, that is $C^s = (R, j^1, ..., j^{i-1}, j^i, ... j^{n-1}), i \in \{1, ..., n-1\}$, where n is the number of joints, j^0 is referred to as the root node R with the representation $R = (t_x, t_y, t_z, r_x, r_y, r_z)$, and $j^i = (r_x, r_y, r_z)$. $(t_x, t_y, t_z) \in \mathbb{R}^3$ and $(r_x, r_y, r_z) \in S^3$ represent translation and rotation vectors, respectively, corresponding to their user-defined axes. Each joint has different degrees of freedom (DOFs), for example, $DOFs(R) = 6$ and $DOFs(j^i) = 3$.

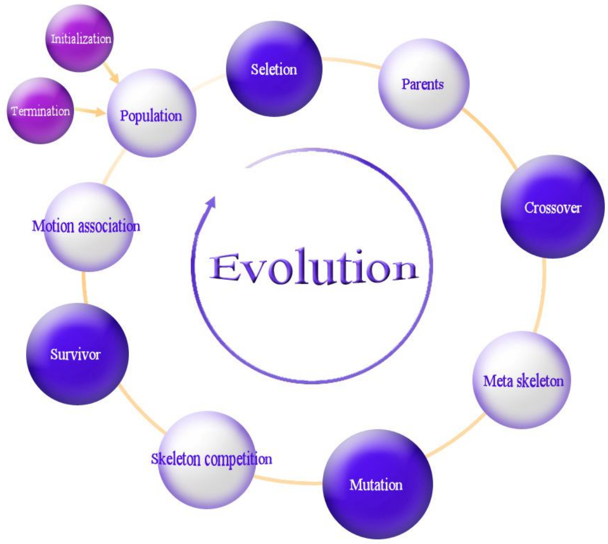


Figure 3: The steps of applying the hierarchical evaluation.

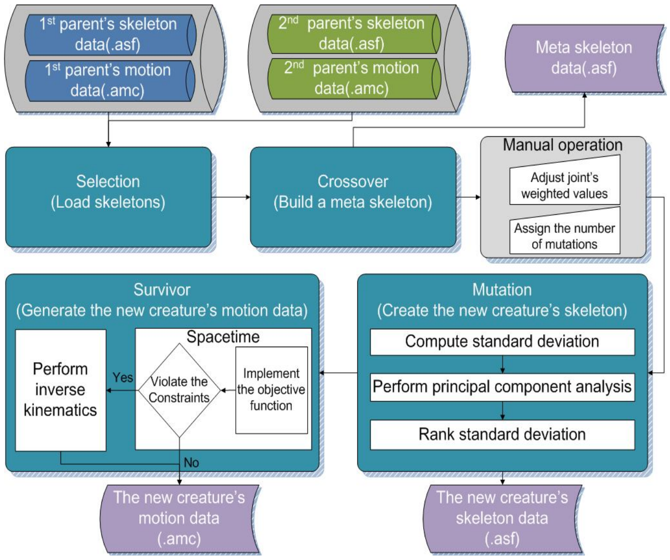


Figure 4: The flowchart of performing the hierarchical evaluation.

With respect to motion data $C^m(t)$, it is defined as $C^m(t) = (R(t), j^1(t), ..., j^{i-1}(t), j^i(t), ..., j^{n-1}(t)),\ t \in \{0, ..., k\}$. Each $C^m(t)$ is represented according to the translation and rotation in a geometric coordination at frame t. Suppose that there are two distinct virtual creatures C_1 and C_2 with different skeleton C_1^s and C_2^s, their skeleton structures may not be completely identical; meanwhile, we are not able to directly transfer one motion to the other one via motion retargetting. Therefore, the first job is to find the correspondence between the skeleton C_1^s and C_2^s, and then we need to initialize the skeletons. Finally, motion transition is carried out to accomplish the motion retargetting.

3.2 Steps for Performing Hierarchical Evolution

In this section, we propose the hierarchical evaluation for constructing a new creature's skeleton with motions. In Figure 4, two source creatures with their skeleton data and motion data are selected by users. After selection, crossover, mutation, and survivor operations, a new target creature has been constructed by means of the hierarchical evaluation. The four operations are described in detail in the following.

3.2.1 Selection

In the selection stage, we need to specify which kinds of animal creatures to be treated as the source objects. The constraint is to use vertebrate animals that their articulated figures meet with the specified constraint's hierarchy, that is $(Root, LowerSpine, UpperSpine, LowerNeck, UpperNeck, Head)$, as shown in Figure 5 (a). With the aid of the constraint's hierarchy, it is easily to carry out the correspondences of joints and bones. The selections of animal creatures are manual in our system. Users can arbitrarily pick up two of the interested vertebrate animals to be the source objects. For example, we select a horse as the first source, and its skeleton is C_1^s that is shown is Figure 5 (b). Figure 5 (c) shows another animal creature, a dolphin, by loading its skeleton C_2^s into the system. Loading two skeletons C_1^s and C_2^s into the system is the first step to carry out the hierarchical evaluation.

3.2.2 Crossover

As shown in Figure 6, different animal creatures have dissimilar articulated spines. We should first find out the difference between these two skeletons by comparing their spine structures of the articulated figures. For example, we analyze the spine hierarchies of a horse and a dolphin according to their spine hierarchical structure, and look for the different and identical segments of the skeletons, as shown in Figure 6 (a). The same identifiers of the segments are retained, and to split the different parts for further operations.

We combine with the same identifiers of segments and append the other different segments to the new spine hierarchy. In other words, the new structure of the skeleton is the meta skeleton in terms of using crossover procedure. After crossing over a horse and a dolphin, we show the spine hierarchy of meta skeleton in Figure 6 (b) and load the meta skeleton of the incomplete new creature into the system that is shown in Figure 6 (c).

3.2.3 Mutation

With the operation of crossing over two animal creatures, a meta skeleton is produced. Its articulated figure includes all joints and bones of source and target objects. The purpose of using mutation is to remove specific redundant joint segments from the meta skeleton. For instant, we can reject the horse's

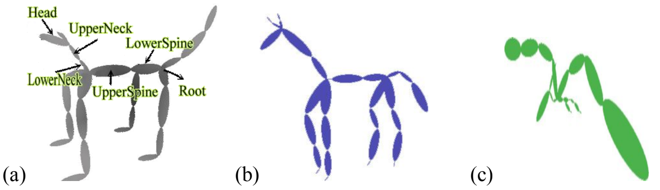


Figure 5: (a) Hierarchy C^{s_b} (b) Loading the skeleton C_1^s (c) Loading the skeleton C_2^s

limbs and tail, but keep the dolphin's main body. Subsequently, we only take into account the joints that are included within the meta skeleton to perform motion transition from the sources to the target. The mutation is an important process for the proposed technique, and its function is to generate an appropriate next-generation skeleton for transiting motion data to it. In addition, we take advantage of degree of freedom (DOF) to analyze the motion data of the source creatures.

$(dirx^i, diry^i, dirz^i)$ is the normalized direction of the joint *i*

$length^i$: the length of the joint *i*

$$x^i = length_i \cdot dirx_i$$
$$y^i = length_i \cdot diry_i$$
$$z^i = length_i \cdot dirz_i$$
$$(x^i(t), y^i(t), z^i(t)) = j^i(t) \cdot (x^i, y^i, z^i)$$
$$X^i(t) = \sqrt{x^i(t)^2 + y^i(t)^2 + z^i(t)^2}$$

$$Cos\theta^i(t) = \frac{X^i(t) \cdot X^i(t-1)}{|X^i(t)| \times |X^i(t-1)|}, \; t = \text{frame\_no., } i = \text{joint\_no., } -1 \le Cos\theta^i \le 1$$
$$\theta^i(t) = Cos^{-1}\frac{X^i(t) \cdot X^i(t-1)}{|X^i(t)| \times |X^i(t-1)|}, \; t = \text{frame\_no., } i = \text{joint\_no., } \theta^i\text{: Rotation angle of the joint } i$$
$$Radian^i(t) = \theta^i(t) \cdot 180 / \pi \tag{1}$$

In Equation (1), each angle θ that is formed by any two vectors of joints, X_i and X_{i-1}, is collected to perform motion analysis. We take Equation (2) that is the standard deviation function to calculate the motion variations of these selected joints.

$$\mu^i = \frac{1}{n}\sum_{i=0}^{n} Radian^i(t), \; t = \text{frame\_no. , } i = \text{joint\_no.}$$
$$\sigma^i = \sqrt{\frac{\sum_{t=0}^{n}(Radian^i(t) - \mu^i)^2}{n}}, \; t = \text{frame\_no. , } i = \text{joint\_no.}$$
$$\sigma^i = w^i \cdot (\sigma^i + 1), \text{ if } w^i \text{ is assigned} \tag{2}$$

θ: Degree of a joint, μ: Average of degrees in a frame interval, n: Total numbers of degrees in a frame interval, w: Weighted value, σ: Variance of a joint

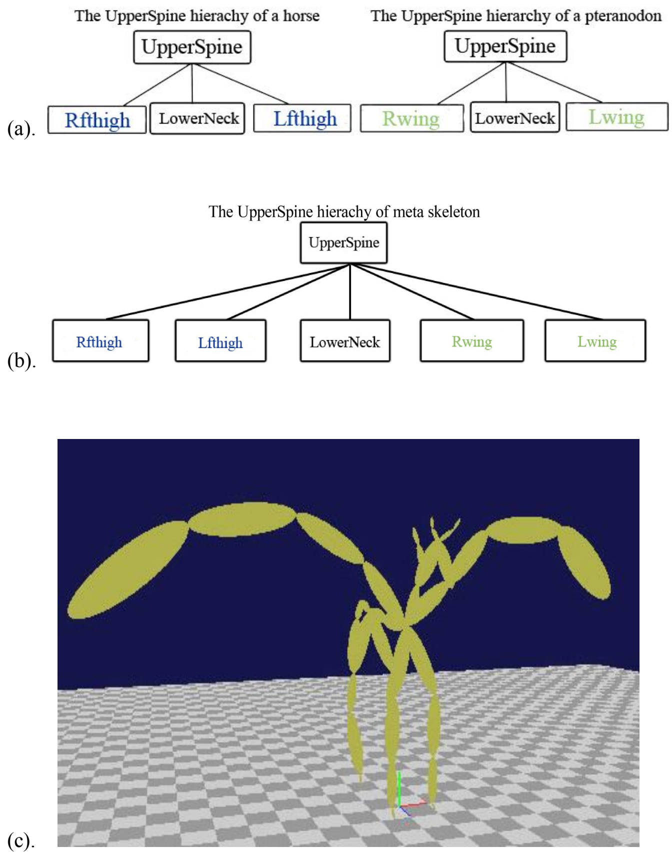


Figure 6: (a) The upper spine hierarchies of the source characters. (b) The upper spine hierarchy of the meta skeleton. (c) The figure of the meta skeleton.

3.2.4 Survivor

The purpose of carrying out survival process is to transfer the motion data of the parent creatures to the target skeleton that has been accomplished the mutation process. We need to analyze two kinds of motion data for synthesizing another new motion data that can be transferred to the target skeleton. In 1997, Gleither presented a method for editing a pre-existing motion and proposed a spacetime constraints solver to consider the entire motion [14]. C_p^m is a target population creature with new synthesizing motions. There are two source creatures with existing motions, C_1^m and C_2^m. We refer to the object function that was proposed by Gleither in 1997 for facilitating the motion blending towards C_p^m. In Equation (3), the parameter t is a specific frame number of a motion animation and x is an user-defined parameter that is used for adjusting the motion variance among C_p^m, C_1^m and C_2^m. Figure 7 shows the result of carrying out motion transition from two source creatures to the new population creature C_p^m. In other words, the motion data of C_p^m comprises the motion data extracted from C_1^m and C_2^m.

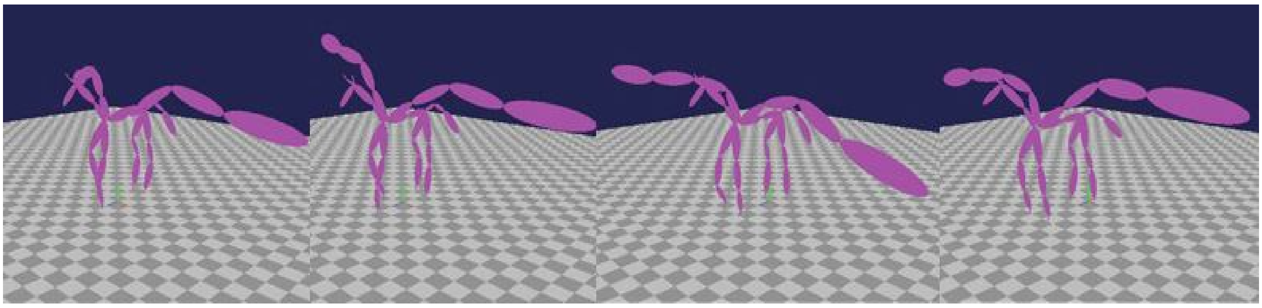


Figure 7: The computed motion for the new creature.

$$\int_t d(t)^2 = \int_t (C_2^m(t) - C_1^m(t))^2$$
$$C_p^m(t) = C_1^m(t) + x \cdot d(t)$$
t : frame no., x : adjustment parameter ,

C_p^m : mutative motion, C_1^m : the 1st motion data

(3)

4. Experimental Results

The major difference between the proposed hierarchical evolution and the other conventional techniques is interactive control while performing the evolution processes. The conventional techniques automatically operated until an optimal solution has been found out or exactly satisfied with their objective functions. For example, a user desires to generate a flying creature with wings.

The conventional techniques need to search for an optimal solution of how to grow specific wing joints that they are able to fly very well; however, it is a time-consuming task and cannot satisfy with the specified requirements of most art designers. In comparison, the proposed method not only supports automatic skeleton constructions but also provides interactive control manner to facilitate to accomplish the hierarchical evolution. Moreover, we make use of Fast Light Toolkit (FLTK) [15] to design our graphical user interface. As shown in Figure 8, a horse, a pterodactyl, and a human being are served as the testing examples for experimenting on the proposed technique. Table 1 lists the number of joints and frame length with respect to these three source creatures. We create a new creature with motions by our system, and the result is shown in Figure 10 and Figure 11.

In this subsection, we experiment on the horse, the pterodactyl, and the new creature to obtain their trajectories of rotation joints. In essence, the motion data of a new creature must be similar to the parents' behaviors, but act some specific differences between the father and the mother. Subsequently, we have to find out minimal motion differences by employing the spacetime function. In other words, the motion data of a new creature can be acquired by referring to the motions of the first creature; meanwhile, we also find out the corresponding motions from the second creature in comparison with the first creature's motions. As a result, the motions of the new creature are analogous to the behaviors of the first creature. From Figure 9 (a) to Figure 9 (d), we can see that the trajectory of the target creature (i.e. the red line) is close to the trajectories of the first creature (i.e. the blue line) as much as possible, but not exactly the same as the blue line. This is because we find out the minimal difference in motion data between the first and the second creatures for synthesizing the motion data of the target creature. The behavior of the target creature is similar to the blending of the first and second creatures.

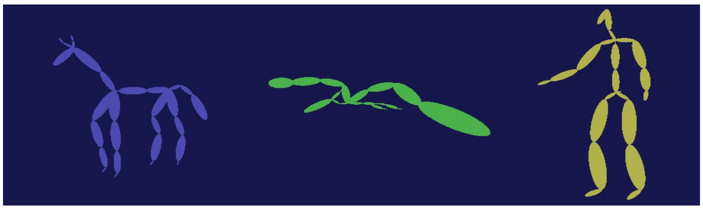
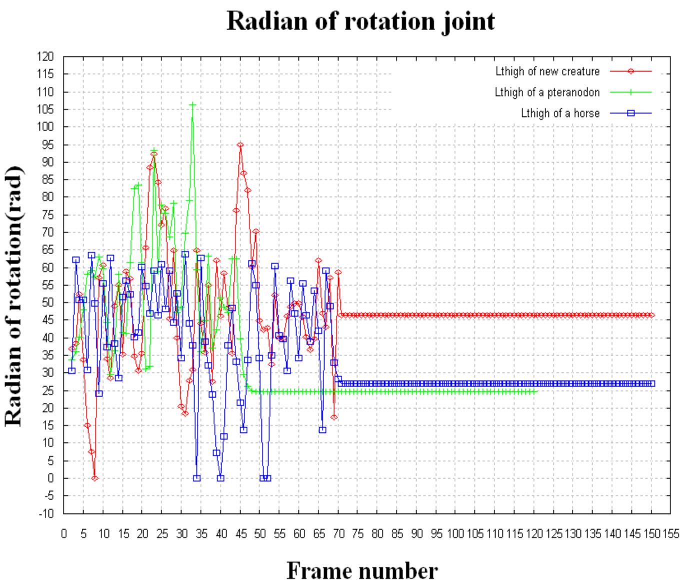


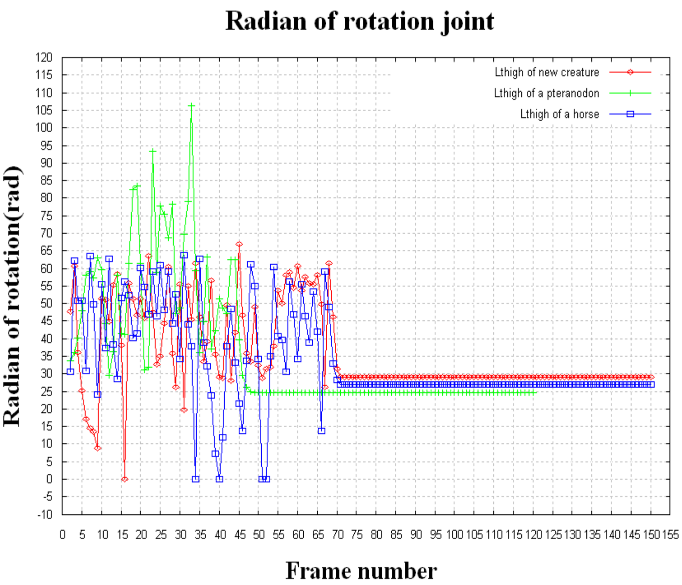
Figure 8: Articulated figures of horse, pterodactyl, and human.

Table 1: The number of joints and frame length with respect to the skeletons of horse, pterodactyl, and human.

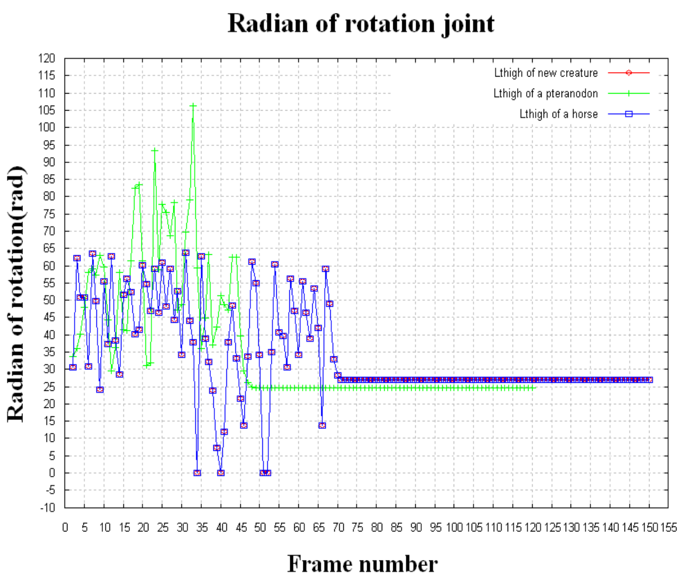
	Number of joints	Frame length
Horse	48	151
Pterodactyl	36	121
Human	31	111



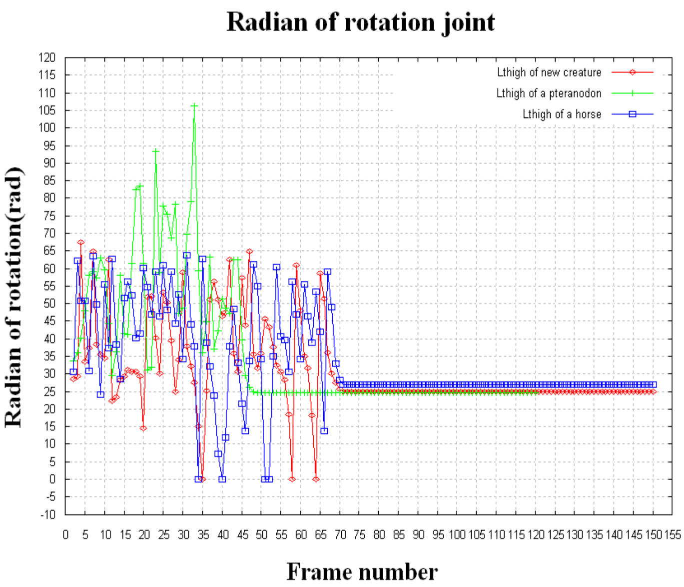
(a) Radian of rotation joint in left thigh when x = -2.0.



(b) Radian of rotation joint in left thigh when x = -0.5.



(c) Radian of rotation joint in left thigh when x = 0.0.



(d) Radian of rotation joint in left thigh when x = 0.5.

Figure 9: Radian of different rotation joints by using various weighted values.

5. Conclusion and Future Works

The paper has proposed a novel approach to accomplishing motion retargetting towards a new virtual skeleton-based creature that is generated by using the hierarchical evaluation. By means of employing selection, crossover, mutation, and survival operations, we can pick up more than one creature to serve as source objects, and make good use of the hierarchical evaluation to construct a new skeleton as the target object. In addition, we carry out motion analysis, including DOF and PCA, to preserve the original motion attributes of the source objects in order to make the new creature's motion more fantastic and with fewer unexpected artifacts. Finally, we apply the motion transition from the source objects to the target object in a very smooth manner. The proposed approach is applicable to most computer game applications when game developers or artists desire to reuse the existing motion data on 3D virtual

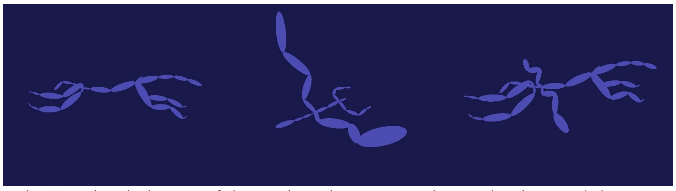


Fig 10: The skeletons of the testing characters, a leopard, a bat, and the new creature.

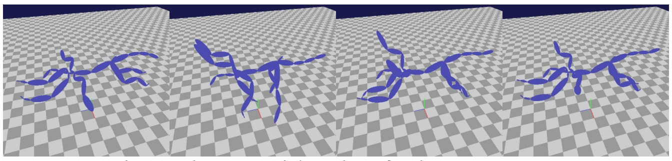


Fig 11: The sequential motions for the new creature.

creatures. In the future work, we will employ dynamic time warping technique to facilitate to the operation of motion blending instead of using the slicing window method.

Acknowledgement

This work is supported in part by the National Science Council at Taiwan, R.O.C., under the project grant number NSC97-2221-E-415-012.

References

[1] Pamela K. Levangie and Cynthia C. Norkin, *Joint Structure and Function: A Comprehensive Analysis*, 4th Edition, ISBN: 0803611919, 2000.

[2] Michael Girard and A. A. Maciejewski, "Computational modeling for the computer animation of legged figures," Proceedings of the 12th annual conference on computer graphics and interactive techniques, pp. 263-270, 1985.

[3] Daniel Sanchez-Crespo Dalmau and Daniel Sanchez-Crespom, *Core Techniques and Algorithms in Game Programming*, New Riders Publish, 2003.

[4] David H. Eberly, *3D Game Engine Design*, Academic Press, 2001.

[5] Jehee Lee and Sung Yong Shin, "A hierarchical approach to interactive motion editing for human-like figures," Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 39–48, 1999.

[6] Michael Gleicher, Hyun Joon Shin, Lucas Kovar, and Andrew Jepsen, "Snap-together motion: Assembling run-time animations," ACM Transactions on Graphics, Vol. 22, Issue 3, pp. 702-702, 2003.

[7] Golam Ashraf and Kok Cheong Wong, "Generating consistent motion transition via decoupled framespace interpolation," Computer Graphics Forum (Proceedings of Eurographics 2000 Conference), pp. 447–456, 2000.

[8] Kwang-Jin Choi and Hyeong-Seok Ko, "On-line motion retargetting," The Journal of Visualization and Computer Animation, Vol. 11, No. 5, pp. 223-235, 2000.

[9] Michael Gleicher, "Retargetting motion to new creatures," Proceedings of the 25th annual conference on computer graphics and interactive techniques, pp. 33-42, 1998.

[10] Alexander Savenko and Gordon Clapworthy, "Using motion analysis techniques for motion retargetting," Proceedings of Information Visualization, IEEE Computer Society Press, pp. 110-115, 2002.

[11] John R. Koza, "Genetic programming", MIT Press, 1992.

[12] L. Gritz, J.K. Hahn, "Genetic programming for articulated figure motion", The Journal of Visualization and Computer Animation, Vol. 6, Issue 3 , pp. 129-142, 1995.

[13] L. Gritz, J.K. Hahn, "Genetic Programming evolution of controllers for 3-D creature animation", Genetic Programming 1997: Proceedings of the 2nd Annual Conference, pp. 139-146, 1997.

[14] Michael Gleicher, "Motion editing with spacetime constraint," Proceedings of the 1997 symposium on interactive 3D graphics, pp. 139-ff, 1997.

[15] FLTK, available through http://www.fltk.org/, July, 2007.