

모바일 장치를 위한 MPEG-4 콘텐츠의 객체 및 장면 트리 변환

(Transforming Objects and a Scene Tree of MPEG-4
Contents for Mobile Devices)

김 상 옥 [†] 김 경 덕 ^{**} 이 속 영 ^{***}
(Sangwook Kim) (Kyungdeok Kim) (Sookyoung Lee)

요 약 본 논문에서는 모바일 장치를 위한 MPEG-4 콘텐츠 저장을 위하여 객체 및 장면 트리 변환을 이용한 방법을 제안한다. 제안하는 방법은 저작시 MPEG-4 콘텐츠의 객체와 장면 트리를 변환하여 모바일 장치에서 재생할 수 있도록 콘텐츠를 재구성한다. 이 방법은 모바일 장치의 작은 인터페이스에서 재생을 효율적으로 할 수 있도록 장면에 표현되는 각 객체를 기하 객체로 변환하고 그 외형을 축소하여 초기 로딩 시간을 감소시키며, 원 객체의 재생은 사용자가 재생 장면에서 해당 기하 객체를 선택할 때 재생된다. 그리고 제안한 방법을 기존 저작 도구에 적용하여, 모바일 장치의 재생기에서 효율적으로 MPEG-4 콘텐츠를 재생시킬 수 있음을 보였다.

키워드 : MPEG-4 콘텐츠, 객체 변환, 장면 트리 변환, 콘텐츠 적응, 콘텐츠 저작

Abstract We propose a method using a transformation of an object and a scene tree in order to author MPEG-4 contents for mobile devices in this paper. The method transforms media objects in the scene into geometry objects in order to reduce initial loading time of the contents, and reduces an external form of each object in a scene of the content for presenting it efficiently on the small interface of mobile devices. Therefore, the contents for mobile devices are reconstructed by the method. An original object in the scene is presented on a mobile device when a user clicks a related geometry object. The method was applied to a conventional authoring tool, so we could find that the method showed an efficient presentation of MPEG-4 contents on mobile devices.

Key words : MPEG-4 contents, Object transformation, Scene tree transformation, Contents adaptation, and Contents authoring

1. 서 론

기존의 MPEG-4 콘텐츠는 단순히 공중 전화망에서 영상 전송 정도에 이용되고 있으나, 현재에는 무선 통신의 발달로 모바일 장치에서 응용이 급증하고 있다. MPEG-4 콘텐츠는 멀티미디어 데이터에 대한 효율적인 전송 및 이용을 위한 표준으로서, 영상의 내용에 대한 이해를 바탕으로 한 내용기반 부호화에 중점을 두고 있

다. 이러한 내용기반 부호화에서는 영상 내용을 각각의 객체 단위로 구성하여 전송하며, 사용자의 의도에 따라 장면속의 객체 조작이 가능하고 콘텐츠의 내용을 사용자의 기호에 맞추어 지원할 수 있다[1-3].

MPEG-4 콘텐츠에서는 장면을 구성하는 멀티미디어 객체들을 독립적으로 다루고, 사용자와의 상호작용과 시간의 흐름에 따른 장면 변화는 시공간적인 정보를 표현하는 장면 기술 언어인 BIFS(Binary Format for Scene)를 이용한다[1]. MPEG-4 콘텐츠의 장면 저작은 BIFS에 대한 전문적인 지식이 필요하므로, 기존의 저작 도구에서는 직관적인 저작 방법이 주로 사용된다. 대부분의 MPEG-4 콘텐츠 저작 도구들은 데스크 탑 컴퓨터에서 사용되기 때문에 이들 저작 도구에서 만들어진 콘텐츠는 데스크 탑 컴퓨터의 재생기에 최적화 되어 있다. 특히 재생 화면 크기는 데스크 탑 컴퓨터와 모바일 장치를 비교하였을 때 많은 차이로 인하여, 데스크 탑의

· 이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음
(KRF-2003-002-D00304)

[†] 정 회 원 : 경북대학교 컴퓨터과학과 교수
swkim@cs.knu.ac.kr

^{**} 정 회 원 : 위덕대학교 컴퓨터멀티미디어공학부 교수
kdkim@uu.ac.kr

^{***} 비 회 원 : 삼성전자 무선사업부 연구원
sy8015.lee@samsung.com

논문접수 : 2004년 2월 28일

심사완료 : 2004년 12월 30일

재생기 화면에서 볼 수 있는 MPEG-4 콘텐츠를 모바일 장치에서는 똑같이 재생하기 어려운 문제가 발생한다. 그러므로 기존 저작 도구에서 생성된 MPEG-4 콘텐츠를 데스크 탑 컴퓨터와 같은 유선 환경은 물론 PDA나 휴대폰 등 다양한 모바일 환경에서도 MPEG-4 콘텐츠의 재생을 위해서는 모바일 장치에 적합한 MPEG-4 콘텐츠의 지원 방법이 필요하다[4-6].

본 논문에서는 모바일 장치를 위한 MPEG-4 콘텐츠 저작을 위하여 객체 및 장면 트리 변환을 이용하는 방법을 제안한다. 제안하는 방법은 모바일 장치의 재생 화면의 크기와 모바일 장치에 전송을 위한 파일 크기를 고려한다. MPEG-4 콘텐츠의 장면은 많은 멀티미디어 객체들을 포함할 수 있으며, 멀티미디어 객체가 많을수록 MPEG-4 콘텐츠 파일의 크기가 커지게 되고 이러한 큰 용량의 콘텐츠는 모바일 장치로 전송하기 어렵다. 또한 모바일 장치의 제한적인 하드웨어 성능으로 많은 멀티미디어 객체들을 한번에 재생하기 어렵다. 그러므로 제안한 방법은 MPEG-4 콘텐츠에 포함된 시각 멀티미디어 객체(비디오, 텍스트, 이미지 객체)들을 기하 객체로 표시하여 장면의 전체적 구성을 나타내고, 초기 전송 정보로 멀티미디어 스트림의 기초 정보만 가지도록 함으로써 초기 전송 파일의 크기를 감소시킨다. 이러한 장면 속 객체의 기하 객체화 및 이에 따른 초기 전송 파일 크기의 감소는 모바일 장치에서 재생을 효율적으로 지원한다.

MPEG-4 기반 콘텐츠에 대한 연구가 현재 활성화되고 있으나, 모바일 장치에 대한 저작 및 스케일링 기술에 대한 연구는 아직 초기 단계이며, MPEG-4 콘텐츠 저작과 콘텐츠 저작 모델에 대한 연구는 다음과 같다.

먼저 MPEG-4 콘텐츠 저작 시스템[1]은 윈도우즈 환경에서 MPEG-4 콘텐츠를 저작하는 시스템으로, 비디오, 이미지, 텍스트 객체들로 시청각 장면을 구성하고 마우스 클릭의 이벤트에 대한 장면 변화를 대화 상자를 이용한 직관적 저작을 지원한다. 그리고 객체 단위로 장면을 구성하는 MPEG-4 콘텐츠의 표현을 위하여 장면 트리를 내부 자료 구조를 이용한다. 장면 트리로 표현된 콘텐츠의 장면은 텍스트 형태의 MPEG-4 장면 기술과 인코딩 과정을 거쳐서 MPEG-4 콘텐츠로 최종 생성된다. MPEG-4 장면 저작 도구[7]는 장면 트리 구조 편집기, 노드 속성 편집기, 이벤트 편집기, BIFS 명령어 편집기, 미리보기 도구로 구성된 윈도우 환경 저작 도구이다. 사용자는 각각의 편집기를 이용하여 콘텐츠를 저작하며, 저작된 콘텐츠는 미리보기 창을 통하여 저작 결과를 확인하여 최종 생성된다. 각 편집기는 멀티미디어 객체의 시간 및 공간적인 속성을 각 편집기를 이용하여 설정하여야한다. 이 저작 도구는 BIFS 명령어와 사용자

상호작용을 지원하기 위한 JAVA 스크립트의 사용도 지원한다. 3D MPEG-4 저작 도구[8]는 OpenGL의 3D 함수들을 이용해서 MPEG-4 콘텐츠를 생성하는 저작 도구로서, 박스, 구, 원뿔 등의 3D 객체와 비디오 및 오디오와 같은 멀티미디어 객체를 이용하여 저작한 후 BIFS 언어로 변환한다. 생성된 BIFS 파일은 MPEG-4 인코더에 의하여 mp4 파일로 생성되고 MPEG-4 브라우저를 이용하여 3D MPEG-4 장면을 미리 보기 등의 기능을 제공한다. MPEG-4 콘텐츠의 복잡도 적용[9]은 MPEG-4 콘텐츠를 모바일 장치에 적용할 때, 콘텐츠의 장면내 객체들을 이벤트 정보와 미디어의 중요도에 따라 객체를 삭제함으로써 모바일 장치에서 재생시 자원의 사용량을 줄이는 방법을 제안한다. 여기서, 장면내 객체의 삭제를 위하여 장면 트리를 이용하였으며, 원 장면 트리에서 삭제된 정보를 포함한 장면 트리로 변환함으로써 모바일 장치에 적용할 수 있는 콘텐츠의 적용 방법을 제안한다. 하지만 객체 삭제에 의하여 정보의 손실을 가지는 단점이 있다. 모바일 멀티미디어 콘텐츠를 위한 저작 도구들[10]은 모바일 콘텐츠의 저작을 모바일 장치에서 직접 저작하는 방법 및 이미 작성된 템플릿을 이용하여 저작하는 방법을 제안한다. 멀티미디어 정보에 대한 모바일 적용의 콘텐츠 모델[6]은 제한적인 대역폭을 가지는 휴대폰 환경과 단말기가 멀티미디어 전송에서 가지는 제약 사항에 대한 연구로서, 새로운 추상 모델과 하이브리드(hybrid) 환경으로 멀티미디어 콘텐츠를 적용시키는 방법이다. 이 모델은 end-to-end 서비스를 최적화하기 위하여 멀티미디어 적용 분류법에 의하여 시맨틱(semantic)과 물리적인 요소들의 레이어 맵핑(mapping) 모델을 기술하고 있다. 적용 분류법으로는 시맨틱 적용과 물리적 적용의 두 분류로 나누며, 시맨틱 적용은 사용자와 서비스 제공자의 선택에 의하여 이루어지고 프레젠테이션의 의미적인 내용에 따라 적용시키는 방법이다. 물리적 적용은 멀티미디어 콘텐츠를 구성하는 객체들을 그들의 특성에 따라 또는 물리적인 QoS에 따라 적용시키는 방법이다. 범용 액세스를 위한 멀티미디어 인터넷 콘텐츠의 적용성[11]에서는 콘텐츠를 객체의 비트 수, 디스플레이 크기, 비트율, 쉐더 요구사항, 하드웨어 요구사항과 같은 항목들로 분석한다. 그리고 멀티미디어 콘텐츠를 구성하는 각각의 객체들을 정보 피라미드라는 프레임으로 제공하여, 콘텐츠를 다중 해상도와 모달리티(modality)로의 표현을 지원한다. 적용적 MPEG-4 콘텐츠의 스케일링[4]에서는 서로 다른 형태의 멀티미디어 데이터를 MPEG-4 콘텐츠로 저작 생산하는 기술과 이를 다양한 서비스 환경에서 적용 가능한 형태로 제공하는 콘텐츠 적용 기술에 대한 연구이다. 제안하는 적용 기술은 MPEG-4 콘텐츠에 포함된 멀티미

디어 객체 단위에서 이루어지는 적응 기술과, 네트워크 환경의 대역폭 변화에 대처하는 MPEG-4 콘텐츠 스트리밍 서비스에 대한 적응 기술을 제안한다. 즉 클라이언트 재생기의 자원 부족에 대하여 콘텐츠를 재생하기 위한 스케일링에 따라 주로 기하 객체를 삭제하고 비디오 및 오디오 객체 중심의 콘텐츠를 동적으로 재구성하여 재생한다.

그러나 이러한 연구들은 PDA와 같은 모바일 장치에서 재생되는 MPEG-4 콘텐츠를 저작할 수 없거나 객체 단위의 재구성이 가능하다는 기초 연구로서, PDA와 같은 모바일 장치에 사용자 상호작용이 가능한 MPEG-4 콘텐츠를 실제로 재생은 어렵다. 대부분의 관련 연구들은 모바일 장치에 적합한 콘텐츠의 저작을 지원하기에는 아직까지 미흡하다. MPEG-4 콘텐츠가 데스크 탑 환경에서 저작이 됨으로써, 저작 화면의 크기가 모바일 장치의 재생 화면 크기와의 차이에 의하여 콘텐츠 저작시 콘텐츠의 변환이 필요하다. 그러므로 본 논문에서는 모바일 장치를 위한 MPEG-4 콘텐츠를 저작하기 위하여, 원 미디어 객체를 기하 객체로 대체하고 대체에 따른 이벤트 정보를 가지도록 객체 및 장면 트리 변환을 이용하는 방법을 제안한다.

본 논문은 구성은 제2장에서는 모바일 장치를 위한 MPEG-4 콘텐츠의 저작과 객체 및 장면 트리의 변환을 설명한다. 제3장에서는 구현 방법과 변환된 객체 및 장면 트리를 가진 콘텐츠를 기존 콘텐츠와 비교 설명한다. 마지막으로 제4장에서 결론을 맺는다.

2. 객체 및 장면 트리의 변환

본 장에서는 MPEG-4 콘텐츠의 저작 방법과 MPEG-4 콘텐츠가 모바일 장치를 지원하기 위한 객체 및 장면 트리의 변환 방법을 기술한다.

2.1 MPEG-4 콘텐츠의 저작

기존 MPEG-4 콘텐츠 저작은 대부분 데스크 탑 컴퓨터에서 콘텐츠가 저작됨으로, 모바일 장치에서 재생시 데스크 탑 컴퓨터의 재생 화면과 모바일 장치의 재생 화면 크기의 차이로 어려운 문제가 발생한다. 그러므로 본 논문에서는 두 가지 측면에서 MPEG-4 콘텐츠의 저작을 고려한다. 먼저 모바일 장치의 재생 화면이 데스크 탑 환경에서보다 화면 크기가 축소됨에 따라 콘텐츠에 포함된 객체를 기하 객체로 대체하고 외형적 크기를 재생 화면 비율에 따라 축소하는 객체 변환을 고려하고, 다음은 모바일 장치에서는 많은 양의 멀티미디어 객체를 재생하기 어려우므로 콘텐츠 스트림의 전송 량 축소로서 장면 트리의 변환을 고려한다. 모바일 장치에서 MPEG-4 콘텐츠 재생을 위한 이러한 고려 사항을 만족시키기 위하여, MPEG-4 콘텐츠의 장면에 포함된 객체

를 분해하고 각 객체에 대하여 기하 객체화 및 외형 크기의 축소 변환과 모바일 장치의 제약 사항에 따른 장면 트리의 변환이 필요하다.

다음 그림 1은 본 논문에서 제안하는 모바일 장치를 위한 MPEG-4 콘텐츠의 저작 과정을 나타낸다. 저작 과정은 장면 저작, 객체 속성 분석, 장면 기술 정보 관리, 객체 및 장면 트리 변환, BIFS 및 객체 기술자 인코딩, mp4 파일 생성 단계로 이루어진다.

먼저 장면 저작 및 객체 속성 분석은 기존 멀티미디어 저작에서와 유사한 과정으로 멀티미디어 객체들을 시각 아이콘으로 나타내어 사용자의 직접 조작에 의하여 장면을 구성한다. 이 때 각 멀티미디어 객체의 가시적 특성을 설정하는 객체 속성은 마우스 클릭이나 텍스트 입력으로 설정한다. 또한 객체의 이벤트는 대화 상자를 이용하여 이벤트 및 행위의 종류를 설정한다.

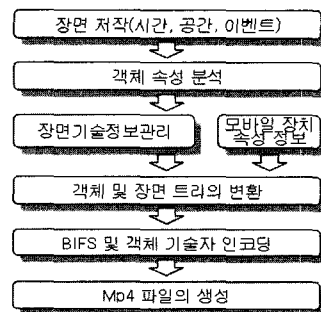


그림 1 모바일 장치를 위한 MPEG-4 콘텐츠 생성 과정

장면 기술 정보 관리 단계에서는 저작된 장면으로부터 장면 트리를 생성 및 관리하며, 시청각 객체와 각 객체의 시공간적인 관계를 나타내는 속성 객체를 장면 트리의 노드로 구성한다. 장면 트리의 관리를 위하여 시청각 객체들을 2차원 기하 객체, 이미지, 비디오 및 오디오 객체로 분류하여 관리하며, 저작 공간에 객체 생성이 이루어질 때마다 장면 트리에 객체 노드가 추가된다. 그리고 객체에 시공간 정보가 추가되면 속성 객체가 생성되어, 장면 트리에서 해당 객체 노드의 하위 노드로 추가되고 객체의 형태, 위치, 재생 시간 등에 대한 정보를 가진다.

객체 및 장면 트리 변환 단계에서는 모바일 장치를 위한 MPEG-4 콘텐츠의 생성을 위하여 콘텐츠 저작에서 생성된 객체와 이들 객체들이 가지는 정보들을 이용하여 장면 트리 노드의 값들을 재조정 한다. 여기서 모바일 정보 속성으로는 모바일 장치의 재생 화면 크기 속성을 이용한다. 대부분의 MPEG-4 콘텐츠 저작이 데스크 탑 컴퓨터에서 저작되므로 객체의 크기 및 위치와 같은 객체 속성 정보들은 모바일 장치에서 곧바로 적용

될 수 없다. 그리고 모바일 장치의 경우 가용 자원이 데스크 탑 컴퓨터 재생기보다 제한적이므로 장면이 포함되는 객체 외형의 축소 변환이 수행되어야 한다. 그림 2는 콘텐츠에서 객체 및 장면 트리의 변환 과정을 나타낸다. 기존 장면 트리에 기술된 각 객체의 위치 및 크기 값의 조정과 함께 이미지 및 비디오 객체의 기하 객체화에 의한 장면 트리의 변환을 나타낸다. 위쪽 박스내 점선으로 표시되는 부분은 이미지와 비디오 객체를 나타내며 아래쪽 박스내 점선으로 표시된 부분은 기하 객체로 변환된 것을 나타낸다. 변환된 기하 객체는 재생시에 사용자의 마우스 클릭에 의하여 원 장면 트리에 기술된 이미지 및 비디오 객체가 재생되도록 이벤트 및 텍스트 노드를 장면 트리에 추가 기술한다. 구체적인 객체 및 장면 트리의 변환 방법은 다음 2.2절에서 기술한다.

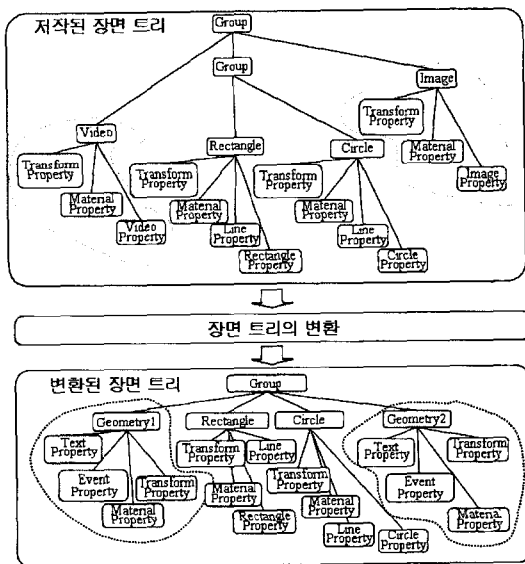


그림 2 장면 트리의 변환

BIFS 및 객체 기술자 인코딩 단계는 변환된 장면 트리를 입력받아 각 객체의 노드 정보 및 속성 노드 정보로부터 BIFS 텍스트 및 객체 기술자를 생성한다. 이 단계에서는 저작된 장면 트리에는 기술된 노드 외에도 저작되지 않는 많은 다른 노드들이 함께 표현되어야 함으로, BIFS 생성규칙을 이용하여 BIFS 문법에 맞도록 관련 노드들이 추가한다. BIFS 생성규칙은 기하 객체, 이미지 객체, 비디오 객체, 오디오 객체에 대한 생성규칙을 가진다. 그리고 각 객체 속성 정보를 가지는 객체 기술자 OD(OD: Object Descriptor)는 멀티미디어 스트림을 필요로 하는 객체에 대해서 특정 스트림을 나타내는 정보를 기술하여 OD 생성규칙에 따라 구문에 맞도록

생성된다. 멀티미디어 스트림을 필요로 하는 객체들로는 이미지 객체, 비디오 객체, 오디오 객체가 있다. OD 정보에는 객체의 ID 값과 그 객체의 URL 값 등으로 이루어지며, BIFS 및 OD는 텍스트 형식으로 기술되어진 후 비트 형식으로 인코딩한다.

mp4 파일 생성 단계에서 mp4 파일은 콘텐츠의 메타 데이터, BIFS, OD를 입력받아 생성된다. 메타 데이터 부분은 저작에 사용된 멀티미디어 객체들의 정보를 나타내는 BIFS와 OD 파일에 대한 위치 및 재생 시작 시간을 가진다. 그리고 오디오 및 비디오 객체인 경우에는 파일 내에서 각 프레임의 위치, 크기, 파일 생성 옵션 정보도 포함한다. 최종적으로 콘텐츠의 메타 데이터, BIFS, OD, 멀티미디어 파일을 함께 멀티플렉싱하여 mp4 파일을 생성한다.

2.2 모바일 장치를 위한 객체 및 장면 트리의 변환

모바일 장치에 적합한 콘텐츠를 제공하기 위하여, 객체의 변환은 장면 트리과 장면 트리로부터 변환에 필요한 객체들의 정보를 추출한 리스트 자료구조를 이용하여 수행한다. 본 논문에서 객체의 변환은 멀티미디어 객체를 기하 객체화하고 이러한 변환에 따른 이벤트와 텍스트 노드가 장면 트리에 추가한다. 여기서 장면 트리에 추가되는 이벤트 노드는 사용자가 실제 재생 화면에서 기하 객체에 마우스를 클릭할 때 해당되는 원 멀티미디어 객체의 재생을 지원하는 이벤트이다. 그리고 텍스트 노드는 생성된 기하 객체의 타입 정보를 나타내는 노드로서, 이미지에 대해서는 'I', 비디오에 대해서는 'V'라는 정보를 가지고 모바일 재생화면에 기하 객체와 함께 표시된다. 그림 3은 모바일 장치를 위한 객체 및 장면 트리의 변환 과정을 나타낸다.

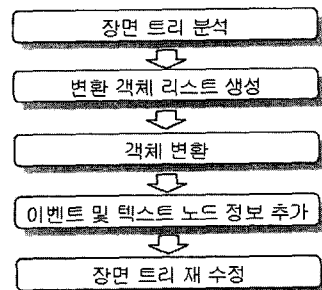


그림 3 객체 및 장면 트리의 변환 과정

객체 및 장면 트리의 변환 과정에서 장면 트리 분석 단계는 장면 트리를 탐색하여 객체 타입, 크기, 위치 정보를 추출한다. 추출된 정보를 이용하여 두 번째 단계인 변환 객체 리스트 생성 단계에서는 축소에 따라 변환시킬 객체들의 리스트인 변환 노드 리스트를 생성한다. 생

성된 변환 노드 리스트는 모바일 장치의 재생기에서 장면을 구성하는 객체들을 변환할 때 필요한 정보를 관리한다. 변환 노드 리스트의 각 노드는 객체의 타입, 위치, 크기, 시간 정보, 이벤트 저작인지 또는 시간 저작인지를 여부를 나타내는 플래그 속성 값으로 구성된다. 여기서 시간 정보는 객체가 화면에 나타날 순서 제어에 이용되며, 시간 저작은 멀티미디어 객체가 재생될 시간을 기술하는 것을 나타낸다. 또한 이 단계에서는 변환 노드 리스트와 관련하여 각 객체의 이벤트 정보를 가진 테이블도 함께 생성된다. 세 번째 단계인 객체 변환 단계에서는 변환 노드 리스트로부터 변환되어야 할 멀티미디어 객체의 기하 객체화 및 외형 축소와 모바일 장치의 재생 화면에서 객체 위치의 재설정을 지원한다. 네 번째 단계에서는 변환된 객체에 관련된 이벤트 및 텍스트 추가 정보를 생성하고, 다섯 번째 단계에서 생성된 추가 정보를 장면 트리에 반영하여 모바일 장치에 적합한 콘텐츠를 재구성을 지원한다.

세 번째 단계를 구체적으로 설명하면 다음과 같다. 객체 및 장면 트리 변환을 위한 축소 비율과 좌표 값의 재설정은 콘텐츠의 재생 장면 크기와 모바일 장치의 재생 화면 비율에 따라 이루어진다. 여기서, 축소 비율은 콘텐츠의 장면에 포함된 객체들의 위치 값을 이용하여 구한다. 즉 재생할 장면에 포함된 객체들 중에 최상단, 최하단, 최좌단, 최우단에 위치한 객체들의 좌표 값을 추출하여 최소 재생 화면의 크기 값을 계산한다. 콘텐츠가 재생하는 실제 장면의 재생 크기가 구해지면 이 크기와 모바일 장치의 화면 크기 값을 비교하여 축소 비율을 계산하고, 콘텐츠에 포함된 각 객체들을 비율대로 축소시킨다. 여기서 계산된 축소 비율은 객체의 크기뿐만 아니라 객체의 좌표 값에도 반영한다. 콘텐츠에 포함된 기하 객체(선, 원, 사각형)의 경우는 계산된 비율대로 축소하며, 그 외 텍스트, 이미지, 비디오 객체는 다음과 같이 변환이 이루어진다. 텍스트 객체의 경우는 모바일 재생기 윈도우의 크기에 맞게 축소했을 경우 축소되는 크기에 따라 글씨가 작아지게 되어 사용자가 알아보기 못하는 경우도 생길 수 있다. 이를 위하여 축소된 텍스트의 크기가 정해진 폰트 이하로 작아지게 되면 사용자가 인식할 수 없으므로 텍스트를 너무 작게 축소하지 않도록 한다. 이미지 객체의 경우는 외형의 크기가 동일한 사각형의 기하 객체로 대치되어 축소한다. 하지만 실제 실행 장면에서는 대치된 기하 객체를 사용자가 선택하면 실제 이미지 크기 그대로 재생한다. 생성된 기하 객체의 속성으로는 사각형 테두리 선 및 재질의 속성은 사각형의 일반적인 속성을 그대로 사용한다. 단, 본 논문에서는 이미지의 실제 크기가 모바일 장치의 재생 화면 크기보다 클 경우에는 고려하지 않으며, 이 때는 장면 트리에서 삭제

한다. 비디오 객체의 경우는 이미지 객체의 경우와 유사하게 외형의 크기가 동일한 사각형의 기하 객체로 대치 및 축소하며, 실제 실행시 사용자 클릭에 의하여 재생한다. 그리고 비디오의 재생 화면의 크기가 대부분 일반적인 모바일 장치의 재생 화면보다 작은 QCIF(144 x 176)와 CIF(288 x 352) 크기를 가짐으로 축소가 필요 없다. 그러나 비디오 품질 면에서는 모바일 장치의 재생기 자원이 부족하여 품질 저하가 발생할 수 있으며, 이러한 문제는 비디오 스케일러빌리티(scalability) 인코딩으로 해결할 수 있으나 본 논문에서는 고려하지 않는다. 버튼 선택에 따라 여러 이미지 객체를 활성화할 경우, 대부분 같은 위치에 재생하므로 여러 이미지 객체를 기하 객체로 나타내면 하나를 제외한 나머지 기하 객체들은 재생 화면에 나타나지 않을 수 있다. 이미지 객체를 대치한 기하 객체의 활성화는 버튼 객체가 포함된 이벤트 처리에 의하여 이루어진다. 버튼 객체에 의한 이미지의 재생에 대한 이벤트의 처리는 이벤트 정보 테이블을 활용한다. 이벤트 정보 테이블은 소스 객체, 목적 객체, 이벤트 타입, 액션 타입에 대한 정보를 가진다. 여기서, 이벤트 타입은 마우스 상태를 나타내며, 액션 타입은 마우스의 상태에 따라 수행되는 행위를 나타낸다. 단, 본 논문에서 사용하는 이벤트 타입은 대부분의 콘텐츠에서 주로 이용하는 마우스 상태에 대해서만 고려한다. 다음 그림 4는 버튼 객체와 이미지 객체를 가진 장면이 기하 객체화 된 상태를 나타낸다. 여기서 I_1 은 이미지 객체를 변환한 기하 객체를 나타낸다.

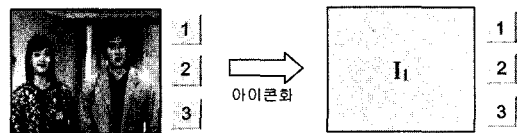


그림 4 버튼 객체에 대한 마우스 이벤트의 예

그리고 그림 5는 그림 4의 변환된 장면에서 버튼 선택에 따라 이미지 객체의 활성화에 대한 이벤트 정보 테이블과 변환 노드 리스트를 나타낸다. 여기서, 소스 및 목적 객체의 *Rect*, *Circle*, *Image*, *Line* 등은 기하 객체를, *Image1*, *Image2*, *Image3* 등은 재생되는 목적 객체를, 이벤트 타입의 *MouseClicked*, *MouseOver*은 마우스 상태를, 액션 타입의 *Active*, *Color*, *Transform*은 목적 객체의 재생, 색깔 변화, 위치 이동을 나타낸다. 즉, 소스 객체인 어떤 이미지 객체가 마우스 클릭에 의하여 목적 객체인 비디오 객체를 재생할 경우, 이미지 객체는 소스 객체가 되고 목적 객체는 비디오 객체로 설정된다. 이 때 이벤트의 타입은 *MouseClicked*이고 액션의 종류는 *Active*가 된다.

이벤트 정보 테이블

소스 객체	목적 객체	이벤트 타입	액션 타입
Rect1	Image1	MouseClicked	Active
Rect2	Image2	MouseClicked	Active
Rect3	Image3	MouseClicked	Active
.....
Circle4	Rect6	MouseOver	Color
Circle5	Line7	MouseClicked	Transform

변환 노드 리스트

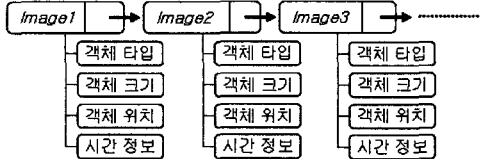


그림 5 이벤트 정보 테이블과 변환 노드 리스트

다음 그림 6은 이미지 객체를 저장할 때 생성되는 원 BIFS를 나타내며, 그림 7은 모바일 장치에서 재생을 위한 객체 및 장면 트리 변환 후의 BIFS를 나타내는 예이다. 그리고 그림 6과 7은 이미지 객체에 대한 부분적인 BIFS를 나타내며, 이벤트에 관련한 BIFS 표현은 그림 8에서 설명한다. 그림 7에 기술한 BIFS가 실제 모바일 장치에서 먼저 화면에 나타나는 부분이며 그림 8의 이벤트를 기술한 BIFS의 수행에 따라 그림 6에 기술한 BIFS에서 원 이미지 객체가 사용자의 마우스 클릭에 의하여 모바일 장치의 재생 화면에 나타난다. 여기서,

```

Group {
  children [
    DEF Switch1000 Switch {
      whichChoice 1
      choice [
        DEF Transform2D1000 Transform2D {
          translation - 236.00 - 142.00
          scale 1.00 1.00
          rotationAngle 0.00
          children [
            children [
              Shape {
                appearance Appearance {
                  texture ImageTexture {
                    url 1
                    repeatS TRUE
                    repeatT TRUE
                  }
                }
                geometry Bitmap { }
              }
            ]
          ]
        }
        DEF TimeS1000 TimeSensor {
          cycleInterval 3.00
          enabled TRUE
          loop TRUE
          startTime 0.00
          stopTime -1.00
        }
        DEF Condi1 Conditional {
          buffer {
            REPLACE Transform2D1000.translation BY - 236.00 143.00
          }
        }
      ]
    }
    ROUTE TimeS1000.isActive TO Condi1.activate
    UPDATE 00 {
      {
        objectDescriptorID 1 muxScript image1-1.scr
      }
    }
  ]
}
    
```

그림 6 이미지 객체의 원 BIFS

```

Group {
  children [
    DEF Switch3000 Switch {
      whichChoice 0
      choice [
        DEF Transform2D3000 Transform2D {
          translation - 236.00 - 142.00
          scale 1.00 1.00
          rotationAngle 0.00
          children [
            Shape {
              appearance Appearance {
                material DEF Material2D3000 Material2D {
                  emissiveColor 0.75 0.75 0.75
                  filled TRUE
                  transparency -1.00
                  lineProps DEF LineProperties3000 LineProperties {
                    lineColor 0.00 0.00 0.00
                    width 1.00
                    lineStyle 0
                  }
                }
              }
            }
            geometry Rectangle {
              size 175.00 316.00
            }
          ]
        }
        DEF TouchS3000 TouchSensor {
          enabled TRUE
        }
      ]
    }
  ]
}
    
```

그림 7 이미지 객체가 변환된 후 BIFS

```

DEF TouchS3000 TouchSensor {
  enabled TRUE
}
DEF Condi1 Conditional {
  buffer {
    REPLACE Switch3000.whichChoice BY 1
  }
}
DEF Condi2 Conditional {
  buffer {
    REPLACE Switch1000.whichChoice BY 0
  }
}
ROUTE TouchS3000.isActive TO Condi1.activate
ROUTE TouchS3000.isActive TO Condi2.activate
    
```

그림 8 이벤트 관련 BIFS

Touch3000은 그림 7의 BIFS를 가리키며, TouchSensor는 마우스 클릭 이벤트에 대한 표현이다. 즉 TouchSensor는 특정 객체에 대하여 마우스의 포인터가 그 객체 위에 놓여지던지 혹은 클릭을 할 경우 특정 행위를 수행하도록 지시하는 BIFS 명령어이다. 그러므로 TouchSensor가 활성화 되면 행위를 정의한 Condi1과 Condi2의 BIFS 명령이 수행된다. Condi1은 기하 객체가 화면에서 보이지 않도록 하는 행위이고, Condi2는 실제 객체를 화면에 나타내는 행위를 정의한다. 즉, 사용자가 기하 객체를 마우스로 클릭하면 선택된 기하 객체는 사라지고, 기하 객체로 대체되었던 실제 객체가 모바일 장치의 화면에 재생된다.

다음 프로시저는 Transformation()은 객체 변환 및 장면 트리 변환 과정을 나타내는 프로시저이다.

```

procedure Transformation()
{ generate the list TNL of transforming nodes from
  the original scene tree;
  // rr : a reduction rate in order to present an
  object on a mobile device.
  for (each object in TNL) do {
    switch(object) {
      geometry_object : { generate a reduced
        geometry object go using rr; }
      image_object : { generate a reduced
        rectangle object ro using rr; }
      video_object : { generate a reduced rectangle
        object ro using rr; }
      text_object : { reduce the text object to using
        rr, but must be visible on the screen;}
    }
    modify attributes(time, source object, target
      object) of the object in the TNL;
    modify the event node table ENT according
      to change of attributes of the object;
    regenerate the transformed scene tree TST
      using the TNL and the ENT,
    and add the event and text nodes which
      relate with the object in the TST;
  }
}
    
```

3. 구현 및 결과

모바일 장치를 위한 MPEG-4 콘텐츠의 객체 및 장면 트리의 변환은 기존 저작 도구[12]를 응용하였다. 그림 9는 저작 도구에서 작성한 콘텐츠를 IM1-2D 재생기에서 재생하는 장면을 나타내며, 그림 10은 그림 9의 MPEG-4 콘텐츠의 장면이 스케일링 되어 288x352 사이즈의 화면을 가진 PDA 재생기[13]에서 재생된 화면을 나타낸다. 여기서, 사용한 PDA는 내장형 리눅스 커널 Kernel 2.4.5-rmk6-np1을 탑재한 iPAQ 3660이며, PDA에서 어플리케이션 개발을 위한 C와 C++ 크로스 컴파일러와 GLIB 라이브러리를 사용하였다[14]. 그리고 MPEG-4 비디오의 렌더링과 사용자 인터페이스를 위하여 QT/Embedded 툴킷을 이용하였다[15]. 그림 10에서 처럼 PDA 재생기에서는 데스크 탑 시스템에 의해 저장된 각 객체는 기하 객체로 표시되며, 사용자가 재생을 원하는 객체를 클릭 함으로써 재생시킬 수 있다.

그림 11은 비디오 객체를 포함한 MPEG-4 콘텐츠를 IM1-2D 재생기와 PDA 재생기[13]에서 재생 과정을 나타낸다.



그림 9 IM1-2D 재생기의 재생 화면

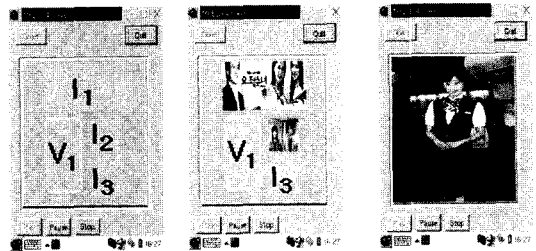


그림 10 PDA 재생기에서의 장면의 아이콘으로 나타난 화면

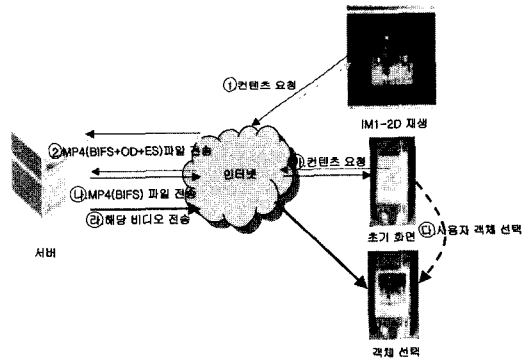


그림 11 비디오 객체를 포함한 MPEG-4 장면의 재생

IM1-2D 재생기는 MPEG-4 장면을 요청하면 서버에서 해당 mp4파일을 전송한다(재생과정: ①→②). 같은 MPEG-4 장면을 PDA 재생기에서 요청하면, 서버에서는 해당 mp4 파일을 전송받고, 이 때 전송받은 mp4 파일은 비디오 객체를 상징하는 기하 객체를 기술한 BIFS 파일만이 포함되어 있다. 그래서 사용자가 비디오 객체를 나타내는 기하 객체를 선택하면 그 이벤트 정보에 의해 서버로부터 비디오 객체를 전송 받아 PDA에서 재생한다(재생과정: ③→④→⑤→⑥).

그림 12와 13은 기하객체와 이미지 객체의 개수에 따른 BIFS 파일의 크기와 mp4 파일의 크기를 비교한 실험이다. 그림 12는 BIFS 파일 크기의 변화량이고 그림 13은 mp4파일 크기의 변화량을 나타낸다.

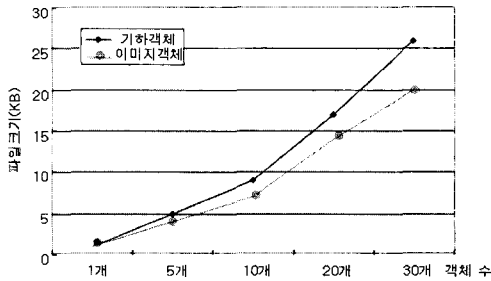


그림 12 BIFS 파일 크기의 변화

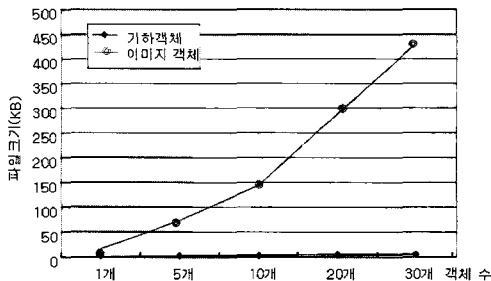


그림 13 mp4 파일 크기의 변화

그림 12에서 나타나듯이 기하객체와 이미지객체 두 경우 모두 객체 개수에 따른 BIFS 파일 크기의 차이는 크지 않음을 알 수 있다. 그러나 그림 13의 mp4 파일의 크기를 비교해 보면 기하 객체의 경우는 객체 개수가 증가함에 따라 mp4 파일 크기 변화량이 작은 반면에 이미지 객체의 경우 객체 개수가 증가함에 따라 mp4 파일 크기가 기하급수적으로 증가함을 나타낸다. 일반적으로 이미지 객체와 같은 멀티미디어 객체들로 구성된 MPEG-4 콘텐츠는 파일 생성에서 BIFS, OD 파일과 각 멀티미디어 객체의 원시 스트림이 함께 멀티플렉싱되어 재생기로 전송된다. 그러므로 멀티미디어 객체가 많을수록 생성된 MPEG-4 콘텐츠의 파일 크기가 증가하게 된다. 하지만 변환된 MPEG-4 콘텐츠의 장면은 기하 객체로 구성됨으로 모바일 재생기에서 BIFS 정보만으로 장면을 표현할 수 있으며 사용자의 선택에 따라 멀티미디어 객체를 재생한다. 이러한 BIFS 파일은 객체 스트림을 포함하지 않으므로, 파일 크기가 작고 네트워크의 대역폭이나 단말기의 자원 사용량을 감소시킨다. 그리고 사용자 QoS에 대한 실험을 위해 30개의 이미지 객체를 포함한 MPEG-4 콘텐츠의 장면에 대해 데스크탑 재생기에 전송이 되는 시간을 측정하여 동일 시간동안 PDA 재생기로 전송이 되는 장면의 이미지 객체 수를 측정하였다. 여기서 PDA의 하드웨어의 사양은 위에서 언급한 사양과 동일하며, 데스크탑 환경은 펜티엄 4에 윈도우 XP를 탑재한 IM1-2D 재생기를 사용하였다.

실험에서 데스크탑 재생기로의 전송시간이 33.67ms로 측정되었고 실험 결과 동일 시간 동안 PDA 재생기에서는 약 10개 정도의 이미지를 선택하여 재생할 수 있었다. 실험에 사용된 이미지는 파일 용량 및 크기가 작은 객체이므로 10개 정도 전송이 가능하지만 실제 장면의 경우 다양한 크기의 이미지를 사용하므로 이 수치보다 더 작은 결과가 나올 수 있다. 하지만 PDA 재생기의 경우 화면 크기가 매우 작다는 점과 하드웨어 사양이 제한적이라는 점을 감안했을 때 10개 정도의 이미지를 재생시킬 수 있다는 점은 상당히 의미 있는 결과이다.

결과적으로 객체와 장면 트리의 변환으로 MPEG-4 콘텐츠는 멀티미디어 객체 스트림을 함께 전송하지 않기 때문에, mp4 파일의 크기가 상대적으로 작아지고 초기 전송되는 BIFS 파일만으로 전체 장면을 구성할 수 있다. 또한 상대적으로 짧은 초기 로딩 시간을 가지는 장점으로, 모바일 장치에서 MPEG-4 콘텐츠를 더욱 효율적으로 재생할 수 있다.

4. 결론

본 논문에서는 모바일 장치를 위한 MPEG-4 콘텐츠를 저장하기 위한 객체 및 장면 트리의 변환 방법을 기술하였다. 제안한 변환 방법은 장면 트리에서 변환되어야 할 객체 정보들을 변환 노드 리스트로 표현하고, 이 변환 노드 리스트의 정보를 이용하여 자동으로 모바일 장치에 재생이 적합한 MPEG-4 콘텐츠를 생성한다. 제안한 방법은 멀티미디어 객체 자체를 변경하는 것이 아니므로 원 콘텐츠의 정보 손실이 거의 없다. 또한 기존 MPEG-4 저작 도구에 쉽게 적용할 수 있으며, BIFS에 대한 전문적인 지식이 없이도 한번의 저작으로 원하는 단말기에 적합한 콘텐츠를 재생할 수 있다.

향후 연구 방향은 멀티미디어 객체의 기하 객체화로 장면의 의미 전달이 부족할 수 있으므로 텍스트 정보의 추가 기술과 이미지 및 오디오/비디오의 확장성 지원 인코딩을 지원하는 기술의 개발이다.

참고 문헌

- [1] K. Cha, H. Kim, and S. Kim., "The Design and Development of MPEG-4 Contents Authoring System," Journal of KISS, Vol. 7, pp. 309-311, 2001.
- [2] WG11(MPEG), MPEG-4 Overview (V.16 La Baule Version) document, ISO/IEC JTC1/SC29/WG11 N3747, October 2000.
- [3] WG11(MPEG), MPEG-4 Overview (V.18 Singapore Version) document, ISO/IEC JTC1/SC29/WG11 N4030, March 2001.

- [4] K. Cha, "A Scalability for Adaptive MPEG-4 Contents," Dissertation for The Degree of Doctor of Philosophy, June 2003.
- [5] S. Lee, K. Cha and S. Kim "An MPEG-4 Contents Authoring for Mobile Devices," Proceedings of 2003 HCI, pp. 402-405, Korea, February 2003.
- [6] M. Metso, A. Koivisto and J. Sauvola, "Content Model for Mobile Adaptation of Multimedia Information," Processing of IEEE 3rd Workshop on Multimedia Signal, pp. 39-44, Sept. 1999.
- [7] M. Shieh, K. Perngand and W. Chen, "The Design and Implementation of A Visual MPEG-4 Scene-Authoring Tool," 2001 Proceedings of Workshop and Exhibition on MPEG-4, pp. 21-24, June 2001.
- [8] P. Daras, I. Kompatsiaris and M. Strintzis, "MPEG-4 Authoring Tool for The Composition of 3D Audiovisual Scenes," Proceedings of Second International Workshop on Digital and Computational Video, pp. 110-117, Feb. 2001.
- [9] K. Cha, "Content Complexity Adaptation for MPEG-4 Audio-visual Scene," IEEE Tans. on Consumer Electrnocs, Vol. 50, No. 2, pp. 760-765, 2004.
- [10] T. Jokela, "Authoring Tools for Mobile Multimedia Content," Proc. of 2003 Int. Conf. on Multimedia and Expo, Vol. 2, pp. 637-640, 2003.
- [11] R. Mohan, J. Smith, and C. Li, "Adapting Multimedia Internet Content for Universal Access," IEEE Transactions on Multimedia, Vol. 1, pp. 104-114, March 1999.
- [12] K. Cha and S. Kim, "MPEG-4 STUDIO: An Object-Based Authoring System for MPEG-4 Contents," Multimedia Tools and Applications, 2003.
- [13] J. Yeou, J. Jung, Y. Shin and S. Kim, "The MPEG-4 Video Player for PDA," Proceeding of the 29th KISS Fall Conference, October 2002.
- [14] <http://www.mizi.com>
- [15] <http://www.trolltech.com>



김 경 덕

1989년 경북대학교 자연과학대학에서 학사학위를 취득. 1991년 경북대학교에서 컴퓨터공학으로 석사학위를 취득. 1999년 경북대학교에서 컴퓨터공학으로 박사학위를 취득. 1991년 2월부터 1996년 5월 까지 (주)웨스트시스템 연구소 연구원. 2000년 3월부터 현재 위덕대학교 컴퓨터멀티미디어공학부 조교수로 재직. 관심분야는 멀티미디어 컴퓨팅, 컴퓨터 언어, 원격 교육 등



이 숙 영

2002년 2월 경북대학교 컴퓨터과학으로 학사학위를 취득. 2004년 2월 경북대학교 컴퓨터과학으로 석사학위를 취득. 2004년 1월부터 현재 삼성전자 연구원으로 재직



김 상 옥

1979년 2월 경북대학교에서 컴퓨터과학으로 학사학위를 취득. 1981년 2월 서울대학교에서 컴퓨터과학으로 석사학위를 취득. 1989년 2월 서울대학교에서 컴퓨터과학으로 박사학위를 취득. 1988년 3월부터 현재 경북대학교 컴퓨터과학과 교수로 재직. 관심분야는 인간과 컴퓨터 상호작용, 컴퓨터언어, 분산 멀티미디어 컴퓨팅 등