Extended Spatial Keyframing for Complex Character Animation

Byungkuk Choi, Mi You, Junyong Noh

Visual Media Lab, Graduate School of Culture Technology - KAIST

335, Gwahangno, Yuseong-gu

Daejeon 305-701, Republic of Korea

Tel. (+82)42 869 2114 Fax. (+82)42 869 2210

email: {litlpoet, anubodhih, junyongnoh}@kaist.ac.kr

Abstract

As 3D computer animation becomes more accessible to novice users, it makes it possible for these users to create high-quality animations. This paper introduces a more powerful system to create highly articulated character animations with an intuitive setup then the previous research, Spatial Keyframing. As the main purpose of Spatial Keyframing was the rapid generation of primitive animation over quality animation, we propose Extended Spatial Keyframing that exploits a global control structure coupled with multiple sets of spatial keyframes, and hierarchical relationship between controllers. The generated structure can be flexibly embedded into the given rigged character, and the system enables the given character to be animated delicately by user performance. During the performance, the movement of the highest ranking controllers across the control hierarchy is recorded in layered style to increase the level of detail for final motions.

Keywords: Performance-driven Animation, Character Articulation, Keyframing

Introduction

Spatial Keyframing(SK) [1] offers a compelling approach for the interactive control of 3D characters, as user movements directly control the timing of animated motions blended with a set of markers. It is a robust tool for the interactive manipulation of 3D characters with the sparsely distributed markers associated with target poses. As the markers store the information of the target poses and are placed with the character in the same 3D space, the markers are known as spatial keyframes.

SK is a revolutionary idea that breaks the convention of existing keyframing. However, the produced motions are too simple to be used for practical character animation. A practical tool should be versatile enough to produce highly articulated complex motions while allowing easy creation of desired animation.

This paper introduces a method for creating high-quality complex character animations. We call this Extended Spatial Keyframing(ESK). ESK provides new features that give great flexibility by embedding a global control structure into a given 3D rigged character with multiple sets of spatial keyframes. A hierarchical relationship between controllers and a layered recording are also used to strengthen the ability of ESK regarding user performance. As a result, novice users can achieve a very flexible structure ready to be animated through an intuitive setup process.

ESK starts with designing multiple sets of spatial keyframes on desirable parts of a given rigged character with hierarchical relationships between the relevant sets. The relationship can be easily organized by creating a new set using the controllers of the existing sets created up to this point. The entire system becomes a complex structure for the given character as the number of desirable articulation parts increases. The hierarchical relationship, however, appropriately reduces the number of controllers by allowing the highest rank controller to move its subordinate controllers automatically.

While designing multiple sets, a global control structure of ESK is automatically constructed in the given rigged character. The complete system manages the entire control inputs such as a global and a local transformation. Once the sets are fully defined, a series of user actions that represents a target animation is recorded in multiple layers. Here, the user performs only with the highest ranking controllers.

As detailed in Section , the quantitative error measurements and the user test clearly indicate that participants with no experience can create a highly articulated character animation within their first hour of using the system, including training time.

The key contribution of this paper lies in how novice users can create highly articulated character animations with minimal user interactions. Low-quality animation for SK is overcome by utilizing a global control structure with multiple sets of spatial keyframes. To deal with a complex structure resulting from multiple sets, a hierarchical relationship and its control method are introduced. Multiple sets of spatial keyframes are recorded layer by layer to add details to the animation as it is created.

Related Work

Making 3D computer animation is now more within the reach of novice users. Several studies related to novice users have been conducted in every category of computer animation. Teddy, a sketching interface for 3D freeform design [2] is an intuitive modeling tool for novice users. The novice user can create 3D polygonal surfaces merely by sketching several 2D freeform strokes. Pinocchio [3] can be used to prepare characters for animation, as it is an automatic rigging system that implants a skeletal-based structure into the character. These approaches focus on simplicity to reduce the complicated procedural tasks of the animation pipeline. Hence, novice users can create animations with little effort. Spatial Keyframing [1] has the same goal of the efficient creation of character animations. It created a character motion with a predefined pose set that required simple user input. The method is highly intuitive but limited to simple character motions.

Performance-driven animations are another approaches targeting for rapid generation of character animation. A user performance is the input to the system to drive the motion of a target object. The sketching-based systems proposed by Popović et al. [4] and Thorne et al. [5] generate a motion path of a target model. They both employ the concept of 'sketching' with a simple interface to match the trajectory of the user input with the target. Terra and Metoyer [6, 7] utilized a model of user performance to adjust the timing of previously keyframed animations. Although the method is valuable considering that mapping the timing is intricate for novice users, it does not provide any means to manipulate spatial data for desirable poses.

A hierarchical approach is widely utilized to handle the growing complexity of control problems. Computer graphics is not an exception. For modeling, Forsey and Bartels [8] introduced a hierarchical B-spline refinement to enhance the surface modeling capability. For control, Liu et al. [9] proposed a hierarchical space-time control to solve space-time constraints efficiently by reformulating functions through the time of the generalized degrees of freedom in a hierarchical wavelet representation. For animation, Lee and Shin [10] adopted a hierarchical approach to edit motions of human-like figures interactively. These methods show the efficiency of the hierarchical approach in obtaining quick results as well as procedural controls for given complex tasks. Hierarchical approaches were also exploited in other research such as the motion control of intelligent agents [11] by Bruderlin et al., in a method known as a vertically structured multilevel abstraction hierarchy.

A layered approach has also been utilized to add details to final motions. Dontcheva et al. [12] introduced layered acting for the first time by piling consecutive user actions into a well-organized character animation. They used a motion capture system and what they characterized as a specific widget to capture the motion of an animator. The layered acting has a great potentiality to support an inexpensive performance-driven system. Neff et al. [13] introduced a novel method of correlation maps with layered inputs via user performance. They also used simple input devices such as a mouse and a keyboard to control complex character motions.

Extended Spatial Keyframing

ESK is built on top of SK. Although they share similar architectural traits, the structure of ESK is much more sophisticated in three ways. The differences include: (1) the automatic embedding of a global control structure into the given rigged character by designing multiple sets of user-defined spatial keyframes, (2) the capability for the generation of hierarchical relationships between controllers, and (3) the recording of animation in multiple layers according to user performance.

Constructing a global control structure with designing multiple sets of spatial keyframes

First, the user imports a 3D rigged character into the system. The user then selects the manipulators of the rigged model followed by a controller generation for the positioning and controlling of new markers. The manipulator is a handle responsible for articulating a character. An example would be an IK handle, a joint, a locator, or a type of a constrained NURBS curve clustered with several parts in Autodesk Maya [14]. The controller represents the intended position of a marker corresponding to a current pose. Once the user is satisfied with the current pose and the position of the controller, the marker registered with the current pose is generated by the simple click of a button. After designing a single set of spatial keyframes with all of the desirable poses with their corresponding markers, the single set begins by holding one controller with several markers and starts to create the blended motions with the set of markers.

Generation of the next set of spatial keyframes is straightforward. However, every generated set should be automatically belonged to the given rigged character and then can construct a global structure that supports both a global and a local transformation. As shown in Fig. 1, by parenting a generated controller to the higher rank controller of the selected manipulators, ESK can find the manipulator related to the global transformation. The root manipulator of the character always has a global transformation in the scene so that only one controller is dealing with global values while others are taking local values on the basis of the root value. In Fig. 1, it is controller 4 that covers a global transformation of the character.

As the goal is to control a complex character model, it is convenient when wanting to control a character part by part. As shown in Fig. 2, to create the crawling motion of an iguana, a rigged character can be divided into the three parts of the spine, the left leg, and the right leg. The movements can then be articulated individually. The same process is repeated for multiple sets of spatial keyframes. The system takes the xyz-coordinates of the controller and a set of spatial keyframes (the xyz-coordinates of markers and associated character poses) as input and returns a blended character pose. SK uses a radial basis function to blend a new pose [1]. ESK uses the same method for interpolation. The interpolation using radial basis functions is briefly introduced in the following, and a detailed explanation can be found in [15, 16]. First, the interpolation function has the form:

$$f(\mathbf{x}_i) = \sum_{j=1}^{k} w_j \phi(\mathbf{x}_i - \mathbf{m}_j) \tag{1}$$

where w_j denotes the weights, $\mathbf{x}_i$ the input vector, $\mathbf{m}_j$ the position of the markers in one set, and k the number of markers, or poses. For a smooth interpolation, Hardy multi-quadrics is employed as the basis function [17].

$$\phi(\mathbf{x}) = \sqrt{\|\mathbf{x}_i\|^2 + d_j^2} \tag{2}$$

$$d_j = \min_{i \neq j} \|\mathbf{x}_i - \mathbf{m}_j\|^2 \tag{3}$$

The distance d_j is measured between $\mathbf{m}_j$ and the nearest $\mathbf{x}_i$. We chose Hardy multi-quadrics after several experiments in Section . The interpolation is smooth, leading to smaller deformations for widely scattered feature points and larger deformations for closely located points [18]. Furthermore, it handles occasional extrapolations well.

The system solves for weights w_j given $f(\mathbf{x}_i)$ which represents the poses at the marker locations. The computed weights w_j are used to interpolate the final position of the character manipulators. Each entry of the selected manipulators requires the construction of a corresponding RBF interpolation system. While SK only allows for the root translation, which is sufficient for a simple motion of a skeletal-based character, ESK allows both translation and rotation of all the manipulators. The user has six degrees of freedom for each manipulator to control the part of the character. This enables the user to manipulate various poses in detail.

Hierarchical relationship and control

The user can organize the hierarchical relationships with more than two sets of spatial keyframes. Selected low ranking sets form a new high ranking set. The low ranking controllers become subject to a high rank and follow the user control of the highest ranking controller. In other words, the performance of the highest rank controller automatically interpolates the position and rotation of the lower-ranked controllers. As a result, the user carries out a simple manipulation of the highest ranking controller for the entire hierarchy.

To create a hierarchical relationship, the user selects low ranking controllers, defines new poses for a higher set, and simply clicks a button to generate a new controller. We employ RBF again for a parent-children relationship.

$$f^{parent}(\mathbf{x}_i) = \sum_{j=1}^{k} w_j^{parent} \phi(\mathbf{x}_i - \mathbf{m}_j^{parent}) \tag{4}$$

$$where, \mathbf{x}_i = \mathbf{c}_i^{child} (i = 1, 2, 3, \ldots, k)$$

Here, $f^{parent}(\mathbf{x}_i)$ represents the defined positions of low ranking controllers at the marker locations of a parent set. The linear system is solved only for the translation of each entry, as the controller of the low ranking set is only manipulated by translation.

This type of set is termed a *synchronizing* set, as its main role is to synchronize more than two individual motions into one and to make subordinate controllers move simultaneously. For example, in the multiple sets of spatial keyframes of the iguana character shown in Section , a synchronizing set for the leg motion can be created by synchronizing parts of the left legs and the right legs (see Fig. 3).

The hierarchical relationship allows easy articulation of the complex motion of included sets by controlling the highest ranking controller. The function for the motion corresponding to the user input can be written as follows:

$$F^{motion}(\mathbf{x}) = (F^{individual} \circ F^{synchronizing} \circ F^{control})(\mathbf{x}) \tag{5}$$

The input $\mathbf{x}$ is the position of the controller as set by the user, and the final result $F^{motion}(\mathbf{x})$ is determined by a set of interpolation functions cascaded together. $F^{control}$ depends on the user performance. $F^{synchronizing}$ and $F^{individual}$ represent the synchronizing and individual control set, respectively.

The strength of the hierarchical control is its efficiency in creating animation. While the input control is a very simple graph containing sequential vector values that represents the x, y, and z positions of the highest ranking controller along the time t, the generated graph of the articulated manipulators is much more complex. Complex graphs of the articulated manipulators also imply that the user should generate them in great detail for a complex motion via traditional keyframing.

Recording animation in multiple layers

The hierarchical control system is not sufficient to express highly articulated motions that deviate from the style imposed by the highest ranking controller. Instead, independent control of various parts of the character would provide great flexibility. The user has access to various controllers across the hierarchy for individual motion. The user begins by selecting one of the available controllers and acts out the motion by moving it across the markers. The translation of the controlled motion is recorded simultaneously in the form of a sketched path with temporal samples. The subsequent user actions with other controllers are recorded layer by layer.

Recording desirable motions is the last step. Complex motions can be generated by layering user trials in ESK. A similar method described in earlier studies [12, 13] is adopted here. As the user acts subsequently with the highest ranking controllers across the hierarchy, the separate motions of the character are recorded continuously in multiple layers. One advantage of ESK compared to one method [12] is related to the number of control parts that must be recorded as layers. As the predefined sets for the desired motions were previously created by this point using spatial keyframes, the user only needs to act with the highest ranking controllers in the sets, as shown in Fig. 4.

The overall motion is combined with multiple results from different layers. The final animation from the articulated controllers can be formulated as follows:

$$y'(t) = y(t) + (\mathbf{x}_j(t) - \mathbf{x}_j(t - dt)) \tag{6}$$

$$y(t) = \sum_{i=1}^{j-1} y_i(t) \tag{7}$$

Here, $y'(t)$ denotes the sum of every jth user performance of the highest ranking controllers. In other words, the current motion $\mathbf{x}_j(t) - \mathbf{x}_j(t - dt)$ with the translation of the controller at time t added to the existing motion $y(t)$ creates the layered animation $y'(t)$. The existing animation $y(t)$ is composed of $(j - 1)$ layers.

While $y'(t)$ represents the layered information of the articulated controllers, the final animation can be computed with the following equation:

$$Let\ \mathbf{Y} = (y'_x, y'_y, y'_z),$$

$$F^{final\ motion}(\mathbf{Y}) = (F^{individual} \circ F^{synchronizing} \circ F^{control})(\mathbf{Y}) \tag{8}$$

Here, $F^{final\ motion}(\mathbf{Y})$ is the result of the motion inherited from a hierarchical set, and $\mathbf{Y}$ is the sum of all animated layers by the articulated controllers.

Quantitative Analysis

Comparison with a temporal keyframing

A conventional approach of keyframing would require tedious manual effort as well as an artistic sense of timing. The goal here is to provide novice users with a mechanism for the creation of a highly articulated complex motion. For ESK to be a viable alternative, we ensured that it can produce the same range of animation created by conventional approaches with less effort and a guarantee of animation quality.

Two animations, natural and exaggerated running motions of a man, were produced. Table 1 shows production data in detail. While temporal keyframing requires the same effort for the natural and the exaggerated motion, ESK entails a fraction of additional time for the same extension. This implies that a great number of extended motions can be generated very efficiently using ESK, especially when the desirable motions of a character are repeatedly reproduced with small variations.

To test the interpolation performance, 25 frames of original animation were initially created by a skilled animator using temporal keyframing. The animation shows a running sequence of a cartoon-style man, and each pose is referenced from a running sequence in a well-known animation book [19]. This was then imitated using ESK. The first row in Fig. 5(Natural motion-original) shows first seven frames of the original animation and from the second to the bottom rows shows the corresponding frames of the imitated animation generated by ESK. The side-by-side comparison reveals a striking resemblance apart from the small errors in frames 3 and 5. As the original animation is keyframed in every frame to express the hopping step of natural running, frames 3 and 5 represent the motion of a man with his legs wide apart. In the imitated animations, however, frames 3 and 5 were interpolated smoothly without any keyframes.

Inspections of Fig. 5 show that a different selection of the basis function does not have much of a visual impact. However, when the controllers are placed outside of the markers in the creation of the exaggerated motion, especially in frames 1, 2, and 4 in Fig. 5(Exaggerated motion-linear), errors become noticeable regarding the blended motion. In particular, linear basis function clearly fails to extrapolate the motion reveals the visual artifacts.

The quantitative accuracy was also measured in a comparison with the original motion. The Mean Absolute Percentage Error (MAPE) [20] was used as the error metric. The error of all of the manipulators in each individual frame of the imitated animation was measured as follows:

$$Position\%Error_{x,y,z}of Each Manipulator = \frac{1}{n}\sum_{j=1}^{n}100\left|\frac{O_i^{x,y,z} - I_i^{x,y,z}}{O_i^{x,y,z}}\right| \tag{9}$$

To measure the error of the rotation angle, the following equation was applied:

$$Angle\%Error_{x,y,z}of Each Manipulator = \frac{1}{n}\sum_{j=1}^{n}100\left|\frac{O_i^{x,y,z} - I_i^{x,y,z}}{Range Angle Limits^{x,y,z}}\right| \tag{10}$$

$O_i^{x,y,z}$ and $I_i^{x,y,z}$ are the 3D positions and angles of the ith manipulator in the original animation and the corresponding imitation, respectively. The notation n denotes the number of frames of the entire animation. Errors of both natural and exaggerated motions were measured to find the optimal basis function of ESK. Intuitively, this error metric measures both the translational and the rotational deviations from the original animation for all the manipulators.

Table 2 shows the errors of all the manipulators of the imitated animation. When ESK imitated the natural motion of running, errors across all basis functions were acceptable except the z values of each leg and the arm IK handle. The z values contain the highest number of errors, as they represent the most varied positions. Sets of spatial keyframes that are more sophisticated would reduce number of z errors. However, even with the numerical errors of the z values, we did not observe any significant visual artifacts. In general, a MAPE of 10% is considered very good, and a MAPE in the range from 20% to 30% or even higher is quite common [20].

In terms of the basis function, Hardy multi-quadrics produced a steadier performance compared to the others. In particular, when ESK imitated the exaggerated motion of running, Hardy multi-quadrics showed the best performance. The notable point of the result is that a linear basis function for ESK performs poorly, as the function fails to extrapolate desirable motions with large errors. This is also not in accordance with the empirical selection of linear basis function of SK [1].

Performance test for complex animation

To verify the robustness of ESK, a task was performed to create complex motions within a short time. The analysis of the animation graphs automatically created by ESK offers the approximate number of keys that would be needed by keyframing. It clearly indicates that keyframing would require much effort compared to ESK. As shown in Table 3, we created 700 frames of a crawling iguana with eight sets of thirty spatial keyframes. This required less than one hour. We achieved very realistic motion of a crawling iguana (see Fig. 6)

The number of temporal keyframes that would be needed for the same animation was approximated using two methods provided in Autodesk Maya [14]. The first is a *bake simulation* that can bake the current animation. The second is a *simplify curves* method that can minimize the keyframes of the generated animation until the value differences of each frame are within the user tolerance levels. By comparing original keyframed data with the simplified data, the minimum number of keyframes was estimated with some degree of tolerable error against the original motion. The approximate data using bake simulation are mainly used to compare the motion accuracy while reducing the keyframes. The approximate data using simplify curves are mainly used to compare the task efficiency while sacrificing motion accuracy.

As the motions are all generated by the highest ranking controllers, no keyframes are placed in the manipulators. For comparison, the original motions were baked with samples at each frame for all channels (translation x, y, and z, and rotation x, y, and z) of the manipulators. When the user places a keyframe in more than three channels at once, instead of placing it in each channel individually, approximate keyframes by the user can be estimated by dividing the total number of keyframes by three degree of freedoms(DOFs). When this approximation is done using bake simulation, approximate keyframes should be divided by a sample interval, as the user does not place a keyframe in every frame during actual tasks. Therefore, the approximate keyframes of the motion were measured as follows:

$$ApproximateKeyframes = \frac{TotalFrames \times TotalDOFs \times TotalManipulators}{3DOFs \times SampleSpace} \tag{11}$$

The approximation using simplify curves requires two user inputs, time tolerance and value tolerance. For each simplified curve, *time tolerance* is the amount (in seconds) that the timing for the keys is averaged, and *value tolerance* is the amount (in working units) that the values of the keys are averaged when the selected curve is simplified [14]. A time tolerance of 4.0 and a value tolerance of 0.5 were used for the curve simplification. While the approximate data by simplifying curves substantially reduced the number of keyframes, the values of each keyframe were quite different compared to the original values. The errors of all of the manipulators in each frame were measured using MAPE as in Section . All approximate keyframe data and measured errors are shown in Table 4 and Table 5, respectively.

As shown in Table 4, the original animation had an enormous number of keyframes because they were generated by the user. In fact, it is impractical to generate them by a temporal keyframing. On the other hand, approximate keyframes showed a plausible number of keyframes for the animation of a similar complexity. However, the number continues to appear inordinately high for a novice user.

The errors in Table 5 represent the differences between the original and simplified animation. The more detailed animation the user desires to create, the more keyframes the animation needs in common. Therefore, though the simplified animation using simplify curves required fewer keyframes compared to both the original and the baked animation, monotonous or unnatural motion may result. The large errors of all the manipulators also meant that the style of the simplified animation differed greatly from that of the original animation.

Results

To demonstrate the effectiveness of ESK, a plug-in for Autodesk Maya (version 2008) [14] was created. The plug-in system is implemented in C++ with the Maya API. Several 3D rigged characters that support many manipulators for complex articulation were also created using the practical rigging technique introduced in several rigging manuals [21, 22].

A novice user with 3D animation was asked to try ESK. Only 10 minutes were required for training, and 20 minutes were necessary to design multiple sets of spatial keyframes using the User Interface provided for the user test. With a given 3D rigged mouse, the artist created 440 frames of a dancing mouse. The user commented that the system was easy to use and the process of creating animation was very intuitive. However, the user also pointed out that the system had inconveniences such as its inability to support a change in the previous pose before finishing the set and its inability to support visual classification of the hierarchy, as the color coding of all of the markers and the controllers are identical. We plan to incorporate the user critiques for future development.

Conclusion and Future Work

Creating animations using ESK is very easy and intuitive. All examples in this paper required only 30 to 50 minutes to design and create final motions.

The main contribution of ESK is that it automatically builds a global control structure by multiple sets of spatial keyframes. A single set of spatial keyframes was one of the critical limitations in SK. Multiple sets of spatial keyframes gave novice users great freedom when designing character motions.

The hierarchical relationship and its control is another key contribution of ESK. A hierarchical relationship allows complex motions with the simple control of the highest ranking controller by synchronizing multiple sets of spatial keyframes. A drawback is that the underlying sets of the hierarchy cannot be used to create different motions that deviate from the hierarchically inherited motion. This problem can be avoided by leaving the sets alone in the individual sets instead of in the hierarchy.

Layered acting and recording motions in multiple layers are not new. However, we applied an earlier approach [12] with modification using multiple layers with multiple sets of spatial keyframes, which gives the user the great advantage of reducing the number of controllers. As the motions can be recorded anytime into temporal keyframes, the user can iteratively record a desirable motion corresponding to the performance. This flexibility is missing in temporal keyframing methods, despite the fact that it is critical for creating motions with emotion.

A comparison between ESK and temporal keyframing showed several advantages of ESK very clearly. In addition, a performance test demonstrated the superiority of ESK compared to SK. The most remarkable feature is efficiency when creating complex motions. After defining multiple sets of spatial keyframes, performance by the user requires the moving of the highest ranking controllers across the hierarchy instead of the tedious placement of temporal keyframes on a time line. The approximate keyframes verify the efficiency of ESK.

Although we have overcome many limitations of SK, ESK still have issues to be addressed in the future. The first of these is related to how expressible ESK is across different types of motions. As with spatial keyframes, it is not easy to switch from one category of motion to another [1]. To create a meaningful story, it is essential to guarantee an easy transition across different types of motions. We plan to investigate this issue in the future.

The interface for 3D control is also an important issue. As with SK, ESK relies on mouse input. The user inevitably controls a high-degree-of-freedom character with a low-degree-of-freedom device. The user loses one degree of freedom when controlling the character with the mouse operation. Mordatch et al. [23] introduced a novel method to control spatial keyframes with a stochastic function and a new interface. We would like to pursue the direction of an interface for a novice user that allows the creation of complex motions easily and cheaply.

Our ultimate target users are professionals. They prefer precise and detailed control over a character, which is something we have yet to achieve. However, we strongly believe that ESK is well suited for the rapid generation of animation for less important characters such as crowds or distant actors. The development of a better interface technique and a more precise control technique will make ESK more versatile even for the creation of a main actor.

Acknowledgements

We would like to thank Jongmin Kim for valuable discussions, and all the members of the KAIST Visual Media Laboratory for their help with useful comments. This work was supported by IT R&D program of MIC/IITA (ITAA1100070400010001000300200, Software Development for Digital Creature).

References

[1] T. Igarashi, T. Moscovich, and J. F. Hughes. Spatial keyframing for performance-driven animation. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 107–115, New York, NY, USA, 2005. ACM.

[2] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[3] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3):72, 2007.

[4] Jovan Popović, Steven M. Seitz, and Michael Erdmann. Motion sketching for control of rigid-body simulations. *ACM Trans. Graph.*, 22(4):1034–1054, 2003.

[5] Matthew Thorne, David Burke, and Michiel van de Panne. Motion doodles: an interface for sketching character motion. *ACM Trans. Graph.*, 23(3):424–431, 2004.

[6] S. C. L. Terra and R. A. Metoyer. Performance timing for keyframe animation. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 253–258, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[7] Sílvio César Lizana Terra and Ronald Anthony Metoyer. A performance-based technique for timing keyframe animations. *Graph. Models*, 69(2):89–105, 2007.

[8] David R. Forsey and Richard H. Bartels. Hierarchical b-spline refinement. *SIGGRAPH Comput. Graph.*, 22(4):205–212, 1988.

[9] Zicheng Liu, Steven J. Gortler, and Michael F. Cohen. Hierarchical spacetime control. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 35–42, New York, NY, USA, 1994. ACM.

[10] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 39–48, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[11] A. Bruderlin, S. Fels, S. Esser, and K. Mase. Hierarchical agent interface for animation. In *Animated Interface Agents Workshop Proceedings of the International Joint Conference on Articial Intelligence (IJCAI'97).*

[12] Mira Dontcheva, Gary Yngve, and Zoran Popović. Layered acting for character animation. *ACM Trans. Graph.*, 22(3):409–416, 2003.

[13] Michael Neff, Irene Albrecht, and Hans Peter Seidel. Layered performance animation with correlation maps. *Computer Graphics Forum*, 26(3):675–684(10), 2007.

[14] Autodesk maya 2008, 2008.

[15] Greg Turk and James F. O'brien. Modelling with implicit surfaces that interpolate. *ACM Trans. Graph.*, 21(4):855–873, 2002.

[16] M.J.L. Orr. Introduction to radial basis function networks. *Center for Cognitive Science, University of Edinburgh, Scotland*, 1996.

[17] Junyong Noh and Ulrich Neumann. Expression cloning. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 277–288, New York, NY, USA, 2001. ACM.

[18] M. Eck. Interpolation methods for reconstruction of 3d surfaces from sequences of planar slices. *CAD und Computergraphik*, 13(5):109–120, 1991.

[19] R. Williams. The animator's survival kit tr897. 5, 2001.

[20] R.A. Yaffee and M. McGee. *An Introduction to Time Series Analysis and Forecasting: With Applications of SAS and SPSS*. Academic Press, 2000.

[21] Brad Clark, John Hood, and Joe Harkins. *Inspired 3D Advanced Rigging and Deformations*. HOMSON, 2005.

[22] S. Inc. *Learning Autodesk Maya 8— The Modeling and Animation Handbook with DVD*.

[23] Igor Mordatch, Patrick Coleman, Karan Singh, and Ravin Balakrishnan. Interface techniques for 3d control of spatial keyframing. In *SIGGRAPH '07: ACM SIGGRAPH 2007 posters*, page 84, New York, NY, USA, 2007. ACM.

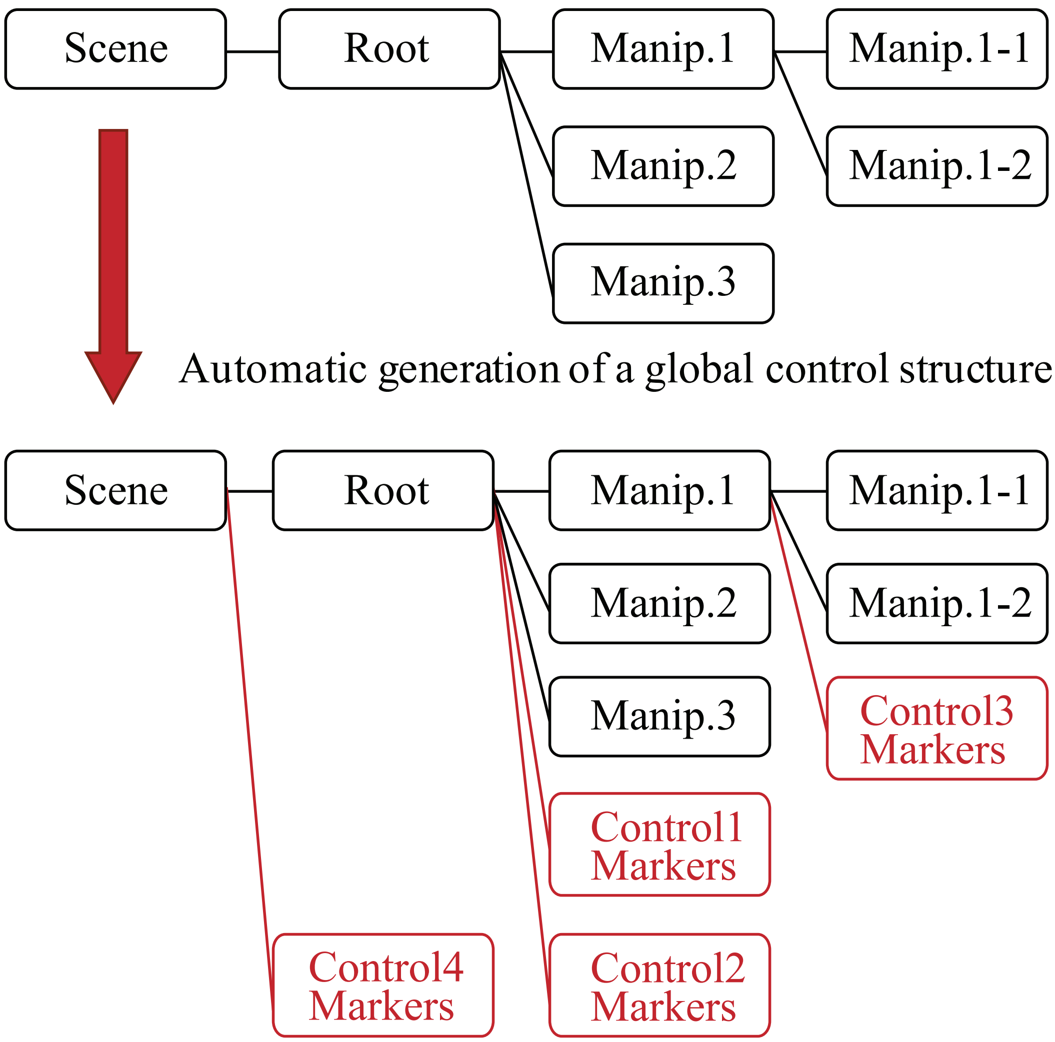

Figure 1: *An example of a global control structure by ESK. Generated controllers are automatically parented to the higher rank controller of the selected manipulators*

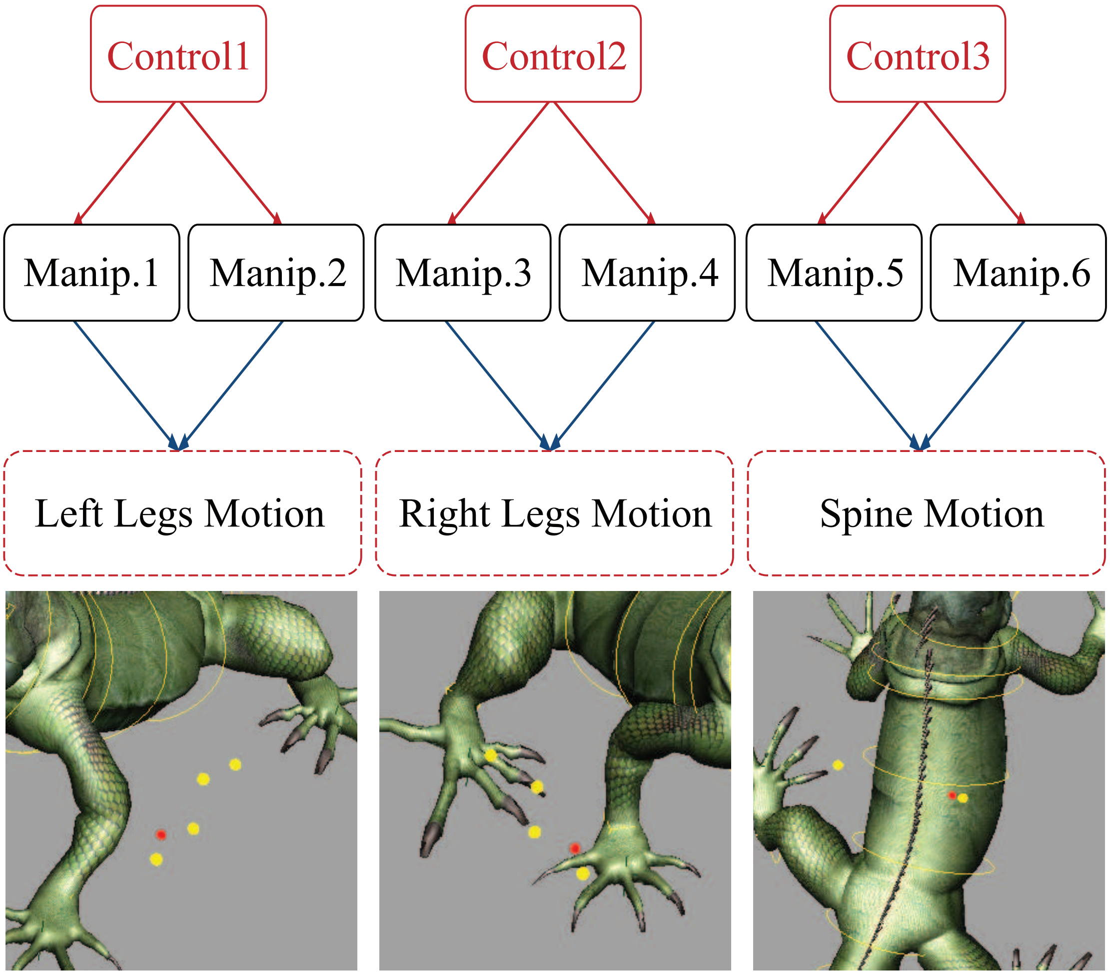

Figure 2: *An example of multiple sets: Each part can be articulated individually with a set of spatial keyframes. The yellow markers indicate spatial keyframes registered with desirable poses, and the red dots are the controllers of each set.*

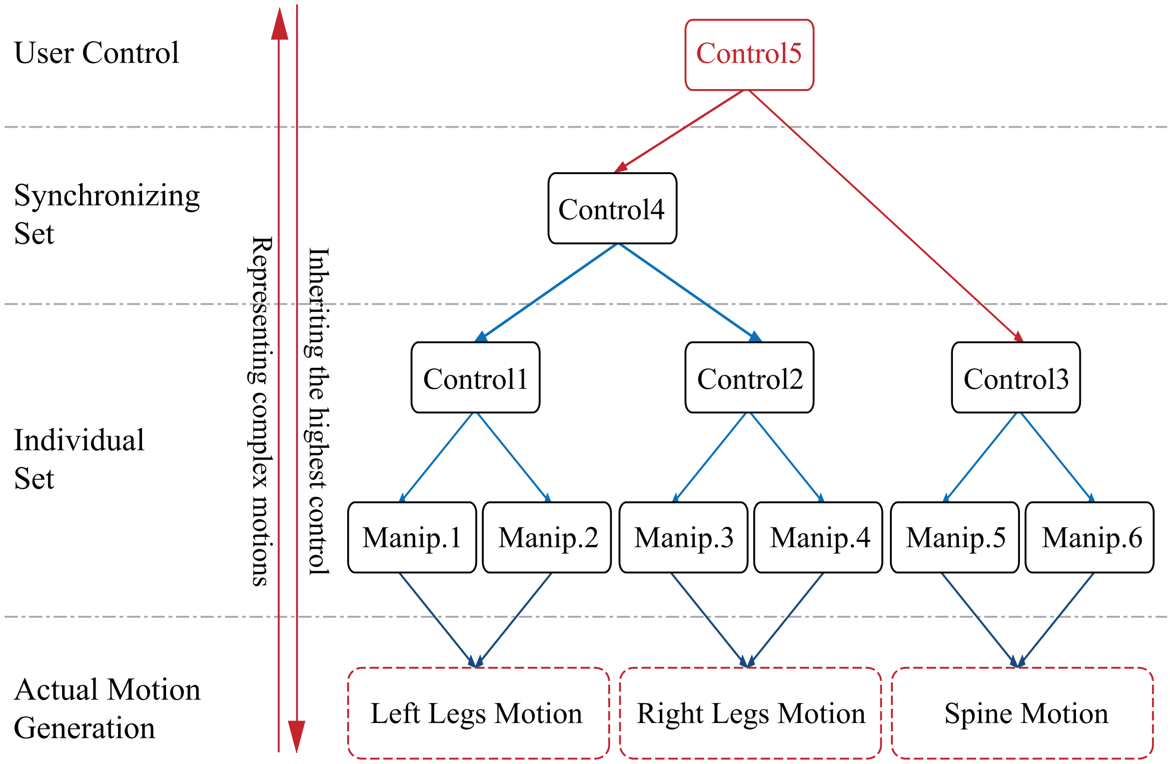

Figure 3: *Hierarchical relationships between controllers in multiple sets*

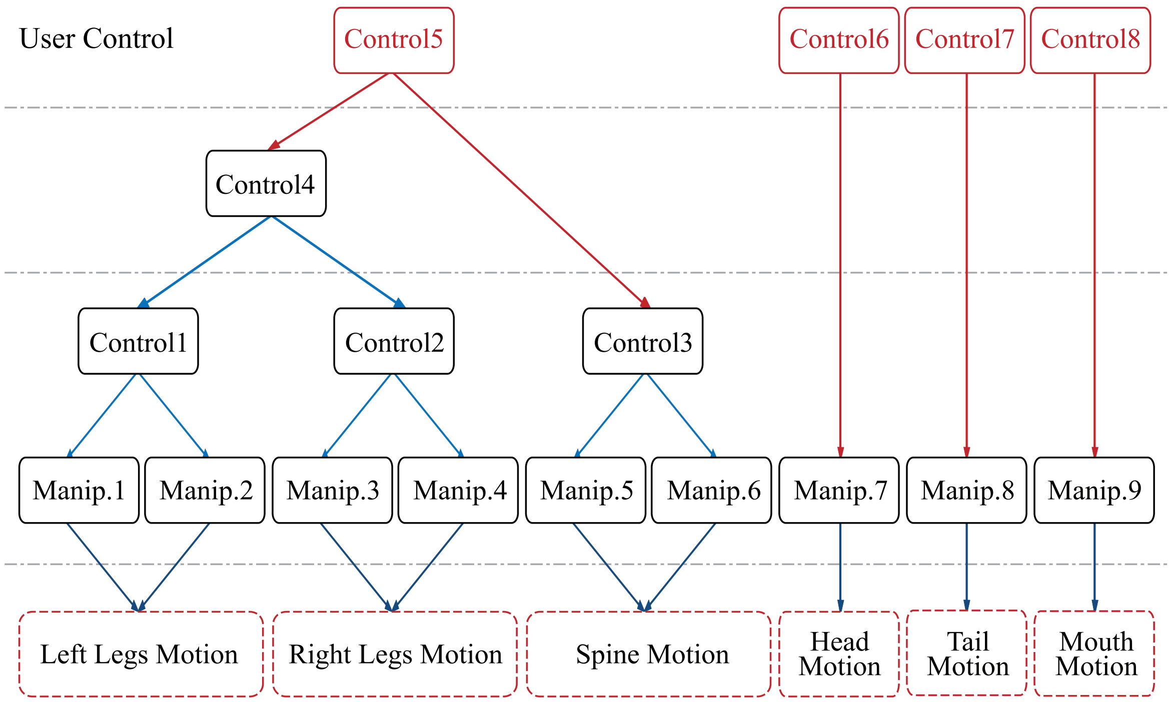

Figure 4: *Highest ranking controllers to record multiple layers*

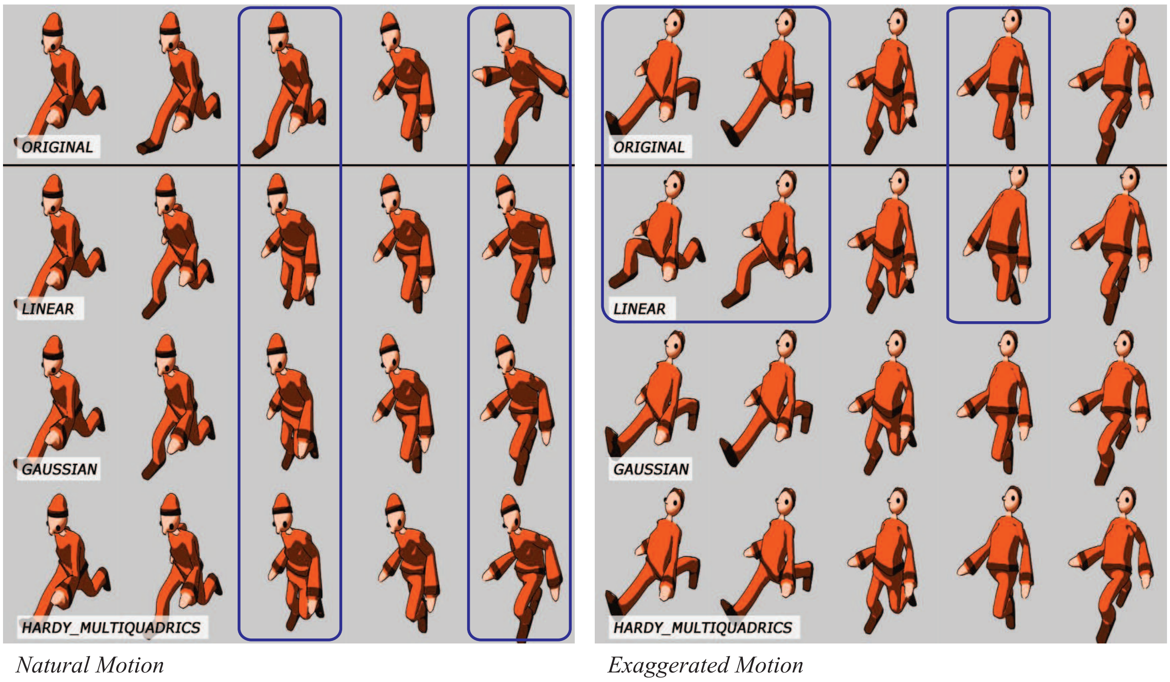

Figure 5: *Visual comparison of the natural motion and the exaggerated motion along to the basis function. The frames in the blue boexes represent the visual differences with errors*

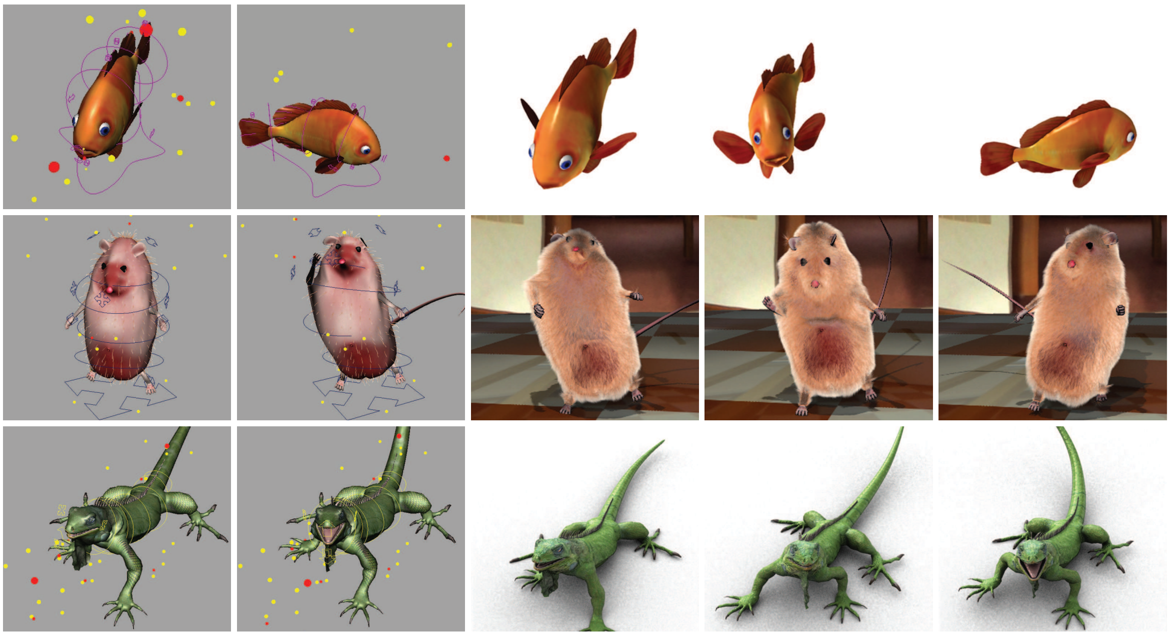

Figure 6: *Examples of a swimming fish(top), a dancing mouse(middle) and a crawling iguana(bottom) created by ESK*

<table>
<tr><th colspan="10">Natural Motion</th></tr>
<tr><td>Method</td><td colspan="4">Temporal Keyframing</td><td colspan="5">Extended Spatial Keyframing</td></tr>
<tr><td># of Keyframes</td><td colspan="4">76</td><td colspan="5">34</td></tr>
<tr><td>Making Time</td><td colspan="4">Approx. 30 min</td><td colspan="5">Approx. 30 min</td></tr>
<tr><td rowspan="5">Description in Detail</td><td colspan="4">Temporal Keyframes</td><td colspan="5">Spatial Keyframes</td></tr>
<tr><td>L Arm</td><td>R Arm</td><td>L Leg</td><td>R Leg</td><td>Legs</td><td>Arms</td><td>Spine</td><td>Head</td><td>Sync</td></tr>
<tr><td>13</td><td>13</td><td>13</td><td>13</td><td>4</td><td>4</td><td>3</td><td>3</td><td>4</td></tr>
<tr><td>Spine1</td><td>Spine2</td><td>Head</td><td rowspan="2"></td><td colspan="5" rowspan="2">*16 temporal keyframes from 3 controllers are also used</td></tr>
<tr><td>13</td><td>5</td><td>6</td></tr>
</table>

⇓ (User Extension)

<table>
<tr><th colspan="6">Exaggerated Motion</th></tr>
<tr><td>Method</td><td colspan="4">Temporal Keyframing</td><td>Extended Spatial Keyframing</td></tr>
<tr><td># of Modification</td><td colspan="4">70</td><td>11</td></tr>
<tr><td>Modification Time</td><td colspan="4">Approx. 25 min</td><td>Approx. 5∼10 min</td></tr>
<tr><td rowspan="5">Description in Detail (modifications)</td><td colspan="4">Temporal Keyframes</td><td rowspan="5">*11 temporal keyframes from 2 controllers are modified. No modification was needed to spatial keyframes</td></tr>
<tr><td>L Arm</td><td>R Arm</td><td>L Leg</td><td>R Leg</td></tr>
<tr><td>13</td><td>13</td><td>13</td><td>13</td></tr>
<tr><td>Spine1</td><td>Spine2</td><td>Head</td><td rowspan="2"></td></tr>
<tr><td>13</td><td>5</td><td>0</td></tr>
</table>

Table 1: *Production data of 25 frames cyclic animations of both a natural and an exaggerated running men*

Errors of the Imitated Animation from the Natural Motion(%)												
Basis Function	Left Arm IK			Right Arm IK			Left Leg IK			Right Leg IK		
	x	y	z	x	y	z	x	y	z	x	y	z
Linear	4.2	9.8	36.7	3.8	14.3	43.4	0.0	11.8	35.6	4.1	13.3	36.4
Gaussian	0.0	3.8	31.8	0.0	11.3	31.5	0.0	10.0	29.6	0.0	10.3	30.7
Hardy	2.8	9.6	27.2	2.3	13.8	36.9	7.3	8.6	27.5	6.7	10.2	28.8
Basis Function	Spine1 Joint (R.A.L=$\pi/3$)			Spine2 Joint (R.A.L=$\pi/3$)			Head Joint (R.A.L=$\pi 2$)					
	x	y	z	x	y	z	x	y	z			
Linear	3.0	1.1	1.8	n/a	n/a	0.0	0.0	n/a	n/a			
Gaussian	3.5	1.8	3.4	n/a	n/a	0.4	0.0	n/a	n/a			
Hardy	3.0	1.2	1.8	n/a	n/a	0.1	0.9	n/a	n/a			

Errors of the Imitated Animation from the Exaggerated Motion(%)												
Basis Function	Left Arm IK			Right Arm IK			Left Leg IK			Right Leg IK		
	x	y	z	x	y	z	x	y	z	x	y	z
Linear	4.1	5.8	43.2	4.0	5.3	12.8	37.5	18.0	19.4	321.0	23.4	30.2
Gaussian	5.4	5.1	7.3	5.6	4.8	20.9	45.4	15.0	17.6	299.7	16.7	39.9
Hardy	1.9	3.4	5.6	1.7	3.0	3.4	21.9	7.2	6.5	18.7	5.5	25.3
Basis Function	Spine1 Joint (R.A.L=$\pi/3$)			Spine2 Joint (R.A.L=$\pi/3$)								
	x	y	z	x	y	z						
Linear	0.3	1.8	2.4	n/a	3.6	0.2						
Gaussian	1.3	4.6	4.5	n/a	1.7	0.4						
Hardy	0.2	0.5	2.6	n/a	0.9	0.2						

Table 2: *Errors of the imitated animation from the natural motion(top) and the exaggerated motion(bottom)*

Crawling Iguana						
Making Time	Approx. 55 min					
# of Sets	8 (synchronizing sets: 2, individual sets: 6)					
# of Spatial Keyframes	30					
Description in Detail	Highest Ranking Controllers					
	Crawling	Head	Mouth	Tail		
	1	1	1	1		
	Synchronizing Sets(keyframes)					
	Spine & Legs			Left & Right legs		
	1(4)			1(4)		
	Individual Sets(keyframes) Included Manipulators to Each Set					
	L Legs	R Legs	Spine	Head	Mouth	Tail
	1(4)	1(4)	1(2)	1(5)	1(2)	1(5)
	2	2	5	1	2	1

Table 3: *Production data of a 700 frames crawling iguana*

Approximate Keyframe Data of the Crawling Iguana			
Total Frames	700		
Total DOFs	6(translation x, y, z + rotation x, y, z)		
Total Manipulators	13		
	Original Data	Bake Simulation	Simplify Curves
Sample Interval	1	4	n/a
Time Tolerance	n/a	n/a	4.0
Value Tolerance	n/a	n/a	0.5
Approx. Keframes	18200	4550	1314

Table 4: *Approximate keyframed data of the crawling iguana*

Errors of the Simplified Animation from the Original Animation(%)												
Simplify Method	Left Front Leg (translation)			Left Front Leg (R.A.L=π)			Left Rear Leg (translation)			Left Rear Leg (R.A.L=π)		
	x	y	z	x	y	z	x	y	z	x	y	z
B.S.	14.9	1.7	13.0	0.0	0.0	0.2	14.0	2.0	5.1	0.0	0.0	0.3
S.C.	32.3	13.2	77.0	0.0	0.1	0.1	25.7	9.2	18.9	0.1	0.1	0.4
Simplify Method	Spine1 (R.A.L=$\pi/3$)			Spine4 (R.A.L=$\pi/3$)			Head (R.A.L=$\pi/2$)			Tail (R.A.L=π)		
	x	y	z	x	y	z	x	y	z	x	y	z
B.S.	11.8	2.0	0.2	3.4	13.8	n/a	0.5	0.5	0.2	0.0	0.0	n/a
S.C.	123.9	12.7	0.5	21.0	157.4	n/a	1.0	0.4	0.6	0.1	0.1	n/a

Table 5: *Errors of the simplified animation*