A

Animated Characters within the MPEG-4 Standard

Françoise Prêteux
Institut National des Télécommunications/Groupe des Ecoles de Télécommunications, France

Marius Preda
Institut National des Télécommunications/Groupe des Ecoles de Télécommunications, France

INTRODUCTION

The first 3D virtual human model was designed and animated by means of the computer in the late '70s. Since then, virtual character models have become more and more popular, making a growing population able to impact the everyday real world. Starting from simple and easy-to-control models used in commercial games, to more complex virtual assistants for commercial[1] or informational[2] Web sites, to the new stars of virtual cinema,[3] television,[4] and advertising,[5] the 3D character model industry is currently booming.

Moreover, the steady improvements within the distributed network area and advanced communication protocols have promoted the emergence of 3D communities[6] and immersion experiences (Thalmann, 2000) in distributed 3D virtual environments.

BACKGROUND

Animated Characters and 3D Standards

Creating, animating, and most of all, sharing virtual characters over Internet or mobile networks require unified data formats. If some animation industry leaders try—and sometimes succeed[7,8]—to impose their own formats in the computer world mainly by making available powerful authoring platforms, the alternative of an open standard is the only valid solution ensuring interoperability requirements, specifically when hardware products are to be built.

A dream of any content producer can be simply formulated as "creating once and reuse for ever and everywhere, in any circumstances." Nowadays, content should be carried by heterogeneous networks (broadcast, IP,[9] mobile), available anywhere and for a large scale of devices (PCs, set-top boxes, PDAs,[10] mobile phones), and profiled with respect to user preferences. All these requirements make that the chain where content is processed more and more complex, and a lot of different actors must interfere: designers, service providers, network providers, device manufacturers, IPR[11] holders, end-users, and so on. For each one, consistent interfaces should be created on a stable and standardized basis.

Current work to provide 3D applications within a unified and interoperable framework is materialized by 3D graphics interchange standards such as VRML[12] ISO/IEC 14772-1:1997 and multimedia 2D/3D standards such as MPEG-4–ISO/IEC 14496. Each one addresses, more or less in a coordinated way, the virtual character animation issue. Moreover, some research groups proposed dedicated languages for modeling virtual faces and bodies, such as Face Markup Language (FML) and Virtual Human Markup Language (VHML). In the VRML community, the H-Anim[13] group released three versions of their specifications (1.0, 1.1 and 2001), as did the SNHC[14] sub-group of MPEG: MPEG-4 Version 1 supports face animation, MPEG-4 Version 2 supports body animation, and MPEG-4 Part 16 addresses the animation of generic and articulated virtual objects (including human-like). In MPEG-4, the specifications dealing with the definition and animation of human avatars are grouped under the name FBA—Face and Body Animation—while those referring to generic models are called BBA—Bone-Based Animation. The next section analyses the main similarities and differences of these two standardization frameworks: VRML and MPEG-4.

The VRML standard provides a textual description of 3D objects and scenes. It focuses on the spatial representation of such objects, while the time behavior is less supported. The major mechanism for supporting animation consists of defining it as an interpolation between key-frames.

The MPEG-4 standard, unlike the previous MPEG standards, does not only cope with highly efficient audio and video compression schemes, but also introduces the fundamental concept of media objects such as audio, visual, 2D/3D natural, and synthetic objects to make up a multimedia scene. As established in July 1994, the MPEG-4 objectives are focused on supporting new ways (notably content-based) of communicating, accessing, and manipulating digital audiovisual data (Pereira, 2002). Thus, temporal and/or spatial behavior can be associated with an object. The main functionalities proposed by the standard address the compression of each type of media objects, hybrid encoding of the natural and synthetic objects, universal content accessibility over various networks, and interactivity for the end-user. In order to specify the spatial and temporal localization of an object in the scene, MPEG-4 defines a dedicated language called BIFS—Binary Format for Scenes. BIFS inherits from VRML the representation of the scene, described as a hierarchical graph, and some dedicated tools such as animation procedures based on interpolators, events routed to the nodes, or sensor-based interactivity. In addition, BIFS introduces some new and advanced mechanisms such as compression schemes to encode the scene, streamed animations, integration of 2D objects, new 3D objects, and advanced time control.

In terms of functionalities related to virtual characters, both VRML and MPEG-4 standards define a set of nodes in the scene graph to allow for a representation of an avatar. However, only the MPEG-4 SNHC specifications deal with streamed avatar animations. A major difference is that an MPEG-4-compliant avatar can coexist in a hybrid environment, and its animation can be frame-based synchronized with other types of media objects, while the H-Anim avatar is defined in a VRML world and must be animated by VRML generic, usually non-compressed, animation tools.

The question that arises now is how to find a good compromise between the need for freedom in content creation and the need for interoperability? What exactly should be standardized, fixed, invariant, and in the mean time, ideally impose no constraints on the designer creativity? The long-term experience that the MPEG community has makes it possible to formulate a straight and solid resolution: in the complex chain of content, producing-transmitting-consuming the interoperability is ensured by only standardizing the data representation format at the decoder side. Pushing this concept to its extreme, an MPEG ideal tool is that one for which two requirements are satisfied: the designer can use any production tool he/she possesses to create the content, and it can be possible to build a full conversion/mapping tool between this content and an MPEG-compliant one. The same principle was followed when MPEG released the specifications concerning the definition and the animation of the virtual characters: there are no “limits” on the complexity of the virtual character with respect to its geometry, appearance, or skeleton and no constraints on the motion capabilities.

MAIN THRUST OF THE ARTICLE

The animation method of a synthetic object is strongly related to its representation model. A simple approach often used in cartoons is to consider the virtual character as a hierarchical collection of rigid geometric objects called segments, and to obtain the animation by transforming these objects with respect to their direct parents.

The second method consists of considering the geometry of the virtual character as a unique mesh and to animate it by continuously deforming its shape. While the former offers low animation complexity with the price of the seams at the joints between the segments, the latter ensures a higher realism of the representation, but requires more computation when synthesizing the animation. Both methods are supported by the MPEG-4 standard and will be detailed in the next sections.

Segmented Character: MPEG-4 FBA Framework

First efforts to standardize the animation of a human-like character (an avatar) within MPEG-4 were finalized at the beginning of 1999 and published as FBA. The approach used here consists of considering the avatar as a hierarchical structure of 3D rigid objects, each one corresponding to an anatomical segment (arm, forearm, etc.). There are no constraints on the geometry of a segment, but the hierarchical structure is predefined. The animation is performed by updating the geometric transformation between the segments. A separate, so-called FBA stream contains the compressed version of those transformations, expressed as extrinsic properties of an anatomical segment, that is, its 3D pose with respect to a reference frame attached to the parent segment.

The orientation of any anatomical segment is expressed as the composition of elementary rotations, namely twisting, abduction, and flexion. For some segments only one or two elementary rotations are supported. Hence, 296 angular joint values are enough to describe any 3D posture of a virtual human-like character. The angular values are specified with respect to the local 3D coordinate system of the anatomical segment. The origin of the local coordinate system is defined as the gravity center of the joint contour common to the considered anatomical segment and its parent. The rotation planes are specified, and/or anatomical segment rotation axes are standardized.

Due to the rigid nature of the anatomical segments, when performing the animation, seams can occur at joints between segments. This effect can be corrected by using local deformation tables that give hints on how to locally update the position of certain vertices on the anatomical segment with respect to the animation parameters (Capin, 1999; Preda, 1999, 2002).

To animate the face of the avatar, FBA uses a surface-based model driven by control points: a standardized number of key points (84) corresponding to the human

Related Content