《主 題》

MPEG-4 SNHC

최 석 림
(세종대학교 전자공학과 교수)

□차 례□

Ⅰ. 서 론

MPEG-4는 디지털 영상 및 음성 압축 분야에서 이미 널리 사용되어지고 있는 MPEG-1과 MPEG-2에 이어 ISO/IEC JTC1 SC29 WG11에서 개발되어지고 있는 차세대 표준이다. 이러한 MPEG-1, MPEG-2 표준들은 CD-ROM상에서 Interactive Video를 구현하고 Digital Television을 가능하게 만들었는데, MPEG-4도 전세계적으로 많은 연구원과 엔지니어의 노력으로 조만간에 또 다른 성과를 이끌어 낼 것이다. MPEG-4는 예전의 MPEG-1,2와는 다르게 CD와 그 이상의 저장 장치와 Internet Web등에서 interactive한 video/audio를 object로 다루게 된다.

ISO/IEC 14496으로 공식적으로 명명될 MPEG-4는 1998년 11월에 공개되어 1999년 1월에 국제 표준이 될 예정으로 현재는 Digital Television과 Interactive Graphics Application, 그리고 World Wide Web의 세 분야를 기반으로 AVOs(Audio/Visual Objects)의 표현과 작성, 그리고 Audio/Visual Scene(그림 1)에서의 Interaction을 목표로 하고 있다.

여기에서는 MPEG-4의 Interactive Graphics Application에서 사용되는 합성된 AVOs를 다루는 SNHC에 대해 SNHC의 정의, SNHC가 다루는 분야와 현재 진행 상황, 그리고 SNHC분야의 세부적인 주제을 하나하나 살펴 보고자 한다.

MPEG-4 SNHC 란 ?

최근까지의 Aural/Visual 코딩은 보고 들을 수 있는 실세계를 2-D 평면위에서 만들어 냈다. 이러한 코딩은 특별한 알고리즘을 가지게 되었고 표준화된 syntax로 표현되었다. MPEG(Motion Picture Expert Group)은 Audio와 Video 정보를 복원하는 방식을 기술하였다. MPEG-4는 지금까지의 MPEG 코딩과는 다르게 Object-Oriented Structure를 적용하여 Audio-Video data를 표현함에 있어서 보다 많은 유연성과 확장성을 제공한다. 특히, MPEG-4 SNHC(Synthetic Natural Hybrid Coding)는 자연영상과 합성영상 정보의 효과적인 표현과 작성을 돕는다.

현재의 SNHC

현재 SNHC가 다루는 영역은 아래와 같다.

- Synthetic Face and Body description, and its Animation
- Text-to-Speech synthesis, and its Facial animation interface
- Media Integration of Text and Graphics
- View dependant Texture Coding
- Static and Dynamic Mesh Coding
- Synthetically Audio

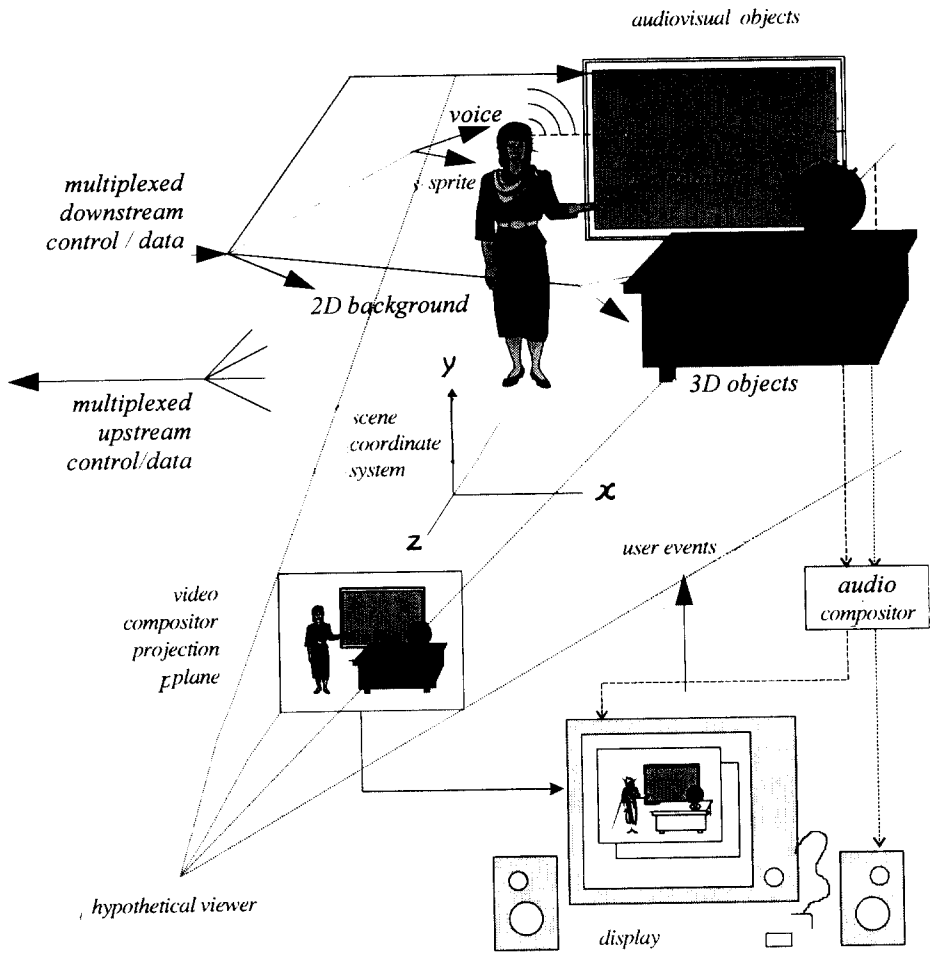

그림 1. MPEG-4 AudioVisual Scene의 예

위의 여섯 가지 분야는 진행 상황에 따라 VM(Verification Model), WD(Working Draft), CD(Commitee Draft), 그리고 IS(International Standard)의 순서로 발전한다. 현재는 위의 분야중에 Facial Coding, MITG, 2-D Mesh Coding, View dependant Texture Coding, Scalable Texture Coding, Synthetically Audio, 그리고 TTS는 WD까지 진행되었으나 나머지 분야는 아직 VM단계에 머무르고 있다.

II. Face & Body Object

합성에 의해 얼굴과 몸체를 표현하기 위해 Face와 Body object를 정의하였다. Face object는 3-D polygon mesh의 형태로 face를 만들어 낸다. 이 때 face의 shape, texture, expression은 일반적으로 뒤에서 설명할 FDP(Facial Definition Parameter)와 FAP(Face Animation Parameter)를 포함하는 bitstream에 의해

제어된다. 구성에 있어서 Face object는 자연스런 표정(뒤에서 정의함)을 가진 일반적인 얼굴을 기본으로 하여 FDP에 의해 이를 변형시켜 원하는 얼굴 형태를 표현한다. 일단 FDP에 따라 한 얼굴 형태의 정보가 전달되면 그 이후의 얼굴의 animation은 bitstream의 FAP만으로 만들어 낼 수 있다. Body object 역시, 3-D polygon mesh의 형태로 가상 body model을 만들어 낸다. Body object도 Face object와 비슷하게 BDP와 BAP를 가진다. 이 중 BDP는 default body를 body의 surface, dimension, texture를 가진 customized body로 변형시켜 준다. Body object는 가상 body model의 기본 자세를 정의하고 있다. 여기서의 기본 자세란 두 발은 앞으로 향하고 팔은 나란히 옆구리에 붙이고 손바닥은 안쪽을 향해 두고 있는 자세이다.

1. Facial Animation Parameter Set

FAP는 얼굴의 움직임의 이론에 기초를 두고 얼굴 근육의 움직임과 밀접한 관계를 가진다. 따라서 자연스런 얼굴 표정(뒤에서 정의함)을 포함하여 기본적인 얼굴 표정을 표현한다. 만약 얼굴의 움직임에 가능하지 않은 값이 정의된다면 만화의 캐릭터같은 표정을 갖게 된다. 여기서는 FAPU(FAP Unit)를 사용하여 얼굴의 움직임을 정의하고 여기에 더해 Viseme Parameter와 Expression Parameter의 두가지 High Level Parameter를 사용하였는데, 먼저 FAPU(그림 2)의 종류를 다음에 보였다.

- IRISDo = Iris diameter; IRISD = IRISDo / 1024
- ESo = Eye separation; ES = ESo / 1024
- ENSo = Eye - Nose separation; ENS = ENSo / 1024
- MNSo = Mouth - Nose separation; MNS = MNSo /1024
- MWo = Mouth width; MW = MWo / 1024

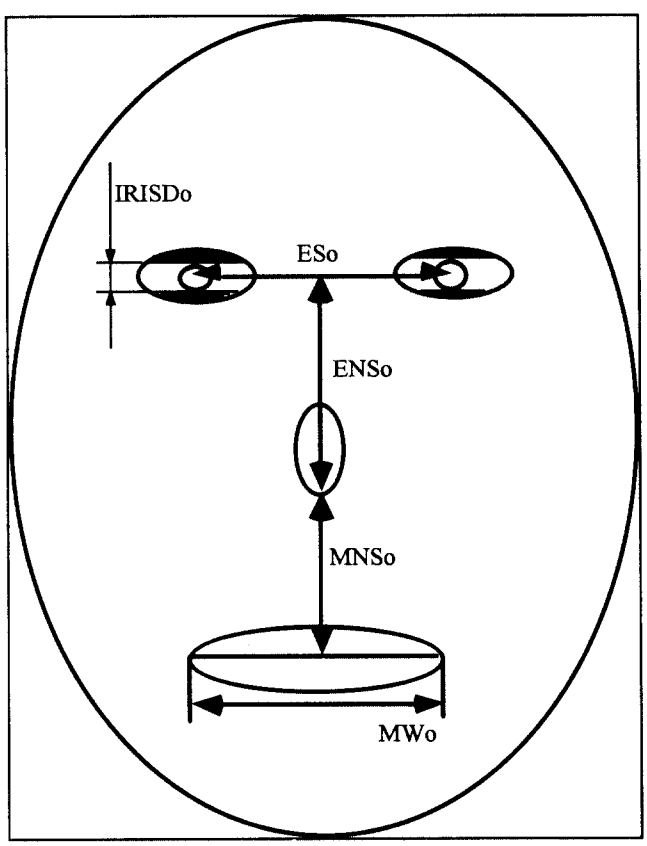

그림 2. The Facial Parameter Unit

이러한 FAPU에 따라서 parameter가 정의되었는데 만약 parameter가 0의 값을 가지면 이것은 neutral face를 가르키게 된다. 다음은 neutral face(그림 3)의 특징과 그 예이다.

- all face muscles are relaxed
- glance is in direction of Z axis
- eyelids are tangent to the iris
- the pupil is one third of IRISD
- the coordinate system is right-handed; head axes are parallel to the world axes
- lips are in contact; the line of the lip is horizontal and at the same height of lip corners
- the mouth is closed and the upper teeth the lower ones

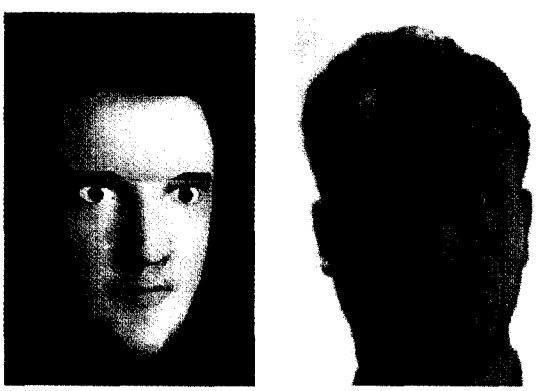

그림 3. Neutral face의 예

FAPs에는 FAPU에 포함된 parameter와 함께 viseme와 expression parameter의 두 high level parameter가 있다. 그 동안 여러 언어에서의 발음에 대한 연구의 결과로 음소의 집합을 정의하는 것이 가능해졌는데 이것을 FA(Facial Animation)과 관련시켜 viseme parameter로 정의하고 있다.

viseme parameter는 다시 viseme의 type을 결정하는 viseme_select와 선택된 viseme의 intensity를 나타내는 viseme_intensity로 나누어진다. 현재는 viseme_select가 15가지로 나누어지지만 앞으로 추가될 예정이다. 또한 FAPU에 포함된 값 중에서 다음의 4가지 high level parameter가 모음을 표현하고 있다.

- Lip Opening Height (LOH)
- Jaw on Y axis (JY)
- Lip Opening Width (LOW)
- Lower Lip Protrusion (LLP)

따라서 모음의 a, e, i, o, u가 나타내는 LOH, JY, LOW, LLP는 특정한 값을 값을 갖게 된다. 이 viseme parameter와 더불어 expression parameter도 expression_select와 expression_intensity로 나누어 진다. expression_select는 1부터 6까지의 값을 가져서 각각 joy, sadness, anger, fear, disgust, surprise를 나타내고 있다. 여기에 expression_intensity가 0부터 15까지, 0은 neutral face를 나타내고, 15는 maximum intensity를 나타내게 된다. 각각의 expression_select에 따라서 얼굴의 texture가 달라지게 된다.

2. Body Animation Parameter Set

이 BAP는 크게 4가지의 subclass로 나눌 수 있다.

- Global Positioning Domain Parameters : 이 parameter는 body에서 특별히 눈에 띄는 point의 전역 좌표 위치와 방위를 가르키는 값으로 쇄골, 어깨, 팔꿈치, 골반, 힙 등의 point에서의 위치와 방위를 가진다.
- Joint Angle Domain Parameters : 이 parameter는 서로 다른 body part를 연결하는 결합각을 포함한다. 이 값을 가지는 결합 관절로는 발가락, 발목, 무릎, 힙, 척추, 어깨 등을 들 수 있다.
- Hand and Finger Parameters : 손은 복잡한 동작을 수행하는 기능을 가지므로 한쪽 손을 25DOF(Degree Of Freedom)로 나누었다.
- High Level Parameters - 이 Parameter는 lower level parameter에 의해 기술될 필요없는 high level 표정, 동작을 정의하는데 사용된다. 이것은 다른 parameter와는 달리 기본이 되는 자세의 동작 제어를 수행하며, 이 parameter에 포함될 수 있는 예로는 procedual한 gesture, 미리 정의된 자세, 그리고 force parameter를 들 수 있다. 이 high level parameter set과 그 입력값을 아직 정의되지 않았으나 이 중에서 force parameter는 힘의 정도와 방향을 나타내어 주는 body animation의 한 부분이 될 것이다.

Body Animation은 위에 정의된 4가지의 parameter subclass와 함께 세부적인 DOF를 정의하고 있다. 이 DOF 역시 크게 4부분으로 나누어 정의하였다.

- Lower body
- At the foot complex

- Upper body
- Hands

3. Facial Definition Parameter Set

FDPs은 주어진 얼굴을 특별한 얼굴로 구체화하는데 사용된다. FDPs은 압축된 FAPs의 스트림에 의해 세션당 한번씩 전송된다. 그러나 만약 decoder가 FDPs을 못받아도 기본의 얼굴과 FAPU의 사용으로 FAP 스트림을 해석할 수 있다. 이것은 방송이나 화상회의 시스템에서 최소한의 작동을 보장해주게 된다.

FDPs는 아래의 내용을 포함하고 있다.

- 3D feature points on the mesh
- 3D mesh
  (with texture coordinates if texture is used)
- texture image
- personal attributes (hair, glasses, age, gender)

위에서 언급한 3D mesh는 face 모델, 그 자체는 아니지만 face 모델의 형태를 정의해주고 있다. 그리고

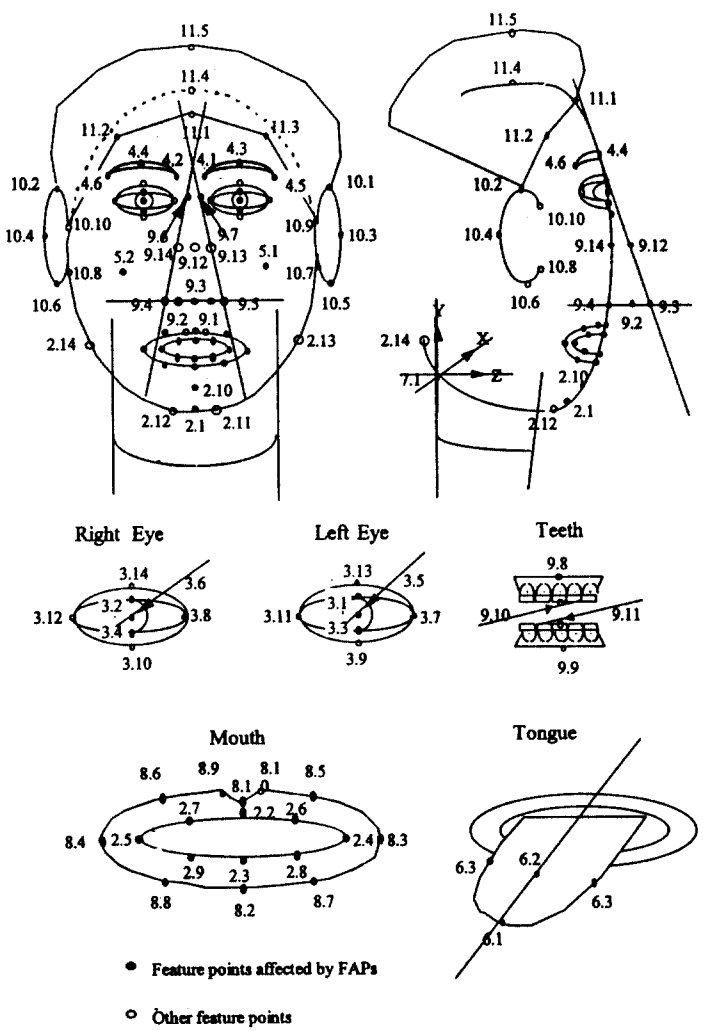

그림 4. FDP feature point set

3D feature point는 그림 4에 보여진 3D mesh에 얼굴 형태를 위치시키는데 사용된다. 그 그림에서 FAP 정의에 포함된 feature point는 FDP mesh feature point 이며 추가된 mesh feature point는 shape와 texture를 일치시키는데 사용된다. 3D mesh나 texture없이 전적으로 feature point에 기초해서 얼굴을 구체화하는 것이 가능하지만 3D mesh나 texture는 얼굴의 visual quality를 잠재적으로 향상시켜 준다.

4. Body Definition Parameter Set

body definition parameter set은 다음의 정보를 포함하고 있다.

1) body surface geometry
- body surface geometry는 3D mesh 전송 메카니즘을 사용하여 download되는데 이러한 surface는 VRML 포맷이 될 것이다.

2) 3D reference points
- body의 요구된 dimension은 body 좌표 시스템에서 body 경계점을 사용해서 정의될 수 있다.

3) texture images (선택적)

4) geometry의 부속 정보
- body surface의 부속 정보는 요구된 변형을 가진 body surface를 위치시키는데 사용되며 attach될 body의 DOF(Degree Of Freedom)의 관점에서 body surface의 local 위치와 방위도 초기화된다.
- body surface와 부속점의 수는 변할 수 있다. 그러므로 body는 실제적인 시뮬레이션 경우에는 많은 surface를 가지며 간단한 body의 경우에는 하나의 surface로 표현될 수 있다.

5) 기타 정보(나이, 성별 등등)

BDP에는 다음 사항들이 기본적으로 가정되었다.

1 human body 모델을 기본 자세로 초기화한다.
- 이 때의 기본 자세는 두 발을 앞으로 향하고 두 팔은 손바닥을 안쪽으로 향하여 옆구리에 붙인 서있는 자세로 이 자세에서의 모든 joint angle은 0값을 가진다.

2 좌표 시스템을 구축한다.
- body 좌표 시스템의 원점은 관절로 이어진 내부 point로 spine origin이 된다.
- 좌표에서의 방위는 왼쪽으로 x point, 위쪽으로 y point, 뒤쪽으로 z point로 정의되었다.

3 calibration과 초기 parameter set이 body dimension을 구체화한다.

III. Text-to-Speech

최근들어 TTS는 interface와 다양한 multi-media application분야에서 많은 관심을 받고 있다. MPEG4 TTS는 입력된 text를 가지고 자연적 음성에서 추출된 prosodic정보를 이용하여 synthetic speech를 화면상 화상과 동기시켜 음성을 재생하도록 한다.

이 TTS를 사용하므로서 narration을 포함하고 있는 multi-media는 자연적인 음성을 녹음하는 불필요한 작업없이도 쉽게 구성되어질 수 있다.

더군다나 FA(FaceAnimation)/AP(Animation Picture)/MP(Moving Picture)와 interface를 가지는 TTS는 더욱 더 풍부한 내용을 수록할 수 있도록 한다. 현재 MPEG4 SNHC활동에서 TTS와 FA/AP/MP에 대한 TTS의 interface가 진행중이다. 이 확장된 TTS는 자연적인 음성에서의 prosodic정보를 이용하여 양질의 합성음성을 재생할수 있으며, interface와 bitstream format은 계층적으로 나누어져 있다. bitstream의 계층적 coding으로 prosodic정보를 가지는 몇 개의 parameter가 사용할수 없다 하더라도 재생시 그것을 어느정도 보완할 수 있게된다.

이러한 Scalable MPEG4 TTS interface에 대한 기본적 개념은 모든정보를 사용자가 요구하는 level에 맞춰 사용할수 있다는 것이다. FA(Facial Animation)와 interface하는 TTS의 일반적인 구조가 아래 그림에 나타나 있다.

이 구조에서 다음과 같은 형태의 interface로 나누어지는 것을 볼 수 있다.

1 Interface for Demux and TTS

Bit stream을 받은 Demux는 다음과 같은 stream을 TTS에 보낸다.
- Input type of TTS which specifies whether TTS is driven with Facial animation or with moving pictures.
- Control commands stream : Control command sequence.
- Input text : text to be synthesized.
- Auxiliary information : Prosodic parameters.
- Lip-shape patterns
- Information for trick mode with moving pictures

2 User interface for TTS

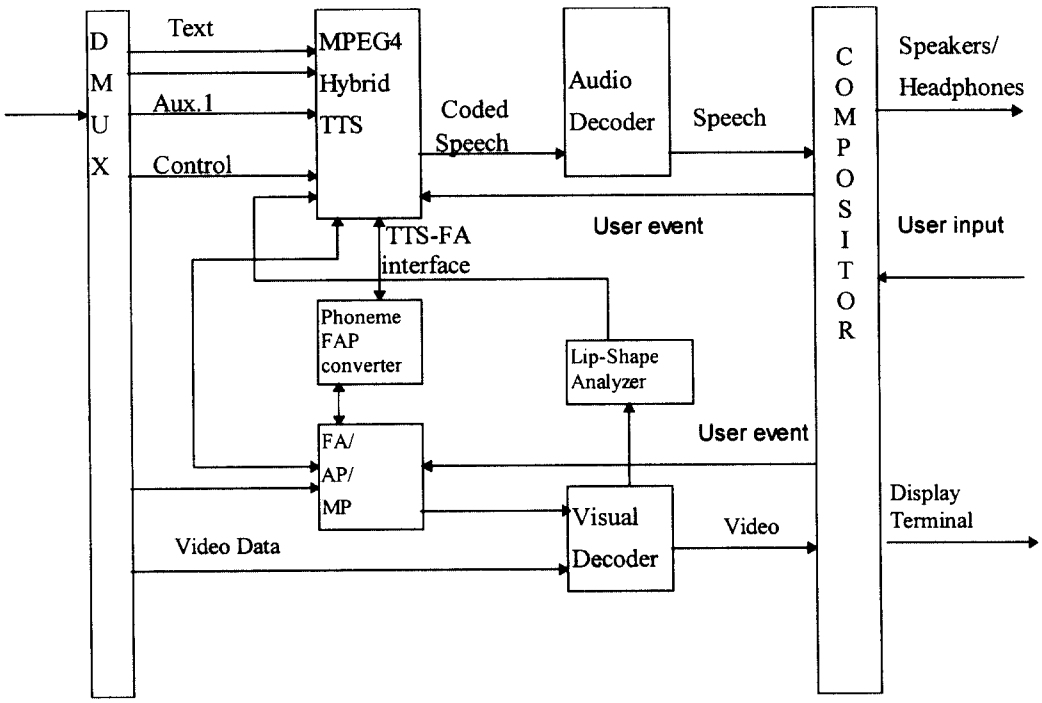

그림 5. MPEG4 TTS architecture

stop, play, voice등 user로부터 받은 명령으로 TTS command는 synthetic speech를 제어한다.

3 Interface for TTS and Face Animation module / Interface for Video Lip-shape Analyzer and TTS

MPEG4 framework상에서 TTS와 FA module은 아래의 두가지 경우에 동기적으로 구동되어 진다.

- TTS receives the lip shape information.
- TTS drives the facial animation module by using a set of phonetic parameters.

4 Interface for TTS and Audio Decoder

Audio Decoder로 보내지는 TTS의 출력은 encoded된 음성과 stop/play/volume control과 같은 몇가지의 기본적인 명령들을 포함한다. 이것은 coded된 자연음성과 audio decoder간의 interface와 비슷하다.

이상의 MPEG-4 TTS의 응용으로 다음의 Application Scenario(MPEG-4 Story Teller on Demand(STOD)를 구성해 볼 수 있다.

- STOD application에서 사용자는 hard disk나 CD에 저장된 story libraries database로부터 story를 선택한다.
- STOD 시스템은 MPEG4 facial animation과 상호작용하는 MPEG4 TTS를 통해 story를 읽게 된다.
- 사용자는 mouse나 keyboard와 같은 user interface를 사용해서 원하는 순간에 정지나 재생을 할 수 있다. 또한 사용자는 gender, age, volume, speech rate등을 선택해 조절할수 있다.

IV. Media Integration of Text and Graphics

MITG는 MPEG-1, MPEG-2, 그리고 MPEG-4와 앞으로의 표준에 부합되는 Video backgrounds에 text와 image, 그리고 graphics를 overlay하는 것을 말한다. Multimedia Presentation에 있어서 text/graphics data는 본래의 video/audio data와 함께 중요한 요소이다. 그래서 MPEG-4가 text/graphics data로 만들어진 Multimedia Presentation을 만들고, 통신하는 기능이 요구된다. 또 다른 중요한 요소는 이러한 text/graphics data를 layer된 spatial & temporal hierarchies에 맞추어 만들어 낼 수 있는 능력이다. 게다가 이러한 overlay 기능은 본래의 video/audio

backgrounds의 absence에도 이용될 수 있어야 한다.

위의 설명한 기능을 구현하는 BIFS(BInary Format for Scene description) nodes는 아래와 같다. 이 nodes는 MPEG-4 Scene안에서 text와 graphics object가 어떻게 구성되고, 스크롤되고 포함되는지를 정의하고 있다.

- Layout node : 이 node는 다양한 정렬 모드에서 child node의 layout을 지정한다.
- Scroll node : child node를 스크롤(Wrapping, Scrolling text, etc)한다.
- Text node : Text string(ISO/IEC 10646에 따른 UCS-2나 UTF-8 문자를 사용)을 포함한다.
- StreamingText node : text를 포함하는 bitstream을 갖는다.
- Font node : font style, size, hspacing/vspacing 등과 child node를 가지나 child node 중에 text child node만이 Font node의 영향을 받는다.
- Blink node : child node를 깜빡거리게 한다.
- Fade node : child node를 사라지게 한다.

다음은 MITG application에서 다루는 SNHC Aural/Visual Scene의 예이다.

V. View dependent Texture Coding

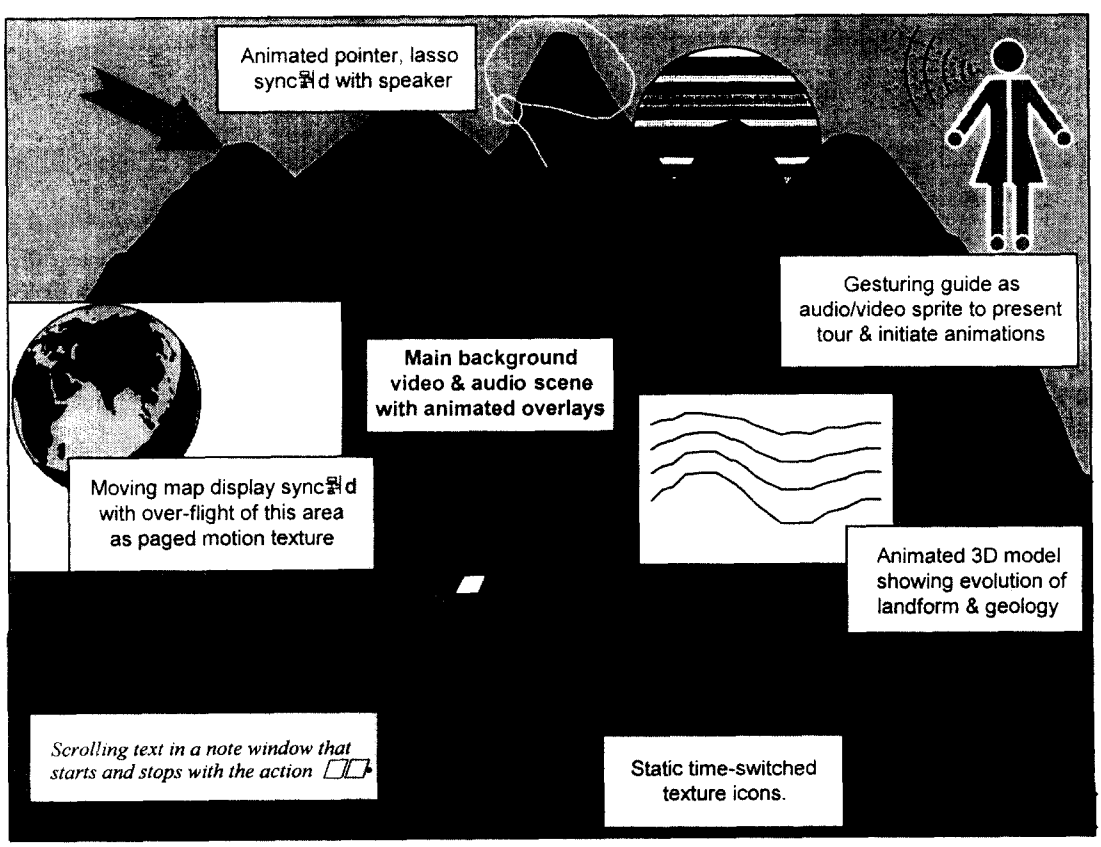

그림 6. SNHC Aural/Visual Scene (MITG의 예)

view dependent texture coding이란 viewpoint가 이동함에 따라 증가하는 texture data를 효과적으로 보내는 방법을 다룬다. 이 기술은 가상 환경으로의 low-delay, low-bandwidth 원격 접속을 허용하여 네트워크를 통해 실제적인 photo-texture data를 보내는 것을 가능하게 한다. decoder에서 coder로 data를 보내는 것은 back-channel을 통해서 이루어지는데 이것은 coder에 현재 viewing condition을 지시하는데 사용된다.

이러한 texture data의 전송시에는 quad mesh에 map된 각 texture 블록마다 아래의 작업을 수행하게 된다.

- Image를 Wavelet 변환 한다.
- 유용한 계수를 고른다.
- 부호화한다.
- 전송한다.
- 복호한다.
- 역변환한다.
- Grid Mesh에 map한다.

view dependent texture coding 투사 방법 또한 orthographic 투사와 perspective 투사의 두가지로 나눌 수 있는데 이 두가지 투사에 따른 구현 방법은 여기서 다루지 않는다. 기본적으로 texture coding 투사의 decoder쪽에서는 texture가 mapped된 3D regular grid mesh 정보를 사용하며, 그 texture mapping 작업은 grid mesh상에서의 정점의 좌표를 사용하여 정의된다.

VI. Static and Dynamic Mesh Coding

1. 2D dynamic Mesh Coding

위에서 언급한바와 같이 현재의 MPEG4에서는 OOP을 이용한 coding이 중요한 개념으로 자리잡고 있다. SNHC분야도 이 개념을 바탕으로 이루어져 있으며, 화면내의 사물(object)들이 각각 하나의 객체로 의미되어져 이들 객체를 중심으로 code/decode되어지기 때문에 불필요한 내용의 data의 전송이 줄어들게 되며 압축에 있어서 상당한 효과를 볼수 있게 되었다. 그러면 이절에서는 OOP를 바탕으로 하는 VO(Video Object)에서 적용되는 2D dynamic mesh coding에 대해서 간략히 알아보기로 하겠다.

1) 2D mesh geometry and motion (de)compression

mesh geometry는 I-VOP(Video Object Plane)에 대해서만 coding되어지며 VOP간의 coding에 대해서는 mesh node point의 motion이 coding되어진다. mesh motion vector를 coding하는 방법에는 두가지가 있다. 첫번째 (simulcast)기법으로 주위노드의 motion vector을 예측에 사용하는 방법이 있고 두번째 (scalable)기법으로는 block-based motion vector를 예측에 사용하는 방법이 있다. 두가지 기법 모두 coding할 motion vector의 예측치를 사용하며 실제예측 error는 VLC를 사용하여 전송되어진다.

(1) Mesh geometry compression

Mesh geometry는 node point의 집합인 Delaunay triangulation을 이용한 2D triangular mesh로 구성되어져 있다. node point 좌표인 Pn = (Xn,Yn)만이 coding되어지며 mesh triangular topology(links between node points)는 coding되지 않는다. 이들 node의 위치를 encoding하기 위해서는 nearest neighbor 기법을 사용하여 차례로 순회(traversal)하며 각 node의 위치는 바로전에 encoding된 node의 위치를 예측에 사용함으로써 encoding된다. node point의 순서는 node가 방문되어진 순서에 따른다. 바로전 node와 현재 node position사이의 차이는 entropy coder를 사용하여 encoding된다. 경계선을 이루는 node들이 먼저 방문되어지고 내부 node들이 그다음 순서로 방문된다. 전체 node point의 개수와 경계를 이루고 있는(boundary) node point의 개수를 전송함으로써 decoder측에서 몇 개의 node point와 boundary node point가 뒤따라 올지를 알 수 있다. 이것으로 모든 node와 polygonal boundary의 재구성을 할수 있게 된다. 이러한 triangular mesh의 예가 다음 그림에 나타나 있다.

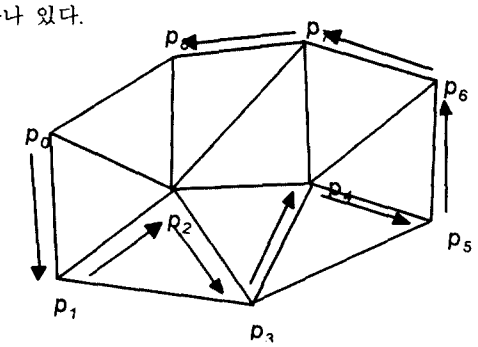

그림 7. Traversal and ordering of a 2D triangular mesh

그림7에서 먼저 node의 수와 boundary node의 수가 encoding된다. 그 다음으로 왼쪽위(top-left)에 있는 node P0가 예측없이 coding되어진다(P0 = (X0,Y0)는 minimum Xn + Yn을 가진 node n으로 정의되어질 수 있다). 다음 순서는 시계방향순으로 boundary node P1이 P1과 P0사이의 차이값으로서 encoding된다. 이런 순서로 다른 boundary node들이 같은 방법으로 encoding되어진다. 그리고 마지막 boundary node와 가장 가까운 encoding되지 않은 내부 node가 발견되고 이들 사이의 차이값이 encoding된다. 이 node는 minimum |Xn - Xlast| + |Yn - Ylast|을 가진 아직 coding되지 않은node n으로 정의되어질 수 있다(여기에서 (Xlast, Ylast)는 바로전에 encoding되어진 node의 좌표를 나타낸다). 모든 node point는 X와 Y좌표를 가지며 Pn = (Xn, Yn)으로 나타낼수 있다. 이들은 바로전 coding된 node point좌표와의 차이를 구해 이들 결과값을 VLC를 사용하여 encoding한다. 그러면 decoder측에서는 단순히 바로전 decode된 node의 위치값을 받아 각 차이값을 더하여 현재node의 위치를 decoding 할 수 있다. 모든 노드의 자리는 receiver측에서 재생되며, constrained Delaunay triangulation이 triangular topology를 얻는데 사용되어진다.

(2) Coding of node motion vectors with spatial prediction

VOP(Video Object Plane) k에서 2D mesh의 각 node point Pn은 2D motion vector Vn을 가진다. Node point motion vector의 공간예측(spatial prediction)은 같은 mesh내에서 encoding된 node point motion vector를 사용하여 예측하는것을 기본으로 한다. 즉 triangle Tk = 〈Pl, Pm, Pn〉중 하나인 node point Pn의 motion vector를 encoding하기위해서 두 개의 node Pl과 Pm의 motion vector Vl과 Vm이 Vn의 예측에 사용되어진다. 그리고 세 개의 node motion vector가 모두 encoding되어진 initial triangle Tk가 있다면, Tk내 두개의 node를 함께 가지는 이웃하는 triangle Tw가 적어도 한 개 이상은 있어야 한다. Tk와 Tw가 공통으로 가지는 두 개의 motion vector가 이미 encoding되어졌기 때문에 Tw의 세 번째 node motion vector는 나머지 두 개의 motion vector를 예측에 사용함으로써 encoding되어진다. Prediction error vector(Vn - Wn)는 VLC로 encoding되어진다. 여기에서 breath-first traversal algorithm을 사용하여 모든 triangle을 방문한다. Mesh의 topology를 미리 알고있기 때문에 순회(traversal)는 encoder와 decoder에서 같은 방식을 사용하여 수행되어질 수 있다. Breadth-first traversal 과정이 아래 그림에 나타나 있다.

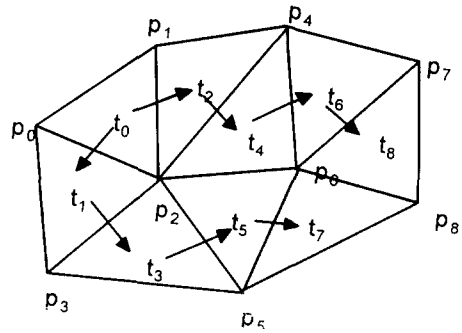

그림 8. Breadth-first traversal in a triangular mesh

Initial triangle은 위에서 정의한것과 같이 왼쪽 위(top-left)에 있는 node를 사용하여 정의되어 진다. 왼쪽위(top-left)에 위치한 node는 mesh의 경계선을 이루고 있으며, 경계선상의 시계방향으로 다음에 위치한 node가 top-left triangle의 한 부분을 차지하고 있다.

Breadth-first traversal은 항상 바로전 triangle 의 오른쪽에 있는 triangle을 방문한다(위 그림에서는 T1이다). 한번도 방문되지 않은 triangle이 방문되어지며, 그중 coding되지 않은 node motion vector만이 encoding되어진다.

(3) Coding of mode motion vectors with prediction from block-based motion vector

현재 표준 video encoder 기법과 MPEG4 video VM은 block-based motion compensation을 사용한다. 16x16 inter macroblock에는 한 개 혹은 네 개의 motion vector가 할당된다. 이 block-based motion vector는 mesh node motion vector coding에서 predictor로서 사용 되어진다. 즉, VOP k에서 VOP k+1로의 node point Pn의 motion vetor Vn을 encoding하기위해서 다음의 단계가 수행된다. VOP k+1에서 Pn을 포함하는 macroblock를 찾은다음 이 block과 이웃 block의 motion vector들을 찾는다. 그리고 이들 block motion vector들의 중간값을 계산하고 그것을 negating하여 prediction vector Wn을 구한다. Prediction error vector Vn - Wn은 VLC coding에 의하여 encoding되어진다. Decoder측에서는 VOP k+1내

에서 같은 macroblock를 찾아서 예측값을 계산하고 node motion vector를 얻기위해 그 예측값을 decoding 된 prediction error vector에 더한다. 위의 단계가 아래의 그림에 나타나 있다.

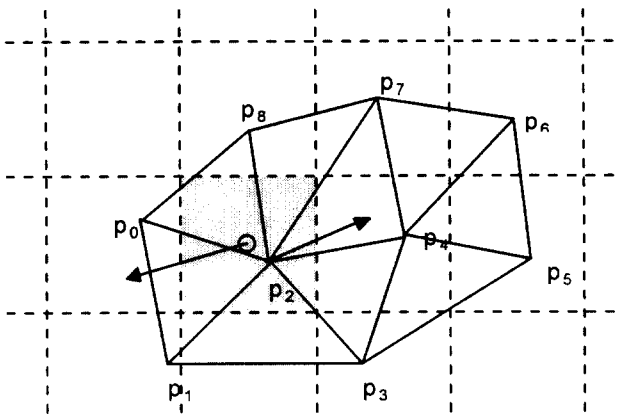

그림 9. The relation between mesh nodes and block motion vectors in node motion vector prediction

위 그림은 node motion vector prediction에서 block-based motion vector로부터의 mesh node와 block motion vector과의 관계를 보여주고 있다. P2의 node motion vector는 VOP k에서 k+1로의 motion vector로 정의되어있다. VOP k+1과 대응하는 block과 block motion vector가 위 그림에서 검게 칠해져 있는 것을 볼 수 있다.

2) Encoder and decoder architectures

여기에는 mesh geometry 와 motion을 압축하는 보조 encoder와 video object shape와 texture를 압축하는 base layer로서의 MPEG4 video 압축 tool을 사용하는 base encoder의 functionality-scalable multimedia encoder 구조에 대해서 알아보겠다. 이 구조는 mesh object의 정의에의해 제공되는 확장 기능들이 필요없는 경우에는 표준 video VM encoder만의 사용도 가능하게 한다. 이들 encoder/decoder 구조는 아래 그림에 잘 나타나 있다.

표준 decoder에서 demultiplexer는 mesh geometry, node point motion vectors, block-based motion vector 및 shape와 video object texture를 분리해낸다. Video object texture가 표준 video VM decoder에서 decode되어지는 동안에 mesh geometry와 node motion bits는 보조(auxiliary) decoder에서 decode된다. 2D mesh object의 움직임(animation)은 각 frame에서 2D mesh geometry 정보를 이용함으로써 수행되어진다. Mesh object interface는 user control parameters와 node point motion vector를 사용하여 2D mesh object를 재생시키게 된다.

VII. MEPG-4의 적용

1. 화상회의

우리는 SNHC의 face/body animation application의

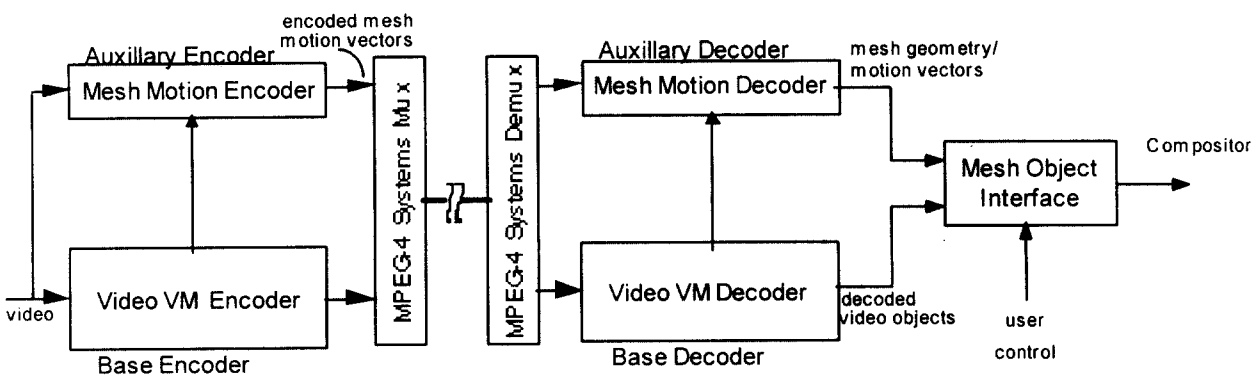

그림 10. Scalable mesh object coding

예로 Virtual Meetiong을 생각할 수 있다. 현재 3D synthetic face로 표현된 간단한 가상 공간에 일곱명이 virtual meetiong을 하고 있다. 아래 그림에서 보이는 대로 일곱 개의 Terminal은 MCU(Multipoint Control Unit)로 연결되어 이중 한 사람은 자신의 데이터를 전송하고 나머지 여섯명의 데이터를 받게 된다. 이 때 animated face를 전송해야 하므로 처음에 FDP를 전송한 후에는 FAP와 코드화된 음성만으로 통신이 이루어 지게 된다. 이 예에서는 각각 2kbps로 FAP와 코드화된 음성이 전송되므로 자신의 Terminal과 MCU는 실제로 28kbps channel로 연결되게 된다.

2. VRML과의 인터페이스

VRML(Virtual Reality Modeling Language)은 말 그대로 3D 가상 공간을 모델링하는 언어로 인터넷상에서 HTML(Hyper Text Markup Language)의 2D에 대응되어 Web상에서 Brower를 통해 VR(Virtual Reality) System을 구축해준다. 이러한 VRML은 MPEG-4 snhc에서의 face와 body의 geometric surface/texture coding에 사용될 것으로 보인다. 실제로 VRML 모델링에서 3D object를 만들 때 VRML 전문 tool을 이용하면 VRML 2.0 규약의 기능을 잘 살릴 수 있으나 모델링 기능이 일반적인 3D 그래픽 프로그램보다 약하기 때문에 3D 그래픽 프로그램을 사용해서 모델링한 후에 만들어진 파일 포맷을 VRML 파일 포맷으로 변환시켜 주는 기능도 고려되고 있다. 이 부분에서 고려될 수 있는 3D 그래픽 엔진을 몇가지 소개하면 SGI의 openGL과 VRML converter기능을 제공하는 RenderWare, 그리고 인텔의 3DR, MS의 다이렉트 3D등을 들수 있다. 이 중 openGL은 2D/3D interactive 그래픽 프로그램 인터페이스로 운영체제나 하드웨어 platform의 제한이 적어 여러 가지 다양한 목적으로 사용될 수 있는 일종의 그래픽 라이브러리이다.

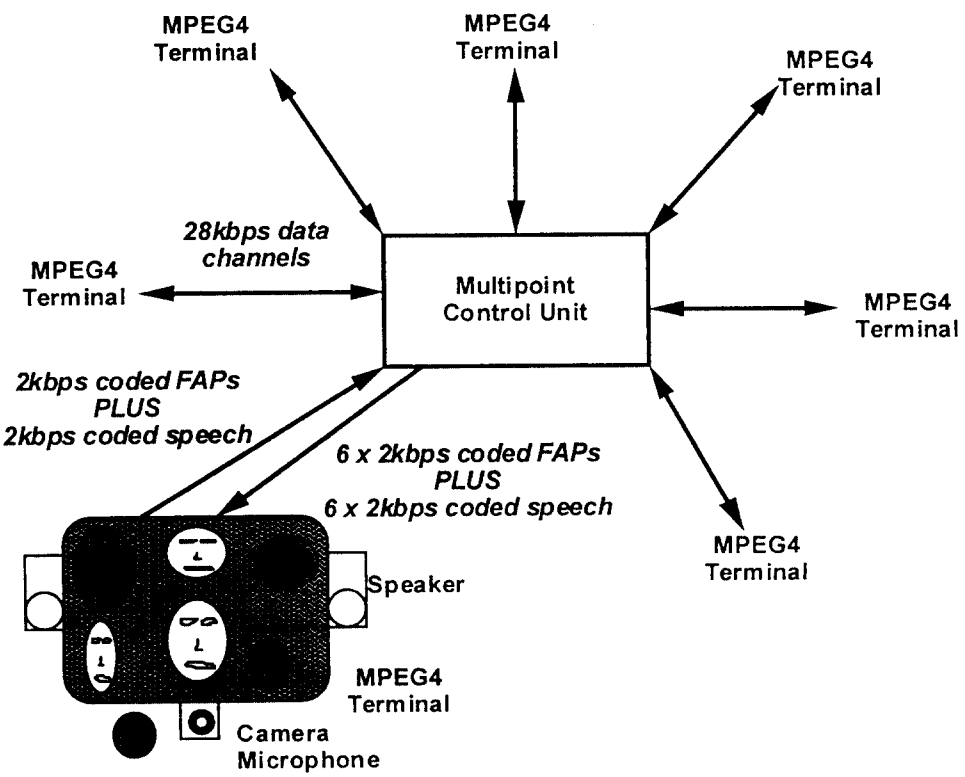

그림 11. Virtual meeting application example

3. 관련 정보

SNHC VM과 Core Experiment, 그리고 WD문서는 MPEG Web site에서 찾아 볼 수 있다. 그러나 현재의 test data나 VM software는 충분히 검증되지 않은 상태여서 공개되지 않고 있다. 아래에 MPEG-4 SNHC Web site URL이 있다.

- MPEG Wed site - http://www.cselt.stet.it/mpeg
- MPEG-4 Systems/MSDL Web site - http://www-elec.enst.fr/msdl/msdl.html
- MPEG-4 SNHC Web site - http://www.es.com/MPEG-4-snhc

그리고 보다 많은 정보를 얻기 위해서 아래의 메일링 리스트를 이용할 수 있다.

- MPEG-4-snhc@es.com

VIII. 맺음말

초기에 동영상과 오디오 데이터를 효율적으로 코드화하고 표현하는데 목표를 두었던 MPEG-1에서부터 MPEG관련 하드웨어와 소프트웨어는 PC user들로부터 많은 사랑을 받아오고 있다. 이것은 표준화된지 얼마 지나지 않아 얻어진 큰 성과로 받아들여지고 있으며 user들의 관심을 사는 중거로 볼 수 있다. 지적재산권 문제로 불거진 Mpeg2 layer3(mp3) 오디오의 경우에 탁월한 음질로 통신망 사용자들에서부터 일반 user들에게까지 MPEG를 알리는데 크나큰 공헌을 한 것이 사실이다. 이 mp3 오디오 데이터 파일의 경우에는 최근에 유료로 전환되어 다운로드를 기다리고 있다. 이와같이 MPEG은 여기서 다룬 MPEG-4와 정보검색을 위한 내용 표현을 다루는 MPEG-7으로 또 한번 우리를 한층 더 발전된 멀티미디어 환경으로 이끌어 주리라 기대한다.

참고문헌

[1] ISO/IEC JTC1/SC29/WG11 N1730 : MPEG-4 Overview
[2] ISO/IEC JTC1/SC29/WG11 N1666 : SNHC Verification Model 4.1
[3] ISO/IEC JTC1/SC29/WG11 N1820 : SNHC Verification Model 5.0
[4] ISO/IEC JTC1/SC29/WG11 N1669m3 : SNHC Frequently Asked Question v1.0
[5] Stephen N. Matsuba and Beruie Roehl, USING VRML, QUE
[6] Richard S. Wright Jr and Michael Sweet, openGL superbible, The Waite Group Inc.

최 석 림

- 1977년~81년 : 서울대학교 전자공학과 학사
- 1981년~83년 : 서울대학교 전자공학과 석사
- 1987년~92년 : Syracuse University 전자공학 박사
- 1983년~85년 : 한국방송공사 기술연구소 연구원
- 1993년~96년 : 현대전자 수석연구원
- 1996년9월~현재 : 세종대학교 전자공학과 부교수