MPEG-4: Audio/Video & Synthetic Graphics/Audio for Mixed Media

Peter K. Doenges [1]
Evans & Sutherland Computer Corp.
Tolga K. Capin [2]
Computer Graphics Lab/Swiss Federal Institute of Technology
Fabio Lavagetto [3]
DIST/University of Genoa
Joern Ostermann [4]
AT&T Research Laboratories
Igor S. Pandzic [5]
MIRALab/University of Geneva
Eric D. Petajan [6]
Bell Laboratories/Lucent Technologies

Abstract

MPEG-4 addresses coding of digital hybrids of natural and synthetic, Aural and Visual (A/V) information. The objective of this Synthetic/Natural Hybrid Coding (SNHC) is to facilitate content-based manipulation, interoperability, and wider user access in the delivery of animated mixed media. SNHC will support non-real-time and passive media delivery, as well as more interactive, real-time applications. Integrated spatial-temporal coding is sought for audio, video, and 2D/3D computer graphics as standardized A/V objects. Targets of standardization include mesh-segmented video coding, compression of geometry, synchronization between A/V objects, multiplexing of streamed A/V objects, and spatial-temporal integration of mixed media types. Composition, interactivity, and scripting of A/V objects can thus be supported in client terminals, as well as in content production for servers, also more effectively enabling terminals as servers. Such A/V objects can exhibit high efficiency in transmission and storage, plus content-based interactivity, spatial-temporal scalability, and combinations of transient dynamic data and persistent downloaded data. This approach can lower bandwidth of mixed media, offer tradeoffs in quality vs. update for specific terminals, and foster varied distribution methods for content that

[1] Evans & Sutherland Computer Corp., 600 Komas Drive, P.O. Box 58700, Salt Lake City, Utah 84158, USA, E-mail: pdoenges@es.com

[2] Computer Graphics Lab (EPFL-LIG), Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland, E-mail: capin@di.epfl.ch, Web: http://ligwww.epfl.ch/

[3] DIST - Department of Telecommunications, Computer and System Sciences, University of Genoa, Via Opera Pia 13, 16145 Genova, Italy, E-mail: fabio@dist.dist.unige.it, Web: http://www.dist.unige.it/

[4] AT&T Research Laboratories, Room HO 4E518, 101 Crawfords Corner Road, Holmdel, NJ 07733-3030, USA, E-mail: ostermann@research.att.com

[5] MIRALab - CUI, University of Geneva, 24 rue du Général-Dufour, CH1211 Geneva 4, Switzerland, E-mail: Igor.Pandzic@cui.unige.ch, Web: http://miralabwww.unige.ch/

[6] Bell Laboratories, Lucent Technologies, Room 2B-231, 600 Mountain Avenue, Murray Hill, New Jersey 07974, USA, E-mail: edp@bell-labs.com

exploit spatial and temporal coherence over busses and networks. MPEG-4 responds to trends at home and work to move beyond the paradigm of audio/video as a passive experience to more flexible A/V objects which combine audio/video with synthetic 2D/3D graphics and audio.

1 Introduction

1.1 Convergence of A/V & 2D/3D

Trends in networking, in decentralization of media production and consumption, and in computer graphics at work and home point toward changes in distributing passive and interactive mixed media. Audio/video and 2D/3D synthetic graphics are merging into hybrid compositions in a variety of formats and platforms that extend the role of television and the PC. This evolution widely spans lower-bitrate applications like video cellular telephony, and higher-bandwidth, networked, interactive, real-time media experiences like distance learning, gaming, and training.

In this discussion, “real-time” means fixed, periodic, deterministic rates of update and refresh in the delivery and presentation of a series of media states, which are fast enough to convey smooth motion or to minimize lag so a user can accomplish a task. “Non-real-time” usually corresponds with non-periodic, non-deterministic update rates or significant lag. “Passive” media may be real-time or non-real-time, but offers no user control of content during an A/V sequence. “Interactive” media, on the other hand, allows the user to manipulate content or to control its display. “Task-loading” of a user means that a media presentation is sufficiently intense that it stimulates sensory/cognitive loading of a user to accomplish a task “realistically” (e.g. skill training, gaming).

A mixed-media coding approach could further improve the efficiency of communication, and offer more bitrate independence (or tolerance) for hybrid media in passive and interactive communications over a range of networks and platforms. Capability is also needed to compose audio/video segments more flexibly as A/V objects, in a manner similar to 2D/3D graphics and synthetic audio. This can infuse audio/video with locality, shape, priority, ranging, scalability, control, and other "objectness" suited to interactivity and content-based access. Mesh-based coding in current video compression work offers gains in functionality that shift the emphasis from achieving strictly lowest bitrates to content manipulation and interactivity.

Emerging silicon and software systems are moving toward delivery of hybrid content for real-time experiences with a high level of integration of computing resources, algorithms, and data primitives to decode, animate, render, and composite scenes. A/V objects can exist as transient or stored data in channels and media such as the Internet, ATM/B-ISDN communications, CD-ROM, on-line modifiable disks that page active data, archival digital libraries, and the memories of servers, decoders, PCs, graphics accelerators, and newer media processors.

1.2 MPEG Focus

The MPEG domain is efficient coding of A/V data to enable delivery and presentation of media experiences. MPEG-1 and 2 established respected standards for coding of real-time audio and video for storage and communications, motivated at each step by CD-ROM and digital television. The primary model relating user and supplier of media involved the streaming of real-time

compressed audio/video data for synchronous decoding and presentation at receivers. This audio/video was displayed at a modest set of fixed formats, bandwidths, and frame rates. Transient content was delivered from server to client and consumed immediately after decompression, with little surviving trace of the data. The content was intended to be primarily natural scenes including, of course, a human presence. The resulting audio and video codecs take statistical advantage of such natural scene content.

MPEG content is distinctive as animated, life-like, absorbing, or entertaining. It may even task-load a user as defined earlier. However in prior MPEG standards, no native support is provided for synthetic objects, although synthetic imagery and audio have utilized MPEG in TV and PC displays at reduced quality without interactivity. MPEG-4 extends to delivery of more complex, structured experiences combining the immediacy of audio/video with downloaded A/V object structures that a user might change with interaction. SNHC seeks to extend the compression inherent in synthetic models as a way to describe natural and synthetic content, and to join audio/video with 2D/3D graphics.

1.3 Industry Forces in Hybrid Media

Synthetic graphics is converging with imaging. In recent years, technical conferences and product developments have trumpeted the convergence of audio/video and 2D/3D, and the coming ubiquity of real-time 3D and media processors for consumer applications. Attention has focused on media graphics, hierarchical complexity management to navigate 3D scenes [1] and images, scalability (multi-resolution video [2], 3D level of detail, and 3D progressive transmission [3]), efficient coding (compression of geometry, radiometry, and structure for 3D models [4],[5],[33]), real-time image manipulation (affine warping), image-based 3D viewing from implicit models (panorama graphics [6], light fields, plenoptic modeling [7]), and networked virtual environments.

Media Digital Signal Processors (DSPs) accelerate signal processing, built-in audio/video codecs, even scene management and geometry processing, as front ends for media pipelines or 2D/3D graphics. Media kits and their Application Programmer Interfaces (APIs) show more attention to spatial audio and aural environments. The illusive set-top box revealed mixed results in testing of centralized multi-channel program servers. Yet it could provide hybrid interactive program guides, media-based Graphical User Interfaces (GUIs), product or recreation catalogs, news filters, and personal tools. Recent announcements by media companies target a different paradigm for the set-top box as an Internet Web server at low cost ($100-150) with a back channel. Also targeted is an 18" satellite dish at 10 Mbit/s for home delivery of such services with a telephone link. The shifts to interaction with downloaded data, to exploring content supported by an adequate back-channel, to TV or PC as family learning resource, and to interactive experiences beyond canned programs, invite interoperable hybrid coding with new functionality and bitrate efficiency.

Some 40 companies are developing hybrid media and 3D chips as well as plug-in boards for PCs and arcades. Miniaturized versions of 3D graphics and layered 2D/3D sprite compositing have appeared in video game consoles. The recent Talisman architecture [8] combines 3D graphics with object motion perspective simulated by affine warping of 3D sprites. This exploits object spatial and temporal coherence, compression/decompression in the pipeline for reduced memory size and bandwidth, and animated object views via layered image compositing. Trade-offs are

made to sacrifice 3D object perspective updates for image quality, for multi-pass special effects, for low latency in real-time interaction, and for more equal treatment of audio, video, 2D, and 3D. Some industry experts talk of 100M PC platform sales by 1998, half with media/3D capability, and many equipped for real-time experiences and networking. Perhaps even the lowly automated teller machine, phone booth, information kiosk, library catalog, therapy device, or lab trainer will benefit from these developments.

A/V content must be geared for speed and communications efficiency. This A/V content will be authored and integrated by far-flung communities, used with interaction (not just consumed), and transmitted between potentially dissimilar authoring and animation systems. The current Web build-out encourages diverse on-line media projects, including mixed media animation at rates striving toward deterministic real-time rates. Examples include Java animation of reactive 2D/3D models for learning or advertising, Virtual Reality Modeling Language (VRML) for immersive 3D model interaction on the Web, and varied media on dedicated networks for special purposes like multi-user gaming and digital studios. Hybrid media, that exhibits standard coded representations for sharing and synchronizing models under real-time control between different clients and servers, will help industry and users. What might help close the real-time gap?

1.4 Synthetic/Natural Hybrid Coding

MPEG-4 addresses the implications of storing and communicating animated media objects in computing environments where content production and interaction are possible for many. SNHC aims to integrate coding of animated 2D/3D computer graphics and synthetic audio with audio/video of natural sound and imagery encoded by signal processing with structures that support their spatial-temporal manipulation. SNHC objectives include compression (synthetics as well as signals), real-time determinacy (if feasible with the local client animation engine, renderer, and/or compositor), media integration (streaming and downloaded), synchronization (multiple animation and streaming sources and rates), geometry/text/etc. streams as well as traditional audio/video streams, spatial and temporal scalability of A/V objects based on a continuous space-time convention, interactivity where applicable, and extensibility.

Implementations could be as deceptively basic as an audio/video pane or sprite with subordinated dynamic 2D graphic overlays. These might be scripted or interactive, and static or moving (e.g. text, icons, 2D renderings). More challenging could be a remotely animated, locally cached 3D virtual environment with embedded (local or remote) streaming objects. Examples include 2D/3D faces, characters, or moving objects, and movies or "flip books," via geometry and texture animations within a 3D world. In both cases, content, composition, and behavior of A/V objects may change as a user interacts with the scene and networked users. A need for reciprocity and equal treatment of A/V objects dawns: hybrid coding should support 2D/3D under audio/video, as well as vice versa. The challenge is careful, step-wise integration of coding with a wide range of bitrates and terminal resources in mind --- not reinventing graphics or behavioral modeling.

1.5 About Paper

This paper describes the objectives and approach for the coding of hybrid animated media and synthetic environments capable of interactivity and scalability. We use some anchor applications to clarify what media representations and codec functionalities are sought for standardization.

SNHC is focused on coding efficiency, and on ensuring effective media delivery in client platforms with a range of animation and rendering capabilities. SNHC is intended to co-exist with modeling tool kits, rendering APIs, and scripting/behavior animation languages.

This paper offers views of the motivations to standardize, target applications, requirements, the system model, what is targeted for standardization, and potential ties to other media standards. Section 2 expands potential SNHC applications, offers associated requirements for coding in the MPEG-4 framework, and discusses a layered approach to standardization that is intended to build progressively on successive steps. Section 3 discusses the system receiver model as well as concerns about what to standardize and how coding tradeoffs can be exploited. Section 4 focuses on geometry compression and its contribution to the efficiency of downloaded and streamed geometry when models are transmitted and animated (analogous to audio/video compression). Section 5 concentrates on facial/body animation with speech and the advantages of model-based parameter streams to animate models relative to transmitting audio/video. Section 6 caps the discussion with media integration and coding challenges associated with 3D virtual environments.

The MPEG-4 standards process is on-going. The work-in-progress is vitally dependent on participation. As a necessary caveat, this paper assembles the current views of some individuals involved in the MPEG-4 process, but does not represent a completed standard nor the official positions of the host organization for standardization which is ISO/IEC JTC1/SC29/WG11.

2 Applications & Requirements

Let us start with some applications that SNHC might serve, with a long view in mind. Then we narrow the focus to a few applications that offer essential requirements for early steps in standardization. These applications are engaging or "immersive." They may even task-load the user in some way. They could involve combinations of real-time streaming and downloaded media, from a network or local disk. These applications rely on streaming audio/video media and synthetic graphics/audio, where effective presentations are best realized with hybrid content, rather than relying on natural or synthetic content alone.

2.1 Sample Applications

Application areas which SNHC might address include:

- Facial animation & correlated speech
  - Facial agents in automated tellers, kiosks, PCs
- Interpersonal & media conferencing
  - Virtual tele-conferencing with multimedia
  - Collaborative work involving 2D/3D design
- Multimedia education & entertainment
  - Knowledge navigation with hybrid media
  - Animated story telling with interaction
  - Distance learning, collaborative tele-teaching
  - Medical (distance diagnostics, surgical training)
- Multimedia presentations with interaction
  - Product, design & service demonstrations

  - Corporate communications & promotion
  - Tele-shopping with 3D models, images, sound
  - Virtual travel agency, real estate promotion
- Digital A/V desktop & networked media workshare
- Virtual studio/set with networked media integration
- Gaming & training with hybrid media
  - Networked virtual environments for teamwork
  - Multi-user simulation & gaming environments
  - Computer-Based Training with A/V + 2D/3D

What are some more specific examples? A CD-ROM encyclopedia shows audio/video clips to a user from an over-flight of a world wonder with synchronized scrolling explanatory notes, an overlay marker or pointer by a naturalist, a scrolling map, a 2D inset showing animated 3D models, and other overlay graphics which come and go in reaction to user selections from a side-bar menu. A student navigates a learning tree in a mixed-media educational title where real-time A/V streams are displayed as panes or sprites synchronized with animated annotations and information graphics. A line or field worker controls a cellular video display linked to a remote "3D manual" in a server which encodes animated views of an interactive work-piece library. An instructor coordinates networked classrooms in a shared multimedia learning process, where natural audio and imagery are combined with synthetic models, content pointers, animation controls, and symbolic representations so students can experience the knowledge. A desk-top hybrid media display supports the home, school, or shop floor, where a user can learn about a layout or product, experience a virtual science lab, or undergo industrial training (process, maintenance, vocational). A desk-top video editing system produces video and audio “objects” with compositing structures, scripts, and "attached" work-in-progress notes that let networked coworkers re-edit and add to the work in subsequent steps of content production. A 3D video game for networked players utilizes clever blends of audio/video, animated 2D/3D sprites, and frugal 3D graphics to challenge users on low-cost mixed-media PCs.

A few anchor applications for SNHC can help abstract essential requirements and develop concrete initial results. Anchor applications for SNHC are audio/video with 2D graphics and interpersonal communication. These two applications provide focus for initial functionality. The first application involves dynamic overlay graphics which are subordinated to audio/video objects to help augment, annotate and explain them. The second is facial/body animation combined with text-to-speech synthesis including initial downloads and streaming data for efficient synchronized speech. These anchors can be extended to other areas like mixed-media conferencing or tele-teaching, or 3D networked virtual environments, at subsequent steps. Overlay graphics were exemplified earlier, while facial animation should be explained.

Facial animation has many uses. An automated teller or kiosk presents animated 2D or 3D polygons with texture, or some video segments guided by branching logic, of a facial agent. The facial agent explains to and asks things of the user in logical connection with a user interface, which is represented on and off by pop-up dynamic overlay graphics. The agent points at and activates the user interface if useful. A therapy device helps a hearing-impaired patient by playing an aural/visual facial agent (as video or polygons) with interaction and rapid rewind. This is accompanied by synchronized text, sign language, or icon-picture-video pop-ups that visualize in easily recognizable forms what is being described by speech articulation of the mouth.

2.2 Requirements Overview

A bottom-up approach has been chosen for MPEG-4 SNHC, focusing in the early phases on two particular applications as a basis for gradual extensions leading to a framework that will be able to cover a broad range of applications. These ultimately involve interactions with audio, video and 2D/3D objects, as well as human representations within 2D/3D environments.

Three levels of extensions are defined. The first level of extension is concerned with the coding of facial and human body animation combined with text-to-speech synthesis. A second level provides media integration of audio/video objects with 2D graphics. The third level integrates 3D synthetic objects, behaviors and interaction in a multi-user 3D virtual environment. At each level of extension, the application focus, requirements and expected standards are specified. Aspects of the first and second level are currently being pursued to support facial/body animation with speech and graphics overlay for audio/video. The next section discusses general requirements for SNHC. The remaining sections deal with each of the extension levels.

2.3 SNHC Object Transmission Protocols

Coding of A/V objects will support more traditional streaming of audio and video with time coding as discussed before, and also the downloading of synthetic models and streaming update of their animation parameters. The coding of spatial and temporal relationships between the downloaded objects and other streaming objects will be supplied with download data, so that media composition is well-defined in the terminal as streamed objects animate jointly with their synthetic partners. Downloading itself includes no temporal synchronization, but the caching of synthetics in the terminal so that all objects are available to initiate a real-time session. A downloaded object will have provision for time coding of animation, which is linked to a streaming object or which otherwise runs in local terminal time during a session. Update streams that can be supplied by a server could include geometry streams (e.g. animated motion or deformation of synthetic model elements). Facial animation can be achieved this way.

2.4 Facial/Body Animation with Speech

We provide an overview of requirements for facial/body animation with speech. In this step, SNHC concentrates on defining a set of parameters to describe facial expressions and body movement with synchronized speech. The decoder could include a 3D articulated facial model to reproduce facial expressions animated by remote parameter streams. The size of the parameter set after compression can be less than 1 Kbit/s, allowing animation at extremely low bitrates. An application is low-bitrate interpersonal conferencing, using image processing to extract the expression parameters from each frame of a talking head video. These parameters then animate a facial model in the receiver after their decoding. Other approaches might include generating lip movements from text or coded speech. The synchronization of audio or speech text with geometry streaming for facial model deformations is stringent (10-20 ms), since audio/visual correlation aids speech intelligibility.

2.4.1 Facial/Body Animation

A model-based approach to the transmission of facial/body information opens breakthroughs both in compression and functionalities. With the small size of the expression and animation parameter set, compression ratios, that are one to two orders of magnitude better than using classical video compression, can be achieved without losing crucial content (e.g. the facial expressions and lip movements). Moreover, achieving high-level functionalities (like movement or scaling) is straightforward since the 3D models of the face and body are known. Stereoscopic visualization of a talking head or body model is also achievable without any bitrate overhead. Body animation leverages all the same principles as facial animation for coding body models and their compressed parametric control streams during a session. Specific spatial models or scripts of behavior for faces and bodies are not standardized, so that application developers can add finesse in the artistic and natural aspects of content expression. A specific approach to modeling these might be accomplished with VRML, Java Media or ActiveX Animation.

The negotiation phase for facial animation establishes the fidelity level for a download or a client-resident face model. It provides set-up of the session including any facial texture and shape attributes as well as tools for transforming a parameter stream into control of a specific face model. Then parameter streaming commences with text or speech audio transmission. Thus a downloaded 2D or 3D model must be sent with very low error rates, so that the model syntax which drives the animation and rendering engines beyond the receiver's decoder is essentially uncorrupted. 3D model data including vertex meshes, transforms, normals, texture maps (if supplied), level-of-detail alternatives for scaling, etc. are all candidates for compression during transmission and decompression into run-ready formats in receiver caches. The parameter sets defining model behavior, their animation during a session, and their compression are standardized.

2.4.2 Text-to-Speech Synthesis & Spatial Audio

SNHC will provide the means to download speech segments represented by spectral or waveform sequences which are used as synthesis units to produce synthetic speech. For instance, given a string of a text with some prosodic parameters such as the F0 (or pitch) contour of the speech, then a decoder with Hybrid Scalable Text-To-Speech (HSTTS) synthesis would concatenate proper synthesis units and synthesize corresponding speech. This HSTTS could be driven by live text streams in the target language with language-focused decoder algorithms.

Hybrid Scalable Text-To-Speech coding can be synchronized with synthetic models. HSTTS coding combines conventional text-to-speech with prosodic parameters that lend more natural intonation (pitch/inflection, volume, duration) to speech reconstruction. SNHC aims to provide a framework for downloaded initiation and streaming control of prosodic-augmented speech synthesis in the terminal. This will be capable of coding prosodic-augmented speech as an audio object. This capability will also embody Text-to-Speech (TTS), which is on the verge of becoming a standard interface, and plays an important role in multimedia technology. Thus narration of multimedia content can be composed without recording speech. This coding standard will be scalable, and will bridge between TTS and natural-sounding speech (e.g. HSTTS). Furthermore, an interface between TTS and facial animation systems will be provided to allow synthetic face models to be driven from speech [32].

Another key objective of SNHC is to provide spatial-temporal integration of audio objects such that synthetic audio coding, and TTS or HSTTS coding, as well as traditional signal coding of

natural and speech audio, can be used together to synthesize aural environments. The requirement is to provide sufficient coded information so that spatial audio reconstruction is possible with special effects that depend on the aural environment and movement of A/V objects. The terminal's decoder and audio rendering resources will then determine the level of fidelity and functionality of audio synthesis supported by a specific platform. The result should be that a terminal with appropriate resources can support spatial-temporal composition of synthetic audio sources (like MIDI), virtual 2D/3D sound emitters, and speech synthesis.

2.5 Media Integration of Audio, Video & Text/Graphics

This capability step is aimed at the initial integration of MPEG-4 Video and Audio Object coding with 2D graphics. Such integration can support the compositing and synchronization of audio/video objects with overlay graphics discussed earlier, the integration of facial animation with time-synchronized multimedia animations (to support tele-teaching, distance learning), or the animation of characters as 2D/3D sprites in layered 2D environments. Figure 1 shows a schematic representation of the audio/video with 2D graphics overlay described earlier.

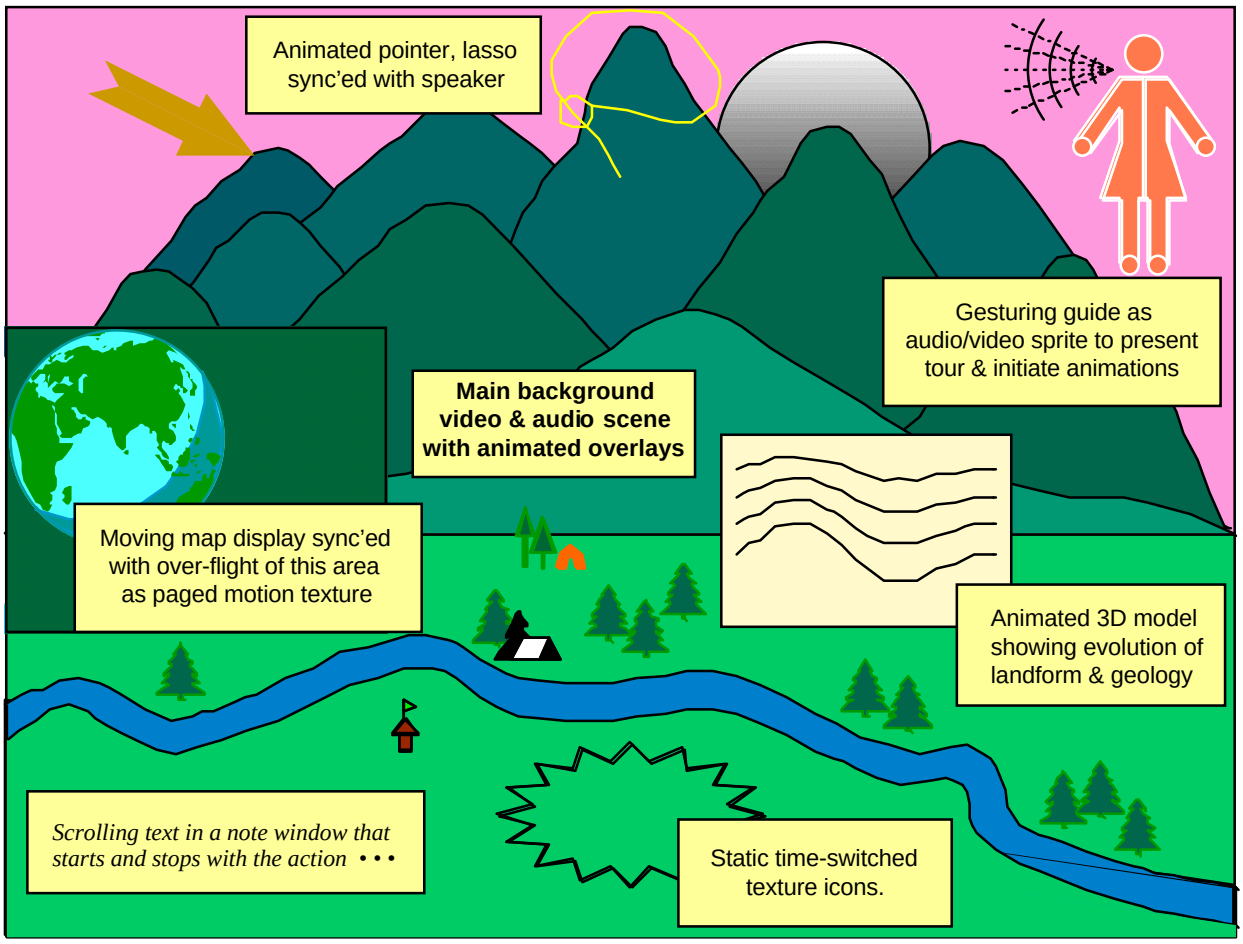

Figure 1. Tour Scene Audio/Video with Graphic Overlays

We may wish to mix streaming and downloaded content from remote servers and the local client platform, such as an audio/video stream from local CD-ROM mapped synchronously onto a downloaded 3D object. An example is video of a face mapped onto its 3D model as animated texture, like a mannequin acting as a screen for a facial movie, or billboard video playing in 3D on a movie or TV screen. Time coding of audio, video, embedded graphics, and text should allow a narrating guide (represented by a sprite or facial animation) with speech and gesture to be synchronized with multimedia playing elsewhere in the presentation. Figure 1 shows an example of a 3D sprite, that is, a gesturing and turning (3D) person captured as (2D) video with synchronized speech audio, and then coded as animated texture with an alpha channel that delineates the boundary of the sprite against its background. In this way, scrolling text, audio/video, animated pointers, flashing icons, and speaker animation can be depicted with natural, synchronous rhythms and events during presentation.

The animation of text and 2D graphics is expected to call on well-established standards in analytic fonts and geometric constructs typical of paint or drawing tools used in desk-top publishing or Web page animation. ITU-T T.126 provides an example of a standard for overlays on still images. Text should be capable of varied fonts, styles, special effects, color, scrolling modes, transparency, streaming, and time synchronization with other A/V objects. 2D graphical objects should include points, open and closed polygonal outlines, rectangles, and ellipses, with attributes and transformation parameters such as rotation, scaling, bounding lines and styles, pen characteristics, highlighting, fill color, texture, and depth ordering.

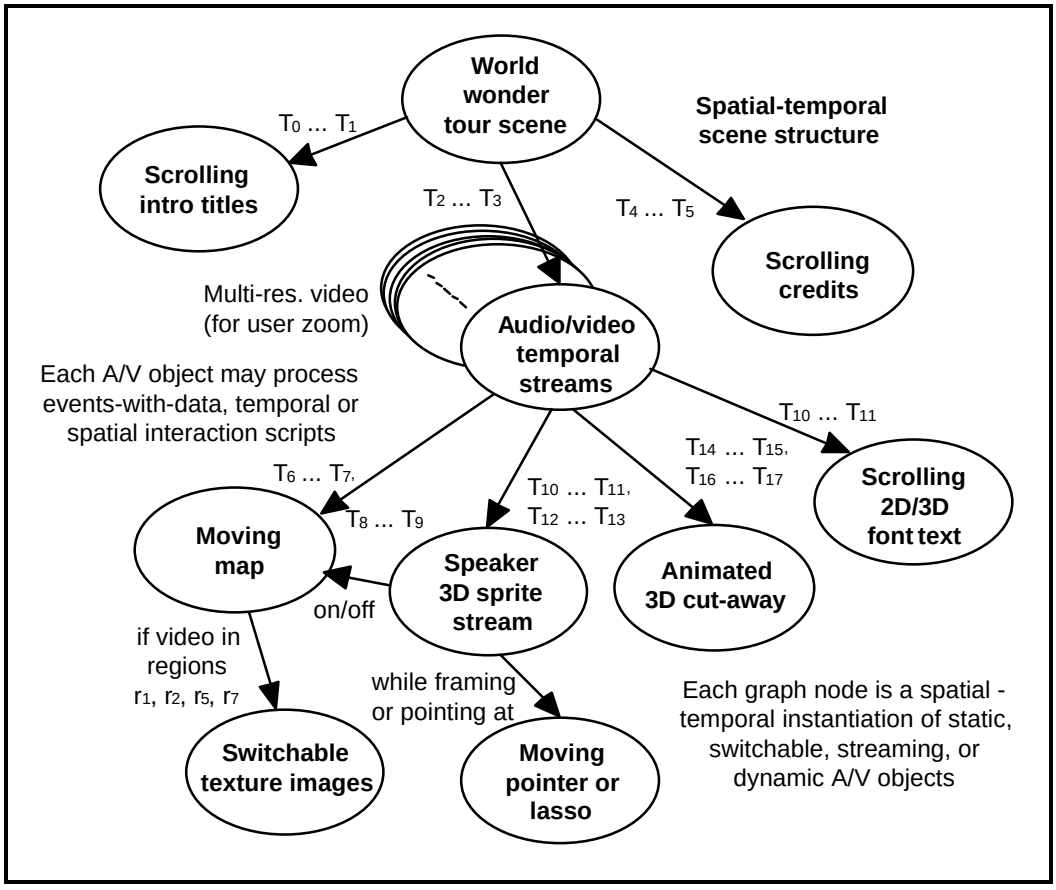

Figure 2. World Wonder Tour Scene Composition

Figure 2 shows a schematic representation of the spatial-temporal scene composition for the example in Figure 1 of audio/video with overlay graphics. Note that the scene composition for this application allows the specification of spatial and time-based dependencies which control where and when specific A/V objects are activated. Consequently, 2D graphical objects exhibit spatial and temporal composition and control with respect to the dominant audio/video scene. Also note that if facial animation with speech is combined with this application, then a real-time stream of facial control and speech parameters would join the audio/video stream as multiplexed A/V data (subject to the realities of channel bandwidth and compression ratios achieved).

2.6 3D Virtual Environments with Multimedia

After MPEG-4 baseline capabilities are established, the ultimate goal of SNHC covers a broad range of applications. This level will allow multiple users to share a 3D Virtual Environment through the network, communicate through speech and vision, and interact with the multimedia content of the environment. The users become a part of the virtual environment, represented by animated avatars consisting of virtual faces and bodies. This allows each user to see other users in the environment, their actions, facial expressions, and their focus of attention. Objects of various types can be integrated within such a Virtual Environment:

- 3D synthetic objects, where the user can introduce new objects into the scene, move or deform them interactively. The objects can have behaviors defined by scripts.

- Video data can be introduced into the 3D environment by mapping the video as texture on 3D objects. The interaction and behavior capabilities of the video-mapped object are preserved, implicitly allowing the manipulation of the video.

- The audio data is introduced as 3D sound sources exhibiting the same interaction and behavior capabilities as other 3D objects.

At this level, MPEG-4 should support scaling of texture to fit bandwidth and decoder rendering capabilities established during the negotiation phase. Texture scaling might be achieved by JPEG compression, sub-band coding, down-sampling, or interpolation of MIPmap image pyramids. Video scalability should also support multi-resolution video under a range of 3D viewing conditions. MPEG-4 should also support progressive downloading of 3D models and texture to display an early version of an incoming scene for low latency or to sustain higher animation rates, following which the user may access higher fidelity versions of the same virtual environment.

The initial standardization of facial/body animation could provide the ground work for more general composition of animated 3D virtual environments that integrate multimedia objects. At this level, MPEG-4 could then support transmission of scenes composed of texture, general 2D/3D models and attributes (lights, ...). An example could be a synthetic human face with lights and texture map in a virtual meeting room. MPEG-4 should here support scaling with level-of-detail for downloaded information, based on scene loading analysis or on feedback from the terminal to server about viewer position, speed of objects, etc. Examples are coarse transmission of an object at the edge of the field of view, paging of objects based on spatial or temporal locality, or thinning of scene content based on priorities specified by the user.

With this last step, MPEG-4 could support interaction and simulation in virtual environments. Examples include rotation around a 3D model of a face (local interaction), change of an object's position in a virtual room shared with other users (remote interaction), collision with or acquisition of moveable objects, or a stereo Doppler effect for an incoming audio object. Capability could exist to support behavior of objects linked to events with data, such as a user action, the satisfaction of a spatial or temporal condition, or a fixed sequence such as a rotating model of the earth. MPEG-4 will support transmission of scripts defining temporal control of a downloaded object, as well as transmission of state parameters between users to permit user interaction and event detection.

Various modeling schema for spatial and temporal media content are embodied in current work such as VRML 2.0, Java Media 2D/3D, and ActiveX Animation. These promise to provide inspiration or direct standards infusion to support the third level of MPEG-4. It is crucial however that such media toolkits provide efficient manipulation of the underlying objects through a media API, so that MPEG-4 can achieve scalability control and spatial-temporal integration of media types with real-time performance where needed and feasible.

3 System Model & Functionality

In this section, we provide a functional overview of an MPEG-4 system with SNHC services in mind. MPEG-4 is concerned with coding of animated data, and thus with spatial-temporal relationships among A/V objects as represented in bitstreams. The requirements of MPEG-4 are sufficiently complex that bitstreams and the higher-level representations they encode should not be designed in isolation from the application environment. It is generally not the domain of MPEG-4 to design a new multimedia modeling or rendering standard, or another animation language. MPEG-4 is intended to provide coding, decompression, multiplexing, synchronization, download and streaming, and composition sufficient to help application developers animate media which can exploit established modeling and rendering standards.

3.1 MPEG-4 Framework

The MPEG-4 System and Description Languages (MSDL) provide a flexible, software-based, dynamically configurable, run-time environment for invoking decompression tools, algorithms, and profiles. Within this environment, MSDL will include a library of A/V object types designed with the foregoing requirements in mind. MSDL defines a class library for media objects, their bitstream description, and their method calls. MSDL supports the building of run-time decoders, algorithms, and related real-time operating profiles in the receiver. MSDL provides for a start-up negotiation phase between server and client to download A/V object types for a specific session along with the method calls to interact with or render them. Then MSDL supports both streams and downloads that instantiate these object types during a session.

MPEG-4 A/V objects will not demand the modeling detail or range of data modalities of objects used in engineering design. A/V objects should be sufficiently expressive to depict surface appearance, essential shape, texture, spatial and motion relationships, and behavior suited for media experiences that are useful to accomplish a task. MPEG-4 object definitions could for example overlap subsets of VRML, ActiveX Animation, Java Media, or other media libraries.

The MSDL architecture group is considering how to integrate other standards, and to ascertain the best fit for MPEG-4 requirements. MSDL is not in itself a graphics API for a rendering engine, but MSDL is capable of building run-time environments that invoke such renderers.

3.2 System Functional Block Diagram

Figure 3 shows the functional architecture and relationships between the major anticipated processes within an MPEG-4 receiver. This includes some optional upload functions and data paths that could be configured to make a transcoder. MPEG-4 constructs are distributed in the left-central portion of the diagram, while other terminal services would typically be provided by browsers, animation engines, rendering/compositing APIs, and windowing environments as supplied. These resources constitute the terminal application and presentation layers.

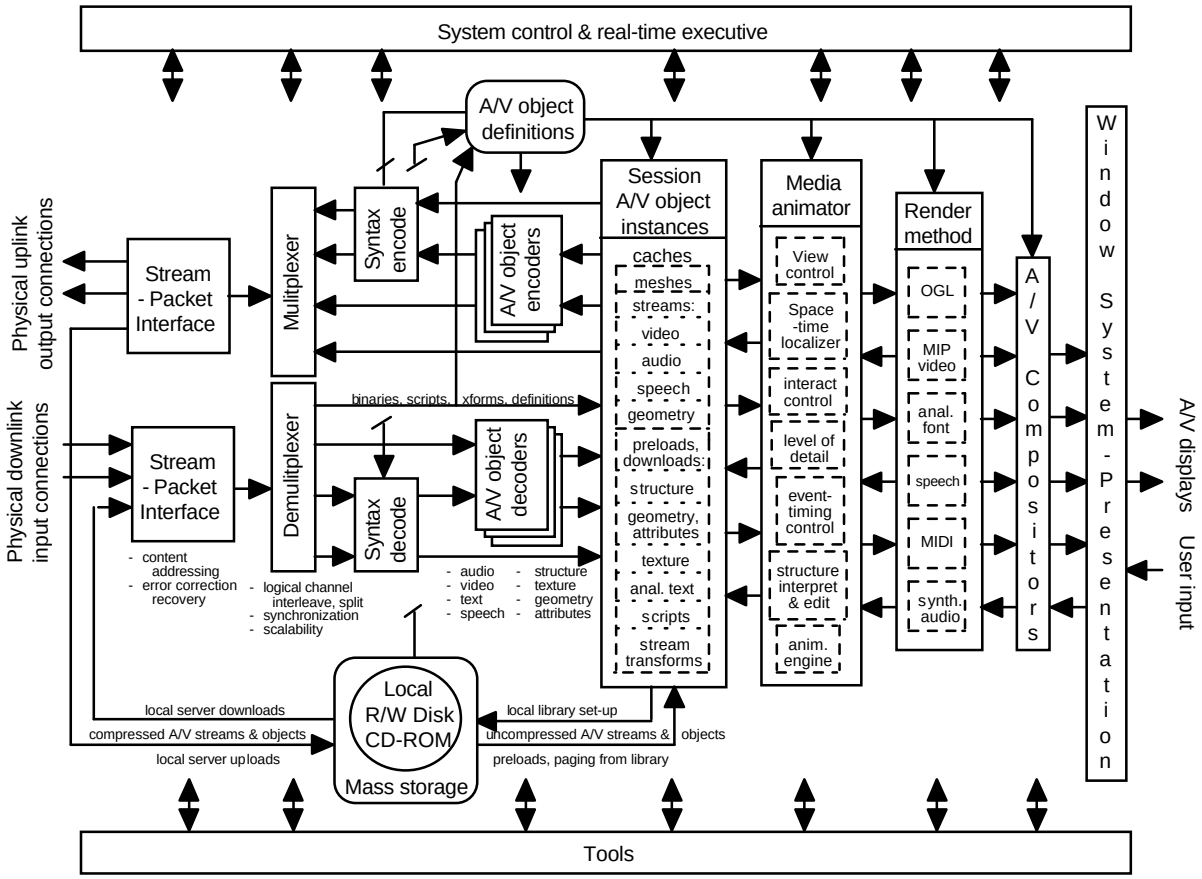

Figure 3. MPEG-4 Receiver Functional Block Diagram

The packet interfaces and mux/demux stages, supported by the system control layer, allow the bitstream(s) to be parsed for content addressing and error correction. Interleaved logical channels are segregated here, including the demultiplexing of progressive transmissions, level-of-detail alternatives, or audio/imagery layers for multi-resolution scalability. Streaming objects are decoded synchronously with buffering for rate leveling if needed. Downloaded objects that persist for a portion of a session in local caches are downloaded asynchronously, including possible decoding of compressed geometry. A/V objects are synchronized throughout the

terminal pipeline, in step with buffer constraints and the space-time alignments implied by object time tags, the session script, or the temporal composition of animated objects.

The data caches in mid-diagram hold instances of A/V objects for a specific session. These caches hold the set of A/V objects which are "local" to the user's current spatial-temporal relationship with the A/V data. Generally, the contents of these caches are paged from network or local disk to meet this locality criteria on a continuing basis. These caches may degenerate to decoder buffers for simpler receivers. In more complex situations, these caches could hold a downloaded scene graph for a portion of the A/V experience, or a decompressed "flip book" for animated texture that does not consume decoder resources. Cache storage also holds persistent objects such as 3D geometry, textures, audio clips and segmentation masks for video processing.

An MPEG-4 receiver is client-centric in the sense that the terminal constantly obtains data to satisfy user locality, and creates the demand for services and caching while queuing remote and local sources to meet receiver needs. A compressed movie is a simple example where client and server are in lock-step timing once the channel is acquired and started. In similar fashion, a transform stream or geometry stream could be decoded to animate a local articulated static 3D model by remote control. In more complex applications, the media animator block determines streaming and downloading needed moment by moment in response to temporal and spatial events detected in a session, including the side effects of user interaction. The media animator also controls the rendering of graphical objects and the compositing of video and audio objects.

3.3 Downloaded vs. Transient A/V Objects

A concept with MPEG-4 is that real-time A/V experiences at the user' s display can be supported with streaming and downloaded data. Animation can be achieved by transmitting time-stamped samples of behavior, or by downloading the behavior as code and then letting the terminal sample the behavior, or conceivably both. Streams can be collected by the terminal from the network with one or more open logical or physical channels, or from local disk that can deliver compressed A/V streams on local busses. Behaviors (as scripts) require very little network bandwidth once downloaded, but require an adequate animation engine in the terminal.

Another distinction with MPEG-4 hybrid coding is the flexibility to vary where a synthetic model, or a streaming object, resides during a session. Synthetic 2D/3D models can be rendered by servers into encoded streams and transmitted as compressed audio/video. They can also be downloaded for large subsequent network bandwidth savings in trade for 2D/3D rendering and compositing resources in the terminal. We have a server-terminal trade-off again, as well as a trade-off in latency during interaction depending on whether the server or terminal acts as host for the reactive behavior of interactive A/V objects. Server-based rendering, and the growth of 2D/3D renderers within terminals, will offer the option to work different ways. Figure 4 shows a sample of an MPEG-4 communication that combines interaction with multiplexing of streaming and downloaded A/V objects.

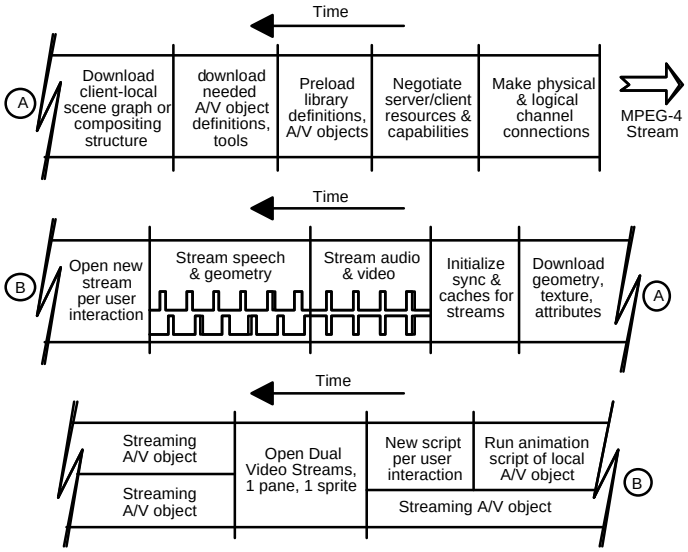

Figure 4. MPEG-4 SNHC Communication Stream

A by-product of such alternative partitioning is the prospect of exploiting temporal or spatial coherence over a network or bus. This might be achieved by connecting a server, capable of multiplexing streams and downloads, with a terminal capable of presenting streams, rendering downloads, or interpolating between animation states of a synthetic model. A terminal could also do predictive rendering of a 2D or 3D synthetic model during intervals when the server does not otherwise provide updates. MPEG-4 coding could facilitate these modes of operation. The application software at the server and client terminal then makes the difference. Compressed A/V streams could also be downloaded to a local disk cache, and then routed into the local MPEG-4 decoder with little or no network loading during the session.

3.4 Media Standards & Technology Stability

Standards can be defined when large communities are ready to communicate in standard ways regardless of the growing diversity of needs and approaches. To maintain pace in current A/V developments, MPEG-4 aims to pick key base abstractions as a foundation, and then specialize the abstractions as applications become clearer or as component technologies mature. Component technology for audio, video, text, 2D information graphics, and 3D graphics is fairly mature today. The modeling of animation, physical action, events, condition sensing, decision-making, and reactions - behavior - is very diverse now, and not so unified or mature within the animation communities. Yet real-time animation is a goal of many MPEG-4 applications.

Many industry sources contribute to animation technology using different mathematical models. Several prominent scripting languages have appeared in the last decade to support near- and real-time animation and simulation. Java combined with VRML 2.0 offers an open, potentially uninhibited framework for behavior programming that resides within a security shell to protect networks. Together they offer a general language for describing behavior with low-level services for linking to and manipulating otherwise static scene structures. ActiveX Animation (once ActiveVRML) imposes discipline on behavior programming through a functional language that models behavior and control of interaction with integrated spatial-temporal hierarchies.

At the boundaries in a network or computing topology, where action must be described and synchronized, this body of techniques relies on streaming time-synchronous data from a server, or on downloading behaviors to initiate their action in a terminal. Achieving temporal scalability requires that either basic communication model code time as a continuum, and then compensate for media complexity, latencies, frame rates, and other constraints in the terminal. This leads to designing MPEG-4 with basic static and sequenced A/V object types, augmented by higher-level scene composition and animation constructs that ride on top. Wherever practical, MPEG-4 will tap A/V object types capable of real-time that have coalesced in the audio, speech, video, text, 2D, and 3D graphics communities. Beyond these services, MPEG-4 should remain open to the thriving experimentation and gradual maturity of behavior programming.

3.5 Synthetic vs. Natural Content

Should a distinction be made between digital models of natural objects and synthetic ones? Contemporary examples argue against such distinctions, although specific coding techniques support natural or synthetic imagery and models better. There is widespread development of 2D/3D synthetic models representing real objects. Photos and video reveal a blurring authenticity about whether content was recorded directly or generated synthetically. Movie content merges real-world and synthetic images that are often unnoticed by audiences. Image processing and feature extraction techniques applied to digital imagery extract the underlying feature abstractions in scenes. Efforts to build complex 3D synthetic environments for movies, gaming, training simulation, etc. often spin-off 2D media derivatives from computer renderings of 3D models. Natural scenery in digital images becomes "photo wallpaper" to decorate the facets of 2D and 3D synthetic models. SNHC should integrate these media types, and make no real distinction about where media components originate, except to ensure quality and bitrates.

3.6 Hybrid Model Complexity

What limits the complexity of SNHC objects? Composing aggregates of audio, video and 2D/3D objects can potentially exceed the fixed content coding of MPEG-1/2. The demultiplexing and decoding buffers, processors, and memory of client terminals, and the quality and bandwidth of network connections, will limit a session or the terminal's ability to absorb downloaded data. Content consisting of a downloaded hierarchy of A/V objects in a terminal may exceed real-time rendering constraints as a function of model complexity and viewing conditions (also different from earlier MPEG). MPEG-4 codes A/V objects in such a way that the negotiation phase for a session, or the pruning of scalable object complexity at run time, can adapt media complexity to suit the terminal. Thus the coding efficiency and the downloading or streaming rates of A/V objects during delivery may be guaranteed. Real-time performance still depends on terminal resources, the specific content, and whether the developer provided for scalability.

MPEG-4 A/V objects or collections of them may thus satisfy a network constraint, and still outpace the terminal's resources. This is different than MPEG-1/2, where certification of compliance involves a verification model for which the network and decoder have well-specified limits on audio/video bandwidth and media complexity that are coded in the stream. The performance of MPEG-4 in such situations should be measured by channel and decoder performance, and not by a terminal's specific animation or rendering limitations. A content

developer must think about temporal, spatial, or object scalability, using the coding features within MPEG-4 to match scalable content to a defined range of terminal resources.

3.7 Multi-Resolution Scalable Imagery & Video

SNHC environments will include imagery and video decoded into static or animated textures applied to 2D or 3D surfaces or planes. Static, animated, and motion textures are used in gaming, facial agent animation, VRML, simulators, etc. to "wallpaper" underlying polygons for photo-realism or artistic expression. Standard MPEG-4 image and video coding should consider the provision of motion and animated texture, progressive transmission, and image pyramid structures. This offers the option to decode discrete scales of texture for manipulation in 2D/3D scenes. Renderers can then support continuous display scalability of texture.

Coding of imagery for insertion in scenes of 2D/3D graphics under widely varied viewing conditions can place special demands on image coding. Texture is compressed by JPEG and MPEG-1/2 quite effectively for coding given resolutions of natural imagery with lossy and lossless options, though scalability is lower than with wavelet coding. Texture compression and decompression, in the mode of JPEG or MPEG-1/2, within database downloads or streams can clearly contribute to network and local bus efficiency.

MPEG-4 Video Objects can be used for variable priority in compositing, video level-of-detail or resolution scalability, temporal scalability, and the ability to multiplex multiple resolution-related video streams in a physical channel within bandwidth and buffer limitations. These video primitives and their spatial-temporal composition can be invoked without synthetic 2D/3D graphics when the terminal compositor handles only video. Depending on capabilities of a specific terminal rendering and windowing environment, this can provide for video zoom and slow-fast control when video pixels map to display pixels in a fairly direct fashion.

Applications may put video in dynamic perspective within a 3D environment, compose a variable picture-within-picture, or zoom-warp continuously in 2D. Media processors and 3D graphics coming in PCs and game consoles over the next few years will make this increasingly possible. This implies the display of video in multimedia and virtual environments where viewing conditions are not confined to predictable relationships between video and display pixels (e.g. fixed-scale video panes). In such cases, rendering systems will attempt to minimize resampling of decoded video and to exploit static or dynamic texture rendering achievable (e.g. with OpenGL or Direct 3D). Rendering of multi-resolution video could access neighboring pairs of resolution for a video stream in an image resolution pyramid, for resampling texture with view-dependent filters during pixel sampling of texture.

MIP mapping (Multum In Parvo) is one example [9] of modeling texture under popular 2D/3D rendering APIs in a ready-to-render image pyramid for bi-linear or tri-linear interpolation in display space. Such texture can be viewed under varied scales, warpings, and perspectives including anisotropic texture in display space. Direct decoding of textures into MIP maps could support their application to 2D/3D sprites and polygons for viewing under varied conditions while preserving image quality. Animated image pyramids obtained during video decoding could help video scalability with image quality in similar fashion. Simulation visuals with textures and videos

inserted in 3D, and ActiveX Animation of picture-within-picture, show the potential importance of scalable static imagery and video as multi-resolution texture.

MPEG-4 is examining how Video Objects can be used to animate texture on 2D/3D polygons. MPEG-4 offers a hierarchical Video Object composition that might be used effectively for spatial-temporal scalability within scene graphs of synthetic environments. MPEG-4 Video Objects might also be used to code moving synthetic imagery (e.g. video of moving models approximating real-world objects, moving information like animated font text or line drawings) when a terminal does not render synthetic content directly. Other issues include representing color spaces, preserving monotonicity in synthetic shading, blocking artifacts, the control video warping effects that are not specifically for compression, etc.

3.8 Naming, Servers, CD-ROM libraries

MPEG-4 use of hierarchical scene composition opens the possibility of making name references to A/V objects that reside in different places to optimize how network and local memory access are balanced. These could include local ROMs or hard disks, removable CD-ROM libraries, servers remote from the terminal, or network "neighborhoods" corresponding to multi-user caches that temporarily store A/V objects frequently accessed by a set of users. These alternatives can help to maintain data rates for hybrid scenes that depend on downloads and streams of A/V objects. SNHC scene compositions could integrate application-specific local and remote A/V objects, where the system layer can search to optimize local access when possible.

4 Geometry Compression

As the capabilities of today's PCs increase, the use of computer graphics becomes more viable for communication applications using a phone line, local area network, or the Internet. For a satisfying user experience in a communication environment, two main conditions should be satisfied:

1. Downloading of computer graphics models, like virtual worlds with their illumination and animation parameters, into the graphics engine must be fast enough. Here the current bottleneck is the slowest communication link between the user display terminal and the server. Most likely this will be the LAN or the link over the Internet.

2. Computer graphics models have to be animated at a high frame rate to support the smooth appearance of motion. Here, the bottleneck is the graphics hardware of the terminal or PC.

The latter condition will be fulfilled by $300 PC graphics boards, announced for late 1996, implementing the OpenGL API, supporting 300K texture-mapped triangles per second. The first condition calls for higher communication bandwidth or compression of computer models. The increase in reliable and affordable networking bandwidth requires change in industry infrastructures that will be a slow, time-consuming process. Computer graphics models are easily larger than several megabytes. Transmission of compressed models and decompression at the terminal will be the enabling technology for interactive use of graphics in communication applications. This is analogous to compression enabling the use of interactive video.

The following section describes basic requirements and basic principles for compression of 3D texture-mapped graphics models, including geometry and texture compression. A subsequent section covers additional requirements such as scalability related to animation.

4.1 Compression of 3D Graphics Models

For the time being, MPEG-4 SNHC assumes that static models consist of illumination, geometry described by polygons, surface properties described by parameters like roughness, shininess, color, and texture. Typically, a scene graph similar to the concept followed in OpenInventor or VRML can be used to describe the model. As far as the amount of data required for describing the model is concerned, the geometry and texture take the largest part. Therefore, the following paragraphs focus on geometry and texture-map compression.

4.1.1 Geometry Compression

For polygon models, a vertex of a polygon is defined in a 3D object coordinate system by means of a 3D vector using three floating point numbers. Each vertex of a model is indexed by a number assigned as its unique identifier. Then, a polygon is defined by a list of such vertex identifiers. Figure 5 shows a triangle within a larger polygonal mesh. Assuming that each vertex is shared by six polygons on average, with a maximum of 1 million vertices/model, the number of bits for describing this triangle would be:

1/6*3vertices/triangle*32bits/vertex+3vertex_ids/triangle*20bits/vertex_id = 76 bits/triangle

This number does not include bits for normals or surface properties. It just indicates that a reasonable size polygon mesh can easily require several hundred Kbits for the basic geometry.

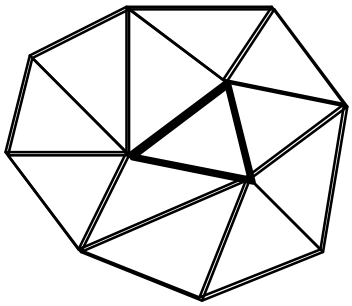

Figure 5. Part of an Irregular Polygon Mesh

The need for geometry compression has recently been recognized by several authors. For lossy encoding of the basic geometry, a spanning tree for vertex positions and triangles defining the polygon mesh is proposed [4]. Vertex positions and normals are quantized, and Huffman encoded. The authors report that a triangle can be described by roughly 2 bits/triangle for vertex positions, structure, normals, and colors. The application of this geometry compression technique to a compressed binary format for VRML 2.0 demonstrates compression ratios of 34-39:1 for anatomical and animal models with fairly high triangle counts, and 65:1 for a mechanical part with very high structural coherence and regularity in the triangular surface mesh.

A different method also using quantization of vertex positions, differential entropy encoding of vertex positions, as well as a special table lookup method for surface normals, reports compression ratios of 6-10:1 while giving only slight losses in object quality [5]. A similar approach is proposed in [33] that describes a near-lossless compression of VRML 1.0 files. On the SNHC Test Data Set used for comparative proposal evaluations within MPEG-4, the algorithm achieved a compression ratio between 15:1 and 63:1. This is achieved by identifying bounding boxes for coordinates, subdividing 3D space into octants (spatial cubes), quantizing coordinates defined by octants, and quantizing floating point numbers for materials and normals to a given precision. Code words are built to allow for an optimal entropy encoding of the data.

4.1.2 Texture Map Compression

Two types of texture maps can be distinguished. Artificial textures are used in repetitive fashion to generate large texture maps like brick walls. Artificial textures can be described by a function generating the values of the texture map or by an image. Natural textures are specific images of natural content used for mapping onto an object surface. In case the texture is transmitted as an image, standard image coding techniques can be applied. If the image is static, JPEG is a method of choice, and is employed in VRML 1.0.

If moving images are used as a sequenced texture map, image sequence coding algorithms like MPEG-1 or H.263 are able to achieve higher compression than the still image codecs. Please note that standardized compression algorithms for textures are available today, whereas there is not yet a standard for geometry compression. VRML 2.0 incorporates MPEG-1 Systems and MPEG-1 Video for movie texture. A general method for animating texture on an object surface could emerge in MPEG-4 including alpha-coding of video to animate and compress a sprite.

4.2 Display & Animation of Large 3D Graphics Models

If large 3D models are to be displayed and animated by the user or by a downloaded script, it is often not feasible to render the entire model if only parts of it are visible, or if details will not be visible due to the display resolution. Using the concept of level of detail in computer graphics or scalability in audio and video coding, rendering of large models becomes feasible on today's PCs.

This level-of-detail concept allows for scene complexity management to render the object at different spatial resolutions depending on the displayed object size. Thus several small polygons are replaced by a larger faster-to-render polygon if visible differences are not distracting in the scene rendering. Similarly, texture maps of different resolutions can be used to limit the amount of filtering for the image synthesis, to avoid aliasing problems, and to speed up access to a first view of the object, as with MIP map texture [9].

This concept of scalability also allows a terminal to display a rough representation of the model as it is downloaded. Thus we download a house, where the outer shell could be downloaded first. Then, while the user is already able to see the exterior, the interior is updated. For communications, the compression algorithms for geometry and texture maps must inherently support this scalability. Progressive geometry has been described [3] which can provide incremental level-of-detail enhancement for a downloading object, as with progressive texture

transmission, where the user benefits from early renderings of arriving objects without waiting for full detail of the object to accumulate. However, to date, this area is still in its infancy.

As far as animation is concerned, MPEG-4 is looking for a standardized method of describing camera/viewpoint animation in order to allow interactive and guided walk-throughs. Furthermore, methods are required which allow display of moving virtual world objects, i.e. the ability to attach scripts to objects is sought.

4.3 What to Standardize

For geometry compression, several proposals already show that the amount of coded data required for representing large polygon models can be reduced by a factor greater than 6:1. Other issues important for communications applications, like scalability and animation, still require intensive investigation. SNHC is pursuing geometry compression, so that model downloads and progressive geometry updates over a network can be greatly economized. The target for standardization is the decoded representation of synthetic objects, and the corresponding bitstream which encodes the model's geometry, structure, attributes, normals, etc. The Compressed Binary Format proposed for VRML 2.0 exemplifies such a standard for efficient 3D model transmission, and offers large gains in compression ratios compared with simple ASCII compression of VRML model syntax. Such a standard could be invoked by MSDL assuming VRML is adopted by MPEG-4. In addition, specific geometry compression tools could be incorporated into an MPEG-4 library, with an API to access the decoder algorithms as such compressed models are de-multiplexed from an incoming MPEG-4 stream.

5 Facial/Body Animation with Speech

5.1 Facial Analysis & Animation

Visual speech information is readily used by humans in a large variety of adverse viewing conditions. This contributes greatly to our ability to understand speech in even the most visually difficult and acoustically noisy situations. In fact, most people have little difficulty conversing in noisy environments when faces are visible. Automatic analysis of talking faces and acoustic speech is now able to provide high-level information about visual speech and facial expressions and gestures. The resulting face feature parameters can be used for speech recognition, talking face animation, and low bitrate coding.

5.2 Face Coding in MPEG-4

One basic objective of MPEG-4 SNHC is soliciting new technology for audio-video synchronization in multimedia applications where talking human faces, either natural or synthetic, are employed for interpersonal communication services, home gaming, advanced multi-modal interfaces, interactive entertainment, or in movie production. Facial sequences, in fact, represent an acoustic-visual source characterized by two strongly correlated components, a talking face and the associated speech, whose synchronous presentation must be guaranteed in any multimedia application. Therefore, the exact timing for displaying a video frame or for generating a synthetic facial image has to be supervised by some form of speech processing performed either as pre-processing before encoding or as post-processing before presentation.

In order to provide a comprehensive view of SNHC potentialities, let us consider a concrete example application in the field of interpersonal communications where a connection is established between two persons over a low bitrate channel. Voice communication is achieved using acoustic speech compression, while two synthetic virtual actors are displayed at each premise, where they are animated by means of synthetic data and by real-time parameter streams estimated from acoustic speech or face analysis. This minimal level of visual information guarantees natural facial expression and coherence of the lip movements with acoustic speech. Within the constraints of the bitrate, each person could send additional information about their face geometry and texture in order to allow the receivers to update the virtual actors or the surrounding virtual environment. In addition, speech synthesis from text or from phonetic transcription could also be used in the communication.

Figure 6 summarizes the major forms of visual and acoustic speech information which are represented and coded in the SNHC architecture. The definition of a standard set of face model parameters eliminates the need to download a particular face model to the receiver which could introduce unacceptable latency in low bitrate applications. The set of standard face model parameters is envisioned to be rich enough to satisfactorily control a wide variety of face models. A full range of facial expressions and speech articulations will be represented by combinations of standard parameters.

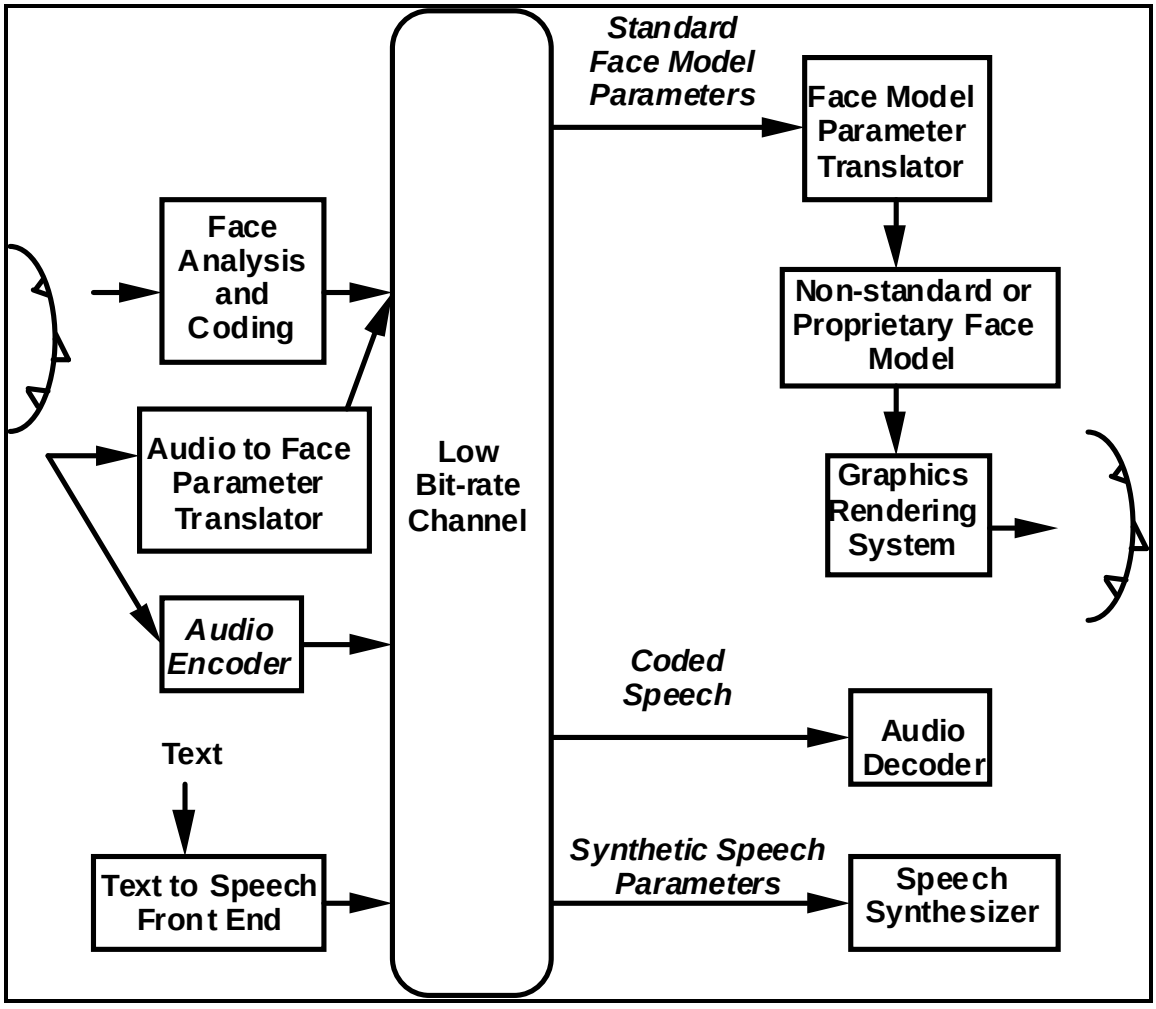

Figure 6. Functional Architecture for Low Bitrate Face Coding

5.3 Face Model Data Hierarchy

Among possible audio-video objects which could be represented and animated in SNHC virtual environments, virtual actors exhibit the highest complexity. With a combination of multi-layer rigid (bones, teeth) and non-rigid (tongue, lips, skin) 3D structures with high texture detail and complex 3D dynamics, the virtual actor must reproduce a wide variety of emotional information associated with a human face. Special attention must be paid to the synchronization of acoustic speech information with coherent visible articulatory movements of the speaker's mouth.

Figure 7 shows the four distinct layers of information associated with face animation coding. The inner-most layer contains only information used in low bitrate applications where a face animation system is resident in the receiver and is not downloaded. The second layer contains static geometric information used to construct the environment and static parts of the face or head. Examples include the 2D or 3D polygon mesh that establishes the structure and geometric shape of the undeformed and static face and lips, or a limited polygon/object backdrop to suggest that the face model occupies a specific place. The third layer contains the face model program, for

transforming parameter streams into facial deformations, in case one does not exist in the receiver before transmission. Finally, the fourth layer contains textures and video which consume large amounts of channel capacity and receiver storage.

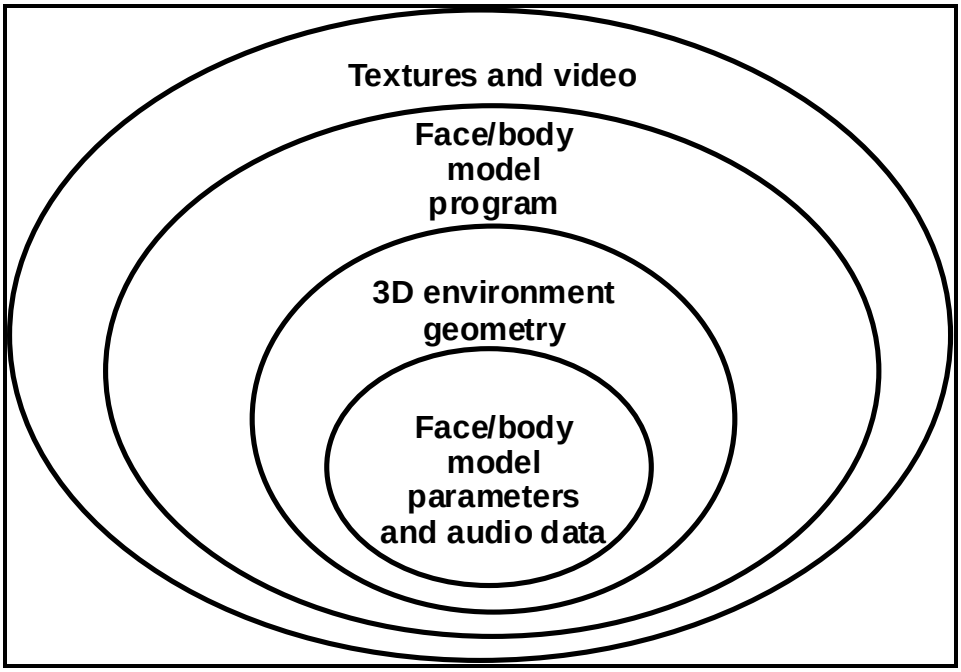

Figure 7. Face & Environment Representation Levels from Lowest (center) to Highest Data Rate

Figure 8 shows how each type of data would be used in a point-to-point communication. First the transmitter, knowing what type of parameter streams it is capable of sending and what corresponding resources are needed to initialize the receiver correctly, requests the characteristics of the receiving terminal in order to establish compatibility and to achieve optimal performance. Then, any initial data is downloaded before presentation. Finally, update data and control parameters are streamed to the receiver using a time-stamp mechanism for synchronization.

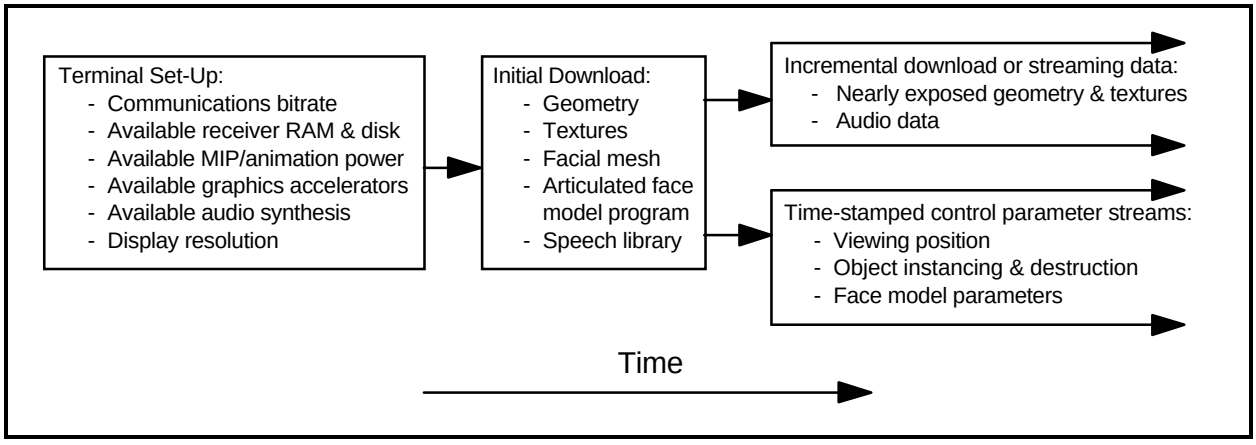

Figure 8. Operational Sequence for

5.4 Bimodality in Speech Production & Perception

Speech production is based on mechanisms of phonation, related to vibration of the vocal cords, and of vocal articulation, related to the time-varying geometry of the vocal tract responsible for the phonemic structure of speech. The vocal tract can assume various shapes by moving the jaw, tongue, lips and velum. In this way, the vocal tract implements a time-varying system capable of filtering the incoming acoustic wave, reshaping its spectrum and modifying the produced sound. Speech is the concatenation of elementary units or phones, generally classified as vowels if they correspond to stable configurations of the vocal tract or, alternatively, as consonants if they correspond to transient articulatory movements. Each phone is then characterized by a few attributes (open/closed, front/back, oral/nasal, rounded/unrounded) which qualify the articulation manner (fricative like /f/, /s/, plosive like /b/, /p/, nasal like /n/, /m/, ...) and articulation place (labial, dental, alveolar, palatal, glottal). Some phones, like vowels and a subset of consonants, are accompanied by vocal cord vibrations and are called "voiced" while other phones, like plosive consonants, are totally independent of vocal cord vibrations and are called "unvoiced".

All sighted humans rely (consciously or unconsciously) on visual speech information to enhance communication, especially in adverse acoustic environments. Lipreading represents the highest synthesis of human expertise in converting visual inputs into words and then into meanings. It consists of a personal database of knowledge and skills constructed and refined by training, capable of associating virtual sounds to specific mouth shapes, generally called "visemes", and therefore inferring the underlying acoustic message. The lipreader's attention is basically focused on the mouth, including all its components like lips, teeth and tongue, but significant help in comprehension comes also from the entire facial expression. Tight synchronization between the visual and acoustic signals is crucial for accurate and effortless speech understanding. For example, the onset of a bilabial plosive ('p') can occur within one frame time (< 20 ms), where an excessive time difference in the receiver between acoustic and visual display can mislead the user.

Audio-visual speech perception and lipreading rely on two perceptual systems working in cooperation so that, in case of hearing impairments or acoustic noise, the visual modality can efficiently integrate or even substitute the auditory modality. Experimental demonstrations have shown that exploitation of the visual information associated with the movements of the speaker's lips improves the comprehension of speech. The Signal-to-Noise Ratio (SNR) is improved by up to 15 dB, and auditory failure is transformed into near-perfect visual comprehension. The visual analysis of the speaker's face provides different levels of information to the observer, improving the discrimination of signal from noise. For example, the opening and closing of the lips is strongly correlated with the signal power and provides useful indications about how to segment the speech stream. While vowels, on one hand, can be recognized rather easily both through hearing and vision, consonants are very sensitive to noise (e.g. 'p' vs. 't') and the visual analysis often represents the only means for successful comprehension.

5.5 Speech Production Modeling

A very large set of acoustic signal observations is necessary to faithfully estimate speech articulation parameters. Classic methods generally aim at an accurate phoneme recognition and at

a consequent synthesis by rule to associate the corresponding viseme. In [10], [11], two estimation algorithms have been proposed for the statistical analysis of the cepstrum space for detecting phonetic clusters, representative of stable configurations of the articulatory organs, and for modeling the transition paths linking clusters, representative of coarticulation. An alternative approach consists of straightforward mapping of the acoustic representation onto predefined mouth-and-lips configurations [12]. This approach, since no coarticulation modeling is taken into account, cannot provide acceptable results when high quality of the visual output is required along with real-time processing of continuous speech. The highest quality model will result from observing both acoustic and visual speech articulation with sufficient spatial, spectral and temporal resolution.

Coding and communication of high quality acoustic-visual speech articulation requires the highest practical temporal accuracy (up to 60 Hz) and a rich set of articulation parameters. And given the variety of speech signal sources (acoustic, visual, and synthetic) and the variety of communication systems (varying bitrates and receiver performance), the set of articulation parameters must be organized in a consistent hierarchy to facilitate scalability as required by SNHC. The definition of the articulation parameter hierarchy remains as a challenge for SNHC.

5.6 Automatic Face Analysis Techniques

Several systems use various lip markers (dots or lipstick) to enable automatic face feature analysis. These approaches are usually designed to capturing visual speech information for scientific investigation of articulation. Early automatic speech-reading systems [13], [14], [15], [16], [17] were designed to demonstrate that visual speech information could improve the performance of acoustic speech recognition systems. Therefore, the lighting and viewing parameters were fixed, and the pixel intensity thresholds were manually set in order to collect the most consistent visual speech data possible. These systems used nostril tracking to reliably track the nose and mouth regions without the use of lip markers. An example of a mouth image sequence captured by one of these systems is shown in Figure 9 (the frames are in raster scan order). This sequence contains four bilabial closures (rows 3,5,6,8), examples of tongue visibility (rows 22,3,7,10), and upper teeth visibility (rows 4,11). The side of the dark patterns are always the inner lip contour (mouth corners). This mouth information was used successfully for speech recognition.

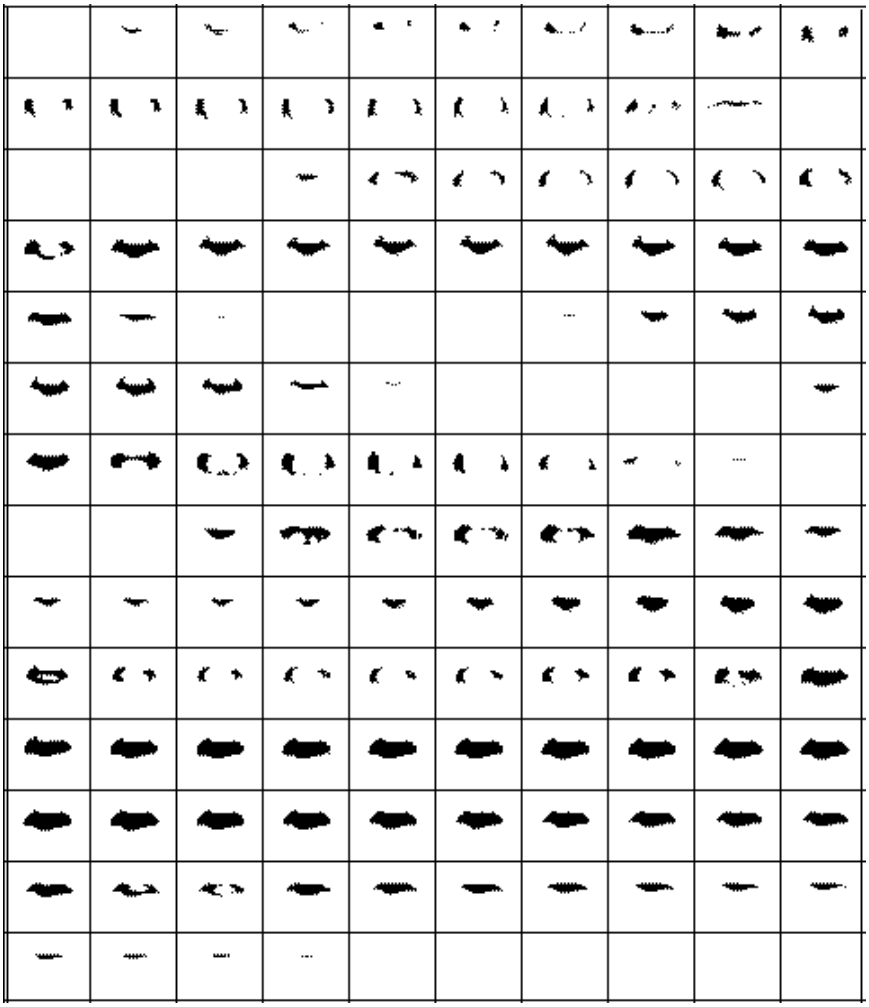

Figure 9. Binary Mouth Images for the Sentence:
"Jane may earn more money by working hard."

Recently, a system has been developed which captures facial features without facial markers and under varying lighting and viewing conditions. This system uses nostril tracking for robust operation in applications which allow appropriate camera placement. The primary objective of this system is to accurately estimate the inner lip contour of an arbitrary speaker under a large range of viewing conditions. An implicit system performance requirement is extremely robust tracking of the face (eyes, nostrils, and mouth). In addition, the system should reliably detect tracking failures and never falsely indicate the positions of the nostrils and mouth. Tracking failures routinely occur due to occlusion from hands (see Figure 11), extreme head rotation, and travel outside of the camera range.

Figure 10. Video Frame with Overlaid Face Features

Figure 11. System Detects & Recovers from Hand Occluding Face

5.7 Video-Driven Face Animation

Head position and inner lip contours can be used to control synthetic head models [18], [19], [20], [21]. The head position is taken as the nostril position and controls the horizontal and vertical head model position relative to the camera model in the graphics rendering system. The head tilt is controlled by either the angle between the eye line or nostril line and the horizontal, but the eye line is less noisy. The final tilt angle value is a running average over a few frames for increased smoothness. The inner lip contour is compared to the inner lip contour of the head model directly.

The head model mouth parameters (e.g., jaw rotation, upper and lower lip position, mouth corner position and mouth scale) are adjusted in combination for each frame until a sufficiently close match is achieved. For greater computational efficiency, the model parameter values from the previous frame are used to constrain the search for the optimal parameter values in the current frame. Since the jaw rotation is not directly estimated from the real face, it is estimated as a small fraction of the vertical mouth opening. Figure 12 was generated using the head tilt and inner lip contours from Figure 10.

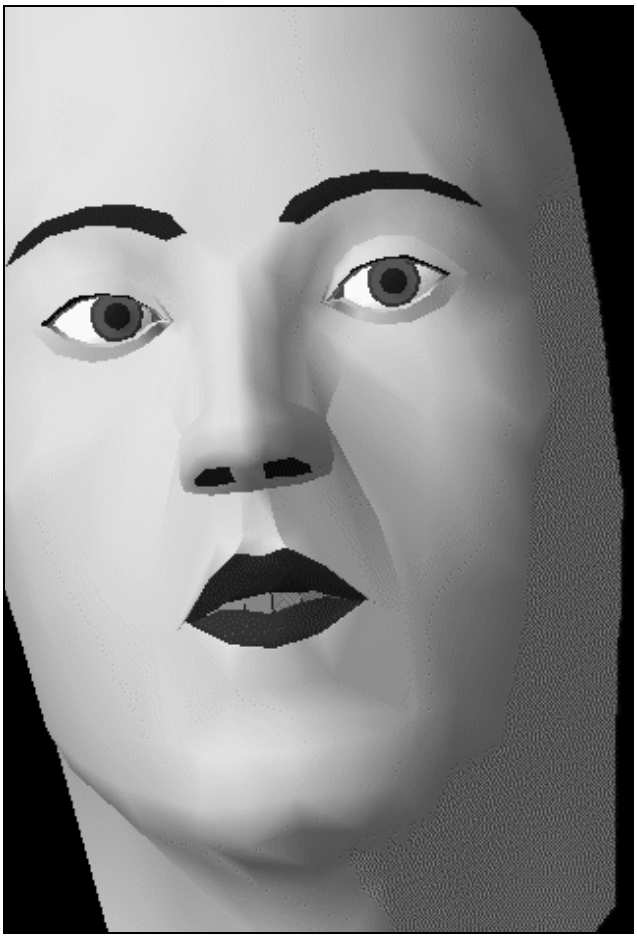

Figure 12. Synthetic Face Driven from Figure 10

5.8 Facial Animation Applications

Small color video cameras are inexpensive and easily positioned for ideal face viewing in many applications. For example, a camera placed just below a computer monitor provides a good view of all facial features (including nostrils) throughout all normal head motions, e.g. looking down at the keyboard and slouching. Other application environments which allow ideal camera position include bank machines, cars, kiosks, point of sale terminals (cash registers), laptop or notebook computers, copying machines, access control stations, aircraft cockpits, and personal digital assistants.

The minimum set of face parameters needed to drive a synthetic talking face is quite small. All of these parameters can be encoded into a very low bitrate signal for transmission over ordinary phone lines using either voice/data modem or data modem with digital audio compression. As shown in Figure 13, the applications for this technology include video conferencing, model-based coding, networked interactive games, tele-video marketing, enhanced public address in noisy environments, entertainment, speech recognition, and enhanced computer/human interaction. If the system is simultaneously used for video coding and speech recognition, then the cost of

implementation is more easily justified. Since the face features are easily transmitted using the telephone network, acoustic-visual speech recognition in the network is feasible.

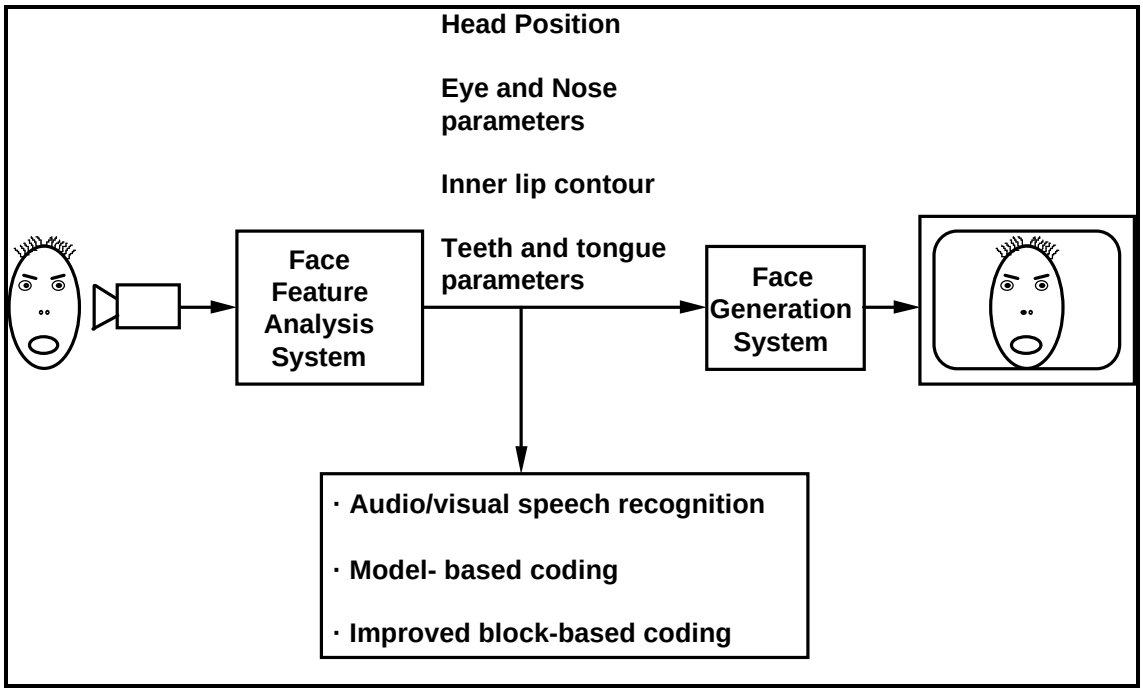

Figure 13. Face Feature Analysis Applications

5.9 Body Representation & Animation

Increasing hardware and network performance together with software technology make it possible to define more complex interfaces with the A/V objects. For more natural interaction, the user should feel that he/she is present in the virtual environment, and should interact using natural gestures and expressions. The Virtual Human body is an important contributor to interaction with the environment and other users sharing the same objects.

Real-time 3D representation of Virtual Human figures has been one of the challenging goals in computer graphics. The most commonly used approach for body modeling is based on a layered approach: skeleton, skin and cloth. The skeleton layer consists of a tree-structured fixed-topology hierarchy of joints connecting limbs, each with minimum and maximum limits. Attached to the skeleton, there is the skin layer, which is responsible for generating 3D skin surfaces of the body. Above the skin surface, a 3D cloth layer can be wrapped. Animation of the virtual body is performed in the skeleton layer, and the above two layers are automatically computed by deforming and transforming 3D vertices. It is also possible to replace these two layers with non-deformed, rigid-body surfaces for faster computation. Various methods are possible for animating the Virtual Humans: key-framing, inverse kinematics, dynamics, and specialized motion control techniques such as walking and grasping.

Virtual Human body modeling shares ties and similarities with facial animation. The body gestures should be synchronized with facial expressions and speech. Therefore it is necessary to consider body animation relative to facial animation. The SNHC approach to body animation is similar to facial animation: the expected standard is a set of parameters for body modeling and animation, and compression of the bitstream for coding body postures during animation.

On the other hand, body animation exhibits dissimilarities with facial animation. There exist various 3D input devices that can be attached to the body, eliminating the need for image processing for feature tracking. For synchronization with audio, body animation is less demanding than facial animation. On the other hand, the virtual body can be used to interact with the objects in the environment; hence it needs synchronization with the object transformations.

The expected standard will possibly consist of the following: body modeling parameters to include global scaling of the body, the ratio between the upper and lower body parts, multiple degrees of motion freedom for different levels of animation detail (the ability to animate simple or complex chains of skeletal kinematics with limited or complex body models, depending on transmitter/receiver resources and on whether general body movement is sufficient or the subtleties of body gesture and motion are needed), and baseline transformations between body coordinate systems.

The animation will be parameterized by defining a compressed bitstream for coding the dynamics of body skeleton posture during the animation. As with facial animation, specific body models will not likely be standardized, but the development of their detail appearance and the animation linkage to standardized parameters will be left to application developers. Downloading or pre-storage of body models in the receiver and their control by standardized parameters will depend on specific content and the session resources of transmitter and receiver.

5.10 What to Standardize

Now we can summarize what should be standardized. For the facial/body animation area, SNHC will standardize the face and body definition parameters, and the animation parameters that are used during a real-time session to provide remote control for bringing varied possible face and body models to life. For the speech coding, there exist several variations of TTS systems for different languages. So the TTS module itself should not be standardized. What should be standardized are the interfaces between the de-multiplexer and TTS, between TTS and the audio decoder, between TTS and facial animation, and the user interface for TTS. The TTS module per se is a black box. So the methods to drive the TTS module from the de-multiplexer and/or user, and the methods to drive facial animation from TTS, can be standardized. The standardization of these TTS APIs defines a minimum (or common) set of parameters for TTS synthesis and should allow application-specific developments and innovation.

6 Networked Collaborative Virtual Environments

In its ultimate future extension, SNHC should provide a framework for the development of multi-user interactive Virtual Environments (VEs) on the network. This type of application allows multiple users to move and interact in a common virtual environment that integrates 3D, 2D, audio, and video objects. The users themselves are represented within the environment using Virtual Humans with articulated faces and bodies allowing perception, identification, interaction

and communication. Networked Collaborative Virtual Environment systems are suitable for numerous collaborative applications ranging from games to medicine. Figure 14 shows an example of a virtual tele-conference with Virtual Humans, 3D, and video objects.

Figure 14. An Example of Networked Collaborative Virtual Environment with Virtual Humans (snapshot from Virtual Life Network [27])

6.1 Principles of Networked Collaborative Virtual Environments

Networked Collaborative Virtual Environments have been a hot topic of research for several years now, and a number of working systems exist [22], [23], [24], [25], [26], [27]. They differ largely in networking solutions, number of users supported, interaction capabilities and application scope, but share the same basic principle illustrated in Figure 15. Each workstation has a copy of the virtual environment. The user can move within the environment and interact with it. All events that have an impact on the environment are transmitted to other sites so that all environments can be updated and kept consistent, giving the impression for the users of being in the same, unique environment. The users become a part of the environment, and they are embodied by a graphical representation that should ideally be human-like.

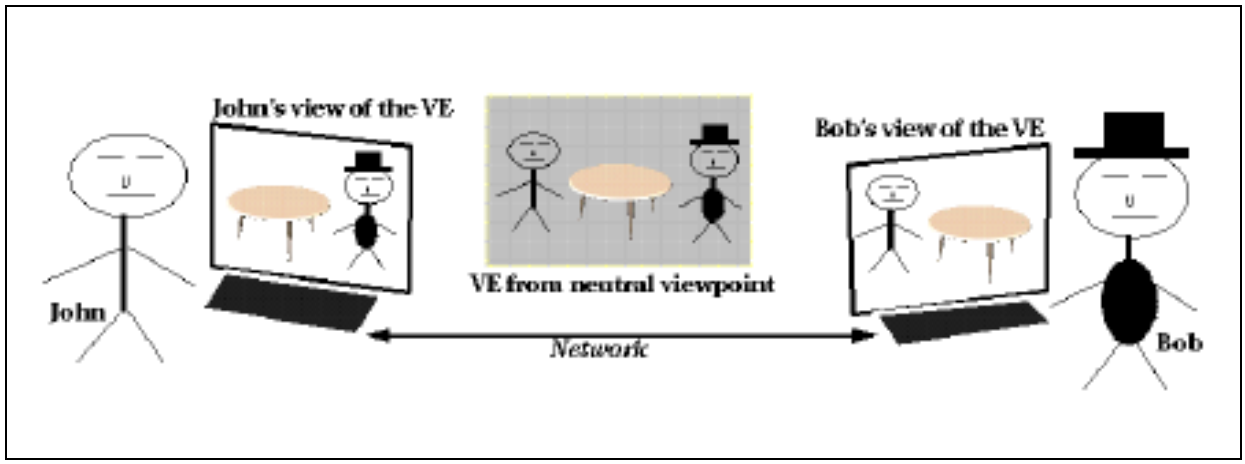

Figure 15. Principles of Networked Collaborative Virtual Environments

6.2 Media Integration

Virtual Environments (VEs) in general, and Networked Collaborative Virtual Environments in particular, provide an ideal framework for the integration of various media objects (3D, 2D, audio, video, images), as well as interaction with these objects. All object types can be treated as special cases of 3D objects, providing a uniform way of perceiving and interacting with objects.

Since VEs are three-dimensional by nature, 3D objects are basic building blocks of such environments. The capability to visualize (render) 3D objects in real time, while interactively changing the point of view, is a basic feature of a VE. In the simplest form of interaction, the user is allowed to move the objects freely in 3D. More interesting are object behaviors achieved by scripting, which can produce much more complicated interactions. For representation and interaction, 2D objects placed in a 3D environment can be regarded as a special case of 3D objects (one dimension is zero).

3D audio rendering techniques support the positioning of the audio objects in 3D within the environment. Such sound sources can be embodied by a graphical representation allowing for interaction in the same manner as for ordinary 3D objects.

Video objects can be pasted (texture-mapped) on any 3D or 2D object in the environment, and the user can interact with them in the same fashion. Moving a video-mapped object moves the video; deformations can be used for zooming or special effects. Semi-transparent video objects can blend with the environment. Images are treated in the same manner as video.

6.3 Virtual Humans

In its initial extension, SNHC concentrates on facial/body animation. This leads to a complete, integrated representation of the human body: the Virtual Humans [28], [29]. Within Networked Collaborative Virtual Environments, the Virtual Humans have several important functions:

- perception (to see if anyone is around)

- localization (to see where the person is)
- identification (to recognize the person)
- visualization of interest focus (to see where the person's attention is directed)
- visualization of actions (to see what the person is doing)
- communication (lip movements in sync with speech, facial expressions, gestures)

Virtual Humans can fulfill all these functions in an intuitive, natural way resembling the manner in which we achieve these tasks in real life.

6.4 What to Standardize

Data Exchange for Virtual Environments

Networking for Collaborative Virtual Environments can be roughly illustrated by Figure 16. The cloud in the middle of Figure 16 interconnects the participating hosts, and can be implemented in various ways: client/server architecture, multi-casting, multiple servers or combinations [30]. As the number of users grows, filtering techniques are necessary to decrease the volume of network traffic by delivering messages to users on an as-needed basis. The solutions to these problems are diverse and may be application-specific, so they are out of SNHC scope.

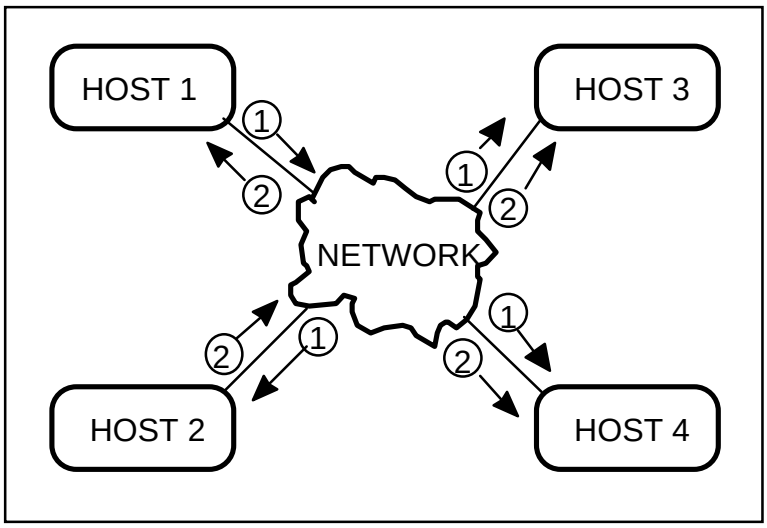

Figure 16. Simplified View of Networking for Collaborative Virtual Environments

SNHC will concentrate on the content transmitted through the network (bubbles in Figure 16): downloads, updates, audio, video, states of objects, etc. The standardization of compressed dynamic state data, transferred efficiently over networks to animate objects and to allow specific clients to detect events arising from interaction with other users or models, could be useful. There is a significant body of work to draw upon concerning dynamic parameters used in gaming, real-time simulation, and Distributed Interactive Simulation for networked connection of multiple

simulators. The coding of such data, as with facial/body animation, would consider compression of parameter streams, time synchronization relative to a system time reference, and scalability.

3D Geometry Compression

As discussed earlier in Section 4, the connection of a single user with a model server to interact with a 3D virtual environment, or the interconnection of multiple users to share experiences in a common 3D synthetic world, raises questions about how the virtual environment model reaches the user's terminal for animation and rendering. In some applications, a user may need a simple model download, followed by local session interaction with that model. In more advanced cases, the user may need progressive downloads of model data as the synthetic environment is navigated. In yet other cases, incremental geometry streams could be transmitted to change a model during a session. This obviously invites compression of structured models with their various attributes, as well as the geometry stream compression mentioned above. SNHC is pursuing geometry compression as discussed in Section 4.

6.5 Application Scope

Networked VEs that integrate various media objects can be of use in an very broad range of applications. We present a non-exhaustive list of examples:

- Virtual teleconferencing with multimedia object exchange
- All sorts of collaborative work involving 3D design
- Multi-user game environments
- Tele-shopping involving 3D models, images, sound (e.g. real estate, furniture, cars)
- Medical applications (distance diagnostics, virtual surgery for training)
- Distance learning/training
- Virtual Studio/Set with Networked Media Integration
- Virtual travel agency

Some application examples are illustrated in Figure 17. Figure 18 shows another example of facial animation based upon video-driven analysis of facial expressions, with real images and the corresponding results of controlling a synthetic 3D face. This again suggests how model-based animation can conserve bandwidth relative to video coding.

Figure 17. Some Examples of Networked Collaborative Virtual Environment Applications: Tele-shopping, Games, Medical Education (snapshots from Virtual Life Network [27])

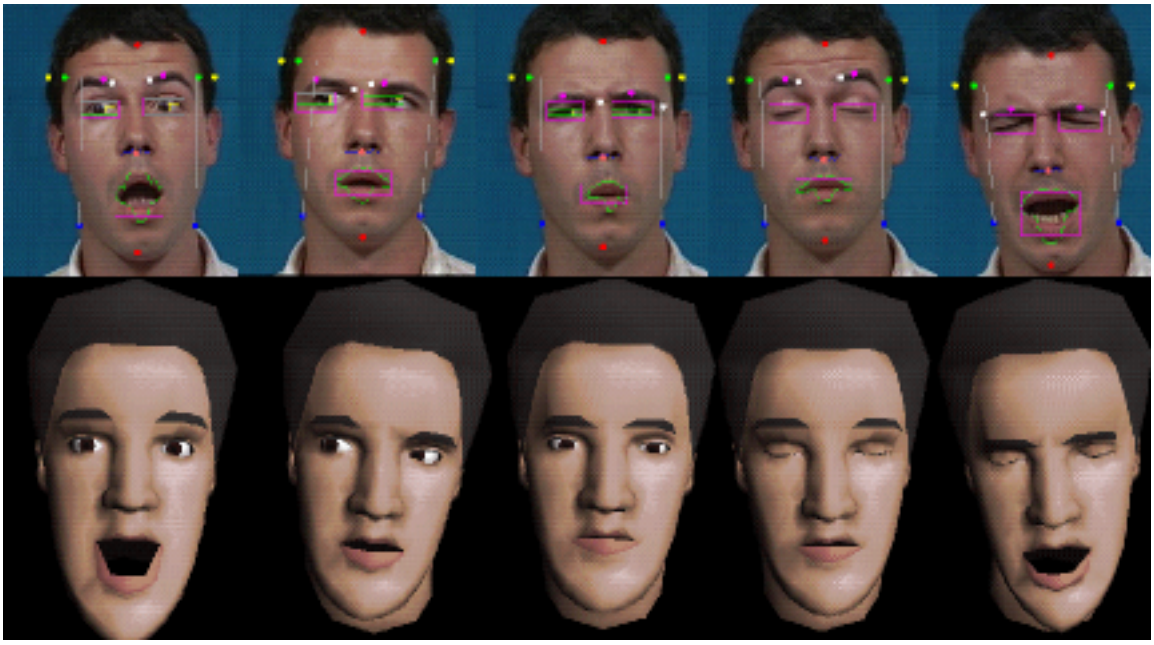

Figure 18. Facial Expressions Coded with 15 Parameters and Synthesized on a Virtual Face [31]

7 Conclusion

MPEG-4 seeks to develop standards of coding for combining audio, video, and 2D/3D graphics to support efficient communication of A/V objects in animated mixed-media delivery. The task is approached in steps, with the initial emphasis on facial/body animation and on combining audio/video capabilities of MPEG-4 with animated 2D overlay graphics. The SNHC effort will rely heavily on other component standards in text, natural and synthetic audio, speech, video, imagery, 2D/3D graphics, and scripting, which are suited for real time. An initial specification for an SNHC Verification Model has been developed dealing with face/body coding, text-to-speech synthesis, media integration, and SNHC audio [34]. A supporting document details a specification for face and body definition and animation parameters [35]. Work accomplished in recent months has been fueled by SNHC proposal submissions on geometry compression, facial and body animation, text-to-speech synthesis, driving facial animation with text/speech, and media integration. Related work proceeds on the MPEG-4 MSDL architecture for augmenting its specification to define APIs for SNHC-related functionalities, invoking external standards where appropriate, and verifying that the multiplexing, synchronization, and A/V object class hierarchy supported by MSDL are consistent with SNHC requirements. Future work is expected to focus on software elements for building SNHC core experiments as MSDL-based demonstrations.

References

[1] J. Shade, D. Lischinski, D Salesin, T. DeRose, J. Snyder, “Hierarchical Image Caching for Accelerated Walk-throughs of Complex Environments,” ACM Computer Graphics Proceedings, Siggraph 96, New Orleans, pp. 75-90, August 1996.

[2] A. Finkelstein, C. Jacobs, D. Salesin, “Multi-Resolution Video,” ACM Computer Graphics Proceedings, Siggraph 96, New Orleans, pp. 281-290, August 1996.

[3] H. Hoppe, “Progressive Meshes,” ACM Computer Graphics Proceedings, Siggraph 96, New Orleans, pp. 99-108, August 1996.

[4] G. Taubin, J. Rossignac, “Geometric Compression Through Topological Surgery,” research report, IBM Research, RC-20340 (#89924), 22 pages, January 1996.

[5] M. Deering, “Geometry Compression,” ACM Computer Graphics Proceedings, Siggraph 95, Los Angeles, pp. 13-20, August 1995.

[6] S. E. Chen, “QuickTime VR - An Image-Based Approach to Virtual Environment Navigation,” ACM Computer Graphics Proceedings, Siggraph 95, Los Angeles, pp. 29-38, August 1995.

[7] L McMillan & G. Bishop, “Plenoptic Modeling: An Image-Based Rendering System,” ACM Computer Graphics Proceedings, Siggraph 95, Los Angeles, pp. 39-46, August 1995.

[8] J. Torborg, J. Kajiya, “Talisman: Commodity Real-time 3D Graphics for the PC,” ACM Computer Graphics Proceedings, Siggraph 96, New Orleans, pp. 353-363, August 1996.

[9] L. Williams, “Pyramidal Parametrics,” ACM Computer Graphics Proceedings, Siggraph 83, pp. 1-11, July 1983.

[10] F. Lavagetto, D. Arzarello, M. Caranzano, "Lip-readable Frame Animation driven by Speech Parameters", IEEE Int. Symposium on Speech, Image Processing and neural Networks, Hong Kong, April 14-16, 1994.

[11] B. Pinkowski, "LPC Spectral Moments for Clustering Acoustic Transients", IEEE Trans. on Speech and Audio Processing, Vol. 1, N. 3, pp. 362-368, 1993.

[12] B. P. Yuhas, M. H. Goldstein Jr. and T. J. Sejnowski, "Integration of Acoustic and Visual Speech Signal Using Neural Networks", IEEE Communications Magazine, pp. 65-71, 1989.

[13] Petajan, E. D., "Automatic Lipreading to Enhance Speech Recognition", Proceedings Globecom Telecommunications Conference, pp. 265-272, IEEE, 1984.

[14] Petajan, E. D., "Automatic Lipreading to Enhance Speech Recognition", Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 40-47, IEEE, 1985.

[15] Petajan, E. D. and Brooke, N. M. and Bischoff, G. J., and Bodoff, D. A. "An Improved Automatic Lipreading System to Enhance Speech Recognition," in "Proc. Human Factors in Computing Systems," pp. 19-25, ACM, 1988.

[16] Goldschen, A., "Continuous Automatic Speech Recognition by Lipreading," Ph.D., George Washington University, 1993.

[17] Goldschen, A. and Garcia, O. and Petajan, E., "Continuous Optical Automatic Speech Recognition," Proceedings of the 28th Asilomar Conference on Signals, Systems, and Computers," pp. 572-577, IEEE, 1994.

[18] Cohen, Michael M. and Massaro, Dominic W., "Modeling Coarticulation in Synthetic Visual Speech," Models and Techniques in Computer Animation, Springer-Verlag, 1993.

[19] Parke, F. I., "A Parametric Model for Human Faces", Ph.D., University of Utah, 1974.

[20] Parke, F. I., "A Model for Human Faces That Allows Speech Synchronized Animation", Journal of Computers and Graphics, 1975.

[21] Parke, F. I., "A Parameterized Model for Facial Animation", IEEE Computer Graphics and Applications, 1982.

[22] Barrus J. W., Waters R. C., Anderson D. B., "Locales and Beacons: Efficient and Precise Support For Large Multi-User Virtual Environments", *Proceedings of IEEE VRAIS*, 1996.

[23] Carlsson C., Hagsand O., "DIVE - a Multi-User Virtual Reality System", *Proceedings of IEEE VRAIS '93*, Seattle, Washington, 1993.

[24] Macedonia M. R., Zyda M. J., Pratt D. R., Barham P. T., Zestwitz, "NPSNET: A Network Software Architecture for Large-Scale Virtual Environments", *Presence: Teleoperators and Virtual Environments*, Vol. 3, No. 4, 1994.

[25] Ohya J., Kitamura Y., Kishino F., Terashima N., "Virtual Space Teleconferencing: Real-Time Reproduction of 3D Human Images", *Journal of Visual Communication and Image Representation*, Vol. 6, No. 1, pp. 1-25, 1995.

[26] Singh G., Serra L., Png W., Wong A., Ng H., "BrickNet: Sharing Object Behaviors on the Net", *Proceedings of IEEE VRAIS '95*, 1995.

[27] D. Thalmann, T. K. Capin, N. Magnenat Thalmann, I. S. Pandzic, “Participant, User-Guided, Autonomous Actors in the Virtual Life Network VLNET”, *Proc. ICAT/VRST ’95*, pp. 3-11.

[28] N. I. Badler, C. B. Phillips, B. L. Webber, "Simulating Humans: Computer Graphics Animation and Control", Oxford University Press, 1993.

[29] Boulic R., Capin T., Huang Z., Kalra P., Lintermann B., Magnenat-Thalmann N., Moccozet L., Molet T., Pandzic I., Saar K., Schmitt A., Shen J., Thalmann D., "The Humanoid Environment for Interactive Animation of Multiple Deformable Human Characters", *Proceedings of Eurographics '95*, 1995.

[30] Funkhouser T. A., "Network Topologies for Scalable Multi-User Virtual Environments", *Proceedings of VRAIS '96*, 1996.

[31] Kalra P., Mangili A., Magnenat Thalmann N., Thalmann D., "Simulation of Facial Muscle Actions Based on Rational Free Form Deformations", *Proc. Eurographics '92*, pp. 59-69, 1992.

[32] Jörn Ostermann, “An Interface for the Animation of Human Heads from Text”, Contribution No. MPEG96/M1197, October 1996 Chicago Meeting of ISO/IEC JTC1/SC29/WG11.

[33] Frank Bossen, “Geometry Compression”, Contribution No. MPEG96/M1236, October 1996 Chicago Meeting of ISO/IEC JTC1/SC29/WG11.

[34] SNHC Ad Hoc Groups, “Draft Specification of SNHC Verification Model 1.0”, Document No. MPEG96/N1364, October 1996 Chicago Meeting of ISO/IEC JTC1/SC29/WG11.

[35] SNHC Face/Body Ad Hoc Group, “Face and Body Definition and Animation Parameters”, Document No. MPEG96/N1365, October 1996 Chicago Meeting of ISO/IEC JTC1/SC29/WG11.