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 Stekettee 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