

컨텐츠 저작 응용을 위한 디지털 자산 관리자의 설계 및 구현

(Design and Implementation of a Digital Asset Manager for
Contents Authoring Applications)

김종수[†] 방수호[†] 정연돈[†] 이재형[†] 김민정[†]
(Jong Soo Kim) (Su Ho Bang) (Yon Dohn Chung) (Jaehyung Lee) (Min Jung Kim)

김명호^{**} 장덕호^{***} 박종승^{***} 오황석^{***}
(Myoung Ho Kim) (Duk Ho Chang) (Jong Seung Park) (Hwang Seok Oh)

요약 디지털 자산이란 이미지, 오디오, 비디오 등과 같이 디지털화된 형태로 존재하는 멀티미디어 정보를 의미한다. 디지털 자산은 저장에 요구되는 방대한 저장 공간 및 자산의 내용 표현에 요구되는 다차원적 정보 특성과 같은 특징들로 인하여, 효율적인 관리를 위한 많은 노력이 필요하다. 본 논문에서는 디지털 자산을 효과적으로 관리하기 위한 디지털 자산 관리자를 구현하였다. 개발된 디지털 자산 관리자는 디지털 자산 중 가장 복잡도가 높은 비디오 자산을 주요 관리 대상으로 하며, 다양한 디지털 자산 관련 응용 환경 중 디지털 컨텐츠 저작을 주요 응용으로 가정한다. 디지털 자산 관리자는 비디오 자산에 대한 효과적인 접근을 위해 계층 구조 모델에 기반을 둔 자산 관리 기능을 제공하고, 저장 시스템과의 독립성을 최대한 보장하며, 빠른 내용 기반 유사성 검색 기능을 지원한다.

Abstract Digital assets denote multimedia information that exists in the form of digitized materials such as images, audio, and video. The management of digital assets demands much effort because of a huge amount of storage space requirement and multidimensional characteristics of the information needed to describe their contents. In this paper, we design and implement a Digital Asset Manager that stores and manages digital assets efficiently. Among the various types of digital assets, we focus on the video asset which has the highest complexity. Our Digital Asset Manager provides various facilities for digital contents authoring applications. In the Digital Asset Manager, video assets are managed by using a hierarchical model in order to ensure efficient accesses to any part of a video asset. Our system also guarantees the independence from the storage platform, and provides a fast content-based similarity search method on the digital assets.

1. 서론

디지털 자산(Digital Asset)이란 디지털화된 형태로 존재하는 멀티미디어 정보를 의미한다. 최근 들어 멀티미디어 저작 등과 같은 컨텐츠 저작 및 네트워크를 이용한 컨텐츠의 제공 등과 같은 응용[1]이 보편화 되면서, 디지털 자산의 정보 가치 및 활용의 중요성이 날로 증대되고 있다. 그러나 디지털 자산은 저장에 요구되는 방대한 저장 공간 및 자산의 내용 기술에 요구되는 다차원적 정보 특성과 같은 특징들로 인하여, 효율적인 관리를 위한 많은 노력이 필요하다. 효과적인 디지털 자산의 관리를

[†] 비회원 : 한국과학기술원 전산학과
jskim@dbserver.kaist.ac.kr
shbang@dbserver.kaist.ac.kr
ydchung@dbserver.kaist.ac.kr
jlee@dbserver.kaist.ac.kr

^{**} 중신회원 : 한국과학기술원 전산학과 교수
mhkim@dbserver.kaist.ac.kr

^{***} 비회원 : 한국전자통신연구원 가상현실연구센터 연구원
dhchang@etri.re.kr
park@etri.re.kr
hsoh@etri.re.kr

논문접수 : 1999년 12월 28일
심사완료 : 2000년 5월 1일

위해서는 방대한 양의 전체 자산 중 내용적 특성에 따라 적합한 자산을 검색하고, 이를 방영 혹은 수정하거나 이를 바탕으로 새로운 자산을 생성하는 작업들이 쉽고 빠르게 수행될 수 있도록 지원되어야 한다. 즉, 디지털 자산을 관리하는 시스템에서는 디지털 자산의 효율적인 저장 방법 및 이에 대한 효과적인 검색 기법의 지원이 필수적이다.

본 논문에서는 효과적인 디지털 자산의 관리를 보장하는 디지털 자산 관리자를 구현하였다. 또한, 디지털 자산 관리자를 효과적으로 활용할 수 있는 각종 지원 도구들도 개발하였다. 개발된 디지털 자산 관리자는 디지털 자산 중 가장 복잡도가 높은 비디오 자산을 주요 대상으로 하고 있으며, 다양한 디지털 자산 관련 응용 환경 중 디지털 컨텐츠 저작을 주요 응용으로 가정한다. 본 논문에서 구현된 디지털 자산 관리자는 다음과 같은 특징을 갖는다.

- 계층 구조(Hierarchy)에 기반을 둔 비디오 자산 관리 방법을 지원한다. 즉, 하나의 비디오 자산을 단일 개체로 관리하지 않고, 비디오-세그먼트-샷으로 구성된 계층 구조를 사용하여 표현 관리한다. 이러한 비디오 자산 계층 구조를 통하여 사용자는 하나의 비디오 개체에서 필요한 일부분에 대한 효과적인 접근 및 활용 수단을 얻게 된다.

- 시스템의 이식성을 높이기 위해 저장 시스템과의 독립성을 최대한 보장한다. 개발된 디지털 자산 관리자는 다양한 하부 저장 시스템을 사용할 수 있도록 설계되었다. 이 때, 저장 시스템의 교체에 따른 부담을 최소화할 수 있도록 시스템을 설계하였다.

- 디지털 자산에 대한 빠른 내용 기반 유사성 검색 기능을 지원한다. 개발된 디지털 자산 관리자에서는 주어진 내용적 특성에 대한 조건을 바탕으로, 주어진 조건에 가장 적합한 k개의 자산을 검색하는 유사성 검색 기능을 지원한다. 이러한 내용 기반 유사성 검색 기능은 VA-File [3] 기법을 바탕으로 구현되었다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 본 논문에서 구현된 디지털 자산 관리자의 구조 및 특징적 기능들에 대하여 기술한다. 3장에서는 디지털 자산 관리자를 활용하는 응용 도구에 대해 알아보고, 4장에서는 디지털 자산 관리자에서 제공되는 내용 기반 유사성 검색 기능에 대하여 설명한다. 5장에서는 디지털 자산 관리 시스템 분야의 기존 연구 사례들을 살펴보고, 마지막으로 연구 결과의 정리 및 향후 계획을 6장에서 제시한다.

2. 디지털 자산 관리자의 설계 및 구현

2.1 비디오 자산의 관리

일반적으로 비디오 자산은 수 메가 바이트에서 수백 메가 바이트에 이르는 매우 큰 용량을 가지고 있다. 따라서, 편집 작업 시 이러한 대용량의 비디오를 단일 개체 혹은 단위로 조작하는 것은 비 효율적이다. 또한, 비디오 자산의 물리적인 구조는 하나의 스트림 형태이지만, 편집 작업을 위한 논리적인 관점에서 보면 그 의미에 따라 여러 구성 단위로 나누어 구성된다고 볼 수 있다. 이러한 이유로 비디오 자산의 관리는 자산의 효과적인 분할에 기반을 두어야 하며, 분할되는 구조 또한 선형적 분할이 아닌 계층적 분할 구조가 적합하다. 본 논문에서 개발한 DAM에서는 이러한 비디오 자산의 논리적 계층 구조를 기반으로 자산을 저장, 관리한다.

본 시스템에서는 '비디오-세그먼트-샷'의 세 단계로 이루어진 계층 구조 모델을 사용한다. 샷은 비디오 자산 계층 구조의 가장 하위 구성 요소로서, 관련된 프레임들의 집합으로 볼 수 있다. 일반적으로 샷을 구분하는 방법은 다양하게 존재하지만, 비디오 장면의 변화 또는 주체의 변화 등과 같은 화면의 변화를 그 단위로 한다. 세그먼트는 서로 연관된 샷들의 집합으로, 사용자의 의도에 따라 구성될 수 있는 단위이다. 즉, 사용자는 동일한 내용에 해당하는 샷들을 묶어 세그먼트를 구성할 수도 있으며, 편집 작업의 단위를 구분하고 지정하기 위하여 세그먼트를 사용할 수도 있다. 비디오는 전체 비디오 자산을 나타내는 단위로 세그먼트들의 집합에 해당한다.

이렇게 하나의 비디오 자산을 '비디오-세그먼트-샷'으로 구성된 계층 구조로 표현함으로써, 사용자는 전체 비디오 자산을 조작할 필요 없이, 자신이 원하는 범위 만큼의 비디오 자산만을 효과적으로 획득하여 조작할 수 있다. 또한, 계층 구조 상의 중간 계층에 해당하는 세그먼트 개념을 이용하여, 특정 샷들의 그룹화 및 이에 기반한 자산 편집 작업을 효과적으로 수행할 수 있다. 예를 들어, 여러 사용자 혹은 팀이 공동으로 하나의 비디오 자산에 대한 편집 작업을 수행하는 경우, 세그먼트는 이러한 사용자 또는 팀 간의 작업 영역을 구분하는 단위로 사용될 수도 있다. 그리고 각각의 계층별로 검색과 브라우징이 가능하도록 하여 사용자가 편집 작업을 할 때 더 효과적으로 검색과 브라우징을 할 수 있도록 하였다.

본 시스템에서는 비디오 계층 구조 상의 구성 요소 중 가장 하위 요소인 샷에 대해서만 물리적 인스턴스를 저장한다. 이 때, 세그먼트 및 비디오에 대한 물리적 인스턴스가 요청되는 경우, 그 하위 샷들을 병합(merge)하여 자동 생성한다. 이로 인하여, 전체적인 저장 영역의 효율적 사용이 가능하며, 계층 구조 요소들의 물리적 인스턴스들에 대한 일관성 유지 문제를 해결할 수 있다.

앞서 제시한 비디오 계층 구조는 비디오 자산에 대한 논리적 표현 모델에 해당한다. 이러한 표현 모델을 실제 시스템에 적용하기 위해서는 제시된 구조를 효과적으로 저장할 수 있는 저장 방법이 필요하다. DAM에서는 비디오 자산의 저장을 위해 다음의 그림 1과 같은 데이터베이스 스키마를 사용한다.

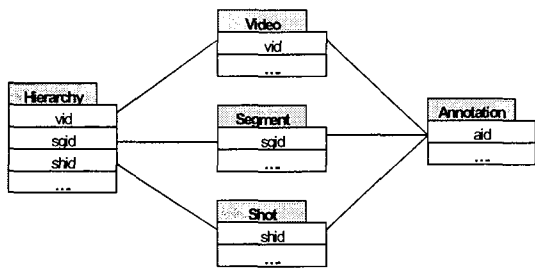


그림 1 비디오 자산의 저장을 위한 스키마 구조

그림 1에서 Video, Segment, Shot은 각각 계층 구조상의 요소인 비디오, 세그먼트, 샷의 정보를 저장하기 위한 테이블에 해당한다. 편의상 테이블의 자세한 속성 정보는 생략하였다. Annotation 테이블은 비디오 자산의 주석 정보를 저장하는 역할을 담당하며, 비디오, 세그먼트, 샷의 각각에 대한 주석 정보를 저장한다. 이와 같이 비디오, 세그먼트 및 샷에 대한 주석 정보를 동일한 테이블에 저장함으로써 주석 정보 기반 검색의 효율적인 수행을 제공할 수 있다. Hierarchy 테이블에는 비디오 자산의 계층 구조를 저장한다. Hierarchy 테이블에는 계층 트리 상의 경로(path), 즉 (비디오 식별자(vid), 세그먼트 식별자(sgid), 샷 식별자(shid))를 하나의 단위로 하여 저장함으로써 구성 요소들 간의 계층 구조 관계를 빠르고 정확하게 검색할 수 있다.

계층 구조 기반의 기본적인 비디오 자산의 관리 방법 이외에도, DAM에서는 가상 자산(virtual asset) 개념을 지원한다. 가상 자산이란 임의의 비디오 자산의 일부를 공유하여 다른 비디오 자산의 일부로서 취급할 수 있는 개념으로 실제 물리적으로는 하나의 인스턴스만이 존재하는 자산이 여러 비디오 자산의 구성원이 될 수 있다. 이와 같은 가상 자산 개념은 뉴스용 자료 화면 편집 등과 같이, 이미 존재하는 여러 디지털 자산들의 일부를 조합하여 새로운 자산을 구성하는 작업 등에 유용하게 사용될 수 있다.

2.2 시스템 구조

디지털 자산 관리자는 상용 DBMS 등과 같은 저장 시

스템의 상부에 위치하여 비디오 클라이언트 응용들로부터의 작업 요청을 처리해 주는 역할을 담당한다. 이러한 DAM의 전체적인 구조는 다음의 그림 2 와 같이 나타낼 수 있다.

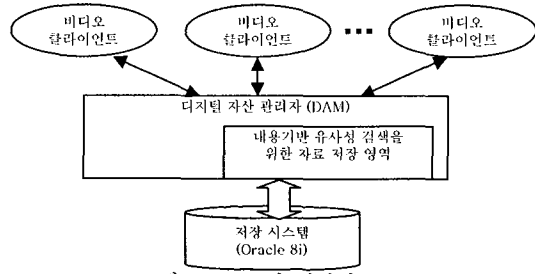


그림 2 DAM의 전체적 구조

그림 2 의 DAM 내부에 위치한 자료 저장 영역은 DAM에서 지원하는 특징적인 기능 중의 하나인 '내용기반 유사성 검색'을 위한 자료 저장 영역을 의미한다. 이에 대해서는 이후에 좀 더 자세히 언급한다.

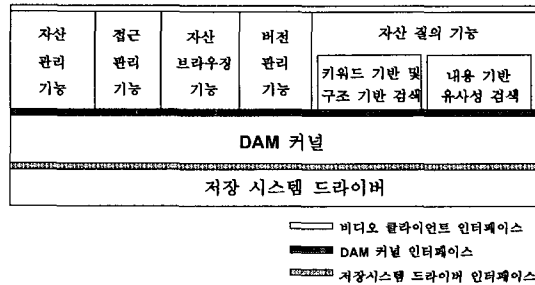


그림 3 DAM의 세부 구조

그림 3은 DAM의 내부 구조를 나타낸다. 기능 구현 모듈, DAM 커널 및 저장 시스템 드라이버의 세 계층으로 구성된 3-Tier 구조로서 각 계층마다 상위로부터의 접근을 위한 인터페이스가 정의된다. DAM의 가장 하부에는 저장 시스템 드라이버(Storage Driver)가 위치한다. 현재 개발된 DAM은 Oracle 8i를 하부 저장 시스템으로 사용하고 있으며, 따라서 Oracle 8i를 위한 저장 시스템 드라이버가 구현되어 있다. 저장 시스템 드라이버 계층과 그 상위 계층 사이에는 저장 시스템 드라이버 인터페이스가 정의되어, 표준화된 접근을 가능하게 한다. DAM에서는 저장 시스템에 종속적인 코드들을 저장 시스템 드라이버 계층에만 한정시켜 사용함으로써 저장 시스템과의 독립성을 보장한다. 상위 계층에는 저장 시스템의 종류에 무관한 기능 코드들만이 위치한다. 따라서,

하부 저장 시스템을 변경하는 경우, 저장 시스템 드라이버 계층만을 수정함으로써 시스템 이식이 쉽게 이루어질 수 있다. 저장 시스템 드라이버 계층의 상위에는 DAM 커널 계층이 위치한다. DAM 커널 계층은 각 기능 구현 모듈에서 사용하는 기능함수들로 구현되어 있으며, DAM 커널 인터페이스를 통해, 상위의 기능 구현 모듈에서 호출하여 사용된다. DAM의 가장 상위 계층에는 DAM에서 지원하는 각 기능들이 구현된 기능 구현 모듈 계층이 위치한다. DAM의 기능 구현 계층은 크게, 자산 관리 기능, 자산 질의 기능, 접근 제어 기능, 버전 관리 기능 및 자산 브라우징 기능의 다섯 가지 모듈로 구분되어 진다. 이들 각 기능 구현 모듈에서 제공하는 기능들이 모여, 비디오 클라이언트 인터페이스(Video Client Interface: 이하 VCI)를 구성하며, 비디오 응용 클라이언트들에서는 이들 VCI 함수들을 사용함으로써 DAM을 통한 자산 관리 및 접근을 수행할 수 있다.

DAM 은 Linux 환경에서 C 언어를 이용하여 구현되었다. 그림 4 는 DAM 에서 서로 다른 계층의 함수들이 어떻게 호출되는 가를 나타낸 인터페이스 호출 예제이다.

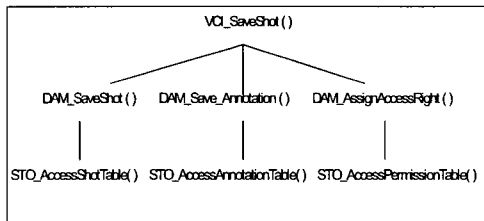


그림 4 DAM 인터페이스 호출 예제

2.3 DAM 의 주요 기능

① 자산 관리 기능

자산 관리 기능은 비디오 자산의 특성과 무관하게, 자료 관리 시스템으로서 기본적으로 갖추어야 하는 기능들을 의미한다. DAM에서 지원하는 기본적 자산 관리 기능은 다음과 같다. 각각의 기능들은 계층 구조 상의 비디오, 세그먼트, 샷 모두에 대해 적용 가능하며, 자산의 계층 구조 정보 및 주석 정보 또한 기본적 자산 관리 기능의 적용 대상이 된다.

- 자산의 삽입, 삭제, 읽기 및 변경 기능
- 가상 자산의 생성 및 삭제 기능
- 자산의 초기 적재(load) 기능

② 검색 기능

DAM에서 지원하는 자산 검색 기능은 키워드 기반 검색(keyword-based search) 과 구조 기반 검색

(structure-based search), 그리고 내용 기반 유사성 검색(content-based similarity search) 이 있다.

키워드 기반 검색은 자산의 주석 정보 및 자산 자체에 대한 정보에 대하여 주어진 키워드를 기반으로 검색을 수행한다. 즉, 사용자는 검색에 사용할 필드의 값이나 해당 필드가 포함하는 문자열들을 입력하게 되며, DAM은 이를 바탕으로 자산들을 지정된 필드에 대한 부분 일치 방식으로 검색하게 된다.

구조 기반 검색은 비디오 자산의 계층 구조를 기반으로 한 검색 방법으로, 특정 비디오를 구성하는 세그먼트 혹은 샷들의 집합을 구하거나, 또는 특정 샷 또는 세그먼트가 소속된 비디오등을 검색한다.

내용 기반 유사성 검색 기법은 질의 개체 또는 조건과 유사한 개체를 검색하는 방법으로 이에 대해서는 3장에서 자세히 기술한다.

③ 접근 제어 기능

DAM은 다수의 사용자가 여러 디지털 자산에 대해 공동의 작업을 수행하는 환경을 가정한다. 따라서, 작업의 효율성과 안정성을 보장하기 위해서, 시스템에서 관리되는 자산에 대한 사용자의 접근을 제어할 수 있어야 한다. DAM에서는 자산 접근의 주제로 '사용자'와 사용자들의 모임에 해당하는 '그룹'의 두 요소를 가정한다. 이 경우 특정 사용자에게 대해서 개인으로서의 접근 권한과 자산이 속한 그룹으로서의 접근 권한이 충돌하는 경우가 발생할 수 있다. DAM에서는 그룹 권한 보다 사용자 권한이 우선한다. DAM에서는 하나의 비디오 자산을 구성하는 비디오, 세그먼트, 샷에 대해 개별적인 접근 권한 부여가 가능하다. 이 때, 계층 구조상의 상/하위 구성 요소간의 접근 권한 충돌 방지를 위하여, 계층 구조상의 특정 요소에 대한 연산 권한은 해당 요소의 모든 하위 요소에 대하여 해당 연산의 권한이 있는 경우에만 설정된다는 규칙을 사용한다.

자산에 대한 접근 제어를 지원하기 위한 데이터베이스 스키마는 다음의 그림 5 와 같다. 그림 5의 AccessRight 테이블은 하나의 자산(assetID)에 대하여 어떤 사용자 또는 그룹(userGroupID)이 특정 연산(opType)을 수행할 수 있는지 여부(perm)를 나타내는 정보를 저장한다.

AccessRight			
assetID	userGroupID	opType	perm

그림 5 접근 관리를 위한 데이터베이스 스키마

접근 관리는 주작업인 사용자의 자산 접근의 전단계

작업이다. 따라서, AccessRight 테이블은 저장 공간의 효율성보다는 접근 가능 여부 판정 시간을 최소화하는 것을 가장 중요한 목표로 설계되었다.

DAM에서 제공되는 자산 접근 제어 관련 기능은 다음과 같다.

- 자산 접근 가능 여부의 조사 기능
- 접근 권한 부여 및 박탈 기능
- 자산 계층 구조 변화에 따른 접근 권한 조정 기능
- 자산 적재 시의 접근 권한 초기화 기능

④ 버전 관리 기능

일반적으로 디지털 자산에 대한 콘텐츠 저작 작업에서는 자산에 대한 변경 및 수정이 빈번하게 일어난다. 또한 저작 작업 자체가 비교적 긴 시간에 걸쳐 일어나므로, 저작 작업 기간 동안의 변경 및 수정 내용에 대한 버전 관리가 필요하다. 특정 자산에 대하여 새로운 버전이 생성되는 경우는, 자산의 일부에 대한 변경, 삽입, 삭제 등 매우 다양하다. DAM에서는 이러한 자산의 변경 시에 변경된 내용을 시간에 따라 기록할 수 있는 버전 관리 기능을 지원한다. DAM에서 지원되는 버전 관리 관련 기능은 다음과 같다.

- 특정 자산에 대한 새로운 버전 관리의 시작 기능
- 새로운 버전의 추가 기능
- 버전 정보 및 물리적 인스턴스의 읽기 기능
- 버전 트리 구조 정보 읽기 기능
- 시스템에서 관리 중인 버전 목록 반환 기능

위에 열거된 기능 중 버전 트리 구조 읽기 및 버전 목록 반환 기능들은 이후에서 설명할 자산 브라우징 기능을 효과적으로 지원해 주는 역할을 담당한다.

⑤ 브라우징 기능

자산 브라우징(browsing)이란 시스템에서 관리되는 비디오 자산들을 여러 가지 방법으로 전체적인 관점에서 살펴 볼 수 있는 기능을 의미한다. DAM은 사용 목적에 따라 다음의 세 가지 형태의 브라우징 방법을 제공한다.

- 사용자 정의 폴더 기반의 브라우징: 폴더(folder)란 사용자가 비디오 자산들을 선택, 등록하여 보관하는 가상의 컨테이너 개념을 의미한다. 즉, 폴더는 사용자 개인 별로 자신이 분류한 기준에 따라 자산들을 구분하여 보관하는 디렉토리의 개념에 해당한다. 이러한 사용자 정의 폴더는 파일 시스템의 디렉토리 구조와 마찬가지로 계층적 구조를 지원하며, 사용자는 폴더에 등록된 자산들을 쉽고 효과적으로 살펴볼 수 있다. 이 때, 폴더와 자산의 등록 관계는 해당 자산의 실제 저장 및 관리와는 무관하다. 즉, 특정 자산이 폴더에서 삭제되는 경우에는

해당 폴더와 관련된 등록 정보만이 삭제될 뿐이며, 자산의 물리적 저장에는 아무런 영향을 미치지 않는다. 사용자 정의 폴더 기반 브라우징을 지원하기 위한 데이터베이스 스키마는 다음 그림 6과 같다.

Register					
Fid	assetType	aid	assetName	assetKeyFrame	assetAnnotation
Folder					
Fid	parentFid	Fname	Desc	OwnerId	

그림 6 폴더 기반 브라우징을 위한 데이터베이스 스키마

사용자 정의 폴더 기반의 브라우징을 지원하기 위하여 DAM에서 지원하는 기능들은 다음과 같다.

- 1) 폴더 내의 개체 관리 기능: 폴더 내의 새로운 개체 등록 및 등록 해제
- 2) 폴더 관리 기능: 폴더의 생성, 삭제 및 이동
- 3) 폴더 검색 기능: 폴더 내에 포함된 자산에 대한 검색 기능

- 비디오 자산의 계층 구조 기반 브라우징: 계층 구조 기반 브라우징이란 하나의 비디오 자산을 구성하는 비디오, 세그먼트, 샷 기반의 계층 구조에 대한 항해(navigation) 방식의 브라우징 방법을 의미한다. 즉, 사용자는 특정 비디오에 속한 세그먼트들 또는 특정 세그먼트들을 구성하는 샷들을 자유롭게 살펴볼 수 있다. 이러한 계층 구조 기반 브라우징 기능은 내부적으로는 구조 기반 검색 기능과 동일한 DAM 커널 함수를 사용하여 구현된다.

- 버전 기반 브라우징: 앞서 언급한 바와 같이 DAM에서는 자산의 변경에 따른 버전 관리 기능을 제공한다. 즉, 논리적인 하나의 자산은 변경 시점에 따라 여러 버전의 물리적인 자산으로 저장된다. 버전 기반 브라우징이란 여러 개의 버전을 갖는 자산에 대해 버전별로 자산의 정보와 변경 시점을 브라우징할 수 있도록 해 주는 기능이다. 즉, DAM에서는 특정 자산의 버전 생성 정보는 나타내는 트리 구조 상을 항해 방식으로 브라우징할 수 있도록 해 준다.

3. DAM의 응용 도구 개발

본 논문에서는 디지털 자산의 관리를 담당하는 DAM 이외에, DAM의 효과적인 활용을 지원하기 위한 두 가지 비디오 클라이언트 응용 도구를 개발하였다. 본 장에서는 자산 입력 도구와 WWW 기반 자산 검색 및 브라우징 도구에 대하여 알아본다.

3.1 자산 입력 도구

자산 입력 도구는 클라이언트 측에 존재하는 비디오 자산을 손쉽게 DAM에 적재할 수 있도록 지원해 주는 도구이다. 개발된 자산 입력 도구는 하나의 비디오 자산을 샷 단위로 분할해주며, 사용자가 비디오 자산의 계층 구조를 지정하고 주석 정보를 입력할 수 있도록 해 준다. 또한 입력된 정보를 DAM에 전송하여 저장하는 기능을 제공한다. DAM 자산 입력 도구는 Visual Basic 응용 프로그램으로 작성되어 있으며, 다음의 그림 7은 자산 입력 도구의 실행 화면 예이다.

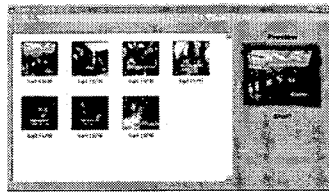


그림 7 자산 입력 도구의 실행 화면

3.2 WWW 기반 자산 검색 및 브라우징 도구

일반적으로 DAM 클라이언트는 DAM에서 제공하는 VCI 함수들을 이용하여 DAM에서 지원하는 기능들을 이용할 수 있다. WWW 기반의 DAM 클라이언트의 경우에도 마찬가지로 VCI 함수들을 사용하는 CGI (Common Gateway Interface)[4] 프로그램이 실제적인 DAM의 클라이언트 응용 프로그램이 된다. WWW 기반 DAM 응용 도구는 다음의 그림 8과 같은 방식으로 동작한다.

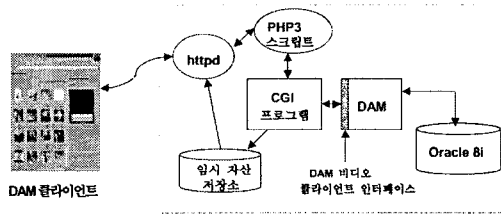
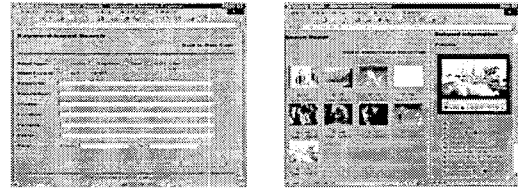
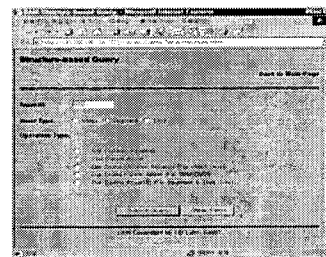


그림 8 WWW 기반 응용 도구의 동작 원리

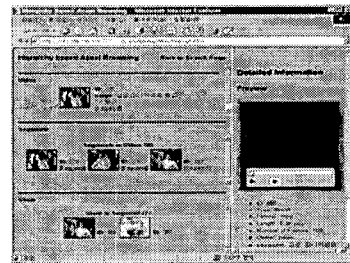
사용자는 웹 브라우저를 통하여 수행하고자 하는 기능 및 관련 정보를 입력하게 되고, 이러한 작업 요청은 httpd 를 통하여 서버측에 전달된다. 서버측에 전달된 작업 요청은 PHP 스크립트[5]를 이용하여 CGI 프로그램으로 전달되고, CGI 프로그램은 VCI 함수를 호출하여 해당 기능을 수행하게 된다. 그림 8에서 임시 자산 저장소 (temporary asset store)는 DAM에서 전달받은 비디오 자산의 물리적 파일들을 저장하여, httpd가 물리적 인스턴



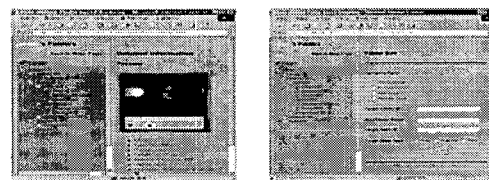
(a) 키워드 기반 검색



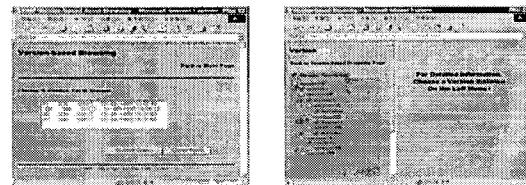
(b) 구조 기반 검색



(c) 계층 구조 기반 브라우징



(d) 사용자 정의 폴더 기반 브라우징



(e) 버전 기반 브라우징

그림 9 WWW 기반 DAM 자산 검색 및 브라우징 도구의 실행 예

스를 클라이언트 측에 전송할 수 있도록 해 주는 저장 영역을 의미한다.

다음의 그림 9는 본 논문에서 개발한 WWW기반 자산 검색 및 브라우징 도구의 다양한 실행 화면 예를 나타낸다. 그림 9의 (a)는 키워드 기반 검색의 검색 인터페이스 및 결과 화면 예를 나타내며, (b)는 구조 기반 검색의 검색 인터페이스를 나타낸다. 그림 9의 (c)는 계층 구조 기반 브라우징의 실행 예를 나타내며, (d)는 사용자 정의 폴더 기반 브라우징의 실행 예와 폴더 편집 인터페이스의 화면 예를, (e)는 버전 기반 브라우징의 실행 예를 나타낸다.

그림 9의 검색 결과 화면 및 브라우징 화면에서 좌측 프레임은 각각 검색 결과의 간략한 정보 또는 폴더 정보, 계층 구조 정보, 버전 정보 등을 나타내기 위한 영역이다. 사용자가 좌측 프레임 상의 특정 자산을 선택할 경우, 해당 자산에 대한 상세 정보가 그림 9 (a)의 결과 화면과 같이 도구의 우측 프레임 영역에 표시된다.

4. DAM의 내용 기반 유사성 검색 기능

본 장에서는 DAM에서 지원하는 내용기반 유사성 검색 기법(content-based similarity search)에 대하여 기술한다. 내용기반 유사성 검색[2]이란 사용자가 질의의 조건으로 표본 자산(seed asset) 혹은 특징(feature) 정보 값을 입력하였을 때, 주어진 표본 자산 혹은 입력 정보와 유사한 k개의 자산을 찾는 검색 방법을 의미한다.

지금까지 유사성 검색을 위한 많은 방법들이 제안되어 왔으며, 분할 색인 구조 (partitioning index structure)가 그 대표적인 예이다. 분할 색인 구조는 전체 데이터 공간을 트리 구조를 바탕으로 분할하여 유사한 데이터들을 분류해 놓은 후, 이를 바탕으로 검색을 수행하는 방법이다. 최근에 제안된 분할 색인 구조의 대표인 연구로서 TV-트리 [6], X-트리[7], SS-트리[8], SR-트리[9] 등이 있다. 그러나 일반적으로 10차원 이상의 고차원 데이터에 대한 유사성 검색에 있어서는 분할 색인 구조의 성능은 선형 탐색(linear scan)보다도 좋지 못하다고 알려져 있다 [3] [10].

본 논문에서 구현한 디지털 자산 관리자에서는 VA (Vector Approximation) 파일 방법[3]에 기반을 둔 내용기반 유사성 검색 기능을 구현하였다. VA 파일 방법은 객체 근사화 (object approximation)에 기초를 둔 방법으로, 지금까지 제안된 유사성 검색 방법 중 가장 우수한 방법으로, 10차원 이상에서도 선형 검색보다 빠른 성능을 보인다고 알려져 있다 [3].

4.1 VA 파일의 개념

일반적으로 다차원 자료는 N-차원 벡터 공간상의 한 점으로 사상될 수 있으며, 두 자료의 유사성은 벡터 공간 상에서의 두 자료에 해당하는 점들간의 거리로 나타내어 질 수 있다. VA 파일(Vector Approximation File)은 다차원 자료에 대한 벡터 근사치의 배열 구조로서, 실제 자료의 정확한 값이 아닌 적은 수의 비트로 표현된 근사값에 대한 빠른 선형 검색을 통하여, 검색 결과에서 제외될 자료들을 빠르게 필터링하는 방법이다.

그림 10은 벡터 공간상의 자료들과 그에 대한 벡터 파일 및 VA 파일 구조의 예를 나타내고 있다.

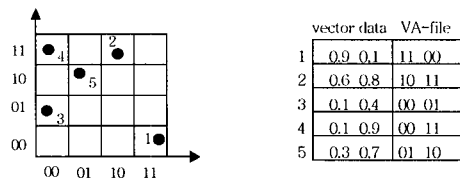


그림 10 VA 파일의 구조

VA 파일을 이용한 유사성 검색은 다음의 두 단계를 통하여 수행된다.

- ① 필터링 단계: 벡터 데이터의 근사치를 이용하여 검색 결과와 거리가 먼 대부분의 자료들을 걸러내는 과정이다. 이 과정은 VA 파일에 대한 선형 탐색을 통하여 이루어지며, 검색 결과에 포함될 가능성이 있는 자료들을 힙(heap) 구조에 저장하게 된다.
- ② 벡터 데이터 접근 단계: 필터링 단계의 수행 후 힙에 저장되어 있는 근사치의 실제 벡터 값을 조사하여, 실제 k개의 검색 결과를 얻는 단계이다.

4.2 DAM에서의 내용 기반 유사성 검색 기법의 구현

DAM에서 유사성 검색 기능이 수행되는 과정은 다음의 그림 11과 같다. DAM에서 관리되는 디지털 자산들이 하부의 저장 시스템을 이용하여 관리되는 데 반하여, 유사성 검색 기능의 수행은 VA 파일이라는 독립적인 저장 구조를 필요로 한다. 따라서, DAM에서는 자산들에 대한 내용 특징 정보들을 VA 파일 구조를 이용하여 따로 관리하게 되며, 이를 이용하여 내용기반 유사성 검색 기능을 지원한다.

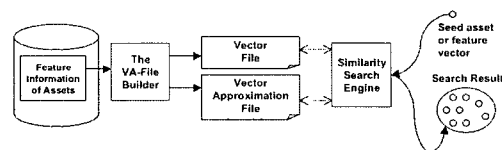


그림 11 DAM에서의 내용 기반 유사성 검색 과정

위의 유사성 검색 기능 지원을 위하여, 본 논문에서 구현된 DAM에서는 다음과 같은 기능들을 구현하였다.

- 표본 자산에 의한 유사성 검색 기능: DAM에서 관리되는 자산 중 사용자가 지정한 자산의 내용 특징(feature) 정보와 가장 유사한 k개의 자산을 검색하는 기능이다. 사용자는 자산의 내용 특징 정보가 갖는 내부적인 구조를 알 필요가 없으므로, 시각적 인터페이스를 통한 유사성 검색의 수행에 적합하다.

- 내용 특징 정보에 의한 유사성 검색 기능: 사용자가 입력한 내용 특징 정보를 바탕으로, 해당 정보와 유사도가 가장 높은 k개의 자산을 검색하는 기능이다.

또한, DAM에서는 VA 파일 정보와 저장 시스템에서 관리되는 자산 정보와의 연계를 위하여 다음의 기능들이 지원된다.

- VA 파일 생성 기능: DAM에서 관리되는 내용 특징 정보를 바탕으로, 이에 대한 VA 파일 구조를 자동적으로 생성해주는 기능이다.

- VA 파일 갱신 기능: DAM에서 내용 특징 정보의 변경이 발생하였을 때, 이에 대한 VA 파일의 내용을 자동적으로 갱신하는 기능이다. 갱신 시의 부담을 최소화하고자 하는 경우에는 주기적으로 VA 파일을 재 생성하는 방법이 사용될 수도 있다.

그림 12는 DAM에서 제공되는 내용기반 유사성 검색 기능을 이용하여 구현한 검색 도구의 실행 예를 나타낸다. 그림의 예는 지정된 자산과 가장 유사한 10개의 자산을 검색한 결과는 나타내고 있으며, 결과 자산들의 유사 정도 및 검색 수행 시간이 표시되었다.

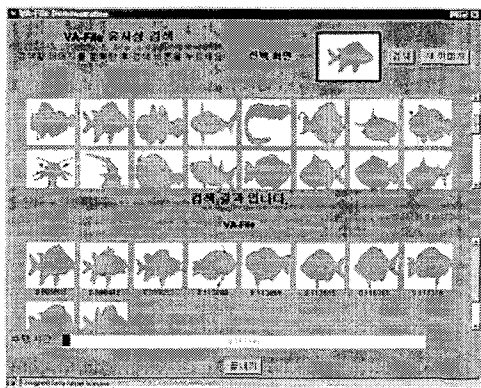


그림 12 DAM의 내용 기반 유사성 검색의 예

4.3 성능 분석

본 논문에서는 구현된 유사성 검색 기법의 성능 측정을

을 실시하였다. 성능 비교를 위하여 분할 색인 구조 기반 검색 기법 중 가장 좋은 성능을 나타낸다고 알려진 SR-Tree[9] 기반 검색법과 수행 시간 및 저장 공간 요구 사항을 비교하였다. 실험에서는 제안자가 공개한 검색 SR-Tree 기반 검색 프로그램 소스[11]를 수정 없이 사용함으로써 결과의 신뢰도를 높이고자 노력하였다. 성능 측정에 사용된 주요 패러미터는 다음과 같다.

- 자산의 개수(N): 10000
- 내용 특징 정보의 차원(Dimension)의 수: 10 ~ 50
- 검색 결과의 개수 (k): 10, 100

그림 13과 14는 각각 k = 10과 k = 100의 경우 유사성 검색에 소요되는 수행 시간을 측정한 결과이다. 또한 그림 x는 VA 파일 기반 방법과 SR-Tree 기반 방법의 저장 공간 요구 사항을 나타낸다. 결과에서 알 수 있듯이,

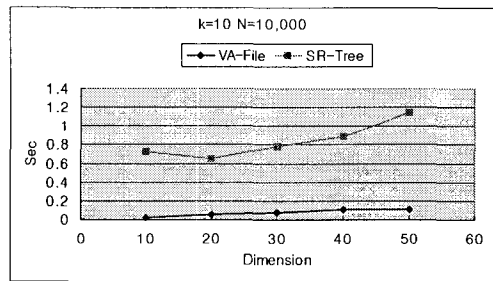


그림 13 유사성 검색 수행 시간 측정 결과

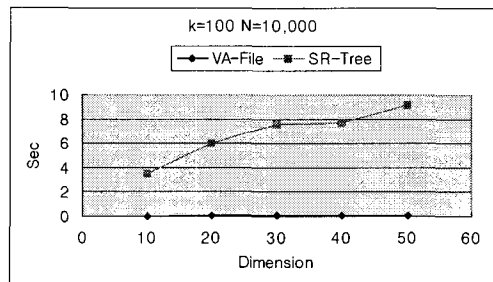
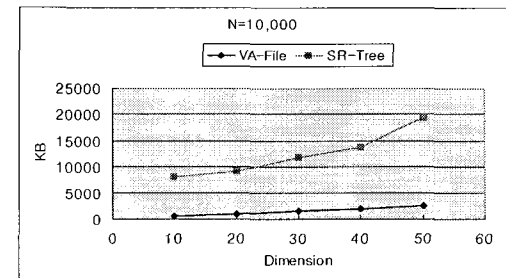


그림 14 저장 공간 요구 사항 측정 결과



DAM에서 구현된 VA 파일 기반 유사성 검색 기법이 기존의 SR-Tree 기법보다 수행 시간 및 저장 공간 효율성 면에서 우수함을 알 수 있다.

5. 관련 연구와의 비교

디지털 자산과 관련된 연구 분야는 자산의 저장 및 관리에서부터 네트워크를 통한 자산의 스트리밍에 이르기까지 매우 다양하다. 이 절에서는 본 논문에서 개발한 디지털 자산 관리자와 직접적인 관련이 있는 기존 연구 사례에 대해서 기술한다. 본 논문에서 개발한 디지털 자산 관리자는 '컨텐츠 저작 및 편집' 응용 환경을 위한 자산 관리 시스템이다. 기존 시스템 중 본 디지털 자산 관리자와 유사한 대표적인 상용 시스템으로는 Excalibur 사의 Screening Room [12]과 Virage 사의 VideoLogger[13]를 들 수 있다.

Screening Room 시스템은 Capture Client, Video Asset Server, Browser Client 및 Edit Client의 네 부분으로 구성되어 있다. 이 중, Capture Client는 특정 비디오 자산에서 화면의 변화가 일어나는 부분들을 구분하여 이야기 진행 방식으로 구성해 주는 기능을 담당하며, Video Asset Server를 이용하여 비디오 자산에 대한 정보 및 물리적 인스턴스가 저장된다. 사용자는 Browser Client를 이용하여 비디오 자산에 대한 검색 및 브라우징을 수행할 수 있다. 브라우징은 Capture Client에서 구성한 이야기 진행 방식에 따라 이루어지며, 키워드 기반 검색 기능과 표본 이미지를 이용한 유사성 검색 기능을 지원한다. 또한, Edit Client를 이용하여 자산의 정보 및 물리적 인스턴스를 편집할 수 있도록 지원한다.

표 1 디지털 자산 관리 시스템의 비교

	본 연구	Excalibur사의 Screening Room	Virage사의 Video Logger
자산 분할방식	3 단계의 계층 구조 분할	전형적 분할	선형적 분할
검색 기능	키워드 검색 및 유사성 기반 검색	키워드 검색 및 유사성 기반 검색	키워드 검색
브라우징 기능	폴더, 계층, 버전 기반의 브라우징	스토리보드 형식의 선형적 브라우징	스토리보드 형식의 선형적 브라우징
접근 제어	지원함	지원하지 않음	지원하지 않음
버전 관리	지원함	지원하지 않음	지원하지 않음
기타 기능	가상 자산 개념	텍스트 추출 기능	다양한 해상도의 비디오 생성 기능

VideoLogger 시스템 또한 Screening Room 과 동일한 이야기 진행 방식에 따른 비디오 자산의 구분을 지원하며, 이야기 진행 방식 기반의 브라우징 기능과 키워드 기반 검색 기능을 지원한다. VideoLogger의 특징적인 기능으로는 여러 해상도에 의한 다양한 비디오 버전의 생성 기능이 있다. 즉, 비디오의 원본 자산과 미리 보기 등의 작업에 사용되는 자산 버전의 해상도를 다양하게 지원함으로써 자산의 재생 시 효율성을 높이고자 하였다.

앞서 언급한 두 시스템은 모두 비디오 자산을 선형적으로 분할하고 관리한다. 즉, 하나의 비디오 자산을 연속된 샷들의 모임으로 저장, 관리하게 된다. 이에 반하여 본 논문에서 구현된 디지털 자산 관리자는 비디오, 세그먼트, 샷으로 구성된 계층 구조를 기반으로 자산을 표현함으로써 자산의 보다 효율적인 접근과 활용을 보장한다. 또한 지원되는 검색 및 브라우징 방법에 있어서도 본 논문에서 구현된 디지털 자산 관리자가 훨씬 다양한 기능을 제공한다.

6. 결론

최근 들어 디지털 콘텐츠를 활용한 다양한 응용 분야가 급속히 발전함에 따라, 디지털 자산의 효과적인 저장 및 관리의 중요성이 대두되고 있다. 효율적인 디지털 자산 관리를 위해서는 방대한 양의 자산에 대한 내용 기반의 빠른 검색 방법과 특정 자산 중 필요한 일부분만을 활용할 수 있는 효과적인 접근 수단이 필수적이다. 본 논문에서는 이러한 효율적인 자산 관리를 보장하는 디지털 자산 관리자를 설계 및 구현하였다. 본 논문에서 구현된 디지털 자산 관리자는 비디오 자산을 계층 구조에 따라 저장함으로써 자산 활용의 효율성을 높였으며, VA 파일 기반의 빠른 내용기반 유사성 검색 기능을 지원한다. 또한 하부 저장 시스템과의 독립성을 보장하는 시스템 구조를 채택함으로써 다양한 시스템 상에서 사용할 수 있는 특징을 지니고 있다.

본 논문에서 개발된 디지털 자산 관리자는 콘텐츠 저작 응용 분야를 목표로 개발되었다. 개발된 디지털 자산 관리자는 멀티미디어 저작 응용 및 방송용 디지털 자료 편집 응용 등과 같은 다양한 응용 분야에서 폭 넓게 활용될 수 있으며, 대용량 디지털 자료 검색 시스템의 기반 기술로도 사용될 수 있다. 향후 연구 분야로서 콘텐츠 저작 이외의 응용 분야를 지원하기 위한 시스템의 기능 확장 및 디지털 자산의 효율적인 저장을 고려한 저장 시스템의 개발이 요구된다.

참고 문헌

[1] E. Badique and A. Barbagli, "Tools for Multimedia Content Creation, Manipulation and Management: The ACTS Perspective" Lecture Notes in Computer Science, 1997.

[2] Thomas Seidl and Hans-Peter Kriegel, "Efficient User-Adaptable Similarity Search in Large Multimedia Databases," VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, 1997.

[3] R. Weber, H. -J. Schek and S. Blott, "A Quantitative Analysis and Performance Study for Similarity Search Methods in High-Dimensional Spaces," In Proc. VLDB, 1998.

[4] J. Dwight, M. Erwin and R. Niles, "Special Edition, Using CGI," QUE Publishing Co., 1997

[5] <http://www.php.net/>

[6] K. -I. Lin, H. Jagadish, and C. Faloutsos, "The TV-tree: An Index Structure for High-dimensional Data," VLDB Journal, 3, 1995.

[7] S. Berchtold, D. A. Keim, and H. P. Kriegel. "The X-Tree: An Index Structure for High-dimensional Data," In proceedings of VLDB, 1996

[8] D. A. White and R. Jain, "Similarity Indexing with the SS-tree," In proceeding of ACM SIGMOD, 1996.

[9] N. Katayama and S. Satoh, "The SR-Tree: An Index Structure for High-Dimensional Nearest Neighbor Queries," In proc. of ACM SIGMOD, 1997

[10] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is Nearest Neighbor Meaningful?," submitted for publication, 1998.

[11] <http://www.rd.nacsis.ac.jp/~katayama/homepage/research/srtree/English.html>

[12] <http://www.excalib.com/>

[13] <http://www.virage.com/>



김종수

1995년 2월 한국과학기술원 전산학과 학사. 1997년 2월 한국과학기술원 전산학과 석사. 1997년 3월 ~ 현재 한국과학기술원 전자전산학과 전산학전공 박사과정 재학중. 관심분야는 시간지원 데이터베이스, OLAP



방수호

1999년 2월 연세대학교 컴퓨터과학과 학사. 1999년 3월 ~ 현재 한국과학기술원 전자전산학과 전산학전공 석사과정 재학중. 관심분야는 이동 컴퓨팅, 멀티미디어



정연돈

1994년 2월 고려대학교 전산과학과 학사. 1996년 2월 한국과학기술원 전산학과 석사. 2000년 한국과학기술원 전자전산학과 전산학전공 박사. 관심분야는 이동 컴퓨팅, 분산 시스템, 멀티미디어, OLAP, DW임.



이재형

1995년 2월 한국과학기술원 전산학과 학사. 1995년 3월 ~ 1998년 5월 LG소프트 연구원. 2000년 2월 한국과학기술원 전산학과 석사. 2000년 4월 ~ 현재 행정자치부 전산사무관시보. 관심분야는 database systems, semi-structured

data.



김민정

1998년 2월 연세대학교 전산과학과 학사. 2000년 2월 한국과학기술원 전산학과 석사. 2000년 2월 ~ LG전자 기술원 재직. 관심분야는 OLAP, Data Warehouse, Data Mining



김명호

1982년 서울대학교 컴퓨터공학과 학사. 1984년 서울대학교 컴퓨터 공학과 석사. 1989년 MICHIGAN 주립대 전산학과 박사. 1989년 MICHIGAN 주립대 연구원. 1989년 ~ 1993년 한국과학기술원 조교수. 1993년 ~ 1999년 한국과학기술원 부교수. 1999년 ~ 현재 한국과학기술원 교수. 1993년 개방형 컴퓨터 통신 연구회(OSIA) 분산트랜잭션처리 분과위(TG-TP)의장. 1993년 ~ 1994년 한국통신기술협회(TTA) 분산 트랜잭션처리 실무위원회 의장. 관심분야는 분산데이터베이스, 분산트랜잭션, 멀티미디어 데이터베이스



장덕호

1981년 성균관대학교 전기공학과 학사. 1983년 성균관대학교 전기공학과 석사. 1991년 ~ 현재 한국과학기술원 박사과정. 1993년 ~ 현재 한국전자통신연구원 선임연구원. 1994년 ~ 1995년 Transarc 사. Visiting Researcher.



박 중 승

1992년 2월 경북대학교 전자계산학과 이
학사. 1994년 포항공과대학교 대학원 전
자계산학과 공학석사. 1999년 2월 포항
공과대학교 대학원 전자계산학과 박사.
1999년 3월 ~ 현재 한국전자통신연구원
선임연구원. 관심분야는 컴퓨터 비전, 패

턴 인식



오 황 석

1992년 2월 경북대학교 자연과학대학 전
자계산학과 졸업(학사). 1994년 2월 한국
과학기술원 전산학과 졸업(공학석사).
1999년 2월 한국과학기술원 전산학과 박
사. 1999년 3월 현재 한국전자통신연구
원 선임연구원. 관심분야는 Video

coding, Parallel processing for video coding, Multimedia
service system