

더블린 코어 모델을 이용한 비디오 데이터의 표현

이 순 희[†] · 김 상 호^{**} · 신 정 훈^{***} · 김 길 준^{****} · 류 근 호^{*****}

요 약

지금까지 대부분의 메타데이터들은 응용 분야에 제한된 부분만을 주로 취급하였다. 그러나 동일한 비디오 데이터를 표현하기 위해서는 동일한 형태의 메타데이터가 필요하고, 이때 비디오 데이터베이스에서 동일한 비디오 데이터에 대하여 서로 다른 여러 개의 메타데이터를 지원해야 하는 문제가 발생한다. 이 논문에서는 이러한 문제를 해결하기 위하여 더블린 코어 모델을 확장하였다. 제안된 비디오 데이터 표현에서는 더블린 코어 모델을 확장한 메타데이터가 비디오 데이터의 구조, 내용 및 조작에 관한 정보를 관리하도록 하였다. 제안된 메타 데이터는 시스템 관리 부분과 사용자 정의 부분을 분리함으로써 응용 분야에 독립적인 모델 구축이 가능하다. 13개의 시간 관계 연산은 더미 샷의 시간 변환 관계를 사용하여 6개로 감소시켰다. 이 감소된 6개의 연산은 역관계를 배제시켜 표현의 일관성을 유지시키고, n-ary 시간 관계의 샷들을 이진관계로 변환시킨다. 그리고 실제 응용 분야에 적용하고 실험하여 확장된 더블린 코어 모델이 응용 분야에 동일한 구조로 메타데이터를 표현하고 동일한 방법으로 검색할 수 있음을 증명하였다.

Representation of Video Data using Dublin core Model

Soon Hee Lee[†] · Sang Ho Kim[†] · Jung Hoon Shin^{***}
Gil Choon Kim^{****} · Keun Ho Ryu^{*****}

ABSTRACT

As most of metadata have been handled on restricted applications, we need a same metadata in order to represent a same video data. However, these metadata make problems that the same video data should be supported by the same metadata. Therefore, in this paper, we extend the Dublin core elements to support the metadata which can solve the problems. The proposed video data representation is managed by the extended metadata of Dublin core model, by using the information of structure, content and manipulation of video data. The thirteen temporal relationship operators are reduced to the six temporal relationship operators by using a dummy shot temporal transformation relationship. The reduced six temporal relationship operators through excluding reverse temporal relationship not only maintain a consistency of representation between a metadata and a video data, but also transform n-ary temporal relationship to binary relationship on shots. We show that the proposed metadata model can be applied to representing and retrieving on various applications as equivalent as the same structure.

키워드 : 비디오 데이터의 표현(Representation of Video Data), 메타데이터(Metadata), 시간 연산자(Temporal Operator)

1. 서 론

인터넷의 확산과 각종 멀티미디어 데이터의 증가로 단순한 텍스트에 의한 검색이 아닌 내용 기반 및 브라우징에 의한 검색을 요구하고 있다. 또한 컴퓨터 기술의 발달로 대용량 저장 장치가 개발되고 있으며, 연산 속도도 매우 빠른 속도로 발전하고 있어 멀티미디어 데이터에 대한 저장 및 검색을 수용할 수 있는 기반이 갖추어지고 있다. 또한 많은 상용 데이터베이스 관리 시스템들도 각종 멀티미디어를 지

원하기 위한 BLOB(Binary Large Object)형 데이터를 취급하며, 이제는 한 개의 레코드에 여러 개의 BLOB형 데이터를 사용할 수 있게 되었다[1]. 그리고 DataBlade라는 기능을 이용하여 비디오 데이터의 표현을 지원하고 있으며 [2], SGML, XML 등의 언어에서는 멀티미디어 문서를 표현하기 위하여 메타데이터의 표준화가 진행되고 있다[3].

비디오 데이터는 이진 형태의 비트 스트림으로 취급된다. 이진 형태의 데이터는 내용에 상관없이 디지털화 되어 BLOB으로 생성된다. 이와 같은 비디오 데이터를 의미 있는 데이터로서 사용하기 위해서는 이들 내용을 식별할 필요가 있다. 이런 비디오 객체들의 내용에 관한 서술을 메타 데이터라고 부른다[4, 5]. 이런 비디오 데이터를 데이터베이스에서 해석하기 위해서는 비디오 데이터에 관한 구조 및 내용들을 식별하여 메타데이터로서 데이터베이스에 저장되

* 이 연구는 한국과학재단 RRC(청주대 정보통신 연구센터)의 지원으로 수행되었음.

† 정 회 원 : 안산공과대학 컴퓨터정보과 교수

** 준 회 원 : 충북대학교 대학원 전자계산학과

*** 준 회 원 : 익산대학교 컴퓨터학과 교수

**** 정 회 원 : 성결대학교 전자상거래학부 교수

***** 정 회 원 : 충북대학교 전기전자컴퓨터공학부 교수

논문접수 : 2000년 10월 24일, 심사완료 : 2002년 5월 13일

어야 한다[6].

메타데이터는 비디오 데이터의 내용, 구조 그리고 프레임에 존재하는 객체들의 각종 정보를 다루며, 이들 정보의 생성은 응용 분야에서 필요한 미디어 타입과 요구되는 정보의 형태에 의존한다. 그리고 메타 데이터는 비디오 관련 처리 기술에 의존하여 비디오 데이터의 카메라 샷(shot), 샷안의 성격, 샷의 배경 등을 식별해야 한다[7, 8]. 또한 내용 기반 검색을 지원하기 위하여 데이터에 포함된 정보들을 추출하여 내용에 대한 사용자 질의를 지원할 수 있도록 적절히 구조화해야 하고[9], 메타데이터의 특성에 따라 분류해야 한다. 아울러, 내용을 설명하는 데이터도 적절히 분류하면 동일한 비디오 정보를 다양한 관점을 제공할 수 있으며, 메타데이터의 관리와 수정도 용이하다[10]. 이와 같이 비디오 데이터에 대한 내용 기반 검색을 지원하기 위하여 메타데이터는 어떤 비디오 데이터 구조도 쉽게 해석하고, 모호하지 않고 쉽게 파싱(parsing)할 수 있어야 한다. 그리고 주석, 데이터의 공간적인 구조, 주관적, 객관적 특징, 또는 사건이나 활동 등을 표현할 수 있어야 한다[11].

더블린 코어 모델(DC 모델 : Dublin Core Model)은 1995년에 웹에서 자원 탐색을 위한 규정을 개발하기 위해 처음 시작되었으며, 사람이 만든 문서에서부터 소프트웨어에서만 사용 가능한 데이터에 이르기까지 거의 모든 범위의 문서를 대상으로 메타데이터에 관한 표준화를 진행하고 있다. 여기서 제안된 요소들은 총괄적이고, 자원 탐색을 지원하는데 유용한 요소들이 널리 채택되었다[3].

DC 모델은 적용 분야를 광범위하게 제시하고 있으나 아직까지 각 분야별로 표준화가 진행되고 있는 단계이며, 특히 시간 관계에 대한 정의가 명확하지 않으며, 비디오 편집을 위한 부분은 일부 효과에 대한 부분에 관한 부분만 제시되고 있다.

AMOS(Active Media Object Stores)에서는 멀티미디어 응용에는 멀티미디어 객체에 관한 메타데이터의 저장에 필요성을 강조하고, 이들 메타데이터들을 타입별로 나누어 메타데이터로 저장하고 있다[12]. AMOS는 비디오 데이터의 자원의 조건을 지정하는 메타데이터와 질적 요소에 관한 메타데이터, 메타 속성, 내용 속성에 관한 메타데이터, 비디오 데이터의 물리적 위치와 내력에 관한 메타데이터를 제공하고 있다. 그러나 이들은 시간 및 공간을 지원하지 못하며, 또한 비디오 편집에 대한 부분을 제시하지 못하고 있으며, 제한된 영역의 메타데이터만을 제시하고 있다.

VideoSTAR(video storage and retrieval)[4, 13]에서는 주석을 위한 내용의 구성은 primary, basic, secondary의 단계로 표현을 하고 있으나, 이런 분류 표현의 단점은 의미 부여가 복잡하고, 내용에 대한 다양한 접근이 미흡하다는 점이다.

Plateau[14]에서는 VOD를 위해 필요한 메타데이터는 잘

정의되어 있으나 체계적인 메타데이터 분류가 취약하고 내용 독립적인 메타데이터가 고려되지 않았다.

이들의 연구는 대부분 특정한 응용 분야에 한정하여 해당 분야에 대한 상세한 부분의 메타데이터까지도 제시하고 있다. 그러나, 이들은 응용 분야에 치우친 메타데이터를 제시하고 있어, 이들 모델은 일반적이지 못하여 다른 응용 분야에 적용할 수 없다는 문제점을 안고 있다. 이렇게 응용 분야별로 서로 다른 구조의 메타데이터를 이용해야 한다면, 비디오 데이터베이스의 관점에서 데이터를 관리하기 위하여 응용 분야별로 다른 구조의 메타데이터를 가져야 하는 문제점이 있다.

따라서 이 논문에서는 비디오 데이터의 시간 관계, 비디오 편집 효과와 메타데이터 적용 분야의 제한성을 해결하기 위하여, 메타데이터의 표준화가 진행되고 있는 DC 모델을 적용한 일반화된 비디오 메타데이터를 제안한다. 제안하는 모델은 DC 모델을 기본으로 하여 시간 영역을 확장하고, 비디오 데이터의 편집도 지원할 수 있도록 한다. 또한 내용 독립적인 메타데이터와 내용 기반 메타데이터가 응용 분야에 독립적이거나 응용분야에 기반한 메타데이터의 독립된 구조를 갖도록 함으로써 관련 응용 분야에 적용 가능한 비디오 메타데이터를 제안한다.

논문의 구성은 제 2장에서는 메타데이터와 관련된 기존 연구를 검토하고, 이들의 문제점을 지적하고 해결점을 제시한다. 그리고 제 3장에서는 DC 모델을 기반으로 비디오 데이터를 표현할 수 있는 메타데이터를 제시하고 제안된 모델의 장단점을 비교 설명하고 4장에서는 제안 모델의 적용례를 설명하고 기존 연구와 비교 검토한다. 마지막으로 5장에서는 결론을 맺는다.

2. 관련 연구

비디오 데이터를 의미 있는 데이터로서 사용하기 위해서는 내용을 식별할 수 있는 문장으로 서술된 메타데이터가 필요하다. 또한 비디오 데이터를 데이터베이스에서 해석하기 위해서는 비디오 데이터에 관한 구조에 관한 내용 독립 메타데이터와 내용 관련 메타데이터를 구분하여 데이터베이스에 저장되어야 한다[6]. 이런 메타데이터를 이용하여 텍스트, 이미지, 동영상과 같은 멀티미디어 데이터를 표현하고자 하는 많은 연구가 수행되었다[4-10, 15]. 그러나 인터넷의 발달으로 보편화되기 시작한 웹에서 정보 검색하는데 필요한 메타데이터에 관한 규정이 논의되고 있는 DC 모델이 있다. 이 모델은 텍스트, 이미지, 동영상 등과 같은 인터넷상의 모든 자원 탐색을 목적으로 연구되고 있으며, 메타데이터의 요소들은 총괄적이고, 자원 탐색을 지원하는데 유용한 핵심 요소들을 포함하고 있다. 이 메타데이터는

제목, 주제, 설명 등의 15개 요소로 이루어져 있으며, 이들 속성에 대한 단순한 값으로 표현하거나 메타데이터의 의미를 세밀히 분류하거나, 일일이 열거된 한정어(qualifier)들로 표현하거나 혹은 메타데이터가 복합적이거나 구조적 값이라는 것을 일일이 나열하여 추가 정보를 채용하여 표현할 수 있다.

<표 1> 더블린 코어 메타데이터의 종류

요 소	설 명
제목(title)	대상이 되는 객체나 일 등의 이름
제작자(creator)	대상이 되는 객체나 일 등을 만들었거나 최고 책임자
제목(subject)	대상이 되는 객체나 일 등을 한마디로 설명해 주는 단어들
주석(description)	대상이 되는 객체나 일 등의 내용이나 목적 등을 자세히 설명
출판자(publisher)	대상이 되는 객체나 일 등을 현재의 형태로 출판 가능하게 하는데 책임있는 대행자나 대행사 또는 사람
기여자(contributor)	대상이 되는 객체나 일 등을 하는데 기여한 편집자, 삽화가와 같은 사람 단 Creator는 제외
날자(date)	현재의 형태로 출판 가능하게 된 날
종류(type)	이미지, 소설 등과 같은 대상이 되는 객체나 일 등을 만든 형태
형식(format)	수행 파일, PS 파일 등과 같은 대상이 되는 객체나 일 등을 만든 형태
식별자(identifier)	대상이 되는 객체나 일 등을 유일하게 식별할 수 있는 문자열, 숫자와 같은 것
출처(source)	다른 원본으로부터 내용을 참조했을 경우에 그 대상
언어(language)	대상이 되는 객체나 일 등에 사용한 언어
관계(relation)	대상이 되는 객체나 일 등이 다른 것들과의 관계
적용범위(coverage)	대상이 되는 객체나 일 등의 공간적 장소나 시간적 기간
권한(rights)	대상이 되는 객체나 일 등을 사용할 수 있는 권한이나 권리

DC 모델은 <표 1>에서 나타낸 것과 같은 메타데이터 요소를 정의하여 사용하고 있다. 이들은 각각의 항목에 대하여 영역별로 세분하여 표시된다. 즉 모든 멀티미디어 데이터를 일관적인 형태로 표현되며, 동시에 각각의 응용 영역별로 필요한 세부 사항들을 추가하여 기술함으로써 각각의 응용 분야에 필요한 메타데이터를 표현한다.

DC 모델은 몇 가지 기본 원칙을 제한하고 있는데, 규정된 필수 데이터 요소 이외에 부차적인 내용이나 특성을 기술 요소로 사용할 수 있는 확장성을 보장하고 있으며, 또한 각 요소의 수록 여부를 강제하지 않는 선택성이라든지, 모든 기술 요소는 반복 사용할 수 있는 반복성과 한정어를 두어 세부 사항을 조정할 수 있다.

DC 모델은 대부분의 응용에서 필요한 기술 요소들을 포함하고 있으며, 이들 요소에 대한 확장을 지원하고 있기 때문에, 많은 응용 분야에 적합한 내용을 수용할 수 있다. 그러나, 비디오 데이터 부분에서의 시간 관계의 표현 및 비디오 데

이터의 편집 부분 등의 구체적인 제시가 미흡한 상태이다.

Plateau[14]는 VOD시스템으로 비디오를 저장한 비디오 데이터베이스를 근거리 네트워크나 인터넷을 통해 사용자에게 제공을 목적으로 하고 있다. Plateau 시스템은 사용자들로부터 설문 조사를 실시하여 비디오 일반 정보, 비디오 개인 정보에 대한 서지 데이터(bibliographic data), 비디오 계층 구조에 대한 구조적 데이터(structural data), 내용 기반 검색 정보를 위한 내용 데이터(content data), 주제어(keyword) 및 표현 방식(representational)에 관한 메타데이터로 구분하고 있다. 그러나 Plateau는 내용 기반 메타데이터는 잘 정의되어 있으나 내용 독립적인 메타데이터는 아주 미약하다.

ViMod(Video Data Model)[16]는 뉴스 제작을 위한 메타데이터 모델로서 질의의 특성을 비디오 내용 특성(qualitative-feature), 비디오 하위 특성(raw-feature), 이미지 하위 특성 그리고 메타 특성(meta-feature)으로 구분하였다. 그러나 ViMod는 색인과 모델링에 있어서 비디오의 하위 레벨 특징들에 관한 메타데이터가 잘 정의되어 있으나, 내용 기반 검색을 위한 상위 레벨의 메타데이터는 미흡하다.

VideoSTAR[4, 13]은 비디오 데이터와 메타데이터의 공유와 재사용을 지원하기 위한 범용 비디오 데이터베이스 시스템이다. VideoSTAR는 미디어 데이터를 제어하기 위한 미디어 관련 데이터(media-specific data), 비디오 문서와 미디어 데이터간의 관계를 정의한 데이터인 구성 데이터(compositional data), 비디오 문서의 버전에 대한 버전 데이터(version data), 비디오 문서의 전체적인 묘사한 서지 데이터, 비디오의 구조를 다룬 구조적 데이터, 내용에 관한 설명한 주석 데이터(content annotation), 비디오, 오디오 데이터에서 자동적으로 추출되는 특징인 내용 특징 데이터(content feature data), 제목에 대한 제목 주석(topic annotations), 추가적인 정보를 기록하는 보충 정보(supplementary information)에 관한 메타 데이터를 제시하고 있다. VideoSTAR는 비디오 데이터의 공유와 재사용성을 위한 일반적인 데이터 모델링을 제공하였으나, 하위 레벨 특징을 갖는 메타데이터가 고려되지 않고 있다.

AMOS에서는 메타데이터들을 타입별로 나누어 메타데이터로 저장하고 있다[12]. 이는 비디오 데이터의 기록 요소, 형태, 크기, 기간(duration), 데이터 속도와 같은 요구되는 자원의 조건을 지정하는 데이터와 지터(jitter), 평균 지연, 속도 비율, 신뢰도와 같은 질적 요소들, 제목, 저자, 날짜, 소스와 같은 메타 속성들로 구분하고 있다. 내용 속성에는 미디어 객체, 미디어 객체의 속성, 주제, 내용 분류, 내용 설명과 같은 미디어 내용들 및 비디오 데이터의 물리적 위치와 내력과 같은 물리적 데이터를 저장하고 있다. 그러나 이들은 시간 및 공간을 지원하지 못하며, 제한된 영역의 메타데이터만을 제시하고 있으며, 편집에 대한 구체적인 메타데이터를 제시하지 않았다.

3. DC 모델을 확장한 비디오 메타데이터 설계

비디오 데이터는 특성상 많은 편집 단계를 걸쳐 하나의 완전한 비디오가 생성된다. 편집중인 하나의 씬의 총 길이는 씬을 구성하고 있는 모든 샷의 길이의 합과 다를 수 있고, 편집 중에는 비디오의 씬의 순서나 하나의 씬 안에서 샷의 순서가 변경되기도 한다. 이와 같은 샷과 샷들의 관계, 씬과 씬들의 관계 등을 표현하기 위하여 다음과 같이 정의한다.

[정의 1] Dummy shot

의미는 전달하지 않지만 일정 시간을 소비하는 샷을 Dummy shot으로 정의한다.

이는 의미 전달이 목적인 일반 샷과 구별된다. 즉 비디오 편집을 위하여 일시적으로 필요한 시간 간격을 유지할 목적으로 생성되는 샷을 Dummy shot이라 정의한다.

[정의 2] 비디오 데이터의 편집 정보를 편집 목록에 기재하기 위한 우선 순서를 다음과 같이 정의한다.

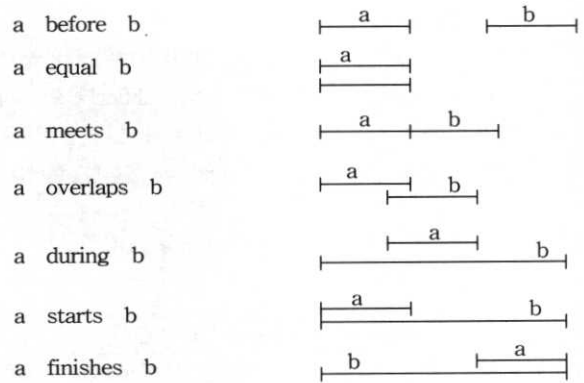
- 시작 시간 우선 원칙 - n개의 임의의 시간 관계를 가진 샷들 가운데 시작 시간이 가장 빠른 샷이 가장 먼저 기재된다.
- 끝 시간 우선 원칙 - 시작 시간이 같은 샷들 중에서는, 끝나는 시간이 빠른 샷이 우선 기재된다.
- 무순위 우선 원칙 - 시작 및 끝나는 시간이 모두 일치되는 샷들은, 임의의 샷이 우선 기재된다.

이는 샷들간의 n-ary 시간 관계를 비디오 편집 목록에 기록할 때에 혼란을 막고 샷들간의 시간 관계 해석이 명확해진다. 정의에 의해 n-ary 시간 관계의 샷들을 하나의 시간축(time-line)상의 이진 관계로 변환하여 가장 근접한 두 개의 샷 사이의 관계로 분리하여 기록된다.

[정의 3] 샷 들간의 시간 관계 연산을 표현하기 위하여 Allen이 정의한 13개의 시간 관계를 정의 1, 2에 의해 <표 2>와 같이 6개의 관계로 다시 정의하여 요약한다.

<표 2> 시간 관계 연산자의 변환

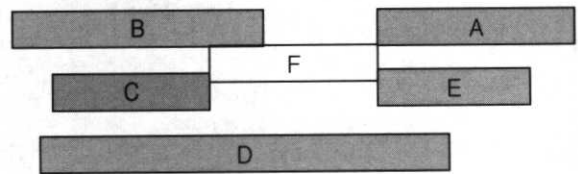
관 계	변환된 관계	적 용 내 용
A before B	A meets B	시작 시간 우선 원칙 Dummy shot도입 A meet Dummy와 Dummy meet B의 관계로 변환
A meets B		시작 시간 우선 원칙으로 역관계를 배제
A overlaps B	A overlaps B	시작 시간 우선 원칙으로 역관계를 배제
A during B	B during A	시작 시간 우선 원칙으로 역관계를 배제
A starts B	A starts B	끝 시간 우선 원칙으로 역관계 배제
A finishes B	B finishes A	시작 시간 우선 원칙으로 역관계를 배제
A equals B	A equals B	무순위 우선 원칙



(그림 1) 시간 관계 연산자

(그림 1)에서 보여 주는 바와 같이 제시한 알렌의 시간 관계[17]는 equal을 제외하고는 모두 역관계가 존재하여 13가지의 시간 관계가 있다. 이를 <표 2>에 제시한 6가지의 관계로 모두 표현되어 비디오 편집 목록에 일관성이 유지된다.

(예 1) 아래의 (그림 2)를 고려하여 비디오 샷의 관계를 표현하자



(그림 2) 비디오 샷들의 시간 관계성의 표현

(그림 2)에서 비디오 데이터의 샷이 삽입되는 순서는 A, B, C, D, E의 순서이다. 이들의 관계는 n-ary 관계를 가지고 있다. 그러나 [정의 2]에 의해 시작 시간 우선 원칙으로 B, D, C, A, E의 순서가 된다. 그러므로 이들은 각각 B, D의 관계, D, C의 관계, C, A의 관계로 정의 될 수 있다. 여기에 Dummy 샷을 정의함으로써, 이는 다시 C, F의 관계로 표시 될 수 있고, 또한 정의에 의해 끝 시간 우선 원칙에 의해 A와 E는 E가 앞서게 된다. 즉 F, E의 관계 및 E, A의 관계로 정의 될 수 있으므로 모든 시간 관계는 6가지 타입으로 단순화 할 수 있다.

이렇게 두 샷간의 관계로 모든 관계를 단순화함으로써 편집 목록에 기재되는 순서 및 이들간의 시간 관계 연산도 단순화된다.

비디오 데이터는 계층적 구조를 가지고 있다. 즉 하나의 완전한 "Video"는 여러 시퀀스(sequence)로 구성되고, 이들 시퀀스는 다시 씬(scene)들로 구성되며, 각각의 씬은 샷들로 구성되어 있으며, 샷은 프레임들의 모임으로 구성된다. 또한 하나의 프레임 안에는 여러 객체들과 배우들이 존재한다. 비디오 데이터를 나타내기 위해서는 이런 구조를 모두 표현해야 한다. 또한 비디오 메타데이터는 내용 기반 검

색을 지원하기 위하여 비디오 정보들을 적절히 구조화하고, 분류되어야 하며, 데이터 구조적 특징, 시공간적인 구조 등을 잘 표현할 수 있어야 한다. 이런 요건들을 표준화에 반영하고 있는 것이 DC 모델이며, 이를 근간으로 하여 비디오 데이터에 필요한 메타데이터를 추가하여, 모든 분야에 적용되는 일반적인 비디오 메타데이터를 설계한다.

[정의 1]~[정의 3]과 비디오 메타데이터의 조건을 고려하여 DC 모델에서 제시하고 있는 15개 항목들을 기본으로 계층적 구조를 갖는 비디오 데이터가 가질 수 있는 정보를 <표 3>과 같이 적용한다.

종류(Type)로는 비디오가 되며 비디오의 하위 층으로 시퀀스가, 시퀀스의 하위 층으로 씬, 씬의 하위 층으로 샷, 샷의 하위 층으로 프레임, 프레임의 하위 층으로 객체가 있는 계층적 구조가 된다. 즉 하나의 프레임을 표현하기 위해서는 "Video.Sequence.Scene.Shot.Frame"으로 계층적 구조로 표현된다.

<표 3> 비디오 메타데이터의 분류

요 소	video	sequence	scene	shot	frame	object
제목(title)	○					
제작자(creator)	○					
주제(subject)	○	○				
출판자(publisher)	○					
형태(type)	○	○	○	○	○	○
주석(description)	○	○	○	○	○	○
날자(date)	○					
식별자(identifier)	○	○	○	○	○	○
출처(source)	○					
공헌자(contributor)	○	○				
형식(format)	○	○	○	○	○	
언어(language)	○					
권한(rights)	○					
관계(relation)	○	○	○	○	○	
적용범위(coverage)		○	○	○	○	

비디오를 구성하고 있는 하위 층들과의 관계를 표현하기 위해서는 관계(relation)를 이용한다. 이들의 관계는 "Video HasPart Sequences"로 표현되며, 역의 관계는 "Sequence IsPartOf Video"가 된다. 이들의 관계처럼 하위 계층의 관계도 모두 이처럼 표현된다.

형식은 데이터를 상연, 조작, 필요한 소프트웨어의 구분하기 위한 정보를 충분히 제공하기 위한 것이므로, 비디오의 일반적인 정보 중에서 전체의 총상연시간(length), 화면의 크기(width, height), 색상수(color), 형태, 상연 속도(framerate) 등을 그 하위 계층으로 수용하게 한다. 이들 중에 형태는 다시 제한자를 이용하여 MPEG, JPEG, AVI 등으로 동영상 지원하는 형태들을 수용한다.

샷의 경우에 전체 길이는 형식의 길이로, 샷의 시작과 끝은

적용 범위의 하위층 요소인 Temporal.Start와 Temporal.End로 표현되고, 이들이 촬영된 장소 역시 Spatial.Place로 표현된다. 프레임안에 있는 객체의 위치는 상대적인 위치를 표현하므로 이는 주석에서 다룬다.

주석은 데이터에 관한 내용, 배경과 목적 등을 자세히 설명하기 위한 부분이며, 또한 시각적 부분과 응용의 기능에 대한 구체적인 내용까지 포함되므로, 이의 하위 계층으로 비디오, 시퀀스, 씬, 샷, 프레임, 객체를 위한 메타데이터를 분류하여 설계하였고 이를 정리한 것을 <표 4>에 나타냈다. <표 4>에서 비디오와 시퀀스는 DC 모델의 15개 항목의 모두 기술되므로 표에 나타내지 않았고, 주석의 편집 목록(editlist) 부분은 씬부분과 샷부분에서 기술한다. 표의 내용에서 카메라 거리에 관한 부분의 "close-up, medium shot, long shot"과 같은 부분은 DC 모델에서 통제어(controlled vocabulary)를 중심으로 기술하였다. 향후 비디오 기술의 발전에 의해 많은 기술 용어가 새로 만들어질 것이며, 이들 모두가 수용 가능하다.

비디오 데이터는 제작 과정상의 편집이 차지하는 중요성이 매우 크다. 이를 나타내기 위하여, 씬의 편집 목록에서는 하나의 씬을 구성하는 샷들의 순서를 기술한다. 이 기술에 참여하는 정보는 각각의 샷의 식별자이며 이는 모든 샷들을 통해 유일성을 가지므로 중복되지 않고, SMPTE나 프레임번호는 해당 샷의 위치를 나타내며 리스트 구조를 가지므로 순서 및 중간의 삭제, 변경, 추가 등이 가능하다. 또한 하나의 씬을 구성하는 샷들과의 관계는 관계에서 기술되며, 이들의 시간적인 관계는 적용 범위 소의 하위층 요소인 Temporal.Start와 Temporal.End로 기술된다. 또한 샷들과의 관계는 모두 하나의 시간축 상에 이루어진다. 시작과 끝을 기술하는 정보는 초, 프레임 번호, 또는 SMPTE 단위로 기술된다. SMPTE는 영화의 촬영이나 편집 상에서 사용하는 표준화된 시간 단위이며, "03 : 25 : 31 ; 18"과 같은 단위로 한편의 비디오의 시작에서 끝나기까지의 연속된 시간으로 표현된다. 즉 편의에 따라 단위는 달라질 수 있지만, 표현하려는 시간상의 전후 또는 겹침 등이 모두 표현될 수 있다. 이로써 편집 상에 필요한 모든 정보가 수용되어, 각종 멀티미디어 개발 도구들도 동일한 메타데이터를 사용하여 비디오 데이터를 사용할 수 있다. 또한 비디오 데이터에 관한 정보는 일부 프레임 또는 객체들의 정보를 보는 것도 중요하지만, 일부 내용을 상연해 보는 것이 더 중요하다. 즉 어떤 정보들을 보는 관점과는 달리 객체의 움직임에 대한 정보를 인식해야 하기 때문이다. 제시한 메타데이터를 이런 목적을 위한 메타데이터를 모두 수용하였다.

이와같은 비디오 메타데이터 중에서 DC 모델의 15개 요소는 <표 3>에서 보여 주었다. 이 15개 요소 중 주석(description)에 해당하는 요소를 확장하여 <표 4>와 같이 정리하여 제시하였다.

<표 4> 비디오 메타데이터의 Description 하위 계층

Type 계층	주석(description)	기술되는 형태 또는 제한자	
Scene	내용(content)	문 장	
	대본(script)	문 장	
	편집 목록(editlist)	Identifier/ SMPTE, frame번호	
	배경(locale)	문 장	
	배역(cast)	문 장	
	사물들(objects)	문 장	
	대표키 프레임(repkey-frame)	JPEG, GIF, MPEG, BMP 등	
Shot	내용(content)	문 장	
	키프레임(key-frame)	JPEG, GIF, MPEG, BMP 등	
	카메라(camera)	거리(distance)	close-up, close shot, medium shot, long shot
		각도(angle)	low, high, eye-level, overhead
		움직임(motion)	dolly forward, dolly back, truck left, truck right, pan left, pan right, tilt up, tilt down, zoom in, zoom out, stationary,
	조명(Light)	문 장	
	편집 목록(editlist)	insertlist	Identifier/SMPTE, frame 번호
		updatelist	Identifier/SMPTE, frame 번호
		deletelist	Identifier/SMPTE, frame 번호
	도입 효과(open transition)	효과	cut, fade, wipe, dissolve
		끝	프레임 번호, 초, SMPTE
마감 효과(close transition)	효과	cut, fade, wipe, dissolve	
	시작	프레임 번호, 초, SMPTE	
프레임 Frame	내용(content)	문 장	
	화면내용(image)	JPEG, GIF, MPEG, BMP 등	
	위치(position)	초, frame 번호, SMPTE	
	색분포(color)	Histogram 값, 문장	
객체 Object	내용(content)	문 장	
	위치(position)	좌표(x, y)	
	모양(shape)	Polygon	
	궤적(trajjectory)	Line	
	속도(speed)	Pixels/frame	
	색상(color)	Histogram 값, 문장	
	질감(texture)	Tamura, SAR feature, Vector	
	부피(volume)	3D polygon	

4. 적용 예 및 결과 분석

4.1 적용 시나리오

설계된 메타데이터를 이용하여 광고에 적용한 시나리오는 다음과 같다.

- 일반적인 부분은 생략하며, 제안된 내용을 중심으로 기술한다.
- 광고는 하나의 시퀀스에 6개의 씬이 있다.
- 씬은 총 15개의 샷으로 구성된 574 프레임인 19초 분량이다.

<표 5>에서는 광고의 시퀀스, 씬, 샷들의 구성 관계와 카메라 동작 및 내용을 보였고, <표 7>에서는 이 광고에 대한 메타데이터를 보였다. 이는 일반적인 기본 DC 모델 부분은 생략하였고, format 부분에서 비디오에 대한 부분을 확장한 것을 나타내었다.

<표 5> 한 광고에 적용 예

시퀀스	씬	샷	카메라	내용
1	1	1	medium	초록색 화초 사이로 보이는 여자가 정면을 보고 앉아 선물을 보고 있다,
		2	medium	초록색 화초 사이로 보이는 여자가 오른쪽을 보고 앉아 선물을 보고 있다,
		3	close	선물에 기뻐하는 모습을 가까이 클로즈업
	2	1	close	선물이 화장품을 확인하는 모습
		1	close	화장품의 내용을 간접 설명(하얀 진흙에 미네랄)
	3	2	close-up	성분이 합쳐진 모습을 설명(클로즈업)
		1	close-up	세안을 위해 용기에서 내용물 나눔
	4	2	close	얼굴에 바름(양분)
		3	close-up	얼굴에 바름(이마와 턱)
		4	close-up	물로 세안함
		1	close-up	세안 후 얼굴 상태에 만족한 모습
	5	2	close	만족해 있는데 전화벨이 울림
		3	close-up	전화 받는 모습을 클로즈업한 장면
		4	close-up	약간 멀어진 모습으로 화장품을 바라보는 듯한 자세
	6	1	close	내용물 및 이름을 보여줌

<표 6> video 타입에 대한 비디오 메타데이터

요 소	video
제목(title)	에경 포인트
제작자(creator)	null
주제(subject)	화이트 머드 폼 화장품 광고
출판자(publisher)	null
종류(type)	Video
주석(descriptor)	
날짜(date)	null
식별자(identifier)	video1
출처(source)	null
기여자(contributor)	null
형식(format)	길이(length)
	19 초
언어(language)	ko
권한(rights)	null
관계(relation)	구성(haspart)
	1 Sequence

시퀀스 중에 씬3의 샷1, 그 중에 한 프레임을 중심으로 기술되는 내용 및 관계 등을 설명한다. 이 경우는 한 씬을 이루는 샷들이 이미 편집이 완료된 상태에서 기술되므로 적용 범위의 Temporal.Start와 Temporal.End를 프레임 번호로 기술한다. 이 표에서 이 샷이 시작되는 프레임을 나타내는 내용을 계층적 구조로 나타내 보면 아래의 식 (1)로 표현된다.

$$\text{Video1.Sequence1.1.Scene1.1.3.Shot1.1.3.1.Coverage.Temporal.Start} = \text{"191 Frame"} \quad (1)$$

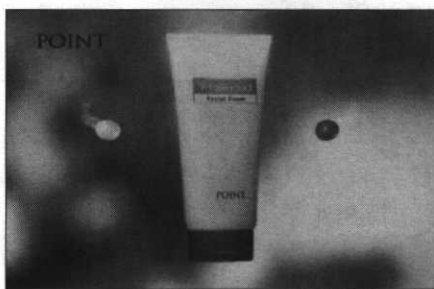
이 샷은, Video1의 첫 번째 시퀀스인 Sequence1.1 중에서 세 번째 씬인 Scene1.1.3, 또 이 씬의 첫 번째 샷임을 나타낸다. 이 샷의 시공간적인 내용을 나타내는 적용 범위의 시간 부분인 Temporal의 시작이 191 프레임을 나타낸다.

<표 6>, <표 7>에서 굵은 글씨체로 나타낸 부분은 응용과 무관하게 모든 비디오 데이터에서 필요한 메타데이터로 분류하였다. 응용에 무관한 부분을 시스템 관리 메타데이터로, 응용에 연관된 부분을 사용자 관리 메타데이터로 분류하였다. 시스템 관리 메타데이터는 모든 종류의 비디오 데이

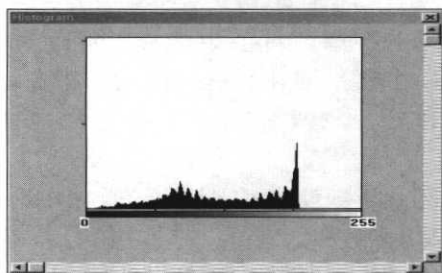
터에 적용되며 경우에 따라서 불필요한 부분은 사용하지 않을 수도 있다. 사용자 관리 메타데이터는 DC 모델의 제안과 같이 항상 확장이 가능하므로 응용 분야에 따라서 확장하여 사용할 수 있다.

<표 7> Shot1.1.3.1에 대한 비디오 메타데이터

요 소		내 용	
종류(type)		Shot	
식별자(Identifier)		Shot1.1.3.1	
주석 (description)	키프레임(key-frame)		
	Frame210		
	내용(content)		
	화장품의 내용을 간접 설명 (하얀 진흙에 미네랄)		
	카메라 (camera)	거리(distance)	medium shot
		각도(angle)	eye-level
		움직임(motion)	stationary
	조명(light)		자연광
	편집 목록 (editlist)	삽입(insertlist)	31, Frame0.0.0.0.142
		변경(updatelist)	45, Frame0.0.0.0.143
삭제(deletelist)		90,00/11/21 : 12 : 23 : 56 : 123	
도입 효과 (open transition)	효과(effect)	fade-in	
	마지막 프레임 (ending frame)	10	
마감 효과 (close transition)	효과(effect)	fade-out	
	시작 프레임 (starting frame)	120	
형식 (format)	크기 (size)	높이(height)	352 pixels
		너비(width)	240 pixels
	색상수(colors)		24 bits
	종류(type)		MPEG
	상연 속도(framerate)		30 frames/sec
해상도(resolution)		72 dpi	
관계 (relation)	haspart		130 Frames
적용범위 (coverage)	시간 (temporal)	시작(start)	00 : 27 : 12 : 00
		종료(stop)	00 : 27 : 16 : 10



(그림 3) Shot1.1.3.1의 키-프레임



(그림 4) 키 프레임의 칼라 히스토그램

(그림 3)은 <표 5>에서 설명한 샷의 키 프레임이며 (그림 4)는 칼라 히스토그램이다. 키 프레임에 대한 메타데이터는 내용은 “중앙에 포인트 화이트 머드 폼이 있고 양쪽에는 진흙과 미네랄이 있다”이며, 화면 내용에 해당하는 것은 (그림 3)에 제시되어 있고, 위치는 프레임 번호 210번, 색분포는 (그림 4)에 제시되어 있다. 이와 같이 표현되는 항목은 응용 분야에 따라서 필요한 항목을 기술하면 된다.

비디오 데이터에 대한 모든 조작은 메타 데이터에 의해 관리되고, 비디오 데이터에 대한 직접적인 연산은 기본적인 정보를 추출하기 위한 연산만 적용되며, 대부분의 연산은 메타 데이터를 통하여 이루어진다. 비디오 데이터에 대한 조작은 메타 데이터에 기록되고 생성 연산에 의해 반영되며 키 프레임에 대한 색인은 데이터베이스에 기록되고 관리된다.

<표 8> 한미르 광고 내용 및 구성

시퀀스	번	샷	프레임번호	내 용 설 명	
1	1	0~87		Hanmir 30a, 검은 배경, 흰 글씨	
	2	88~137		핑클의 4인이 노래를 부르고 있다.	
	3	138~187		노래를 부르고 있다 (다른 각도의 촬영)	
	4	188~263		컴퓨터 보이고, 팬클럽 검색하기, 검색엔진에서 “핑클”을 검색	
	5	264~283		검색엔진에서 검색이 시작된다	
	6	284~337		검색 결과는 “서핑클럽”이 검색되었다.	
	7	338~377		잘못된 검색엔진 선택은 힘들고 어렵다는 것을 서핑을 하는 모습으로 보임, 실내에서 서핑시작	
	8	378~425		한미르로 안찾았지?	
	9	426~439		“핑클”이 아닌 “서핑클럽”이 검색되었음을 강조	
	10	440~465		서핑이 계속된다(혼돈이 시작)	
	2	11	466~495		구성원들이 모두 차례로 출발함
	12	496~513			
	13	514~515			짧은 샷으로 빠른 진행을 보임
	14	516~519			
3	15	520~566		zoom-out, 실내가 아닌 바다 밖으로 나옴 (정보의 바다)	
	16	567~582		바다에 서핑함	
	17	583~590		한명이 바다에 빠짐	
	18	591~602		또 한명 빠짐	
	19	603~606		또 한명 빠짐	
	20	607~630		또 한명 빠짐	
2	4	21	631~706	바닷가로 겨우 나옴, 기진 맥진, 꼭 한미르로 사용할 것을 강조	
	5	22	707~738	먼 화면으로 네명이 서핑중, 한 명씩 찍음	
	6	23	739~796	한미르 화면 등장, 핑클 검색	
	7	24	797~818	결과 화면 나옴, 한 명씩 보임, 검색메뉴 한 개씩 보임 메뉴: 전화번호, 지도검색 설명	
		25	819~864	메뉴: 일본웹 검색 설명	
	26	865~920	메뉴: 부동산.증권정보 설명		
	27	921~958	메뉴: 쇼핑 설명		
	28	959~1004	모두 보임, 마무리함		

편집은 비디오 데이터의 특성중의 하나이다. 영상물은 촬영도 중요하지만, 이를 편집을 통하여 완성하는 것이다. 이런 비디오 편집 정보는 메타 데이터를 통하여 유지하고 관리된다. 편집중인 비디오의 모든 정보는 메타 데이터에 저장되어 관리되며, 새로운 생성에 의해 모든 메타 데이터 정보를 반영할 때까지 유지된다.

외부 파일로부터 입력된다면, 입력 데이터의 기본적인 구성 요소들은 파일로부터 추출되어 자동으로 입력된다. 샷에 대한 입력 데이터 중에서 format에 관한 내용은 화면의 크기, 색상수, 파일 타입, 상영 속도는 모두 시스템에 의해 자동 입력된다. 시스템 관리 데이터는 반드시 입력되어야 하고, 사용자 관리 메타 데이터는 임의로 입력 가능하다.

예로 선택한 광고는 MPEG로 되어 있으며, 2개의 시퀀스, 7개의 씬, 27개의 샷으로 구성되어 있다. <표 8>에는 이들의 계층적 구조와 샷을 구성하는 프레임 번호, 내용을 개략적으로 표현한다.

4.2 예

앞 절에서 기술한 시나리오를 기반으로 다음과 같은 질의어를 통하여 비디오 데이터를 입력하고 검색 할 수 있다.

【예 1】 “c:\MyData\상업광고\한미르30A.mpeg”에 있는 광고물인 “한미르 30A”편을 입력하라. 이는 2개의 시퀀스를 가지고 있다.

```
insert into video
set ( title = "한미르30A", source = "c : \MyData\상업광고\한미르
30A.mpeg", relation.haspart = 2 )
file "c : \MyData\상업광고\한미르30A.mpeg"
```

이는 처음 비디오형의 메타 데이터를 생성시키며, from절로부터 기본 정보인 상영 길이를 추출하여 입력하게 된다. 기타 정보는 반자동, 또는 수동으로 입력할 수 있다. 식별자는 비디오형의 처음이므로 Video1로 자동 생성된다. 또한 2개의 시퀀스 정보를 관계에 삽입한다.

【예 2】 “한미르 30A”의 첫 번째 시퀀스는 3개의 씬으로 되어 있으며, 프레임 0번에서 630까지로 되어 있다.

```
insert into sequence
set ( format.length = get_SMPTE( 0, 630 ), relation.haspart = 3 )
file "c : \MyData\상업광고\한미르30A.mpeg"
where video.title = "한미르 30A" and sequence = 1
```

이는 이미 생성된 비디오에 1번째 시퀀스 정보를 삽입하고 있다. sequence1.1은 3개의 씬을 가진 정보와 전체 길이를 get_SMPTE 함수에 의해 계산되어 입력된다. 630 프레임은 “00 : 00 : 21 : 000”의 SMPTE 값을 갖는다. 이는 21초를 나

타내며, 초 이하의 단위는 프레임 수를 나타낸다.

【예 3】 “한미르 30A”의 첫 번째 시퀀스중 2번째 씬은, 프레임 378번에서 519까지이며, 이 씬은 7개의 샷으로 구성되어 있다.

```
insert into scene
set ( Repframe = extract_rep_keyframe ( 378,519 ), relation.haspart
= 7, format.length = get_SMPTE( 378, 519 ))
file "c : \MyData\상업광고\한미르30A.mpeg"
where video.title = "한미르 30A" and sequence = 1, scene = 2
```

이는 scene1.1.2를 삽입하는 과정이다. 아직 편집 목록은 null 값을 가지며, 동일한 원본 파일을 사용하므로 Coverage.Temporal.start 와 Coverage.Temporal.stop 는 SMPTE로 계산된 값에 의해 지정된다. 만일 독립된 다른 파일이라면, 해당 파일의 SMPTE 값이 입력되며, 식별자는 scene0.0.2가 된다. 또한 Source에 외부 파일이 등록된다.

【예 4】 외부 파일 “\video\shot\abc.mpg”를 scene1.1.2의 2번째 샷으로 입력하라. 효과는 처음 5 프레임은 fade in으로 마지막 프레임 10개는 fade out으로 하라.

```
insert into shot
set( keyframe = extract_keyframe ( "\video\shot\abc.mpg" ),
opentransition.effect = "fade in", opentransition.ending = 5,
closetransition.effect = "fade out",
closetransition.start = Video_Length ( "\video\shot\abc. mpg" )-10,
format.size = Video_Size ( "\video\shot\abc.mpg" )
format.colors = Video_Colors ( "\video\shot\abc.mpg" )
format.framerate = Frame_Rate ( "\video\shot\abc.mpg" ) source =
( "\video\shot\abc.mpg" )
file "\video\shot\abc.mpg"
where video.title = "한미르 30A" and sequence = 1, scene = 2,
shot = 2
```

이 예제는 많은 연산이 사용되었다. 파일 “abc.mpg”는 지금까지 한번도 참조되지 않았기 때문에 모든 정보를 새로이 추출하여 입력되어야 하고, 기존의 씬과는 독립된 파일이므로 원본(Sorce)에 관한 기록이 필요하다. 다만 소속된 씬이 있으므로 샷의 식별자는 shot1.1.2.2로 된다.

비디오 데이터에 대한 검색은 일반 문자형 데이터를 검색하는 것과는 다르게 비디오의 구성을 알 수 있어야 하며, 검색할 수 있는 항목이 어떤 것이지도 알아야만 한다. 그러므로 단순 검색뿐만 아니라, 단계적인 질의를 통하여 검색할 수 있는 다단계 전략을 사용한다. 각 단계를 하나씩 설명하면,

- 1) 정확한 질의를 하면, 해당되는 항목의 결과를 사용자에게 보여준다.
- 2) 무엇에 대한 검색일지 모른다면, 비디오 메타 데이터의 계층적 구조를 보여 주어, 질의를 할 수 있도록 유도한다.

- 3) 적절한 질의를 재구성할 수 있도록 한다.
- 4) 검색 결과가 샷이나 클립, 또는 그 이상이면 브라우징과 상영을 연결한다.
- 5) 질의는 분류하여 저장, 향후 질의의 유형으로 이용한다.

【예 5】 춘향전의 1번 시퀀스의 3 번째 씬을 모두 보여라

```
select *
from m in video
where sequence = 1 and scene = 3 and title = "춘향전"
display//상영, 생략되면 한 화면 즉 키 프레임 또는 대표키 프레임
```

이 질의는 비디오 “춘향전”에서 1번째 시퀀스의 3번째 씬을 찾아서, 씬 전체를 상영해 준다. 상영의 방법은 지정하지 않았으므로, 일반적인 방법인 한 화면에 모든 정보를 상영한다.

【예 6】 춘향전의 어떤 구조를 가지고 있는지 보여라

```
select m.structure
from m in video
where title = "춘향전"
```

이 질의는 “춘향전”에 대한 메타 데이터를 참조하여 시퀀스, 씬, 샷의 수를 구성 관계로 나타낼 것이다. 질의 결과를 이용하여 비디오의 하위 계층에 대한 상세한 질의를 유도할 수 있다.

【예 7】 지금까지 편집된 비디오 “가을비”를 이용하여, 완전한 한편의 비디오 “가을비 2”를 생성하라.

```
select CreateVideo(m, "가을비 2")
from m in video
```

이 질의에 의하여, 모든 편집 정보를 반영한 한편의 비디오가 생성된다. 메타 데이터의 정보는 모두 사용되며, 원본 비디오 데이터에 이들 정보를 반영한다. 생성된 “가을비 2”의 메타 데이터는 편집 정보에 관련된 정보를 제외한 모든 메타 데이터가 원본의 데이터와 동일한 내용을 갖는다.

【예 8】 강수철 감독이 만든 영화중 60분 이하인 영화 제목을 보여라.

```
Select m.title
from m in video
where m.format.length < 3600 and m.creator = "강수철"
```

이 질의는 시간 연산을 통하여, 강수철 감독이 제작한 길이가 60분 이하인 모든 영화 제목을 나열한다.

【예 9】 영화 가을비 중에서 처음 2분간을 시간축 상으로 상영하라.

```
Select extract(m, 1, 120)
from m in video
where m.title = "가을비"
display time-line
```

이 질의는 비디오 “가을비”의 처음 2분간을 편집 연산자를 이용하여 추출하고, 결과를 상영 연산에 의해 시간축 상에서 프레임들이 나열된 상태로 상영한다. 이는 기존의 단순 시간 질의어에서, 상영을 지원하는 질의어로 확장된다.

4.3 결과 분석

이 논문에서 제안한 모델은 정의된 비디오 데이터형을 근거로 하여 샷, 씬, 시퀀스 및 비디오의 메타 데이터를 구성한 모델로, 구조적 비디오 데이터를 제시하였다. 또한 비디오 데이터의 구조적 정보와 조작 정보도 함께 관리하였다. 이는 비디오 데이터의 독립성을 유지할 수 있으며, 비디오 객체의 삽입, 삭제, 갱신이 자유롭고, 데이터베이스가 제공하는 데이터베이스에 의한 해석, 공유, 문제 복구 등의 모든 이점을 이용할 수 있다.

시간 속성은 비디오 데이터의 중요한 측면이다. VideoSTAR은 질의를 구현할 때, 시간적 관계들을 이용하는데 필요한 시간적 데이터베이스에 기반한 대수식을 제공해 준다. OVID, VideoSTAR는 뉴스 비디오를 대상으로 하고 있으나, 시간 속성은 기본적 시간 관계인 시작 시간과 종료 시간인 간격 관계에 한정하고 있다. 그러나 비디오의 시간적 특성을 고려하여 시점 관계의 시간속성을 지원해야 한다.

이 논문에서는 시간의 두 가지 측면 즉 시점 및 간격 관계를 모두 지원하며, 비디오 데이터의 특성을 고려하여 13개의 시간 연산자를 6개로 변환하여 적용하였고, 변환된 시간 관계를 이용하여 비디오 편집을 지원하는 등 시간 속성의 효율성을 높였다.

VideoSTAR의 내용 기반 메타 데이터 모델에서 주석을 위한 내용의 구성은 primary, basic, secondary의 층화 단계로 표현을 하고 있으나, 이런 분류 표현의 단점은 의미 부여가 복잡하고, 내용에 대한 다양한 접근이 미흡하다는 점이다. 또한 메타 데이터의 형태가 person, location, event 등의 세 가지 형태로 고정되어 비디오 데이터의 내용을 설명하는데 표현의 한계가 있다. 동일한 비디오 스트림을 설명하는 메타 데이터일지라도 하나의 스키마를 이용하여 설명할 수 없다. 즉 각각의 메타 데이터를 분리하여 메타 데이터가 분산되기 때문에 통합적인 메타 데이터 관리가 불가능하다. 또한 뉴스에 한정된 메타 데이터를 제안하고 있다. OVID모델은 객체 지향 구조를 기반으로 한 비디오 객체 모델로서 스키마 구조를 갖지 않고 연산식 형태로 구성되었다. 이 모델은 임의의 속성에 대한 비디오 객체의 추가가 가능하나 모델들은 구조의 확장성의 측면에서 상대적으로 또 다른

수식을 제안해야 하는 문제를 안고 있다.

또한 의미적인 접근 단위가 되는 비디오 객체는 각 객체가 생성된 이후에는 상호 독립적으로 유지되므로 비디오 객체간에 동적으로 속성이 상속될 수 없다. 따라서 비디오 객체간에 중첩되는 부분이라도 사용자에게 의해 독립적인 비디오 객체로 정의되어 의미 기술이 되지 않은 경우 접근이 불가능하다. 또한 사용자마다 다양하게 정의할 수 있는 속성 이름의 일관된 관리에 대한 언급 없이, 모든 속성 값을 하나의 속성 값 트리로 관리하여 많은 속성 값이 정의되면 속성 값 트리가 너무 커져 사용자의 속성 값 선택을 효율적으로 지원할 수 없는 문제가 따른다. 즉 대부분의 모델은 메타데이터를 일부 응용 분야에 치중하여 구성하고 있어 확장이 곤란하고, 일부 모델은 구성의 일관성이 없이 임의의 속성을 추가하는 등 구조가 확정되어 있지 못하다.

이 논문에서는 웹에서 멀티미디어를 표현하고 검색하고자 표준화가 진행되고 있는 더블린 코어 모델을 확장하여 비디오 메타 데이터를 설계하였다. 또한 메타 데이터의 구성을 시스템 관리 부분과 사용자가 정의하여 추가할 수 있는 부분을 분리하여 설계하였으며 응용 분야에 독립적 사용이 가능하며 유연성이 뛰어나다.

QBIC은 비디오를 샷으로 분할하고 각 샷에 대한 대표 프레임 추출하여 각종 내용 기반 검색을 위한 메타 데이터의 구성이 잘되어 있지만 비디오 데이터의 조작에 관한 언급은 없다. OVID는 구간 상속에 의한 공유는 잘 지원하지만 역시 비디오 데이터에 대한 조작은 지원하지 않는다. VideoSTAR는 컨텍스트 개념을 도입하여 공유가 매우 용이하며, 비디오 데이터에 decompose에 의한 분해등의 일부 조작 기능을 지원한다.

그러나 이 논문에서 제안한 모델은 비디오 데이터를 구성하고 있는 임의의 단위로 삽입, 삭제, 갱신과 비디오 편집을 메타 데이터에 의하여 지원한다.

기존의 OVID, VideoSTAR, QBIC은 단순한 비디오 데이터의 저장, 질의 및 브라우징의 관점에서 질의어를 지원하고 있다. 이들 모델에서 제안되고 있는 질의에 관한 대수식은 단편적인 질의나 실험적 브라우징을 위해 유용할 수 있으나 다양한 응용지원을 위해서는 일반화된 형식을 갖는 것이 바람직하다. VideoSTAR에서는 부분적 편집만이 가능하여 비디오 편집에는 활용될 수 없어 비디오 데이터베이스의 효율적 활용은 제시하고 있지 못하다.

그러나 이 논문에서는 질의어를 확장하여 비디오의 편집 및 상영을 지원하였다. 이는 비디오 데이터를 이용한 멀티미디어 저작 도구와 같은 응용 시스템의 개발과 비디오 데이터베이스에 의한 비디오 데이터의 공유가 가능하며, 제안된 모델의 효율적인 활용을 기대할 수 있다.

제시한 예와 같이 제안한 메타데이터는 DC 모델을 확장한

모델로 계층적 구조를 가지고 있으며, 표준화된 기술 방법에 의해 확장되어, 어떤 응용 분야에서도 동일한 구조의 메타데이터로 표현되어 데이터베이스에 수용하여 동일한 방법으로 검색할 수 있는 이점이 있다.

이 연구에서 제안한 모델은 메타데이터를 기존의 분류법과는 다르게 응용 독립적인 메타데이터는 시스템 관리 메타데이터로 응용 종속 메타데이터는 사용자 관리 메타데이터로 분류하였으며, 기존의 메타데이터에서 제시되지 못했던 비디오 편집에 대한 상세한 부분을 제시하였다. 또한 알렌의 13개 시간 관계를 6개로 변환시켜 적용시킴으로서 메타데이터상의 편집 목록을 작성함에 일관성 유지할 수 있도록 제안하고 있다.

지금까지 기술한 제안 모델과 기존 모델을 비교한 내용을 요약하면 <표 9>와 같다.

지금까지의 연구에서 제안된 모델들이 VOD나 뉴스, 스포츠 경기 등과 같은 하나의 특정 응용 분야만을 고려한 메타데이터를 제시하였다. 그러나 이 연구에서는 응용 분야에 무관하게 적용시킬 수 있도록 모든 분야에서 공통적으로 적용되는 시스템 관리 메타데이터와 응용 분야에 밀접한 사용자 관리 메타데이터로 분류하였으며 표준화가 진행되고 있는 DC 모델을 확장하였으며 지금까지의 연구에서 응용 분야에 치우쳤던 비디오 메타데이터를 모든 비디오 데이터에 적용될 수 있도록 일반화 하고자 하였다.

<표 9> 기존 연구와의 비교 분석

행목 \ 모델	AMOS	VideoSTAR	Plateau	DC 모델	제안 모델
비디오 데이터 표현	Continus Long Field	Stream Interval	계층 구조	video, scene 등 구조적	video, scene 등 구조적
시간 지원	지원안됨	시작과 종료 지원	지원안됨	시점 지원	시점, 간격 지원
메타 데이터의 구성	자원의 조건, 절적 요소, 내용 속성 등	분산되어 구조가 통합되어 있지 못함	내용기반 위주 내용 독립적이지 못함	내용, 지적소유, 구현에 관한 15개 요소, 확장성	DC기반 표준안 준수 구조적, 확장 용이 시스템, 사용자 정의로 구분하여 관리함
주석 기능	자유로운 주석	자유로운 주석	일반정보, 개인정보에 한함	요소별 자유로움	응용 분야와 무관 자유로움
비디오 편집	지원안됨	일부 지원	지원안됨	효과부분만 지원	자유로운 편집 (삽입 삭제등)
응용분야	VOD	뉴스분야	VOD	웹, 전자문서	모든 비디오 데이터
장 점	외부 저장 매체의 관리	비디오 데이터의 변경 및 재사용	VOD의 내용 기반 메타데이터	확장이 용이 적용 분야별 확장	자유로운 비디오 편집 시간지원 확장이 용이

5. 결 론

지금까지 연구된 비디오 데이터 모델에서 제안한 메타데이터는 대부분 특정 응용 분야에 한정된 부분만을 주로 다루어 왔다. 이들의 연구에 의하면 동일한 내용의 비디오 데이터를 표현하기 위하여 응용 분야별로 각기 다른 구조와 다른 내용의 메타데이터가 필요하였다. 그러나 응용 분야별로 다른 구조와 내용을 갖는 메타데이터는 비디오 데이터베이스의 관점에서는 서로 다른 구조를 갖는 메타데이터를 지원해야 하는 문제점이 있다.

따라서 이 논문에서는 이러한 문제점을 해결하기 위하여 DC 모델을 기본으로 비디오 데이터를 표현하기 위하여 시간 영역을 확장하고, 비디오 편집을 지원할 수 있으며, 비디오 메타데이터를 시스템 관리 메타데이터와 사용자 관리 메타데이터로 분류함으로써 응용과 무관한 표준화된 비디오 메타데이터를 정의하였다. 그리고 실제 응용에 적용할 수 있도록 비디오 메타데이터를 설계하였다.

시간의 역관계를 포함한 13개의 시간 관계 연산을 더미 샷을 이용한 시간 변환 관계를 사용하여 6개로 감소시켜, 비디오 편집 목록 작성에 일관성을 유지하도록 하였다. 그리고 실제 응용인 광고에 적용하고 실험하여 확장된 DC 모델이 응용 분야에 동일한 구조로 메타데이터를 표현할 수 있음을 증명하였다.

이는 데이터베이스에서 비디오데이터를 표현하기 위한 메타데이터의 표준화로 응용분야에 영향을 받지 않는 시스템 구축으로 데이터의 공유와 관리에 일관성을 유지할 수 있게 되었다. 이러한 메타데이터를 이용하면 각종 멀티미디어 데이터를 이용하는 WWW, XML 등의 분야의 데이터를 데이터베이스에서 저장하고 관리하는데 있어서 규정된 필수 메타데이터외에 부차적인 구조를 사용할 수 있는 확장성이 있으며 각 요소들의 수록에 선택성이 보장되며 반복성과 한정어를 두어 세부 사항들을 조정할 수 있는 구조를 갖는 메타데이터로 표현 될 수 있으므로 동일한 비디오 데이터를 응용 분야에 따라 다른 구조를 갖는 메타데이터를 사용하지 않고 통합 할 수 있을 것이다.

이 논문은 이러한 비디오 데이터를 표현하기 위한 메타데이터를 표준화하는 관점에서 제안하였으며, 향후에는 메타데이터를 더욱 확장하여 비디오 데이터도 일반 문자와 같이 어떤 응용 분야에서도 같은 내용으로 인식하며, 데이터베이스 안에서 관리될 수 있는 일반화된 비디오 데이터형을 지원할 수 있도록 연구를 지속할 것이다.

참 고 문 헌

[1] Oracle8i, "Oracle8i interMedia Audio, Image, and Video User's Guide and Reference," <<http://sitelicense.arizona.edu/~oradoc/DOC/index.htm>>, 1999.

[2] Marian Gallagher, Kimberly Bostrom and Michelle Caisse, "Video Foundation DataBlade Module User's Guide, Version 1.2," INFORMIX Press, 1998.

[3] The Dublin Core Metadata Initiative, "Dublin Core Metadata Element Set, Version 1.1 : Reference Description," <<http://purl.org/dc/documents/>>, 1999.

[4] R. Hjelsvold, "Video Information Contents and Architecture," In Proceedings of the 4th International Conference on Extending Database Technology, Cambridge, UK, pp.259-272, Mar., 1994.

[5] R. Hjelsvold R. Midstrum, "Modeling and Querying Video Data," In Proceedings of the 20th International Conference on VLDB, pp.686-694, Oct., 1994.

[6] B. Prabhakaran, "Multimedia Database Management Systems," Kluwer Academic Publishers, Boston, pp. 1-84, 1997.

[7] S. W. Smoliar and H. J. Zhang, "Content-Based Video Indexing and Retrieval," IEEE Multimedia, 1(2), pp. 62-72, 1994.

[8] R. Jain and A.Hampapur, "Metadata in Video Databases," No.4, ACM SIGMOD RECORD, pp.27-33, December, 1994.

[9] T. D. C. Little, G. Ahanger, R. J. Folz, et al., "A Digital Video-On-Demand Service Supporting Content-Based Queries," In Proceedings of ACM Multimedia '93 Anaheim, U.S.A, August, 1993.

[10] 최주영, "메타데이터 확장을 통한 효율적인 뉴스 비디오 정보 검색 모델", 석사 학위 논문, 한국과학기술원, 1997.

[11] 이연배, 류근호, "시간지원 데이터베이스의 영상화를 위한 접속시스템의 설계 및 구현", 정보처리학회논문지, 제1권 제2호, pp.163-171, 1994.

[12] Silvia Hollfelder and Hyo-Jun Lee, "Data Abstraction for Multimedia Database Systems," In German National Research Center Technical Report No.1075, 1997.

[13] R. Hjelsvold, VideoSTAR - A Database for Video Information Sharing, Ph.D. Thesis, Norwegian Institute of Technology, Nov., 1995.

[14] L. A. Rowe, J. S. Boreczky, and C. A. Eads, Indexes for User Access to Large Video Databases, In Proc. of the IS&T/SPIE Symposium on Electronic Imaging Science and Technology, Conf. on Storage and Retrieval for Image and Video Databases II, San Jose, CA, Feb., 1994.

[15] 김기욱, 김형주, "비디오 주석 시스템의 설계 및 구현", 정보과학회논문지(B), 제24권 제6호, pp.588-597, 1997.

[16] R. Jain and A. Hampapur, Metadata in Video Databases, SIGMOD Record, Vol.23, No.4, pp.27-33, Dec., 1994.

[17] J. F. Allen, "Maintaining knowledge about temporal intervals," Communications of the ACM, Vol.26, pp.832-843, Nov., 1983.



이 순 희

e-mail : shlee@ansantc.ac.kr
1987년 한양대학교 전자과(공학사)
1993년 광운대학교 전산대학원(이학석사)
2001년 충북대학교 대학원 전자계산학과
(이학박사)

1987년~1993년 현대전자(주) 전산실 대리
1993년~1996년 현대정보기술(주) 기술개발팀장
1996년~현재 안산공과대학 컴퓨터정보과 교수
관심분야 : 멀티미디어 데이터베이스, 멀티미디어



김 길 준

e-mail : kimid@sungkyul.edu
1975년 숭실대학교 전자계산학과 졸업
1981년 숭실대학교 대학원 전자계산학과
공학석사
1991년 숭실대학교 대학원 전자계산학과
공학박사

1993~현재 성결대학교 전자상거래 학부 부교수
관심분야 : 전자상거래, 자연어 처리, 멀티미디어



김 상 호

e-mail : shkim@dblab.chungbuk.ac.kr
1997년 충북대학교 컴퓨터학과 졸업
1999년 충북대학교 대학원 전산학과 석사
1999년~현재 충북대학교 대학원 전산학과
박사과정

관심분야 : 시공간 데이터베이스, Web
Visualization, 시공간 뷰 등



류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대학교 전산과 졸업
1980년 연세대학교 공학대학원 전산전공
공학석사
1988년 연세대학교 대학원 전산전공 공학
박사

1976년~1986년 육군군수지원사 전산실(ROTC장교), 한국전자통신
연구소(연구원), 한국방송통신대 전산학과(조교수) 근무
1989~1991년 Univ. of Arizona 연구원(TempIS Project)
1986년~현재 충북대학교 전기전자컴퓨터공학부 교수
관심분야 : 시간지원 데이터베이스, 시공간 데이터베이스, Tem-
poral GIS, 지식기반 정보검색시스템, 데이터 보안, 데
이터 마이닝, Bio-Informatics 등



신 정 훈

e-mail : shinjh@iksan.ac.kr
1982년 숭실대학교 전자계산학과 공학사
1991년 충북대학교 대학원 전산통계학과
이학석사
1999년 충북대학교 대학원 전자계산학과
이학박사

1992년~현재 익산대학 컴퓨터학과 부교수
관심분야 : 시간지원 시스템, 지식베이스 시스템, 운영체제