

이스트맨 코닥
미국 뉴욕주 로체스터

(72) 발명자

백승엽
미국10027뉴욕주뉴욕리버사이드드라이브530어파트먼트6제이

베니테즈아나
미국10027뉴욕주뉴욕웨스트119번스트리트400어파트먼트9에프

창시푸
미국10027뉴욕주뉴욕리버사이드드라이브560어파트먼트18케이

리칭셴
미국뉴욕주호손티제이.왓슨리서치센터아이비엠내

스미스존알.
미국뉴욕주호손티제이.왓슨리서치센터아이비엠내

베르그만로렌스디.
미국뉴욕주호손티제이.왓슨리서치센터아이비엠내

푸리아털
미국뉴저지주레드뱅크에이티앤티랩스

황귀양
미국뉴저지주레드뱅크에이티앤티랩스

쥬디스찰리
미국뉴욕주로체스터이스트맨코닥내

(74) 대리인

유미특허법인

(56) 선행기술조사문헌

05144685

05579444

* 심사관에 의하여 인용된 문헌

심사관 : 손영태

전체 청구항 수 : 총 64 항

(54) 이미지 디스크립션 시스템 및 방법

(57) 요약

이미지 콘텐츠를 기술하기 위한 시스템 및 방법은 객체 집합(24), 객체 계층(26) 및 엔티티 관계 그래프들(28)을 포함하는 이미지 디스크립션 레코드들을 규정한다. 이미지 콘텐츠의 경우, 이미 객체들은 전역 객체들(O0 8) 및 지역 객체들(O1 2 및 O2 6)을 포함할 수 있다. 이미지 객체들은 차례로 다수의 특정 디스크립터들에 의해 정의되는 다수의 상이한 클래스들(36, 38 및 40)에 의해 정의된다. 객체 집합의 객체들 사이의 관계는 객체 계층(26) 및 엔티티 관계 그래프(28)에 정의된

다. 이미지 디스크립션 레코드들은 검색 엔진들, 필터들 및 파일저장 시스템(archive system)과 같은 컴퓨터 애플리케이션들에 의해 그 다음의 접근 및 프로세싱을 위해 이미지 정보의 콘텐츠 및 콘텍스트를 기술하는 표준 수단(standard vehicle)을 제공한다.

대표도

도 2

특허청구의 범위

청구항 1.

이미지 정보로부터 디스크립션 레코드(description record)를 생성하는 시스템에 있어서,

상기 이미지 정보를 수신하는 적어도 하나의 이미지 입력 인터페이스;

상기 이미지 정보를 수신하는 상기 적어도 하나의 이미지 입력 인터페이스와 연결되며,

상기 이미지 정보로부터 이미지 객체 디스크립션들을 생성하도록 이미지 객체 추출 프로세싱을 수행함으로써 상기 이미지 정보를 처리하고,

이미지 객체 계층 디스크립션들을 생성하도록 객체 계층 구축 및 추출 프로세싱을 수행함으로써 상기 생성된 이미지 객체 디스크립션들을 처리하며,

엔티티 관계 그래프 디스크립션들을 생성하도록 엔티티 관계 그래프 생성 프로세싱을 수행함으로써 상기 생성된 이미지 객체 디스크립션들을 처리하고,

상기 이미지 객체 디스크립션들, 상기 이미지 객체 계층 디스크립션들 및 상기 엔티티 관계 그래프 디스크립션들을 포함하는 적어도 하나의 디스크립션 레코드가 상기 이미지 정보 내에 포함된 콘텐츠를 표현하도록 생성되는 컴퓨터 프로세서; 및

상기 프로세서와 동작 가능하게 연결되며 상기 적어도 하나의 디스크립션 레코드를 저장하는 데이터 저장 시스템(data storage system)

을 포함하는 디스크립션 레코드 생성 시스템.

청구항 2.

제1항에 있어서,

상기 이미지 객체 추출 프로세싱과 상기 객체 계층 구축 및 추출 프로세싱은 병렬로 수행되는 디스크립션 레코드 생성 시스템.

청구항 3.

제1항에 있어서,

상기 이미지 객체 추출프로세싱은,

상기 이미지 정보 내의 각 이미지를 상기 이미지 내의 영역(region)으로 분할하는 이미지 세그먼트 프로세싱; 및

하나 이상의 상기 영역에 대해 하나 이상의 특징 디스크립션들을 생성하는 특징 추출 및 주석 프로세싱을 포함하며,

상기 생성된 이미지 객체 디스크립션들을 하나 이상의 상기 영역에 대해 상기 하나 이상의 특징 디스크립션들을 포함하는 디스크립션 레코드 생성 시스템.

청구항 4.

제3항에 있어서,

상기 하나 이상의 특징 디스크립션은 미디어 특징들, 시각적 특징들 및 의미적 특징들로 이루어지는 그룹에서 선택되는 디스크립션 레코드 생성 시스템.

청구항 5.

제4항에 있어서,

상기 의미적 특징들은 누가(who), 대상(what object), 행동(what action), 장소(when), 때(when), 이유(why), 코드 다운로드(code downloading) 및 텍스트 주석(text annotation)으로 이루어지는 그룹에서 선택되는 적어도 하나의 특징 디스크립션에 의해 정의되는 디스크립션 레코드 생성 시스템.

청구항 6.

제4항에 있어서,

상기 시각적 특징들은 색채(color), 텍스처(texture), 위치(position), 크기(size), 형상(shape), 움직임(motion), 코드 다운로드 및 방위(orientation)로 이루어지는 그룹에서 선택되는 적어도 하나의 특징 디스크립션에 의해 정의되는 디스크립션 레코드 생성 시스템.

청구항 7.

제4항에 있어서,

상기 미디어 특징들은 파일 포맷(file format), 파일 크기(file size), 색채 표현(color representation), 해상도(resolution), 데이터 파일 위치(data file location), 저자(author), 창작(creation), 스케일러블 층(scalable layer), 코드 다운로드 및 양식 트랜스코딩(modality transcoding)으로 이루어지는 그룹에서 선택되는 적어도 하나의 특징 디스크립션에 의해 정의되는 디스크립션 레코드 생성 시스템.

청구항 8.

제1항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 시각적 특징 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션들을 생성하는 디스크립션 레코드 생성 시스템.

청구항 9.

제1항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 의미적 특징 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션들을 생성하는 디스크립션 레코드 생성 시스템.

청구항 10.

제1항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 미디어 특징 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션들을 생성하는 디스크립션 레코드 생성 시스템.

청구항 11.

제1항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션들을 생성하며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 디스크립션 레코드 생성 시스템.

청구항 12.

제1항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션들을 생성하며,

상기 이미지 객체 계층 디스크립션들은 복수의 계층적 레벨을 갖는 디스크립션 레코드 생성 시스템.

청구항 13.

제12항에 있어서,

복수의 계층적 레벨을 갖는 상기 이미지 객체 계층 디스크립션들은 클러스터링 계층(clustering hierarchies)들을 포함하는 디스크립션 레코드 생성 시스템.

청구항 14.

제13항에 있어서,

상기 클러스터링 계층들은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 관계들에 기초하며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 디스크립션 레코드 생성 시스템.

청구항 15.

제1항에 있어서,

상기 엔티티 관계 그래프 생성 프로세싱은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 관계들에 기초하여 상기 이미지 객체 디스크립션들의 엔티티 관계 그래프 디스크립션들을 생성하며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 디스크립션 레코드 생성 시스템.

청구항 16.

제1항에 있어서,

상기 이미지 객체 디스크립션들을 수신하여 부호화된 디스크립션 정보로 부호화하는 부호기(encoder)를 추가로 포함하며,

상기 데이터 저장 시스템은 상기 부호화된 디스크립션 정보를 상기 적어도 하나의 디스크립션 레코드로서 저장하도록 작용하는 디스크립션 레코드 생성 시스템.

청구항 17.

제1항에 있어서,

상기 이미지 객체 디스크립션들, 상기 이미지 객체 계층 디스크립션들 및 상기 엔티티 관계 그래프 디스크립션들은 이미지 디스크립션을 형성하도록 서로 결합되며,

상기 이미지 객체 디스크립션들을 수신하여 부호화된 디스크립션 정보로 부호화하는 부호기를 추가로 포함하며,

상기 데이터 저장 시스템은 상기 부호화된 디스크립션 정보를 상기 적어도 하나의 디스크립션 레코드로서 저장하도록 작용하는 디스크립션 레코드 생성 시스템.

청구항 18.

제17항에 있어서,

상기 부호기는 이진 부호기를 포함하는 디스크립션 레코드 생성 시스템.

청구항 19.

제17항에 있어서,

상기 부호기는 XML 부호기를 포함하는 디스크립션 레코드 생성 시스템.

청구항 20.

이미지 정보로부터 디스크립션 레코드(description record)를 생성하는 방법에 있어서,

상기 이미지 정보를 수신하는 단계;

상기 이미지 정보로부터 이미지 객체 디스크립션들을 생성하도록 이미지 객체 추출 프로세싱을 수행하여 상기 이미지 정보를 처리하는 단계;

이미지 객체 계층 디스크립션들을 생성하도록 객체 계층 구축 및 추출 프로세싱을 수행하여 상기 이미지 객체 디스크립션들을 처리하는 단계;

엔티티 관계 그래프 디스크립션들을 생성하도록 엔티티 관계 그래프 생성 프로세싱을 수행하여 상기 생성된 이미지 객체 디스크립션들을 처리하고, 상기 이미지 객체 디스크립션들, 상기 이미지 객체 계층 디스크립션들 및 상기 엔티티 관계 그래프 디스크립션을 포함하는 적어도 하나의 디스크립션 레코드가 상기 이미지 정보 내에 포함된 콘텐츠를 표현하도록 생성되는 단계; 및

상기 적어도 하나의 디스크립션 레코드를 저장하는 단계

를 포함하는 디스크립션 레코드 생성 방법.

청구항 21.

제20항에 있어서,

상기 이미지 객체 추출 프로세싱과 객체 계층 구축 및 추출 프로세싱 단계는 병렬로 실행되는 디스크립션 레코드 생성 방법.

청구항 22.

제20항에 있어서,

상기 이미지 객체 추출 프로세싱 단계는,

상기 이미지 정보 내의 각 이미지를 상기 이미지 내의 영역들로 분할하는 이미지 세그멘테이션 프로세싱; 및

하나 이상의 상기 영역들에 대해 하나 이상의 특징 디스크립션들을 생성하는 특징 추출 및 주석 프로세싱을 포함하며,

상기 생성된 이미지 객체 디스크립션들은 상기 하나 이상의 영역들에 대해 상기 하나 이상의 특징 디스크립션들을 포함하는 디스크립션 레코드 생성 방법.

청구항 23.

제22항에 있어서,

미디어 특징들, 시각적 특징들 및 의미적 특징들로 이루어지는 그룹에서 상기 하나 이상의 특징 디스크립션을 선택하는 단계를 포함하는 디스크립션 레코드 생성 방법.

청구항 24.

제23항에 있어서,

상기 의미적 특징들은 누가(who), 대상(what object), 행동(what action), 장소(when), 때(when), 이유(why), 코드 다운로드(code downloading) 및 텍스트 주석(text annotation)으로 이루어지는 그룹에서 선택되는 적어도 하나의 특징 디스크립션에 의해 정의되는 디스크립션 레코드 생성 방법.

청구항 25.

제23항에 있어서,

상기 시각적 특징들은 색채(color), 텍스처(texture), 위치(position), 크기(size), 형상(shape), 움직임(motion), 코드 다운로드 및 방위(orientation)로 이루어지는 그룹에서 선택되는 적어도 하나의 특징 디스크립션에 의해 정의되는 디스크립션 레코드 생성 방법.

청구항 26.

제23항에 있어서,

상기 미디어 특징들은 파일 포맷(file format), 파일 크기(file size), 색채 표현(color representation), 해상도(resolution), 데이터 파일 위치(data file location), 저자(author), 창작(creation), 스케일러블 층(scalable layer), 코드 다운로드 및 양식 트랜스코딩(modality transcoding)으로 이루어지는 그룹에서 선택되는 적어도 하나의 특징 디스크립션에 의해 정의되는 디스크립션 레코드 생성 방법.

청구항 27.

제20항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱 단계는 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 시각적 특징 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션을 생성하는 디스크립션 레코드 생성 방법.

청구항 28.

제20항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱 단계는 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 의미적 특징 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션들을 생성하는 디스크립션 레코드 생성 방법.

청구항 29.

제20항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱 단계는 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 미디어 특징 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션들을 생성하는 디스크립션 레코드 생성 방법.

청구항 30.

제20항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱 단계는 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션들을 생성하며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 디스크립션 레코드 생성 방법.

청구항 31.

제20항에 있어서,

상기 객체 계층 구축 및 추출 프로세싱 단계는 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 관계들에 기초하여 상기 이미지 객체 디스크립션들의 이미지 객체 계층 디스크립션들을 생성하며,

상기 이미지 객체 계층 디스크립션들은 복수의 계층적 레벨을 갖도록 형성되는 디스크립션 레코드 생성 방법.

청구항 32.

제31항에 있어서,

복수의 계층적 레벨을 갖는 상기 이미지 객체 계층 디스크립션들은 클러스터링 계층들을 포함하도록 형성되는 디스크립션 레코드 생성 방법.

청구항 33.

제32항에 있어서,

상기 클러스터링 계층들은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 관계들에 기초하도록 형성되며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 디스크립션 레코드 생성 방법.

청구항 34.

제20항에 있어서,

상기 엔티티 관계 그래프 생성 프로세싱 단계는 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 관계들에 기초하여 상기 이미지 객체 디스크립션들의 엔티티 관계 그래프 디스크립션들을 생성하며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 디스크립션 레코드 생성 방법.

청구항 35.

제20항에 있어서,

상기 이미지 객체 디스크립션들을 수신하여 부호화된 디스크립션 정보로 부호화하는 단계와, 상기 부호화된 디스크립션 정보를 상기 적어도 하나의 디스크립션 레코드로서 저장하는 단계를 추가로 포함하는 디스크립션 레코드 생성 방법.

청구항 36.

제20항에 있어서,

이미지 디스크립션들을 형성하기 위해 상기 이미지 객체 디스크립션들, 상기 이미지 객체 계층 디스크립션들 및 상기 엔티티 관계 그래프 디스크립션들을 결합하는 단계, 상기 이미지 객체 디스크립션들을 수신하여 부호화된 디스크립션 정보로 부호화하는 단계, 및 상기 부호화된 디스크립션 정보를 상기 적어도 하나의 디스크립션 레코드로서 저장하는 단계를 추가로 포함하는 디스크립션 레코드 생성 방법.

청구항 37.

제36항에 있어서,

상기 부호화 단계는 이진 부호화 단계를 포함하는 디스크립션 레코드 생성 방법.

청구항 38.

제36항에 있어서,

상기 부호화 단계는 XML 부호화를 포함하는 디스크립션 레코드 생성 방법.

청구항 39.

대응하는 이미지 정보 내에 포함된 이미지 콘텐츠를 표현하는 적어도 하나의 디스크립션 레코드를 갖는 디지털 정보를 포함하는 컴퓨터로 판독 가능한 매체에서 있어서,

상기 적어도 하나의 디스크립션 레코드는,

이미지 객체 추출 프로세싱을 이용하여 상기 이미지 정보로부터 생성된 하나 이상의 이미지 객체 디스크립션;

객체 계층 구축 및 추출 프로세싱을 이용하여 상기 생성된 이미지 객체 디스크립션들로부터 생성된 하나 이상의 이미지 객체 계층 디스크립션; 및

엔티티 관계 그래프 생성 프로세싱을 이용하여 상기 생성된 이미지 객체 디스크립션들로부터 생성된 하나 이상의 엔티티 관계 그래프 디스크립션

을 포함하는 컴퓨터로 판독 가능한 매체.

청구항 40.

제39항에 있어서,

상기 이미지 객체 디스크립션들, 상기 이미지 객체 계층 디스크립션들 및 상기 엔티티 관계 그래프 디스크립션들은 하나 이상의 특징 디스크립션들을 포함하는 컴퓨터로 판독 가능한 매체.

청구항 41.

제40항에 있어서,

상기 하나 이상의 특징 디스크립션들은 미디어 특징들, 시각적 특징들 및 의미적 특징들로 이루어지는 그룹에서 선택되는 컴퓨터로 판독 가능한 매체.

청구항 42.

제41항에 있어서,

상기 의미적 특징들은 누가(who), 대상(what object), 행동(what action), 장소(when), 때(when), 이유(why), 코드 다운로드(code downloading) 및 텍스트 주석(text annotation)으로 이루어지는 그룹에서 선택되는 적어도 하나의 특징 디스크립션에 의해 정의되는 컴퓨터로 판독 가능한 매체.

청구항 43.

제41항에 있어서,

상기 시각적 특징들은 색채(color), 텍스처(texture), 위치(position), 크기(size), 형상(shape), 움직임(motion), 코드 다운로드 및 방위(orientation)로 이루어지는 그룹에서 선택되는 적어도 하나의 특징 디스크립션에 의해 정의되는 컴퓨터로 판독 가능한 매체.

청구항 44.

제41항에 있어서,

상기 미디어 특징들은 파일 포맷(file format), 파일 크기(file size), 색채 표현(color representation), 해상도(resolution), 데이터 파일 위치(data file location), 저자(author), 창작(creation), 스케일러블 층(scalable layer), 코드 다운로드 및 양식 트랜스코딩(modality transcoding)으로 이루어지는 그룹에서 선택되는 적어도 하나의 특징 디스크립션에 의해 정의되는 컴퓨터로 판독 가능한 매체.

청구항 45.

제39항에 있어서,

상기 객체 계층 디스크립션은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 시각적 특징 관계들에 기초하는 컴퓨터로 판독 가능한 매체.

청구항 46.

제39항에 있어서,

상기 이미지 객체 계층 디스크립션은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 의미적 특징 관계들에 기초하는 컴퓨터로 판독 가능한 매체.

청구항 47.

제39항에 있어서,

상기 이미지 객체 계층 디스크립션은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 미디어 특징 관계들에 기초하는 컴퓨터로 판독 가능한 매체.

청구항 48.

제39항에 있어서,

상기 이미지 객체 계층 디스크립션은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 관계들에 기초하며, 복수의 계층적 레벨을 갖는 컴퓨터로 판독 가능한 매체.

청구항 49.

제48항에 있어서,

복수의 계층적 레벨을 갖는 상기 이미지 객체 계층 디스크립션들은 클러스터링 계층들을 포함하는 컴퓨터로 판독 가능한 매체.

청구항 50.

제49항에 있어서,

상기 클러스터링 계층들은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체의 관계들에 기초하며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 컴퓨터로 판독 가능한 매체.

청구항 51.

제39항에 있어서,

상기 엔티티 관계 그래프 디스크립션들은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 관계들에 기초하며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 컴퓨터로 판독 가능한 매체.

청구항 52.

제39항에 있어서,

상기 이미지 객체 디스크립션들은 부호화된 디스크립션 정보의 형태인 컴퓨터로 판독 가능한 매체.

청구항 53.

제39항에 있어서,

상기 이미지 객체 디스크립션들, 상기 이미지 객체 계층 디스크립션들 및 상기 엔티티 관계 그래프 디스크립션들은 부호화된 디스크립션 정보의 형태로 서로 결합되는 컴퓨터로 판독 가능한 매체.

청구항 54.

제53항에 있어서,

상기 부호화 디스크립션 정보는 이진 부호화 정보의 형태인 컴퓨터로 판독 가능한 매체.

청구항 55.

제53항에 있어서,

상기 부호화 디스크립션 정보는 XML 부호화 정보의 형태인 컴퓨터로 판독 가능한 매체.

청구항 56.

제12항에 있어서,

복수의 계층적 레벨을 갖는 상기 이미지 객체 계층 디스크립션들은 복수 레벨의 추상화 계층들(abstraction hierarchies)을 포함하도록 형성되는 디스크립션 레코드 생성 시스템.

청구항 57.

제56항에 있어서,

상기 복수 레벨의 추상화 계층들은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 관계들에 기초하도록 형성되며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 디스크립션 레코드 생성 시스템.

청구항 58.

제31항에 있어서,

복수의 계층적 레벨을 갖는 상기 이미지 객체 계층 디스크립션들은 복수 레벨의 추상화 계층들을 포함하도록 형성되는 디스크립션 레코드 생성 방법.

청구항 59.

제58항에 있어서,

상기 복수 레벨의 추상화 계층들은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 관계들에 기초하도록 형성되며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 디스크립션 레코드 생성 방법.

청구항 60.

제48항에 있어서,

복수의 계층적 레벨을 갖는 상기 이미지 객체 계층 디스크립션들은 복수 레벨의 추상화 계층들을 포함하도록 형성되는 컴퓨터로 판독 가능한 매체.

청구항 61.

제60항에 있어서,

상기 복수 레벨의 추상화 계층들은 상기 이미지 객체 디스크립션들에 의해 표현되는 이미지 객체들의 관계들에 기초하도록 형성되며,

상기 관계들은 시각적 특징 관계들, 의미적 특징 관계들 및 미디어 특징 관계들로 이루어지는 그룹에서 선택되는 컴퓨터로 판독 가능한 매체.

청구항 62.

제3항 또는 제4항에 있어서,

상기 하나 이상의 특징 디스크립션은 코드 다운로드의 편의를 위해 추출 및 매칭 코드에 대한 포인터들(pointers)을 포함하는 디스크립션 레코드 생성 시스템.

청구항 63.

제22항 또는 제23항에 있어서,

상기 하나 이상의 특징 디스크립션은 코드 다운로드의 편의를 위해 추출 및 매칭 코드에 대한 포인터들을 포함하는 디스크립션 레코드 생성 방법.

청구항 64.

제40항 또는 제41항에 있어서,

상기 하나 이상의 특징 디스크립션은 코드 다운로드의 편의를 위해 추출 및 매칭 코드에 대한 포인터들을 포함하는 컴퓨터로 판독 가능한 매체.

명세서

기술분야

본 발명은 멀티미디어 정보를 기술하는 기술에 관한 것으로, 더욱 자세하게는 이미지 정보 및 그러한 정보의 콘텐츠(content)를 기술하는 기술에 관한 것이다.

배경기술

전세계적인 인터넷 및 지역적인 네트워크의 광범위한 이용의 성숙과 더불어 소비자들과 회사들은 디지털 멀티미디어(digital multimedia) 정보를 점점 접근 가능하게 되어 왔다. 그에 따라, 이러한 증대하는 가공되지 않은 정보집단(mass of raw information)으로부터 유용한 정보를 추출할 수 있도록 하는 디지털 멀티미디어 정보를 처리, 필터링, 검색 및 조직화하는 시스템의 개발이 점점 더 중요해지고 있다.

인스턴트 애플리케이션(instant application)을 파일링(filing)하는 경우에, 소비자들과 회사들이 텍스트 정보(textual information)를 검색할 수 있도록 하는 솔루션(solutions)이 있다. 실제로, yahoo.com, goto.com, excite.com 및 다른 사이트들이 제공하는 수많은 텍스트 기반 검색 엔진(text-based search engines)을 월드 와이드 웹(World Wide Web)에서 이용할 수 있으며, 이들은 그러한 정보 검색 기술에 대한 요구의 중요성을 나타내는 웹사이트들 가운데 가장 많이 방문되는 사이트들이다.

공교롭게도, 멀티미디어 콘텐츠(multimedia content)의 존재에 대한 디스크립션(description)이 일반적으로 인식되지 않는 경우에, 상기한 것은 멀티미디어 콘텐츠에 대해서는 진실이 아니다. 이점에 있어서, 화상(picture)에 포함된 비디오 객체들(video objects)의 색채(color) 정보, 텍스처(texture) 정보 및 형상(shape) 정보와 같은 특징들을 사용하여 사용자가 화상을 검색할 수 있도록 해주는 멀티미디어 데이터베이스를 제공하기 위한 시도들이 있어 왔다. 그러나, 20세기를 마감하는 시점에서, 멀티미디어 콘텐츠의 존재에 대한 디스크립션이 널리 인식되지 않는 경우, 인터넷 또는 다른 지역적이거나 국부적인 네트워크에서 멀티미디어 콘텐츠를 일반 검색하는 것은 아직 불가능하다. 또한, 멀티미디어 콘텐츠에 대한 검색의 필요는 데이터베이스에 한정되는 것이 아니라 디지털 방송 텔레비전과 멀티미디어 전화(telephony) 등의 다른 애플리케이션들로 확장한다.

한 산업계에서 그러한 표준 멀티미디어 디스크립션 프레임워크(framework)를 개발하기 위한 폭넓은 시도가 MPEG(Motion Expert Group)의 MPEG-7 표준화 노력을 통해 있어 왔다. 1996년 10월에 발표된 MPEG-7은 멀티미디어 검색, 필터링, 브라우징(browsing) 및 요약(summarization)과 같은 콘텐츠 중심(content-focused) 애플리케이션을 촉진하기 위하여 멀티미디어 데이터에 대한 콘텐츠 디스크립션의 표준화를 지향한다. MPEG-7 표준의 목적에 대한 보다 완전한 기술(記述)은 표준화를 위한 국제 기구의 문서 ISO/IEC JTC1/SC29/WG11 N2460(Oct. 1998)에 포함되어 있으며, 그 내용은 본 명세서에 참조되어 본 명세서의 일부를 이룬다.

MPEG-7 표준은 디스크립터의 구조(descriptor structures)("디스크립션 스킴들(description schemes)"로서 참조됨) 및 다양한 유형의 멀티미디어 정보를 기술하기 위한 그들의 관계는 물론, 디스크립터들의 표준 집합(standard set)을 명기하는 것을 목적으로 한다. 또한 MPEG-7은 디스크립터들에 대한 "디스크립션 스킴들" 및 그들의 관계는 물론, 다른 디스크립터들을 규정하기 위한 표준화 방법들을 제안한다. 이 디스크립션, 즉 디스크립터 및 디스크립션 스킴들의 조합은 사용자가 관심 있는 자료(material)에 대해 신속하고 효율적인 검색과 필터링을 할 수 있도록 콘텐츠 그 자체와 연관될 것이다. 또한 MPEG-7은 디스크립션 스킴을 명기하기 위한 언어, 즉 디스크립션 정의 언어(Description Definition Language, DDL)의 표준화 및 멀티미디어 콘텐츠 디스크립션의 이진 부호화를 위한 스킴들의 표준화를 제안한다.

인스턴트 애플리케이션을 파일링하는 경우, MPEG은 장래에 MPEG-7 표준으로의 통합을 위해 필요한 디스크립션 스킴들을 최적으로 구현할 기술을 위해 권유하는 제안이다. 그러한 최적화된 디스크립션 스킴들을 제공하기 위하여, 세가지 다른 멀티미디어 애플리케이션 협정을 고려할 수 있다. 이것들은 분산 프로세싱 시나리오(distributed processing scenario), 콘텐츠 교환 시나리오 및 멀티미디어 콘텐츠의 개인화된 뷰잉(personalized viewing)을 허용하는 포맷이다.

분산 프로세싱과 관련하여, 디스크립션 스킴은 모든 플랫폼(platform), 모든 공급자(vendor) 및 모든 애플리케이션과 관계없이 멀티미디어 자료의 디스크립션은 서로 교환하는 능력을 제공하여야 하며, 이것은 멀티미디어 콘텐츠의 분산 프로세싱을 가능하게 할 것이다.

상호 운용 가능한(interoperable) 콘텐츠 디스크립션들의 표준화는 멀티미디어 프로세서들, 편집기들, 검색 시스템들, 필터링 에이전트들 등과 같은 여러 분산 애플리케이션에 플러그인 될 수 있는 여러 소스(sources)로부터의 데이터를 의미할 것이다. 이 애플리케이션들 중 몇몇은 표준화된 멀티미디어 데이터 디스크립션과 함께 작용할 수 있는 멀티미디어 도구의 제공자들인 하위 산업계(sub-industry)를 이루는 제3자에 의해 제공될 수 있다.

사용자는 콘텐츠 및 소정의 저레벨(low-level) 프로세싱 또는 고레벨(high-level) 프로세싱에 의해 얻은 콘텐츠와 관련 인덱싱 데이터(indexing data)를 다운로드하기 위해 여러 콘텐츠 제공자의 웹사이트들에 접근할 수 있어야 하고, 나아가 사용자의 개인적인 관심사에 따라, 이질적인 데이터 디스크립션을 특정 방식으로 조작하는 도구들(예: 자바 애플릿(Java applets))을 다운로드하기 위해 몇몇 도구 제공자들의 웹사이트에 접근할 수 있어야 한다. 이러한 멀티미디어 도구의 한 예가 비디오 편집기(video editor)일 것이다. MPEG-7을 따르는(compliant) 비디오 편집기는 각 비디오와 관련된 디스크립션이 MPEG-7을 따르는 경우, 다양한 소스들로부터 비디오 콘텐츠의 조작 및 처리가 가능하게 될 것이다. 각 비디오는 카메라 움직임(motion), 장면 컷(scene cuts), 주석들(annotations) 및 객체 세그먼테이션들(object segmentations)과 같은 세부 디스크립션(description detail)의 정도의 변화가 따를 수 있다.

두 번째 시나리오는 상호 운용 가능한 콘텐츠 디스크립션 표준이 이질적인 멀티미디어 데이터베이스들 사이의 멀티미디어 콘텐츠의 교환이기 때문에 매우 유익할 것이다. MPEG-7은 존재하는 멀티미디어 자료의 디스크립션을 표현, 교환, 번역 및 재사용하는 수단을 제공하는 것을 지향한다.

현재, TV 방송국들, 라디오 방송국들 및 다른 콘텐츠 제공업자들은 막대한 양의 멀티미디어 자료를 처리하고 저장한다. 이러한 자료는 현재 텍스트 정보 및 독점적인(proprietary) 데이터베이스들 사용하여 수동으로 기술된다. 상호 운용 가능한 콘텐츠 디스크립션이 없는 경우, 콘텐츠 사용자들은 각 방송국이 사용한 디스크립션을 그 자신의 독점적인 스킴으로 수동 번역하기 위해 인력 투자가 필요하다. 멀티미디어 콘텐츠 디스크립션의 교환은 모든 콘텐츠 제공자들이 동일한 콘텐츠 디스크립션 스킴들을 채택한다면 가능할 수 있다. 이것은 MPEG-7의 목적 중 하나이다.

마지막으로, 디스크립션 스킴들을 사용하는 멀티미디어 플레이어 및 뷰어들은 사용자들에게 사용자들이 형성한 데이터의 멀티플 뷰(multiple views)와 같은 혁신적인 능력(capabilities)을 갖도록 제공하여야 한다.

이상의 예는 MPEG-7에 기초한 표준화된 방식으로 생성된(delivered) 충분히 구조화된 데이터에 대해 사용 가능하다는 점에서 단지 힌트일 뿐이다. 공교롭게도, 현시점에서 분산 처리, 콘텐츠 교환 또는 개인화된 뷰잉 시나리오를 총체적으로 만족시킬 수 있는 이용 가능한 종래 기술은 없다. 특히, 종래 기술은 일반적인 특성이나 의미 관계(semantic relationships)에 기초한 멀티미디어 정보에 포함된 콘텐츠를 포착(capturing)하기 위한 기술을 제공하거나, 또는 그러한 콘텐츠를 조직화하기 위한 기술을 제공하기에는 부족하다. 따라서, 일반적인 멀티미디어 정보를 위한 효율적인 콘텐츠 디스크립션 스킴들에 대한 기술이 필요하다.

발명의 상세한 설명

본 발명은 이미지 콘텐츠를 위한 디스크립션 시스템을 제공하는 것을 목적으로 한다.

본 발명의 다른 목적은 이미지 콘텐츠를 위한 확장 가능한 디스크립션 시스템을 제공하는 것이다.

본 발명의 또 다른 목적은 이미지 콘텐츠를 위한 스케일러블(scalable)한 디스크립션 시스템을 제공하는 것이다.

본 발명의 또 다른 목적은 MPEG-7 표준과 같은 소정의 제안된 미디어 표준들의 요구조건을 충족시키는 이미지 콘텐츠를 위한 디스크립션 시스템을 제공하는 것이다.

또한 본 발명은 이미지 콘텐츠를 기술하는 방법을 제공하는 것을 목적으로 한다.

본 발명의 다른 목적은 확장 가능한 이미지 콘텐츠 디스크립션 방법을 제공하는 것이다.

본 발명의 또 다른 목적은 스케일러블한 이미지 콘텐츠 디스크립션 방법을 제공하는 것이다.

본 발명의 또 다른 목적은 MPEG-7 표준과 같은 소정의 제안된 미디어 표준들의 요구조건을 충족시키는 이미지 콘텐츠 디스크립션 방법을 제공하는 것이다.

본 발명의 추가적인 목적들, 특징들 및 이점은 본 발명의 실시예를 나타내는 첨부 도면들과 이하의 상세한 설명에 의해 명백해 질 것이다.

실시예

도면 전체에, 명백히 규정되어 있지 않든 동일한 도면 부호 및 문자(character)는 동일한 특성, 요소(elements), 구성(components) 또는 도시된 실시예의 일부를 표시한다. 또한, 이제 본 발명을 도면을 참조하여 상세하게 설명하는 동안에 예시한 실시예들에 관하여도 설명될 것이다. 첨부된 청구범위에 의해 규정되는 바와 같은 본 발명의 사상 및 범위를 벗어나지 않고, 설명된 실시예들에 대한 변경 및 변형이 가능할 것이다.

본 발명은 이미지에 대한 디스크립션 스킴(Description Scheme, DS)을 구성하며, 여기서 일반적인 이미지 데이터를 나타내는 간단하지만 강력한 구조들이 사용된다. 비록 본 발명의 디스크립션 스킴은 이미지 콘텐츠를 기술하는 모든 유형의 표준과 함께 사용될 수 있지만, 본 발명의 바람직한 실시예는 MPEG-7 표준과 함께 사용된다. 비록 디스크립션 정의 언어(Description Definition Language, DDL)가 본 발명의 DS를 구현하기 위해 사용될 수 있지만, 본 발명의 바람직한 실시예에는 특별히 월드 와이드 웹 애플리케이션을 위해 개발된 SGML(Standard Generalized Markup Language, ISO 8879)의 합리화된 부분집합인 XML(eXtensible Markup Language)을 이용한다. SGML은 문서에 사용된 태그 셋(tag set)과 이 태그들을 나타내는 구조적인 관계를 명기함으로써 문서가 그 자신의 문법을 기술한다는 의미에서 문서들이 자체 기술(self-describing)할 수 있도록 한다. XML은 언어 내에 기본적인(key) SGML 이점을 계속 유지하며, 완전한 SGML 보다 훨씬 더 배우기 쉽고, 사용하기 쉬우며 구현하기 쉽도록 설계된다. XML에 대한 완전한 기술은 월드 와이드 웹 컨소시엄의 XML에 대한 웹페이지, <http://www.w3.org/XML>에서 구할 수 있으며, 그 내용은 본 명세서에 참조되어 본 명세서의 일부를 이룬다.

본 발명의 디스크립션 스킴에서 사용하는 이미지 특징을 부여하는 주 구성요소는 객체들, 특징 분류들(feature classifications), 객체 계층들, 엔티티 관계 그래프들, 복수 레벨의 추상화(abstraction), 코드 다운로드 및 양식 트랜스코딩(modality transcoding)이며, 이 모두는 이하에서 더욱 상세하게 설명될 것이다. 본 발명의 디스크립션 스킴에서, 이미지 문서는 객체들과 객체들 사이의 관계들의 집합으로 표현된다. 각 객체는, 일반적으로 미디어 특징들(media features), 시각적 특징들 및 의미적 특징들 등의 카테고리들로 그룹을 만드는 하나 이상의 관련된 특징을 가질 수 있다. 각 특징은 외부 추출(external extration) 및 유사성 매칭코드(similarity matching code)를 지시(pointing)함으로써 코드 다운로드를 촉진시킬 수 있는 디스크립터들을 포함할 수 있다. 객체들 사이의 관계는 객체 계층 및 엔티티 관계 그래프를 이용하여 기술할 수 있다. 객체 계층들은 또한 복수 레벨의 추상화를 포함할 수 있다. 양식 트랜스코딩은, 사용자 단말기들이 상이한 솔루션 및/또는 상이한 양식으로 동일한 이미지 콘텐츠를 수신하는 상이한 능력들(예를 들어, 팜파일럿(palmpilots), 셀룰러 폰 또는 다른 형태의 개인용 컴퓨터와 같은)을 가질 수 있도록 한다.

전술한 바와 같이, 본 발명의 이미지 디스크립션 시스템의 바람직한 실시예는 MPEG-7 표준과 함께 사용된다. 이 표준에 따라, 본 발명의 바람직한 실시예는 다른 차원을 따라 정의될 수 있는 이미지 콘텐츠의 여러 레벨을 기술함에 있어, 기본 엔티티로서 객체들을 사용한다. 예를 들어, 객체들은 이미지 영역들이나 이미 영역들의 그룹들을 기술하는 데 사용될 수 있다. 고레벨 객체들은 차례로 의미적인 또는 시각적인 특징들에 기반한 원시 객체 그룹들을 기술하는 데 사용될 수 있다. 게다가, 상이한 유형의 특징들은 상이한 레벨의 객체들과 관련하여 사용될 수 있다. 이를테면, 시각적인 특징들은 이미지 콘텐츠 내의 물리적인 구성요소에 대응하는 객체에 적용될 수 있으며, 이에 반하여 의미적인 특징들은 객체의 모든 레벨에 적용될 수 있다.

또한, 본 발명의 이미지 디스크립션 시스템은 유연성(flexibility), 확장 가능성(extensibility), 확장성(scalability) 및 사용의 편의성을 제공한다. 유연성 강화를 위하여, 본 발명은 인스턴스를 생성하는(instantiated) 이미지 디스크립션 시스템의 일부가 효율적인 특징들의 범주화(categorization)와 효율적인 계층 클러스터링을 위하여 객체들의 클러스터링을 이용할 수 있도록 하며, 또한 효율적인 링킹(liking), 임베딩(embedding) 및 외부 특징 디스크립터들과 실행 코드의 다운로드를 지원할 수 있도록 한다. 또한 본 발명은 다른 도메인(domains)에 대한 새로운 요소(elements)를 생성시키는 데 사용될 디스크립션 스킴에서 정의된 요소를 허용함으로써 확장 가능성을 제공한다. 확장성은 객체 계층들을 사용하는 기준(criteria)의 모든 임의 집합(arbitrary set)에 기초한 복수의 추상화 레벨을 정의하는 본 발명의 능력에 의해 제공된다. 이 기준들은 시각적 특징들(예를 들어, 크기 및 색채), 의미적인 관련성(semantic relevance)(예를 들어 사용자 관심사항 프로파일과의 관련성) 및/또는 서비스 품질(예를 들어, 미디어 특징들)에 관하여 명기될 수 있다. 본 발명은 구성요소들, 즉 객체들, 특징 클래스들, 객체 계층들 및 엔티티 관계 그래프들의 최소 집합을 명기하기 때문에 사용에 편리하다. 또한, 객체들과 특징들은 모듈러(modular) 및 유연한(flexible) 방식으로 추가될 수 있다. 게다가, 상이한 유형의 객체 계층들 및 엔티티 관계 그래프들은 유사한 방식으로 각각 정의될 수 있다.

본 발명의 이미지 디스크립션 시스템 하에서, 이미지는 객체 계층들 및 엔티티 관계 그래프들에 의해 다른 객체와 관련되는 이미지 객체들의 집합으로 표현된다. 이 객체들은 외부 추출 및 유사성 매칭 코드와 링크될 수 있는 복수의 특징을 가질 수 있다. 이 특징들은 예를 들어 미디어 특징들, 시각적 특징들 그리고 의미적 특징들로 범주화된다. 이미지 객체들은 다수의 상이한 객체 계층들로 조직화될 수 있다. 둘 이상의 객체들 사이의 비 계층적인 관계들(non-hierarchical relationships)은 하나 이상의 상이한 엔티티 관계 그래프를 사용하여 기술될 수 있다. 큰 이미지에 포함된 객체들의 경우, 이러한 객체들의 클러스터링 및 뷰잉에 있어 복수 레벨의 추상화는 객체 계층들을 이용하여 구현될 수 있다. 이러한 이미지들의 클러스터링 및 뷰잉에 있어 복수 레벨의 추상화는, 예를 들어 미디어 특징들, 시각적 특징들 및/또는 의미적 특징들에 기초할 수 있다. 미디어 특징의 한 예에서는 사용자가 만족스러운 양식 및 해상도(resolutions) 내의 동일한 이미지 콘텐츠에 접근하는 상이한 단말기 사양(terminal specifications)을 가질 수 있도록 하는 양식 트랜스코딩을 포함한다.

본 발명의 이미지 디스크립션 시스템의 특징들 및 작용(operation)은 이하에서 더욱 자세하게 나타날 것이다. 도 1a 내지 도 1c는 본 발명의 이미지 디스크립션 시스템에 따른 이미지 예에 대한 디스크립션의 일례를 도시한다. 도 1a는 이미지 객체들의 집합의 일례와 이 객체들에 대응하는 객체 특징들의 예를 도시한다. 더욱 상세하게는, 도 1a는 도시된 사진 예에 대한 특징들(10)의 예는 물론이고, O0(8)(즉, 예시된 사진 전체)에 포함된 이미지 객체(2)(즉, O1)("사람 A"), O2(6)("사람 B") 및 O3(4)("사람들")을 도시한다. 도 1b는 도 1a에 도시한 이미지 객체들에 대한 공간 객체 계층의 일례를 나타낸 것이며, O0(8)(사진 전체)은 O1(2)("사람 A") 및 O2(6)("사람 B")를 포함하도록 도시된다. 도 1c는 엔티티 관계(E-R) 그래프를 나타낸 것이며, O1(2)("사람 A")은 그 왼쪽에 위치하고 약수하는 O2(6)("사람 B")로서 특징이 부여된다.

도 2는 종래의 UML(Unified Modeling Language) 포맷과 표기법(notation)을 이용하는 본 발명의 이미지 디스크립션 시스템을 도표로 표현한 일례를 도시한다. 특히, 도 2에 도시된 다이아몬드형 기호는 합성 관계(composition relationship)를 나타낸다. 각 요소와 관련된 범위(range)는 합성 관계의 빈도(frequency)를 나타낸다. 특히, 술어(nomenclature)

"0...*" 는 "0(zero) 이상"을 의미하고, 술어 "1...*" 는 "1 이상"을 의미한다. 다음의 논의에서, 문자 "<" 및 ">" 사이에 나타나는 텍스트는 이하에 나타내는 XML의 바람직한 실시예에서 참조된 요소들의 특징 부여를 의미한다. 도 2에 도시한 바와 같은 본 발명의 이미지 디스크립션 시스템에서, 이미지 디스크립션을 나타내는 이미지 요소(22)(`<image>`)는 이미지 객체 집합 요소(24)(`<image_object_set>`)를 포함하고, 또 하나 이상의 객체 계층 요소들(26)(`<object_hierarchy>`) 및 하나 이상의 엔티티 관계 그래프들(28)(`<entity_relation_graph>`)을 포함할 수 있다. 각각의 이미지 객체 집합 요소(24)는 하나 이상의 이미지 객체 요소들(30)을 포함할 수 있다. 각각의 이미지 객체 요소(30)는 미디어 특징 요소들(36), 시각적 특징 요소들(38) 및/또는 의미적 특징 요소들(40)과 같은 하나 이상의 특징을 포함할 수 있다. 각각의 객체 계층 요소(26)는 객체 노드 요소(32)를 포함하고, 그 각각은 차례로 하나 이상의 추가적인 객체 노드 요소(32)를 포함한다. 엔티티 관계 그래프(28) 각각은 하나 이상의 엔티티 관계 요소(34)를 포함한다. 각 엔티티 관계 요소(34)는 차례로 관계 요소(44)를 포함하고, 또한 하나 이상의 엔티티 노드 요소(42)를 포함할 수 있다.

객체 계층 요소(26)는 엔티티 관계 그래프(28)의 특수한 경우이다. 본 발명의 이미지 디스크립션 시스템의 바람직한 실시예는 엔티티 관계 그래프(28) 외에 객체 계층 요소들(26)을 포함하는데, 이는 객체 계층 요소(26)가 검색에 있어 엔티티 관계 그래프(28)보다 더 효율적인 구조이기 때문이다. 또한 객체 계층 요소(26)는 복합 객체(composite objects)를 정의하는 가장 자연스런 방법이며, MPEG-4 객체들은 계층적 구조들을 사용하여 구성된다.

유연성과 보편성(generality)을 극대화하기 위하여, 본 발명의 이미지 디스크립션 시스템은 객체들의 정의를 객체들 사이의 관계를 기술하는 구조들과 분리시킨다. 따라서, 동일한 객체는 상이한 객체 계층들(26) 및 엔티티 관계 그래프들(28)에

서 나타날 수 있다. 이것은 하나 이상의 객체 계층(26) 및/또는 엔티티 관계 그래프(28)에서 나타나는 객체들에 대한 특징들의 바람직하지 못한 복사(duplication)를 방지한다. 또한, 객체는 객체 계층(26) 또는 엔티티 관계 그래프(28)와 같은 임의의 관계 구조(relational structure)에 포함될 객체에 대한 요구 없이도 정의될 수 있으며, 객체들의 추출 및 객체들간의 관계는 상이한 단계에서 시행될 수 있으므로 이미지 콘텐츠의 분산 프로세싱을 허용한다.

도 1a 내지 도 1c 및 도 2를 참조하면, 이미지 객체(30)는 이미지 중의 하나 이상의 임의 영역을 참조하므로 공간 내에서 연속적이거나 불연속적일 수 있다. 도 1a 내지 도 1c에서, O1(2)("사람 A"), O2(6)("사람 B") 및 O0(8)(즉 사진)은 오직 하나의 연속적인 관련 영역을 갖는 객체들이다. 한편, O3(4)("사람들")은 공간 내에서 서로 분리되는 복수 영역으로 구성되는 객체의 일례이다. 전역(global) 객체는 전체 이미지에 공통인 특징들을 포함하는데 반해, 지역(local) 객체는 그 이미지의 특정 부분에 대한 특징들만 포함한다. 따라서, 도 1a 내지 도 1c에서, O0(8)은 도시된 전체 이미지를 나타내는 전역 객체인 반면, O1(2), O2(6) 및 O3(4)은 전체 이미지 내에 포함된 사람 또는 사람들을 각각 나타내는 지역 객체이다.

본 발명과 관련하여 사용될 수 있는 여러 형태의 객체들에는 색채 또는 텍스처와 같은 시각적인 특징들에 의해 정의되는 객체인 시각적 객체들; 미디어 객체들; 의미적 객체들; 및 의미적, 시각적 그리고 미디어 특징들의 조합에 의해 정의된 객체들을 포함한다. 따라서, 객체의 유형은 그 객체를 기술하는 데 사용된 특징들에 의해 정해진다. 그 결과, 새로운 형태의 객체들이 필요에 따라 추가될 수 있다. 또한 상이한 유형의 객체들은 MPEG-7 표준에 의해 지원되는 계승 관계(inheritance relationships)를 사용함으로써 이러한 일반적인 객체들로부터 생성될 수 있다.

도 2에 도시한 바와 같이, 이미지에 기술된 모든 이미지 객체 요소들(30)의 집합(<image_object>)은 이미지 객체 집합 요소(24)(<image_object_set>)와 함께 포함된다. 각 이미지 객체 요소(30)는 이미지 디스크립션 내에 유일한 식별자(identifier)를 가질 수 있다. 이 식별자와 객체 유형(예를 들어, 지역 또는 전역)은 각각 객체 요소 ID 및 유형 속성(attributes)으로 표현된다. 도 1a 내지 도 1c에 도시된 이미지를 기술하기 위한 객체들의 집합 예를 구현하는 일례를 아래에 XML 리스트로 나타낸다. 이하에 나타내는 모든 XML 리스트에서, 문자"<!--" 와 ".>" 사이에 나타나는 텍스트는 XML 코드에 대한 주석을 의미한다.

```
<image_object_set>
  <image_object id="O0" type="GLOBAL"> </image_object> <!-- Photograph -->
  <image_object id="O1" type="LOCAL"> </image_object> <!-- Person A -->
  <image_object id="O2" type="LOCAL"> </image_object> <!-- Person B -->
  <image_object id="O3" type="LOCAL"> </image_object> <!-- People -->
</image_object_set>
```

도 2에 도시된 바와 같이, 예를 들어 이미지 객체들(30)은 그것들의 특징들에 의해 전달되는 정보에 따라 함께 그룹 특징들인 세 가지 특징 클래스 요소들을 포함할 수 있다. 이러한 특징 클래스 요소들의 예에는 미디어 특징들(36)(<img_obj_media_features>), 시각적 특징들(38)(<img_obj_visual_features>) 및 의미적 특징들(<img_obj_semantic_features>)을 포함한다. 표 1은 이러한 각 특징 클래스들에 대한 특징들의 일례를 리스트로 나타낸다.

표 1 : 특징 클래스들 및 특징들의 예	
특징 클래스	특징들
의미적	텍스트 주석, 누가(who), 대상(what object), 행동(what action), 이유(why), 때(when), 장소(when)
시각적	색채(color), 텍스처(texture), 위치(position), 크기(size), 형상(shape), 방위(orientation)
미디어	파일 포맷(file format), 파일 크기(file size), 색채 표현(color representation), 해상도(resolution), 데이터 파일 위치(data file location), 저자(author), 창작(creation)

이미지 객체 요소(30)의 특징 클래스들에 포함된 각 특징 요소는 MPEG-7 표준에 따라 디스크립터들을 포함할 것이다. 아래의 표 2는 표 1에 나타난 소정의 시각적 특징들의 예와 관련될 수 있는 디스크립터들의 예를 나타낸다. 또한 표 2에 나타난 것과 같은 특정 디스크립터들은 외부 추출 및 유사성 매칭 코드와의 링크를 포함할 수 있다. 비록 표 1 및 표 2가 특징들 및 디스크립터들의 예를 나타내었지만, 본 발명의 이미지 디스크립션 시스템은 확장 가능한 방식 및 모듈러 방식으로 각 객체에 대한 특징들 및 디스크립터를 얼마든지 포함할 수 있다.

표 2 : 시각적 특징들 및 관련 디스크립터들의 예	
특징 클래스	특징들
색채	색채 히스토그램, 두드러진 색채(dominant color), 색채 코히어런스 벡터(color coherence vector), 비주얼 스프라이트 색채(visual sprite color)
텍스처	Tamura, MSAR, 에지 방향 히스토그램, DCT 계수 에너지들, 비주얼 스프라이트 텍스처
형상	바운딩 박스(bounding box), 이진 마스크(binary mask), 크로마 키(chroma key), 다각형 형상, 푸리에 형상, 경계, 크기, 대칭, 방위(orientation)

아래에 나타낸 XML의 예는 특징들과 디스크립터들이 이미지 객체(30) 내에 포함되도록 정의될 수 있는 방법의 일례를 나타낸 것이다. 특히, 아래의 예는 도 1a 내지 도 1c에 도시한 전역 객체 OO과 관련된 특징들(10), 즉 두 개의 의미적 특징("장소" 및 "때"), 한 개의 미디어 특징("파일 포맷") 그리고 한 개의 시각적 특징("색채 히스토그램(color histogram)" 디스크립터를 갖는 "색채")의 예를 정의한다. 객체는 아래의 예에서 나타낸 바와 같이 각각의 의미적 카테고리 내의 다른 개념(<concept>)에 의해 기술될 수 있다.

```
<image_object id="OO" type="GLOBAL"> <!-- Global object: Photograph -->
  <img_obj_semantic_features>
    <where>
      <concept> Columbia University, NYC </concept>
      <concept> Outdoors </concept>
    </where>

    <when> <concept> 5/31/99 </concept> </when>
  </img_obj_semantic_features>
  <img_obj_media_features>
    <file_format> JPG </file_format>
  </img_obj_media_features>
  <img_obj_visual_features>
    <color>
      <color_histogram>
        <value format="float[166]"> .3 .03 .45 ... </value>
      </color_histogram>
    </color>
  </img_obj_visual_features>
</image_global_object>
```

도 2에 도시한 바와 같이, 본 발명의 이미지 디스크립션 시스템에서 객체 계층 요소(26)는 미디어 특징들(36), 시각적 특징들(38), 의미적 특징(40) 또는 이들의 임의 조합과 같은 상이한 기준들에 기초하여 이미지 객체 집합(24)의 이미지 객체들(30)을 조직화하는 데 사용될 수 있다. 각 객체 계층 요소(26)는 링크(33)를 경유하여 이미지 객체 집합(24)의 이미지 객체 요소(30)를 참조하는 객체 노드들(32)의 트리(tree)를 구성한다.

객체 계층(26)은 하나 이상의 자식 노드에서 부모 노드로의 견제 관계(containment relation)를 포함한다. 이 견제 관계는, 예를 들어 미디어 특징들(36), 시각적 특징들(38) 및/또는 의미적 특징들(40)과 같은 사용되고 있는 특정 객체 특징들에 의존하는 다수의 상이한 유형일 수 있다. 예를 들어, 도 1b에 도시한 공간 객체 계층은 시각적 특징, 즉 공간 위치(spatial)와 관련하여 생성되기 때문에 시각적인 견제를 기술한다. 도 3a 및 도 3b는 두 개의 추가적인 객체 계층의 예를 도시한다. 특히, 도 3a는 도 1a에 도시한 이미지 객체에 대한 계층의 일례를 표 1에 나타낸 바와 같은 "누구(who)" 의미 특징에 기초하여 나타낸다. 따라서, 도 3a에서 O3(4)("사람들")는 O1(2)("사람 A") 및 O2(6)("사람 B")를 포함하도록 도시된다. 도 3b는 표 1에 나타낸 바와 같은 색채 및 형상 시각적 특징들의 예에 기초한 계층의 일례를 도시한다. 도 3b에서, O7(46)("피부 톤 (skin tone) & 형상")은 예를 들어 어떤 특정된 색채 및 형상 제약들(constraints)을 충족하는 객체의 대응 영역

(corresponding region)이 되도록 정의될 수 있다. 따라서, 도 3b는 O7(46)("피부 톤 & 형상")이 O4(48)("얼굴 영역1") 및 O6(50)("얼굴 영역2")을 포함하도록 도시한다. 상이한 특징들을 결합시키는 객체 계층들(26)은 또한 애플리케이션 시스템들의 광범위한 요구들을 충족하도록 구성될 수 있다.

또한 도 2에서 도시된 바와 같이, 각 객체 계층 요소(26)(`<object_hierarchy>`)는 객체 노드들(ONs)(32)의 트리를 포함한다. 이 객체 계층들은 또한 임의 스트링 속성 유형(optional string attribute types)을 포함할 수 있다. 만약 이러한 스트링 속성 유형들이 존재하며, 시소러스(thesaurus)는 애플리케이션들이 존재하는 계층들의 유형을 정할 수 있도록 이 스트링 속성 유형들의 값을 제공할 수 있다. 모든 객체 노드(32)(`<object_node>`)는 링크(33)를 경유하여 이미지 객체 집합(24)의 이미지 객체(30)를 참조한다. 이미지 객체들(30)은 또한 링크(33)를 경유하여 그들을 참조하는 객체 노드들(32)을 반대로 참조할 수 있다. 이 양방향성 링크 매커니즘은 이미지 객체 집합(24)의 이미지 객체들(30)에서 객체 계층(26)의 대응한 객체 노드들(32)로, 또 그 반대로의 효율적인 횡단(transversal)이 가능하도록 한다. 각각의 객체 노드(32)는 이미지 객체의 유일한 식별자를 사용함으로써 속성(object_ref)을 이용하여 이미 객체(30)를 참조한다. 각각의 객체 노드(32)는 또한 속성의 형태로 유일한 식별자를 포함할 수 있다. 이 객체 노드들(32)에 대한 유일한 식별자들은 객체들(30)이 다른 속성(object_node_ref)을 사용하여 그들을 참조하는 객체 노드들을 반대로 참조할 수 있도록 한다. 도 1b에 도시한 공간 객체 계층의 예를 구현하는 XML의 일례를 아래와 같이 표현된다.

```
<object_hierarchy type="SPATIAL"> <!-- Object hierarchy: spatial hierarchy -->
  <object_node id="ON0" object_ref="O0"> <!-- Photograph -->
    <object_node id="ON1" object_ref="O1"> </object_node> <!-- Person A -->
    <object_node id="ON2" object_ref="O2"> </object_node> <!-- Person B -->
  </object_node>
</object_hierarchy>
```

또한 객체 계층들(26)은 클러스터링 계층들을 만들고 복수 레벨의 계층화를 생성하는 데 사용될 수 있다. 예를 들어 위성 사진 이미지와 같은 비교적 큰 이미지를 기술함에 있어, 보통 이러한 이미지들에 포함된 그 많은 객체들을 효율적인 방식으로 그리고 스케일러블 방식으로 기술하고 검색할 때, 대개 문제가 발생한다. 클러스터링 계층들은 이 문제의 해결책(solution)을 제공하기 위해 본 발명의 이미지 디스크립션 시스템과 관련하여 사용될 수 있다.

도 4a 및 도 4b는 클러스터링 계층 스킴 사용의 일례를 도시하며, 여기서 객체들은 그 각각의 크기(`<size>`)에 기초하여 계층적으로 클러스터링된다. 특히, 도 4a는 예를 들어 위성 사진 이미지와 같은 비교적 큰 이미지 표현을 도시하며, 여기서 객체들 O11(52), O12(54), O13(56), O14(58) 및 O15(60)는 큰 이미지에 포함된, 예를 들어 지구 표면의 호수와 같은 변화하는 크기의 이미지 객체를 나타낸다. 도 4b는 도 4a에 도시한 객체들에 대한 크기 기반(size-based) 클러스터링 계층의 일례를 나타내며, 여기서 객체들 O11(52), O12(54), O13(56), O14(58) 및 O15(60)는 도 4a에 도시된 객체들을 나타내고, 객체들 O16(62), O17(64) 및 O18(66)은 도 4b에 도시된 클러스터 계층에 대한 크기 기반 기준들(criteria)을 명기하는 객체들을 나타낸다. 특히, 객체 O16(62), O17(64) 및 O18(66)은 예를 들어 객체 계층(26)의 중간 노드들(intermediate nodes)(32)을 의미할 수 있으며, 중간 노드들은 이미지 객체들(30)과 같이 표현된다. 이 객체들은 도시된 클러스터 계층에 같이 그룹 짓기 위하여 사용된 크기 특징과 관련된 기준들, 조건들(conditions) 및 제약들(constraints)을 포함한다. 도 4b에 도시된 특정한 예에서 객체 O16(62), O17(64) 및 O18(66)은 크기에 기초한 세 가지 계층적 레벨을 갖는 클러스터링 계층을 형성하는 데 사용된다. 객체 O16(62)은 클러스터링 계층을 형성하는 크기 기준들을 나타낸다. 객체 O17(64)은 50 단위(unit)보다 작은 크기 기준들의 두 번째 레벨을 나타내고, 여기서 이러한 단위는 픽셀로 나타낼 수 있으며, 객체 O18(66)은 10 단위보다 작은 크기 기준들의 세 번째 레벨을 나타낸다. 도 4b에 도시된 바와 같이, 객체 O11(52), O12(54), O13(56) O14(58) 및 O15(60)는 각각 소정 단위 수의 특정 크기를 갖는 것으로 특징이 부여된다. 유사하게, 객체 O13(56), O14(58) 및 O15(60)는 10 단위보다 작은 특정 크기를 갖는 것으로 특징이 부여된다.

비록 도 4a 및 도 4b가 기준들의 단일 집합, 즉 복수의 특징들에 관계하는 상이한 기준들을 사용하는 크기, 복수 클러스터링 계층들만에 기초하는 단일 클러스터링 계층의 일례를 도시하였지만, 모든 이미지에 대해 사용될 수 있다. 예를 들어, 이러한 클러스터링 계층들은 미디어 특징들, 시각적 특징들 및/또는 의미적 특징들의 임의 조합에 기초하여 객체들과 함께 그룹으로 만들 수 있다. 이 절차(procedure)는 이미지들을 시각적 정보 검색 엔진들에 같이 클러스터링하기 위해 사용된 절차와 유사하다. 전체 큰 이미지 내에 포함된 각 객체는 객체 집합(24)의 이미지 객체(30)로 할당되고, 또한 미디어 특징들(36), 시각적 특징들(38) 또는 의미적 특징들(40)과 같은 어떤 관련된 특징들에 할당될 수 있다. 객체 계층(26)의 중간 노드들(32)은 이미지 객체들(30)로 표현되고, 또한 객체들을 특정 레벨에서 함께 그룹으로 만들기 위해 사용되는 하나 이상의 특징들과 관련된 기준들, 조건들 및 제약들을 포함한다. 이미지 디스크립션은 임의 개수의 클러스터링 계층들을 포함할 수 있다. 도 4a 및 도 4b에 도시된 클러스터링 계층의 예는 아래의 XML로 구현된 일례로 표현될 수 있다.

```

<image>
  <image_object_set>
    <image_object type="LOCAL" id="O11"> <!-- Real objects of the image -->
      <size> <num_pixels> 120 </num_pixels> </size>
    </image_object> <!-- Others objects -->
    <image_object type="LOCAL" id="O17"> <!--Intermediate nodes in the
hierarchy-->
      <size> <num_pixels> <less_than> 50 </less_than> </num_pixels>
    </size>
    </image_object> <!-- Others objects -->
  </image_object_set>
  <object_hierarchy>

    <object_node id="ON11" object_ref="O16">
      <object_node id="ON12" object_ref="O11" />
      <object_node id="ON13" object_ref="O12" />
      <object_node id="ON14" object_ref="O17">
        <object_node id="ON15" object_ref="O13" />
        <object_node id="ON16" object_ref="O14" />
        <object_node id="ON17" object_ref="O18">
          <object_node id="ON18" object_ref="O15" />
        </object_node>
      </object_node>
    </object_node>
  </object_hierarchy>
</image>

```

도 4a 및 도 4b의 복수 클러스터링 계층의 예에 도시된 바와 같이, 또 아래의 표 3에 나타낸 바와 같이, 도시된 객체들의 크기에 기초하는 3 레벨의 추상화(abstraction)가 정의되어 있다. 이 복수 레벨 추상화 스킴은 도 4a에 도시된 이미지에서 객체들을 검색하고 뷰잉하기 위한 스케일러블 방법을 제공한다. 이러한 접근 방법은, 예를 들어 다양한 의미적 클래스들과 같은 다른 특징들에 기초하는 복수의 추상화 레벨을 표현하는 데 사용될 수 있다.

추상화 레벨	객체들
1	011, 012
2	011, 012, 013, 014
3	011, 012, 013, 014, 015

도 4a 및 도 4b에 도시된 클러스터링 계층들 및 복수 레벨의 추상화의 예에 대한 추가적인 XML 실시예는 첨부서류 A에 포함된다.

비록, 이러한 계층적 구조들이 이미지 검색의 목적에는 적합하지만, 객체들 사이의 일정한 관계들은 이러한 구조들을 사용하여 적절하게 표현될 수 없다. 따라서, 도 1c 및 도 2에 도시된 바와 같이, 본 발명의 이미지 디스크립션 시스템은 또한 객체들 사이의 보다 복잡한 관계들을 상세하게 설명하기 위해 엔티티 관계(E-R) 그래프(28)를 사용한다. 엔티티 관계 그래프(28)는 하나 이상의 엔티티 노드(42)와 그들 사이의 관계를 나타낸 그래프이다. 아래의 표 4는 각각의 특정한 예는 물론이고 그러한 관계 유형의 몇 가지 다른 예를 나타낸다.

관계 유형	관계들

공간적	방향적	위의, 아래의, 왼쪽의, 오른쪽의, 상부 왼쪽의, 상부 오른쪽의, 하부 왼쪽의, 하부 오른쪽의
	위상적 (topological)	인접하는(adjacent to), 이웃하는(neighboring to), 가까이에, 내에(within), 포함하다(contain)
의미적	관계 있는(relative of), 속하는, 일부, 관련이 있는(related to), 동일한, ..인(is A), 구성하는	

엔티티 관계 그래프들은 어떤 일반적인 구조일 수 있으며, 또한 다양한 계승 관계(inheritance relationships)를 이용함으로써 어떤 특정한 애플리케이션을 위해 커스텀화(customized)될 수 있다. 도 1c에 도시된 엔티티 관계 그래프의 예는 도 1a에 도시된 객체들 O1(2) 및 O2(6) 사이의 공간적인 관계의 예, 즉 "왼쪽의"와 의미 관계의 예, 즉 "약수하는"을 기술한다.

도 2에 도시된 바와 같이, 본 발명의 이미지 디스크립션 시스템은 0(zero) 또는 그 이상의 엔티티 관계 그래프들(28) (<entity_relation_graph>)을 상술(specification)할 수 있도록 한다. 엔티티 관계 그래프(28)는 하나 이상의 엔티티 관계 요소들(34)(<entity_relation>)의 집합을 포함하며, 또한 두 개의 선택 속성(optional attributes), 즉 엔티티 관계 그래프(28)에 의해 표현되는 결합관계(binding)를 기술하기 위한 유일한 식별자 ID와 스트링 유형을 포함한다. 이러한 유형들에 대한 값은 예를 들어 시소러스(thesaurus)에 의해 제공될 수 있다. 각 엔티티 관계 요소(34)는 하나의 관계 요소(44) (<relation>)를 포함하며, 또한 하나 이상의 엔티티 노드 요소들(42)(<entity_node>)과 하나 이상의 엔티티 관계 요소들 포함한다. 이 관계 요소(44)는 기술되는 특정한 관계를 포함한다. 각 엔티티 노드 요소(42)는 속성, 즉 object_ref를 사용하여 링크(43)를 경유하여 이미지 객체 집합(24)의 이미지 객체(30)를 참조한다. 링크(43)를 경유하여, 이미 객체들(30)은 또한 속성(entity_code_ref)을 사용하여 그 이미지 객체들(30)을 참조하는 엔티티 노드들(42)을 반대로 참조할 수 있다.

도 1c의 엔티티 관계 그래프(28)의 예에 도시된 바와 같이, 엔티티 관계 그래프(28)는 객체O1(2)("사람 A")와 객체O2(6) ("사람 B") 사이의 두 개의 엔티티 관계(34)를 포함한다. 첫 번째, 이러한 엔티티 관계(34)는 객체O1(2)이 객체O2(6)에 대하여 어떻게 위치되는지에 관한(즉 "왼쪽의") 공간적 관계(44)를 기술한다. 두 번째, 도 1c에 도시된 이러한 엔티티 관계(34)는 어떻게 객체O1(2)이 객체O2(6)와 "약수하는"지의 의미 관계(44)를 기술한다. 도 1c에 도시된 엔티티 관계 그래프의 예를 XML로 구현한 일례를 다음에 나타낸다:

```

<entity_relation_graph>
  <entity_relation> <!-- Spatial, directional entity relation -->
    <relation type="SPATIAL.DIRECTIONAL"> Left Of </relation>
    <entity_node id="ETN1" object_ref="O1"/> <entity_node id="ETN2" object_ref="O2"/>
  </entity_relation>
  <entity_relation> <!-- Semantic entity relation -->
    <relation type="SEMANTIC"> Shaking hands with </relation>
    <entity_node id="ETN3" object_ref="O2"/> <entity_node id="ETN4"
object_ref="O1"/>
  </entity_relation>
</entity_relation_graph>

```

효율을 높이기 위하여, 엔티티 관계 요소들(34)은 또한 도 2에 도시된 바와 같이 하나 이상의 다른 엔티티 관계 요소들(34)을 포함할 수 있다. 이것은 일련의 중첩 병렬 순차 관계(nested parallel sequential relationships)를 사용하여 상이한 미디어 문서들을 동기화(synchronizes)하는 SMIL(Synchronized Multimedia Intergarion Language)에서 사용되는 것과 같은 엔티티 관계들의 중첩(nested) 그래프들을 효율적으로 생성할 수 있도록 한다.

객체 계층(26)은 엔티티-관계 그래프(28)의 특별한 형태이므로, 엔티티-관계 그래프(28)를 이용하여 구현할 수 있으며, 여기서 엔티티들은 견제 관계들에 의해 관련된다. 견제 관계는 표 4에 나타난 것들과 같은 위상적인 관계들(topological relationships)이다. 엔티티-관계 그래프(28)의 특별한 형태인 객체 계층(26)을 도시하기 위하여, 도 1b에 도시된 객체 계층(26)의 예는 엔티티-관계 그래프(28)처럼 XML을 사용하여 아래와 같이 표현된다.

```

<entity_relation_graph>
  <entity_relation>
    <relation type="SPATIAL"> Contain </relation>
    <entity_node object_ref="O0"/> <entity_node object_ref="O1"/>
  </entity_relation>
  <entity_relation>
    <relation type="SPATIAL"> Contain </relation>
    <entity_node object_ref="O0"/> <entity_node object_ref="O2"/>
  </entity_relation>
</entity_relation_graph>

```

도 1b에 도시된 예는 어떻게 객체 O0(8)(사진 전체)이 객체 O1(2)("사람 A") 및 객체 O2(6)("사람 B")를 공간적으로 포함하는가를 기술한다. 따라서, 특정 요구조건들에 기초하여, 애플리케이션들은 엔티티 관계 그래프(28)의 내포적인 구조(comprehensive structure)의 편의성(convenience)을 활용하거나 또는 대안적으로 객체 계층들(26)의 효율(efficiency)을 활용함으로써 사용하는 계층들을 구현할 수 있다.

예를 들어, 미디어 특징들(36), 시각적 특징들(38) 또는 의미적 특징들(40)과 같은 특징들의 모든 유형에 관련된 이미지 디스크립터들의 경우, 본 발명의 이미지 디스크립션 시스템은 또한 아래의 XML의 예에 도시된 바와 같이, 코드 다운로드를 촉진하기 위하여 추출 및 유사성 매칭 코드와의 링크들을 포함할 수 있다. 이 링크들은 독점적인 디스크립터들(proprietary discriptors)을 사용하는 상이한 소스들(sources)로부터 이미지 콘텐츠의 효율적인 검색 및 필터링을 위한 메커니즘을 제공한다. 본 발명의 이미지 디스크립션 시스템의 이미지 디스크립터 각각은 특정한 디스크립터에 대한 추출 및 유사성 매칭 코드와 관련한 정보를 포함하는 디스크립터 값과 코드 요소(code element)를 포함할 수 있다. 이 코드 요소(<element>)는 또한 코드를 실행하기 위한 입력 파라미터들(<input_parameters>) 및 출력 파라미터들(<output_parameters>)의 디스크립션은 물론이고, 실행 가능한 파일들의 포인터들(<location>)을 포함할 수 있다. 코드의 유형에 관한 정보(즉, 추출 코드 또는 유사성 매칭 코드), 코드 언어(예를 들어 Java 또는 C와 같은) 및 코드 버전이 코드 요소의 특정한 속성으로 정의될 수 있다.

아래에 나타낸 XML로 구현한 예는 이른바 Tamura 텍스처 특징이라 하는 디스크립션을 제공하며, H.Tamura, S. Mori 그리고 T. Yamawaki가 발표한 "Textual Features Corresponding to Visual Percetion"(IEEE Transactions on Systems, Man and Cybernetics, Vol. 8, No. 6, June 1978)의 내용 전부는 본 명세서에 참조되어 본 명세서의 일부를 이룬다. Tamura 텍스처 특징은 특정한 특징 값, 즉, 거침(coarseness), 콘트라스트(contrast) 및 방향성(directionality)을 제공하고, 또한 특징 추출 및 유사성 매칭을 위한 외부 코드와의 링크들을 제공한다. 아래에 나타낸 특징 추출의 예에서는 또한 입력 및 출력 파라미터들에 관한 추가적인 정보가 제공된다. 이러한 디스크립션은 예를 들어 메타검색 엔진(metasearch engine)으로부터 텍스처 질의에 응답하는 검색 엔진에 의해 생성될 수 있다. 이때 이 메타검색 엔진은 사용자에 대한 결과들의 동질성 리스트를 생성하기 위하여 다른 검색 엔진들로부터 수신된 결과들에서 동일한 특징 디스크립터를 추출하기 위한 코드를 사용할 수 있다. 다른 경우에는, 특정 특징 값들이 아니라 추출 및 유사성 매칭 코드만이 포함할 수 있다. 필요한 경우 이러한 인스턴스들, 필터링 에이전트들은 프로세싱 동안에 특징 값들을 추출하기 위해 사용될 수 있다.

아래에 나타낸 XML로 구현한 예는 또한 디스크립터들에 대해 외부적으로 정의된 디스크립션 스킴들이 본 발명의 이미지 디스크립션 시스템에 내포될 수 있고 결합될 수 있도록 한다. 아래의 예에서, 크로마 키(Croma Key) 형상 특징에 대한 외부 디스크립터는 XML 명칭공간(namespaces)을 사용하여 이미지 디스크립션 내에 내포된다(imported). 이 프레임워크, 새로운 특징들, 특징들의 유형 및 이미지 디스크립터들의 사용은 확장 가능한 그리고 모듈러 방식으로 용이하게 포함될 수 있다.

```

<texture> <tamura>
  <tamura_value coarseness="0.01" contrast="0.39" directionality="0.7"/>
  <code type="EXTRACTION" language="JAVA" version="1.1"> <!-- Link extraction
code -->
    <location> <location_site href="ftp://extract.tamura.java"/> </location>
    <input_parameters> <parameter name="image" type="PPM"/>
</input_parameters>
    <output_parameters>
      <parameter name="tamura texture" type="double[3]"/>
    </output_parameters>
  </code>
  <code type="DISTANCE" language="JAVA" version="4.2"> <!-- Link similarity code --
>
    <location> <location_site href="ftp://distance.tamura.java"/> </location>
  </code>
</tamura> </texture>

<shape> <!-- Import external shape descriptor DTD -->
  <chromaKeyShape xmlns:extShape "http://www.other.ds/chromaKeyShape.dtd">
    <extShape:HueRange>
      <extShape:start> 40 </extShape:start> <extShape:end> 40
    </extShape:end>
  </extShape:HueRange>
</chromaKeyShape>
</shape>

```

본 발명의 이미지 디스크립션 시스템은 또한 양식 트랜스코딩(modality transcoding)을 지원한다. 여러 사용자들에게 이미지 콘텐츠를 전송하여야하는 콘텐츠 방송국의 인스턴스 예에서, 방송국은 사용자들의 다양한 단말기 요구조건 및 대역폭 조건을 수용하기 위하여 이미지 콘텐츠를 상이한 미디어 양식 및 해상도로 트랜스코딩 하여야 한다. 본 발명의 이미지 디스크립션 시스템은 지역 및 전역 객체들 모두와 관련된 양식 트랜스코딩을 제공한다. 이 양식 트랜스코딩은 미디어 양식(media modality), 해상도 및 문제의 이미지 객체들의 트랜스코딩된 버전(versions)의 위치를 트랜스코딩 하거나, 또는 선택적으로 외부 트랜스코딩 코드에 링크한다. 문제의 이미지 디스크립터는 또한 상이한 사용자 단말기들의 요구조건을 충족시키기 위하여 이미지 객체를 다른 양식 및 해상도로 트랜스코딩하기 위한 코드를 지시할 수 있다. 아래에 나타난 XML로 구현한 예는 이미지 객체에 대해 트랜스코딩된 오디오를 제공하는 것을 도시한다.

```

<image_object type="GLOBAL" id="O0">
  <img_obj_media_features>
    <location> <location_site href="Hi.gif"/> </location>
    <modality_transcoding>
      <modality_object_set>
        <modality_object id="mo2" type="AUDIO" resolution="1">
          <location><location_site
href="Hi.au.xml"?o1/></location>
        </modality_object>
      </modality_object_set>
    </modality_transcoding>
  </img_obj_media_features>
</image_object>

```

본 발명의 이미지 디스크립션 시스템의 문서 유형 정의(Document Type Definition, DTD) 추가적인 예는 첨부서류 A에 제공된다.

도 5는 본 발명의 이미지 디스크립션 시스템을 구현하는 컴퓨터 시스템의 예를 블록도로 나타낸 것이다. 도시된 컴퓨터 시스템은, 예를 들어 이미지 입력 인터페이스(404)를 경유하여 이미지 콘텐츠를 표현하는 디지털 데이터를 수신하는 컴퓨터 프로세서부(402)를 포함한다. 이와는 달리, 디지털 이미지 데이터는 원격 소스(remote source)로부터 양방향 통신 입/출력(I/O) 포트(406)를 경유하여 프로세서부(402)에 전달될 수 있다. 이 이미지 콘텐츠는 또한 임의의 광 데이터 저장 시스템이나 자기 데이터 저장시스템과 같은 이 기술분야에서 공지된 비휘발성 컴퓨터 미디어(408)로부터 프로세서부(402)에 전달될 수 있다. 프로세서부(402)는 예를 들어 종래의 개인용 컴퓨터 시스템 및 워크스테이션에서 일반적으로 채용되는 표준 SVGA 모니터 및 비디오 카드와 같은 대개 적합한 인터페이스 회로와 고해상도 모니터를 포함하는 이미지 디스플레이 시스템(410)에 데이터를 제공한다. 키보드 및 예를 들어 마우스, 트랙볼, 광 펜이나 터치 스크린과 같은 디지털 포인팅 장치(digital pointing device)와 같은 사용자 입력 장치는 사용자와 컴퓨터의 상호작용을 위하여 프로세서부(402)에 연결된다. 도 5의 컴퓨터 시스템의 예는 또한 프로세싱 작업을 하는 동안에 프로세서부(402)가 접근할 수 있는 휘발성 및 비휘발성 컴퓨터 메모리(414)를 포함할 것이다.

도 6은 본 발명의 이미지 디스크립션 시스템을 구현하기 위하여 도 5에 도시된 컴퓨터 시스템에 의해 시작되는 프로세싱 작업들을 추가로 도시한 흐름도이다. 디지털 이미지 데이터(310)는 링크(311)를 경유하여 컴퓨터 시스템에 입력된다. 적당한 애플리케이션 소프트웨어의 제어를 받는 컴퓨터 시스템은 블록(320)에서 이미지 객체 추출을 수행하며, 이미지 객체들(30) 및 예를 들어 미디어 특징들(36), 시각적 특징들(38) 및 의미적 특징들(40)과 같은 관련 특징들이 생성된다. 이미지 객체 추출(320)은 전자동 프로세싱 작업, 반자동 프로세싱 작업 또는 예를 들어 사용자 입력 장치(412)를 경유하는 것과 같이 사용자 상호작용에 의해 주로 정의되는 객체들의 경우에는 거의 수동 작업의 형태를 취할 수 있다.

바람직한 실시예에서, 이미지 객체 추출(320)은 두 개의 보조적인 작업, 즉 블록(325)에 도시된 이미지 세그멘테이션 작업과 블록(326)에 도시된 특징 추출 및 주석 작업으로 구성된다. 이미지 세그멘테이션 단계(325)의 경우, 디지털 이미지들을 하나 이상의 공통된 특징을 공유하는 영역들로 분할하는 임의의 영역 추적 기술(region tracking technique)이 사용될 수 있다. 또한, 특징 추출 및 주석 단계(326)의 경우, 분할된 영역들로부터 특징을 생성한 모든 기술이 사용될 수 있다. 영역 기반(region-based) 클러스터링 및 검색 서브시스템이 자동화된 이미지 세그멘테이션과 특징 추출에 적합하다. 이미지 객체 세그멘테이션 시스템은 반자동화된 이미지 세그멘테이션 및 특징 추출 시스템의 일례이다. 이와는 달리 수동 세그멘테이션 및 특징 추출이 사용될 수 있다. 이 시스템의 예에서, 이미지 세그멘테이션(325)은 예를 들어 이미지 객체들(30)을 생성할 수 있고, 특징 추출 및 주석 달기(326)는 예를 들어 이미지 객체들(30)과 관련된 특징들, 예를 들어 미디어 특징들(36), 시각적 특징들(38) 및 의미적 특징들(40)과 같은 특징들을 생성할 수 있다.

객체 추출 프로세싱(320)은 하나 이상의 이미지 객체들(30)을 포함하는 이미지 객체 집합(24)을 생성한다. 이때, 이미지 객체 집합(24)의 이미지 객체들(30)은 추가 프로세싱을 위하여 블록(330)에 도시된 바와 같은 객체 계층 구축 및 추출 프로세싱 및/또는 블록(336)에 도시된 바와 같은 엔티티 관계 그래프 생성 프로세싱의 형태로 링크(321, 322, 324)를 경유하여 제공될 수 있다.

바람직하게는, 객체 계층 구축 및 추출(330)과 엔티티 관계 그래프 생성(336)은 링크(327)를 통해 병렬로 발생된다. 이와 달리, 이미지 객체 집합(24)의 이미지 객체들(30)은 링크(323)를 경유하여 객체 계층 구축 및 추출(330)과 엔티티 관계 그래프 생성(336)을 우회하도록 지시 받을 수 있다. 따라서 객체 계층 구축 및 추출(330)은 하나 이상의 객체 계층들(26)을 생성하고, 그러므로 엔티티 관계 그래프 생성(336)은 하나 이상의 엔티티 관계 그래프들(28)을 생성한다.

프로세서부(402)는 이미지 객체 집합(24), 객체 계층들(26) 및 엔티티 관계 그래프들(28)을 문제의 이미지 콘텐츠에 대한 이미지 디스크립션 레코드에 병합한다. 그런 다음, 이 이미지 디스크립션 레코드는 데이터베이스 기억장치(340)에 직접적으로 저장되거나, 선택적으로 링크들(343, 361)을 경유하여 이진 부호기(360)에 의해 먼저 압축된 다음 저장되거나, 또는 링크들(341, 351)을 경유하여 XML 부호기(350)에 의해 디스크립션 정의 언어(예를 들어 XML)로 부호화된 다음 저장된다. 일단 이미지 디스크립션 레코드들이 데이터베이스 기억장치(340)에 저장되면, 레코드들은 예를 들어 검색, 필터링 및 파일보관(archiving) 애플리케이션들과 같은 다른 애플리케이션들(370)에 의한 접근 및 사용에 유용한 포맷으로 사용할 수 있도록 유지된다.

도 7을 참조하면, 본 발명의 이미지 디스크립션 시스템 상에서 구현될 수 있는 클라이언트 서버 컴퓨터 시스템의 일 실시 예를 제공한다. 시스템(100)의 아키텍처(architecture)는 클라이언트 컴퓨터(100)와 서버 컴퓨터(120)를 포함한다. 서버 컴퓨터(120)는 디스플레이 인터페이스(130)와 질의 디스패처(query dispatcher)(140), 실행 데이터베이스(150), 질의 번역기들(160, 161, 165), 목표 검색 엔진들(170, 171, 175) 및 멀티미디어 콘텐츠 디스크립션 시스템(200, 201, 205)을 포함하며, 이들에 대해서는 이하에서 더욱 자세하게 설명할 것이다.

한편, 이 예시적인 클라이언트 서버 실시예에 언급할 다음의 설명(disclosure)은 이 기술분야의 당업자라면 특정한 시스템 배치는 수많은 공지된 지역 또는 분산 아키텍처를 포함하도록 본 발명의 범위 내에서 변경될 수 있음을 이해하여 한다.

상업적으로 사용 가능한 메타검색 엔진은 사용자들을 자동으로 그리고 투명하게(tansparently) 복수의 텍스트 기반 검색 엔진들과 링킹하는 게이트웨이(gateway)로 작용한다. 도 7의 시스템은 이러한 메타검색 엔진들의 아키텍처가 성장하고, 상이한 클래스들의 사용자 질의들에 대한 그들의 성능을 등급 매김으로써 다수의 온라인 멀티미디어 검색 엔진들을 능동적으로 선택하고 인터페이스하도록 설계된다. 따라서, 상업적으로 이용 가능한 메타검색 엔진들의 질의 디스패처(140), 질의 번역기(160, 161, 165) 및 디스플레이 인터페이스(130)는 본 발명에 사용될 수 있다.

디스패처(140)는 사용자의 질의를 수신함에 따라 실행 데이터베이스(150)를 조사함으로써 질의를 받게 될 목표 검색 엔진들을 선택한다. 이 데이터베이스(150)는 각각의 지원되는 검색 옵션에 대한 과거의 질의 성공 및 실패에 대한 실행 스코어를 포함한다. 질의 디스패처는 사용자의 질의, 예를 들어 검색 엔진들을 활성화(enable) 시키는 트리거 색채(trigger color)가 될 색채 정보를 찾는 질의를 충족시킬 수 있는 검색 엔진들(170, 171, 175)만을 선택한다. 검색 엔진들(170, 171, 175)은 예를 들어 검색 엔진(170)과 연관된 클라이언트(172)와 같은 클라이언트-서버 관계로 배치될 수 있다.

질의 번역기들(160, 161, 165)은 선택된 검색 엔진들의 인터페이스에 적합하도록 사용자의 질의를 적당한 스크립트들로 번역한다. 디스플레이 구성요소(130)는 각 검색 엔진의 결과를 병합하기 위해 실행 스코어를 사용하며, 그것을 사용자에게 보여준다.

본 발명에 따르면, 사용자가 인터넷이나 지역적인(regional) 또는 국부적인(local) 네트워크의 시각적 콘텐츠를 능동적으로 검색할 수 있도록 하기 위해, 검색 질의들은 본 발명에 의해 생성되는 멀티미디어 콘텐츠 디스크립션에 의해 만들어지거나 보기(example) 또는 스케치(sketch)에 의해 만들어진다. 검색 엔진(170, 171, 175) 각각은 검색 엔진이 접근할 수 있는 멀티미디어 정보 콘텐츠를 기술하고 검색을 실행하기 위하여 예를 들어 아래에 기술된 디스크립션 스킴들과 같은 디스크립션 스킴을 사용한다.

멀티미디어 정보에 대한 콘텐츠 기반 검색 질의를 실행하기 위하여, 디스패처(140)는 사용자가 질의에서 선택한 것을 만족시키는 것을 확보하기 위해 각 검색 엔진(170, 171, 175)에 의해 사용되는 멀티미디어 콘텐츠 디스크립션 시스템(200)을 통해 질의 디스크립션을 조화시킬 것이다. 그런 다음 디스패처(140)는 실행 데이터베이스(150)를 조사함으로써 질의 받게 될 목표 검색 엔진들(170, 171, 175)을 선택할 것이다. 예를 들어, 사용자는 색채로 검색하기를 원하고 어떤 검색 엔진이 모든 색채 디스크립터들을 지원하지 않는 경우, 특정한 검색 엔진은 질의에 소용되지 못할 것이다.

다음에, 질의 번역기들(160, 161, 165)은 질의 디스크립션을 각각의 선택된 검색 엔진에 적합한 디스크립션으로 개조할 것이다. 이 번역은 또한 각각의 검색 엔진에서 사용 가능한 디스크립션 스킴들에 기초될 것이다. 이 태스크(task)는 표준 디스크립터들에 대한 실행 추출 코드나 특정 검색 엔진들로부터 변환 디스크립터들로 다운로드된 추출 코드를 요구할 수 있다. 예를 들어, 사용자가 166 빈즈(bins)의 색채 코히어런스(color coherence)를 사용하는 객체의 색채 특징을 명기하면, 질의 번역기는 그것을 각 검색 엔진이 사용하는 특정 색채 디스크립터들(예를 들어 색채 코히어런스와 x 빈즈의 색채 히스토그램)로 번역할 것이다.

그 결과를 사용자에게 보여주기 전에, 질의 인터페이스는 비교하고 등급을 매기기 위해 모든 결과 디스크립션들을 동종의 것으로 번역하여 각 검색 옵션으로부터의 결과를 병합할 것이다. 다시, 표준 디스크립터들에 대한 유사성 코드 또는 검색 엔진들로부터 다운로드된 유사성 코드는 실행될 필요가 있다. 사용자 선택들은 어떻게 그 결과를 사용자에게 표시할 것인지 결정할 것이다.

도 8을 참조하여 각 검색 엔진(170, 171, 175)에 의해 사용되는 본 발명에 따른 디스크립션 시스템(200)을 설명한다. 본 명세서에 개시된 바람직한 실시예에서, 멀티미디어 콘텐츠를 기술하기 위해 XML을 사용한다.

디스크립션 시스템(200)은 멀티미디어 아이템들(205)의 수집을 위한 풍부하고 다양한 디스크립터들을 생성하기 위해 몇 개의 멀티미디어 프로세싱, 분석 및 주석 서브시스템(210 ~ 280)을 포함한다. 각 서브시스템을 차례로 설명한다.

첫 번째 서브시스템(210)은 자동으로 분할된 비디오 시퀀스의 영역들에 대해 색채, 텍스처, 움직임, 형상 및 크기와 같은 시각적 특징들을 추출하는 영역 기반 클러스터링 및 검색 시스템이다. 시스템(210)은 예를 들어, 디졸브(dissolve), 페이드 인/아웃(fade in/out), 와이프(wipe) 등의 급전하거나 과도적일 수 있는 장면 변화 검출에 의해 비디오를 개별 샷(shots)으로 분해한다. 각각의 샷에 대해, 시스템(210)은 전역 움직임(즉, 두드러진 배경의 움직임)과 카메라 움직임을 추

정하고, 그런 다음 각 영역에 대한 상이한 시각적 특징을 산출하는 쏫 내의 전체 프레임들의 영역들을 분할, 검출 및 추적한다. 각각의 쏫에 대해, 이 시스템에 의해 생성된 디스크립션은 시각적 특징들과 움직임 특징들 그리고 카메라 움직임을 갖는 영역들의 집합이다. 영역 기반 클러스터링 및 검색 시스템(210)의 완전한 기술(記述)은 본 발명과 동일한 출원인이 출원한(co-pending) PCT 출원번호 PCT/US98/09124(출원일: 1998년 5월 5일, 명칭: "An Algorithm and System Architecture for Object-Oriented Content-Based Video Search")에 포함되어 있으며, 그 내용은 본 명세서에 참조되어 본 명세서의 일부를 이룬다.

본 명세서에 사용된 바와 같이, "비디오 클립(video clip)"은 식별 가능한 속성들을 갖는 하나 이상의 비디오 객체들(예를 들어 야구 방망이를 스윙하는 야구선수, 바다를 가로질러 움직이는 파도타기 보드, 또는 초원을 가로질러 달리는 말과 같은 것으로서 이에 한정되지는 않음)을 갖는 비디오 정보의 일련의 프레임들을 참조할 것이다. "비디오 객체"는 하나 이상의 관심 있는 특징들, 예를 들어 텍스트, 색채, 움직임 또는 형상과 같은 특징들에 있어 동질인 픽셀들의 인접한(contiguous) 집합이다. 그러므로, 비디오 객체는 적어도 하나의 특징에 있어 일관성을 나타내는 하나 이상의 비디오 영역들로 형성된다. 예를 들어, 걷는 사람(여기서, 사람이 "객체"이다)의 쏫은 형상, 색채 및 텍스트와 같은 기준들이 상이한 인접한 영역들의 집합(collection)으로 분할될 것이지만, 모든 영역들은 그들의 움직임 속성에 있어 일관성을 나타낼 수 있다.

두 번째 서브시스템(220)은 효율적으로 그리고 자동으로 MPEG 압축된 영역에서 직접적으로 얼굴을 검출하는 MPEG 도메인 얼굴 검출 시스템(MPEG-domain face detection system)이다. 인간의 얼굴은 이미지와 비디오에서 중요한 주제이다. 비디오 콘텐츠의 이해를 위해 시청자에게 중요 정보를 제공하는 인간의 얼굴은 뉴스, 다큐멘터리, 영화 등 어디에나 있다. 이 시스템은 얼굴 라벨들(labels)을 갖는 영역들의 집합을 제공한다. 이 시스템(220)의 완전한 기술(記述)은 PCT 출원번호 PCT/US97/20024(출원일: 1997년 11월 4일, 명칭: "A Highly Efficient System for Automatic Face Region Detection in MPEG Video")에 포함되어 있으며, 그 내용은 본 명세서에 참조되어 본 명세서의 일부를 이룬다.

세 번째 서브시스템(230)은 비디오 시퀀스들에서 의미적 객체들을 추적하기 위해 자동적인 세그멘테이션이 사용자 입력과 함께 통합되는 비디오 객체 세그멘테이션 시스템이다. 일반적인 비디오 소스들의 경우, 이 시스템은 사용자가 트레이싱 인터페이스(tracing interface)를 사용함으로써 근사한 객체 경계(approximate object boundary)를 정의할 수 있도록 한다. 근사한 객체 경계가 주어지면, 이 시스템은 자동으로 이 경계를 다듬고(refine) 비디오의 다음 프레임들에서 객체의 움직임을 추적한다. 이 시스템은 복잡한 객체들, 빠르고 간헐적인 움직임, 복잡한 배경, 다수의 움직이는 객체들 및 부분적인 폐색(occlusion)을 포함하며, 현존하는 접근법(approaches)을 사용하여 모델링하기 어려운 많은 현실 세계(real world) 상황을 취급하기에 충분히 강건하다(robust). 이 시스템에 의해 생성된 디스크립션은 텍스트로 수동 주석이 달린, 관련된 영역들과 특징들을 갖는 의미적 객체들의 집합이다. 이 시스템(230)의 완전한 기술(記述)은 미국 특허 출원번호 09/405,555 (출원일: 1998년 9월 24일, 명칭: "An Active System and Algorithm for Semantic Video Object Segmentation")에 포함되어 있으며, 그 내용은 본 명세서에 참조되어 본 명세서의 일부를 이룬다.

네 번째 서브시스템(240)은 쏫 경계들, 움직이는 객체들, 객체 특징들 및 카메라 움직임 등을 추출하기 위해 압축된 MPEG 비디오 스트림들(streams)을 분석하는 계층적 비디오 브라우징 시스템이다. 이 시스템은 또한 비디오의 직관적인 시각화 및 편집을 위해 계층적 쏫 기반(shot_based) 브라우징 인터페이스를 생성한다. 이 시스템(240)의 완전한 기술(記述)은 PCT 출원번호 PCT/US97/08226(출원일: 1997년 5월 16일, 명칭: "A Method and Architecture for Indexing and Editing Compressed Video Over the World Wide Web")에 포함되어 있으며, 그 내용은 본 명세서에 참조되어 본 명세서의 일부를 이룬다.

다섯 번째 서브시스템(250)은 수동 텍스트 주석들의 표제어(entry)이다. 이것은 종종 장면 분류를 위해 시각적 특징들 및 텍스트 특징들과 통합하는 것이 바람직하다. 예를 들어 Clarinet과 같은 온라인 뉴스 소스들의 이미지인 경우, 각 이미지들과 연관된 캡션(captions) 또는 기사(articles)의 형태로 텍스트 정보가 있다. 이 텍스트 정보는 디스크립션에 포함될 수 있다.

여섯 번째 서브시스템(260)은 저레벨 시각적 특징들에 기초하여 이미지들 및 비디오 쏫들의 고레벨 의미적 분류를 위한 시스템이다. 이 시스템의 핵심은 규칙 유도(rule induction), 최근접 이웃 분류(nearest neighbor classification)와 같은 다양한 기계 학습 기술(machine learning techniques)로 구성된다. 이 시스템은 {자연 풍경(nature landscape)}, {도시/교외(city/suburb)}, {실내(indoor)} 및 {야외(outdoor)}와 같은 이미지들 및 비디오 장면들을 고레벨의 의미적인 장면 클래스들로 분류하는 데 사용된다. 이 시스템은 동일한 의미적 장면 클래스들의 집합에 대해서조차도 어떤 코퍼스(corpus)와는 잘 작용할 수 있지만, 다른 코퍼스와는 잘 작용하지 않는 고정된 규칙들의 집합을 발견하였기 때문에 기계 학습 기술들에 집중한다. 때문에 이 시스템은 핵심은 기계 학습 기술들에 기초하며, 이 시스템은 각 코퍼스의 예들을 사용하여 시스템을 훈련시킴으로써 상이한 코포라(corpora)에 대한 고성능을 실현하도록 적용될 수 있다. 이 시스템이 생성한 디스크립션은 각각의 이미지 또는 비디오 시퀀스의 쏫들과 관련된 각각의 키프레임(keyframe)에 대한 장면 클래스를 나타낸다.

이 시스템(260)의 완전한 기술(記述)은 S. Paek 등의 "Integration of Visual and Text based Approaches for the Content Labeling and Classification of Photographs"(ACM SIGIR'99 Workshop on Multimedia Indexing and Retrieval, Barkeley, CA(1999))에 포함되어 있으며, 그 내용은 본 명세서에 참조되어 본 명세서의 일부를 이룬다.

일곱 번째 서브시스템(270)은 모델 기반 이미지 분류 시스템(model-based image classification system)이다. 많은 자동 이미지 분류 시스템은 클래스-명세 알고리즘(class-specific algorithms)이 분류를 수행하는 데 사용되는 미리 정의된 클래스들의 집합에 기초한다. 이 시스템(270)은 사용자들이 자신들의 클래스들을 정의할 수 있도록 하며, 자동으로 시각적 모델들을 기억하는(learn) 데 사용되는 예제들을 제공한다. 이 시각적 모델들은 자동으로 분할되는 영역들, 자신들이 관련된 시각적 특징들 및 자신들의 공간적 관계들에 기초한다. 예를 들어, 사용자는 첫 번째 사람이 푸른 정장을 입고 갈색 소파에 앉아 있고, 두 번째 사람이 그 앉은 사람의 오른쪽에 서있는 초상화의 시각적 모델을 생성할 수 있다. 이 시스템은 분류하는 동안에 느린 학습법(lazy-learning), 결정 트리(decision trees) 및 전개(evolution program)의 조합을 사용한다. 이 시스템이 생성한 디스크립션은 텍스트 주석들의 집합, 즉 각 이미지에 대한 사용자 정의 클래스이다.

이 시스템(270)의 완전한 기술(記述)은 A. James 등의 "Model Based Classification of Visual Information for Content-Based Retrieval"(Symp. Elec. Imaging: Multimedia Proc. and App.- Storage Retrieval and for Image and Video Database VII, IS&T/SPIE'99(1999))에 포함되어 있으며, 그 내용은 본 명세서에 참조되어 본 명세서의 일부를 이룬다.

예를 들어 디스크립션들 또는 디스크립션들의 일부를 생성하는 데 사용되는 협력자들(collaborators)의 서브시스템들과 같은 다른 서브시스템들(280)이 멀티미디어 콘텐츠 디스크립션 시스템(200)에 추가될 수 있다.

작업(operation)에서, 이미지 및 비디오 콘텐츠(205)는 정지된 이미지들이나 움직이는 비디오의 데이터베이스, 브라우저 인터페이스(206)로부터 콘텐츠를 수신하는 버퍼(buffer), 또는 라이브 이미지(live image)나 녹화 방송(video transmission)의 저장소(receptacle)일 수 있다. 이 서브시스템들(210 내지 280)은 위에서 설명한 바와 같이, 자동으로 분할된 영역들의 저레벨 시각적 특징들, 사용자 정의 의미적 객체들, 고레벨 장면 특성(properties), 분류 및 관련된 테스트 정보를 포함하는 디스크립션(211 ~ 281)을 생성하기 위해 이미지 비디오 콘텐츠(205)에 작용한다. 일단 이미지나 비디오 아이템에 대한 모든 디스크립션들이 생성되고 블록(29)에서 통합되면, 이 디스크립션들은 다음에 검색 엔진(170)이 접근하는 데이터베이스(295)에 입력된다.

특정한 서브시스템들, 즉 영역 기반 클러스터링 검색 서브시스템(210)과 비디오 객체 세그먼테이션 시스템(230)은 전체 디스크립션 생성 프로세스를 실행할 수 있으며, 한편 나머지 서브시스템들은 프로세싱 중에 서브시스템(210, 230)에 의해 호출(call)될 수 있는 프로세스의 일부만을 실행한다. 유사한 방식으로, 서브시스템들(210, 230)은 프로세스 내의 특정 태스크들에 대해 서로 호출될 수 있다.

비록 본 발명을 특정 실시예와 관련하여 설명하였지만, 개시된 실시예들에 대한 다양한 변경, 치환 및 개조가 첨부한 청구 범위에 나타낸 바와 같은 본 발명의 사상 및 범위를 벗어나지 않고 가능함을 알 것이다.

첨부서류 A: 이미지 디스크립션 스킴들의 문서 유형 정의

image_ds.dtd:

```
<!-- Image DS -->

<!-- Entities are like macros. They can be referenced by using the notation
"%EntityName;" -->

<!-- For clarity, we have chosen not to reference them in this DTD, but they are
referenced in other DTDs derived from this one. -->

<!ENTITY % image_object_elements "
    img_obj_media_features?,
    img_obj_semantic_features?,
    img_obj_visual_features?">

<!ENTITY % ref_image_object_attributes "
    id ID #IMPLIED
    object_ref IDREF #IMPLIED
    object_node_ref IDREFS #IMPLIED
    entity_node_ref IDREFS #IMPLIED">

<!ENTITY % img_obj_media_features_elements "
    location?, file_format?, file_size?, resolution?, modality_transcoding?">

<!ENTITY % img_obj_semantic_features_elements "
    text_annotation?, ?, who?, what_object?, what_action?, where?, why?,
when?">

<!ENTITY % img_obj_visual_features_elements "
```

```

image_scl?, color?, texture?, shape?, size?, position?, motion?">

<!-- Image element -->
<!ELEMENT image (image_object_set, object_hierarchy*, entity_relation_graph*)>
<!ATTLIST image
    id ID #IMPLIED>

<!-- Image object set element -->
<!ELEMENT image_object_set (image_object+)>

<!ELEMENT image_object (img_obj_media_features?, img_obj_semantic_features?,
    img_obj_visual_features?">
<!ATTLIST image_object
    type (LOCAL|GLOBAL) #REQUIRED
    id ID #IMPLIED
    object_ref IDREF #IMPLIED
    object_node_ref IDREFS #IMPLIED
    entity_node_ref IDREFS #IMPLIED>

<!ELEMENT img_obj_media_features (location?, file_format?, file_size?,
resolution?,
    modality_transcoding?)>

<!ELEMENT img_obj_semantic_features (text_annotation?, who?, what_object?,
what_action?, where?, why?, when?)>

```

```

<!ELEMENT img_obj_visual_features (image_scl?, color?, texture?, shape?, size?,
position?,
                                motion?)>

<!-- Object hierarchy element -->
<!-- The attribute type is the hierarchy binding type -->
<!ELEMENT object_hierarchy (object_node)>
<!ATTLIST object_hierarchy
    id ID #IMPLIED
    type CDATA #IMPLIED>

<!ELEMENT object_node (object_node*)>
<!ATTLIST object_node
    id ID #IMPLIED
    object_ref IDREF #REQUIRED>

<!-- Entity relation graph element-->
<!-- Possible types of entity relations and entity relation graphs:
    - Spatial: topological, directional
    - Temporal: topological, directional
    - Semantic -->
<!ELEMENT entity_relation_graph (entity_relation+)>
<!ATTLIST entity_relation_graph
    id ID #IMPLIED
    type CDATA #IMPLIED>

<!ELEMENT entity_relation (relation, (entity_node | entity_node_set |
entity_relation)*)>

```

```
<!ATTLIST entity_relation
    type CDATA #IMPLIED>

<!ELEMENT relation (#PCDATA | code)*>

<!ELEMENT entity_node (#PCDATA)>
<!ATTLIST entity_node
    id ID #IMPLIED
    object_ref IDREF #REQUIRED>

<!ELEMENT entity_node_set (entity_node+)>

<!-- External object location DTD -->
<!ENTITY % location SYSTEM "location.dtd">
%location;

<!-- External file feature DTD -->
<!ENTITY % file SYSTEM "file.dtd">
%file;

<!-- External modality transcoding DTD -->
<!ENTITY % modality_transcoding SYSTEM "modality_transcoding.dtd">
%modality_transcoding;

<!-- External object annotation DTD -->
<!ENTITY % text_annotation SYSTEM "text_annotation.dtd">
%text_annotation;
```

```

<!-- External scalable image DTD -->
<!ENTITY % image_scl SYSTEM "image_scl.dtd">
%image_scl;

<!-- External object color DTD -->
<!ENTITY % color SYSTEM "color.dtd">
%color;

<!-- External object texture DTD -->
<!ENTITY % texture SYSTEM "texture.dtd">
%texture;

<!-- External object shape DTD -->
<!ENTITY % shape SYSTEM "shape.dtd">
%shape;

<!-- External object size DTD -->
<!ENTITY % size SYSTEM "size.dtd">
%size;

<!-- External object position DTD -->
<!ENTITY % position SYSTEM "position.dtd">
%position;

<!-- External object motion DTD -->
<!ENTITY % motion SYSTEM "motion.dtd">
%motion;

```

```

<!-- External object time DTD -->
<!ENTITY % time SYSTEM "time.dtd">
%time;

<!ENTITY mpeg7 "ISO/IEC JTC1/SC29/WG11 MPEG-7">

<!-- Image DS End -->

```

location.dtd:

```

<!-- Description of resources' location-->

<!-- Objects, image, videos can be located/accessed at different locations -->
<!ELEMENT location (location_site*)>

<!-- One location site -->
<!ELEMENT location_site EMPTY>
<!ATTLIST location_site
    href CDATA #REQUIRED
    title CDATA #IMPLIED>

<!ELEMENT code (location*)>
<!ATTLIST code
    type (EXTRACTION|DISTANCE) "EXTRACTION"
    language (C|JAVA|PERL) #REQUIRED
    version CDATA #REQUIRED>

<!-- Description of resources' storage location -->

```

file.dtd:

```

<!-- Description of resources' file specifications-->

<!-- File format -->
<!ELEMENT file_format (#PCDATA)>

<!-- File size -->
<!ELEMENT file_size (#PCDATA)>

<!ELEMENT resolution (res_width, res_height)>
<!ELEMENT res_width (#PCDATA)>
<!ELEMENT res_height (#PCDATA)>

<!-- Description of resources' file specifications -->

```

modality_transcoding.dtd:

```

<!-- Description of modality transcoding agent -->

<!-- Representation of image objects in different modalities (audio, text) and
resolutions -->
<!-- We reuse the defined structure entity_relation_graph -->
<!ELEMENT modality_transcoding (modality_object_set, entity_relation_graph?)>

<!ELEMENT modality_object_set (modality_object*) >

```

```

<!ELEMENT modality_object (location) >
<!ATTLIST modality_object
    id ID #REQUIRED
    type (AUDIO|VIDEO|TEXT|IMAGE|SYNTHETIC_VIDEO) #REQUIRED
    resolution CDATA #IMPLIED
    object_ref IDREF #IMPLIED>

<!-- Description of semantic transcoding agent -->
    
```

text_annotation.dtd:

```

<!-- Text annotation features -->

<!ELEMENT text_annotation (name | who | where | why | when | what_object |
what_action)*>

<!-- Name annotation -->
<!ELEMENT name (concept*, code*)>

<!-- People annotation -->
<!ELEMENT who (concept*, code*)>

<!-- Location annotation -->
<!ELEMENT where (concept*, code*)>

<!-- Event annotation -->
<!ELEMENT why (concept*, code*)>
    
```

```

<!-- Date annotation -->
<!ELEMENT when (concept*, code*)>

<!-- Object annotation -->
<!ELEMENT what_object (concept*, code*)>

<!-- Action annotation -->
<!ELEMENT what_action (entity_relation*, concept*, code*)>

<!-- Concept -->
<!ELEMENT concept (#PCDATA | code)*>
<!ATTLIST concept
    language CDATA "english"
    generation (automatic|manual) "manual">

<!-- Text annotation features end -->

```

image_scl.dtd

```

<!-- Image scalability features -->

<!ELEMENT image_scl (image_sclobj, code*)
<!ELEMENT image_sclobj (img_obj_scltype, img_obj_mode, img_obj_numlayers,
codref, subsamp_factor, img_obj_shape?)>

<!ELEMENT img_obj_scltype EMPTY>
<!ATTLIST img_obj_scltype
    typeinfo (DATPARTITION|SPATIAL|SNR) #REQUIRED>

```

```

<!-- Image scalability (subtype) mode features -->
<!ELEMENT img_obj_mode EMPTY>
<!ATTLIST img_obj_mode
    modeinfo CDATA #REQUIRED>

<!ELEMENT numlayers EMPTY>
<!ATTLIST img_obj_numlayers
    numval #REQUIRED>

<!ELEMENT codref EMPTY>
<!ATTLIST codref
    layernum #REQUIRED>

<-- subsampling ratio n/m for horizontal and vertical directions -->
<!ELEMENT subsamp_factor EMPTY>
<!ATTLIST subsamp_factor
    hor_factor_n CDATA #REQUIRED
    hor_factor_m CDATA #REQUIRED
    vert_factor_n CDATA #REQUIRED
    vert_factor_m CDATA #REQUIRED>

<!ELEMENT img_obj_shape(shape)>

<!-- scalability features end -->

```

color.dtd:

```

<!-- Color features -->

```

```

<!ELEMENT color (color_hist | luv_color)*>

<!-- Color histogram feature -->
<!ELEMENT color_hist (color_hist_value, code*)>
<!ATTLIST color_hist
    length CDATA #REQUIRED
    color_space (RGB|OHTA|HSV|LUV) #REQUIRED
    quantization (uniform|non-uniform) #REQUIRED>

<!ELEMENT color_hist_value (#PCDATA)>
<!ATTLIST color_hist_value
    format CDATA #REQUIRED>

<!-- LUV color feature -->
<!ELEMENT luv_color (luv_color_value, code*)>

<!ELEMENT luv_color_value (luv_bin*)>
<!ATTLIST luv_color_value
    length CDATA #IMPLIED>

<!ELEMENT luv_bin ((l, u, v)?)>
<!ATTLIST luv_bin
    l CDATA #REQUIRED
    u CDATA #REQUIRED
    v CDATA #REQUIRED>

<!ELEMENT l (#PCDATA)>
<!ELEMENT u (#PCDATA)>

```

```
<!ELEMENT v (#PCDATA)>

<!-- Color features end -->
```

texture.dtd:

```
<!-- Texture features -->

<!ELEMENT texture (text_tamura*)>

<!-- Tamura texture feature -->
<!ELEMENT text_tamura (tamura_value, code*)>

<!ELEMENT tamura_value (coarseness, contrast, orientation)?>
<!ATTLIST tamura_value
    coarseness CDATA #REQUIRED
    contrast CDATA #REQUIRED
    orientation CDATA #REQUIRED>

<!ELEMENT coarseness (#PCDATA)>
<!ELEMENT contrast (#PCDATA)>
<!ELEMENT orientation (#PCDATA)>

<!-- Texture features end -->
```

shape.dtd:

```
<!-- Shape features -->
```

```

<!ELEMENT shape (eigen_analysis | contour | shape_geom | shape_chkey |
shape_poly | shape_bitmap | shape_fourier)*>

<!-- Eigenvalue analysis shape feature -->
<!ELEMENT eigen_analysis (eigen_analysis_value, code*)>

<!ELEMENT eigen_analysis_value (aspect_ratio, circularity, angle)>

<!ELEMENT aspect_ratio (#PCDATA)>
<!ATTLIST aspect_ratio
    value CDATA #REQUIRED>

<!ELEMENT circularity (#PCDATA)>
<!ATTLIST circularity
    value CDATA #REQUIRED>

<!ELEMENT angle (#PCDATA)>
<!ATTLIST angle
    value CDATA #REQUIRED>

<!-- Outline -->
<!ELEMENT contour (contour_value*, code*)>

<!ELEMENT contour_value (perimeter, outline)>

<!ELEMENT perimeter (#PCDATA)>
<!ATTLIST perimeter
    value CDATA #REQUIRED>

```

```

<!ELEMENT outline (point2d*)>
<!ATTLIST outline
  length CDATA #IMPLIED>

<!ELEMENT point2d (point_x, point_y)?>
<!ATTLIST point2d
  x CDATA #REQUIRED
  y CDATA #REQUIRED>

<!ELEMENT point_x (#PCDATA)>
<!ELEMENT point_y (#PCDATA)>

<!-- External geometry shape features -->
<!ENTITY % shape_geom SYSTEM "shape_geom.dtd">
%shape_geom;

<!-- External shape features DTD based on chromakeyed texture -->
<!ENTITY % shape_chkey SYSTEM "shape_chkey.dtd">
%shape_chkey;

<!-- External polygon shape feature -->
<!ENTITY % shape_polyg SYSTEM "shape_polyg.dtd">
%shape_polyg;

<!-- External bitmap shape feature -->
<!ENTITY % shape_bitmap SYSTEM "shape_bitmap.dtd">
%shape_bitmap;

```

```

<!-- External fourier shape feature -->
<!ENTITY % shape_fourier SYSTEM "shape_fourier.dtd">
%shape_fourier;

<!-- Shape features end -->

```

shape_geom.dtd:

```

<!-- Shape geometry and moment based properties or features -->

<!ELEMENT shape_geom (boundary | sizevalue | symmetry | orientation2d |
orientation3d)*>

<!-- shape geometry based properties: boundary, sizevalue, symmetry -->

<!ELEMENT boundary EMPTY>
<!ATTLIST boundary
    perimeter CDATA #REQUIRED>

<!ELEMENT sizevalue EMPTY>
<!ATTLIST sizevalue
    area CDATA #REQUIRED>

<!ELEMENT symmetry EMPTY>
<!ATTLIST symmetry
    type_fold CDATA #REQUIRED>

<!-- shape moment based properties: orientation2d-->

```

```

<!ELEMENT orientation2d EMPTY>
<!ATTLIST orientation2d
    angle CDATA #REQUIRED)>

<!-- orientation 3D features -->
<!ELEMENT orientation3d (roll, pitch, yaw) >
<!ELEMENT roll EMPTY>
<!ATTLIST roll
    value CDATA #REQUIRED
    unit CDATA #FIXED "DEGREES" >
<!ELEMENT pitch EMPTY>
<!ATTLIST pitch
    value CDATA #REQUIRED
    unit CDATA #FIXED "DEGREES" >
<!ELEMENT yaw EMPTY >
<!ATTLIST yaw
    value CDATA #REQUIRED
    unit CDATA #FIXED "DEGREES" >

<!-- shape property features end -->

```

shape_chkey.dtd:

```

<!-- Shape based on chromakeyed texture features -->
<!ELEMENT shape_chkey (chromakey)>

<!-- chromakey shape features -->
<!ELEMENT chromakey(chromakey_value, code*)>

```



```
<!-- Polygon Shape features end -->
```

shape_bitmap.dtd

```
<!-- Bitmap Shape features -->
```

```
<!ELEMENT shape_bitmap (offset, size, bitmap*)>
```

```
<!-- Offset (top,left) value -->
```

```
<!ELEMENT offset EMPTY>
```

```
<!ATTLIST offset
```

```
  xoff CDATA #REQUIRED
```

```
  yoff CDATA #REQUIRED>
```

```
<!-- Window size -->
```

```
<!ELEMENT winsize EMPTY>
```

```
<!ATTLIST winsize
```

```
  xsize CDATA #REQUIRED
```

```
  ysize CDATA #REQUIRED>
```

```
<!-- Bitmap -->
```

```
<!ELEMENT bitmap EMPTY>
```

```
<!ATTLIST bitmap
```

```
  mask CDATA #REQUIRED>
```

```
<!ELEMENT contourlength EMPTY>
```

```
<!ATTLIST contourlength
```

```
  length CDATA #REQUIRED>
```

```
<!-- Bitmap Shape features end -->
```

shape_fourier.dtd:

```
<!-- DFT Shape features -->

<!ELEMENT shape_fourier (length, quantizer, numcoeff, dft*)>

<!-- Contourlength -->
<!ELEMENT lenght EMPTY>
<!ATTLIST length
    value CDATA #REQUIRED>

<!-- Quantizer stepsize of coefficients -->
<!ELEMENT quantizer EMPTY>
<!ATTLIST quantizer
    value CDATA #REQUIRED>

<!-- Number of Coefficients-->
<!ELEMENT numcoeff EMPTY>
<!ATTLIST numcoeff
    value CDATA #REQUIRED>

<!-- value of each DFT coefficient -->
<!ELEMENT dft EMPTY>
<!ATTLIST dft
    x CDATA #REQUIRED
    y CDATA #REQUIRED
```

```
<!-- DFT Shape features end -->
```

size.dtd:

```
<!-- Size features -->

<!-- Size -->
<ELEMENT size (size_dimensions | size_num_pixels)>

<!-- Dimensions (X,Y) -->
<ELEMENT size_dimensions (size_x, size_y)?>
<ATTLIST size_dimensions
    x CDATA #REQUIRED
    y CDATA #REQUIRED>

<ELEMENT size_x (#PCDATA)>
<ELEMENT size_y (#PCDATA)>

<!-- Number of pixels -->
<ELEMENT size_num_pixels (#PCDATA)>
<ATTLIST size_num_pixels
    value CDATA #REQUIRED>

<!-- Size features end -->
```

position.dtd:

```
<!-- Position features -->
```

```

<!ELEMENT position (segmentation_mask_analysis*)>

<!-- Segmentation mask position feature -->
<!ELEMENT segmentation_mask_analysis (segmentation_mask_analysis_value,
code*)>

<!ELEMENT segmentation_mask_analysis_value (left-top_vertex, centroid)>

<!ELEMENT left-top_vertex EMPTY>
<!ATTLIST left-top_vertex
    x CDATA #REQUIRED
    y CDATA #REQUIRED>

<!ELEMENT centroid EMPTY>
<!ATTLIST centroid
    x CDATA #REQUIRED
    y CDATA #REQUIRED>

<!-- Position features end -->

```

첨부서류 B: 도 3의 인텍싱 계층 및 복수의 추상화 레벨들에 대한 XML의 예

ex_index_hierarchy.xml:

```

<image>
  <image_object_set>
    <!-- List of objects in the image -->
    <image_object type="LOCAL" id="o1">
      <size> <size_num_pixels> 120 </size_num_pixels> </size>
    </image_object>
    <image_object type="LOCAL" id="o2">
      <size> <size_num_pixels> 140 </size_num_pixels> </size>
    </image_object>
    <image_object type="LOCAL" id="o3">
      <size> <size_num_pixels> 45 </size_num_pixels> </size>
    </image_object>
    <image_object type="LOCAL" id="o4">
      <size> <size_num_pixels> 35 </size_num_pixels> </size>
    </image_object>
    <image_object type="LOCAL" id="o5">
      <size> <size_num_pixels> 9 </size_num_pixels> </size>
    </image_object>
    <!-- Intermediate nodes in the indexing hierarchy -->
    <image_object type="LOCAL" id="o6">
      <size> <size_num_pixels /></size>
    </image_object>
    <image_object type="LOCAL" id="o7">

```

```

<size> <size_num_pixels>
    <less_than> 50 </less_than>
</size_num_pixels> </size>
</image_object>
<image_object type="LOCAL" id="o8">
    <size> <size_num_pixels>
        <less_than> 10 </less_than>
    </size_num_pixels> </size>
</image_object>
</image_object_set>
<!-- Indexing hierarchy based on area of objects (number of pixels) -->
<!-- The first object of the hierarchy contains the features used in the indexing
-->
<!-- The other intermediate nodes include constraints on the feature values -->
<object_hierarchy>
    <object_node id="on1" object_ref="o6">
        <object_node id="on2" object_ref="o1" />
        <object_node id="on3" object_ref="o2" />
        <object_node id="on4" object_ref="o7">
            <object_node id="on5" object_ref="o3" />
            <object_node id="on6" object_ref="o4" />
            <object_node id="on7" object_ref="o8">
                <object_node id="on8" object_ref="o5" />
            </object_node>
        </object_node>
    </object_node>
</object_node>
</object_hierarchy>
</image>

```

도면의 간단한 설명

도 1a는 본 발명의 이미지 디스크립션 시스템의 이미지의 일례를 나타내는 도면이다.

도 1b는 본 발명의 이미지 디스크립션 시스템의 객체 계층(object hierarchy)의 일례를 나타내는 도면이다.

도 1c는 본 발명의 이미지 디스크립션 시스템의 엔티티 관계 그래프(entity relation graph)의 일례를 나타내는 도면이다.

도 2는 본 발명의 이미지 디스크립션 시스템의 일례를 나타내는 블록도이다.

도 3a는 본 발명의 이미지 디스크립션 시스템의 객체 계층의 일례를 나타내는 도면이다.

도 3b는 본 발명의 이미지 디스크립션 시스템의 객체 계층의 다른 예를 나타내는 도면이다.

도 4a는 본 발명의 이미지 디스크립션 시스템의 이미지의 일례를 나타내는 도면이다.

도 4b는 본 발명의 이미지 디스크립션 시스템의 클러스터링 계층(clustering hierarchy)의 일례를 나타내는 도면이다.

도 5는 본 발명의 이미지 디스크립션 시스템의 일례를 나타낸 블록도이다.

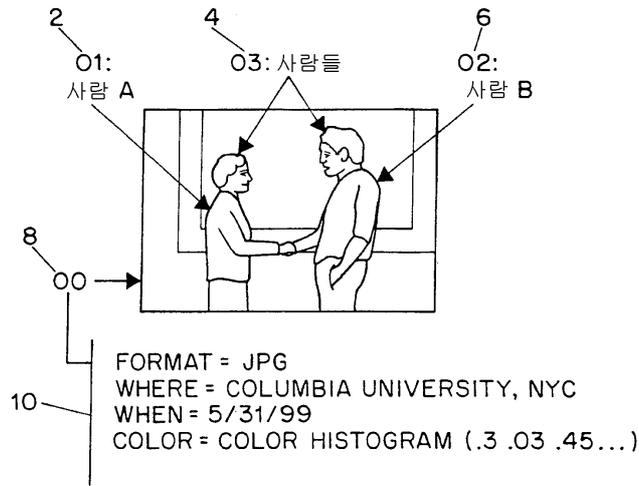
도 6은 본 발명의 이미지 디스크립션 시스템의 프로세스 흐름을 나타낸 블록도이다.

도 7은 본 발명의 이미지 디스크립션 시스템의 일례를 나타내는 블록도이다.

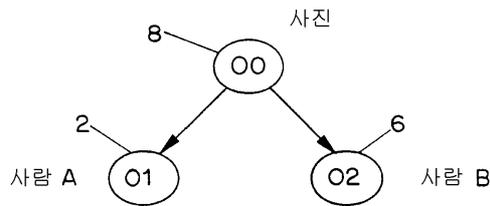
도 8은 본 발명의 이미지 디스크립션 시스템의 다른 예를 나타내는 블록도이다.

도면

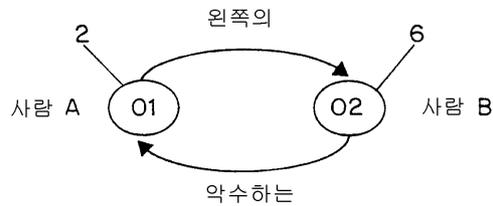
도면1a



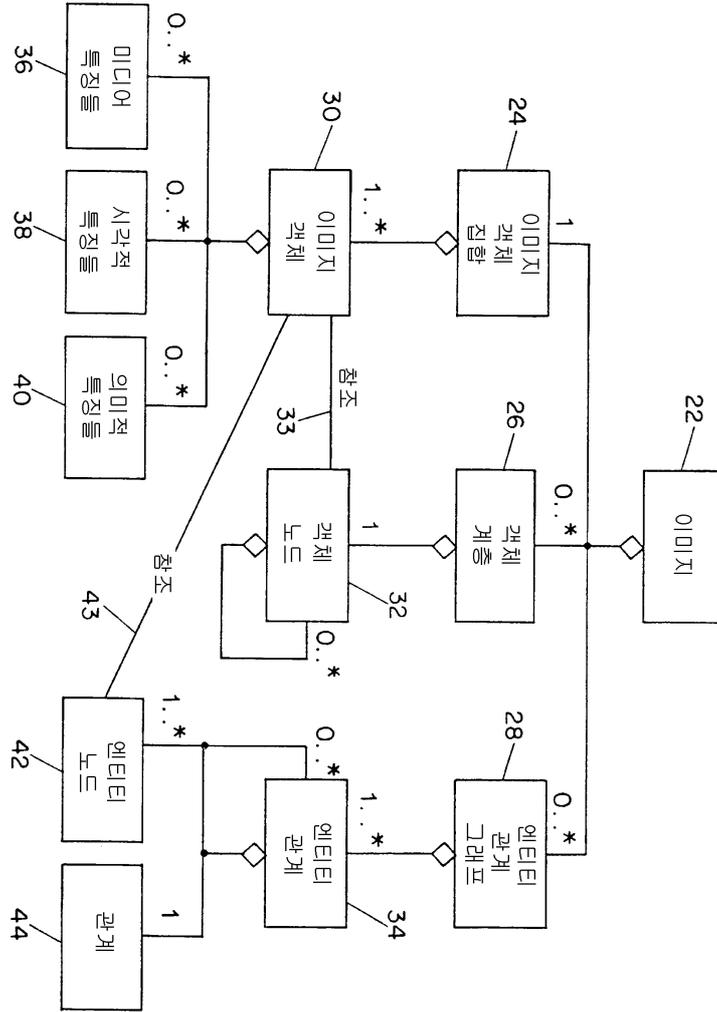
도면1b



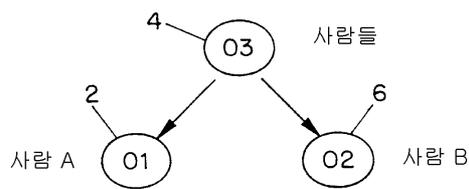
도면1c



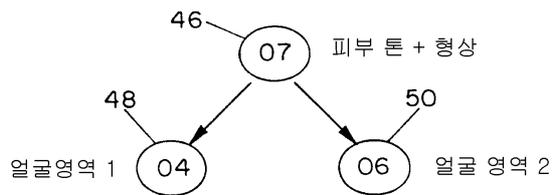
도면2



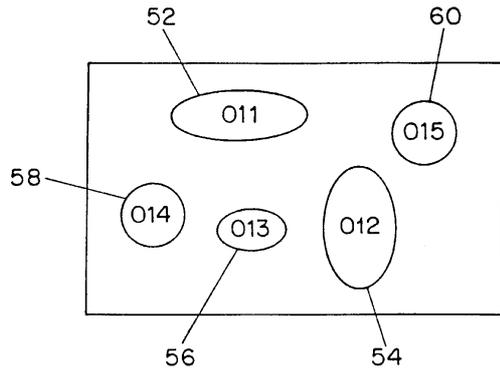
도면3a



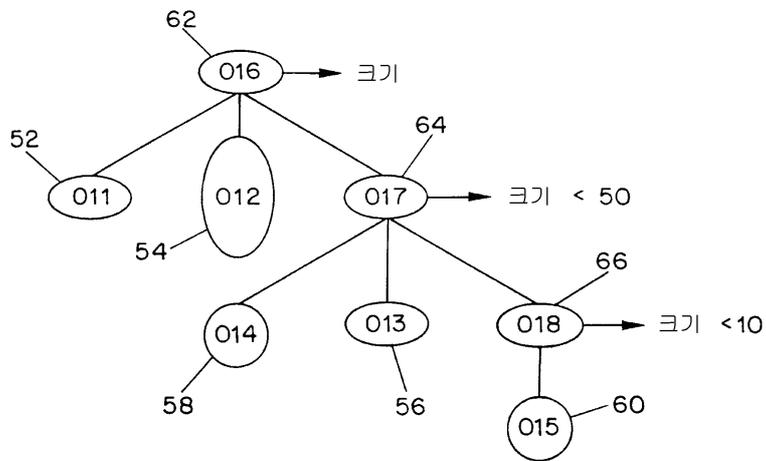
도면3b



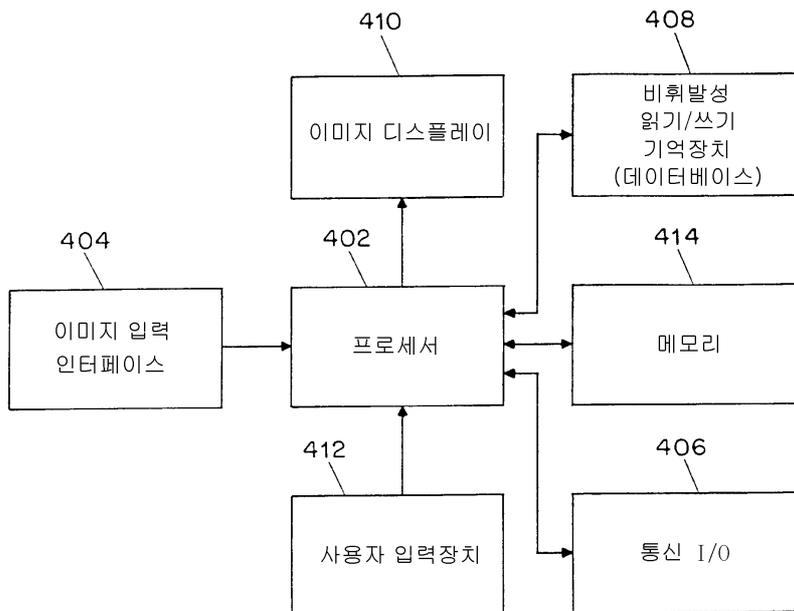
도면4a



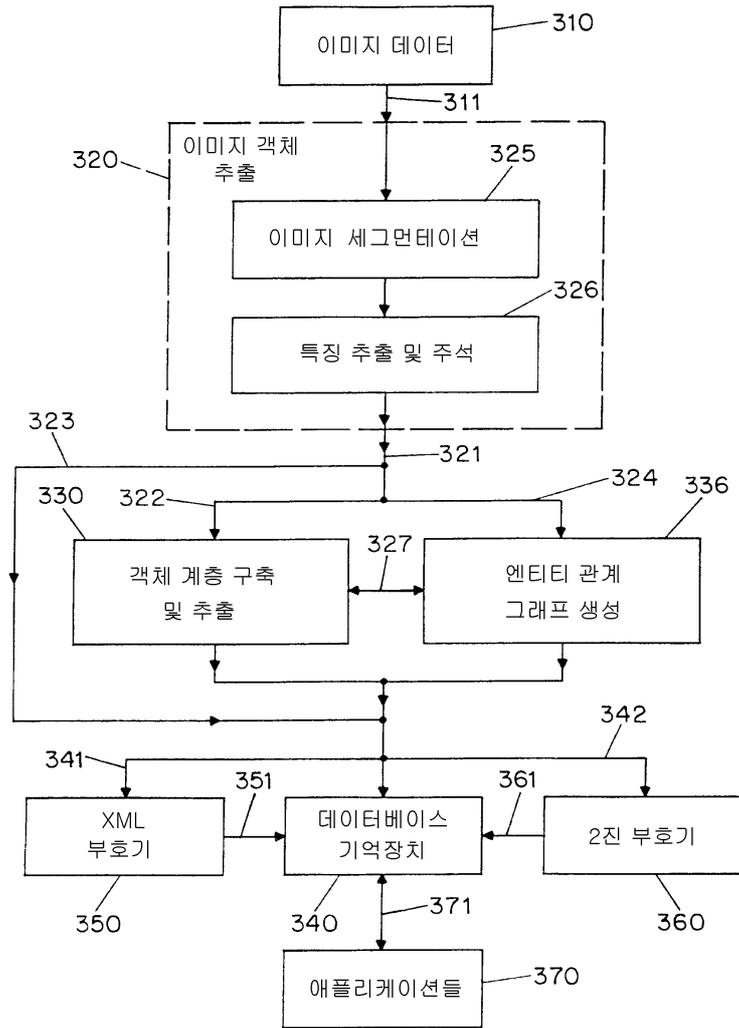
도면4b



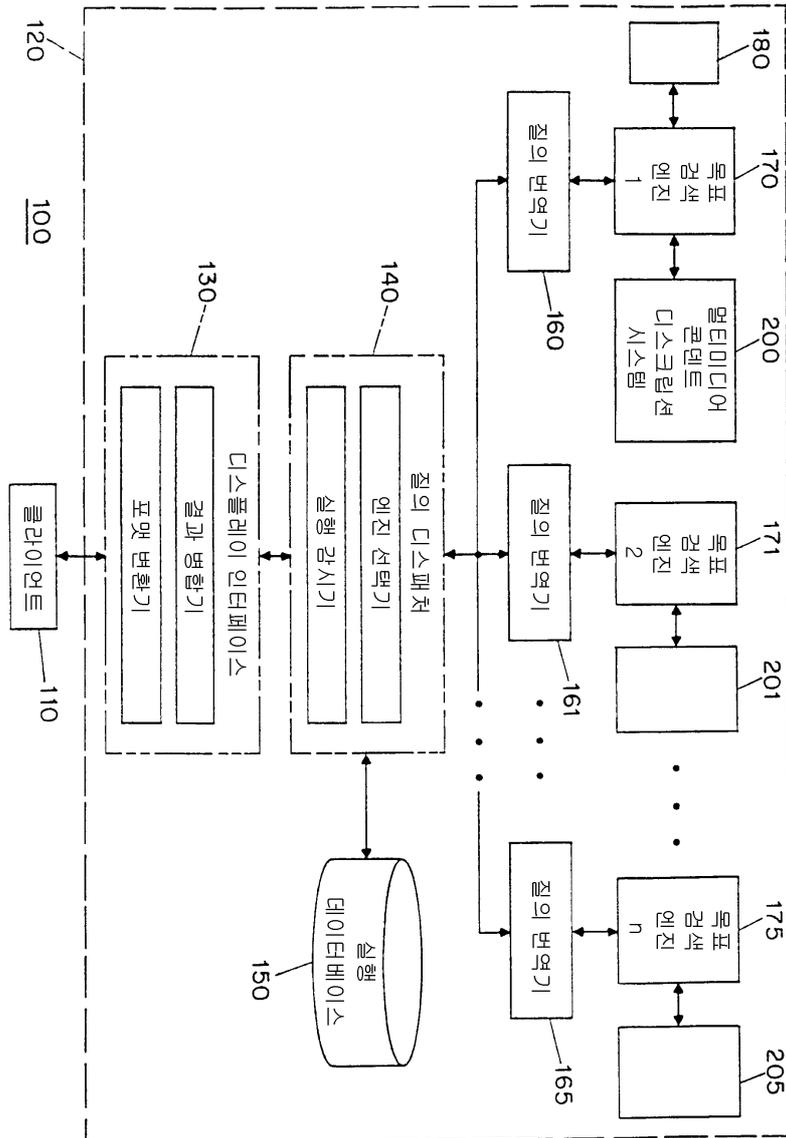
도면5



도면6



도면7



8페이지

