논문 2007-44CI-1-11

MPEG-4 시스템 기반의 다시점 전환 시스템 구조 및 재생기 구현

(Multi-View Point switch System Structure & Implementation of Video player in MPEG-4 based)

이 준 철*, 이 정 원**, 장 용 석**, 김 승 호**

(Jun-Cheol Lee, Jung-Won Lee, Yong-Seok Chang, and Sung-Ho Kim)

요 약

본 논문은 현재 MPEG-4의 3차원 오디오/비디오(3-Dimensional Audio Video, 3DAV) 기술표준에서 다시점 비디오(Multi-view video)서비스를 제공할 수 있는 객체기술자(Object Descriptor)와 기초스트림기술자(Elementary Stream Descriptor)의 구조를 제안한다. 기존의 MPEG-4 시스템 상에서 확장영역을 사용하여 다시점 동영상 서비스를 제공 할 수 있는 객체기술자와 기초스트림의 구조를 각각 정의 하여 분류한 후 각 경우에 대해 분석한다. 기존 시스템의 확장만으로 송수신측과 연계되어 상관관계가 고려된 다시점 비디오 서비스 제공하는 것이 부적합하다는 것을 보인다. 그리고 다시점 영상 전송 시 수신측에서 각 시점간의 상관관계를 고려하여 시점 스위칭을 할 수 있는 새로운 객체 기술자를 추가한 구조를 제안한다. 이를 통하여 다시점 비디오 서비스에서 사용자 요구에 따른 시점 전환을 가능하게 하면서, 필요한 시점에 대한 정보만을 전송해서 수신측에 부하를 줄일 수 있다.

Abstract

This paper suggests structures of the Object Descriptor and the Elementary Stream Descriptor that provide multi-view video services in 3-Dimensional Audio Video technical standards of current MPEG-4. First, it defines that the structures of the Object Descriptor and the Elementary Stream Descriptor on established MPEG-4 system, then distributes individually, and analyzes that. But extension of established system is inappropriate for providing multi-view audio video services connected transmissions and receptions. And, this paper suggests a structure of new Object Descriptor able to switch viewpoints that considers the correlation between each viewpoints, when multi-view video is transmitted. By means of that, it is able to switch viewpoints according to a requirement of a user in a multi-view video services, and reduce overheads for transmitting information about necessary viewpoint.

Keywords: 3DA/V, Multiview Video, Multiview Point Switching

I. 서 론

현재 MPEG에서는 사용자의 시청 실감을 한층 더 높일 수 있는 방법으로 3차원 오디오/비주얼 기술에 대한 표준화가 진행 중이다. 카메라 기술, 광대역 전송 및 저장 기술, 디스플레이 기술, 그래픽 처리 기술 등이 크게 발전함에 따라 3차원 AV 분야도 실제 내용에 있어서는 다양한 형태의 영상 기술을 포함하고 있다. 안경식, 무안경식 등의 다양한 3차원 디스플레이 기술이 개발되고 있으며, 스테레오 영상 또는 이의 확장형인 평행형 다시점(parallel multi-view) 동영상 서비스가 등장하고 있다. 또한, 컴퓨터 처리속도와 저장장치의 용량이 급속히 증가됨에 따라, 컴퓨터 그래픽 기술을 활용한 3차원 그래픽 실감(virtual reality) 영상과, 그래픽 배경에 실

* 정회원, 부천대학 인터넷과
(Dept. of Internet Information Science, Bucheon college)
** 정회원, 경북대학교 컴퓨터공학과
(Dept. of Computer Engineering, Kyungpook National University)
접수일자: 2006년9월12일, 수정완료일: 2007년1월12일

사 영상 객체를 혼합하거나 그 반대로 실사 배경에 그래픽 객체가 혼합된 혼성 실감(mixed reality) 영상 기술이 발전하고 있다. 이와 더불어, 주변 360도 전방향의 파노라마 영상을 실시간으로 획득할 수 있는 카메라 기술이 개발되어 전방위(Omni-view) 동영상 획득이 가능해졌으며, 실제로 볼록 거울을 이용한 어안(fish-eye)형 카메라는 보안 감시 분야에 활용되고 있다. 광대역 전송 및 고용량 저장이 가능해 짐에 따라, 여러 대의 카메라를 사용하여 하나의 객체를 여러 각도에서 찍는 집중형 다시점 영상 기술이 가능해 졌다. 집중형 다시점 영상은 스포츠 중계와 공연 실황 녹화 등에 사용될 수 있다[1][2].

3차원 비디오는 두 가지 관점에서 접근할 수 있다. 첫 번째는 영상의 깊이 정보를 사용해서 영상의 일부가 화면으로부터 튀어나오는 느낌을 사용자가 느끼도록 한다. 두 번째는 사용자에게 다양한 시점을 제공하여 현실감을 느끼도록 구성된 다시점 비디오로써 한 대 이상의 카메라를 통해 촬영된 영상들을 기하학적으로 교정하고 공간적인 합성을 통해 여러 방향의 다시점을 사용자에게 제공하는 것을 말한다.

다시점 비디오는 고화질의 개념을 뛰어넘는 더욱 사실감이 넘치는 영상을 제공하기 때문에 사용자들은 미디어에 몰입감을 더욱 느끼게 되고 광고, 교육, 의료, 국방, 오락 등의 분야에서 영상정보 전달 효과를 크게 높일 수 있다. 이러한 특징으로 다시점 비디오는 사용자에게 전방향의 영상을 제공할 수 있는 옴니비디오, N개의 카메라로부터 입력된 영상을 선택할 수 있는 시점 스위칭, 기존 2차원 비디오와 비교해 사용자 주위의 장면에 대한 보다 넓은 시야를 제공하는 파노라마 등 다양한 형태를 나타내고 있다. 그러나 이러한 장점에도 불구하고 영상 획득시 필수적으로 존재하는 카메라간의 동기, 데이터 양, 고가의 장비가 요구되는 문제점으로 인하여 다양한 서비스 개발이 제한되고 있으며 영상 획득 방법이나 그 후의 영상 처리에 있어서 복잡한 시스템 구조를 가지고 있다.

본 논문에서는 3차원 A/V 기술 중 다시점 비디오에서 각 시점에서 입력되는 영상을 사용자와의 상호 작용으로 시점의 방향을 바꿀 수 있는 시점 전환 기술을 제안하고자 한다. 현재 다시점 동영상 코덱에 대한 표준은 제정되지 않았으나, 앞으로 이를 제어하기 위한 MPEG-4 시스템상의 새로운 구조가 필요하다. 이를 제어하기 위해 기존의 MPEG-4 시스템에서의 확장영역을 고려한 방법을 분석하고, 상관관계를 고려한 새로운 객체기술자가 추가 된 방법을 제안한다. 실험은 ENST에서 개발된 MPEG-4 플레이어 OSMO4를 사용한다[3].

Ⅱ장에서는 다시점 동영상 서비스를 위한 기본적인 MPEG-4 시스템의 구조와 고려 될 수 있는 구조에 대한 제안과 분석을 한다. 그리고 상관관계를 고려한 새로운 객체기술자를 추가한 구조를 보인다. Ⅲ 장에서는 실험결과, Ⅳ 장은 결론과 향후 연구 과제로 마무리 짓는다.

Ⅱ. MPEG-4 시스템 구조와 다시점 전환을 위한 확장영역을 사용한 구조 제안 및 분석

1. MPEG-4 시스템 구조

MPEG-4는 기존의 영상 압축 부호화 표준과는 달리 컴퓨터 합성 영상, 그래픽, 합성 오디오를 복합적으로 부호화하기 때문에, 기존의 화면 기반의 부호화는 달리 영상 및 오디오를 객체(object)기반으로 부호화한다. 이러한 객체 기반의 다양한 콘텐츠를 사용하기 위해 MPEG-4 시스템은 콘텐츠 전체의 구성을 나타내는 장면 기술(Scene Description)정보와 이 장면에 포함된 개개 콘텐츠 객체의 종류 및 비트스트림을 나타내는 객체 기술(Object Description)정보로 콘텐츠 전반적인 구성과 특성을 표현한다.

그림 1은 기본적인 MPEG-4의 SDM(System Decoder Model)이다. 이는 MPEG-4 시청각 장면을 재현하는 단말기의 개념적인 동작을 보여주고 있다. MPEG-4 시스템은 구조적으로 송신측에서는 시청각 장면이 압축된 후 동기정보가 부가되고 하나 또는 그 이상의 스트림으로 다중화 하는 전달계층으로 보내진다. 수신측에서는 이러한 스트림이 그림 1의 스트림 멀티플렉스 인터페이스(Stream Multiplex Intreface)에서

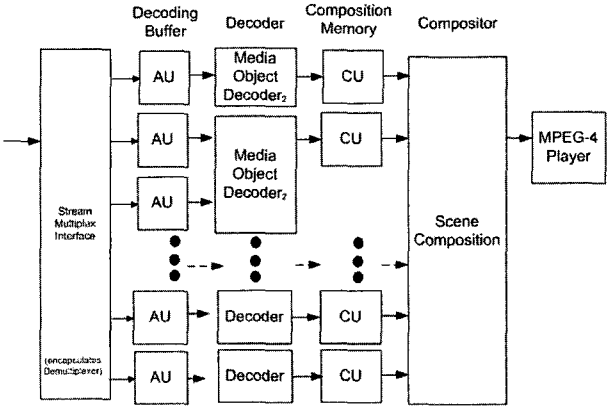


그림 1. SDM 구조
Fig. 1. System Decoder Model.

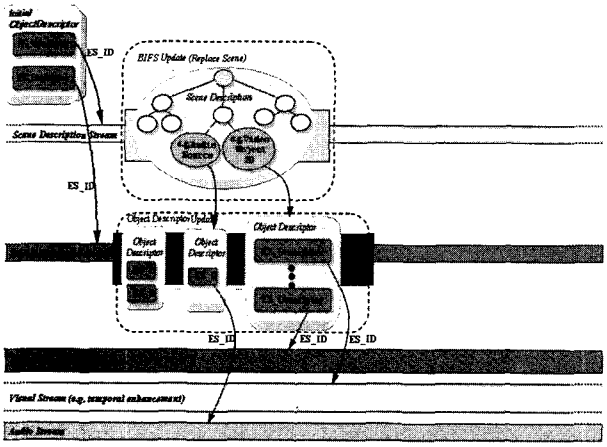


그림 2. 객체 기술자 프레임워크
Fig. 2. The object descriptor framework.

역 다중화 되고 디코딩 버퍼를 지나 각 스트림에 맞는 디코더를 통해 역 압축되어 여러 개의 미디어 객체를 만들어 낸다. 그리고 이 미디어 객체를 장면기술과 동기화(synchronization) 정보에 따라 장면 구성자(Scene Composition)를 통해 합성되어 최종 사용자에게 재현한다. 또한, 사용자가 마우스 등을 이용하여 객체를 움직이거나 다른 장면으로 이동할 수 있다.[4-7]

현재의 MPEG-4 시스템의 객체기술자는 2차원 동영상처리에 초점을 맞추어 정의됨으로써 현실감 및 입체감이 떨어지는 2차원 디스플레이어 방식만을 서비스 하고 있기 때문에 사용자에게 제한적인 서비스를 제공한다. 그림 1 의 SDM구조에서 볼 수 있듯이 디코더에서 하나의 이상의 스트림이 들어와도 하나의 미디어 오브젝트로 복호화 된다. 기존의 MPEG-4 시스템은 하나의 노드에 단일한 객체 기술자를 가리키고, 이를 통해 실제 미디어 오브젝트를 가리킨다. 따라서 일반적인 MPEG-4 기반으로는 다시점 동영상의 전송 및 디스플레이에 대한 한계성을 극복하지 못한다.

그림 2는 장면을 기술할 때 사용된 노드와 객체 기술자 그리고 미디어 기초스트림과의 관계를 쉽게 이해되도록 나타낸 객체 기술자 프레임워크(framework)이다. 최초에 초기화 객체 기술자(Initial Object Descriptor)를 통해 장면 기술자와 객체 기술자 스트림에 접근하고, 장면 기술자와 객체 기술자를 구성하여 실제 스트림까지 접근하는 과정을 나타낸다.

2. 다시점 동영상 서비스를 위한 객체 기술자/부호화 스트림 구조 분석

기술자는 크게 두 가지로 고려 될 수 있다. 기존 개념을 유지한 상태로 다시점 동영상 전송을 위한 시스템을 구축하는 것과 다시점 동영상 정보를 위해서 객체에 대한 정의를 새롭게 하는 것이다.

본 논문에서는 다시점 동영상 정보를 위한 시스템 구축에서 최소한의 변화를 주며, 시점간의 상간관계를 고려할 수 있는 새로운 객체 기술자를 추가하는 방법을 제안 한다. 각각의 시점에 대해서는 기존에 사용하던 객체기술자를 그대로 사용하고, 각 시점에 대한 객체 기술자를 제어하기 위해 슈퍼 객체 기술자 개념인 MVP_OD(Multiview Point Object Descriptor)를 정의하여 다시점 비디오의 시점전환을 제공한다. 먼저 각 다시점 동영상 서비스를 제공하기 위해 고려 될 수 있는 구조에 대해 분석 후 제안 된 객체기술자/기초스트림기술자의 구조와 그 문제점을 제시하고 이를 보완 할 수 있는 새로운 객체를 사용한 다시점 전환 기술의 실험 결과를 보인다.

다시점 동영상에서 전송되는 각 시점 별 영상 정보는 크게 두 가지 경우로 생각 할 수 있다. 모든 시점들의 영상 정보가 MVC(Multi View CODEC)을 통해 하나의 기초스트림에 포함되어 전송되는 경우와 각 시점 별로 MVC를 통해 각각의 기초스트림으로 전송되는 두 가지의 경우로 생각할 수 있다[7][9]. 그림 3은 각 시점의 영상정보를 MVC로 인코딩 하고 하나의 기초스트림을 통합해서 전송되는 경우 고려되는 SDM구조를 나타낸다.

각 시점에서 들어오는 영상정보를 MVC을 사용하여 중복성이 제거된 하나의 기초스트림으로 통합하고 다른 정보와 Mux되어 전송된다. 그리고 수신측에서 DeMux를 통해 각각 적합한 디코딩 버퍼(Decoding Buffer)에 들어가고, MVC_ES(MVC Elementary Stream)는 MVC을 통해 각 시점의 비디오 스트림의 형태로 컴포지터(compositor)로 들어가서 user interaction에 따른 시점에 전환에 의해 해당 시점의 미디어 객체가 렌더링 되어서 MPEG-4 플레이어로 재생된다. 이 때 하나의 시점을 보여주기 위해 모든 시점의 비디오 스트림을 복호

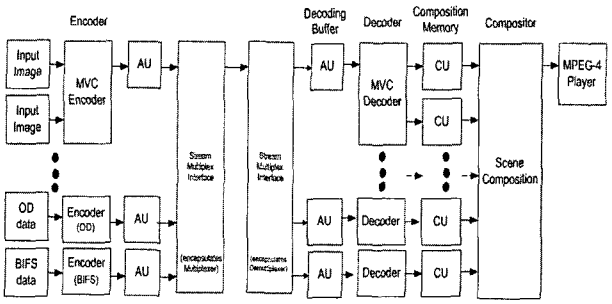


그림 3. 하나의 기초스트림인 경우의 SDM 구조
Fig. 3. SDM of case that is one ES.

시켜 시스템의 부하를 주지 않고, 사용자가 원하는 시점의 정보만을 복호하고, 선택된 시점간의 실감적인 시점 변화를 보여주기 위한 인터폴레이션(interpolation)을 제공하기 위한 시점간의 상관관계가 정의되어야 한다.

그림 4는 MVC에서 하나의 기초스트림으로 통합되어 전송 될 경우의 객체 기술 프레임 워크이다. 그림에서 알 수 있듯이 하나의 객체기술자는 하나의 MVC_ES을 가리킨다. MVC_ES는 MVC 디코더에 의해서 디코딩되고, 각 시점별 비디오 스트림이 생성된다. 이때 이를 제어할 수 있는 객체기술자와 기초스트림 기술자에 새로운 필드와 처리 과정이 필요하다.

그림 4에서 OD내의 ES_D(ES_Descriptor)에 있는 ES_ID를 통해 실제 스트림에 접근한다. 그러나 MVC 스트림은 여러개의 시점으로 이루어져 있다. MPEG-4 시스템 구조 상으로는 ES내에 있는 각 시점의 비디오 스트림을 접근 할 수 없다. 따라서 새로운 노드와 객체기술자를 추가하지 않고 다시점 동영상 서비스를 제공하기 위해 BIFS에서 영상을 표현하는 노드인 MovieTexture 노드와 OD/ES_D에 다음 세 가지를 고려하여야한다. 우선 MVC와 송신측에 user interaction에 의해 선택된 시점의 정보를 넘겨 줄 수 있고, 각 비디오 스트림간의 상관관계를 표현 할 수 있어야 한다. 그리고 카메라 파라미터 정보와 배치 타입 개수에 대한 정보를 고려해야 한다.

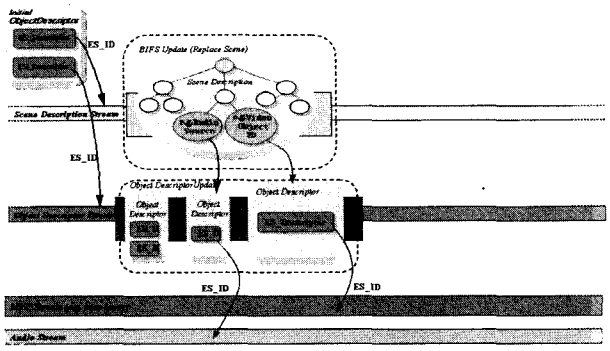


그림 4. 하나의 기초스트림인 경우의 객체 기술 프레임 워크

Fig. 4. The Object Descriptor Framework that has one ES.

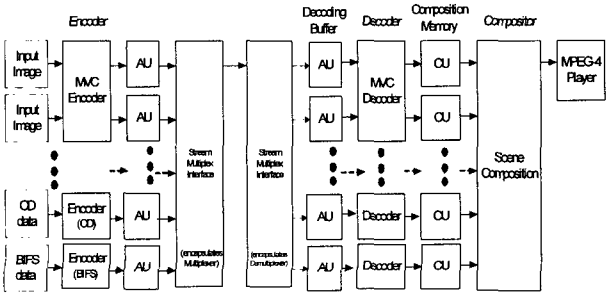


그림 5. 각각의 기초스트림인 경우의 SDM 구조

Fig. 5. SDM of case that is individual ES.

그림 5는 각 시점의 영상정보가 MVC로 인코딩되어 각각의 기초스트림으로 전송될 경우의 SDM 구조이다. 송신측에서 각 시점별로 들어오는 영상을 MVC 인코더를 통해 압축하고 다른 정보의 기초스트림과 함께 Mux를 통해 직렬화 되어 전송되고 수신측에서 DeMux를 통해 디코딩 버퍼(decoding buffer)로 들어가서 각각의 MVC 디코더에서 디코딩하고, 복호화한 후 user interaction을 의해 선택된 비디오 스트림을 컴포지션(composition), 렌더링(rendering) 해서 MPEG-4 플레이어로 재생한다. 이 때 모든 시점에 대해서 복호함으로써 시스템에 부하가 생기는 것을 줄이기 위해, 그리고 선택된 시점간의 실감적인 시점 변화를 보여주기 위한 인터폴레이션을 제공하기 위한 시점간의 상관관계가 정의 되어야 한다.

그림 6은 각각의 기초스트림인 경우의 객체 기술 프레임 워크를 나타낸다. 각 시점의 영상정보가 객체기술자를 통해서 각각의 기초스트림을 가리키는 것에서 알 수 있듯이 기존의 MPEG-4 시스템의 객체기술자 기반으로 이벤트를 처리하는 것이 아니라 기초스트림 기반으로 처리되는 것을 알 수 있다. 따라서 이를 제어 할 수 있는 객체기술자와 기초스트림기술자에 새로운 필드와 처리과정이 필요하다.

다음의 그림 7은 MPEG-4 시스템의 객체기술자와 기초스트림기술자 구조이다. 기존 MPEG-4 시스템의 확장을 통해 다시점 동영상 서비스를 제공하기 위해서는 extDescr을 사용해서 필드를 확장 시켜야 한다. 그리고 시점에 따른 user interaction 처리 과정에 따라 두

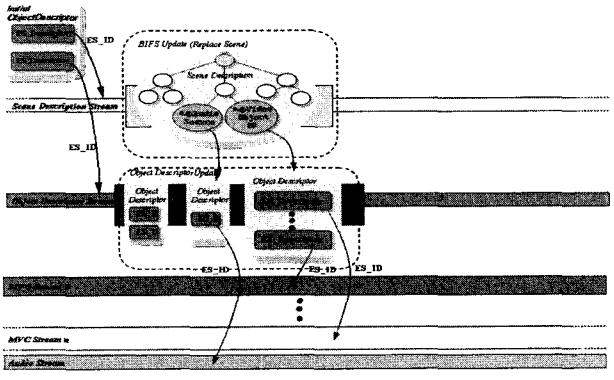


그림 6. 각각의 기초스트림인 경우의 객체 기술 프레임 워크

Fig. 6. The Object descriptor framework that has inividual ES.

가지 경우로 나누어진다.

첫 번째는 기초스트림의 단위에서 시점변화에 대한 처리를 하는 것이고 두 번째는 객체기술자에서 처리하는 경우로 볼 수 있다. 두 가지 경우에 따라 그림 7의 개체기술자와 기초스트림기술자 구조에서 확장영역을 사용해서 필요한 필드가 추가된다.

다음 그림 8은 MP4 파일 재생 도구인 OSMO플레이어상에서 기본적인 MPEG-4 시스템이 독립된 영상을 재생할 때의 구조이다. 기본적인 MPEG-4 시스템에서 이벤트 처리의 기본 단위가 객체기술자이기 때문에 기초스트림 단위에서 여러 개의 시점에 대한 기초스트림을 제어 하거나, 하나의 기초스트림 기술자 내에 있는 각 시점들의 비디오 스트림을 제어하기 위해서는 user interaction에 따라 기초스트림기술자, 객체기술자에서 이벤트를 처리할 수 있는 과정이 요구 된다. 노드 역시 각 시점 별로 기초스트림에 접근하거나, 객체스트림 내의 각 비디오 스트림에 접근 할 수 있는 독립적인 texture handler가 필요하다. 기존의 MPEG-4 시스템의 이벤트 처리 과정을 보면 동영상 스트림 데이터를 핸들링 하기 위해서는 texture와 관련된 노드의 texture

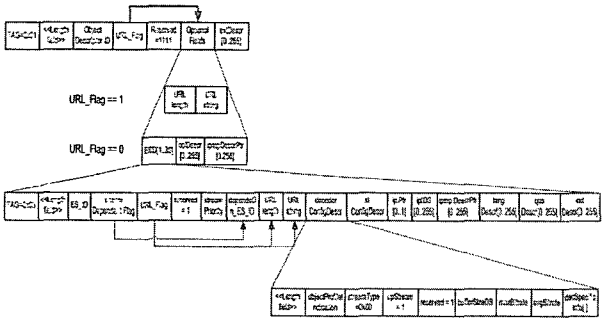


그림 7. MPEG-4 시스템의 객체기술자와 기초스트림 기술자의 구조

Fig. 7. The Object Descriptor and Elementary Stream Descriptor structure of MPEG-4 System.

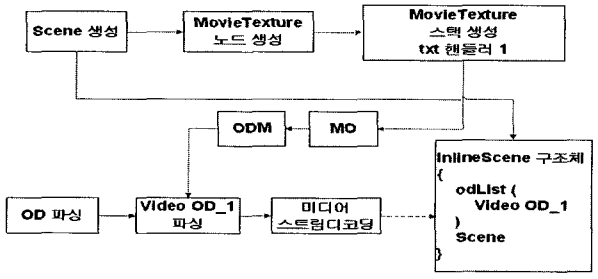


그림 8. OSMO플레이어에서 MPEG-4 시스템의 실행 구조

Fig. 8. Execution structure of MPEG-4 system in OSMO player.

handler의 자료구조를 사용하고, 발생하는 이벤트를 처리하기 위해서 texture_xx계열의 API를 이용한다.

texture_xx계열의 API는 texture handler에 등록된 MO(Media Object)를 통해서 ODM(Object Descriptor Manager)과 연결되어 있다. ODM은 객체기술자당 하나씩 생성되며 객체기술자 내에 속한 ES_ID에 해당하는 채널과 코덱을 관리한다. 노드는 MO를 통해서 객체기술자와 연결되고 노드에서 이벤트처리는 객체기술자 단위로 수행되며 ES_ID 또는 비디오 스트림으로 처리하기 위해서는 texture계열의 API의 구현과 ODM, MO, MC(Media Control), MM(Media Manager)계열의 수정이 필요하다. 다음 가, 나 항에서는 하나의 기초스트림으로 처리 되는 경우와 각각의 기초스트임으로 처리되는 경우에 대한 객체/기초스트림 기술자에 대한 확장 구조를 고려하고 분석한다.

가. 객체기술자가 하나의 기초스트림을 가지는 경우

(1) 기초 스트림 단위에서 Interaction을 제어할 때

기본적으로 MPEG-4 시스템에서는 기초스트림단위에서 시점 전환을 위한 제어를 할 수 있는 필드와 루틴이 없다. 따라서 기초스트림 단위에서 처리하기 위해서는 MVC와 송신측에 사용자 인터랙션에 의해 선택된 시점의 정보를 넘겨 줄 수 있는 필드, 동영상 스트림간의 상관관계를 표현 할 수 있는 필드와 처리 루틴 카메라 개수와 배치 타입에 대한 정보, 각 카메라에 대한 파라미터 정보등 다시점 동영상 처리를 위해 고려된 네 가지 사항에 적합한 필드 추가와 기초스트림내의 각 시점에 대한 비디오 스트림에 접근 가능한 루틴이 필요하다.

그리고 MovieTexture 노드에 user interaction을 받아들여, 시점에 대한 정보를 가질 수 있도록 새로운 필드를 추가해야 한다. 그림 9는 노드에서 시점 전환을

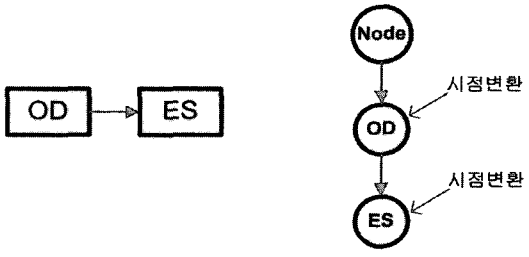


그림 9. 하나의 기초스트림을 가지는 구조일 경우의 객체/기초스트림 기술자 단위에서 처리

Fig. 9. Processing in Object/Element Stream Descriptor unit of structure that have one Element Stream.

위해 생성된 user interaction을 기초스트림 기술자 단위와 객체 기술자에서 처리하기 위해 확장 영역을 사용해서 필요한 필드가 추가 된 객체기술자/기초스트림기술자의 구조이다. 노드에서는 해당 시점에 정보가 들어있는 필드가 추가 되어야하고, 이 정보에 따라 기초스트림내의 각 시점의 비디오 스트림에 접근해서 처리되기 위한 실행구조가 추가되어야 한다.

기존의 MPEG-4 시스템의 실행구조인 그림 6과는 달리 Texture handler에 MES(Media Elementary Stream)가 등록되고 ESM(Elementary Stream Manager)을 통해 실제 기초스트림에서 MVC에 의해 복호화 된 각 비디오 스트림의 리스트와 연결되어 실행할 수 있는 루틴이 추가되어야한다. 그림 10은 확장된 객체/기초스트림 기술자의 구조를 보이고, 표 1에는 추가된 필드와 루틴의 명칭과 역할이 기술되어 있다.

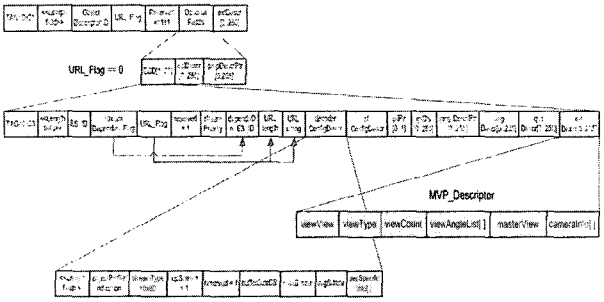


그림 10. 확장 된 MPEG-4 시스템의 객체기술자와 기초스트림 기술자의 구조

Fig. 10. Structure of Elementary Stream Descriptor and Object Descriptor of extended MPEG-4 System.

(2) 객체기술자 단위에서 시점 전환을 제어할 경우

다시점 영상 전환에 대한 사용자 인터랙션을 처리하는 객체기술자가 하나의 기초스트림을 가지는 구조에서의 경우(그림 9 참조)이다. 기초스트림에서 시점 전환을 제어하는 경우와 같이 객체기술자 자체 내에 사용자 인터랙션에 따른 기초스트림내의 비디오 스트림을 처리할 수 있는 루틴이 존재하지 않는다. 이를 처리하기 위해

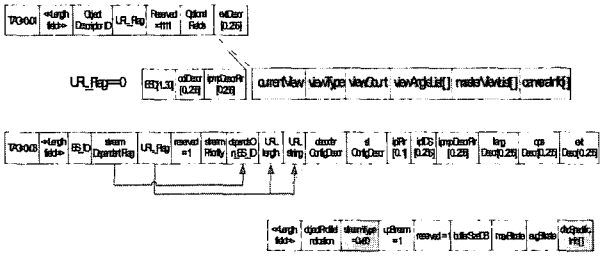


그림 11. 객체 기술자 단위에서 확장 된 MPEG-4 System의 객체/기초스트림 기술자의 구조

Fig. 11. Structure of Elementary Stream Descriptor and Object Descriptor of extended MPEG-4 System.

표 1. 객체기술자/기초스트림기술자의 확장된 필드 영역

Table 1. OD/ESD's extended field & Processing process.

	확장된 필드명		설 명
OD에서 추가된 Field	없음		Field가 추가 되지 않는다. Node에서 user interaction이 발생하면 OD를 거치지 않고 직접 ES에서 처리되기 때문이다
ES_D에서 추가된 Field	currentView		Node에서 발생한 user interaction 정보를 전달 받아 실제 MVC에서 출력되는 각 시점별 Video stream을 지칭할 정보를 가진다.
	viewCount		다시점 동영상을 입력 받는 카메라의 개수를 나타낸다.
	viewType		다시점 동영상 입력 시 사용되는 카메라 배치구조를 나타낸다.
	viewAnglelList[]		각 카메라 간에 이루는 각도 정보를 리스트로 나타낸다. 이는 각 시점간에 시점 전환 시 자연스러운 전환을 위해 Interpolation되는 영역에 가상 카메라를 위한 정보이다
	masterViewList[]		각 시점의 Master 시점이 되는 카메라의 정보를 나타낸다. 즉 다른 카메라 영상 정보 참조 없이 독립적으로 복호화되는 카메라 시점을 나타낸다.
	cameraInfo[]		각 카메라가 가지는 카메라 정보
	Decoder ConfigDesc	streamType	전송된ES가 MVC에 의해 각 시점별 영상의 정보를 가진 ES임을 나타내는 새롭게 추가된 stream_Tag정보를 가진다. 사용자 정의 영역의 "0x20"의 값을 가진다.
		decSpecificInfo	MVC에 대한 정보를 가진다.
추가된 루틴	MES (Media Elementary Stream)		Node의 Text 핸들러에 등록되어 기존의 MO와 같이 Text 핸들러에 등록되어 같이 ESM을 통해 실제 ES내의 각 시점별 video stream과 연결된다.
	ESM (Elementary Stream Manager)		ODM에서 하는 역할과 유사한 기능을 한다. Node의 Txt 핸들러에 추가된 MES 과 ESM을 통해서 ES_D의 currentView가 가리키는 시점의 video stream 과 실제적으로 연결시켜 주는 기능을 한다.

서는 기존의 MPEG-4 시스템에서 객체기술자의 시점 전환에 필요한필드와 기초스트림 내의 해당 시점의 비디오 스트림을 접근할 수 있는 루틴이 추가되어야 한다. 그리고 시점간의 상관관계에 대한 필드 역시 새롭게 추가 되어야 한다.

그림 11은 노드에서 시점 전환을 위해 생성된 사용자 인터랙션을 객체기술자 단위에서 처리하기 위해 확장 영역을 사용해서 필요한 필드를 추가한 객체/기초스트림 기술자의 구조이다.

기초스트림 단위에서 처리하는 경우와는 달리 객체기술자에 필요한 정보에 대한 필드를 추가하고, 기초스트림 기술자내의 "decoderConfig Descr" 에는 MVC 스트림을 인식할 수 있는 'stream Type'에 대한 정의와 디코딩 정보가 추가 되어야 한다.

MPEG-4 시스템 실행 구조는 한 개의 기초스트림에서 객체기술자 단위로 처리한다. 그러므로, 각 시점 별 비디오 스트림을 출력하는 MVC 미디어 디코더와 객체기술자 단위에서 시점 전환 제어와 상관관계 따른 정보 통해 기초스트림 내의 각 비디오 스트림까지 제어 할 수 있는 처리 루틴인 ESM이 그림 6에 추가 되어야 한다. 표 2에서 추가된 필드와 처리 과정을 기술 한고 있다.

나. 각 시점의 영상 정보가 각각의 기초스트림을 통해 전송되는 경우

(1) 객체기술자에 각 시점별 기초스트림이 연결될 때 기초스트림 단위에서 시점 전환을 제어할 경우

객체기술자 하나에 기초스트림이 여러 개인 경우로 MPEG-4 시스템에서는 기초스트림 단에서는 이벤트 처리를 할 수 없다. 기본적인 MPEG-4 시스템이 이벤트 처리의 기본 단위는 객체기술자이기 때문에 기초스트림단위에서 여러 개의 시점에 대한 기초스트림을 제어하기 위해서는 기초스트림에서 이벤트를 처리할 수 있는 과정이 요구되기 때문이다. 또한, 시스템은 각각의 기초스트림 기술자가 노드에서 발생되는 이벤트를 직접

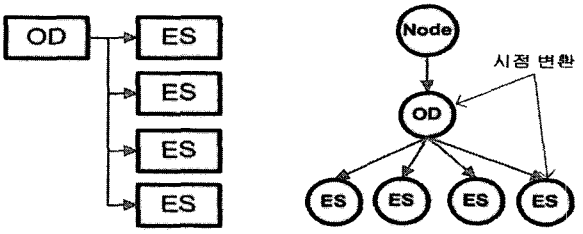


그림 12. 각 시점 별 기초스트림이 연결되는 경우 객체/기초스트림 기술자 단위에서 처리

Fig. 12. Processing in Object/Element Stream Descriptor unit that has individual ES.

표 2. 객체기술자/기초스트림 기술자의 확장된 필드 영역

Table 2. OD/ESD's extended field & Processing process.

	확장된 필드명		설 명
OD에서 추가된 Field	current View		Node에서 발생한 user interaction 정보를 전달 받아 실제 MVC에서 출력되는 각 시점별 Video stream을 지칭할 정보를 가진다.
	viewCount		다시점 동영상을 입력 받는 카메라의 개수를 나타낸다.
	viewType		다시점 동영상 입력 시 사용되는 카메라 배치구조를 나타낸다.
	viewAnglelList[]		각 카메라간에 이루는 각도 정보를 리스트로 나타낸다. 이는 각 시점간에 시점 전환 시 자연스러운 전환을 위해 Interpolation되는 영역에 가상 카메라를 위한 정보이다
	masterViewList[]		각 시점의 Master 시점이 되는 카메라의 정보를 나타낸다. 즉 다른 카메라 영상 정보 참조 없이 독립적으로 복호화되는 카메라 시점을 나타낸다.
	cameraInfo[]		각 카메라가 가지는 카메라 정보
ES_D에서 추가 된 Field	없음		Field가 추가 되지 않는다. Node에서 user interaction이 발생하면 OD에서 처리 되기 때문이다
	Decoder ConfigDesc	streamType	전송된 ES가 MVC에 의해 각 시점별영상의 정보를 가진 ES임을 나타내는 새롭게 추가된 stream_Tag정보를 가진다. 사용자 정의 영역의 "0x20"의 값을 가진다.
		decSpecificInfo	MVC에 대한 정보를 가진다.
추가된 루틴	ESM (Elementary Stream Manager)		ODM에서 하는 역할과 유사한 기능을 한다. Node의 Texthandler에 추가된 MO와 ODM을 통해서 연결 된 ES의 currentView가 가리키는 시점의 Video stream 과 실제적으로 연결시켜 주는 기능을 한다.

적으로 처리할 수 있는 구조가 되어야 하며, 노드 역시 각 시점 별로 기초스트림에 접근할 수 있는 독립적인 texture handler 가 필요하다. 기존의 MPEG-4 시스템이 이벤트 처리 과정을 보면 동영상 스트림 데이터를 핸들링 하기 위해서는 texture와 관련된 노드의 texture handler라는 자료구조를 사용하고, 발생하는 이벤트를 처리하기 위해서 texture_xx계열의 API를 이용한다. texture_xx계열의 API는 texture handler에 등록된 MO(Media Object)를 통해서 ODM(Object Descriptor Manager)과 연결되어 있다. ODM은 객체기술자 당 하나씩 생성되며 객체기술자 내에 속한 ES_ID에 해당하는 채널과 코덱을 관리한다. 노드는 MO를 통해서 객체기술자와 연결되고 노드에서 이벤트처리는 객체기술자 단위로 수행되며 ES_ID로 이를 처리하기 위해서는 texture계열의 API 구현과 ODM, MO, MC(Media Control), MM(Media Manager)계열의 수정이 필요하다. 그림 13은 노드에서 시점 전환을 위해 생성된 user interaction에 의해 기초스트림 단위에서 처리하기 위해 확장 영역을 사용해서 필요한 필드가 추가 된 객체/기초스트림 기술자구조이다. 각 기초스트림 기술자에서 반복되는 정보를 줄이기 위해 currentView, View Type field는 객체기술자에 추가한다.

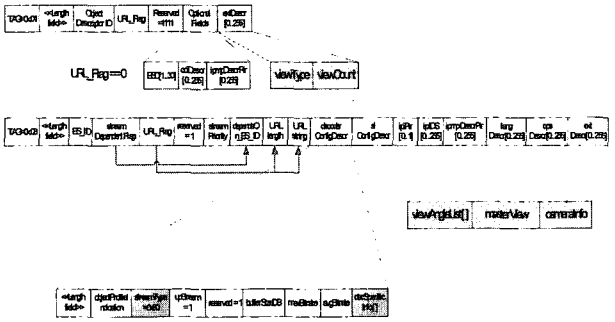


그림 13. 확장 된 MPEG-4 시스템의 객체기술자와 기초스트림 기술자의 구조

Fig. 13. Structure of Elementary Stream Descriptor and Object Descriptor of extended MPEG-4 System.

노드에서 시점 전환을 위해 생성된 user interaction에 의해 각 기초스트림내의 비디오 스트림에 접근해서 처리되기 위한 루틴이 실행구조에 추가되어야 한다. 기존의 MPEG-4 시스템의 실행구조와 달리 text handler에 MES가 등록되고 ESM을 통해 각 기초스트림에서 복호화된 각 비디오 스트림과 연결되어 실행되어야한다. 표 3은 추가된 필드와 루틴에 대해 기술 되어 있다.

(2) 객체기술자에 각 시점별 기 시점 전환 Interaction을 제어할 경우

하나의 객체기술자에 각 시점에 대한 여러 개의 기초스트림이 존재하는 구조에서 객체기술자 단위에서

표 3. 객체기술자/기초스트림 기술자의 확장된 필드 영역

Table 3. OD/ESD's extended field & Processing process.

	확장된 필드명		설 명
OD에서 추가된 Field	viewCount		다시점 동영상을 입력 받는 카메라의 개수를 나타낸다.
	viewType		다시점 동영상 입력 시 사용되는 카메라 배치구조를 나타낸다.
ES_D에서 추가된 Field	cameraInfo[]		각 카메라의 가지는 카메라 정보
	viewAngleList[]		각 카메라 간에 이루는 각도 정보를 리스트로 나타낸다. 이는 가 시점간에 시점 전환 시 자연스러운 전환을 위해 Interpolation되는 영역에 가상 카메라를 위한 정보이다
	masterView		각 시점의 master 시점이 되는 카메라 의 정보를 나타낸다. 즉 다른 카메라 영상 정보 참조 없이 독립적으로 복호화 되는 카메라 시점을 나타낸다.
	Decoder ConfigDesc	streamType	전송된ES가 MVC에 의해 각 시점별 영상의 정보를 가진 ES임을 나타내는 새롭게 추가된 stream_Tag정보를 가진다. 사용자 정의 영역의 "0x20"의 값을 가진다.
		decSpecificInfo	MVC에 대한 정보를 가진다.
추가된 루틴	MES (Media Elementary Stream)		Node의 Text 핸들러에 등록되어 기존의 MO와 같이 Text 핸들러에 등록되어 같이 ESM을 통해 실제 각 ES내의 video stream과 연결된다.
	ESM (Elementary Stream Manager)		ODM에서 하는 역할과 유사한 기능을 한다. Node의 Txt 핸들러에 추가된 MES 과 ESM을 통해서 ES_D의 currentView가 가리키는 시점의 video stream 과 실제적으로 연결시켜 주는 기능을 한다.

시점 전환을 처리하는 경우(그림 12 참조) 기존의 MPEG-4 시스템의 이벤트 처리의 기본 단위인 객체기술자를 사용하지만, 객체기술자 자체 내에는 사용자 인터랙션에 따라 기초스트림임을 선택하는 과정이 존재하지 않으므로 새로운 과정의 추가가 필요하다. 그리고 객체기술자에는 각 시점간의 상관관계를 표현 할 수 있는 정보가 존재 하지 않기 때문에 해당 정보를 추가시켜야 한다. 그림 14는 다시점 동영상 서비스를 제공

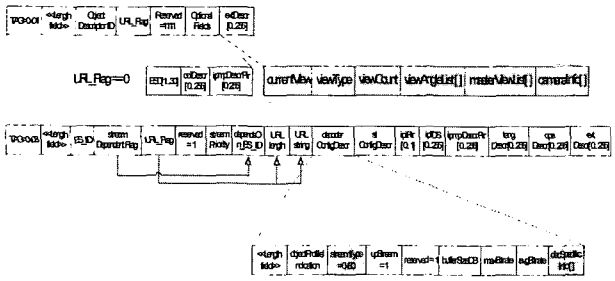


그림 14. 각 시점 별 기초스트림이 존재 할 때 객체기술자 단위에서 처리하는 경우 기초스트림기술자와 객체기술자 구조

Fig. 14. Structure of ES and OD when it is processed in OD unit, there is individual ES.

하기 위해 추가되어야 하는 객체/기초스트림 기술자의 구조를 나타낸다.

노드에서 시점 전환을 위해 생성된 user interaction에 의해 객체기술자 단위에서 각 기초스트림내의 비디오 스트림에 접근해서 처리되기 위한 루틴이 추가된 실행구조 가 필요하다. 기존의 MPEG-4 시스템의 구조인 text handler에 MO가 등록되고 ODM을 통해 각 기초스트림에서 복호화 된 각 비디오 스트림과 연결되는 것과는 달리 객체기술자에서 시점전환에 따른 기초스트림내의 비디오 스트림과 연결되어 실행되어야 한다. 표 4에서 추가된 필드와 루틴에 대한 설명이 되어 있다.

(3) 각 시점별 객체기술자를 사용하여 시점 전환을 제어 할 경우

각 시점별로 하나의 객체기술자와 기초스트림을 가지는 구조일 때, 객체기술자 단위에서 시점 전환을 처리하는 경우 기존의 MPEG-4 시스템의 이벤트 처리의 기본 단위인 객체기술자를 사용한다. 각각의 시점별로 객체기술자가 생성되고, 노드에서 발생하는 user interaction에 따라 객체기술자가 선택되어 해당 시점

표 4. 객체기술자/기초스트림 기술자의 확장된 필드 영역

Table 4. OD/ESD's extended field & Processing process.

	확장된 필드명		설 명
OD에서 추가된 Field	current View		Node에서 발생한 user interaction 정보를 전달 받아 실제 MVC에서 출력되는 각 시점별 Video stream을 지칭할 정보를 가진다.
	viewCount		다시점 동영상을 입력 받는 카메라의 개수를 나타낸다.
	viewType		다시점 동영상 입력 시 사용되는 카메라 배치구조를 나타낸다.
	viewAnglelList[]		각 카메라간에 이루는 각도 정보를 리스트로 나타낸다. 이는 각 시점간에 시점 전환 시 자연스러운 전환을 위해 Interpolation되는 영역에 가상 카메라를 위한 정보이다
	masterViewList[]		각 시점의 Master 시점이 되는 카메라의 정보를 나타낸다. 즉 다른 카메라 영상 정보 참조 없이 독립적으로 복호화되는 카메라 시점을 나타낸다.
	cameraInfo[]		각 카메라가 가지는 카메라 정보
ES_D에서 추가 된 Field	없음		Field가 추가 되지 않는다. Node에서 user interaction이 발생하면 OD에서 처리 되기 때문이다
	Decoder ConfigDesc	streamType	전송된ES가 MVC에 의해 각 시점별영상의 정보를 가진 ES임을 나타내는 새롭게 추가된 stream_Tag정보를 가진다. 사용자 정의 영역의 "0x20"의 값을 가진다.
		decSpecificInfo	MVC에 대한 정보를 가진다.
추가된 루틴	ESM (Elementary Stream Manager)		ODM에서 하는 역할과 유사한 기능을 한다. Node의 Txt 핸들러에 추가된 MO와 ODM을 통해서 연결 된 ES의 currentView가 가리키는 시점의 Video stream 과 실제적으로 연결시켜 주는 기능을 한다.

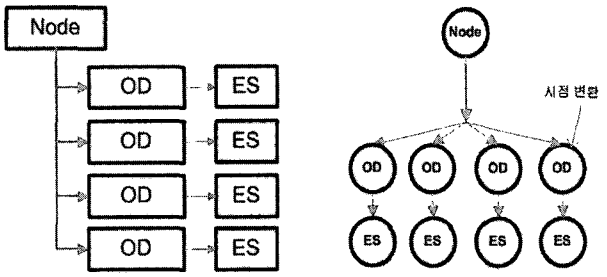


그림 15. 각 시점별 하나의 객체기술자와 기초스트림으로 구성되는 경우

Fig. 15. A case that it is consist of one OD and one ES about individual view-point.

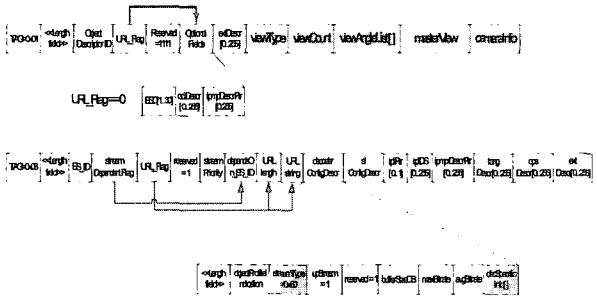


그림 16. 각 시점별 하나의 객체기술자와 기초스트림으로 구성되는 경우의 구조

Fig. 16. Structures of case that is consisted of one Object Descriptor and Element Stream Descriptor about individual view-point.

이 재생된다. 각 객체기술자에는 각 시점간의 상관관계를 표현 할 수 있는 정보가 존재 하지 않기 때문에 해당 정보를 추가 시켜야 한다. 그러나 하나의 노드에는 하나의 객체기술자가 연결되어, 영상을 뿌려주는 MovieTexture 노드에 있는 texture handler에 등록되는 MO를 통해 디코딩된 영상을 관리하게 된다. 그러나 기존의 MovieTexture 노드에는 하나의 texture handler만 존재하기 때문에 각각의 시점의 영상의 MO가 등록될 수 있는 texture handler가 추가 되어야 한다. 또한 하나의 노드에는 하나의 객체기술자만이 연결 될 수 있으므로 여러 개의 객체기술자가 등록될 수 있는 Movietexture 노드가 필요하게 된다. 그러나 각 시점간의 상관관계를 표현하기 위해서는 각 시점간의 객체기술자가 전송되어 디코딩 된 이후 가능하게 되므로 실시간으로 시점 전환이 불가능하다. 하나의 객체기술자와 기초스트림은 한 시점에 대한 디코딩정보를 가지고 있으므로 모든 시점간의 상관관계를 알 수 없다. 그러므로 사용자 인터랙션에 따른 시점 전환의 상관관계를 송신측으로 전송해서 필요한 시점에 대한 정보를 전송받도록 하기 위해서는 각 시점에 대한 정보를 모두 받은 후 그에 해당하는 객체/기초스트림 기술자를 디코딩 후에 가능하다. 그림 16은 다시점 동영상 서비스를 제공 하고 위해 추가되어야 하는 객체기술자

표 5. 객체기술자/기초스트림 기술자의 확장된 필드 영역

Table 5. OD/ESD's extended field & Processing process.

	확장된 필드명		설 명
OD에서 추가된 Field	current View		Node에서 발생한 user interaction 정보를 전달 받아 실제 MVC에서 출력되는 각 시점별 Video stream을 지칭할 정보를 가진다.
	viewCount		다시점 동영상을 입력 받는 카메라의 개수를 나타낸다.
	viewType		다시점 동영상 입력 시 사용되는 카메라 배치구조를 나타낸다.
	viewAnglelList[]		각 카메라 간에 이루는 각도 정보를 리스트로 나타낸다. 이는 각 시점간에 시점 전환 시 자연스러운 전환을 위해 Interpolation되는 영역에 가상 카메라를 위한 정보이다
	masterViewList[]		각 시점의 Master 시점이 되는 카메라의 정보를 나타낸다. 즉 다른 카메라 영상 정보 참조 없이 독립적으로 복호화되는 카메라 시점을 나타낸다.
	cameraInfo[]		각 카메라가 가지는 카메라 정보
ES_D에서 추가된 Field	없음		Field가 추가 되지 않는다. Node에서 user interaction이 발생하면 OD에서 처리되기 때문이다
	Decoder ConfigDesc	streamType	전송된 ES가 MVC에 의해 각 시점별영상의 정보를 가진 ES임을 나타내는 새롭게 추가된 stream_Tag정보를 가진다. 사용자 정의 영역의 "0x20"의 값을 가진다.
		decSpecificInfo	MVC에 대한 정보를 가진다.

/기초스트림 기술자의 구조를 나타낸다.

각 시점별 하나의 객체기술자와 기초스트림으로 구성되는 경우일 때 시스템의 실행과정에서도 사용자 인터랙션에 따라 이벤트를 처리 할 수 있는 필드와 루틴이 필요하다. 표 5는 추가된 필드와 루틴에 대한 설명을 하고 있다, 그리고 그림 15와 16에서 알 수 있듯이 객체기술자내에 시점 정보, 개수, 상관관계 리스트, 마스터 시점, 카메라 정보 등이 반복되는 문제점이 있다.

지금까지 다섯 가지 경우에 기존의 MPEG-4 시스템에서 확장 영역을 사용해서 다시점 동영상의 시점 변환 서비스를 위한 구조를 살펴보았다. 기본적으로 확장영역을 사용한다면 시점 전환 서비스가 가능하다는 것을 알 수 있었다. 그러나 기본적인 MPEG-4 시스템을 처리과정과는 크게 다르게 된다. 또한, 상관관계를 고려하지 못한다. 2절에의 그림 3과 5의 SDM 구조에서 알 수 있듯이 기존의 시스템에 확장영역을 사용하면 사용자 인터랙션이 발생할 때 해당 시점의 상관관계에 대한 정보를 미리 알 수 없기 때문에 현재 시점에 필요한 정보 외에 카메라 전체의 정보를 모두 전송 받아야한다. 이로 인해 시점 전환 시 일시적인 딜레이가 발생할 수 있다. 그리고 시점의 수가 적은 경우 미디어 객체를 디코딩하기 위한 과부하가 적지만, 실제 다시점 서비스의 경우 수십 개의 시점을 사용하기 때문에 전송 시 과부하와 디코딩 시에도 많은 자원낭비가 발생할 수 있다. 전체 시점에 대한 상관관계가 미리 전송되어 사용자 인터랙션에 의해 선택된 시점과 상관관계가 있는 시점을 전송 받아서 처리할 수 있는 구조가 되어야한다. 다음 Ⅲ장에서 이를 해결하기위해 새로운 객체기술자와 노드를 추가하여 해결한다.

Ⅲ. MPEG-4 System 기반의 상관관계가 고려된 객체 기술자 구조 제안

다시점 동영상 전환 서비스를 제공하기 위해 본 논문에서는 MVP_OD(MultiViewPointObjectDescriptor)와 MVPT(MultiViewPointTextue)를 추가한다.

기존의 MovieTexture는 하나의 비디오 영상 파일에 포함된 시간에 따라 변화하는 질감, 그림을 설정하고 비디오 영상과 질감, 그림 대응에 대한 매개변수를 조정하여 texture handler로 이벤트를 처리 할 수 있었다. 그러나 다시점 비디오 영상 구현을 위해서는 여러 개의 texture handler를 통해 각 시점 별 이벤트를 처리 할 수 있어야 한다. 그림 18는 다시점 비디오를 위한 노드

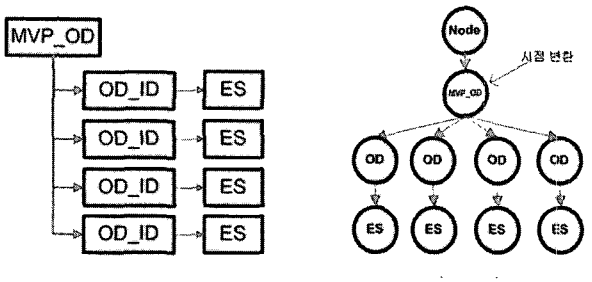


그림 17. MVP_OD에 객체기술자단위로 interaction하는 경우
Fig. 17. When do interaction by OD unit to MVP_OD.

MultiViewPointTexture {

exposedField	SFBool	loop	FALSE
exposedField	SFFloat	speed	1
exposedField	SFTime	startTime	0
exposedField	SFTime	stopTime	0
exposedField	MFString	url	NULL
field	SFBool	repeatS	TRUE
field	SFBool	repeatT	TRUE
eventOut	SFTime	duration_changed	
eventOut	SFBool	isActive	
exposedField	SFInt32	currentView	

}

그림 18. MVPT 노드 구조
Fig. 18. Structure of MultiViewPointTextue Node.

구조로써 현재 가리키는 시점 정보를 포함하고 있는 currentView필드를 포함하고 있다.

그리고 MVP_OD 는 MVPT 노드를 통해서 이벤트가 발생되면, 발생한 이벤트를 처리하기 위해서 OD_ID를 가지고 있는 odList를 가진다. odList에는 기초스트림을 가지는 객체기술자들로 구성된다. 따라서 각 시점의 정보를 가지는 객체기술자가 파싱되기 전에 MVP_OD가 파싱되어서 다시점에 대한 전체적인 정보를 얻게 된다. 여기서 얻은 시점간의 상관관계 정보를 통해 사용자 인터랙션으로 선택된 시점에 필요한 정보만을 전송 받아 디코딩된다. 그림 19는 객체 기술 프레임 워크를 나타낸 것이다. 최초에 MPEG-4 시스템에서 IOD(Initial Object Descriptor)가 처음으로 디코딩이 된다. 이를 통해 장면 기술자와 객체 기술자가 구성되고 객체기술자를 통해 ES_ID 참조되는 곳에 데이터를 가져와서 재생된다. 그러나 제안 된 구조는 다시점 전환 서비스를 제공하는 MVPT 노드와 MVP_OD를 통해 각 시점의 OD를 접근한다. 그림 20은 다시점 서비스를

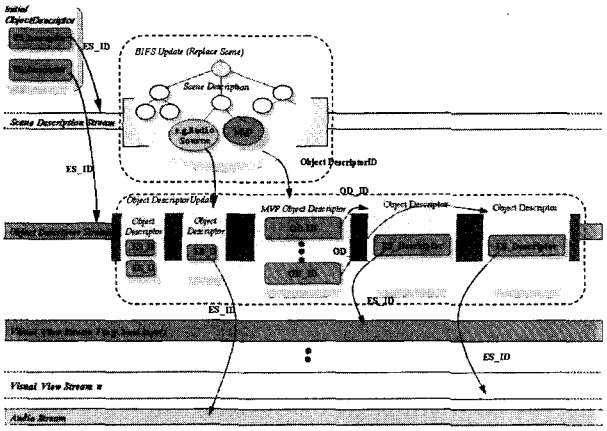


그림 19. 객체 기술 프레임 워크
Fig. 19. The Object Descriptor Framework.

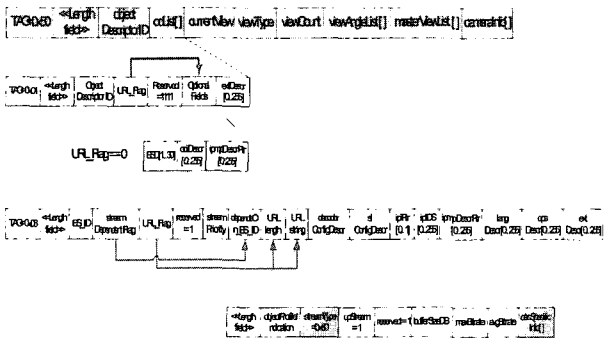


그림 20. 제안한 시스템의 OD/ES_D의 구조
Fig. 20. Structures of Object Descriptor/Elementary Stream Descriptor of system that is proposed.

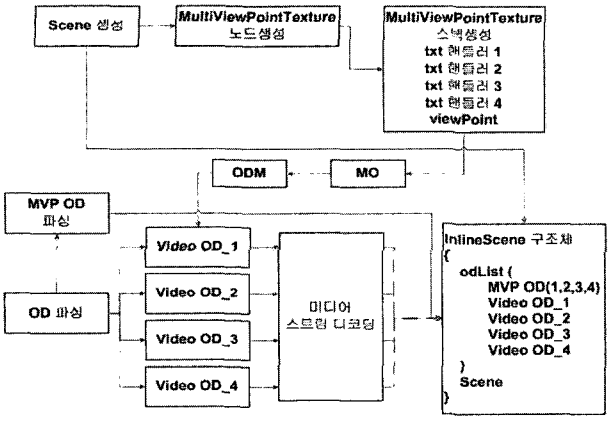


그림 21. MVP_OD 일 때 시스템의 실행 과정
Fig. 21. In case use MVP_OD in OSMO player, execution process of system.

제공하기 위한 객체기술자/기초스트림 기술자의 구조를 나타낸다.

그림 21은 제안된 OSMO플레이어로 MVP_OD를 이용하는 실행과정을 나타낸 것이다. MVP_OD는 MVPT 노드를 통해서 이벤트가 발생하고, 발생한 이벤트를 처리 하기 위해서 OD_ID와 각 시점의 ES_ID를 가지고 있는 odList를 가진다. odList에는 기초스트림을 가지는 객체 기술자들로 구성된다. 노드의 texture hander에 등록된 Media Object를 통해서 ODM과 연결되어 있고, 4개의 texture handler는 각 시점을 나타낸다. 여기서 하나의 handler가 선택되고. ODM은 객체기술자당 하나씩 생성되며, 객체 기술자내에 있는 기초스트림에 해당하는 채널과 코덱을 관리한다. 노드는 MO를 통해서 객체기술자와 연결되고 객체기술자가 가지고 있는 ES_ID를 통해 기초스트림에 접근한다.

IV. 실 험

다시점 전환 시스템 재생기는 GPAC 0.4 버전을 기반으로 구현 하였고, 다시점 비디오를 재생하기 위해서 소스를 변경하였다. GPAC은 ANSI C로 개발 되었고 MPEG-4 시스템 (ISO/IEC 14496-1) 표준을 기반으로 하는 멀티미디어 프레임워크이다. 실험은 제안된 구조인 MVP_OD일 때 객체기술자 단위로 시점 전환을 제어하는 경우로 실험을 하였다. 구현은 OSMO4를 기반으로 새로운 객체기술자인 MVP_OD를 인식할 수 있고, 이를 처리할 수 있는 루틴을 추가하였다.

그림 22는 객체기술자에 각 시점 별 기초스트림이 존재 할 때 추가된 MVP_OD를 통해 user interaction에 의한 시점전환 결과이다. 그림 23 (a)는 비교 결과로써, 일반적인 동영상을 나타내며, (b)는 이 논문에서 제안하는 다시점을 제공하는 영상에서 사용자 상호작용이 있을 때 시점이 자연스럽게 전환되는 것을 나타낸 결과이다. 사용자가 ES1의 영상을 t 시각에 시청도중 인터랙션을 통해 ES2의 시점에서 바라보는 t+1의 영상을 동

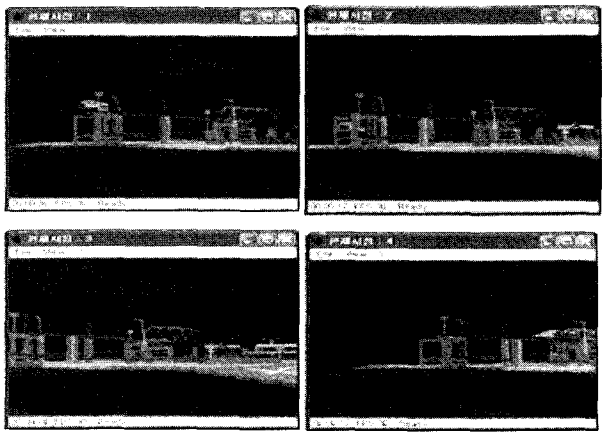
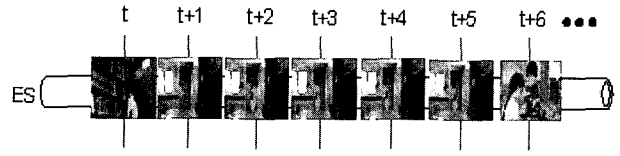
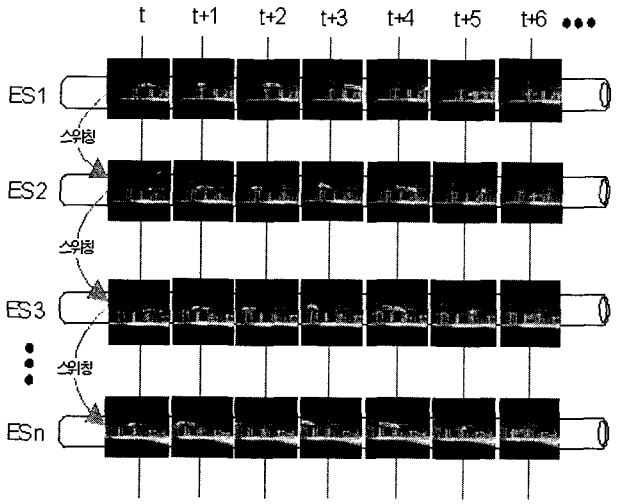


그림 22. 수정된 OSMO재생기 실행과정
Fig. 22. Execution process of modified OSMO player.



(a) 일반적인 OSMO 재생기
(a) General OSMO player



(b) 다시점 동영상 재생기 실험 결과
(b) Experiment result of MultiViewPoint OSMO player

그림 23. 기존의 재생기와 제안된 재생기의 결과 비교
Fig. 23. General player and result comparison of proposed player.

기에 맞추어 사용자 볼 수 있다. 다시점 비디오를 재생할 수 있는 재생기는 N개의 시점까지 고려되었고, 네 개의 시점을 실험 하였다. 시점 스위칭은 키보드의 입력과 마우스를 통해 받게 된다. 마우스를 클릭하면 다른 시점으로 전환되고, 키보드의 "R"키를 누르면 시점 1번부터 4번까지 자연스럽게 스위칭 된다.

V. 결론 및 향후 연구 방향

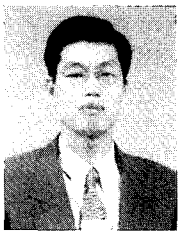
지금까지 MPEG-4 시스템과 다시점 동영상 기술에 대해서 알아보았다. MPEG은 동영상 데이터를 다루는데 적합한 도구이나, MPEG-2에서 다시점 동영상을 전송하기에는 무리가 있었다. 따라서 MPEG-4 기반으로 다시점 동영상을 전송하는 방법을 고려하게 되었다. 기존의 MPEG 4 플레이어들 역시 다시점 동영상을 전송하기 위한 정보들이 부족했다. 그래서 새로운 객체기술자를 추가하여 OSMO4 플레이어에 입력되는 MP4 파일이 다시점 동영상임을 알 수 있도록 하여, OSMO4에서 다시점 동영상 시점 전환이 가능하도록 하였다..

실험에서는 네 개의 영상에 대한 스위칭을 수행했으나, N개의 다시점 동영상으로도 확장이 가능하다. 그러나 아직 MPEG-4 3DAV 그룹에서 요구하는 요구사항은 모두 만족하지 못하고 있다. 앞으로 3DAV에서 제시하는 요구사항을 모두 만족하면서, 다시점 개의 동영상을 재생하는데 과부하가 소요되지 않는 방법으로의 모색이 필요하다. 추후 각 시점의 카메라 정보를 세부적으로 정의 하고, 앞으로 표준화 될 MVC에 따라 시점간의 상관관계정보를 제공하는 구조로 보완해야 한다. 이를 통해 3차원적으로 다시점 동영상 서비스가 가능하도록 해야 할 것이다.

참 고 문 헌

[1] M. Mambo, K. Usuda and E. Okamoto, "Proxy Signature : Delegation of the Power to Sign Messages," In IEICE Trans. Fundamentals, Vol. E79-A, No. 9, pp. 1338-1353, Sep., 1996.
[2] N7071, "Application and Requirements for 3DAV"
[3] GPAC, http://gpac.sourceforge.net
[4] N5539,"Applications and Requirements for 3DAV"
[5] ATTEST, http://www.iti.gr/db.php/en/projects/ATTEST.html
[6] Coding of moving pictures and associated audio : MPEG-4 Systems, ISO/IEC JTC1/SC29/WG11 N3054
[7] Coding of moving pictures and audio : MPEG-4 Systems, ISO/IEC JTC1/SC29/WG11 N3198
[8] Text for ISO/IEC FCD 14496-1 Systems, ISO/IEC JTC/SC29/WG11 N2201
[9] Requirements on Multi-view Video Coding v.4, ISO/IEC JTC/SC29/WG11 N7282
[10] Survey of Algorithms used for Multi-view Video Coding(MVC), ISO/IEC JTC/SC29/WG11 N6909
[11] Coding of audio-visual objects : Systems, ISO/IEC JTC/SC29/WG11 N4848
[12] MPEG, http://mpeg.telecomitalialab.com/
[13] ISO/IEC JTC1/SC29/WG11 (ISO/IEC 14496-1) W4848
[14] N5878, "Report on 3DAV Exploration"
[15] ISO/IEC 14496-1/Coding of audio-visual objects : Systems, final committee draft, ISO/IEC JTC/SC29/WG11 N2501

저 자 소 개

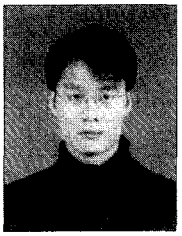


이 준 철(정회원)

1988년 경북대학교 자연과학대학 통계학과 졸업.
1991년 경북대학교 대학원 컴퓨터 공학과 석사.
1994년 경북대학교 대학원 컴퓨터 공학과 박사 수료.
1995년~현재 부천대학 인터넷과 부교수.
<주관심분야 :감시시스템, 멀티미디어, 다시점 동영상>

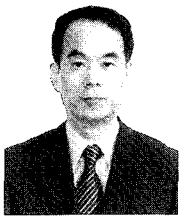
장 용 석(정회원)

1999년 부산외국어대학교 컴퓨터 공학과(공학사).
2001년 경북대학교 컴퓨터공학과 (공학석사).
2006년 경북대학교 컴퓨공학과 박사 졸업.
<주관심분야 :Metro Ethernet, Ethernet PON, WDM PON, MPEG >



이 정 원(정회원)

2004년 경성대학교 컴퓨터과학과 졸업.
2005년 경북대학교 컴퓨터공학 석사 졸업.
2006년~현재 경북대학교 컴퓨터공학 박사과정
<주관심분야 :Residential Ethernet, MPEG, 감시시스템, 이미지 처리>



김 승 호(정회원)-교신저자

1981년 경북대학교 전자공학과 졸업.
1983년 한국과학기술원 전산학과 석사.
1994년 한국과학기술원 전산학과 박사.
1985년~현재 경북대학교 컴퓨터공학과 정교수.
<주관심분야 :알고리즘, 감시시스템, 멀티미디어, 다시점 동영상>