KEYFRAME-BASED SUBACTORS

L.Forest, D.Rambaud, N.Magnenat-Thalmann, D.Thalmann

MIRALab
HEC/IRO
Université de Montréal
Montréal, Canada

Abstract

The concept of keyframe-based subactor attempts to span two major types of animation: parametric keyframe animation and algorithmic animation. In a keyframe-based subactor, all parameter values are defined by interpolation, however, if there is a law defined for one parameter, this law is applied and values computed by interpolation are ignored. The application of keyframe-based subactors to human motion is also discussed.

keywords: subactor, procedural law, parametric keyframe animation, algorithmic animation

Résumé

On introduit le concept de sous-acteur basé sur des dessins-clés pour tenter de concilier deux types principaux d'animation: l'animation paramétrique à dessins-clés et l'animation algorithmique. Dans un sous-acteur basé sur des dessins-clés, toutes les valeurs de paramètres sont définies par interpolation; cependant si une loi est définie pour un paramètre, cette loi s'applique et les valeurs calculées par interpolation sont ignorées. On décrit aussi une application de ces sous-acteurs basés sur les dessins-clés dans le domaine de l'animation de personnages tridimensionnels.

mots-clés: sous-acteur, loi procédurale, animation paramétrique à dessins-clés, animation algorithmique

Introduction

There have been two major approaches in the design of animation control (Steketee and Badler 1985; Parke 1982; Zeltzer 1985; Magnenat-Thalmann and Thalmann 1985) : keyframe animation and algorithmic animation. The concept of **keyframe-based subactor** attempts to span both types of animation.

Keyframe animation consists of the automatic generation of intermediate frames, called inbetweens, based on a set of keyframes supplied by the animator. There are two fundamental approaches to keyframe animation: **shape-interpolation** and **parametric keyframe animation**.

Shape interpolation is the three-dimensional analog of two-dimensional keyframing, introduced by Burtnyk and Wein (1971). Inbetween frames are computed by interpolating between the data points of the two objects.

In parametric keyframe animation systems (Steketee and Badler 1985; Parke 1982) inbetween frames are generated by interpolating the transformation parameters and transforming objects.

In algorithmic animation, motion is algorithmically described. Physical laws are applied to parameters of the objects. Control of these laws may be given by programming as in ASAS (Reynolds 1982) and MIRA (Magnenat-Thalmann and Thalmann 1983) or using an interactive director-oriented approach as in the MIRANIM (Magnenat-Thalmann et al 1985) system. With such an approach, any kind of law may be applied to the parameters. For example, the variation of a joint angle may be controlled by kinematic laws as well as laws based on dynamic analysis (Badler 1984; Armstrong and Green 1985; Wilhelms and Barsky 1985).

In keyframe animation, there are often undesirable effects such as lack of smoothness and discontinuities in motion. To reduce these effects, alternate methods to a linear interpolation have been proposed by Baecker (1969), Burtnyk and Wein (1976), Reeves (1981), Kochanek and Bartels (1984) However, according to Steketee and Badler (1985), with shape interpolation, there is no totally satisfactory solution to the deviations between the interpolated image and the object being modeled. Unless animators spend their time to digitize almost each frame.

Algorithmic animation is an excellent approach for most motions, however it tends to be complex for specifying human motions. Moreover, kinematic laws may be sometimes completely unrealistic and laws based on dynamic analysis are generally very expensive.

The concept of keyframe-based subactor

An actor as defined by Reynolds (1982) is a graphical entity with a given role to play. A subactor (Magnenat-Thalmann and Thalmann, 1985b) is an entity which is dependent on an actor. This means that all motions applied to an actor are also applied to all its subactors. The reverse is not true. There are also two other advantages to the subactor approach:

1. Any new subactor may be inserted as dependent on an existing actor.
2. Motions of different subactors may be coordinated and synchronized within an actor.

A subactor is a variable of type subactor, which is a data abstraction formulation of a class of entities composed of objects and internal transformations applied to them. Formally a subactor communicates with other entities by means of parameters, which may be time-dependent.

In a keyframe-based subactor, all parameter values are defined by interpolation, however, if there is a law defined for one parameter, this law is applied and values computed by interpolation are ignored. This approach has great advantages. Most of the parameters may be controlled by the keyframe process, which is less expensive in fact; however more realistic effects may be performed on selected parameters.

Keyframe-based subactors have been implemented as an extension of the MIRANIM system.

Subactors in the MIRANIM system

MIRANIM is an advanced system which allows the creation, manipulation and animation of realistic images. The most important features of MIRANIM are as follows:

- basic geometric primitives
- ruled and free-form surfaces
- multiple cameras and stereoscopy
- actor motions
- multiple lights and spots, shadows (Magnenat-Thalmann and Thalmann, 1985)
- transparency, three-dimensional texture, fractals, particle systems

Image rendering may be performed by a scanline z-buffer algorithm or a ray tracing algorithm.

MIRANIM is mainly based on three components:
1) the object modelling and image synthesis system BODY-BUILDING
2) the director-oriented animation editor ANIMEDIT
3) the actor-based sublanguage CINEMIRA-2

ANIMEDIT is a scripted system; the director designs a scene with decors, actors, cameras and lights. Each of these entities is driven by animated variables, which are, in fact, state variables following evolution laws. CINEMIRA-2 allows the director to use programmers to extend the system. The great advantage of this is that the system is extended in a user-friendly way. This means that the director may immediately use the new possibilities. An entity programmed in CINEMIRA-2 is directly accessible in ANIMEDIT. This not only extends the system, but also enables specific environments to be created. For animation, CINEMIRA-2 allows the programming of five kinds of procedural entities: objects, laws of evolution, actor transformations, subactors, animation blocks.

A CINEMIRA-2 subactor is dependent on an actor which may be transformed in ANIMEDIT by a list of global transformations like translation, rotations, shear, scale, color transformation, flexion, traction. Several actors like these may participate in the same scene with other actors implemented using only algorithmic animation, cameras, lights and decor.

Application of keyframe-based subactors to human motion

A new system has now been designed and implemented: BODY-MOVING; this is a parametric key-frame animation system in which human bodies are mainly controlled by joint angles. BODY-MOVING is an interactive program that allows the user to build any sequence of motion for a given three-dimensional character. Actually, motion is controlled by 50 joint angles. A keyframe is specified by modifying values for these angles from the previous keyframe in the sequence. Corrections may be done vertically for any keyframe, or horizontally for a given parameter in each keyframe. The animator may look at parameter values for any keyframe or interpolated frame. He/she also may obtain a wire-frame view of the human bodies for any frame.

For each parameter, interpolation may be computed linearly or using bicubic splines (Kochanek and Bartels 1984).

Once the motion of the three-dimensional character is designed, the character needs to be covered with surfaces. In our experimental system, we try to completely separate the topology of the surfaces from the wire-frame model. This means that parts of the human bodies may be designed using ruled surfaces such as revolution surfaces, free-form surfaces or three-dimensional reconstructed surfaces obtained from digitized projections. The system transforms the surfaces according to the wire-frame model assuring an automatic continuity between the different surfaces. This correspondance is based on a changing of reference systems independent of the segment length. This means that for the same set of surfaces, several bodies of different sizes may be obtained according to the segment length in the wire-frame models. This technique may be considered as a three-dimensional skeleton technique (Burtnyk and Wein 1976).

For example consider a point between the elbow and the wrist; when we change the reference system, it is important to notice that both parts may be bent and/or twisted. This means that the surface must be extended on the external side of the elbow and twisted at the wrist, while preserving continuity.

Integration strategy

The integration of parametric keyframe animation and algorithmic animation has been performed considering that any human body designed with BODY-MOVING is a subactor in MIRANIM. This subactor has 50 real parameters, grouped in 17 three-dimensional vector parameters where each component is an angle. Each vector parameter is identified by a name; for example LEFTSHOULDER is the three-dimensional vector that controls the motion of the shoulder. If there is no law defined for this parameter, the interpolated values are taken. If there is a law defined for the parameter, this law is applied and values computed by interpolation are ignored. This approach of keyframe-based subactor has great advantages. Most of the angles may be controlled by the keyframe process, which is less expensive in fact; however more realistic effects may be performed on selected angles. For example, laws based on dynamics have been implemented using similar equations to those described by Armstrong and Green (1985). Of course, to obtain an angle following a law based on the dynamic analysis, dynamic properties like masses, forces, inertia matrices and torques have to be supplied. Intrinsic properties of bodies like masses and moments of inertia may be given at the creation of the surfaces in BODY-BUILDING. Forces and torques have to be specified as parameters of the laws. With our approach, expensive computations are performed only when absolutely necessary. Our approach to the integration of the different techniques is as follows: the joint angles must vary according to the values calculated by BODY-MOVING, but it is also possible to have one or more angles following a predefined law or programmed with the CINEMIRA-2 sublanguage.

To control algorithmically the evolution of an angle, the animator may use three kinds of laws: predefined laws, CINEMIRA-2 analytical laws and functions of a previous state. In this latter case, an evolution law may be completely changed at any time (and consequently at any frame); this allows the animator to adapt the evolution of a joint angle to a new situation.

The integration approach has another important application: the relation between MIRANIM actors and human characters generated by BODY-MOVING. The typical case is when the value of a parameter (angle) of the human character have to be derived from data for an actor. For example, the MIRANIM actor is a ball and the human character receives the ball on the head. In this case, the motion of the character has to be controlled using data about the ball. To solve this case, our approach is to predefine functions which return at any frame the value of any parameter. These laws may be then applied to any animated variable which drives the motion of others actors, cameras or lights.

Fig.1 shows the integration of human keyframe-based subactors into the MIRANIM system.

Acknowledgements

The research was supported by the Natural Sciences and Engineering Council of Canada, the Art Council of Canada and the Government of Quebec.

References

Armstrong, W.W. and Green, M. (1985) **The Dynamics of Articulated Rigid Bodies for Purposes of Animation**, The Visual Computer, Vol.1, No 4, pp.231-240.

Badler, N.I. (1984) **Design of a Human Movement Representation Incorporating Dynamics**, Technical Report, Department of Computer and Information Science, University of Pennsylvania.

Baecker, R. (1969) **Picture-driven Animation**, Proc. AFIPS Spring Joint Computer Conf., Vol.34, pp.273-288.

Burtnyk, N. and Wein, M. (1971) **Computer-generated Key-frame Animation**, Journal of SMPTE, 80, pp.149-153.

Burtnyk, N. and Wein, M. (1976) **Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation**, Comm. ACM, Vol.19, No10, pp.564-569.

Forest, L.; Magnenat-Thalmann, N. and Thalmann, D. (1986) **Integrating Keyframe Animation and Algorithmic Animation of Articulated Bodies**, Proc. Computer Graphics Tokyo '86, Springer, Tokyo.

Fortin, M.; Léonard, N.; Magnenat-Thalmann, N. and Thalmann, D. (1985) **Animating Lights and Shadows**, Computer-Generated Images, Springer, Tokyo Berlin Heidelberg New York, pp. 45-55.

Kochanek, D. and Bartels, R. **Interpolating Splines with Local Tension, Continuity and Bias Tension**, Proc. SIGGRAPH '84, pp.33-41.

Magnenat-Thalmann, N. and Thalmann, D. (1983) **The Use of High-Level Graphical Types in the MIRA Animation System**, IEEE Computer Graphics and Applications, Vol. 3, No 9, pp.9-16.

Magnenat-Thalmann N. and Thalmann, D. (1985) **Computer Animation: Theory and Practice**, Springer-Verlag, Tokyo

Magnenat-Thalmann, N. and Thalmann, D. (1985b) **Subactor Data Types as Hierarchical Procedural Models for Computer Animation**, Proc. EUROGRAPHICS '85, Nice, France, North Holland, pp.121-128

Magnenat-Thalmann, N.; Thalmann, D. and Fortin, M. (1985) **MIRANIM: An Extensible Director-Oriented System for the Animation of Realistic Images**, IEEE Computer Graphics and Applications, Vol. 5, No 3, pp. 61-73

Parke, F.I. **Parameterized Models for Facial Animation**, IEEE Computer Graphics and Applications, Vol.2, No9, 1982, pp.61-68

Reeves, W. **Intbetweening for Computer Animation Utilizing Moving Point Constraints**, Proc. SIGGRAPH '81, Vol.15, No3, 1981, pp.263-269.

Reynolds, C.W. (1982) **Computer Animation with Scripts and Actors**, Proc. SIGGRAPH'82, pp.289-296

Steketee, S.N.; Badler, N.I. **Parametric Keyframe Interpolation Incorporating Kinetic Adjustment and Phrasing Control**, Proc. SIGGRAPH '85, pp. 255-262.

Wilhelms, J.P. and Barsky, B.A. (1985) **Using Dynamic Analysis to Animate Articulated Bodies such as Humans and Robots**, Springer Tokyo Berlin Heidelberg New York, pp.209-229.

Zeltzer, D. (1985) **Towards an Integrated View of 3D Computer Animation**, The Visual Computer, Springer, Vol.1, No4, pp.249-259.

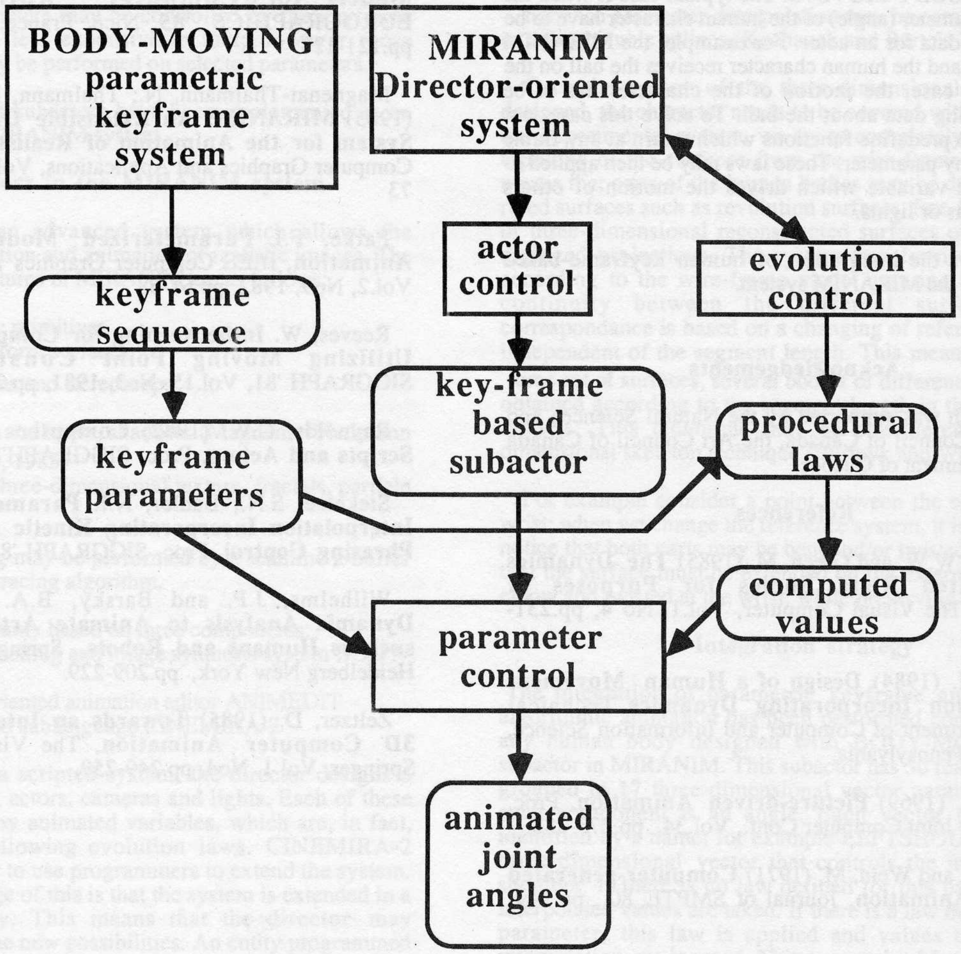

Fig.1 The integration of keyframe-subactors into the MIRANIM system