

# 디지털 방송을 이용한 실시간 다채널 이미지갤러리

□ 박연선, 정문열 / 서강대학교 영상대학원 미디어공학과

# Ubiquitous

## 요약

본 논문은 디지털 TV에서 시청자가 보고 있는 동영상에서 관심 있는 화면을 캡춰하여 저장하고, 그 저장된 이미지들을 미술관이나 박물관 같이 꾸며진 환경에서 감상하듯이 다시 볼 수 있는 자신만의 이미지 갤러리를 만드는 것을 목적으로 한다. 또한, 컴퓨터 안에 있는 이미지를 PDA나 휴대폰 등에 쉽게 업로드할 수 있는 것과 같이 디지털 TV에서도 이 이미지들을 가공하거나 이메일을 통해 다른 사람에게 전달할 수 있다.

본 논문에서는 디지털 TV에 이러한 갤러리 기능을 추가하기 위한 데이터 방송 애플리케이션의 설계와 구현을 기술한다. 이 애플리케이션은 디지털 방송이 시작되면서 가능해진 본격적인 데이터 방송을 응용한 독특한 예이다. 본 논문에서는 이러한 데이터 방송을 구현하는데 필요한 요소를 기술하고, 애플리케이션을 구현하는 과정에서 해결해야 했던 문제점들을 기술한다.

## 1. 서론

디지털 방송(Digital Broadcasting)이란, 프로그램의 제작, 송출, 수신에 이르기까지 전과정의 디지털화를 말하는 것으로 기존의 아날로그 전송 방식보다 비디오, 오디오를 약 4배정도 압축할 수 있다는 장점을 통해 같은 대역폭을 이용해서 고화질, 고음질, 다채널 방송을 가능하게 하는 방송을 말한다[1]. 이로서 기존에 대역폭의 문제로 아주 적은 양의 데이터만을 부가정보로 보낼 수 있었던 데이터 방송은 여러가지 다양한 구성을 갖춘 서비스들을 제공하는 것이 가능해졌다. 기존의 방송에서 시청자들이 수동적인 자세로 TV를 감상할 수밖에 없었던 것에 반해 디지털 방송에서는 시청자와 상호작용할 수 있는 상호작용적 콘텐츠를 방송하고 리턴 채널(return channel)을 통한 시청자의 의견과 반응을 실시간적으로 고려할 수 있다. 데이

터 방송이 시작되면서 기존의 오디오, 비디오의 동영상 서비스 외에 상호작용적 (interactive) 데이터 방송 콘텐츠에 많은 사람들이 관심을 가지고 있다.

본 논문은 이렇게 시청자와의 실시간 상호작용 (interactivity)을 가능하게 하고 시청자로 하여금 참여의지를 높이도록 해주는 데이터 방송 애플리케이션의 한 예를 구현하고, 이 구현과정에 필요했던 요소와 문제점들을 기술한다. 본 논문에서 구현한 데이터 방송 애플리케이션은 현재 방송되고 있는 동영상에 자신이 원하는 장면이 나왔을 때 그 화면을 실시간으로 캡처하고 저장할 수 있도록 해주는 기능을 가진 것이다. 이 애플리케이션을 통해 시청자는 캡처한 이미지를 셋탑박스 (Set-top Box, STB) 하드디스크에 저장하거나 리턴 채널을 통해 서버에 저장하고 필요할 때 다시 불러오기를 통해 TV에서 미술관이나 박물관 처럼 이미지들을 감상할 수 있다. 즉, 본 애플리케이션은 TV 안의 개인 이미지 갤러리를 구현한 것이다. 현재 본 논문에서 구현된 부분은 아니지만 리턴 패스를 이용하여 접속한 서버를 통하여 저장된 이미지들을 이미지 후처리 등의 작업을 통해 원하는 형태로 가공하거나 이메일을 통해 전달하는 것도 가능하다. 또한, 추후 셋탑박스에 하드디스크가 내장된다면 개인의 셋탑박스에 직접 이미지들을 저장할 수도 있을 것이다.

본 애플리케이션은 DVB-MHP 표준을 바탕으로 구현하였으며 그 표준에는 Java의 JMF 1.0만을 지원하는 바, 아직 화면을 캡처하는 API는 추가되지 않은 상태이다. 이 문제를 해결하는 한 방법은 TV 동영상에서 사용자가 화면을 캡처한 시점을 리턴 채널을 통해 서버에 보내고 서버에서 동일 동영상의 해당 화면을 찾아 캡처하는 것이

다. 본 논문에서는 DVB-MHP 표준에서 지원하는 Java의 기능이나 하드디스크 등 셋톱의 사양이 본 데이터 방송 애플리케이션을 제작하기에는 아직 부족하므로, 이를 퍼스널 컴퓨터(PC)상에서 구현하였다. 그리고 캡처된 셋톱박스상에서 이미지를 네비게이션 할 수 있는 기능을 갖춘 애플리케이션을 제작해 봄으로써 추후 디지털 TV에서 동일한 애플리케이션을 개발할 때 필요한 개발 방향을 제시하였다.

본 논문은 데이터 방송 개발 환경, 이러한 개발 환경 위에서 구축된 이미지 갤러리 어플리케이션 구현과 결과, 추후 연구 과제로 구성되어 있다.

## 2. 데이터 방송의 개발 환경

현 시점에서 데이터 방송을 기획, 제작하는 일은 데이터 방송이 어떻게 이루어지는가와 전송되는 트랜스포트 스트림(Transport Stream, TS)에 대한 이해가 없이는 어려움이 있다. 현재 DVB-MHP 표준에 맞게 데이터 방송을 제작 가능하게 해주는 틀이 개발되어 있기는 하나, 단순한 기능과 상호작용을 가진 데이터 방송외에는 제작할 수 없는 상황이다. 따라서 엔지니어들이 직접 Java 프로그램을 작성할 수 밖에 없는 실정이다.

본 논문의 애플리케이션도 순수 핸드 코딩을 통해 구현을 하였고 이는 여러가지 방송 규약 및 프로그래밍에 대한 기반 지식이 있어야 가능한 일이었다. 이에 DVB-MHP 표준의 데이터 방송을 개발하는데 필요한 기본 사항들, 본 논문에서 기획, 개발, 제작한 애플리케이션을 구현하는데 필요한 사항들에 대해 먼저 살펴보고자 한다.

### 1) DVB-MHP

DVB-MHP는 Digital Video Broadcasting-Multimedia Home Platform의 약어로서 유럽식 디지털 방송(DVB)의 데이터 방송 규약을 말한다. MHP에서 제안하는 API를 이용하면 가정용 단말기인 셋탑박스(STB), TV, 퍼스널 컴퓨터(PC)와 그 주변장치, 그리고 가정용 디지털 네트워크를 모두 수용하는 수신기에서 부가방송(enhanced broadcasting), 양방향 서비스(interactive service), 그리고 인터넷 접근(internet access) 등의 서비스를 구현할 수 있다(2,3).

### 2) Xlet Program

DVB-J 어플리케이션은 DVB-MHP API를 이용하여 자바(JAVA) 언어를 가지고 제작하는 어플리케이션으로 JAVA TV API에서 정의한 Xlet interface를 구현하여 만드는데, Xlet이라고도 한다(4).

Xlet은 메인(Main)함수를 가지는 어플리케이션(application)이라기 보다는 사실 인터넷의 애플릿

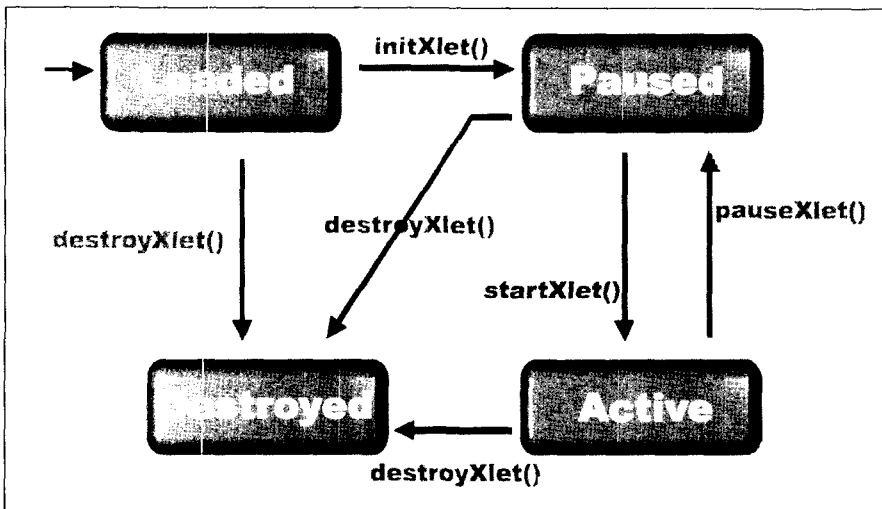
(applet)에 가깝다고 할 수 있다. Xlet은 어플리케이션 매니저(application manager)를 통해 그 Lifecycle이 제어되는데, Xlet이 각각의 상태(State)에 따라 실행되는 방식이 마치 애플릿이 실행되는 방식과 유사하다. Xlet의 lifecycle은 <그림 1>과 같다.

### 3) STB(Set-top Box)

셋탑박스는 위성을 통해 수신된 Mpeg-2 방송신호를 전달받아 디코딩(decoding)하여 TV 화면에서 보여주는 디지털 수신기로서 트랜스포트 스트림을 수신하여 처리하는 과정은 <그림 2>와 같다.

### 4) JMF(Java Media Framework)

JMF는 기존의 자바 프로그래밍 중에서 멀티미디어 제어관련 API 및 SDK를 보강한 것으로 Time-based Media를 모두 다룰 수 있으며 비디오, 오디오의 캡처, 저장, 전송, 스트리밍 등을 처리할 수 있다. Time-based media의 가장 기본적인 형태



<그림 1> Xlet Life Cycle(4)

는 audio, video가 있다[5].

MHP 셋탑박스는 트랜스포트 스트림 형태로 들어오는 동영상을 제어하는 하드웨어 player를 가지고 있다. 그리고 Xlet 프로그램에서 특정 방송 트랜스포트 스트림을 가리키는 Locator에 대한 JMP player를 생성하면 셋탑박스의 하드웨어 player를 사용하는 JMF player가 생성된다. 그렇기 때문에 본 논문에서 하고자 하는 동영상 이미지 캡처는 JMF player와 관련이 깊다. JMF player를 설정하고 제어하는 방법에 관해서는 제 3장에서 자세하다.

### 5) HAVi & DVB UI(User Interface)

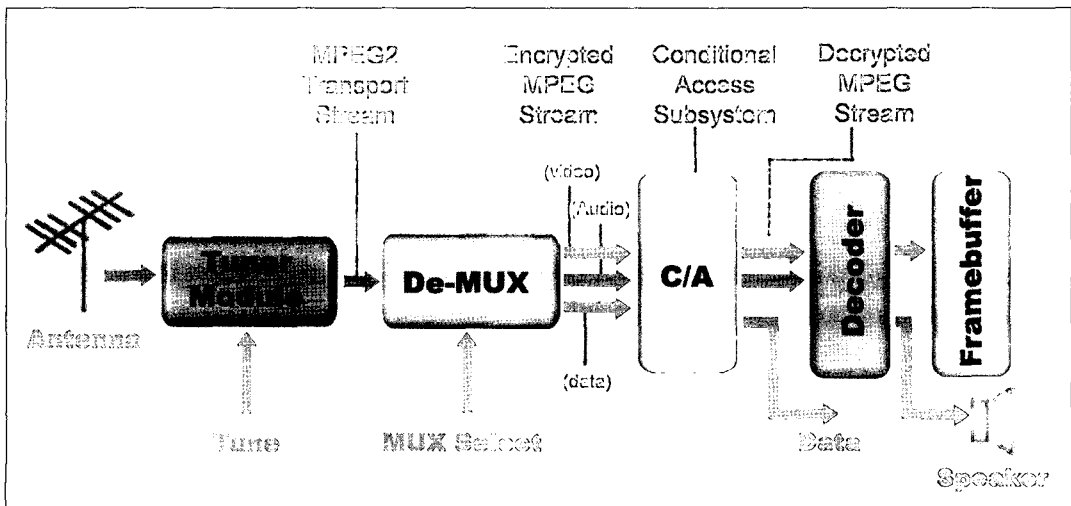
HAVi UI는 Home Audio Video Interoperability Architecture User Interface의 약어로서 기존의 모든 전자제품(Consumer Electronic(CE)) UI를 구성할 때 이용되는 Interface 이다. 이름의 약자가 의미하듯이 가전제품들의 네트워킹 및 호환을 지원하는 것이 목적인데, Java AWT의 확장된 개념이라

고 보면 된다. 디지털 방송용 셋탑박스에서 이 spec을 사용하여 UI를 구성해야 하는 이유는 HAVi UI가 TV와 친숙한 사용자 인터페이스를 지원하기 때문이다. 특히, 기존의 TV기반 시스템에서 그래픽스/비디오 관련 기능들과 오버레이들을 가능하게 해준다[6][7].

Alpha값을 갖는 그래픽 컴포넌트를 동영상 백라운드와 오버랩시키는 문제를 해결하려면 HAVi UI뿐만 아니라 DVB UI를 이용해야 한다. Alpha값을 갖는 컴포넌트와 동영상을 반투명하게 만드는 작업은 DVB UI중에 DVBColor Class를 이용하여 만들 수 있다[7].

### 3. 실시간 대화형 이미지 갤러리

본 논문에서 이미지 갤러리라 함은 현재 TV상에서 시청하고 있는 동영상 중에서 자신이 소유하고자 하는 장면을 캡처하여 이미지로 저장하고 이렇게 저장된 여러 개의 이미지들을 마치



< 그림 2 > 셋탑박스(Set-top Box)를 통한 방송수신 흐름도(4)

미술관 혹은 박물관에 놓인 그림이나 사진들을 감상하듯이 다시 보기를 통하여 화면상에 전시되는 것들을 감상할 수 있도록 만들어진 공간을 말한다. 여기서 전시되는 공간은 TV 상의 화면이 되고 미리 저장해 두었던 이미지들을 한곳에 불러모아 감상할 수 있도록 구성되어진다. 이때 리모트 컨트롤을 사용한 이미지들의 네비게이션을 구현하고 또한 선택한 이미지에 대하여 여러가지 조작을 가능하게 해준다. 여기서 이미지의 조작이라 함은 따로 이미지를 합성하거나 혹은 필터링을 통한 이미지 효과를 첨가하여 이미지에 대한 개인의 취향에 따른 조작을 통해 결과 이미지를 따로 저장할 수 있도록 해주어 단순한 사진 이미지 뿐만 아니라 그림과 같은 효과를 가진 이미지들도 만들 수 있도록 해주는 것을 말한다. 이렇게 나온 결과 이미지들을 자신이 보내고 싶은 사람에게 이메일이나 휴대폰을 통한 전송을 할 수 있도록 함으로써 디지털 미디어라는 통합을 디지털 TV에서도 이룰 수 있다. 현재 본 논문에서 테스트한 환경, 셋탑박스 상황에 따라 이메일이나 휴대폰으로의 전송은 향후 미래에 가능해지겠지만 우선은 PC 상에서 가상적으로 애플리케이션을 구현하여 실험해보았다. 또한, 셋탑박스에서 실행되는 Xlet 프로그램을 구현하고 갤러리 공간이 TV 상에서 어떻게 보여지는가를 PC에서의 모의 실험을 통해 논문에서 제안한 아이디어의 가능성을 타진해보았다.

실제 셋탑박스 안에서 직접 트랜스포트 스트림으로부터 보여지는 화면의 한 정지 영상을 캡처하여 이미지 처리하는 API들은 현재 나와 있는 셋탑박스 안에 탑재되어 있는 Java VM에서 지원하고 있지 않다. 따라서 PC 상에서 구현한 것과 동일한 테스트는 할 수 없었다. 이 문제에 대한 분석과 해결방

안에 대하여 뒤에 따로 설명을 하겠다.

### 1) 이미지 갤러리 개발 배경

이미지 갤러리를 기획하는 단계에서는 앞으로 서비스가 실시될 예정인 데이터 방송에 대한 사전 조사를 하고 그에 따른 하나의 아이디어에 착안하여 생각을 정리해나갔다. 데이터 방송이 시작되면 가능해질 여러가지 서비스들의 종류와 내용을 살펴보면 크게 4가지 정도의 분류가 가능해진다[1].

#### 1) 독립형 단방향 서비스

#### 2) 프로그램 연동형 단방향 서비스

#### 3) 프로그램 연동형 쌍방향 서비스

#### 4) 프로그램, 인터넷, 휴대폰 연동형 쌍방향 서비스

이 가운데 본 논문에서 구현하고자 하는 애플리케이션은 프로그램, 인터넷, 휴대폰 연동형 쌍방향 서비스에 속하는 것으로서 TV 시청 중에 원하는 이미지를 저장한 후 이를 인터넷을 통한 이메일로의 전송이나 휴대폰으로의 전송을 통해 PC 서버와 휴대폰간의 쌍방향 서비스들을 가능하게 한다.

### 2) 이미지 갤러리 구성

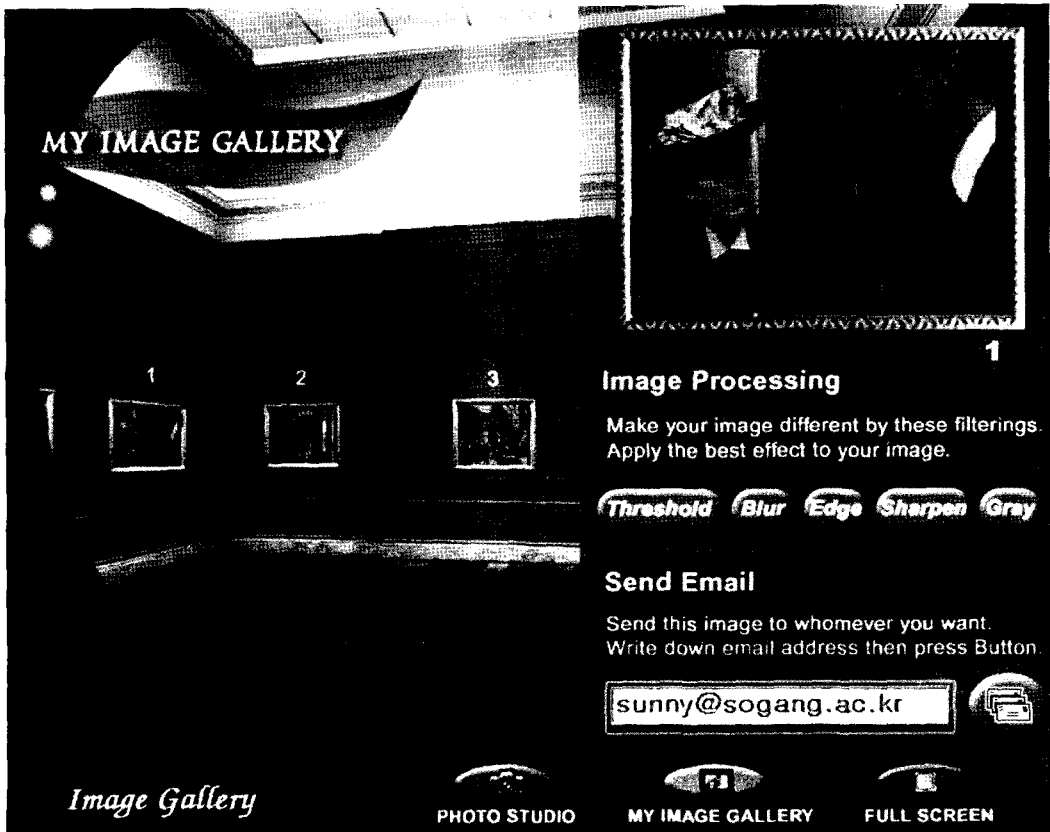
동영상에서 정지이미지를 캡처하면 작은 액자에 그 순간 캡처된 이미지가 보여지게 되고 저장하고 싶은 이미지일 경우 번호를 지정하여 각각 9장의 이미지를 선별하여 저장할 수 있다. 이렇게 저장된 9장의 이미지들은 각기 갤러리라는 공간안에서 다시 보기를 할 수 있는데 이러한 공간을 세계의 다른 색으로 구성된 방으로 만들고 각 방에는 3장의 이미지를 전시하게 된다. 또한 그 안에서 이미지의 후처리를 하고 이메일 전송을 할 수 있도록 인터페

이스를 구성하였다. 이러한 이미지 갤러리 공간을 구성하기 위해서 3D view로 보여지는 갤러리 공간을 만들고 이를 이미지로 저장하여 보는 이에게 공간감을 줄 수 있도록 배경을 구성하였다. 각 갤러리 공간 안에 전시되는 그림들로 나타내지는 액자들에 캡처한 이미지들이 올려지고 그 이미지들 중에서 자신이 보고 싶은 이미지를 선택하면 큰 액자에 다시 불러오게 되어 큰 이미지로 보여지게 된다. 보는 이로 하여금 넓은 갤러리 공간 안에서 자신의 이미지들을 감상하듯 볼 수도 있고 큰 그림으로 불러들여 볼 수 있어 작은 이미지로만 보게 되는 답답함을 해소하도록 노력하였다. 이러한 공간 구성으로 이

루어지는 갤러리 중에 한 공간 전시 이미지를 보면 다음과 같다.

### 3) 이미지 갤러리 구현 방법

위에서 구성한 기획에 맞추어 이미지 갤러리를 구현하는 방법에 대해 일련의 절차를 설명해보고자 한다. 비디오 영상을 Xlet 상에서 제어할 수 있도록 하기 위한 JMF player를 만들고 그 player에게 동영상의 실행(play)되는 시간을 알려주고 컨트롤할 수 있는 Control Component와 TV 화면 상에 보여질 공간을 지정해주는 VisualComponent를 할당해주고, 화면 사이즈를



〈그림 3〉 Application으로 구현한 Image Gallery

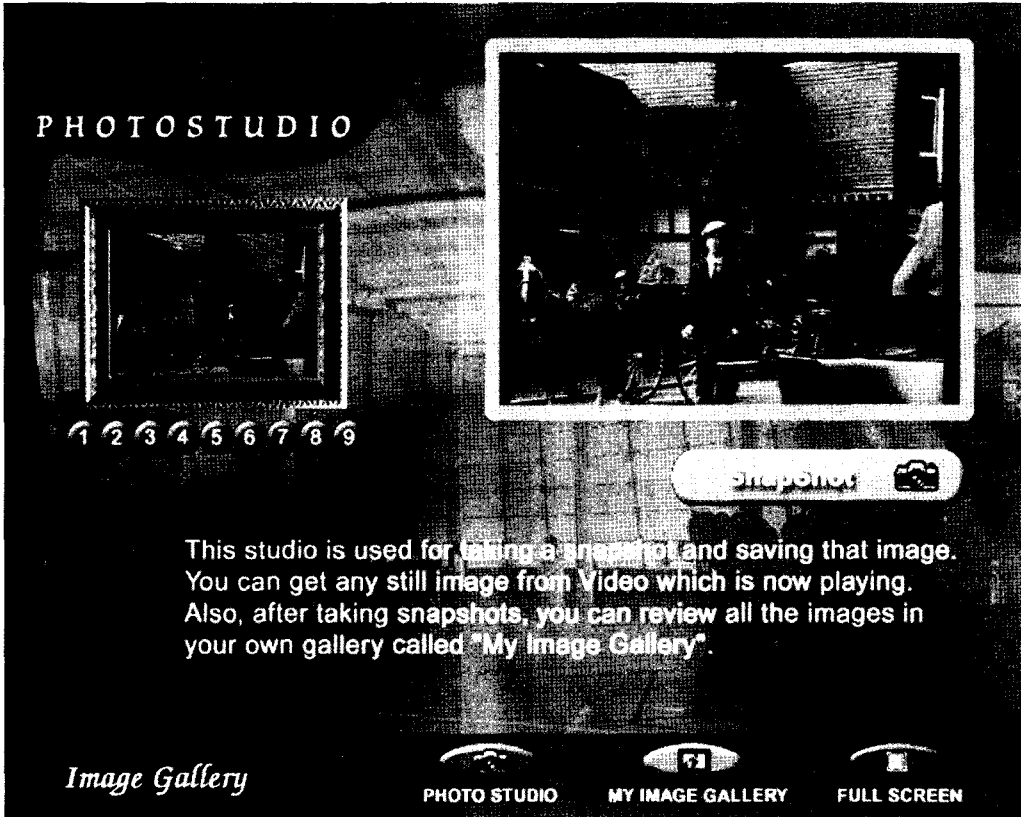
조절하는 Control, 한 프레임(frame)의 정보를 가져와서 버퍼에 저장을 하도록 제어하는 Control 등을 구현한다. 이 기능은 현재 테스트한 환경인 셋탑박스에서는 불가능한 기능으로서 이를 위한 API들을 셋탑박스 안에 Java VM이 지원하고 있지 않기 때문이다. 이것에 대한 언급은 뒤에 자세히 다루기로 한다. 이렇게 하여 한 장면을 캡처하게 되고 그것을 하나의 이미지로 만들어 저장하게 된다. 이렇게 저장된 이미지는 따로 하드에 보관되어지고 서비스 내에 개인 이미지 갤러리 보기는 버튼을 통하여 다시 보기할 수 있게 된다. 이때, 보여지는 환경은 동영상은 가려지고 전체 화면이 독립형으로 돌아가는 데이터 서비스가 되는데 전체적인 분위기를 마치 미술관이나 박물관과 같은 형태로 꾸미게 되어, 보는 시청자로 하여금 그 동안 저장해 두었던 이미지들을 마치 작품을 감상하는 기분으로 보도록 해주는 UI로 구성되게 된다. 이렇게 저장된 이미지에 또 다른 가공을 통해서 나만의 개성적인 그림을 갖도록 할 수도 있는데 기존에 캡처된 이미지는 단지 사진과 같은 분위기를만 가질 수 있는 것에 반해 필터링 효과를 첨가시켜 사진을 그림처럼 바꾸어 줄 수도 있다. 이렇게 최종 과정을 거쳐 나온 이미지는 하드에 다시 저장되어 개인 이미지 갤러리에서 다시 보기를 통해 감상할 수도 있고 이렇게 저장된 이미지를 다른 사람에게 이메일이나 휴대폰을 통하여 전달할 수도 있는데 이것은 현재 테스트하고 있는 셋탑박스 상에서 아직 구현이 되지 않아 추후 과제로 남기게 되었다. 본 논문에서는 이를 가상적으로 구현하기 위하여 PC에서 구현하고 모의실험하였다.

### 동영상에서 화면 캡처하기

본 논문에서 테스트한 Philips 셋탑박스안에는

기본적으로 아무런 데이터 방송 서비스가 이루어지지 않을 때 디코더에서 나온 비디오를 바로 화면에 보여주는 Background JMF Player와 그래픽 레이어(layer) 상에 올라가는 동영상을 play해 주기 위해 제공되는 AWT Component JMF player가 있다(7). Xlet 안에서 현재 TV 화면에서 play되는 동영상을 제어할 수 있는 player를 만들 수 있는데 이를 위해 서비스 컨텍스트로부터 기본적으로 돌아가고 있는 Background JMF player를 얻어온다. 이렇게 Xlet상에서 구현한 player는 동영상을 제어하는 여러가지 기능들을 가지고 있는데 기본적으로 제공되는 컨트롤(Control)에는 두가지가 있다. 첫 번째 컨트롤은 화면의 사이즈를 줄여 그래픽 레이어와 동시에 보여지게 할 때 동영상의 사이즈를 제어할 수 있는 AWTVideoSizeControl이고 두 번째 컨트롤은 동영상을 선택하는 MediaSelectControl이다. 본 논문에서는 캡처하는 기능을 가진 player를 만들기 위해 화면 캡처를 위한 컨트롤로서 FrameGrabbingControl을 추가했다. 이 컨트롤은 현재 테스트한 셋탑박스 안에서는 해독할 수 없으므로 PC상에서 구현하여 실험하였다. 현재 play되는 동영상의 한 프레임에 대하여 그 동영상의 포맷(Format)과 인코딩(Encoding) 방식에 대한 정보를 알 수 있으므로, 현재 프레임의 정보를 버퍼에 저장한다. 여기서 버퍼에 들어 있는 정보는 다시 Image객체의 instance로 전환되어 화면에 보여질 수 있는 가능한 형태로 바뀌게 된다.

PC상에서 구현한 Image Gallery에서 Snapshot이라는 버튼을 누르면 현재 play되는 동영상에서 버튼을 누른 순간의 프레임 정보를 가져와 그 한 장면을 캡처하게 된다. 캡처한 순간의 이미지는 바로 옆에 액자 안에 다시 불러와 보여주게



〈 그림 4 〉 Application으로 구현한 Photo Studio

되고 자신이 원하는 화면일 경우 지정된 번호에 저장할 수 있다.

#### 4) 이미지 갤러리 구현상의 문제점과 해결방안

현재 본 논문에서 구현하려고 하였던 모든 기능들은 데모용 PC 애플리케이션에선 완벽히 구현이 되었지만 셋탑박스에서 실행되는 Xlet 프로그램으로는 완벽히 구현이 되지 못하였다. 그 이유중의 하나는 PC가 지원하는 JMF player와 셋탑박스가 지원하는 JMF player의 기능차이이다. 트랜스포트 스트림을 셋탑박스에서 받아들여 TV 상에 보여주는 player가 PC 상에서 구현되는 JMF player와 개념

적으로 흡사해서 셋탑박스의 player를 JMF player로 간주하기는 하지만, 실제로는 그 기능이 차이가 많다. 즉 셋탑박스의 player에는 동영상의 한 화면을 캡처하는 기능이 없다. 이를 해결하는 한 방법은 PC와 셋탑박스 간의 서버 클라이언트의 네트워크 통신을 이용하는 것이다. 셋탑박스의 Xlet이 동영상에서 영상을 캡처하기 위해 시청자가 버튼을 클릭한 순간의 시간정보인 NPT(Normal Play Time)를 얻어 온 다음, 이를 서버로 보내면 서버의 player는 그 NPT를 MediaTime으로 하고 이 시각에 해당되는 이미지를 캡처하여 클라이언트인 셋탑박스에게 보내주는 방법이다[8]. 이러한 시스템은 셋탑박스에



서 이미지를 캡처할 수 있다고 하더라도 그 이미지 자체를 셋탑박스 안에서 처리하는데 무리가 있을 경우 또한 셋탑박스 안에 하드가 없을 경우 서버와의 통신을 통해 저장하고 이미지 처리를 수월하게 하도록 도와주는 방법으로도 이용될 것이라 본다.

구현상의 문제 이외에 거론될 수 있는 문제가 있다면 캡춰한 이미지의 저작권 문제인데, 이 부분은 방송을 제공하는 콘텐츠 제공업체 혹은 콘텐츠 소유권자와 합의하여 셋탑박스 안에 저장되어지는 이미지의 종류나 수에 관련해 제약사항을 두거나 계약을 하는 방식으로 해결해 나갈 수 있을 것이라 생각된다. 셋탑박스는 거의 한 가정내에서 사용하는 개인적인 물건이기 때문에 집안 내에서 개인적인 목적으로 사용할 경우는 크게 문제시 되지 않겠지만 이미지들이 외부로 유출될 경우는 워터마크를 이용한 방법, 이미지에 대한 이용료 부가 등 제한을 두는 여러가지 방법들을 생각해볼 수 있겠다. 추후 이 문제에 관해서는 구현이 이루어진 이후 콘텐츠 제공업체와 효과적인 해결방안에 대한 구체적인 논의가 이루어져야 할 것으로 생각된다.

#### 4. 결론 및 추후 연구

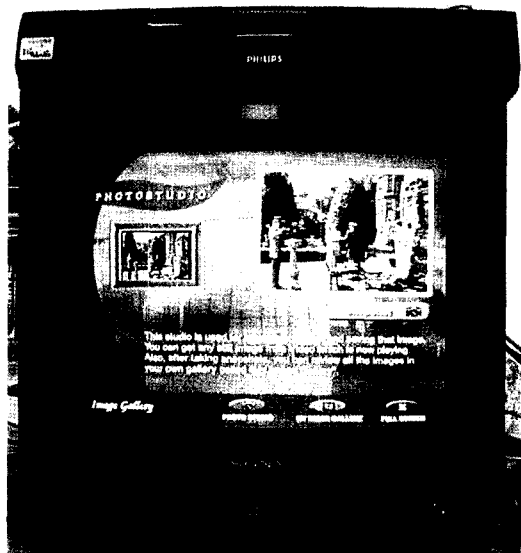
본 논문에서 기획한 데이터 방송, 즉 동영상에서 원하는 이미지를 캡처하고 이를 이메일로 전송하는 프로그램을 기획하고, PC의 애플리케이션으로 구현 실험하였다. 그리고 캡춰된 이미지들로 구성된 갤러리를 셋탑박스에서 Xlet 프로그램을 이용하여 구현하고 실험 하였다.(그림 5)

PC 애플리케이션에서는 실제 보여지는 영상 내에서 원하는 장면이 나왔을 경우 클릭하는 순간의 프레임을 캡처하여 이미지로 저장하고 이 이미지를 액자에 넣어 갤러리공간에서 전시한다. 이러한 갤

러리 안에서는 이미지들을 필터링 처리하여 여러가지 효과를 적용해보고 시청자가 원하는 이미지를 따로 저장할 수도 있고 또 이를 이메일로 자신에게 혹은 이미지를 보내주고 싶은 다른 사람에게 이메일로 전송할 수 있도록 구현하였다.

PC 애플리케이션에서 구현된 모든 기능들이 현재 테스트하고 있는 셋탑박스 환경 하에서는 구현이 되지 않기 때문에 셋탑박스 환경에서는 TV에서 갤러리를 보여주는 것에 초점을 맞추어 구현하였다. 이는 여러 이미지들과 동영상간의 관계를 테스트해보는 좋은 계기가 되었으며 실제 TV 상에서 보여질 때의 사용자 인터페이스에 대한 실험에도 많은 도움이 되었다.

PC 애플리케이션의 많은 기능들을 탑재한 셋탑박스를 만들려면 API와 Java VM의 환경, 즉 미들웨어를 관장하는 프로그램 코딩을 하거나 문제의 해결방안으로 제시한 서버 클라이언트의 통신을 통해 원하는 기능을 구현해 낼 수 있을 것이라 생각된다.



〈그림 5〉 Xlet을 셋탑박스에서 실행한 모습

● 참고문헌 ●

- (1) Data Broadcasting, The Technology and the Business, Lars Tvede, Peter Pircher, Jens Bodenkamp, AIRCODE 정책개발실 옮김, 2001
- (2) Report on API Migration, DigiTAG (Digital Terrestrial Television Action Group), 2001
- (3) The Multimedia Home Platform - an overview, J.-P. Evain EBU Technical department, 1998
- (4) JavaTV™ API Technical Overview : The JavaTV API whitepaper, version 1.0, Release Candidate D, BartCalder, Sun Microsystem, 2000
- (5) Java™ Media Framework API Guide, Sun Microsystems, 1998-9
- (6) The Havi Specification, www.havi.org
- (7) Multimedia Home Platform 1.0.1, DVB Bluebook A057 Rev.1, ETSI, 2000
- (8) '디지털 방송에서 T-Commerce를 위한 Video와 동기화된 Data Steam 생성과 해석기 구현' 이규혁, 2002

필자소개



박연선

- 1997년 2월 : 서울대학교 미술대학 공예학과 졸업
- 2002년 2월 : 서강대학교 영상대학원 미디어공학과 졸업(공학석사)
- 2002년 2월 : (주)알티캐스트 인사 애플리케이션 개발팀 선임연구원
- 주관심분야 : 디지털 위성방송, 방송용 애플리케이션 UI



정문렬

- 1980년 : 서울대학교 자연대학 계산통계학과 졸업(학사)
- 1982년 : 카이스트 전산학과 졸업(석사)
- 1982년~1986년 : 국방관리연구소 연구원 역임
- 1992년 미국 펜실베이니아 대학(Univ of Pennsylvania) 전산학과 졸업(박사)
- 1992년~1994년 : 일본 구주공과대학 정보공학부 조교수
- 1994년~1999년 : 숭실대학교 컴퓨터학부 부교수 역임
- 한국컴퓨터그래픽스학회 이사, 한국멀티미디어학회이사, 한국정보과학회 컴퓨터그래픽스 운영위원회 부회장 역임
- 2004년 현재 : 서강대학교 영상대학원 미디어공학과 부교수
- 주관심분야 : 게임, 애니메이션의 주요요소인 가상 캐릭터의 표정과 몸동작, 그리고 의상을 컴퓨터 기술로 재현하는 것과 대화형 TV(interactive TV) 콘텐츠 제작기술 및 제작이다.