主題

TTS를 이용한 멀티미디어 통신서비스

광운대학교 전자공학과 안병호, 유지상
한국전자통신연구원 이승훈, 김상훈

차 례

I. 서론
II. TTS 및 아바타의 구조와 원리
III. TTS연동 아바타를 이용한 멀티미디어 통신서비스 응용
IV. 결론

I. 서 론

음성은 인간의 의사소통 수단으로 가장 편리하게 사용되는 매체이다. 따라서 지난 수십년간 인간과 컴퓨터와의 자연스러운 통신을 위해 음성언어를 이용한 휴먼인터페이스 기술을 연구해 왔고, 최근에는 제한된 영역이지만 이러한 기술이 실생활에 적용되기 시작하고 있다. 특히 TTS기술은 음성의 어느 기술보다 상용화에 더 접근된 기술이며 각종 텍스트 정보를 음성으로 변환하는 서비스에 응용할 수 있어 낮은 비트율을 요구하는 분야에 효율적인 음성전달의 방법이 될 수 있다. 이와 함께 동영상 압축의 국제표준인 MPEG에서는 자연영상/음향의 효율적인 고품질 압축/복원 기술을 중심으로 이미 표준화가 완료된 MPEG-1이나 MPEG-2와 달리 현재 표준화가 진행중인 MPEG-4의 경우 자연영상/음향뿐만 아니라 합성영상/음향도 그 대상으로 하고 있으며 압축/복원 방식도 구성요소별로 최적의 기술을 적용할 수 있도록 지향하고 있다. 이러한 MPEG-4의 표준화대상에는 FA(Face animation), BA(Body animation), 합성음향, TTS등이 포함되어 이를 이용한 화상회의의 방법, 가상환경에의 적용 및 그에 관한 효율적인 통신방법이 연구되고 있는 실정이다.

본 논문에서는 TTS와 MPEG-4 기준을 따르는 아바타의 구조와 원리에 대해서 살펴보고 TTS와 아바타를 연동한 멀티미디어 통신 서비스의 응용 예를 살펴보고자 한다.

II. TTS 및 아바타의 구조와 원리

1. TTS의 구조 와 원리

TTS기술은 크게 언어처리, 운율처리, 신호처리 3부분으로 나눌 수 있다. 〈그림 1〉

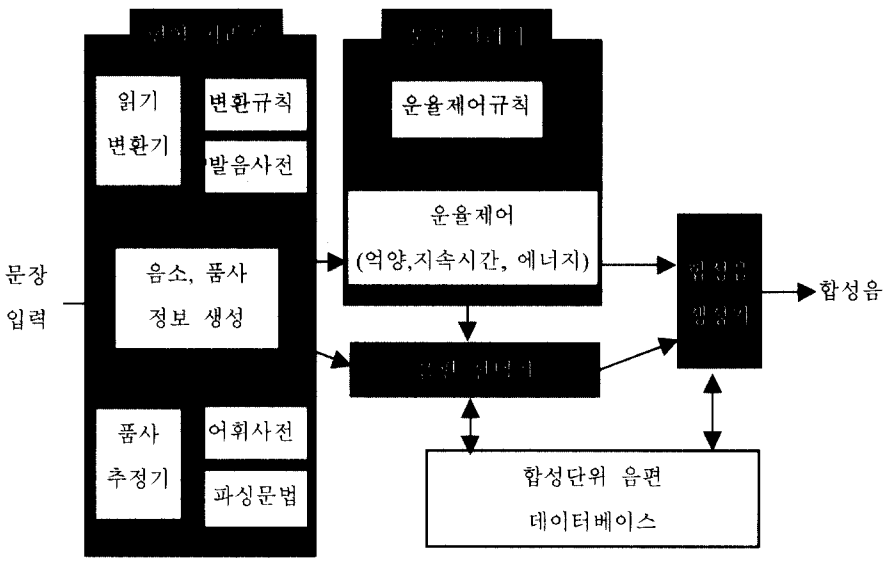


그림 1. TTS의 구조

1) 언어처리

언어처리부에서는 소리생성을 위한 음소와 운율정보를 텍스트에서 추출하며 한글이외의 문자를 처리하는 전처리와 발음나는대로 변환하는 글자/음운변환기, 문장의 구조를 분석하는 구문분석기로 이루어져 있다. 전처리에서는 심볼, 영문자, 한자, 숫자 등을 한글로 변환한다. 형태소 분석기에서는 어휘사전을 이용하여 한글로 변환된 텍스트를 형태소 단위로 분리하게되며, 각 단어에 대해 한글 텍스트, 품사, 형태소 분리 결과, 형태소의 종류 등에 관한 정보가 출력된다. TTS 시스템에서 구문분석기는 자연스러운 운율정보를 생성하기 위해 문장의 구문정보를 추출해 내는 역할을 한다. 운율생성을 위한 정보는 각 구의 위치, 문장성분, 상호 결합관계 등을 말한다. 문장성분 추출은 형태소 분석결과를 이용한다. 글자/음운변환기는 한글 읽기규칙을 구현한 것으로서, 음성합성시스템이 한글을 올바르게 읽을 수 있도록 소리나는대로 바꿔주는 것인데, 기본원칙은 교육부에서 고시한 표준어 규정의 표준발음법을 따른다.

2) 운율처리

운율처리는 운율정보를 이용하여 실제 억양의 컨투어, 음소 지속시간, 에너지 컨투어, 휴지길이를 생성하는 과정이다. 휴지구간은 각 문장, 절, 구, 어절 경계에서 오게 되는데 합성음의 의미전달, 자연성에 매우 중요하다. 휴지구간 예측은 어절간 품사정보의 관계로부터 끊어읽기 강도를 예측한다. 일반적으로 끊어읽기 강도는 3~5가지 단계로 나눌 수 있으며, 어절간 품사정보의 bigram, trigram 또는 HMM 방법을 사용하여 예측한다.

음소지속시간 제어규칙은 단일 화자가 발성한 여러 개의 문장에 대해 관측된 지속시간을 맞추도록 규칙이 작성된다. 규칙의 작성은 문장의 구조와 음운환경에 따라서 각 합성단위의 지속시간이 신축되는 비율과 듣는 사람이 변별적 자질을 인지할 수 있는 최소 지속시간 제어를 할 수 있는 방향으로 이루어진다. 그리고 발성속도, 리듬의 변화에 따른 지속시간 제어를 위한 규칙도 작성된다[1].

억양제어 규칙으로 주로 사용되는 Fujisaki 모델은 구 제어(Phrase control) 명령어와 악센트 제어(Accent control) 명령에 따라 임펄스와 스

텝 입력이 가해져 저대역 필터된 곡선이 생성되는데 이 곡선이 예측된 기본주파수가 된다. 최근에는 억양모델링 방법에 있어서도 변화를 가져왔는데 ToBI 시스템과 같은 대량의 운율 데이타베이스로부터 통계적 분석에 의한 억양 패턴의 정형화 및 심볼화를 통하여 분석 및 규칙화를 하고 있다[2].

3) 신호처리

연결합성방식에서의 스펙트럼 생성기는 합성 데이터베이스로부터 합성단위를 추출, 연결하고 운율값(피치, 지속시간, 에너지)을 합성단위에 적용하여 운율을 실현하는 것으로 근래에는 TD-PSOLA (Time Domain-Pitch Synchronous Overlap & Add) 방식이 주로 사용된다[3]. 이 방식은 파라미터 합성방식보다 합성음이 명료할 뿐만 아니라 자연성을 대폭 향상시킬 수 있다. 최근에는 대용량 운율 데이터베이스를 이용한 합성방식이 제안되어 복수개의 합성단위 중 앞뒤 연결왜곡 및 운율왜곡이 최소가 되게 최적 합성단위를 선정, 합성단위에 운율조절을 하지 않고 직접 연결하여 합성한다. 특히 합성단위를 자동으로 제작하여 새로운 화자에 대한 합성음을 짧은 시간내에 생성할 수 있다.

2. 아바타생성의 구조와 원리

아바타란 대리자의 개념으로 화상회의나 가상공간안에 자신의 대리자를 참여시키는 방법으로서 현재 MPEG-4 SNHC그룹[4]등에서 활발한 연구 및 그에 관한 표준화가 진행되고 있다. MPEG-4 첫 번째 version의 표준안에서는 얼굴 객체(face object)를 정의하여 수용하고 있으며 FDP(Face Definition Parameter), FAP(Face Animation Parameter)라는 얼굴정의 파라미터와 얼굴 애니메이션 파라미터들로 얼굴 객체를 기술하고 있다.

1) FDP, FAP 및 FAPU의 정의 및 구조

FDP는 얼굴의 정면에서 얻어진 81개의 feature들의 point로 정의되어 있는데 각 feature point는 X, Y, Z축의 3차원 공간에서의 위치 좌표로 정의된다. MPEG-4 SNHC 그룹에서 현재까지 제공되고 있는 FDP는 총 6개의 set으로 이, 혀 등이 제외된 61개의 feature point가 제공되는 jim_n, claude_n, chen_n 등의 3개의 set와 모든 feature point가 제공되는 charles, MIRAFace, ISTmodel 등 3개의 set으로 되어 있다.

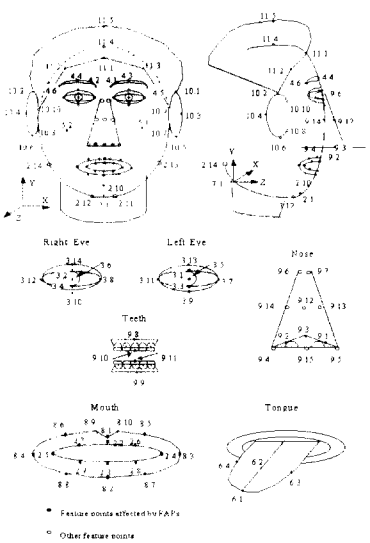


그림 2. MPEG-4에서 정의된 FDP(Face Definition Parameter)

FAP는 FDP의 움직임 정보를 가지는 66개의 low level parameter와 음운에 따른 얼굴의 움직임, 표정에 따른 얼굴의 움직임을 나타내는 2개의 high level parameter로 구성되어 있다. 또한 FDP가 달라짐에 따라 FAP 값의 올바른 동작의 보장을 위해 6개의 FAPU(Face Animation Parameter Unit)을 FDP의 좌표 값으로부터 계산하여 정의하고 FDP, FAP를 이용한 애니메이션의 기본 단위로 사용하게 된다.

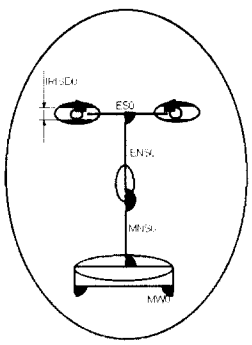


그림 3. FAPU(face animation parameter units)

FAP는 각 FDP에 대응하는 움직임 정보를 FAPU의 값으로 가지고 있기 때문에 각 frame을 구성하는 FAP에서 mask에 의해 활성화되는 FAP값을 사용하여 FDP의 값을 바꾸어 주게된다. MPEG-4 SNHC FBA 그룹에서 제공하는 FAP data는 입의 움직임과 얼굴의 표정을 포함하는 10여개 정도이며 그림 4에 그중 jim_n.fdp와 emotion.fap에 의해 만들어진 얼굴 및 애니매이션된 300번째 frame을 보였다.

2) OpenGL을 이용한 얼굴객체의 구현

OpenGL은 API(Application Programming Interface)함수로 구성되어 있는 C언어 라이브러리로 고속 rendering기능과 texture mapping 등의 다양한 그래픽 기능을 내장하고 있고 여러 platform에 porting되어 광범위하게 쓰이고 있으며 각 platform에 알맞게 짜여진 보조 toolkit과 함께 사용되어 진다[5].

OpenGL을 사용하여 얼굴 객체를 구현하기 위해서는 보조 feature point들의 좌표를 추가하고 보조 좌표와 FDP를 사용하여 삼각형 mesh 정보를 생성해 내야 한다. 이 삼각형 mesh 정보를 glVertex3f라는 함수를 통해 OpenGL에 전달해 주고 얼굴의 각 부분에 대한 적당한 색 정보를 glColor3f함수로 전달해 주어 얼굴 객체를 rendering 시키게 된다. 이를 통해 rendering 되어진 얼굴 객체는 FAP의 각 frame을 읽는 부분과 각 frame FAP정보에 따라 FDP를 변화시키는 과정을 거치게 되며 이 과정을 반복하여 변화되어진 정보를 가지고 rendering시킴으로 애니메이션이 되어진다. 〈그림 5〉

3) 사실감 있는 얼굴 객체의 구현

그림 5에서와 같이 3차원 그래픽을 통해 얻어진 얼굴 객체는 응용 시스템에서 활용될 때 사실감이 떨어지는 단점이 있다. 이 문제를 해결하고 자연스

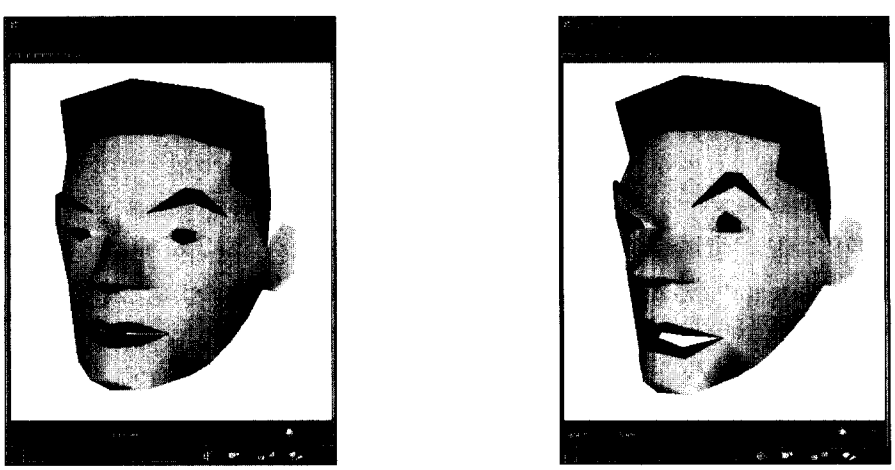


그림 4. FDP와 FAP를 이용한 애니메이션의 한 프레임

그림 5. OpenGL을 이용하여 애니메이션된 Charles라는 얼굴의 모습

러운 객체를 구현하기 위해서 3차원 그래픽으로 생성된 객체에 조명 효과와 smooth shading 기능 등 다양한 방법을 적용하게 되는데 가장 효과적인 방법이 실세계에서 얻어진 영상이나 합성영상을 객체의 표면에 붙여 사실감을 향상시키는 텍스쳐 맵핑(texture mapping) 방법이라 할 수 있다.

MPEG-4에서는 얼굴 객체의 구현과 관련하여 추가 사항으로 얼굴의 텍스쳐를 사용할 수 있도록 정의하고 있으나 MPEG-4 SNHC에서 제공되는 FDP에는 얼굴 객체를 생성할 때 사용하는 텍스쳐와 텍스쳐 맵핑에 필요한 좌표 정보를 제공하지 않는다. 따라서 본 논문에서는 기준 FDP를 정하고 이 기준 FDP로부터 텍스쳐의 좌표 정보를 정의하여 새로운 FDP로 변형시키는 방법을 사용하여 변형된 FDP에 위에서 언급한 것과 동일한 방법으로 얼굴 객체를 생성하고 애니메이션을 구현하는 방법을 적용하였다. 이 과정에서 텍스쳐 정보를 얻을 수 없고 특히 애니메이션 구현 시 필요하지 않은 머리 뒷부분의 텍스쳐 맵핑은 제외시켰다. 또한 텍스쳐 맵핑의 속도향상과 효과적인 텍스쳐 맵핑을 위해 위에서 정의한 대부분의 보조 좌표를 이용하는 대신 텍스쳐 맵핑에 중요한 요소가 되는 얼굴의 경계 부분에 10개, 이 부분에 8개, 혀 부분에 4개의 보조좌표를 새로이 추가하여 총 103개의 FDP 좌표를 사용하였다. 그리고, 텍스쳐 맵핑을 위해 얼굴정면 사진을 스캐닝하여 256 x 512 크기의 bitmap 형태의 얼굴 텍스쳐 영상을 얻었으며 이들 텍스쳐로부터 FDP와 새로이 추가한 보조 좌표의 X축과 Y축 성분 값을 찾아서 텍스쳐 좌표 정보로 사용하였다[6].

위에서 얻은 텍스쳐와 텍스쳐 좌표 정보를 사용하여 FDP의 값이 주어진 텍스쳐에 따라 변형되는데 얼굴의 모양을 결정하는 중요한 요소인 X축과 Y축의 정보는 이미 텍스쳐 좌표 정보에 있으므로 기준 FDP의 X축과 Y축 성분 값은 텍스쳐 좌표 정보의 X축과 Y축 값으로 대치하고 Z축 정보는 기준 FDP의 정보를 변경하지 않고 그대로 사용하였다. 이렇게 변형된 FDP는 텍스쳐의 모양을 가능한 한 원형 그대로 반영할 수 있으므로 얼굴의 형태를 정확하고 자연스럽게 표현할 수 있도록 해준다. 텍스쳐 정보에 따라 적절하게 변형된 FDP는 위에서 이미 언급한 OpenGL 기반 얼굴객체 생성 방법과 애니메이션 구현 방법을 사용하여 얼굴 객체를 만들고 애니메이션 하는데 사용되었다. 그림 6에서는 텍스쳐 맵핑 기술을 사용하여 애니메이션 된 얼굴 모습중 하나의 프레임을 보여주고 있다.

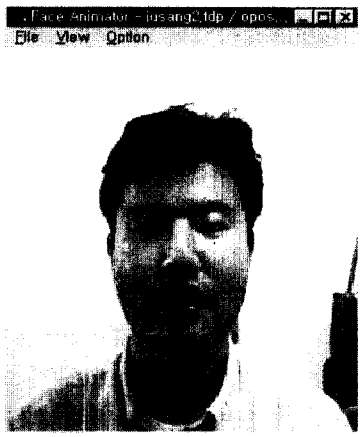


그림 6. 텍스쳐 맵핑을 통해 애니메이션된 얼굴모습

3. TTS 연동 아바타

아바타를 TTS와 연동시키기 위해, TTS 프로그램이 동작할 때 입 모양의 애니메이션을 위해 미리 정해진 개수의 입 모양 패턴 bitmap 영상에서 FAP를 찾아내었으며 이들 정보를 파일의 형태로 미리 저장하였다. 저장된 FAP 화일은 TTS 프로그램의 동작과 함께 얼굴 객체가 생성되는 과정에서 FDP 파일과 함께 내부적으로 불려지며 이 때 TTS에서 사용되는 언어에 대응되는 입 모양 FAP를 자동으로 선택할 수 있도록 만들었다.

본 논문에서는 FAP 값을 수동으로 구하기 때문에 입 부분의 동작을 정의하는 16개의 FAP를 윗입술, 아래입술, 입술 양끝의 움직임으로 분리하고 동작에 따라 6개의 그룹으로 나누어 같은 FAP를 할당하고 계산함으로써 단순화된 FAP 정보를 만들 수 있었다. FAP의 값은 TTS에서 사용되는 입 모양 종류와 음운의 개수에 따라 달라지는 데 현재는 한국어 86개, 영어와 일어에 대해 각각 114, 18개의 음운에 대응되는 FAP을 만들어 사용하였다. 표 1에 입 모양에 해당되는 FAP의 6개 그룹을 나열하였다[7].

그룹 1	윗입술의 세로축 움직임
	8.1, 8.5, 8.6, 8.9, 8.10, 2.2, 2.6, 2.7
그룹 2	아래입술의 세로축 움직임
	8.2, 8.7, 8.8, 2.3, 2.8, 2.9
그룹 3	입술 양끝의 가로축 움직임
	8.3, 8.4, 2.4, 2.5
그룹 4	입술 양끝의 세로축 움직임
	8.3, 8.4, 2.4, 2.5
그룹 5	윗이의 세로축 움직임
	9.8, 9.10
그룹 6	아래이의 세로축 움직임
	9.9, 9.11

표 1. 입모양 FAP의 6개 그룹

TTS와의 연동시에 정확하고 자연스러운 애니메이션을 위해 입 부분의 움직임에는 두 음운 사이에 존재하는 움직임 양을 계산하여 선형보간법을 이용하여 보간을 하였고 얼굴 전체의 상하좌우 움직임도 적절하게 구현하여 자연스러움을 더하였다.

Ⅲ. TTS연동 아바타를 이용한 멀티미디어 통신서비스 응용

TTS를 이용한 아바타의 효과적 이용을 위해 standalone형태가 아닌 네트워을 이용한 다자간의 통신을 구현해 보았다. 최대 4명까지의 참가자가 각자의 아바타와 TTS를 이용하여 발화자가 글자입력창을 통해 입력한 글을 듣게 되는 환경이다.

구현된 시스템은 클라이언트/서버의 구조로서, 하나의 서버에 다수의 클라이언트가 접속할 수 있도록 구성하였다. 기본적으로 TCP/IP를 이용하고 임의의 포트를 지정하여 클라이언트들이 접속할 수 있게 하였다. 일반적인 화상회의는 많은 정보를 실시간적으로 보내야 하기 때문에 얼마간의 손실이나 오류가 있어도 UDP를 사용할 수밖에 없었다. 그러나 이번에 구현된 시스템에서는 주고받는 정보가 1KByte

이내 이므로 정보의 정확한 전달을 보장하는 TCP를 사용하여도 실시간적인 회의 환경이 이루어 질 수 있다[8].

그림 7에서 보듯이 서버의 기능은 중계자의 입장으로서 클라이언트가 정보를 보내오면 모든 클라이언트들에게 똑같이 전송시켜주는 기능과 함께 회의의 상황을 보여주게 된다. 서버는 무조건적으로 접속된 클라이언트들에게 정보를 중계하게 되므로 회의 도중 새로운 클라이언트가 접속하여도 무리 없이 새로운 참석자의 참여를 이룰 수 있고, 회의 도중 회의를 그만두는 참석자가 발생하더라도 그 즉시 모든 회의 참여자에게 그 사실을 알릴 수 있다. 새로운 회의 참가자가 발생했을 시, 그 동안의 대화 내용을 새로운 참가자에게 알려주기 위해 서버는 회의의 내용을 저장하게 되고 또한 서버는 대화를 감시하고 있다가 두 참가자가 동시에 발언을 하거나 한 참가자의 발언 중에 다른 참가자가 발언을 할 경우 기다리게 하는 등 회의의 중재기능도 가지고 있다. 이러한 서버의 기능을 효과적으로 하기 위해 서버는 정보를 보낸 클라이언트에게도 똑같은 정보를 보냄으로서 서버의 저장 및 중재 기능의 제어를 받게 하였으며 클라이언트는 자신이 생성한 정보가 아니라 무조건 서버에게 전달받은 정보만으로 정보를 갱신하도록 구성하였다.

클라이언트는 아바타와 TTS 그리고 텍스트 입력창으로 구성되어 있으며, 화자는 이 텍스트 입력창을 통하여 발언을 하게 된다. 회의 시작 시 자신의 아바타를 선정하거나 캡쳐하여 서버에게 전달함으로써 자신의 회의 참석여부를 알리는 초기화를 수행하여야 한다. 이를 구현하기 위하여 network 패킷 안에 63Byte의 헤더를 두어 캡쳐한 얼굴의 BMP와 FDP 화일, 미리 지정해 놓은 아바타의 사용여부 그리고 화자의 발언이 담겨있는 패킷인지를 구분하도록 하였다. 또, Network전송시 무리를 주지 않기 위해 7000Byte 이상의 정보는 총 정보의 Byte 수를 7000으로 나누어 계산된 패킷 수 만큼의 정보를 분리시켜 전송하였다. 그러기 위해 정보의 총 Byte 수와 이번 패킷에 전송하는 패킷의 양을 헤더에 표시하여, 수신시 정보를 병합하는데 이용하였다. 그리고 1Byte의 Boolean 형태의 검사 부분을 헤더에 첫 부분에 두어 참가자가 회의 도중 회의에서 나갈 경우에 대비하도록 하였다. 만일 참가자가 자신의 얼굴에 대한 BMP 화일을 전송하지 않고 미리 만들어져 있는 아바타를 사용하고자 할 경우 패킷의 종류를 SELECT로 하여 전체 크기 부분에 선

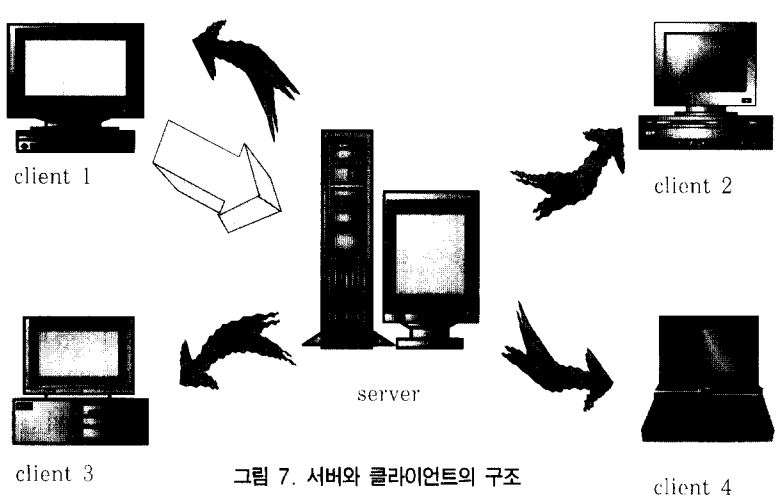


그림 7. 서버와 클라이언트의 구조

접속여부[1]	패킷 의 종류[2]	전체 크기[3]	보내는 크기[4]
송신자 이름[5]			
전송 내용[6]			

1. 접속여부 : 1(접속중) , 2(접속중지) [1Byte]
2. 패킷의 종류 : BMP, FDP, SPEAK, SELECT [4Byte]
3. 전체크기 : [4Byte] , 4.보내는크기 : [4Byte]
5. 송신 이름 : [50Byte] , 6.전송내용 : [6937Byte]

그림 8. 전송 패킷의 구조

정한 아바타의 번호를 송신하여 선택한 아바타가 상대편 화면에 보여지게 구성하였다. 그림 9에 전송 패킷의 구조를 보였다.

위 과정에 의하여 회의상태를 초기화하면, 화자의 발언 내용을 아바타가 렌더링 할 수 있는 준비가 되고, 회의시 발언하고자 할 때 패킷의 종류를 SPEAK로 바꾸어 전송하면 상대편의 아바타는 패킷의 종류를 확인하고 전송 내용을 TTS에 넘겨주게 된다.

TTS는 아바타와 독립적인 프로그램으로 동작하도록 되어 있으며, 아바타로 부터 전송받은 텍스트를 이용하여 음성합성을 수행한다. TTS는 확장성 및 응용성을 고려하여 마이크로소프트사의 SAPI (Speech API) 규격에 맞도록 구현하였으며 아바타와 파일방식을 사용하여 정보를 주고 받게된다. 이때, TTS는 발언하고자 하는 상대편의 얼굴을 합성음과 동기를 맞추어 움직이도록 하기 위해서 음소 단위로 정해진 시간마다 얼굴정보를 아바타에게 보내주고, 아바타는 얼굴정보를 폴링하면서 바뀌는 얼굴정보에 따라서 얼굴모양을 움직이게 된다. 이렇게 함으로써 실제로 아바타가 말을 하는 것과 같은 효과를 얻을 수 있다. 그림 9는 4명의 참가자가 접속했을시의 모습을 만화캐릭터를 이용한 아바타를 사용하여 구현하여 보았다.

그림 9. 실행된 서버와 클라이언트

Ⅳ. 결 론

본 논문에서는 TTS연동 아바타의 구조와 원리를 알아보았고 이를 기반으로 한 TTS와 MPEG-4 SNHC의 표준안을 따르는 아바타를 이용하여 다자간의 화상회의시스템을 구현하여 보았다. 구현한 시스템을 데스크탑 PC에서 실험한 결과, 아바타와 TTS의 연동 측면에서 만족할 만한 성능을 얻을 수 있었다. 그러나 현재의 시스템은 아직도 랜더링 기술, 합성음성의 명료성 및 자연성 개선 문제로 실용화 단계까지는 거리가 있다고 판단된다. 하지만 이러한 기술들은 향후 story-teller on demand, 가상현실분야에서 음성을 이용한 아바타의 의사표현도구등 그 응용분야가 무척 다양하다. 따라서 더욱 자연스러운 TTS와 아바타를 이용한 멀티미디어 기술확보를 위해서, 복잡한 메쉬구조를 가진 자연스러운 아바타생성기술 및 랜더링기술, 명료한 TTS 음성재생기술들을 개발하고 지적재산권을 확보한다면 향후 좋은 결과를 기대할 수 있을 것이다.

※ 참 고 문 헌

[1] H. Klatt, "Review of Text-To-Speech Conversion for English", *Jounal of the Acoustic Society of America (JASA)*, Vol. 82, pp. 737-793.

[2] Eds, G. Bailly, *Talking Machines, Theories, Models and Designs*, Elsevier : North Holland, 1992.

[3] 이정철, 운율과 합성단위 최적화를 이용한 한국어 합성음 명료도와 자연성 개선 연구, 서울대 박사학위논문, 1998.8.

[4] ISO/IEC JTC1/SC29/WG11 N1902," INFORMATION TECHNOLOGY CODING AUDIO-VISUAL OBJECTS : VISUAL 14496-2 Committe Draft," Oct. 1997.

[5] Richard S. Wright, Jr., and Micheal Sweet, *OpenGL SuperBible*, Waite Group Press, 1996.

[6] 이주상, 유지상, "MPEG-4 SNHC기반 얼굴 객체의 구현," *Telecommunication Review*, 제8권 3호, pp.400~409, 1998.

[7] Keith Waters and Thomas M. Levergood, "DECface : An Automatic Lip-Synchronization Algorithm for Synthetic Faces," *Digital Equipment Corporation Cambridge Research Lab Technical Report* 93/4, 1993.

[8] Craig Hunt, *TCP/IP Network Administration*, O'Reilly & Associates Inc., 1994.

안 병 호

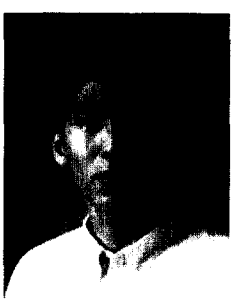
1998년 한림대학교 전자공학과(공학사)
1998년~현재 광운대학교 전자공학과 석사과정
※주관심분야 : 영상처리, 영상인식, 가상현실 등

유 지 상

1985년 서울대학교 전자공학과(공학사)
1987년 서울대학교 전자공학과(석사)
1993년 미국 퍼듀대학교 전기공학과(Ph. D.)
1994년 현대전자산업(주) 선임연구원
1997년 한림대학교 전자공학과 조교수
1997년~현재 광운대학교 전자공학과 조교수
※주관심분야 : 영상신호처리, 비선형신호처리, 컴퓨터 비젼 등

김 상 훈

1990년 연세대학교 전기공학과(공학사)
1992년 KAIST 전기 및 전자공학과(석사)
1992년~현재 한국전자통신연구원 음성언어팀 선임연구원
※주관심분야 : 음성합성, 음성인식 등



이 승 훈

1989년 고려대학교 전자공학과(공학사)
1991년 고려대학교 전자공학과(석사)
1991년~현재 한국전자통신연구원 멀티모달 I/F팀 선임연구원
※주관심분야 : 음성합성, 음성인식, 음성응용 시스템 등