

OODBMS 기반 확장 용이 MPEG 비디오 브라우저

(Extendible MPEG Video Browser based on OODBMS)

송용준[†] 김형주^{**}

(Yong-Jun Song) (Hyoung-Joo Kim)

요약 최근 대용량성과 비정형성을 특징으로 하는 멀티미디어 데이터의 사용이 크게 늘어나면서 효율적인 저장 관리 기법이 필요하게 되었다. 그에 대한 해결 방법의 한가지로서 OODBMS를 이용한 멀티미디어 데이터의 저장 관리 연구가 활발히 진행되고 있는데, 본 연구에서는 OODBMS를 이용하여 비디오 데이터의 표준인 MPEG 데이터 관리 기법을 연구하고, 그러한 데이터를 여러 사용자들이 효율적으로 접근하여 편리하게 사용할 수 있고 향후 확장이 용이한 환경을 제공하는 것을 목표로 한다. 이를 위해 본 논문에서는 대용량의 멀티미디어 데이터를 효율적으로 저장 관리하는 SOP OODBMS를 기반으로 클라이언트/서버 구조의 확장 용이 MPEG 비디오 브라우저를 제안하여, 사용자들이 편리하게 MPEG 비디오 데이터를 이용할 수 있으며 개발자가 새로운 기능이나 새로운 비디오 포맷의 지원을 쉽게 추가할 수 있도록 하였다.

Abstract The efficient method for storing and managing large-scale multimedia data is very crucial because multimedia data have characteristics such as being unnormalized and complicated. A flurry of works has been done on multimedia data management using OODBMS. This paper aims at the efficient management of MPEG data, which is the standard video data format, using OODBMS and the provision of environment where several users can access to use the multimedia data efficiently and the developers can extend the system easily. Thus we present the extendible MPEG video browser with client/server architecture based on SOP OODBMS which can efficiently store and manage the multimedia data with high volume. With the presented MPEG video browser, users can manipulate MPEG video data conveniently and the developers can easily add new functions or new video formats.

1. 서론

최근 컴퓨터 기술의 급속한 발전으로 인해서 이전의 문자 위주의 사용자 환경에서 벗어나 이미지, 그래픽, 오디오 및 비디오 등과 같은 멀티미디어 데이터를 이용하여 점차 실세계와 유사한 사용자 환경을 제공하는 것이 자연스러운 추세이다. 현재 컴퓨터 구조 분야, 통신 분야, 데이터베이스 분야, 사용자 인터페이스 분야 등

많은 관련 분야에서 활발한 연구가 수행되고 있지만, 비정형성과 대용량성을 특징으로 하는 멀티미디어 데이터를 효율적으로 처리하여 저장 관리하는 것은 아주 어려운 문제이며 앞으로 해결해야 하는 문제들이 많이 남아있다. 특히, 비디오 데이터는 포함하는 정보량이나 실제 활용성으로 볼 때 멀티미디어 데이터의 핵심이라 할 수 있지만 관련 기술의 발전은 가장 느린 편이다.

비디오 데이터는 이미지 데이터인 프레임들의 연속으로 구성되는데, 사람이 자연스러운 비디오로 느끼기 위해서는 1초에 15개~30개의 프레임들이 보여져야 한다. 따라서 비디오는 단 1분 분량이라 할지라도 900개~1,800개라는 많은 수의 프레임들을 갖게 되는데, 이와 같은 대용량의 비디오 데이터를 그대로 사용하기 보다는 압축 기법을 이용함으로써 저장에 필요한 디스크 크기를 줄이고 통신상에서의 전송 속도를 향상시키는 것

· 본 연구는 한국과학재단의 목적기초 특정연구 No.95-0100-23-04-3

"다중매체 요구형 시스템 기술에 관한 연구"의 지원에 의한 것임

† 비 회 원 : 서울대학교 컴퓨터공학과

yjsong@oopsla.snu.ac.kr

** 종신회원 : 서울대학교 컴퓨터공학과 교수

hjk@oopsla.snu.ac.kr

논문접수 : 1997년 9월 30일

심사완료 : 1998년 4월 15일

이 일반적이다. 이러한 비디오 압축 기법의 표준으로 이 웃하는 이미지들간의 보상(compensation)을 이용하여 압축률을 크게 향상시킨 MPEG 압축 방법[5]이 주로 사용된다. 하지만 이와 같은 MPEG 압축 기법을 사용한다고 하더라도 압축된 비디오 데이터의 크기는 여전히 다른 미디어에 비하여 훨씬 크기 때문에 기존의 운영체제가 제공하는 단순한 파일 시스템 상에서 그 데이터를 관리하는 것은 비효율적이다. 더구나 하나의 MPEG 비디오 데이터가 아닌 많은 수의 MPEG 비디오 데이터를 저장 관리해야 하는 경우에는 더욱 더 비효율적이므로 새로운 데이터 관리 기법이 필요하다. 이러한 문제점에 대한 해결 방법의 하나로 효율적인 데이터 저장 관리 기능을 제공하는 데이터베이스 관리 시스템(DBMS)을 사용한다.

오늘날 가장 널리 사용되고 있는 관계형 데이터베이스 관리 시스템(RDBMS)은 숫자, 문자 등의 기본 데이터 타입에 대하여 효율적인 저장 관리 기법을 제공하지만, 멀티미디어 데이터와 같은 비정형적 타입의 데이터를 처리하기에는 부적합하다. 이와 같은 RDBMS의 단점을 극복하여 새로운 타입과 기능의 확장성이 뛰어난 객체지향 데이터베이스 관리 시스템(OODBMS)에 대한 연구가 활발히 이루어지고 있으며, 아울러 OODBMS를 이용한 멀티미디어 데이터의 저장 관리 연구 및 그것들을 기반으로 한 멀티미디어 응용 연구 또한 활발히 수행되고 있다[10].

이와 같은 OODBMS를 이용하여 MPEG 비디오 데이터를 효율적으로 저장 관리하고 사용자에게는 편리한 비디오 데이터 사용 환경을 제공하며 아울러 향후 확장이 용이한 시스템을 본 논문에서는 "OODBMS 기반 확장 용이 MPEG 비디오 브라우저"라 하여 크게 다음과 같은 네가지 연구 분야로 나누어 다룬다.

- OODBMS에서의 MPEG 비디오 데이터의 효율적 저장 관리 : 대용량의 MPEG 비디오 데이터를 저장 관리하기 위해서 OODBMS를 사용하는 것이 바람직하다. 이를 위하여 본 연구에서는 MPEG 비디오 데이터를 효율적으로 저장 관리하기 위한 OODBMS에서의 클래스 설계 및 구현에 대하여 다룬다.
- 클라이언트/서버 구조에서의 MPEG 비디오 데이터의 효율적 전송 : 저장된 데이터를 통신망을 통하여 효율적으로 전송하는 것이 바람직하다. 이를 위하여 본 연구에서는 클라이언트/서버 구조에서 MPEG 비디오 데이터를 효율적으로 전송하기 위한 기법들에 대하여 다룬다.

- 사용자 환경에서의 MPEG 비디오 데이터의 편리한 조작 : 사용자에게 쉬운 인터페이스와 다양한 기능들을 제공하는 것이 바람직하다. 이를 위하여 본 연구에서는 MPEG 비디오 브라우저를 위한 편리한 사용자 인터페이스에 대한 연구와 MPEG 비디오 데이터의 디코딩 및 여러가지 조작 기능들을 다룬다.
- 확장 용이 시스템 환경 . 향후 새로운 기능의 추가나 새로운 포맷의 데이터 지원이 쉬워야 한다. 이를 위해서 본 연구에서는 확장 용이 MPEG 비디오 브라우저의 객체 지향 설계 및 구현에 대하여 다룬다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 관련 연구들을 살펴보고 3장에서 본 논문에서 다룬 MPEG 비디오 브라우저의 전반적인 특성들에 대하여 살펴본다. 다음으로 4장에서는 OODBMS를 이용한 MPEG 비디오 브라우저의 설계에 대하여 소개하며, 그 실제 구현에 대하여 5장에서 소개한다. 6장에서는 구현된 시스템의 확장성에 대하여 간단히 살펴보고 마지막으로 7장에서는 결론으로서 본 연구 결과를 요약하고 향후 연구 계획에 대하여 소개한다

2. 관련 연구

본 연구는 본 연구실에서 개발한 OODBMS 기반의 주석 기반 비디오 검색 시스템인 VIRON[4]에서의 비디오 브라우저가 갖는 다음과 같은 문제점들을 해결하기 위해서 시작되었다.

- 비디오 브라우저의 독립성 결여 : VIRON에서의 비디오 브라우저는 검색 시스템의 일부분으로 포함된 형태로 내부적으로도 시스템의 다른 모듈들과 많은 연관성을 갖고 있다. 따라서 비디오 브라우저를 필요로 하는 다른 응용들에서는 재사용하기가 힘들다는 문제점을 갖는다. 이를 해결하기 위해 독립적이며 확장 가능한 비디오 브라우저의 개발이 필요하다
- OODBMS 활용성 미약 : VIRON에서는 OODBMS를 이용하여 비디오 데이터 및 관련 논리적 정보를 저장하고는 있지만 그 정보를 단순한 스트림 형태로 저장하기 때문에 실제 그 정보의 해석은 응용 시스템 차원에서 수행하는 비효율성이 존재한다. 이를 해결하기 위해 객체지향 데이터 모델을 이용하여 각 데이터 상태(값)와 관련 연산들을 함께 연관시킴으로써 보다 효율적인 시

스텝 구현이 가능하다.

일반적으로 비디오 브라우저는 단독으로 연구되는 것이 아니라 다른 연구의 한 부분으로 연구되는 것이 대부분이다. 본 연구에서도 비디오 브라우저 뿐만 아니라 OODBMS에서의 비디오 저장 관리, 클라이언트/서버간의 데이터 전송 문제, 그리고 비디오 브라우저의 확장성 문제를 함께 다루고 있는데, 관련 연구 분야로는 다중 사용자에게 여러 비디오 데이터를 동시에 서비스하기 위한 VOD(Video-On-Demand) 연구, 비디오 데이터의 효율적 코딩 및 디코딩을 위한 MPEG 데이터 연구, 그리고 비디오 데이터의 효율적 저장, 관리 및 표현을 위한 비디오 데이터 모델 연구를 들 수 있다.

먼저 VOD 서비스는 비디오 데이터를 저장 관리하는 서버로부터 사용자가 원하는 비디오를 검색하여 클라이언트에게 전송하여 사용자가 원하는 대로 조작할 수 있다는 점에서 본 연구와 유사하다. 하지만 대규모 상용 서비스를 목적으로 하는 VOD 서비스는 통신망을 통하여 많은 사용자에게 다양한 비디오를 원하는 방식대로 볼 수 있도록 하는 실시간 대화형 서비스로서 방대한 양의 MPEG 비디오들을 효율적으로 저장하고 모든 사용자 요구들을 실시간으로 처리하는 것이 필수적이며, 사용자가 선택한 비디오에 대한 데이터를 읽는 작업을 주로 수행하기 때문에 질의 최적화, 트랜잭션 처리, 권한 검사 등과 같은 DBMS가 제공하는 주요 기능들과는 별로 관련이 없다. 그 대신 VOD에서는 보다 빠른 데이터 처리를 위하여 RAID 기반의 화일 시스템을 주로 사용하며, 사용자의 요구를 즉시 처리할 수 있는 실시간성을 제공하는데 많은 연구가 수행되고 있다[6,11,12]. 따라서 VOD 연구 분야는 본 논문에서 다루고자 하는 DBMS 기반의 비디오 브라우저와는 다소 차이가 있다고 할 수 있다

다음 관련 연구 분야인 MPEG 데이터 연구에서는 MPEG 포맷 데이터를 생성하고 해석하는 MPEG 인코더/디코더[5], 반대 방향으로 효율적으로 재생하기 위한 역방향 MPEG 데이터 재생[7] 등의 연구가 수행되고 있다. 본 연구에서는 기존의 MPEG-1 비디오 화일에 대한 디코더를 이용하여 DBMS 내의 MPEG 비디오 데이터를 재생할 수 있도록 확장하였으며, 역방향 재생으로는 각 GOP(Group of Picture)마다 한 장면씩 보여주는 방식을 제공한다.

마지막으로 실세계의 비디오 정보를 컴퓨터 상의 데이터로 저장, 관리 및 표현하기 위한 구조와 연산들을 정의한 비디오 데이터 모델에 관련된 대표적인 연구들

로는 Oomoto와 Tanaka가 제안한 비디오 검색을 위한 객체지향 모델인 OVID[3], Hjelvold와 Midtstraum이 제안한 비디오 태이타 모델 및 확장된 질의어[8,9], Day, Dagtas 및 Iino가 제안한 비디오 데이터를 위한 객체지향 개념 모델[13], 그리고 김 기병 등이 제안한 다중 계층 비디오 모델인 MuVi[1] 등이 있다. 현재 본 연구에서는 이와 같은 구체적인 비디오 모델까지는 연구되지 않았고, 그 대신 OODBMS가 제공하는 객체지향 개념을 이용하여 비디오 정보를 크게 나누어 실제 비디오 데이터를 저장하는 물리적 비디오 객체와 비디오 각 프레임에 대한 정보를 저장하는 논리적 비디오 객체로 표현한다.

3. MPEG 비디오 브라우저 특성

이 장에서는 먼저 MPEG 비디오 데이터의 특성을 소개한 후에 그러한 특성을 효율적으로 처리하기 위해 OODBMS에 기반한 MPEG 비디오 브라우저가 갖춰야 할 특성들을 각 연구 분야별로 나누어 살펴본다.

3.1 MPEG 데이터의 특성

MPEG 데이터는 물리적으로는 비트 스트림 형태로 인코딩되어 있으며, 논리적으로는 그림 1과 같이 한 장면을 나타내는 프레임들의 연속으로 구성되는데, 각 프레임은 인코딩 방법에 따라 크게 I, P, 그리고 B 프레임 타입으로 나뉘어진다. I 프레임은 JPEG과 유사한 방법에 의해 독립적으로 코딩된 타입이며, P 프레임은 이전의 I 또는 P 프레임으로부터 모션 보상 기법을 이용한 전방향(forward) 예측 코딩되었으며, B 프레임은 이전의 I 또는 P-프레임과 이후의 I 또는 P-프레임 양측으로부터 보간 기법을 이용한 양방향 예측 코딩된 타입이다. 따라서 I 프레임은 단독으로 디코딩되어 재생될 수 있는 반면에 P 프레임과 B 프레임은 관련된 다른 프레임들과 함께 디코딩되어야 하는 의존 관계를 갖는다.

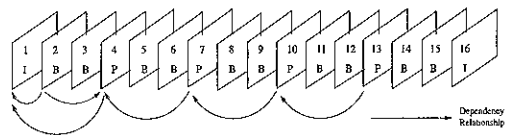


그림 1 MPEG 프레임들의 의존 관계

이와 같이 MPEG 데이터는 서로 의존 관계를 갖는 I, P, B-프레임들로 구성되기 때문에 임의 프레임 재생이나 역방향 재생을 위해서는 하나의 MPEG 데이터를 단순히 하나의 연속된 데이터로 처리하는 것과 함께 MPEG 데이터의 특성에 따라 여러개의 논리적인 단위

들로 나누어 처리하는 것이 필요하다.

3.2 MPEG 데이터 관리를 위한 OODBMS 특성

앞에서 살펴본 것과 같은 특성들을 갖는 MPEG 데이터의 효율적 관리를 지원하기 위한 OODBMS의 대표적인 특성들은 다음과 같다.

- 사용자 정의 타입 지원 : 사용자가 원하는 임의의 타입과 관련된 각 연산을 클래스로 정의할 수 있다. 따라서 MPEG 데이터에 대한 클래스를 정의함으로써 DBMS 차원에서 MPEG 데이터의 물리적 특성뿐만 아니라 논리적 특성까지 효율적으로 표현할 수 있다.
- BLOB(Binary Large Object) 지원 : 대용량의 데이터를 비트 스트림으로 저장하기 위한 BLOB을 지원하고, 그것들을 효율적으로 처리하기 위한 insert, delete, update 등의 인터페이스를 지원한다. 이러한 BLOB을 기반으로 MPEG 데이터 타입을 정의할 수 있다
- 객체 관리 기능 : 사용자가 정의한 클래스에 대한 인스턴스로서 생성된 객체 단위로 데이터를 저장 관리한다. 객체는 실제 데이터의 상태를 포함하며, 해당 클래스에 정의된 멤버함수들을 사용하여 객체의 상태를 효율적으로 관리할 수 있다. 따라서, 각 MPEG 데이터는 사용자가 정의한 MPEG 클래스의 객체로 생성되어 효율적으로 저장 관리될 수 있다.

3.3 MPEG 데이터 전송을 위한 클라이언트/서버 특성

대용량의 MPEG 데이터를 여러 사용자가 효율적으로 이용할 수 있도록 다음과 같은 특성들을 갖는 클라이언트/서버 구조가 바람직하다.

- 필요한 데이터의 부분 전송 : 사용자의 대기 시간을 최소화하기 위하여 전송할 데이터의 크기를 줄이는 것이 바람직하다. 사용할 데이터를 한번에 모두 전송하기보다는 여러 부분으로 나누어서 필요할 때 하나씩 전송하는 것이다.
- 효율적 캐쉬 관리 : 서버는 클라이언트로부터의 데이터 요청을 받으면 일단 그 데이터가 캐쉬에 있는지 조사하여 없을 때에만 디스크로부터 읽어 들여 먼저 캐쉬에 저장한 다음에 클라이언트에게 전송한다. 이와 같이 최근에 디스크로부터 읽어 들인 데이터를 캐쉬에 저장 관리함으로써 다음에 그 데이터가 필요한 경우에 디스크로부터 다시 읽어 들일 필요가 없어진다. 클라이언트 시스템에서도 전송받은 데이터를 캐쉬에 저장 관리함으로써 같은 효과를 얻을 수 있다.

- 데이터의 지역(local)저장 : 사용자가 자주 사용하는 데이터라면 서버로부터 그 데이터를 매번 전송 받을 것이 아니라, 일단 한 번 전송받은 후에 그 데이터를 클라이언트가 직접 저장 관리하면서 필요할 때 바로 사용하는 것이 보다 효율적이다.

3.4 MPEG 데이터 조작을 위한 사용자 환경 특성

사용자가 원하는 MPEG 비디오 데이터를 편리하게 이용하기 위해서는 다음과 같은 특징들이 바람직하다.

- 다양한 조작 기능 : MPEG 데이터에 대한 기본적인 재생 및 정지 기능뿐만 아니라 빠른 재생, 역방향 재생, 한 장면씩 보기, 임의 장면 보기 등 여러가지 다양한 조작 기능들을 제공하여 사용자가 원하는 장면을 쉽게 볼 수 있도록 한다
- 편리한 사용자 인터페이스 : 사용자에게 친숙한 TV, VTR과 유사한 그래픽 사용자 인터페이스(GUI)를 제공하여 편리하게 사용할 수 있도록 한다. 또한 VTR에서는 제공하지 않는 기능들, 예를 들어 임의 장면 보기, 비디오 선택 등을 위해서 사용자가 직관적으로 알 수 있는 인터페이스를 제공한다.
- 데이터 입력 및 저장 기능 : DB 구축자 또는 특정 사용자에게 자신이 보유하고 있는 MPEG 데이터를 DB로 등록할 수 있도록 하여 보다 많은 MPEG 데이터를 제공한다. 또한 서버로부터 전송받은 MPEG 데이터를 사용자 화일로 저장하는 기능을 제공하여 보다 쉽게 MPEG 데이터를 이용할 수 있도록 한다.

3.5 시스템의 확장 용이성

시스템의 유지 보수의 편의성 및 활용성을 높이기 위해서 다음과 같은 확장성들이 바람직하다.

- 새로운 기능의 추가 : 새로운 비디오 조작 기능이나 새로운 비디오 검색 기법 등의 추가가 용이하도록 한다.
- 새로운 포맷의 지원 : MPEG 비디오 포맷뿐만 아니라 향후에 MOV, AVI 등의 다른 포맷의 비디오를 지원하도록 확장 가능하도록 한다.
- 확장 용이한 응용 개발 환경 : 다른 응용 시스템들을 개발할 때 기존의 시스템을 이용하거나 그 시스템을 구성하는 클래스들을 재사용할 수 있도록 한다.

4. SOP MPEG 비디오 브라우저 설계

본 장에서는 3장에서 살펴본 특성들을 고려하여 OODBMS에 기반한 MPEG 비디오 브라우저를 설계한

다. 먼저 기반 OODBMS로 사용될 SOP에 대하여 소개하고, SOP 상에서의 MPEG 데이터 타입 지원을 위한 클래스 설계를 다룬다. 다음으로 SOP에 기반한 클라이언트/서버 구조의 MPEG 비디오 브라우저를 설계하고, 마지막으로 사용자 인터페이스 설계를 다룬다.

4.1 SOP OODBMS 소개

본 연구실에서 개발한 SOP(SNU OODBMS Platform)는 ODMG-93 표준을 따르는 국내 최초의 다중 사용자용 OODBMS로서, ODL(Object Definition Language) 및 OQL(Object Query Language)을 지원하며 지속성 객체 지향 응용 프로그램의 작성을 위해 C++, Smalltalk 및 CLOS 바인딩을 제공한다. 특히 SOP는 90년대 소프트웨어 개발의 가장 중요한 개념으로 자리잡은 객체 지향 개념을 기반으로 설계, 구현되어 관리 및 유지 보수는 물론 새로운 기능의 추가가 매우 용이하다.

SOP는 효율적인 객체의 저장 및 관리 기능을 제공하는 객체 지향 저장 시스템(Soprano)[2], 스키마 관리 및 진화를 담당하는 스키마 관리자, 사용자 질의를 처리하는 OQL 처리기, 그리고 객체 지향 데이터베이스 언어 전위처리기 등의 엔진 모듈과 그래픽 환경에서의 스키마 정의 및 질의 처리 등을 지원하기 위한 다양한 사용자 지원 도구들로 구성된다. 또한 SOP에는 2,3차원의 공간데이터, 이미지, 음성, 동화상 등의 멀티미디어 데이터에 대한 프로토타입 질의 처리기와 비정형적인 텍스트의 정보 검색 도구도 제공되어 멀티미디어 데이터베이스 구축이 가능하다.

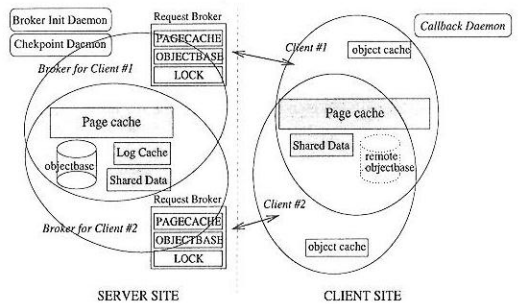


그림 2 Soprano의 클라이언트-서버 구조

이러한 SOP의 구성 모듈들 중에서도 특히 객체의 저장, 관리, 트랜잭션의 병행 제어, 복구, 캐싱 등의 핵심적인 작업을 수행하는 Soprano는 여러 다양한 응용을 지원하기 위해 설계된 클라이언트/서버 구조의 객체 저장 시스템으로서 크게 물리적 장치를 다루는 부분과 객

체 캐쉬 관리 부분, 그리고 다양한 지속성 객체 클래스들로 구성되어 있다. Soprano는 그림 2에서 보듯이 각 클라이언트마다 하나의 서버 프로세스가 연결되어 서비스를 제공하는 다중 프로세스 구조를 갖는다.

먼저 서버 프로세스는 페이지 캐쉬의 요구를 수행하는 페이지 캐쉬 요구 중개자(request broker), 디스크 연산을 대신하는 ObjectBase 요구 중개자, 로크(lock) 요구를 처리해주는 로크 요구 중개자 등 여러 요구 중개자들로 구성된다. 즉, 서버 프로세스는 클라이언트 프로세스에게 페이지의 할당 및 반환, 읽기, 쓰기, 로크의 획득 및 반환 등의 연산을 대신 처리하고 그 결과를 전달한다.

클라이언트 프로세스는 초기화시 Broker Init Daemon에 의해 생성된 요구 중개자와 연결하여 클라이언트에서 수행할 수 없는 모든 서비스를 요구 중개자를 통해 수행한다. 클라이언트 프로세스는 요구한 객체를 서버 프로세스로부터 페이지 단위로 전송받아 각각 객체 캐쉬와 페이지 캐쉬에 저장 관리하면서, 그 객체에 대해 가능한 모든 작업을 수행한다.

4.2 SOP에서의 MPEG 클래스 설계

본 절에서는 SOP에서의 MPEG 비디오 데이터를 처리하기 위해서 객체지향 데이터 모델을 이용한 MPEG 클래스 설계에 대하여 설명한다.

SOP의 Soprano에서 다루는 객체의 크기는 기본적으로 페이지 크기로 제한되어 있다. 하지만 MPEG 데이터와 같은 가변적이며 비정형적인 대용량의 멀티미디어 데이터를 처리하기 위해서는 BLOB구조에 기반을 둔 타입의 지원이 필수적이다. Soprano에서는 BLOB에 기반을 둔 LargeObj 클래스를 제공하는데, LargeObj 객체는 그 크기에 제한이 없으며 임의의 위치에서 데이터의 삽입이나 삭제가 가능하다.

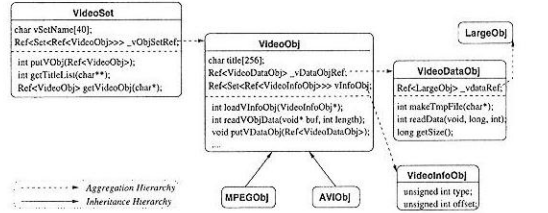


그림 3 MPEG 클래스 스키마

본 연구에서는 그림 3과 같은 구조의 MPEG 클래스 스키마를 설계하였다. 물리적 비디오 데이터는 LargeObj 객체를 포함하는 VideoDataObj 객체에 저장되어

현재 지정된 위치에서부터 원하는 크기만큼의 데이터를 읽는 readData() 멤버함수 등을 갖는다. 비디오 데이터는 여러개의 논리적인 부분으로 나뉘어져 그 각각은 VideoInfoObj 객체에 저장되는데, MPEG 데이터에 대한 논리적인 비디오 분할의 기준으로는 몇 개의 I 프레임들 구간으로 구성된 GOP 단위나 각 프레임 단위가 있는데, 본 연구에서는 각 프레임 단위를 사용한다.

이러한 물리적 데이터와 논리적 데이터들을 효과적으로 관리하는 것이 바로 VideoObj 클래스로서, VideoDataObj 객체와 VideoInfoObj 객체를 집합에 대한 포인터를 포함하며 그것들을 관리하기 위한 여러 멤버 함수들을 갖는다. 이 때, VideoObj 클래스는 추상(abstract) 클래스로 정의되어 실제 객체를 만들지는 않고 하위 클래스들을 위한 정보를 제공하는 상위 클래스로서 의미를 갖는데, 이 클래스의 하위 클래스로 MPEG 포맷 비디오를 위한 MPEGObj 클래스, MOV 포맷 비디오를 위한 MOVObj 클래스 등 다양한 비디오 포맷의 클래스들을 정의할 수 있다.

이제까지 하나의 비디오 데이터를 위한 클래스들을 살펴보았는데, DBMS에 저장 관리될 많은 비디오 데이터를 효율적으로 관리하기 위한 관리자가 필요하다. 이러한 관리자 클래스가 바로 VideoSet 클래스로서, 이 클래스는 새로운 비디오를 추가하거나 기존의 비디오를 삭제하는 기능, 비디오들의 목록을 순차적으로 살펴보고 선택하는 기능 등을 멤버 함수로 제공한다.

4.3 클라이언트/서버 구조의 MPEG 비디오 브라우저 설계

본 절에서는 앞에서 설명한 클라이언트/서버 구조의 SOP OODBS와 MPEG 관련 클래스들을 기반으로 각 모듈별로 설계한 MPEG 비디오 브라우저에 대하여 소개한다(그림 4 참조).

먼저 MPEG 비디오 데이터의 저장 및 전송을 담당하는 MPEG 비디오 브라우저의 서버 시스템으로 앞에서 설명한 SOP 서버를 그대로 이용한다. 이는 SOP 서버가 앞에서 살펴본 비디오 브라우저 서버로서의 특정한 비디오 데이터의 효율적 저장 관리 및 클라이언트가 요구하는 비디오 데이터의 효율적 전송 기능을 제공하기 때문이다.

다음으로 MPEG 비디오에 대한 검색 및 비디오 재생을 처리하는 MPEG 비디오 브라우저의 클라이언트 시스템은 다음과 같은 6개 모듈들로 구성되어 있다.

- SOP 클라이언트 모듈 : 서버와의 통신을 담당하며 전송받은 데이터를 캐쉬에 저장 관리한다.
- MPEG 디코더 모듈 : MPEG 포맷의 비디오 데

이타를 해석하여 사용자에게 다양하게 보여주는 기능을 담당한다.

- MPEG DB 구축자(populator) 모듈 : 메모리 상의 MPEG 데이터를 MPEG 관련 객체로 만들어 SOP 데이터베이스로 구축하는 일을 담당한다.
- MPEG 객체 관리자 모듈 : MPEG 관련 객체를 여러 부분으로 나누어 전송받아서 메모리 상에서 하나의 비디오 정보로 관리한다.
- MPEG 화일 관리자 모듈 : 사용자의 디스크에 있는 MPEG 화일을 메모리로 읽어들이어 DB 구축을 위해 MPEG DB 구축자에게 넘겨주는 작업과 MPEG 비디오 조작을 위해 MPEG 디코더로 넘겨주는 작업, 그리고 MPEG 객체 관리자가 SOP로부터 읽어들이는 MPEG 관련 객체를 넘겨받아 사용자 디스크에 지역 화일로 저장하는 작업을 담당한다.
- 사용자 인터페이스 모듈 : DB 또는 화일 시스템으로부터 원하는 비디오 데이터를 선택하고, 사용자 입력에 따른 그 처리 결과를 보여준다.

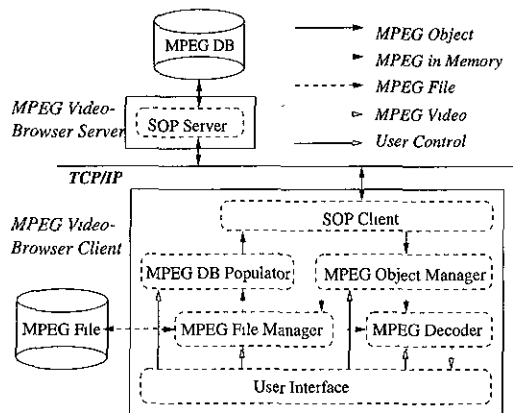


그림 4 MPEG 비디오 브라우저의 전체 구조

이와 같이 SOP에 기반한 클라이언트/서버 시스템은 SOP가 제공하는 효율적인 캐쉬 기능을 이용하고, MPEG 객체 관리자와 MPEG 화일 관리자에서 MPEG 데이터의 부분 전송과 지역 저장을 각각 담당하도록 하였으며, 시스템이 각 모듈별로 설계되어 구현 및 유지, 보수가 용이하다는 장점을 갖는다.

4.4 MPEG 비디오 브라우저의 사용자 인터페이스 설계

MPEG 비디오 브라우저의 사용자 인터페이스는 UNIX환경에서 널리 사용되는 GUI인 X윈도우를 기반으로 그림 5와 같이 사용자에게 친숙한 TV, VTR과 유

사하고 다양한 기능들을 제공하도록 설계되었는데, 크게 다음과 같은 네부분으로 나뉘어진다.



그림 5 사용자 인터페이스 설계

- 메뉴 영역 : 사용자가 선택가능한 기능들을 위한 풀-다운(pull-down) 방식의 메뉴들을 제공한다. 그 기능들로는 비디오 선택 및 재생, 정지, 빨리보기 등 기본적인 MPEG 비디오 조작에 관련된 기능들과 DB 구축, 화일 저장, 도움말 기능, 그리고 시스템의 종료 등이 있다. 사용자는 이러한 메뉴들을 마우스 또는 키보드로 선택할 수 있다.
- 비디오 디스플레이 영역 : TV의 브라운관과 같이 실제 MPEG 비디오를 보여주는 영역이다. 만일 MPEG 비디오가 기본으로 설정된 이 영역의 크기보다 클 경우에는 자동으로 필요한만큼 윈도우의 크기가 확장될 것이다.
- 스케일 영역 : 스케일은 기다란 가로 막대 안을 움직일 수 있는 조그마한 사각형인 슬라이더 (slider)를 포함한 형태로 슬라이더는 전체 비디오 프레임들 중에서 현재 프레임 위치를 나타내며 그 프레임 번호를 슬라이더 위에 보여준다. 사용자는 슬라이더를 임의의 위치로 옮김으로써 원하는 비디오 프레임을 바로 볼 수 있다.
- 버튼 영역 : 가장 대표적인 기능들을 사용자가 바로 사용할 수 있도록 버튼으로 제공한다. 먼저 버튼 영역(1)에서는 비디오 선택, 재생, 정지 등의 MPEG 비디오 조작 기능들을 원격 비디오 선택, 재생, 정지, 이전 프레임 보기, 다음 프레임 보기, 빨리보기, 그리고 비디오 초기화 기능을 위한 버튼들을 포함한다. 여기서 각 버튼들은 VTR의 버튼들에서 볼 수 있는 기호들을 이미지로 포함하여 사용자가 쉽게 그 기능을 알 수 있도록 한다. 다

음으로 버튼 영역(2)에서는 사용자 디스크의 MPEG 화일 선택, 선택된 MPEG 화일의 DB 구축 기능 등의 기타 주요 기능들을 위한 버튼들을 포함한다.

5. SOP MPEG 비디오 브라우저 구현 및 실행 예

본 장에서는 4장에서 살펴본 MPEG 비디오 브라우저의 설계를 바탕으로 구현한 실제 시스템에 대하여 설명하고, 그 실행 예를 살펴본다.

5.1 시스템 구현

MPEG 비디오 브라우저의 서버 시스템은 이미 언급했듯이 기존의 SOP 서버 시스템을 기반으로 앞에서 설계한 MPEG 클래스들을 정의하여 MPEG 데이터를 지속성 객체(persistent object)로 저장 관리할 수 있도록 하였다. 이 때, MPEG 데이터의 논리적인 정보를 위한 VideoInfoObj 클래스는 각 프레임의 타입과 위치를 저장하도록 정의되었다.

MPEG 비디오 브라우저의 클라이언트 시스템의 개발 환경으로 C/C++과 X윈도우/Motif를 이용하여 내부 기능들과 외부 GUI를 각각 구현하였다.

각 모듈을 간단히 살펴보면 먼저 서버와 관련된 모든 작업들을 담당하기 위해 SOP 클라이언트를 이용하며, MPEG 디코더 모듈은 UC. Berkeley에서 개발한 MPEG-1 디코더 라이브러리 버전 2.2를 기반으로 본 연구 특성에 맞게 변경하였다. 이 라이브러리는 C언어 및 X 윈도우용 프로그램으로 개발되어 본 연구에서 개발한 C++ 및 Motif용 프로그램과의 통합 및 SOP와의 통합 작업이 필요했으며, 이전/다음 프레임 보기, 임의 접근 등의 새로운 기능들을 추가하였다. 그리고, MPEG 화일 관리자, MPEG 객체 관리자, MPEG DB 구축자, 그리고 사용자 인터페이스 모듈들은 모두 4장에서의 설계 특성들을 만족하도록 구현되었다.

그림 6에서는 이와 같이 구현된 클라이언트/서버 시스템의 작동 과정을 보여준다.

구현된 시스템은 먼저 클라이언트의 연결 요청에 대한 서버의 확인이 이루어진 이후에 실제 연결이 설정되면 서버가 관리하는 모든 비디오에 대한 제목, 객체 식별자와 같은 정보를 클라이언트에게 전송함으로써 시스템의 초기화가 완료된다.

다음으로 사용자는 클라이언트가 보여주는 비디오들의 목록 중에서 원하는 것을 선택하면 클라이언트는 서버에게 데이터 요청을 하고, 그 결과로 서버는 해당 비디오 데이터를 검색하여 일단 일정 크기의 시작 부분만

을 전송한다. 클라이언트는 비디오의 첫장면을 보여주고 사용자의 입력을 기다리며, 사용자의 입력에 따라 새로운 비디오 부분이 필요할 때 서버에 요청한 후에 그 데이터를 전송받아 사용한다.

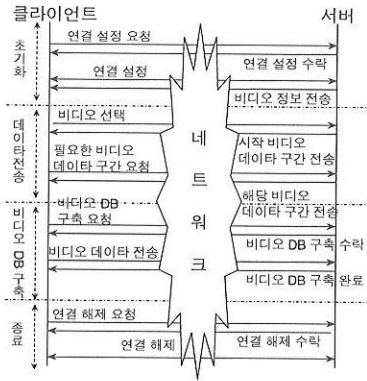


그림 6 클라이언트/서버 시스템의 작동 과정

비디오 데이터베이스 구축이 목적인 사용자를 위해 클라이언트는 먼저 서버로부터 DB 구축 허락을 받은 후에 자신의 디스크에 저장된 MPEG 화일을 MPEG 관련 객체로 생성하고 서버에게 전송하여 서버가 비디오 데이터를 저장 관리하도록 한다.

시스템의 사용을 끝내려면 클라이언트는 서버에게 연결 해제 요청을 해서 허락을 받으면 할당받은 시스템 자원을 반환한 후에 연결을 해제하게 된다.

5.2 시스템 실행 예

MPEG 비디오 브라우저의 서버 시스템은 SOP 서버를 실행시킴으로써 작동되며, 그 이후에 사용자는 MPEG 비디오 브라우저의 클라이언트를 실행시킴으로써 이용가능하다.

MPEG 비디오 브라우저를 이용한 MPEG 비디오 조작에 앞서 다음과 같은 과정으로 MPEG 비디오 데이터베이스가 구축되어야 한다. 먼저 사용자가 MPEG 화일 선택 버튼을 눌러 원하는 MPEG 화일을 선택한 후에 그 아래 버튼을 누르면 비디오 제목 입력 대화 상자가 나타난다. 그 상자에 사용자가 원하는 제목을 입력한 후에 OK 버튼을 누르면 시스템이 이미 같은 제목의 비디오 데이터가 있는지 검사하여 만일 없는 경우라면 관련 객체들을 생성하여 DB로 구축하게 된다.

이와 같은 DB 구축 예를 그림 7에서 보여주고 있는데, 현재 사용자 디스크의 "/tmp/C++_2.mpg"라는 MPEG 파일을 선택한 후에 "C++ 두번째 강의" 제목으

로 SOP 상의 MPEG DB로 구축하는 과정이다. 이 때, 선택한 MPEG 화일에 대하여 사용자는 재생, 멈춤, 초기화 기능과 같은 기본적인 조작을 수행할 수도 있다. 사용자는 MPEG DB에 구축된 비디오의 제목 리스트를 보고 원하는 비디오를 선택한 후에 다양한 조작을 할 수 있다.

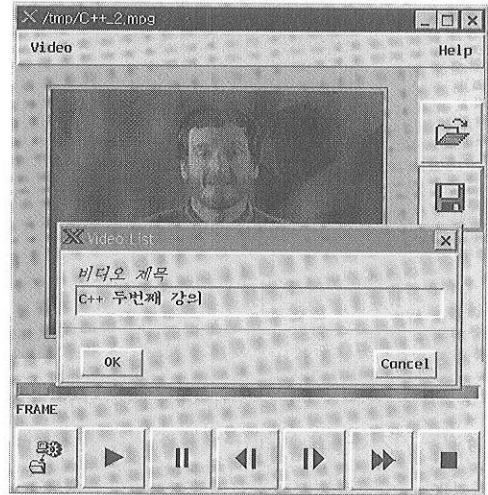


그림 7 MPEG 화일의 DB 구축 예

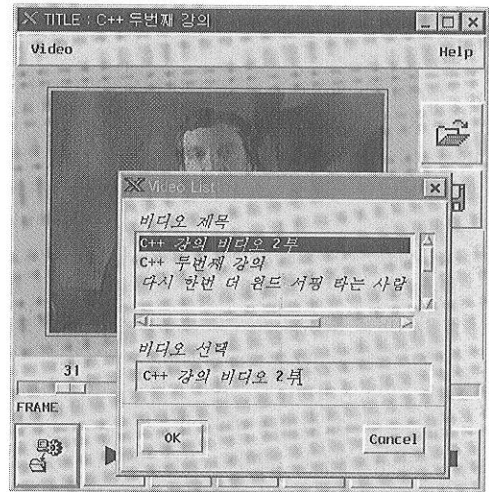


그림 8 MPEG 화일의 선택 및 재생 예

그림 8에서는 "C++ 두번째 강의"란 제목의 비디오 재생 과정을 보여주는데, 현재 스케일 영역에서 보듯이 31번째 프레임을 재생 중이다. 또한, 새로운 비디오를

선택하는 과정도 함께 보여주는데, 원격 비디오 선택 버튼을 눌러 나타난 비디오 목록 중에서 “C++ 강의 비디오 2부”란 제목의 비디오를 선택한 것이다.

6. MPEG 비디오 브라우저의 확장성

이 장에서는 구현된 MPEG 비디오 브라우저의 확장성에 대하여 설명한다.

먼저 새로운 기능의 추가가 용이하다. 예를 들어 2배속 재생 기능을 추가하려면 외부에서 호출할 수 있도록 VideoObj 클래스 내에 play2speed() 멤버함수를 정의하고 MPEGObj 클래스에 실제 함수를 구현한 다음 사용자 인터페이스에 필요한 버튼을 추가하여 그 버튼이 눌렸을 때 현재 사용되는 MPEGObj 객체의 play2speed() 멤버함수가 수행되도록 연결시키면 된다.

다음으로 새로운 비디오 타입으로의 확장이 용이하다. 시스템의 각 모듈은 MPEG 데이터를 처리하기 위해서 VideoObj 클래스의 멤버함수를 호출하게 되어 있다. 따라서 MPEG 비디오 타입을 위해 추상 클래스인 VideoObj 클래스의 하위 클래스로 MPEGObj 클래스를 정의한 것과 같이 새로운 비디오 타입을 위해서도 VideoObj의 하위 클래스로 정의하여 필요한 멤버 함수들을 정의하면 각 모듈에서는 기존의 방식으로 새로운 포맷의 비디오를 처리할 수 있다.

마지막으로 본 연구에서 정의한 MPEG 관련 클래스를 재사용하거나 확장하여 새로운 MPEG 관련 응용을 개발할 수 있으며, MPEG 비디오 브라우저 자체로도 비디오 검색 시스템과 같은 응용 시스템에 필요한 비디오 재생기로 사용될 수도 있다.

7. 결론 및 향후 연구

본 연구에서는 OODBMS를 이용한 대용량의 MPEG 비디오 데이터의 효율적 저장 관리 및 편리한 사용자 환경을 제공하며 향후 새로운 기능의 추가나 새로운 포맷의 비디오의 지원이 용이한 “확장 용이 MPEG 비디오 브라우저”를 목표로 그것이 갖추어야 할 특성들을 분석하고, 그에 따른 설계 및 구현에 대하여 다루었다.

먼저 MPEG 비디오 브라우저가 갖추어야 할 바람직한 특성들을 OODBMS내에서의 MPEG 비디오 데이터의 효율적 관리 분야, 클라이언트/서버 구조에서의 MPEG 비디오 데이터의 효율적 전송 분야, 사용자 환경에서의 MPEG 비디오 데이터의 편리한 조작 분야, 그리고 확장 용이 시스템 환경 분야의 네분야로 나누어 제시하였다.

제안된 특성들을 고려하여 본 연구실에서 개발 중인

SOP를 기반 OODBMS로 하여 클라이언트/서버 구조의 MPEG 비디오 브라우저를 설계하고 구현하였다. 먼저 우수한 데이터 타입 확장성을 갖는 SOP에 MPEG 관련 클래스들을 정의하여 대용량의 MPEG 데이터를 효율적으로 저장 관리하도록 하였다. 그러한 MPEG 데이터베이스를 효율적으로 접근하기 위해 SOP에 기반한 클라이언트/서버 구조를 갖는 시스템을 구현하고, 아울러 SOP에서 캐시를 이용한 데이터 관리 기능과 MPEG 데이터를 논리적 단위로 나누어 관리하면서 필요한 부분만을 선택적으로 전송하도록 하여 대용량의 MPEG 비디오 데이터의 효율적인 전송을 가능하게 하였다. 또한 사용자들이 시스템을 쉽게 사용할 수 있도록 X윈도우/Motif에 기반한 TV 모양의 GUI를 제공하며 주요 기능들을 VTR의 버튼 형태의 이미지 버튼들로 제공하였다. 마지막으로 MPEG 비디오 브라우저는 객체 지향 개념과 C++을 이용하여 기본 데이터 타입을 클래스들로 정의하고 그 클래스들을 기반으로 각 기능들을 모듈들로 구현하여 유지 보수 및 확장이 용이하다는 특징을 갖는다

향후 연구 계획으로 현재 MPEG-1 비디오만을 지원하는데, MPEG-2 레벨까지 지원하도록 확장할 예정이며, 제목과 같은 단순한 정보를 이용하여 원하는 비디오를 선택하는 현재의 수준에서 벗어나 실제 비디오의 내용에 의한 검색이 가능한 비디오 검색 시스템으로 확장할 계획이다. 그리고, SOP 서버를 그대로 사용하는 현재의 MPEG 비디오 브라우저의 서버 시스템을 여러 사용자의 이용 정보들을 효과적으로 처리할 수 있는 지능적인 서버 시스템으로 확장하는 연구가 필요하며, 구현된 시스템을 이용하여 다양한 응용 시스템들을 개발할 예정이다.

감사의 글

본 연구를 함께 수행하며 본 논문 작성에 많은 도움을 준 Multimedia팀원들과 SOP팀원들께 깊이 감사드립니다.

참고 문헌

- [1] 김기병, 김형주, “내용 기반 검색 및 주석 기반 검색을 통합하는 비디오 데이터 모델의 설계 및 구현”, 한국정보과학회 학회지, 제3권, 제2호, pp.115-126, 1997.
- [2] 안정호, 이강우, 송하주, 김형주, “Soprano:객체 저장 시스템의 설계 및 구현”, 한국정보과학회학회지, 제2권, 제3호, 1996.
- [3] Eitetsu Oomoto and Katsumi Tanaka, “OVID: Design and Implementation of a Video-Object

- Database System," IEEE Trans. on Knowledge and Data Engineering, Vol.5, Num.4, pp.629-643, 1993,
- [4] Ki-Wook Kim, Ki-Byoung Kim, and Hyoung-Joo Kim, "VIRON: An Annotation-Based Video Information Retrieval System," In the proceedings of the 20th COMSAC, 1996.
- [5] L.A.Rowe, K.D.Patel, B.C.Smith and K.Liu, "MPEG Video in Software: Representation, Transmission, and Playback", IS&T/SPIE symposium on Elec. Imaging Science and Technology: San Jose, CA, pp.1-11, 1994.
- [6] Mourad AN, "Issues in the Design of a Storage Sever for Video-on-Demand," ACM Multimedia Systems, Vol.4, Num.2, pp.70-86, 1996.
- [7] M.S. Chen, D.D. Kandlur, "Stream Conversion to Support Interactive Video Playout," IEEE Multimedia, Vol.3, Num.2, 1996.
- [8] Rune Hjelsvold, "Video Information Contents and Architecture," The 4th International Conference on Extending Database Technology, March 1994.
- [9] Rune Hjelsvold and Roger Midtstraum, "Modelling and Querying Video Data," In Proceedings of 20th International Conference on VLDB, pp.686-694, Santiago, Chile, October 1994.
- [10] S. Khoshafian, A.B. Baker, "Multimedia and Imaging Databases," Morgan Kaufmann Publishers, 1996.
- [11] T.D.C. Little and D. Venkatesh, "Prospects for Interactive Video-on-Demand," IEEE Multimedia, Vol.1, Num.3, pp.14-24, 1994.
- [12] Y.N.Doganata and A.N. Tantawi, "Making a Cost-Effective Video Server," IEEE Multimedia, Vol.1, Num.4, pp.22-30, 1994.
- [13] Young Francis Day, et al., "Object-Oriented Conceptual Modeling of Video Data," Proceedings of ICDE, pp.401-408, 1995.



송 용 준

1992년 2월 서울대학교 컴퓨터공학과 졸업. 1994년 2월 서울대학교 컴퓨터공학과 석사. 1994년 3월 ~ 1996년 2월 한국통신 멀티미디어 연구소 근무. 1996년 3월 ~ 현재 서울대학교 컴퓨터공학과 박사과정 재학중. 관심분야는 객체지향 시스템, 멀티미디어 시스템, 데이터베이스

김 형 주

제 4 권 제 3 호(C) 참조