

효율적인 의미 검색을 위한 동영상 데이터 모델링

정 미 영[†] · 이 원 석^{††}

요 약

본 논문에서는 많은 양의 정보를 복합적으로 포함하는 비디오 데이터의 의미적인 접근 효율을 높이고 저장 공간을 절약하기 위한 관리 방안을 제안한다. 비디오 데이터를 단순히 화일 단위의 비트열로 간주하여 처리하는 것이 아니라 원하는 부분을 속성을 통해 의미를 기술함으로써 접근할 수 있도록 한다. 비디오 데이터는 복잡한 의미를 내포하며 의미기술에 필요한 속성이 변할 수 있으므로 데이터베이스 시스템의 고정된 스키마를 통한 의미기술 방법과 유동적인 속성 집합을 통한 의미기술 방법을 병합하여 의미 기술하도록 지원하며 사용자마다 의미 표현 방법에 차이가 있을 수 있으므로 일관된 속성값을 선택할 수 있도록 속성과 속성값을 체계적으로 관리하는 방안을 제안한다. 또한 비디오 데이터베이스에 저장된 비디오 데이터간에 의미상속을 통해 접근 효율을 높이는 한편 비디오 데이터를 물리적으로 분리하거나 공유하여 새로운 프리젠테이션을 구성할 수 있도록 함으로써 저장공간을 효율적으로 사용할 수 있도록 지원한다.

Modelling of Video Data for Effective Content-based Retrieval

Mi-Young Jeong[†] · Won-Suk Lee^{††}

ABSTRACT

In this paper, we present an efficient way to describe the complex meaning embedded in video data for the content-based retrieval in a video database. Instead of viewing the data stored in a video file as a continuous bit stream, we associate the desired interval in the stream with a set of attribute and value pairs which describe its meaning. Since the meaning of video data can be complex and can change dynamically, it is necessary to combine the fixed schema of attributes used in the traditional database system with a description method through a set of attribute and value pairs that can be defined dynamically. As the content of video data can be expressed differently according to the view point of a user, it is important to maintain the meaning of the set of attribute and value pairs consistently for different users. This paper proposes the effective way to manage the set of attribute and value pairs. In addition, it also describes a way to define a new video presentation by separating a video stream physically or by sharing the portion of a bit stream, and the new method can minimize the required storage space.

1. 서 론

컴퓨터와 통신 그리고 데이터 압축기술의 발달로 동영상 데이터를 이용한 다양한 서비스가 가능하게 되었다[1, 2]. 비디오 데이터는 데이터의 특성상 많은 양의 정보를 복합적으로 포함할 수 있으므로 초고속 통신망 환경을 갖는 정보화 사회에서는 기존의 문자 데이터를 기본으로 한 정보 표현 방식에서 복합적이

※이 논문은 1996년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

† 정 회 원: 한국통신 연구개발본부 연구원

†† 정 회 원: 연세대학교 컴퓨터과학과

논문접수: 1996년 10월 14일, 심사완료: 1997년 3월 20일

고 사용이 편리한 비디오 데이터의 활용이 증가될 것이다. 따라서 비디오 데이터의 서비스 질을 향상 시키기 위해서는 비디오 데이터를 단순히 화일 단위의 압축된 비트열로 간주하여 처리하는 것보다 원하는 비디오 데이터의 내용을 기술하여 의미적인 접근이 가능하도록 하는 방안이 연구되어야 한다.

비디오 데이터에 나타나는 개체는 그 개체의 일반적인 성질을 나타내는 속성들을 가지며 비디오 데이터는 2차원 공간에서 이러한 개체들의 다양한 움직임을 그 의미로 갖는다. 또한 각 개체의 움직임 정보는 시간축을 중심으로 연속된 형태로 변경된다. 따라서 비디오 데이터의 의미 기술은 각 개체 단위의 위치 및 움직임 정보 뿐만 아니라 화면상에 동시에 나타나는 개체간의 연관된 움직임 또는 상태를 자세하게 기술할 수 있도록 지원해야 만이 많은 양의 비디오 데이터로 구성된 비디오 데이터베이스에서 원하는 정보를 갖는 화면을 정확히 추출할 수 있다.

이미지 데이터의 형태 인식에 대한 연구는 많이 진행되어 왔으나 비디오 데이터에 나타나는 복잡한 형태에 대한 자동 인식 시스템의 연구가 미비한 상황에서 비디오 데이터에 대한 의미적 접근에 관한 연구는 의미 속성집합을 통해 다양한 키워드를 부여하는 의미기술 방법과 텍스트에 의한 의미기술 방법을 중심으로 진행되고 있다. 텍스트에 의한 의미기술 방법은 의미부여는 용이하나 사용자에 따라 주관적으로 부여된 의미를 기초로 검색작업을 수행해야 하므로 일관된 구문분석이 선행되어야 한다. 따라서 다양한 의미를 내포하는 비디오 데이터에 일반적으로 적용하기에는 아직까지 검색 효율이 떨어진다[7, 8].

현재 비디오 데이터에 대한 검색 방법의 연구는 비디오 데이터에서 의미를 기술하고자 하는 범위(interval)를 정의하여 사용자가 키워드를 부여하는 방법으로 이루어지고 있고 대부분의 국내외 연구는 의미를 기술하는 속성 집합을 고정된 데이터베이스 스키마로 정의하여 동영상 데이터에 접근하도록 하는 방법[3, 4]과 일정한 스키마 없이(schemaless) 의미기술 속성을 동적으로 추가/삭제함으로써 유동적인 속성집합에 의해 접근하는 방법[5, 6, 13, 14]중 하나로 진행되고 있다. 그러나, 비디오 데이터베이스에 저장되는 비디오 데이터는 제목, 화일형식 및 개체의 일반적인 성질들과 같이 정형화된 속성 집합으로 표현할 수 있

는 의미와 화면에 나타나는 개체의 표정, 동작 및 상황과 같이 비디오 데이터에서 부분적으로 나타나면서 동적으로 변화되는 의미를 갖는다. 동적으로 변화되는 의미는 매우 다양하며 모델링하고자 하는 의미가 응용분야의 특성에 따라 다를 수 있으므로 데이터베이스를 구축할 때 정형화하여 고정된 스키마로 표현하는 것은 거의 불가능하다. 그러므로 비디오 데이터의 부분적인 의미를 보다 자세히 묘사하기 위해서는 유동적인 속성집합에 의한 의미기술 방법의 지원이 필수적이다. 그러나 고정된 스키마가 없이 모델링되는 동적인 정보는 스키마가 고정되어 정의되지 않았기 때문에 속성의 이름이나 속성의 도메인이 모호할 수 있으며 동적으로 속성이 삭제 또는 추가될 수 있다. 따라서, 속성의 선택이나 동일한 속성에 대한 속성값을 사용자의 표현 방식에 따라 다르게 부여할 수 있으므로 검색의 효율을 위해서 일관된 값의 선택이 가능하도록 속성 및 속성값의 체계적인 관리가 선행되어야 하지만 기존의 유동적인 속성집합을 사용하는 의미기술 방법들은 체계적인 속성 및 속성값의 관리를 고려하지 않았다.

비디오 데이터의 검색은 비디오 데이터에 부분적으로 주어진 동적인 의미뿐만 아니라 개체의 일반적인 성질이나 화일형식 등과 같이 고정된 스키마로 표현된 의미를 복합적으로 사용하여 질의를 형성할 수 있도록 지원하는 것이 데이터베이스에 저장된 데이터를 더욱 포괄적으로 검색할 수 있게 한다. 또한 동적인 의미는 화면에 나타나는 내용을 기술하는 의미로서 화면에 나타나는 개체의 크기 및 움직임 방향 등과 같이 움직임 자체를 기술하는 의미와 움직임의 결과로 표현되는 추상화된 의미가 있다. 이러한 두가지 의미는 의미 자체가 다름으로 이원화하여 모델링할 수 있도록 지원하여 상호 연계에 의한 검색을 지원하는 것이 더욱 풍부한 정보에 의한 검색을 지원할 수 있다. 하지만 대부분의 내용에 기초한 검색에 대한 연구는 동적인 의미만을 검색 대상으로 제한하며 동적인 의미도 움직임 자체에 대한 의미와 추상화되어 표현된 의미를 뚜렷하게 구별하지 않는다.

동일한 비디오 데이터에서 정의된 두 범위는 시간적인 범위 내포관계[9](temporal interval inclusion relationship)에 따라서 내포하는 범위에 부여된 속성이 내포되는 범위에 상속되는 성질을 갖는다. 이러한 성

절을 이용하여 범위간의 의미 공유를 통해 사용자에 의해서 정의되지 않은 부분으로의 접근이 가능하도록 하여 접근 효율을 향상시킬 수 있다. 또한 동영상 데이터는 압축되었다 하더라도 크기가 매우 크므로 비디오 데이터에서 극히 일부분만이 의미적인 검색 단위로 정의되는 경우 사용하지 않는 부분을 영구적으로 저장하는 것은 매우 비효율적이며 다수의 비디오 데이터에 정의된 부분들을 공유·결합하여 새로운 프리젠테이션을 정의할 수 있도록 함으로써 저장공간을 절약할 수 있다. 그러나 대부분의 연구는 의미 상속을 고려하지 않거나[3, 4, 7, 8] 고려했다 하더라도 제한된 의미 상속만을 지원하고 효율적인 저장 공간의 활용을 위한 비디오 데이터의 물리적인 분리를 고려하지 않았다[5, 6].

OVID 시스템[6]은 의미를 기술하고자 하는 비디오 데이터의 부분을 비디오객체로 정의하고 각 비디오 객체의 의미는 고정된 스키마가 없이 해당 비디오 객체의 의미를 나타내는 속성 및 속성값의 유동적인 집합으로 표현한다. 또한 사용자는 각 속성의 특성을 고려하며 속성의 상속 여부를 결정한다. 속성값간의 연관관계는 정의된 모든 속성값들에 대한 일반화/상세화 관계를 하나의 트리로 표현한 속성값 트리를 통해 유지한다. 속성값 트리는 내부적인 의미사전으로 사용되어 합병(merge) 및 중첩(overlap)연산에 의해 새롭게 정의된 비디오객체에 상속 가능한 속성과 속성값을 상속한다. OVID에서 의미적인 접근 단위가 되는 비디오객체는 각 객체가 생성된 이후에는 상호 독립적으로 유지되므로 비디오객체간에 동적으로 속성이 상속될 수 없으며 비디오 객체간에 중첩되는 부분이라도 사용자에 의해 독립적인 비디오객체로 정의되어 의미기술 되지 않은 경우 접근이 불가능하다. 또한 사용자마다 다양하게 정의할 수 있는 속성이름의 일관된 관리에 대한 언급이 없으며 모든 속성값을 하나의 속성값 트리로 관리함으로써 많은 속성값이 정의되면 속성값 트리가 너무 커져 사용자의 속성값 선택을 효율적으로 지원할 수 없다.

Algebraic video system[5]은 의미를 기술하고자 하는 부분을 비디오 대수학으로 표현된 다양한 비디오 표현식(video expression)으로 정의하여 해당 부분의 의미를 OVID에서와 같이 유동적인 속성집합에 의해 의미를 표현한다. 여러 비디오 표현식은 또 다른 표

현식으로 계층적으로 통합되어 의미 기술 될 수 있으며 이러한 표현식의 계층적 구조화는 새로운 프레젠테이션을 정의하는 방법도 된다. 하지만 속성값간의 관계를 유지하지 않으므로 사용자간에 일관된 속성값 선택의 지원이 어렵고 의미 상속에 있어서는 OVID와 달리 표현식의 계층구조의 변화에 따라 동적으로 변화된 의미를 상속할 수 있다. 하지만 상속되는 의미는 동일한 표현식의 계층 구조에 정의된 의미만을 상속할 수 있어 표현식에 해당되는 부분에 대해 다른 계층 구조에서 정의된 의미는 공유될 수 없다.

기존 연구들의 위와 같은 문제점을 보완하고 비디오 데이터의 특성에 맞는 질 높은 서비스를 제공하기 위해 본 연구는 비디오 데이터의 의미 기술 및 검색에 필요한 속성을 분류하고 각 분류마다 그 특성에 적합한 관리 방법을 제시한다. 각 속성 및 속성값의 저장 및 관리에 있어서 기존의 관계형 데이터베이스 관리시스템(DBMS)[10]의 기능을 최대한 활용할 수 있도록 한다. 또한 이러한 속성 관리 방법에 적합한 비디오 데이터 자체의 관리 측면으로 비디오 데이터를 물리적 또는 논리적으로 분리할 수 있게 하여 불필요한 부분을 제거하거나 동일한 부분을 공유함으로써 저장장소를 효율적으로 활용할 수 있도록 하며 논리적으로 내포하는 부분의 의미 상속을 지원하고 논리적으로 분리된 여러 부분을 통합하여 새로운 뷰를 정의할 수 있게 지원함으로써 질 높은 비디오 데이터 서비스를 가능하게 한다.

본 논문은 다음과 같이 구성된다. 2장에서는 동영상 데이터의 의미기술에 필요한 속성들을 분류하고 속성의 성질에 따른 관리 방법을 설명하며 연속된 비디오 데이터를 물리적으로 또는 논리적으로 분리하는 단위가 되는 비디오 데이터의 계층적 구조와 각 구조화 단위를 설명한다. 3장에서는 각 구조화 단위에 대한 의미 기술 방법과 구조화 단위간의 의미 상속 방법을 기술한다. 4장에서는 유동적인 속성 집합을 체계적으로 관리하기 위한 가변속성 관리자에 대해 설명하고 5장에서는 관계형 모델에 기초한 비디오 데이터베이스의 구성과 가능한 질의 형태를 예를 통해 살펴본 후 6장에서 결론과 향후 연구 방향을 제시한다.

2. 비디오 데이터 구조화

2.1 의미 속성의 분류

비디오 데이터를 기술하기 위한 속성의 분류는 (그림 1)과 같이 크게 5가지로 구분할 수 있으며 이들의 특성은 다음과 같다.

① 물리 속성

비디오 데이터의 해상도 및 화일위치 등과 같이 비디오 데이터의 물리적인 성질을 나타내기 위한 속성.

② 정적 속성

비디오 장르, 제목, 요약, 제작기간 및 제작자와 같이 화면에 부분적으로 나타나는 동적인 내용과 무관하게 비디오 화일 또는 뷰(view) 전체를 기술하는 속성.

③ 개체 속성

비디오 데이터에 나타나는 개체의 나이, 이름 및 크기 등과 같이 개체의 일반적인 성질(property)을 기술하기 위한 속성.

④ 개체 동적 속성

비디오 데이터의 특정 부분에서 화면상에 동적으로 표현되는 개체의 위치, 크기, 움직임 방향 및 소리 크기와 같이 단일 개체가 화면의 x-y 좌표를 중심으로 연속적으로 보여지는 정보를 기술하는 속성.

⑤ 내용기술 속성

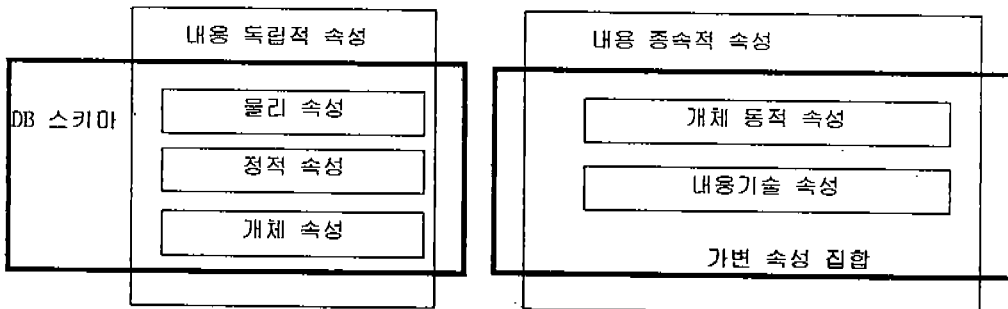
비디오 데이터의 특정부분에서 개체의 행위, 상태 또는 표정이나 개체간의 상호 움직임의 결과로 발생하는 상황, 사건이나 공간적 배경 등의 내용을 기술하는 속성이며 비교적 긴 시간에 걸친 추상적인 의미를 기술하기 위한 속성.

물리속성, 정적속성 및 개체속성은 내용 독립적인

속성으로 데이터베이스 구축시 정적으로 정형화할 수 있는 속성이므로 비디오 데이터베이스가 구축되는 해당 응용분야의 데이터베이스 스키마로 고정하여 정의한다. 반면에 내용 종속적인 개체동적속성과 내용기술속성은 응용분야의 특성이나 사용자의 관점에 따라 다양하게 표현될 수 있고 사용자에 의해 유동적으로 추가 및 삭제될 수 있으므로 일관된 속성값의 선택이 가능하도록 별도의 가변속성 관리 방법으로 체계적으로 관리한다. 개체동적속성의 의미는 응용 분야에 존재하는 개체가 화면에 나타나는 모양을 기술하는 속성이므로 속성이름, 속성값, 개체이름으로 표현되며 응용분야에서 정의된 개체이름이 속성이름으로 정의된 개체동적속성에 대해 속성값을 갖는다. 여기서 개체 이름은 응용분야의 특성에 따라 응용분야의 개체를 유일하게 나타낼 수 있는 값으로 간주한다. 반면에 내용기술속성에서는 특정 개체와 연관된 의미 정보를 표현할 때는 이와 동일한 방법을 사용하고 상황이나 사건과 같이 특정 개체에 대한 의미를 나타내지 않을 때는 개체이름을 사용하지 않는다.

2.2 비디오 데이터의 계층적 구조화와 구조화 단위

비디오 데이터베이스에 저장되는 비디오 데이터는 저장공간의 효율적인 활용 및 다양한 검색 단위와 프리젠테이션을 정의하기 위해 계층적으로 구조화 되어 유지된다. 비디오 데이터는 실제 비디오 화일의 비트열을 갖는 물리세그먼트, 물리세그먼트에서 특정 범위를 논리적으로 분리한 논리세그먼트, 그리고 논리세그먼트들의 공유·결합을 통해 정의되는 비디



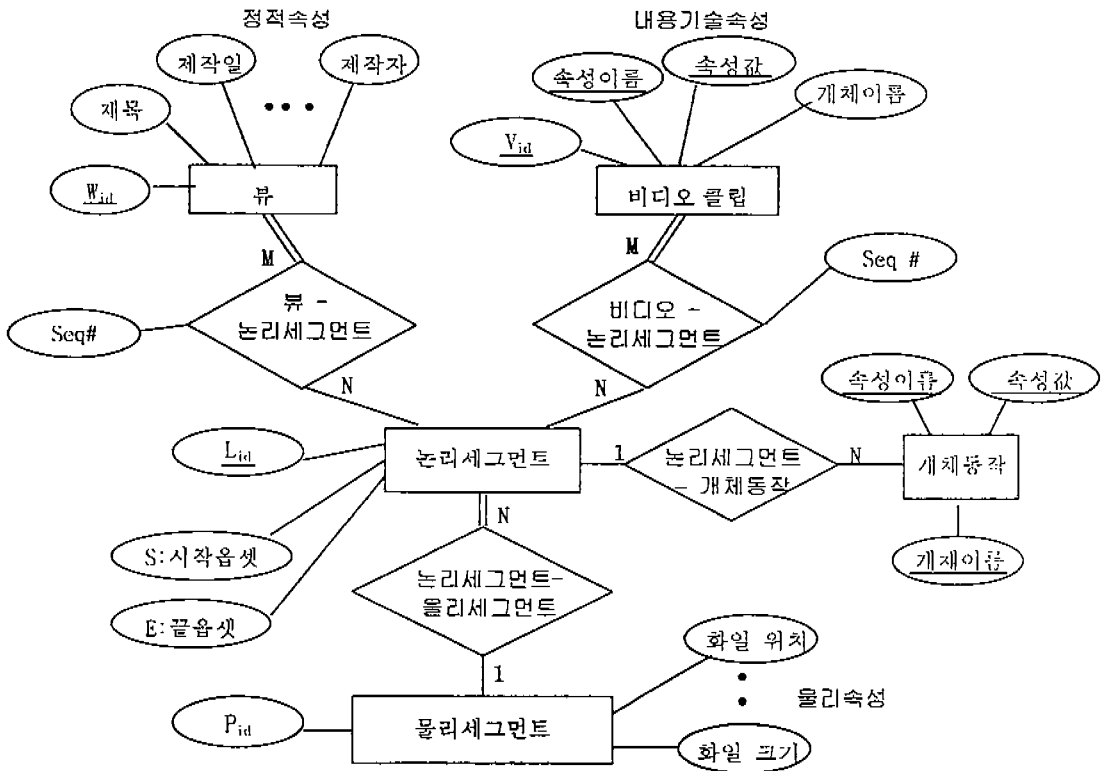
(그림 1) 속성의 분류
(Fig. 1) Types of attributes

오컬립, 그리고 비디오오컬립과 유사한 구조를 갖지만 새로운 프리젠테이션을 정의하는 뷰로 구조화된다.

물리세그먼트는 독립적으로 디코딩이 가능한 화일 단위로 정의되며 새로운 비디오 화일을 비디오 데이터베이스에 추가하고자 할 때 해당 화일의 위치 및 해상도와 같은 물리적 속성을 데이터베이스 스키마에 저장하고 실제 데이터를 화일시스템에 저장함으로써 정의된다. 또한 시스템에 정의된 모든 물리세그먼트는 각각 유일한 식별자(P_{id})를 갖는다. 논리세그먼트는 특정 물리세그먼트내에서 해당되는 부분의 시작 오프셋(offset) S 와 끝 오프셋 E 로 표현되며 모든 논리세그먼트도 물리세그먼트와 같이 각각 고유의 유일한 식별자(L_{id})로 구분된다. 오프셋은 해당 비디오 화일의 프레임(frame) 번호로 표현되며 MPEG[11, 12]와 같이 압축된 경우에는 GOP(group of pictures)로 표현된다. 논리세그먼트는 개체동적속성을 정의하는

단위이면서 동시에 비디오오컬립과 뷰를 정의하는 기본 단위가 된다. 비디오오컬립은 내용기술속성을 정의하는 단위이며 물리세그먼트로부터 정의된 논리세그먼트들의 순차적인 리스트(list)로 표현된다. 뷰의 구조는 비디오오컬립과 동일하지만 뷰의 사용 목적이 화면상의 의미를 기초로 한 검색이 아닌 여러 비디오 화일의 부분을 공유·결합하여 새로운 비디오 프로그램을 정의하는데 있으므로 뷰 전체를 기술하는 정적속성이 정의된다. 비디오오컬립과 뷰도 각각 유일한 비디오오컬립 식별자(V_{id})와 뷰 식별자(W_{id})를 갖고, 구성되는 각 논리세그먼트에 순차번호(Seq#)를 정의하여 리스트의 순서를 표시한다. 또한 사용자간에 효율적인 의미정보의 공유를 위해 시스템 차원에서 새로운 구조화 단위의 정의시 이미 정의된 동일한 구조화 단위가 이미 존재하는 경우 새롭게 정의하지 않는다.

비디오 데이터베이스에 정의된 검색 대상은 뷰 집



(그림 2) 구조화 단위의 개체-관계도
(Fig. 2) Entity-Relationship diagram for structuring units

합, 비디오클립 집합 그리고 논리세그먼트 집합이며 이들의 실제 데이터는 논리세그먼트 집합과 물리세그먼트 집합간의 대응(mapping)에 의해 정의된다. 하나의 물리세그먼트에 대해 서로 다른 범위를 갖는 다수의 논리세그먼트가 정의될 수 있는 반면 논리세그먼트는 하나의 물리세그먼트에서 정의된다. 따라서 물리세그먼트 집합과 논리세그먼트 집합은 1:N 관계를 갖는다. 비디오클립과 뷰는 논리세그먼트들의 순차적인 리스트로 표현되므로 하나 이상의 논리세그먼트로 뷰 또는 비디오클립이 정의될 수 있으며 논리세그먼트는 여러 뷰 또는 비디오클립에서 공유될 수 있다. 따라서 비디오클립 집합과 뷰 집합은 각각 논리세그먼트 집합과 M:N 관계를 갖는다. (그림 2)는 구조화 단위에 대한 개체-관계도(Entity-Relationship Diagram)를 보여주며 이에 해당하는 관계형 데이터베이스 스키마는 다음과 같이 설계할 수 있다.

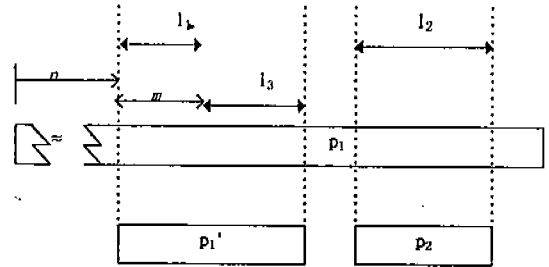
- 물리세그먼트(P_{id} , 파일이름, 위치, 해상도, ...)
 - ← 물리속성
- 논리세그먼트(L_{id} , P_{id} , S, E)
- 개체동작(L_{id} , 속성이름, 속성값, 개체이름)
 - ← 개체동작속성
- 비디오클립(V_{id} , 속성이름, 속성값, 개체이름)
 - ← 내용기술속성
- 뷰(W_{id} , 제목, 제작일, 제작자, ...)
 - ← 정적속성
- 비디오-논리세그먼트(V_{id} , L_{id} , Seq#)
- 뷰-논리세그먼트(W_{id} , L_{id} , Seq#)

여기서 개체이름은 비디오 데이터베이스가 구축되는 응용분야에서의 개체이름으로 간주할 수 있으며 이를 통하여 응용분야에 있는 개체의 일반적인 성질과 비디오 데이터베이스에 정의된 의미가 연결될 수 있다.

2.3 구조화 단위의 연산 작업

비디오클립과 뷰는 논리세그먼트를 공유할 수 있으므로 동일한 데이터를 부분적으로 갖는 비디오클립이나 뷰마다 동일한 데이터를 별도로 저장할 필요가 없다. 또한 특정 물리세그먼트에서 더이상 사용하지 않는 부분이 있고 그 크기가 크다면 이를 저장하는 것은 저장공간의 낭비를 초래하므로 논리세그먼트...

트들이 정의되어 사용되는 부분을 독립적으로 디코딩이 가능한 새로운 물리세그먼트들로 정의하고, 사용되지 않는 나머지 부분을 제거할 수 있도록 지원한다. 이 과정에서 각 논리세그먼트의 시작 및 끝 옵셋을 물리세그먼트의 변경에 맞게 조정해야 한다. (그림 3)은 논리세그먼트가 정의된 물리세그먼트를 삭제 연산의 수행시 논리세그먼트의 옵셋 변경이 이루어지는 과정을 보여준다. 물리세그먼트 p_1 이 삭제되는 경우 그 위에 정의된 논리세그먼트 l_1, l_2, l_3 은 시스템에 의해 새롭게 정의된 물리세그먼트 p_1', p_2 를 참조하는 새로운 범위로 변경된다. 즉 논리세그먼트 l_3 의 시작 옵셋은 물리세그먼트 p_1 에 대해서는 $n+m$ 이지만 새로운 물리세그먼트 p_1' 에 대해서는 m 이 되고 논리세그먼트 l_1 의 시작 옵셋은 새로운 물리세그먼트 p_1' 에 대해서 0이 된다.



(그림 3) 물리세그먼트의 분리
(Fig. 3) Separation of a physical segment

논리세그먼트는 연속된 물리세그먼트에서 정의가 가능함으로 이미 정의된 논리세그먼트들에 적용가능한 연산작업은 union_L, intersection_L 및 difference_L 작업이 있다. 중복된 부분을 갖는 두 논리세그먼트 l_1 과 l_2 에 대해 union_L(l_1, l_2) 작업은 l_1 과 l_2 의 최대 범위를 찾는 연산 작업이고, intersection_L(l_1, l_2)은 중복된 부분만을 찾는 연산 작업이다. 반면에 difference_L(l_1, l_2)은 l_1 의 범위에서 l_2 의 범위를 제외한 부분을 찾는 작업이다. 이와 유사하게 두개의 비디오 클립간의 중복 및 포함 범위를 찾는 연산 작업이 있다. 또한 비디오클립은 논리세그먼트들의 리스트로 표현되므로 기존의 비디오 클립으로 새로운 클립을 정의하는 연산으로는 concatenation_V과 pro-

jection_V이 있다. concatenation_V(v_1, v_2)는 v_1 의 논리세그먼트 리스트에 v_2 의 리스트가 연결된 새로운 비디오클립을 생성하는 연산으로 만약 중복된 논리세그먼트가 존재하면 오직 한번만 포함시킨다. projection_V(v, l)은 비디오클립 v 에 있는 논리세그먼트 l 을 갖는 새로운 비디오클립을 생성한다.

3. 의미기술 및 의미상속

3.1 의미 기술

사용자에 의해 동적으로 정의되는 내용종속속성은 사용자의 관점에 따라 유동적으로 새롭게 정의할 수 있으며 정의된 속성의 도메인(domain)도 고정된 범위를 갖지 않고 해당 속성으로 추가된 모든 속성값들을 그 범위로 갖는다. 이러한 특성을 갖는 속성의 활용면에서 사용자간의 일관된 속성과 속성값의 선택 방법은 4절에서 자세히 기술하기로 하고, 이 절에서는 각 구조화 단위에서 속성과 속성값의 관리 방법을 자세히 설명하기로 한다.

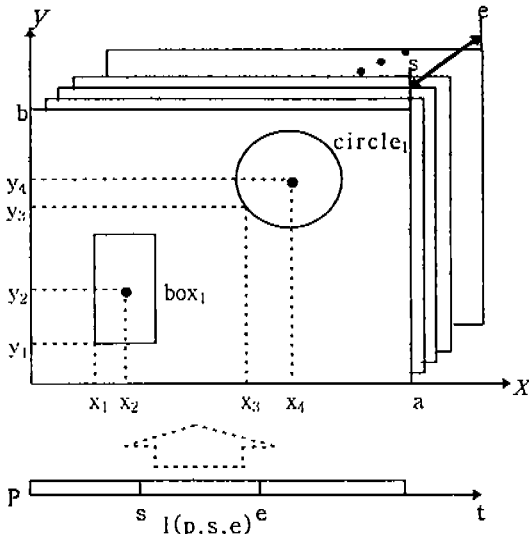
논리세그먼트에 정의되는 개체동적속성은 화면에 나타나는 개체의 모양을 기술하는 속성으로 화면상에 나타나는 개체의 중심 위치, 중심의 움직임 방향,

개체의 크기 또는 색, 그리고 소리의 크기 등과 같은 정보를 나타낸다. (그림 4)는 직사각형 개체 box_1 과 원 개체 $circle_1$ 이 물리세그먼트 p 에서 음셋 s 에서 e 까지의 부분에 정지되어 나타날때 이들 개체의 위치 정보를 나타내는 논리세그먼트 l 에는 다음과 같은 개체동적속성과 속성값을 정의할 수 있다.

비디오클립에 정의되는 내용기술속성도 개체동적속성과 유사한 방법으로 정의된다. (그림 5-(a))는 서로 다른 물리세그먼트로부터 정의된 논리세그먼트를 결합하여 비디오클립을 정의한 예를 보여주고 (그림 5-(b))는 동일한 물리세그먼트로부터 정의된 논리세그먼트를 결합하여 비디오클립을 정의한 예를 보여준다. 그림에서 w_{id} 는 뷰 식별자이며 각 뷰에 정적 속성으로 제목만을 표현하였지만 이밖에 다른 정적속성들과 함께 고정된 스키마로 뷰의 의미를 기술할 수 있다.

3.2 의미 상속

두개의 논리세그먼트들간에 존재할 수 있는 시간적인 관계는 한 논리세그먼트의 범위가 다른 논리세그먼트의 범위를 포함하는 내포관계와 두 논리세그먼트의 범위가 서로 중첩되는 부분이 존재하는 중첩



개체동적속성

Lid	속성이름	속성값	개체이름.
1	x-center	x_2	box_1
1	y-center	y_2	box_1
1	x-min	x_1	box_1
1	y-min	y_1	box_1
1	x-max	x_1+x_2	box_1
1	y-max	y_1+y_2	box_1
1	x-center	x_3	$circle_1$
1	y-center	y_3	$circle_1$
1	x-min	x_3	$circle_1$
1	y-min	y_3	$circle_1$
1	x-max	$x_4+(x_4-x_3)$	$circle_1$
1	y-max	$y_4+(y_4-y_3)$	$circle_1$

(그림 4) 논리세그먼트 정의 예
(Fig. 4) Example of logical segments

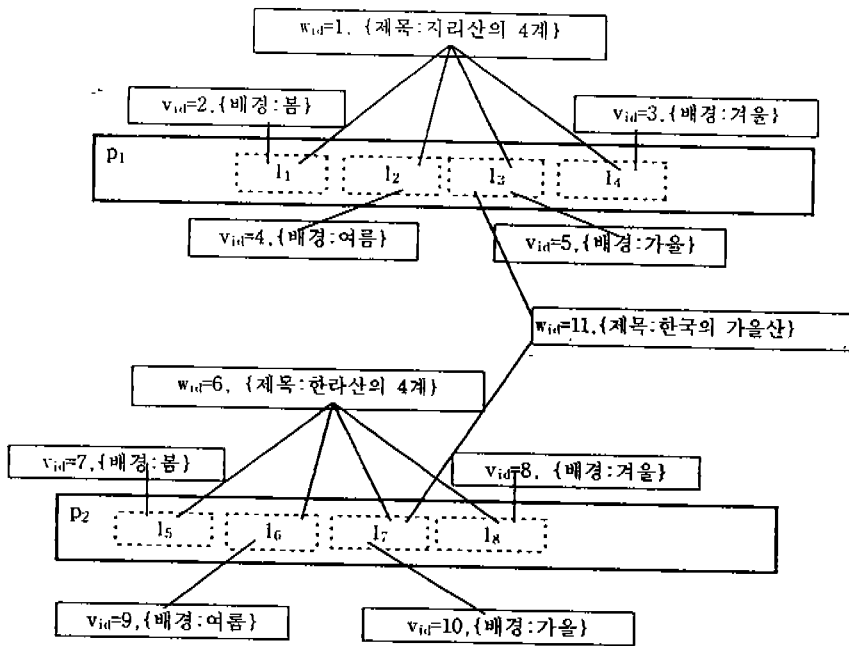


그림5-(a)

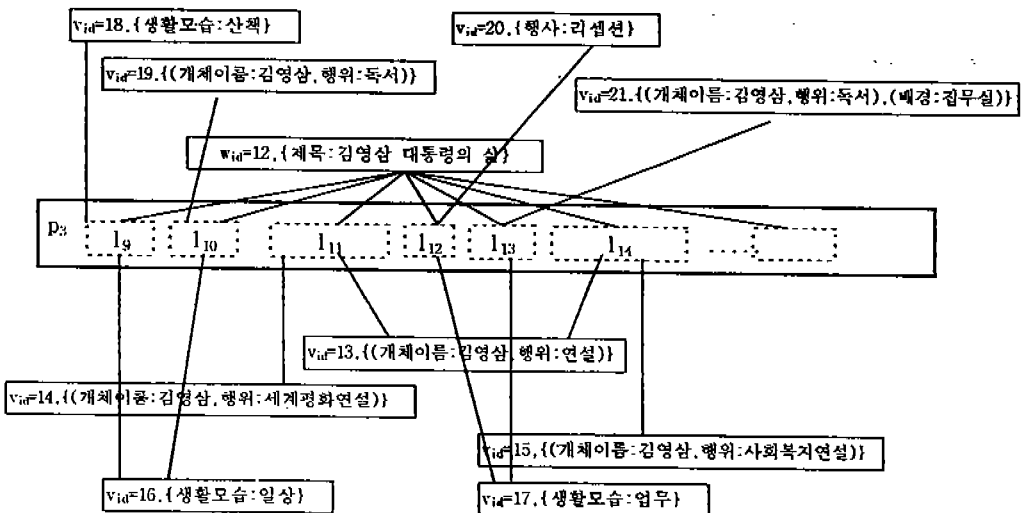


그림5-(b)

(그림 5) 예제 비디오클립
(Fig. 5) Example of video clips

관계가 있다. 두 논리세그먼트에서 내포관계가 성립할 때 내포하는 논리세그먼트에 정의된 개체동적속성과 속성값은 내포되는 논리세그먼트에 묵시적으로 상속될 수 있다. 그 이유는 내포관계가 성립되는 두 논리세그먼트에 주어진 의미는 동일한 영상에 대해 사용자가 그 의미를 기술하였고 큰 범위를 갖는 논리세그먼트에 주어진 개체의 개체동적속성은 당연히 더 작은 범위에 전반적으로 나타나는 의미이기 때문이다. 따라서 논리세그먼트에 주어진 의미는 자체에 주어진 자체의 의미와 내포관계에 의해 상속받은 상속의 의미로 구분할 수 있다. 또한 중첩관계를 갖는 두 논리세그먼트간에도 서로의 의미를 상속할 수 있으며, 중첩된 부분이 사용자에게 의해서 별도의 논리세그먼트로 정의되지 않았어도 각 논리세그먼트에 주어진 개체동적속성의 결합을 통해 독립적으로 접근이 가능하다.

개체동적속성은 화면에 나타나는 형태를 기술하는 속성이므로 동일한 물리세그먼트에서 내포 관계가 성립될 때 범위의 중복시 모든 의미를 상속할 수 있다. 반면에 내용기술속성과 속성값은 해당 비디오클립에 전체적으로 나타나는 개체동적속성과 속성값들의 추상화된 의미로 표현된다. 일반적으로 두 비디오클립이 내포관계를 가질 때 내포하는 클립의 의미보다 더 추상화된 의미로 기술된다. 만약 추상화 정도의 차이가 상당히 클 때 내포하는 클립 의미를 내포되는 클립에 상속하는 것은 내포되는 클립의 의미를 적절히 표현 하지 못할 수 있으며 일반적으로 내포관계를 갖는 두 비디오 클립에 주어진 의미의 추상화 차이는 각 클립의 크기의 차에 비례한다고 볼 수 있다. 이러한 관점에서 볼 때, 내용 기술 속성과 속성값의 상속은 OVID와 같이 속성을 정의할 때 해당 속성의 상속 여부를 사용자가 정의하는 방법과 위에서 기술한 방법처럼 속성에 관계없이 두 비디오클립의 추상화 정도의 차이를 파악하여 선별적으로 상속하는 방법이 있을 수 있다. 사용자에게 의한 정의 방법은 속성의 상속 여부가 명확하지 않을 수 있고 추상화 정도를 파악하여 상속하는 방법은 불필요한 의미를 상속할 수 있다.

사용자에게 의한 정의 방법은 OVID에서 사용한 방법을 그대로 사용할 수 있으므로 본 논문에서는 비디오 클립의 크기 차이에 따라 사용 여부를 결정하는

방법을 중심으로 기술하기로 한다. 내포관계를 갖는 두 비디오클립에서 내포되는 비디오클립 V_1 의 크기가 내포하는 비디오클립 V_2 의 크기에 비해 매우 작을 때는 V_2 에 주어진 의미는 V_1 의 의미에 비해 너무 많이 추상화된 의미일 수 있으므로 묵시적인 의미 상속은 두 비디오클립의 크기 비율이 미리 정해진 상수보다 클 때 이루어 지도록 한다. 즉 다음의 수식이 만족했을 때 의미 상속은 이루어지며 V_1 의 상속의 의미는 V_2 의 내용기술속성 및 속성값을 포함한다.

$$\frac{size(v_1)}{size(v_2)} > \delta_1 \quad (0 \leq \delta_1 < 1)$$

위의 식에서 $size(v)$ 는 비디오클립 v 의 크기를 나타내며 δ_1 은 비디오클립간의 상속을 결정하는 비율로 포함상속비율이라고 정의한다. 포함상속비율의 값을 적절히 할수록 비디오클립간의 의미 상속이 제한되지만 어느 정도 추상화 단계의 차이가 적은 의미만을 상속할 수 있고 반면에 크게 할수록 의미상속은 많이 되지만 상속되는 의미가 상속받은 클립의 의미에 적합하지 않을 수 있다.

두 비디오클립이 중첩관계를 가질 때 각 클립에 주어진 의미로 중첩된 부분을 독립적으로 접근할 수 있다. 중첩된 부분은 사용자에게 의해 별도의 비디오클립으로 정의되지 않았어도 두 비디오클립의 의미를 내포관계에 의해 상속받은 것으로 간주할 수 있다. 따라서 내포관계에서와 같이 중첩된 부분의 의미상속은 각 비디오클립의 크기에 비해 중첩되는 부분의 크기 비율이 정해진 비율보다 클 때에만 가능하도록 제한한다. 즉 두 비디오클립 V_1 과 V_2 이 다음을 만족할 때 $V_1 \cap V_2$ 에 상속된다.

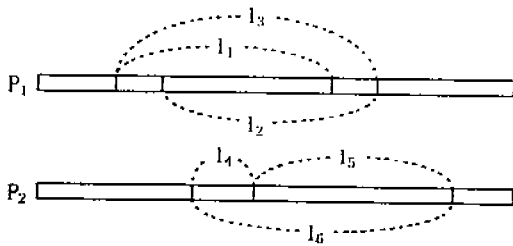
$$\frac{size(v_1 \cap v_2)}{size(v_1)} > \delta_2 \quad (0 \leq \delta_2 < 1)$$

이와 유사하게 V_2 의 의미는 다음이 만족될 때 $V_1 \cap V_2$ 에 상속된다.

$$\frac{size(v_1 \cap v_2)}{size(v_2)} > \delta_2 \quad (0 \leq \delta_2 < 1)$$

위의 식에서 δ_2 는 중첩상속비율이라고 정의하며 그 특성은 포함상속비율과 같다.

(그림 6)은 두개의 물리세그먼트 p_1 과 p_2 에 정의된 논리세그먼트와 비디오클립을 보여준다. l_3 에 주어진 개체동적 의미는 l_1 과 l_2 로, l_6 의 개체동적 의미는 l_4 와 l_5 로 상속이 가능하다. 이와 더불어 l_1 과 l_2 의 중복된 부분은 별도의 독립적인 논리세그먼트로 정의되지 않았지만 이들의 개체동적 의미를 검색조건으로 하여 l_1 과 l_2 의 중복된 부분을 접근할 수 있다. 또한 $\delta_1 = \delta_2 = 0.7$ 일때 비디오클립 v_1 에 주어진 내용기술속성은 v_2 와 v_3 로 상속이 가능하지만 v_4 와 v_5 로는 상속되지 않고 v_2 의 의미도 v_4 나 v_5 에 상속되지 않는다. 또한 v_2 와 v_3 는 이들이 중첩되는 부분의 크기가 중첩 상속 비율보다 크므로 v_2 와 v_3 의 의미는 이들이 중첩되는 부분으로 상속이 가능하다.



- | | |
|-----------------------|--------------------|
| $l_1 = (P_1, 10, 30)$ | $V_1 = [l_3, l_6]$ |
| $l_2 = (P_1, 15, 35)$ | $V_2 = [l_1, l_6]$ |
| $l_3 = (P_1, 10, 35)$ | $V_3 = [l_2, l_5]$ |
| $l_4 = (P_2, 50, 55)$ | $V_4 = [l_6]$ |
| $l_5 = (P_2, 55, 80)$ | $V_5 = [l_4, l_1]$ |
| $l_6 = (P_2, 50, 80)$ | |

(그림 6) 논리세그먼트와 비디오클립의 의미상속
(Fig. 6) Description inheritance in logical segments and video clips

이와같이 논리세그먼트나 비디오클립간의 내포 및 중첩관계를 파악함으로써 다음과 같은 두 가지 특징이 있다. 중첩관계의 파악은 사용자에게 의해서 의미가 기술되지 않은 부분으로의 접근이 가능하도록 지원할 수 있고 내포관계에 의한 의미상속은 서로 다른 사용자에게 의해 기술된 의미를 공유할 수 있게 지원함으로써 풍부한 의미정보에 의해 보다 정확한 비디오 데이터로의 접근이 가능하다.

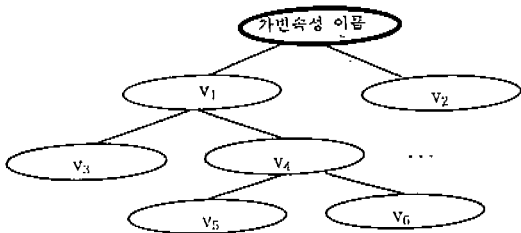
이러한 상속의미는 상속받은 의미를 해당 논리세그먼트나 비디오클립에 독립적으로 복사해서 저장하

거나 필요시 동적으로 파악할 수 있다. 의미를 복사하는 방법은 저장공간의 낭비를 초래하고 새로운 논리세그먼트나 비디오클립이 정의되면 이들의 상속 여부를 검사하여 갱신해야 하는 단점이 있으나 검색시 의미 상속관계를 동적으로 파악할 필요가 없으므로 처리시간이 절약되고 상속하는 비디오클립이 삭제됨에 관계없이 상속받은 의미를 유지할 수 있는 장점을 지닌다. 반면에 필요시 의미 상속이 이루어지도록 하는 방법은 새로운 비디오클립이 추가될 때 상속관계를 찾는 오버헤드가 없고 복사하여 두는 방법보다 저장공간이 절약되지만 검색 작업을 수행할 때 상속관계를 파악해야 하는 오버헤드가 따르고 상속할 수 있는 비디오클립이 삭제되는 경우 상속될 수 있는 의미를 잃어버린다는 단점이 있다. 일반적으로 새로운 논리세그먼트나 비디오클립을 생성하는 작업보다 검색 작업이 더욱 빈번히 수행되므로 상속의미를 독립적으로 복사하여 저장하는 것이 더욱 효율적일 수 있지만, OVID에서와 같이 사용자가 상속 여부를 결정하지 않고 크기 비교에 의한 자동적인 상속인 경우에는 상속 의미가 자체의미 보다 의미의 표현이 부정확할 수 있다. 따라서 논리세그먼트나 비디오클립에서 상속의미와 자체의미를 개념적으로 분리하여 검색 질의 자체도 상속의미를 포함하는 질의와 자체의미만을 대상으로 한 질의로 이원화하여 지원하도록 한다. 이러한 환경에서 상속의미는 독립적으로 복사되어 저장되지 않으므로 상속의미는 자체의미로 검색된 논리세그먼트 또는 비디오 클립간의 시간적인 내포관계를 파악하여 추출하게 된다.

4. 가변속성 관리자 및 의미적인 색인

내용기술속성이나 개체동적속성은 유동적으로 추가 및 삭제될 수 있으므로 가변속성 관리자로 관리하며 이러한 속성들의 속성값도 서로 다른 사용자가 일관되게 선택할 수 있도록 의미적인 색인을 유지함으로써 체계적으로 속성과 속성값을 유지한다. 의미적인 색인은 (그림 7)과 같이 가변속성 값들을 속성값간의 의미적인 연관성에 따라서 트리형태로 표현하며 이러한 트리를 가변속성트리라고 정의한다. 가변속성트리는 가변속성 이름을 루트(root)로 하고 그 하위 레벨에 타원으로 표현되는 속성값 노드가 위치하는

구조를 갖는다. 속성값간의 관계는 일반화 관계와 상세화 관계로 나눌 수 있으며 트리상의 자식노드는 부모노드보다 더 특수화된 속성값을 갖게된다. 예를 들어 (그림 7)에서 v_1 은 v_3 보다 상위 레벨에 해당되므로 보다 일반적인 값을 갖는다. 가변속성트리는 속성과 속성값 선택을 용이하게 하며 서로 다른 사용자간에 일관성을 보장해 줄 수 있다.

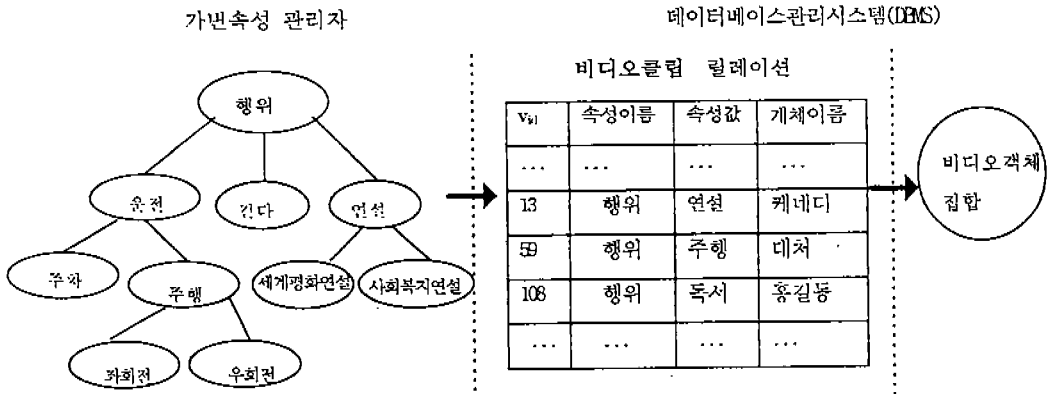


(그림 7) 가변속성 트리
(Fig. 7) A flexible attribute tree

가변속성관리자는 비디오데이터베이스에서 동적으로 정의된 속성들과 그들의 값을 사용자가 편리하게 검색할 수 있도록 하는 인터페이스로 간주할 수 있다. 트리에 정의된 속성값 노드는 실제 속성값에 해당되는 논리세그먼트나 비디오클립과 연결되며 이들간의 연결 정보는 개체동적 속성인 경우 2절에서

기술한 개체동작릴레이션인 개체동작 스티마에 저장되고, 내용기술속성인 경우에는 비디오클립의 의미 정보를 저장하는 스키마인 비디오클립에 저장된다. 사용자에게 의해 정의된 각 속성에 대해 한개의 가변속성트리기가 존재하지만 이들의 정보는 속성의 특성에 따라 개체동작 스키마 또는 비디오클립 스키마에서 통합되어 관리된다. 따라서 사용자가 검색을 위해서 활용 가능한 속성은 가변속성트리의 루트로 정의된 속성들이며, 이들 각각의 도메인(domain)은 해당 트리의 모든 자식 노드가 된다. (그림 8)은 내용기술속성의 하나인 '행위'에 대한 가변속성트리의 예를 보여 준다. 비디오클립 스키마의 개체이름 속성은 여러 데이터형의 개체이름이 복합적으로 나타날 경우 별도의 속성을 두어 해당 튜플의 개체 이름의 도메인형을 나타냄으로써 서로 다른 도메인을 갖는 속성값들을 저장할 수 있다.

사용자는 논리세그먼트나 비디오클립에 의미를 부여하기 위해 가변속성트리를 선택하고 원하는 속성값 노드를 선택한다. 만일 원하는 속성이 정의되지 않았을 경우 새로운 가변속성트리를 정의하고, 적절한 속성값이 없는 경우 속성값을 트리상의 알맞는 위치에 추가할 수 있다. 이렇게 동적으로 추가되는 속성이나 속성값은 해당 릴레이션의 갱신작업으로 반영된다.



(그림 8) 가변속성 값과 비디오클립과의 연결
(Fig. 8) Association between video clips and the values of their flexible attributes

5. 비디오 데이터베이스의 구성 및 질의 형태

5.1 시스템 구성도

비디오 데이터베이스를 유지하기 위한 시스템의 전체 구성도는 (그림 9)와 같고 크게 가변속성관리자와 고정 데이터베이스 스키마 그리고 화일시스템으로 구성된다.

5.2 질의 형태

사용자는 (그림 10)과 같이 논리세그먼트나 비디오 클립의 검색을 위해 가변속성 관리자를 통해 가변속성 트리에서 원하는 속성값을 선택함으로써 질의를 형성하거나 고정된 데이터베이스 스키마에 대해 직접적으로 SQL(Structured Query Language)문을 통해 질의를 표현할 수 있다. 사용자는 질의 결과로 일반적인 속성값이나 각 구조화 단위를 유일하게 나타내는 식별자를 얻게된다.

5.2.1 가변속성관리자를 통한 질의

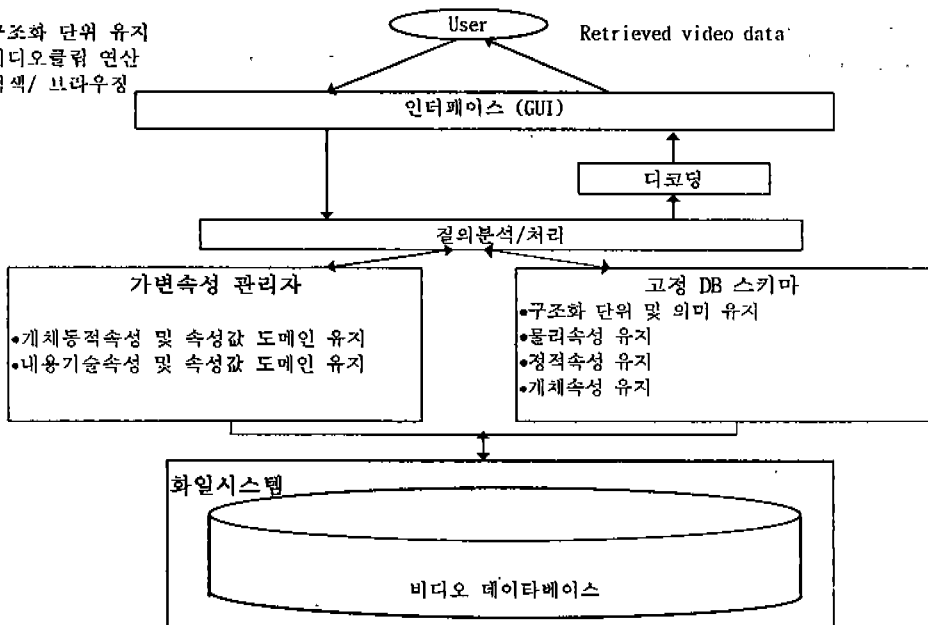
가변속성관리자에 의해 관리되는 개체동적속성이

나 내용기술속성에 기초한 질의의 유형으로는 각 논리세그먼트나 비디오클립에 주어진 자체의미만을 검색 대상으로 하는 단순질의, 포함 또는 중첩관계로 상속된 의미를 검색 대상으로 포함하는 상속질의로 구분할 수 있다. 단순질의는 원하는 속성이 정의된 가변속성트리를 선택하고 화면에 나타난 속성트리를 통해 원하는 속성값을 선택하고 특정 개체에 대한 의미인 경우 개체의 이름을 사용하여 질의를 형성한다. 이렇게 형성된 질의는 내부적으로 개체동작이나 비디오클립 릴레이션에 대한 SQL문으로 변환되어 논리세그먼트나 비디오클립을 검색할 수 있다. 예를 들어 김영삼에 대해서 “행위”라는 가변속성을 선택하고 속성트리에서 “연설”인 값을 선택했다면 내부적으로 Q1과 같은 SQL문으로 변환되어 질의가 처리된다.

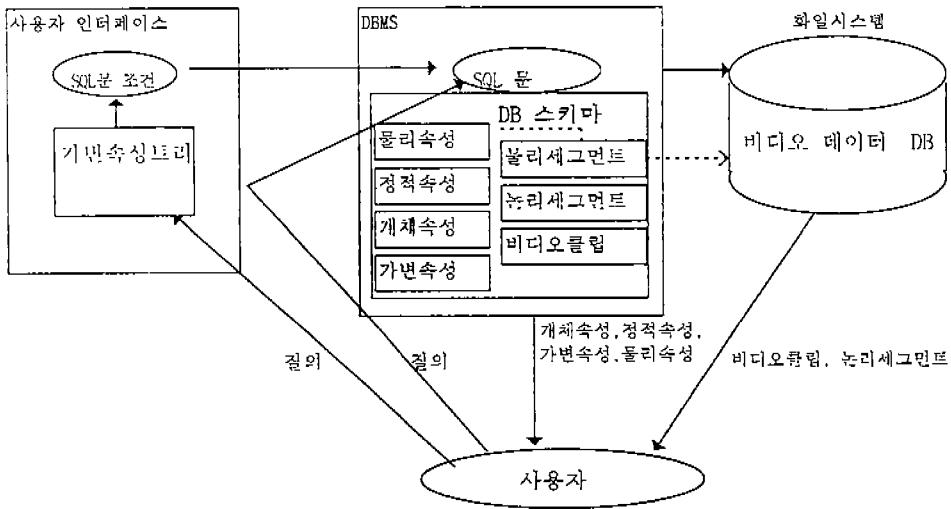
```

Q1: select Vid
    from 비디오클립
    where 속성이름='행위' and 속성값='연설'
    and 개체이름='케네디
    
```

- 구조화 단위 유지
- 비디오클립 연산
- 검색/ 브라우징



(그림 9) 비디오 구조화 시스템
(Fig. 9) Video structuring system



(그림 10) 비디오 데이터베이스 검색 유형
(Fig. 10) Querying patterns in the video database

질의의 조건은 다수의 가변속성트리에서 표현된 조건을 AND 또는 OR 논리연결자로 연결하여 표현할 수 있다. OR 연결자로 표현된 두 조건은 하나의 SQL문으로 표현이 가능하나 AND 연결자로 표현된 조건은 각 조건을 SQL문으로 변경한 후 각 SQL문을 집합 연산자인 교집합(intersection)으로 연결한 SQL문으로 변환된다. 예를 들어 개체 box_1 의 중심이 axb 화면상에서 $x < a/2, y < b/2$ 부분에 나타나는 논리세그먼트를 찾는 질의는 Q2와 같이 표현할 수 있다.

Q2: (SELECT L_d
FROM 개체동작
WHERE $x\text{-center} < a/2$ and 개체이름 = box_1)
∩
(SELECT L_d
FROM 개체동작
WHERE $y\text{-center} < b/2$ and 개체이름 = box_1)

동일한 가변속성트리에서 한개 이상의 속성값노드를 선택할 수 있으며 이때 선택된 값들은 SQL의 where 절에서 OR 연결자로 연결된 조건을 형성한다. 가변속성트리의 속성값노드는 일반화/상세화 관계를 가지며 특정 노드 이하의 서브트리에 해당하는 속성값

노드들을 모두 선택할 수도 있다.

상속질의는 논리세그먼트나 비디오클립에 정의된 자체의미와 내포 또는 중첩관계에 의해 상속된 상속의미를 검색대상으로 한다. 논리세그먼트에 대한 상속질의는 질의를 구성하는 각 가변속성트리의 검색조건을 단순질의로 처리하여 얻은 결과 논리세그먼트집합들을 대상으로 2절에서 기술된 구조화 단위 연산 작업을 사용하여 동일한 물리세그먼트에서의 내포 및 중첩관계를 파악함으로써 해당 검색 조건을 만족하는 논리세그먼트들을 검색한다. 검색 결과는 중첩관계로 인해 사용자에게 의해 정의된 논리세그먼트가 아닌 물리세그먼트 내에 존재하는 범위가 될 수도 있다. 예를 들면 (그림 4)에서 box_1 이 t_1 부터 t_3 까지 나타나고 해당 부분을 논리세그먼트 I_1 으로 정의하고 circle이 t_2 부터 t_4 까지 나타나고 해당 부분을 논리세그먼트 I_2 로 정의했다면($t_2 < t_3$) 두 개체가 동시에 나타나는 부분을 검색 하고자 하면 단순 질의의 결과로 얻은 I_1 과 I_2 에서 시간적인 내포관계에 따라 사용자에게 의해 별도로 정의된 논리세그먼트가 아니고 물리세그먼트 p에 대해 시작 읍셋이 t_2 면서 끝 읍셋이 t_3 인 범위를 결과로 얻는다. 비디오클립의 상속질의도 동일한 방법으로 처리하지만 의미상속에 있어서 이미 정해진 포함상속비율이나 중첩상속비율을 고려하여

상속 여부를 결정한다.

5.2.2 SQL문을 통한 질의

SQL을 사용한 비디오 데이터베이스의 검색은 2절에서 정의한 릴레이션인 물리세그먼트, 논리세그먼트, 개체동작, 비디오클립, 뷰, 비디오-논리세그먼트 및 뷰-논리세그먼트에 대해 표현할 수 있으며 특정 개체에 대한 질의시 개체의 일반적인 성질을 갖는 응용분야의 릴레이션이 될 수 있다. 예를 들어 제목이 '한국의 가을 산'으로 정의된 뷰를 검색하기 위한 질의는 Q3와 같고 가변속성 "배경"에 대해 "바다"를 배경으로 하는 비디오클립을 검색하기 위한 SQL문은 Q4와 같다.

Q3:

```
select Wid
from 뷰
where 제목='한국의 가을산'
```

Q4:

```
select Vid
from 비디오클립
where 속성='배경' and 속성값='바다'
```

또한 물리세그먼트 p의 t₁에서 t₂까지의 부분에 정의된 모든 논리세그먼트를 찾는 질의는 Q5와 같이 표현할 수 있고, 비디오클립 v의 논리세그먼트 리스트를 찾는 질의는 Q6와 같이 표현할 수 있다.

Q5:

```
SELECT Lid
FROM 논리세그먼트
where Pid=p and S>t1 and E<t2
```

Q6:

```
SELECT Lid
FROM 비디오-논리세그먼트
WHERE Vid=v
ORDER BY Seq#
```

SQL문을 통해 표현된 질의와 가변속성관리자를 통해 표현된 질의는 union/intersection 등과 같은 집합연산(set operation)으로 연결하여 표현할 수 있다. 예를 들어 케네디가 바다를 배경으로 연설하는 비디오클립은 Q1 ∩ Q4로 표현할 수 있다. 또한 응용분야에 정치인(이름, 나이, 성별, 국명, 정치활동시기)라는 릴레이션이 있을 때 한국의 남자 정치인이 연설하는

행위를 담은 비디오클립을 검색하기 위한 질의는 Q7과 같이 표현할 수 있다.

Q7:

```
select Vid
from 비디오클립
where 속성이름='행위' and 속성값='연설' and
개체이름 in(select 이름
from 정치인
where 국명='한국' and 성별='남자')
```

6. 결론 및 향후 연구 방향

비디오 데이터는 데이터의 특성상 많은 양의 정보를 복합적으로 포함할 수 있으므로 향후 초고속 정보통신망 환경을 갖는 정보화사회에서는 기존의 문자 데이터를 기본으로 한 정보 표현 방식에서 복합적이고 사용이 편리한 비디오 데이터의 활용이 증가될 것으로 예상된다. 비디오 데이터는 양이 방대하고 다양한 의미를 기술하기 위해 많은 속성이 필요하므로 저장공간을 효율적으로 사용하면서 효과적인 내용기초 검색이 가능하도록 비디오 데이터를 모델링함으로써 비디오 데이터의 활용을 가속화 시키고 서비스의 질을 향상시킬 수 있다. 본 연구는 복합적 의미를 갖는 비디오 데이터로부터 의미적 분리를 통한 새로운 프리젠테이션의 정의와 효율적인 의미적 접근 방안을 제시하기 위해 비디오 데이터를 계층적으로 구조화하고 각 구조화 단위간의 매핑정보와 동적인 속성을 데이터베이스화하여 관리함으로써 관계형 데이터베이스 관리시스템의 기능을 최대한 활용하면서 비디오 데이터에 기술된 다양한 의미정보를 체계적으로 관리하는 방법을 제시하였다. 또한 가변속성관리자를 두어 동적인 속성 추가와 서로 다른 사용자간에 일관된 속성값 선택을 가능하게 하고 방대한 의미 정보간의 관계를 갖는 의미사전을 사용하지 않고 동적으로 논리세그먼트나 비디오클립간에 의미를 상속하여 의미 접근을 강화하는 방안을 제안하였다.

다양한 움직임 정보를 모델링해야 되는 개체동적 속성은 속성의 특성에 알맞는 추상 데이터형으로 표현하는 것이 의미 표현이나 활용에 있어서 더 용이할 수 있다. 따라서 복합 데이터형에 의한 모델링 및 메소드(method)를 원활히 지원하는 객체지향 데이터베이스

스에서 연구가 필요하며 이와 동시에 객체 단위의 모델링 및 접근 방법을 강화함으로써 객체 단위로 압축된 비디오 데이터들의 융합에 의한 비디오 데이터의 편집 등 다양한 분야에서 활용될 수 있을 것이다.

참 고 문 헌

[1] R. Price, "MHEG: An introduction the future International Standard for Hypermedia Object Interchange," ACM Proceeding of Multimedia, CA, USA, 1993.

[2] D. Deloddere, W. Verbiert and H. Verhille, "Interactive Video on Demand", IEEE communications Magazine, May 1994.

[3] Yoshinobu Tomomura et al, "Structured Video Computing", IEEE Multimedia, Vol. 1, No. 3, Fall 1994.

[4] T. D. C. Little, G. Ahanger, "A Digital On-Demand Video Service Supporting Content-Based Queries", ACM Proceeding Multimedia 93, /6/93/CA, USA

[5] R. Weiss, A. Duda, and D. K. Gifford, "Composition and Search with a Video Algebra", IEEE Multimedia, Vol. 2, No. 1, Spring, 1995.

[6] Eitetsu Oomoto and Katsumi Tanaka, "OVID: Design and Implementation of a Video-Object Database System", IEEE trans. on knowledge and data engineering, vol. 5, No. 4, August 1993.

[7] Rune Hjelsvold and Roger Midtstraum, "Modeling and Querying Video Data", Proceedings of the 20th VLDB Conference Santiago, Chile, 1994.

[8] M. Washisaka, at al, "Video/Text Linkage System Assisted by a Concept Dictionary and Image Recognition," IEEE Proceedings of Multimedia, 1996.

[9] Thomas D.C.Little, "Interval-Based Conceptual Models for Time-Dependent Multimedia Data", IEEE transactions on knowledge and data engineering, vol. 5, No. 4, August 1993.

[10] Ramez Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems," The Benjamin/Cummings Publishing Company Inc., 1994.

[11] 권주한, 임명식, 정제창, "MPEG Video 표준해설", 한국통신학회지 제11권 제8호, 1994년 8월

[12] 김용한, 안상미, 안치득, "MPEG-4표준화 동향 및 전망", 한국통신학회지 제11권 제8호, 1994년 8월

[13] T. Kuo and A. Chen, "A Content-Based Query Language for Video Database," IEEE Proceedings of Multimedia, 1996.

[14] Stephen W. Smoliar and HongJiang Zhang, "Content-Based Video Indexing and Retrieval", IEEE Multimedia, Vol. 1, No. 2, Summer, 1994.



정 미 영

1994년 2월 덕성여자대학교 전산학과 졸업(학사)
 1996년 8월 연세대학교 대학원 전산과학과 졸업(석사)
 1996년 8월~현재 한국통신 연구개발본부 연구원

관심분야: 멀티미디어 데이터베이스, 데이터 모델링



이 원 석

1985년 Boston Univ. 컴퓨터공학 학사
 1987년 Purdue Univ. 컴퓨터공학 석사
 1990년 Purdue Univ. 컴퓨터공학 박사
 1992년 3월~1992년 12월 삼성

전자 선임연구원

1993년~현재 연세대학교 컴퓨터과학과 조교수
 관심분야: 비디오 데이터 모델링, 연역 데이터베이스, 멀티미디어 데이터베이스, 객체지향 데이터베이스