(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0105908
(43) 공개일자 2013년09월26일

(51) 국제특허분류(Int. Cl.)
H04N 7/26 (2006.01) *G11B 20/10* (2006.01)
(21) 출원번호 10-2013-7020778(분할)
(22) 출원일자(국제) 2004년12월08일
심사청구일자 없음
(62) 원출원 특허 10-2012-7033797
원출원일자(국제) 2004년12월08일
심사청구일자 2012년12월26일
(85) 번역문제출일자 2013년08월06일
(86) 국제출원번호 PCT/US2004/041667
(87) 국제공개번호 WO 2005/057906
국제공개일자 2005년06월23일
(30) 우선권주장
10/731,809 2003년12월08일 미국(US)

(71) 출원인
디브이엑스, 엘엘씨
미국 캘리포니아 95050 산타 클라라 데 라 크루즈 블러바드 2830
(72) 발명자
리 아담 에이치.
미국 92121 캘리포니아주 샌디에고 아파트먼트 152 워터리지 서클 10194
브라네스 제이슨
미국 92127 캘리포니아주 샌디에고 콩코드 리지 테라스 15885
(뒷면에 계속)
(74) 대리인
유미특허법인

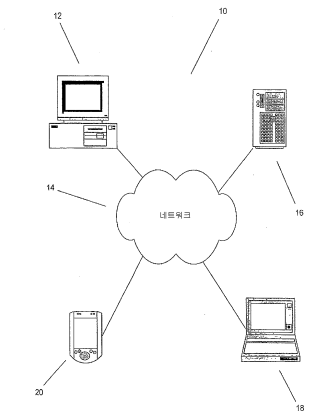
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **멀티미디어 분배 시스템**

(57) 요 약

본 발명은 멀티미디어 파일과 이러한 멀티미디어 파일을 생성, 분배 및 이용하는 방법에 관한 것이다. 본 발명에 따른 멀티미디어 파일은 이 파일의 컨텐츠에 관련된 "메타데이터"와 파일의 컨텐츠에 액세스하기 위한 메뉴 인터페이스를 생성하는데 이용될 수 있는, 다수의 비디오 트랙, 다수의 오디오 트랙, 다수의 서브타이틀 트랙, 및 데이터를 얻을 수 있다. 본 발명에 따른 멀티미디어 파일은 비디오 트랙, 오디오 트랙, 서브타이틀 트랙 및 파일 외부의 "메타데이터"를 포함한다. 본 발명의 멀티미디어 파일의 실시예에서는 복수 개의 부호화된 비디오 트랙을 포함한다.

대 표 도 - 도1



(72) 발명자

더럼 크리스토퍼

미국 92128 캘리포니아주 샌디에고 아파트먼트 100 스토니 피크 드라이브 11729

세이어 티모시

미국 92128 캘리포니아주 샌디에고 이글스 크릭 코트 11202

코테 벤자민

미국 92056 캘리포니아주 우스터 드라이브 오션사이드 4045

톰슨 다리우스

미국 92007 캘리포니아주 카디프 바호 스트리트 440

로타 제롬

미국 92129 캘리포니아주 샌디에고 도마우스 로드 12396

마요 알리스테어

미국 92117 캘리포니아주 샌디에고 클레어몬트 드라이브 3114

그랍 에릭 윌리엄

미국 92106 캘리포니아주 샌디에고 아파트먼트 104 로즈크랜스 스트리트 1220

도날드슨 제랄드

미국 92117 캘리포니아주 샌디에고 노르위치 스트리트 5515

히치콕 헤더

미국 92024 캘리포니아주 엔시니타스 힐크레스트 드라이브 376

차바리아 데미안

프랑스 34600 베다리에우스 슈만 드 디오 6

플로이드 마이클 존

미국 92128 캘리포니아주 샌디에고 솔 서미트 드라이브 13765

버트 아산

미국 92604 캘리포니아주 어바인 헤리티지 4

퍼지 브라이언

미국 90069 캘리포니아주 로스앤젤레스 산타 모니카 블러버드 #1211 8721

특허청구의 범위

청구항 1

도 1에 도시된 바와 같은 멀티미디어 파일 복호화 시스템.

명 세 서

기 술 분 야

[0001] 본 발명은 멀티미디어 파일의 부호화, 전송 및 복호화에 관한 것으로서, 더 구체적으로는, 단일의 오디오 트랙 및 단일의 비디오 트랙 외에 다른 트랙을 포함할 수 있는 멀티미디어 파일의 부호화, 전송 및 복호화에 관한 것이다.

배 경 기 술

[0002] 인터넷이 발달함에 따라, 표준화된 생성, 분배 및 표시 가능한 멀티미디어 정보용의 파일의 형식에 대한 개발도 촉진되고 있다. 통상적으로, 단일의 멀티미디어 파일은 단일의 비디오 트랙과 단일의 오디오 트랙을 포함한다. 멀티미디어 정보가, 고용량의 DVD(Digital Video Disk) 등의 휴대 가능한 미디어에 기록되어 있는 경우, 많은 수의 비디오 트랙, 오디오 트랙 및 서브타이틀(subtitle) 트랙을 제공하기 위하여 다수의 파일(multiple file)이 이용될 수 있다. 대화형(interactive) 메뉴를 생성하는데 이용될 수 있는 정보를 포함하는 추가의 파일이 제공될 수 있다.

발명의 내용

해결하려는 과제

[0003] 본 발명은 멀티미디어 파일과, 이러한 멀티미디어 파일을 생성, 분배 및 복호화하기 위한 시스템을 제공하는 것이다.

과제의 해결 수단

[0004] 본 발명의 실시예는 멀티미디어 파일과, 이러한 멀티미디어 파일을 생성, 분배 및 복호화하기 위한 시스템을 포함한다. 본 발명의 특징에 의하면, 멀티미디어 파일은 복수 개의 부호화된 비디오 트랙(video track)을 포함한다. 본 발명의 다른 특징에 의하면, 멀티미디어 파일은 복수 개의 부호화된 오디오 트랙(audio track)을 포함한다. 본 발명의 또 다른 특징에서는, 멀티미디어 파일이 하나 이상의 서브타이틀 트랙을 포함한다. 또 다른 특징으로서, 멀티미디어 파일은 부호화된 "메타데이터"(meta data)를 포함한다. 또 다른 특징으로서, 멀티미디어 파일은 부호화된 메뉴 정보를 포함한다.

[0005] 본 발명의 실시예에 따른 멀티미디어 파일은, 복수 개의 부호화된 비디오 트랙을 포함한다. 본 발명의 다른 실시예에서는, 멀티미디어 파일이 복수 개의 연결된 'RIFF'(Resource Interchange File Format) 청크를 포함하며, 각각의 부호화된 비디오 트랙은 개별의 'RIFF' 청크에 포함된다. 또한, 비디오는 사이코비주얼 인핸스먼트(psychovisual enhancement)를 이용하여 비디오를 부호화하고, 각각의 비디오 트랙은 적어도 하나의 오디오 트랙과 연관된다.

[0006] 다른 실시예에서, 각각의 비디오 트랙은 'RIFF' 청크 내에서 일련의 'video' 청크로서 부호화되며, 각각의 비디오 트랙과 동반하는 오디오 트랙은, 연관된 비디오 트랙의 'video' 청크를 포함하는 'RIFF' 청크 내에서 인터리브된 일련의 'audio' 청크로서 부호화된다. 또한, 각각의 'video' 청크는 비디오 트랙으로부터 단일 프레임의 비디오를 생성하는데 이용될 수 있는 정보를 포함하며, 각각의 'audio' 청크는, 오디오 트랙의 'video' 청크를 이용하여 생성된 프레임에 동반하는 부분으로부터 오디오 정보를 포함한다. 또한, 'audio' 청크는 'RIFF' 청크 내에서 대응하는 'video' 청크의 앞에 인터리브된다.

[0007] 본 발명의 실시예에 따라, 멀티미디어 파일을 부호화하는 시스템은 복수 개의 비디오 트랙을 부호화하고, 부호화된 비디오 트랙을 연결하며, 부호화 및 연결된 비디오 트랙을 단일의 파일에 기록하도록 구성된 프로세서를 포함한다. 다른 실시예에서, 프로세서는 각각의 비디오 트랙이 개별의 'RIFF' 청크 내에 포함되도록 비디오 트

랙을 부호화하도록 구성되어 있다. 또, 프로세서는 사이코비주얼 인핸스먼트를 이용하여 부호화되도록 구성되어 있다. 또한, 각각의 비디오 트랙에는 각각 오디오 트랙이 하나 이상 관련되어 있다.

[0008] 다른 실시예에서, 프로세서는, 각각의 비디오 트랙을 'RIFF' 청크 내에서 일련의 'video' 청크로서 부호화하며, 각각의 비디오 트랙에 동반하는 하나 이상의 오디오 트랙을, 관련된 비디오 트랙의 'video' 청크를 포함하는 'RIFF' 청크 내에서 인터리브된 일련의 '오디오' 청크로서 부호화하도록 구성되어 있다.

[0009] 또 다른 실시예에서, 프로세서는, 각각의 'video' 청크가 비디오 트랙으로부터 단일 프레임의 비디오를 생성하는데 이용될 수 있도록 비디오 트랙을 부호화하며, 각각의 '오디오' 청크가 비디오 트랙으로부터 생성된 'video' 청크를 이용하여 생성된 프레임에 동반하는 오디오 트랙 부분으로부터 오디오 정보를 포함하도록, 비디오 트랙과 관련된 오디오 트랙을 부호화하도록 구성되어 있다. 또한, 프로세서는 'RIFF' 청크 내에서 대응하는 'video' 청크의 앞에 각각의 '오디오' 청크를 인터리브한다.

[0010] 본 발명의 실시예에 따라, 복수 개의 부호화된 비디오 트랙을 포함하는 멀티미디어 파일을 복호화하는 시스템은, 멀티미디어 파일로부터 정보를 추출하도록 구성된 프로세서를 포함한다. 이 프로세서는 멀티미디어 파일 내에 포함된 부호화된 비디오 트랙의 수에 관련된 정보를 추출하도록 구성된다.

[0011] 다른 실시예에서, 프로세서는 부호화된 비디오 트랙을 'RIFF' 청크 내에서 찾아내도록 구성된다. 또한, 부호화된 비디오 트랙 중, 부호화된 제1 비디오 트랙은 표준 4cc 코드(FourCC code)를 갖는 제1 'RIFF' 청크 내에 포함되며, 부호화된 제2 비디오 트랙은 전용의 4cc 코드를 갖는 제2 'RIFF' 청크 내에 포함되고, 전용의 4cc 코드는 표준 4cc 코드의 처음 2개의 문자를 자신의 마지막 2개의 문자로서 갖는다.

[0012] 다른 실시예에서, 각각의 부호화된 비디오 트랙은 개별의 'RIFF' 청크 내에 포함된다.

[0013] 다른 실시예에서, 복호화된 비디오 트랙은 멀티미디어 파일을 생성할 때 부호화되었던 원래의 비디오 트랙과 유사하고, 복호화된 비디오 트랙과 원래의 비디오 트랙 간의 편차 중 적어도 일부는, 비디오 트랙의 프레임 중 어두운 부분에 위치하며, 복호화된 비디오 트랙과 원래의 비디오 트랙 간의 편차 중 적어도 일부는, 비디오 트랙의 고속 동작 장면에 위치한다.

[0014] 추가의 실시예에서, 각각의 비디오 트랙은 적어도 하나의 오디오 트랙과 연관되어 있다.

[0015] 다른 실시예에서, 프로세서는, 'RIFF' 청크 내에서 일련의 'video' 청크를 복호화함으로써 비디오 트랙으로부터 비디오를 표시하며, 비디오 트랙의 'video' 청크를 포함하는 'RIFF' 청크 내에서 인터리브된 일련의 'audio' 청크를 복호화함으로써, 각각의 비디오 트랙에 동반하는 오디오 트랙으로부터 오디오를 생성하도록 구성된다.

[0016] 또 다른 실시예에서, 프로세서는, 비디오 트랙의 단일 프레임을 생성하기 위하여 각각의 'video' 청크로부터 추출된 정보를 이용하며, 'video' 청크를 이용하여 생성된 프레임과 동반하는 오디오 트랙의 일부를 생성하기 위하여 각각의 'audio' 청크로부터 추출된 정보를 이용하도록 구성된다. 또한, 프로세서는 'RIFF' 청크 내에 연관되어 있는 'video' 청크의 앞에서 'audio' 청크를 찾아내도록 구성된다.

[0017] 본 발명의 실시예에 따른 멀티미디어 파일은, 일련의 부호화된 비디오 프레임; 및 부호화된 비디오 프레임 사이에서 인터리브한 부호화된 오디오를 포함하며, 부호화된 오디오는 2개 이상의 오디오 정보 트랙을 포함한다.

[0018] 또 다른 실시예에서, 오디오 정보 트랙의 적어도 하나는 복수 개의 오디오 채널을 포함한다.

[0019] 다른 실시예에서, 멀티미디어 파일에 포함된 오디오 파일의 수를 식별하는 헤더 정보를 추가로 포함하고, 오디오 정보 트랙의 적어도 하나에 관한 기술 정보(description information)를 포함하는 헤더 정보를 추가로 포함한다.

[0020] 또 다른 실시예에서, 각각의 부호화된 비디오 프레임은 부호화된 오디오 정보에 의해 처리되며, 비디오 프레임의 앞에 배치되는 부호화된 오디오 정보는, 부호화된 비디오 프레임과 동반하는 각각의 오디오 트랙의 일부에 대한 오디오 정보를 포함한다.

[0021] 다른 실시예에서, 비디오 정보는 멀티미디어 파일 내에 청크로서 저장된다. 또 다른 실시예에서, 비디오 정보의 각각의 청크는 단일의 비디오 프레임을 포함한다. 또, 오디오 정보는 상기 멀티미디어 파일 내에 청크로서 저장되며, 2개의 개별적인 오디오 트랙으로부터의 오디오 정보는 오디오 정보의 단일의 청크 내에 포함되지 않는다.

[0022] 또 다른 실시예에서, 비디오 청크는 상기 각각의 오디오 트랙으로부터 적어도 하나의 오디오 청크에 의해 분리

된다. 또, 비디오 청크를 분리하는 오디오 청크는, 오디오 청크에 따르는 비디오 청크 내에 포함된 비디오 정보와 동반하는 오디오 청크의 일부에 대한 오디오 정보를 포함한다.

[0023] 본 발명의 실시예에 따라, 멀티미디어 파일을 부호화하는 시스템은, 비디오 트랙을 부호화하며, 복수 개의 오디오 트랙을 부호화하고, 비디오 트랙으로부터의 정보를, 복수 개의 오디오 트랙으로부터의 정보와 인터리브하며, 인터리브된 비디오 및 오디오 정보를 단일의 파일에 기록하도록 구성된 프로세서를 구비한다.

[0024] 다른 실시예에서, 오디오 트랙의 적어도 하나는 복수 개의 오디오 채널을 포함한다.

[0025] 또 다른 실시예에서, 프로세서는, 부호화된 오디오 트랙의 수를 식별하는 헤더 정보를 부호화하며, 헤더 정보를 단일의 파일에 기록하도록 구성된다.

[0026] 또 다른 실시예에서, 프로세서는, 부호화된 오디오 트랙의 적어도 하나에 관한 기술 정보를 식별하는 헤더 정보를 부호화하며, 헤더 정보를 단일의 파일에 기록하도록 구성된다.

[0027] 다른 실시예에서, 프로세서는 비디오 트랙을 비디오 정보의 청크로서 부호화한다. 또한, 프로세서는 각각의 오디오 트랙을 오디오 정보의 일련의 청크로서 부호화한다.

[0028] 다른 실시예에서, 오디오 정보의 각각의 청크는 단일의 오디오 트랙으로부터 오디오 정보를 포함한다. 또한, 프로세서는 오디오 정보의 청크를 비디오 정보의 청크 사이에 인터리브하도록 구성된다.

[0029] 다른 실시예에서, 프로세서는, 'video' 청크 내의 비디오 정보와 동반하는 각각의 오디오 트랙의 일부를 'audio' 청크로 부호화하도록 구성되며, 'video' 청크를 'audio' 청크와 인터리브하여, 'video' 청크에 포함된 비디오 정보와 동반하는 각각의 오디오 트랙으로부터 오디오 정보를 포함하는 'audio' 청크가 각각의 'video' 청크보다 앞서 배치되도록 구성된다. 또한, 프로세서는, 단일의 비디오 프레임이 각각의 비디오 청크 내에 포함되도록, 비디오 트랙을 부호화한다.

[0030] 다른 실시예에서, 프로세서는 범용의 프로세서이다.

[0031] 또 다른 실시예에서, 프로세서는 전용 회로이다.

[0032] 본 발명의 실시예에 따라, 복수 개의 오디오 트랙을 포함하는 멀티미디어 파일을 복호화하는 시스템은, 멀티미디어 파일로부터 정보를 추출하도록 구성된 프로세서를 포함하며, 프로세서는 멀티미디어 파일 내에 포함된 오디오 트랙의 수에 관한 정보를 추출하도록 구성된다.

[0033] 다른 실시예에서, 프로세서는 복수 개의 오디오 트랙으로부터 단일의 오디오 트랙을 선택하도록 구성되며, 프로세서는 선택된 오디오 트랙으로부터 오디오 정보를 복호화하도록 구성된다.

[0034] 다른 실시예에서, 오디오 트랙의 적어도 하나는 복수 개의 오디오 채널을 포함한다.

[0035] 또 다른 실시예에서, 프로세서는 오디오 트랙의 적어도 하나에 관한 기술 정보를 포함하는 멀티미디어 파일에 있는 헤더로부터 정보를 추출하도록 구성된다.

[0036] 본 발명의 실시예에 따라, 멀티미디어 정보를 통신하기 위한 시스템은, 네트워크; 멀티미디어 파일을 포함하며, 서버를 통해 네트워크에 접속되는 기억 장치; 및 네트워크에 접속되는 클라이언트를 포함하며, 클라이언트는 서버로부터 멀티미디어 파일의 전송을 요청할 수 있고, 멀티미디어 파일은 적어도 하나의 비디오 트랙과 비디오 트랙과 동반하는 복수 개의 오디오 트랙을 포함한다.

[0037] 본 발명에 따른 멀티미디어 파일은, 일련의 부호화된 비디오 프레임; 및 부호화된 비디오 프레임 사이에 인터리브된 적어도 하나의 부호화된 서브타이틀 트랙을 포함한다.

[0038] 다른 실시예에서, 적어도 하나의 부호화된 서브타이틀 트랙은 복수 개의 부호화된 서브타이틀 트랙을 포함한다.

[0039] 또 다른 실시예에서, 멀티미디어 파일에 포함된 부호화된 서브타이틀 트랙의 수를 식별하는 헤더 정보를 추가로 포함한다.

[0040] 다른 실시예에서, 부호화된 서브타이틀 트랙의 적어도 하나에 대한 기술 정보를 갖는 헤더 정보를 추가로 포함한다.

[0041] 다른 실시예에서, 각각의 서브타이틀 트랙은 일련의 비트맵을 포함한다.

[0042] 또 다른 실시예에서, 각각의 서브타이틀 트랙은 일련의 압축된 비트맵을 포함한다. 또한, 각각의 비트맵은 런

랭스 부호화(run length encoding)를 이용하여 압축된다.

[0043] 다른 실시예에서, 일련의 부호화된 비디오 프레임은 일련의 비디오 청크로서 부호화되며, 각각의 부호화된 서브타이틀 트랙은 일련의 서브타이틀 청크로서 부호화되고, 각각의 서브타이틀 청크는 디스플레이 상에 텍스트로서 표현될 수 있는 정보를 포함한다.

[0044] 다른 실시예에서, 각각의 서브타이틀 청크는 단일의 서브타이틀에 관한 정보를 포함한다.

[0045] 또 다른 실시예에서, 각각의 서브 타이틀 청크는 비디오 시퀀스의 서브타이틀이 중첩되어야 하는 부분에 관한 정보를 포함한다. 또한, 각각의 서브타이틀 청크는 디스플레이의 서브타이틀이 위치하여야 하는 부분에 관한 정보를 포함한다.

[0046] 다른 실시예에서, 각각의 서브타이틀 청크는 상기 서브타이틀의 컬러에 관한 정보를 포함한다. 또한, 컬러에 관한 정보는 컬러 팔레트를 포함한다. 또한, 서브타이틀 청크는, 제1 컬러 팔레트에 관련된 정보를 포함하는 제1 서브타이틀 청크와, 제1 컬러 팔레트에 관련된 정보를 대체하는 제2 컬러 팔레트에 관한 정보를 포함하는 제2 서브타이틀 청크를 구비한다.

[0047] 본 발명의 실시예에 따라, 멀티미디어 파일을 부호화하는 시스템은, 비디오 트랙을 부호화며, 적어도 하나의 서브타이틀 트랙을 부호화하고, 비디오 트랙으로부터의 정보를 적어도 하나의 서브타이틀 트랙으로부터의 정보와 인터리브하며, 인터리브된 비디오 및 서브타이틀 정보를 단일의 파일에 기록하도록 구성된 프로세서를 포함한다.

[0048] 다른 실시예에서, 적어도 하나의 서브타이틀 트랙은 복수 개의 서브타이틀 트랙을 포함한다.

[0049] 또 다른 실시예에서, 프로세서는, 멀티미디어 파일에 포함된 서브타이틀 트랙의 수를 식별하는 헤더 정보를 부호화하여 단일의 파일에 기록하도록 구성되어 있다.

[0050] 다른 실시예에서, 프로세서는, 적어도 하나의 서브타이틀 트랙에 관련된 기술 정보를 부호화하여 단일의 파일에 기록하도록 구성되어 있다.

[0051] 다른 실시예에서, 비디오 트랙은 비디오 청크로서 부호화되며, 적어도 하나의 서브타이틀 트랙의 각각은 서브타이틀 청크로서 부호화된다. 또한, 서브타이틀 청크의 각각은 비디오 트랙의 일부와 동반하는 단일의 서브타이틀을 포함하며, 인터리버는 서브타이틀 청크 내의 서브타이틀과 동반하는 비디오 트랙의 일부를 포함하는 비디오 청크의 앞에 각각의 서브타이틀 청크를 인터리브하도록 구성된다.

[0052] 다른 실시예에서, 프로세서는 서브타이틀을 비트맵으로서 부호화하여 서브타이틀 청크를 생성하도록 구성된다.

[0053] 다른 실시예에서, 서브타이틀은 압축된 비트맵으로서 부호화된다.

[0054] 또 다른 실시예에서, 비트맵은 런 랭스 부호화를 이용하여 압축된다. 또한, 프로세서는 각각의 서브타이틀 청크에, 서브타이틀이 중첩되어야 하는 디스플레이의 부분에 관련된 정보를 포함한다.

[0055] 또 다른 실시예에서, 프로세서는, 각각의 서브타이틀 청크에, 서브타이틀이 위치되어야 하는 디스플레이의 부분에 관련된 정보를 포함한다. 프로세서는, 각각의 서브타이틀 청크에, 서브타이틀의 컬러에 관한 정보를 포함한다.

[0056] 또 다른 실시예에서, 컬러 관련 정보는 컬러 팔레트를 포함한다. 또한, 서브타이틀 청크는 제1 컬러 팔레트에 관한 정보를 포함하는 제1 서브타이틀 청크와, 제1 컬러 팔레트에 관련된 정보를 대체하는 제2 컬러 팔레트에 관한 정보를 포함하는 제2 서브타이틀 청크를 구비한다.

[0057] 본 발명의 실시예에 따라, 멀티미디어 파일을 복호화하는 시스템은, 상기 멀티미디어 파일로부터 정보를 추출하도록 구성된 프로세서를 포함하며, 프로세서는 적어도 하나의 서브타이틀 트랙이 존재하는지 여부를 판정하도록 멀티미디어 파일을 조사하도록 구성된다.

[0058] 다른 실시예에서, 적어도 하나의 서브타이틀 트랙은, 복수 개의 서브타이틀 트랙과, 멀티미디어 파일 내의 서브타이틀 트랙의 수를 판정하도록 구성된 프로세서를 구비한다.

[0059] 다른 실시예에서, 프로세서는 멀티미디어 파일로부터 서브타이틀 트랙의 수를 식별하는 헤더 정보를 추출하도록 구성된다.

[0060] 또 다른 실시예에서, 프로세서는 멀티미디어 파일로부터 적어도 하나의 서브타이틀 트랙에 관한 기술 정보를 추

출하도록 구성된다.

[0061] 다른 실시예에서, 멀티미디어 파일은 비디오 청크로서 부호화된 적어도 하나의 비디오 트랙을 포함하며, 멀티미디어 파일은 서브타이틀 청크로서 부호화된 적어도 하나의 서브타이틀 트랙을 포함한다. 또한, 각각의 서브타이틀 청크는 단일의 서브타이틀에 관련된 정보를 포함한다.

[0062] 또 다른 실시예에서, 각각의 서브타이틀은 서브타이틀 청크에 비트맵으로 부호화되며, 프로세서는 비디오 트랙을 복호화하도록 구성되고, 프로세서는 비디오 시퀀스의 일부분의 위에 비트맵을 중첩시킴으로써 비디오 프레임을 표시하도록 구성된다.

[0063] 다른 실시예에서, 서브타이틀은 압축된 비트맵으로서 부호화되며, 프로세서는 비트맵을 압축해제하도록 구성된다. 또한, 프로세서는 런 랭스 부호화(run length encoding)된 비트맵을 압축해제하도록 구성된다.

[0064] 다른 실시예에서, 각각의 서브타이틀 청크는 비디오 트랙의 서브타이틀이 중첩되어야 하는 부분에 관한 정보를 포함하며, 프로세서는 서브타이틀 청크 내의 정보에 의해 표시되는 각각의 비디오 프레임 상에 서브타이틀의 비트맵을 중첩시킴으로써 표시를 위한 비디오 프레임의 시퀀스를 생성하도록 구성된다.

[0065] 또 다른 실시예에서, 각각의 서브타이틀 청크는, 서브타이틀이 위치하여야 하는 프레임 내의 위치에 관련된 정보를 포함하며, 프로세서는 서브타이틀 청크 내의 정보에 의해 표시된 각각의 비디오 프레임 내의 위치에 서브타이틀을 중첩시키도록 구성된다.

[0066] 다른 실시예에서, 각각의 서브타이틀 청크는 서브타이틀의 컬러에 관련된 정보를 포함하며, 프로세서는 서브타이틀 청크 내의 컬러 정보에 의해 표시되는 하나 이상의 컬러에 서브타이틀을 중첩시키도록 구성된다.

[0067] 또 다른 실시예에서, 서브타이틀 청크 내의 컬러 정보는 컬러 팔레트를 포함하며, 프로세서는 서브타이틀의 비트맵 내에서 이용되는 컬러 정보를 획득하기 위하여 컬러 팔레트를 이용하여 서브타이틀을 중첩시키도록 구성된다.

[0068] 다른 실시예에서, 서브타이틀 청크는 제1 컬러 팔레트에 관한 정보를 포함하는 제1 서브타이틀 청크와, 제2 컬러 팔레트에 관한 정보를 포함하는 제2 서브타이틀 청크를 구비하며, 프로세서는 제1 청크가 처리된 이후 서브타이틀의 비트맵에 이용된 컬러에 관한 정보를 획득하기 위하여 제1 컬러 팔레트를 이용하여 서브타이틀을 중첩시키도록 구성되며, 제2 청크가 처리된 이후 서브타이틀의 비트맵에 이용된 컬러에 관한 정보를 획득하기 위하여 제2 컬러 팔레트를 이용하여 서브타이틀을 중첩시키도록 구성된다.

[0069] 본 발명의 실시예에 따라, 멀티미디어 정보를 통신하기 위한 시스템은, 네트워크; 멀티미디어 파일을 포함하며, 서버를 통해 네트워크에 접속되는 기억 장치; 및 네트워크에 접속되는 클라이언트를 포함하며, 클라이언트는 서버로부터 멀티미디어 파일의 전송을 요청할 수 있고, 멀티미디어 파일은 적어도 하나의 비디오 트랙과 비디오 트랙과 동반하는 적어도 하나의 서브타이틀 트랙을 포함한다.

[0070] 본 발명의 실시예에 따른 멀티미디어 파일은, 일련의 부호화된 비디오 프레임; 및 부호화된 메뉴 정보를 포함한다. 또한, 부호화된 메뉴 정보는 청크로서 저장된다.

[0071] 다른 실시예에서, 메뉴 정보의 적어도 2개의 개별적인 'menu'(메뉴) 청크를 추가로 구비한다. 또한, 적어도 2개의 개별적인 'menu' 청크는, 적어도 2개의 개별적인 'RIFF' 청크에 포함된다.

[0072] 다른 실시예에서, 'menu' 청크를 갖는 제1 'RIFF' 청크는 표준 4cc 코드를 포함하며, 'menu' 청크를 갖는 제2 'RIFF' 청크는 전용의 4cc 코드를 포함하는데, 표준 4cc 코드의 첫 번째 2개의 문자는 전용의 4cc 코드의 마지막 2개의 문자처럼 나타난다.

[0073] 또 다른 실시예에서, 적어도 2개의 개별적인 'menu' 청크는 단일의 'RIFF' 청크에 포함된다.

[0074] 다른 실시예에서, 'menu' 청크는 일련의 메뉴를 기술하는 청크와, 일련의 메뉴와 관련된 미디어를 포함하는 'MRIF' 청크를 포함한다. 또한, 'MRIF' 청크는 비디오 트랙, 오디오 트랙 및 오버레이 트랙을 구비하는 미디어 정보를 포함한다.

[0075] 또 다른 실시예에서, 일련의 메뉴를 기술하는 청크는, 전체 메뉴 시스템을 기술하는 청크; 메뉴를 언어로 그룹화하는 적어도 하나의 청크; 개별적인 메뉴 디스플레이와 이에 동반하는 배경 오디오를 기술하는 적어도 하나의 청크; 메뉴 상의 버튼을 기술하는 적어도 하나의 청크; 스크린 상의 버튼의 위치를 기술하는 적어도 하나의 청크; 및 버튼과 관련된 다양한 동작을 기술하는 적어도 하나의 청크를 포함한다.

[0076] 또 다른 실시예에서, 제2 파일에 대한 링크을 추가로 포함하며, 부호화된 메뉴 정보가 상기 제2 파일 내에 포함된다.

[0077] 본 발명의 실시예에 따라, 멀티미디어 파일을 부호화하기 위한 시스템은, 메뉴 정보를 부호화하도록 구성된 프로세서를 포함하며, 프로세서는 부호화된 비디오 트랙 및 부호화된 메뉴 정보를 갖는 멀티미디어 파일을 생성하도록 구성된다.

[0078] 다른 실시예에서, 프로세서는 메뉴의 객체 모델(object model)을 생성하며, 객체 모델을 구성 파일(configuration file)로 변환하고, 구성 파일을 청크로 파싱(parsing) 처리하며, 미디어 정보를 구비하는 AVI(Audio Video Interleave) 파일을 생성하고, AVI 파일 내의 미디어를 'MRIF' 청크로 인터리브하며, 파싱 처리된 청크를 'MRIF' 청크에 연결하여 'menu' 청크를 작성하도록 구성된다.

[0079] 다른 실시예에서, 프로세서는 객체 모델을 이용하여 제2의 더 작은 'menu' 청크를 생성하도록 구성된다. 또한, 프로세서는 제2 메뉴를 부호화하도록 구성된다.

[0080] 또 다른 실시예에서, 프로세서는 부호화된 제1 메뉴를 제1 'RIFF' 청크에 삽입하고, 부호화된 제2 메뉴를 제2 'RIFF' 청크에 삽입한다. 또한, 프로세서는 단일의 'RIFF' 청크에 부호화된 제1 메뉴 및 제2 메뉴를 포함한다. 또한, 프로세서는 제2 파일 내의 부호화된 메뉴에 대한 참조를 멀티미디어 파일에 삽입한다.

[0081] 본 발명의 실시예에 따라, 멀티미디어 파일을 복호화하기 위한 시스템은, 멀티미디어 파일로부터 정보를 추출하도록 구성된 프로세서를 포함하며, 프로세서는 부호화된 메뉴 정보가 포함되어 있는지 여부를 판정하기 위해 멀티미디어 파일을 조사하도록 구성된다. 또한, 프로세서는 'RIFF' 청크 내의 'menu' 청크로부터 메뉴 정보를 추출하도록 구성된다. 또한, 프로세서는 'menu' 청크 내에 저장된 비디오 정보를 이용하여 메뉴 디스플레이를 구성한다.

[0082] 다른 실시예에서, 프로세서는 'menu' 청크에 저장된 오디오 정보를 이용하여 메뉴 디스플레이와 동반하는 배경 오디오를 생성하도록 구성된다.

[0083] 다른 실시예에서, 프로세서는 'menu' 청크로부터의 비디오 정보의 위에 'menu' 청크를 오버레이 처리함으로써 메뉴 디스플레이를 생성하도록 구성된다.

[0084] 본 발명의 실시예에 따른 멀티미디어 정보를 통신하기 위한 시스템은, 네트워크; 멀티미디어 파일을 포함하며, 서버를 통해 네트워크에 접속되는 기억 장치; 및 네트워크에 접속되는 클라이언트를 포함하며, 클라이언트는 서버로부터 멀티미디어 파일의 전송을 요청할 수 있고, 멀티미디어 파일은 부호화된 메뉴 정보를 포함한다.

[0085] 멀티미디어 파일은 일련의 부호화된 비디오 프레임; 및 멀티미디어 파일에 관한 부호화된 메타데이터를 포함하며, 부호화된 메타데이터는 주어부, 술어부, 객체부 및 출처부를 구비하는 적어도 하나의 문을 포함하여 이루어진다.

[0086] 다른 실시예에서, 주어부는 메타데이터에 의해 기술되는 파일, 아이템, 개인 또는 조직을 식별하는 정보를 포함한다.

[0087] 또 다른 실시예에서, 술어부는 주어부의 특성을 나타내는 정보를 포함한다. 또한, 객체부는 술어부에 의해 식별된 주어부의 특성을 기술하는 정보를 포함한다. 출처부는 문의 소스에 관련된 정보를 포함한다.

[0088] 또 다른 실시예에서, 주어부는 타입과 값을 구비하는 청크에 해당하며, 값은 정보를 포함하고, 타입은 청크가 리소스인지 또는 익명 노드인지를 나타낸다.

[0089] 다른 실시예에서, 술어부는 타입 및 값을 구비하는 청크에 해당하며, 값은 정보를 포함하고, 타입은 값 정보가 서술형 URI(Universal Resource Identifier)인지 또는 정규의 리스트 엔트리인지 여부를 나타낸다.

[0090] 다른 실시예에서, 객체부는 타입, 언어, 데이터 타입 및 값을 구비하는 청크에 해당하며, 값은 정보를 포함하고, 타입은 값 정보가 EUC-KR(8 bit UCS Transformation Format) 문자인지, 문자 정수인지 또는 문자 XML 데이터인지 여부를 나타내며, 데이터 타입은 값 정보의 타입을 나타내고, 언어는 특정 언어를 식별하는 정보를 포함한다.

[0091] 또 다른 실시예에서, 출처부는 타입 및 값을 구비하는 청크에 해당하며, 값은 정보를 포함하고, 타입은 값 정보가 문의 출처라는 것을 나타낸다.

[0092] 다른 실시예에서, 부호화된 데이터의 적어도 일부는 이진 데이터로서 표현된다.

[0093] 또한, 부호화된 데이터의 적어도 일부는 64 비트의 ASCI 데이터로서 표현된다.

[0094] 다른 실시예에서, 부호화된 데이터의 적어도 제1 부분은 이진 데이터로서 표현되며, 부호화된 데이터의 적어도 제2 부분은 다른 형식으로 표현된 데이터를 포함하는 추가의 청크로서 표현된다.

[0095] 다른 실시예에서, 추가의 청크는 각각 단일 부분으로 된 메타데이터를 포함한다.

[0096] 본 발명의 실시예에 따라 멀티미디어 파일을 부호화하기 위한 시스템은, 비디오 트랙을 부호화하도록 구성된 프로세서를 포함하며, 프로세서는 멀티미디어 파일에 관련된 메타데이터를 부호화하도록 구성되어 있고, 부호화된 메타데이터는 주어부, 술어부, 객체부, 및 출처부를 구비하는 적어도 하나의 문을 포함하여 이루어진다. 또한, 주어부는 메타데이터에 의해 기술되는 파일, 아이템, 개인 또는 조직을 식별하는 정보를 포함한다. 또한, 술어부는 주어부의 특성을 나타내는 정보를 포함한다. 객체부는 술어부에 의해 식별된 주어부의 특성을 기술하는 정보를 포함한다. 출처부는 문의 소스에 관련된 정보를 포함한다.

[0097] 다른 실시예에서, 프로세서는 주어부를 타입 및 값을 포함하는 청크로서 부호화하도록 구성되어 있으며, 값은 정보를 포함하고, 타입은 청크가 리소스인지 또는 익명 노드인지 여부를 나타낸다.

[0098] 다른 실시예에서, 프로세서는, 타입 및 값을 구비하는 청크로서 술어부를 부호화하도록 구성되어 있으며, 값은 정보를 포함하고, 타입은 값 정보가 서술형 URI인지 또는 정규의 리스트 엔트리인 여부를 나타낸다.

[0099] 또 다른 실시예에서, 프로세서는 객체부를, 타입, 언어, 데이터 타입 및 값을 구비하는 청크에서 부호화하도록 구성되어 있으며, 값은 정보를 포함하고, 타입은 값 정보가 EUC-KR 문자인지, 문자 정수인지 또는 문자 XML 데이터인지 여부를 나타내며, 데이터 타입은 상기 값 정보의 타입을 나타내고, 언어는 특정 언어를 식별하는 정보를 포함한다.

[0100] 다른 실시예에서, 프로세서는 출처부를 타입 및 값을 포함하는 청크로서 부호화하도록 구성되어 있으며, 값은 정보를 포함하고, 타입은 값 정보가 문의 출처부라는 것을 나타낸다.

[0101] 다른 실시예에서, 프로세서는 멀티미디어 파일에 관련된 메타데이터의 적어도 일부를 이진 데이터로서 부호화하도록 구성된다.

[0102] 다른 실시예에서, 프로세서는 멀티미디어 파일에 관련된 메타데이터의 적어도 일부를 64 비트의 ASCI 데이터로서 부호화하도록 구성된다. 또한, 프로세서는 멀티미디어 파일에 관련된 메타데이터의 적어도 제1 부분을 이진 데이터로서 부호화하고, 멀티미디어 파일에 관련된 메타데이터의 적어도 제2 부분을 다른 형식으로 표현된 데이터를 포함하는 추가의 청크로서 부호화하도록 구성된다.

[0103] 다른 실시예에서, 프로세서는 추가의 청크를 단일 부분으로 이루어진 메타데이터로 부호화하도록 구성된다.

[0104] 본 발명의 실시예에 따라, 멀티미디어 파일을 복호화하기 위한 시스템은, 멀티미디어 파일로부터 정보를 추출하도록 구성된 프로세서를 포함하며, 프로세서는 멀티미디어 파일에 관련된 메타데이터 정보를 추출하도록 구성되어 있으며, 메타데이터 정보는 주어부, 술어부, 객체부 및 출처부를 구비하는 적어도 하나의 문을 포함한다.

[0105] 다른 실시예에서, 프로세서는 메타데이터에 의해 기술되는 파일, 아이템, 개인 또는 조직을 식별하는 정보를 주어부로부터 추출하도록 구성된다. 또한, 프로세서는 술어부로부터 주어부의 특성을 나타내는 정보를 추출하도록 구성된다. 또한, 프로세서는 술어부에 의해 식별된 주어부의 특성을 기술하는 정보를 객체부로부터 추출하도록 구성된다.

[0106] 다른 실시예에서, 프로세서는 문의 소스에 관련된 정보를 출처부로부터 추출하도록 구성된다.

[0107] 또 다른 실시예에서, 주어부는 타입 및 값을 포함하는 청크에 해당하며, 프로세서는 타입을 조사함으로써 청크가 주어부 정보를 포함하는 것을 식별하도록 구성되어 있고, 프로세서는 값으로부터 정보를 추출하도록 구성된다.

[0108] 다른 실시예에서, 술어부는 타입 및 값을 포함하는 청크에 해당하며, 프로세서는 타입을 조사함으로써 청크가 술어부 정보를 포함하는 것을 식별하도록 구성되어 있고, 프로세서는 값으로부터 정보를 추출하도록 구성된다.

[0109] 다른 실시예에서, 객체부는 타입, 언어, 데이터 타입 및 값을 포함하는 청크에 해당하며, 프로세서는 타입을 조사함으로써 청크가 객체 정보를 포함하는 것을 식별하도록 구성되어 있고, 프로세서는 값에 포함된 정보의 데이

터 타입을 판정하기 위해 데이터 타입을 조사하도록 구성되어 있으며, 프로세서는 데이터 타입에 의해 표시되는 타입의 정보를 값으로부터 추출하도록 구성되어 있고, 프로세서는 특정의 언어를 식별하는 정보를 언어로부터 추출하도록 구성되어 있다.

[0110] 또 다른 실시예에서, 출처부는 타입 및 값을 포함하는 청크에 해당하며, 프로세서는 타입을 조사함으로써 청크가 출처부 정보를 포함하는 것을 식별하도록 구성되어 있고, 프로세서는 값으로부터 정보를 추출하도록 구성되어 있다.

[0111] 다른 실시예에서, 프로세서는 메타데이터 문으로부터 정보를 추출하고, 정보의 적어도 일부를 표시하도록 구성된다.

[0112] 다른 실시예에서, 프로세서는 메타데이터를 이용하여 메모리 내에 방향성 라벨부착 그래프(DLG: Directed-Labeled Graphs)를 나타내는 데이터 구조를 구성한다.

[0113] 다른 실시예에서, 프로세서는, 복수 개의 문에 대하여 주어부, 술어부, 객체부 및 출처부 중 적어도 하나를 조사함으로써, 메타데이터를 통해 정보를 검색하도록 구성된다.

[0114] 다른 실시예에서, 프로세서는 검색의 결과를 그래픽 사용자 인터페이스의 일부로서 표시하도록 구성된다. 또한, 프로세서는 외부 장치로부터의 요청에 따라 검색을 수행하도록 구성된다.

[0115] 또 다른 실시예에서, 멀티미디어 파일과 관련된 메타데이터 정보의 적어도 일부는 이진 데이터로서 표현된다.

[0116] 다른 실시예에서, 멀티미디어 파일과 관련된 메타데이터 정보의 적어도 일부는 64 비트의 ASCI 데이터로서 표현된다.

[0117] 또 다른 실시예에서, 멀티미디어 파일에 관련된 메타데이터 정보의 적어도 제1 부분은 이진 데이터로서 표현되며, 멀티미디어 파일에 관련된 메타데이터 정보의 적어도 제2 부분은 다른 형식으로 표현된 데이터를 포함하는 추가의 청크로서 표현된다. 또한, 추가의 청크는 단일 부분으로 된 메타데이터를 포함한다.

[0118] 본 발명의 실시예에 따라, 멀티미디어 정보를 통신하기 위한 시스템은, 네트워크; 멀티미디어 파일을 포함하며, 서버를 통해 네트워크에 접속되는 기억 장치; 및 네트워크에 접속되는 클라이언트를 포함하며, 클라이언트는 서버로부터 멀티미디어 파일의 전송을 요청할 수 있고, 멀티미디어 파일은 멀티미디어 파일에 관련된 메타데이터를 포함하며, 메타데이터는 주어부, 술어부, 객체부 및 출처부를 구비하는 적어도 하나의 문을 포함한다.

[0119] 본 발명의 실시예에 따른 멀티미디어 파일은, 적어도 하나의 부호화된 비디오 트랙; 적어도 하나의 부호화된 오디오 트랙; 복수 개의 부호화된 텍스트 문자열을 포함하며, 부호화된 텍스트 문자열이 적어도 하나의 비디오 트랙과 적어도 하나의 오디오 트랙의 특성을 기술한다.

[0120] 다른 실시예에서, 복수 개의 텍스트 문자열은, 비디오 트랙 또는 오디오 트랙의 동일한 특성을, 상이한 언어를 이용하여 기술한다.

[0121] 다른 실시예에서, 적어도 하나의 부호화된 서브타이틀 트랙을 추가로 포함하며, 복수 개의 부호화된 텍스트 문자열은 서브타이틀 트랙의 특성을 기술하는 문자열을 포함한다.

[0122] 본 발명의 실시예에 따라, 멀티미디어 파일을 작성하기 위한 시스템은, 적어도 하나의 비디오 트랙을 부호화하고, 적어도 하나의 오디오 트랙을 부호화하며, 부호화된 오디오 트랙의 적어도 하나를 상기 비디오 트랙과 인터리브하고, 적어도 하나의 비디오 트랙과 적어도 하나의 오디오 트랙의 다수의 특성의 각각을 복수 개의 언어로 기술하는 텍스트 문자열을 삽입하도록 구성된 프로세서를 포함한다.

[0123] 본 발명의 실시예에 따라, 부호화된 오디오, 비디오 및 텍스트 문자열을 포함하는 멀티미디어 파일을 표시하기 위한 시스템은, 부호화된 텍스트 문자열을 멀티미디어 파일로부터 추출하며, 텍스트 문자열을 이용하여 풀 다운(pull down) 메뉴 디스플레이를 생성하도록 구성된 프로세서를 포함한다.

발명의 효과

[0124] 본 발명으로 멀티미디어 파일과, 이러한 멀티미디어 파일을 생성, 분배 및 복호화하기 위한 시스템을 제공할 수 있다.

도면의 간단한 설명

[0125] 도 1은 부호화, 분배 및 복호화하기 위한 본 발명의 실시예에 따른 시스템을 나타내는 도면.

도 2a는 본 발명의 실시에에 따른 멀티미디어 파일의 구조를 나타내는 도면.

도 2aa는 본 발명의 다른 실시예에 따른 멀티미디어 파일의 구조를 나타내는 도면.

도 2b는 본 발명의 실시예에 따른 'hdrl' 리스트 청크의 개념도.

도 2c는 본 발명의 실시예에 따른 'strl' 청크의 개념도.

도 2d는 본 발명의 실시예에 따른 멀티미디어 파일의 'DXDT' 청크를 저장하기 위해 할당된 메모리의 개념도.

도 2da는 본 발명의 실시예에 따른 멀티미디어 파일의 'DXDT' 청크에 포함될 수 있는 '메타데이터'의 개념도.

도 2e는 본 발명의 실시예에 따른 "DMNU' 청크의 개념도.

도 2f는 본 발명의 실시예에 따른 'WowMenuManager' 청크 내에 포함되는 메뉴 청크의 개념도.

도 2g는 본 발명의 실시예에 따른 'WowMenuManager' 청크 내에 포함되는 메뉴 청크의 개념도.

도 2ga는 'DMNU' 청크 내에 포함되는 다양한 청크 사이의 관계를 나타내는 개념도.

도 2h는 본 발명의 실시예에 따른 멀티미디어 파일의 'movi' 리스트 청크의 개념도.

도 2i는 DRM을 포함하는 본 발명의 실시예에 따른 멀티미디어 파일의 'movi' 리스트 청크의 개념도.

도 2j는 본 발명의 실시예에 따른 'DRM' 청크의 개념도.

도 3a는 본 발명의 실시예에 따른 멀티미디어 파을 생성하기 위한 시스템의 블록도.

도 3b는 본 발명의 실시예에 따른 'DXDT' 청크를 생성하기 위한 시스템의 블록도.

도 3c는 본 발명의 실시예에 따른 'DMNU' 청크를 생성하기 위한 시스템의 블록도.

도 3d는 본 발명의 실시예에 따른 미디어 모델의 개념도.

도 3da는 본 발명의 실시예에 따라 소형 메뉴를 자동으로 생성하는데 이용될 수 있는 미디어 모델로부터의 객체(object)의 개념도.

도 3e는 본 발명의 실시예에 따라 오디오를 리청크(re-chunk)하는데 이용될 수 있는 처리과정을 나타내는 흐름도.

도 3f는 본 발명의 실시예에 따른 비디오 인코더의 블록도.

도 3g는 본 발명의 실시예에 따라 I 프레임 상에서의 평활한 사이코비주얼 인핸스먼트를 수행하는 방법을 나타내는 흐름도.

도 3h는 본 발명의 실시예에 따라 매크로블록 SAD 사이코비주얼 인핸스먼트를 수행하기 위한 처리과정을 나타내는 흐름도.

도 3i는 본 발명의 실시예에 따라 단일 패스 레이트 제어를 위한 처리과정을 나타내는 흐름도.

도 3j는 본 발명의 실시예에 따라 N번째 패스 VBV 레이트 제어를 수행하기 위한 처리과정을 나타내는 흐름도.

도 4a는 본 발명의 실시예에 따라 멀티미디어 파일로부터 요청된 멀티미디어 정보를 위치시키고 멀티미디어 정보를 표시하기 위한 처리과정을 나타내는 흐름도.

도 4b는 본 발명의 실시예에 따른 디코더의 블록도.

도 4c는 본 발명의 실시예에 따라 표시한 메뉴의 예를 나타내는 도면.

도 4d는 본 발명의 실시예에 따라 도 4c에 나타낸 디스플레이를 생성하는데 이용되는 정보 소스를 나타내는 개념도.

발명을 실시하기 위한 구체적인 내용

[0126] 도면을 참조하여, 멀티미디어 파일을 복호화, 전송 및 복호화할 수 있는 본 발명의 실시예에 대하여 설명한다. 본 발명의 실시예에 따른 멀티미디어 파일은, 다중의 비디오 트랙, 다중의 오디오 트랙, 다중의 서브타이틀 트

랙, 파일의 컨텐츠에 접근하기 위한 메뉴 인터페이스를 생성하는데 이용될 수 있는 데이터, 및 파일의 컨텐츠에 관련된 "메타데이터"(meta data)를 포함할 수 있다. 본 발명의 몇몇 실시예에 따른 멀티미디어 파일은, 이 파일과 관련되지 않은 비디오 트랙, 오디오 트랙, 서브타이틀 트랙 및 "메타데이터"에 대한 기준을 포함한다.

[0127] 1. 시스템의 설명

[0128] 도 1에는, 본 발명의 실시예에 따라 파일을 부호화, 분배 및 복호화하기 위한 시스템을 나타낸다. 이 시스템(10)은 네트워크(14)를 통해 다른 다양한 연산 장치에 접속되는 컴퓨터(12)를 포함한다. 네트워크에 접속될 수 있는 장치(이하, 단순히 '장치'라고 함)에는, 서버(16), 랩탑 컴퓨터(18) 및 개인 휴대정보 단말기(PDA: personal digital assistant)(20)가 포함된다. 다양한 실시예에서, 장치와 네트워크 사이의 접속은 유선 또는 무선으로 가능하며, 다양한 네트워크 프로토콜을 사용하여 구현될 수 있다.

[0129] 동작에 있어서, 컴퓨터(12)는 본 발명의 실시예에 따라 멀티미디어 파일을 부호화하는데 이용될 수 있다. 컴퓨터(12)는, 또한 본 발명의 실시예에 따라 멀티미디어 파일을 복호화 및 분배하는데에도 이용될 수 있다. 이러한 컴퓨터는 피투피(P2P: peer-to-peer) 네트워크를 포함하는 다양한 파일 전송 프로토콜을 이용하여 파일을 분배할 수 있다. 또한, 컴퓨터(12)는 본 발명의 실시예에 따라 멀티미디어 파일을 서버(18)에 전송할 수 있으며, 서버(18)는 다른 장치가 파일에 접근할 수 있게 되어 있다. 다른 장치로는, 컴퓨팅 장치나 전용의 디코더 장치도 포함할 수 있다. 본 예에서는, 랩탑 컴퓨터와 PDA가 도시되어 있다. 다른 실시예에서는, 디지털 셋톱박스, 데스크톱 컴퓨터, 게임 머신, 가전 제품 및 그와 다른 장치가, 네트워크에 접속되어 멀티미디어 파일을 다운로드해서 복호화할 수 있다.

[0130] 일실시예에서, 장치는 네트워크를 통해 서버로부터 멀티미디어 파일을 액세스한다. 다른 실시예에서, 장치는 피투피 네트워크를 통해 다수의 컴퓨터로부터 멀티미디어 파일을 액세스한다. 몇몇 실시예에서, 멀티미디어 파일은 디스크 드라이브, CD-ROM, DVD 등의 휴대형 기억 장치에 기록될 수 있다. 많은 실시예에서, 전자 장치는 휴대형 기억 장치에 기록된 멀티미디어 파일을 액세스할 수 있다.

[0131] 2. 파일 구조의 설명

[0132] 본 발명의 실시예에 따른 멀티미디어 파일은, 워싱턴주 레드몬드에 소재한 마이크로소프트(Microsoft Corporation) 및 뉴욕주 아몬크에 소재한 아이비엠(IBM)에서 정한 자원교환 파일형식(RIFF: Resource Interchange File Format)과 호환 가능하도록 구성될 수 있다. RIFF는 멀티미디어 데이터 및 이와 관련된 정보를 기억하기 위한 파일 형식이다. RIFF 파일은 통상적으로 파일을 식별하여 후속하는 파일의 남은 길이(즉, file_length - 8)를 제공하는 8 바이트의 RIFF 헤더를 가진다. RIFF 파일의 잔여부분 전체는 "청크"(chunk)와 "리스트"(list)를 포함하여 이루어진다. 각각의 청크는 그 타입을 식별하며 그 다음에 이어지는 데이터 바이트 길이를 제공하는 8 바이트의 청크 헤더를 포함한다. 각각의 리스트는 리스트의 타입을 식별하며 후속하는 데이터의 길이(바이트 단위)를 나타내는 8 바이트의 리스트 헤더를 갖는다. 리스트에서의 데이터는 청크 및/또는 리스트를 포함한다(결과적으로 청크 및/또는 다른 리스트를 포함할 수 있다). RIFF 리스트를, "리스트 청크"(list chunks)라고도 한다.

[0133] AVI(Audio Video Interleave) 파일은, RIFF 파일의 형식에 따르기는 하지만 멀티미디어 데이터를 특정의 형식으로 갖는 식별자가 정의된, 다양한 청크와 리스트를 포함하는 특별한 형식의 RIFF 파일이다. AVI 포맷은 마이크로소프트(Microsoft Corporation)에서 개발 및 정의된 것이다. AVI 파일은 멀티미디어 데이터를 AVI 형식으로 출력할 수 있는 인코더를 이용하여 생성되는 것이 일반적이다. AVI 파일은, 총괄적으로 AVI 디코더로서 알려져 있는 소프트웨어 그룹 중 임의의 것에 의해 복호화되는 것이 일반적이다.

[0134] RFII 및 AVI 포맷은, 이들이 소정의 파일 포맷의 일부에 해당하는 청크 및 리스트만을 정의하지만, 파일이 RIFF 및/또는 AVI 디코더에 의해 판독할 수 없도록 하지 않으면서, RIFF 및/또는 AVI 파일 포맷 정의와 관계없는 리스트 및/또는 청크를 포함한다는 점에서 유연성을 갖는다. 실제로, AVI(마찬가지로 RIFF) 디코더는, 이들이 AVI 파일 포맷 정의에서는 발견하지 못한 헤더 정보를 포함하는 리스트와 청크를 그냥 무시하도록 하여 구현된다. AVI 디코더는 비(非)-AVI 청크와 리스트를 통해 계속해서 판독할 수 있어야 하며, AVI 디코더의 동작이 완만해져야 하지만, 그렇지 않다면, 이들은 아무 영향도 미치지 않으면 AVI 디코더 의해 무시된다.

[0135] 본 발명의 실시예에 따른 멀티미디어 파일을 도 2a에 도시한다. 예시된 멀티미디어 파일(30)은 문자 세트 청크('CSET' chunk)(32), 정보 리스트 청크('INFO' 리스트 chunk))(34), 파일 헤더 청크('hdrl' 리스트 chunk)(36), 메타데이터 청크('DXDT' chunk)(38), 메뉴 청크('DMNU" chunk)(40), 정크 청크('junk'

chunk)(41), 영화 리스트 청크('movi' list chunk)(42), 선택적 인덱스 청크('idxl' chunk)(44), 및 부차적 메뉴 청크('DMNU' chunk)(46)를 포함한다. 이들 청크 중 일부와 그외 다른 부분은 AVI 파일 포맷에서 정의하고 있으나, AVI 파일 포맷에 포함되지 않는 부분도 있다. 이하, 모두는 아니지만 많은 경우에 있어서, AVI 파일 포맷의 일부로서 정의하고 있는 청크 또는 청크의 일부에 대하여 설명한다.

[0136] 본 발명의 실시예에 따른 다른 멀티미디어 파일을 도 2aa에 나타낸다. 멀티미디어 파일(30')은 도 2a에 나타낸 멀티미디어 파일과 유사하지만, 파일에 다중의 연결된 'RIFF' 청크가 포함되어 있다는 점이 다르다. 'RIFF' 청크는 부차적 'DMNU' 청크(46)를 배제할 수 있으며, 메뉴 정보를 'DMNU' 청크(46')의 형태로 포함할 수 있는, 도 2a에 도시한 것과 유사한, 'RIFF' 청크를 포함할 수 있다.

[0137] 도시한 예에서, 멀티미디어는 다중의 연결된 'RIFF' 청크를 포함하며, 제1 'RIFF' 청크(50)는, 문자 세트 청크('CSET' chunk)(32'), 정보 리스트 청크('INFO' 리스트 chunk)(34'), 파일 헤더 청크('hdrl' 리스트 chunk)(36), 메타데이터 청크('DXDT' chunk)(38), 메뉴 청크('DMNU" chunk)(40), 정크 청크('junk' chunk)(41), 영화 리스트 청크('movi' list chunk)(42), 선택적 인덱스 청크('idxl' chunk)(44)를 포함한다. 제2 'RIFF' 청크(52)는, 다른 메뉴 청크('DMNU' chunk)(46')를 포함한다. 'RIFF' 메뉴 청크(52) 이후에, 추가의 타이틀을 포함하는 추가의 'RIFF' 청크(54)가 포함될 수 있다. 추가의 'RIFF' 청크는 독립적인 미디어를 호환 가능한 AVI 파일 형식으로 포함할 수 있다. 일실시예에서, 제2 메뉴 청크(46')와 추가의 'RIFF' 청크는 4개의 문자 코드(AVI 포맷에서 정의되어 있음, 이하 설명함)로 특정된다. 이 4개의 문자 코드 중 처음 2개의 문자는 2개의 제2 문자를 나타내며, 다음 2개의 문자는 2개의 제1 문자를 나타낸다.

[0138] 2.1 'CSET' 청크

[0139] 'CSET' 청크(32)는 마이크로소프트에서 만든 AVI(Audio Video Interleave: 오디오 비디오 겸용)의 파일 포맷(AVI file format)에서 정의한 청크이다. 'CSET' 청크는 멀티미디어 파일의 문자 세트와 언어 정보를 정의한다. 'CSET' 청크를 포함할지 여부는 본 발명의 실시예에 따라 선택적이다.

[0140] 본 발명의 실시예에 따른 멀티미디어 파일은, 'CSET' 청크를 이용하지 않고, RFC 3066 Language Specification(언어 사양)과 조합한 디폴트에 의해 설정된 문자에 대해서는 유니코드 컨소니엄(Unicode Consortium)에서 정의하고, 언어 정보에 대해서는 IETF(Internet Engineering Task Force)에서 정의한 EUC-KR을 이용한다.

[0141] 2.2 'INFO' 리스트 청크

[0142] 'INFO' 리스트 청크(34)는 멀티미디어 파일의 컨텐츠를 식별하는데 도움을 주는 정보를 저장할 수 있다. 'INFO' 리스트는, AVI 파일 형식으로 정의되며, 멀티미디어 파일에 포함할지 여부는 본 발명의 실시예에 따라 선택적이다. 'DXDT' 청크를 포함하는 많은 실시예에서는 'INFO' 리스트 청크를 포함하지 않는다.

[0143] 2.3 'hdrl' 리스트 청크

[0144] 'hdrl' 리스트 청크(38)는 AVI 파일 형식으로 정의되며, 멀티미디어 파일 내의 데이터의 형식 관련 정보를 제공한다. 'hdrl' 리스트 청크 또는 유사한 설명 정보를 갖는 청크는 일반적으로 포함되어야 한다. 'hdrl' 리스트 청크는 각각의 비디오 트랙, 각각의 오디오 트랙 및 각각의 서브타이틀 트랙에 대한 청크를 포함한다.

[0145] 단일의 비디오 트랙(62), 2개의 오디오 트랙(64), 외부 오디오 트랙(66), 2개의 서브타이틀 트랙(68) 및 외부 서브타이틀 트랙(70)을 포함하는 본 발명의 일실시예에 따른 'hdrl' 리스트 청크(38)의 개념도를 도 2b에 나타낸다. 'hdrl' 리스트(60)는 'avih' 청크를 포함한다. 'avih' 청크(60)는 파일 내의 스트림의 수, 멀티미디어 파일 내에 포함되는 비디오의 폭과 높이와 같은 전체 파일에 대한 글로벌(전역) 정보를 포함한다. 'avih' 청크는 AVI 파일 형식에 따라 구현될 수 있다.

[0146] 'avih' 청크에 추가로, 'hdrl' 리스트는 각각의 오디오, 비디오 및 서브타이틀 트랙에 대한 스트림 기술어 리스트(stream descriptor list)를 포함한다. 일실시예에서, 스트림 기술어 리스트는 'strl' 청크를 이용하여 구현된다. 본 발명의 일실시예에 따른 'strl' 청크는 도 2c에 도시되어 있다. 각각의 'strl' 청크는 멀티미디어 파일 내의 각각의 트랙을 기술하는 기능을 한다. 멀티미디어 파일 내의 오디오, 비디오 및 서브타이틀 트랙에 대한 'strl' 청크는, 'strh' 청크(92), 'strf' 청크(94), 'strd' 청크(96) 및 'strn' 청크(98)를 참조하는 'strl' 청크를 포함한다. 이들 모든 청크는 AVI 파일 형식에 따라 구현될 수 있다. 특별히 중점을 둘 것은, 미디어 트랙의 타입을 특정하는 'strh' 청크(92)와, 비디오가 디지털 권리 관리에 의해 보호되는지 여부를 나타내기 위해 변형 가능한 'strd' 청크(96)이다. 본 발명의 실시예에 따라 디지털 권리 관리를 다양하게 구현하는

것에 대하여 이하 설명한다.

[0147] 본 발명의 실시예에 따른 멀티미디어 파일은 추가의 오디오 트랙 또는 추가의 서브타이틀 트랙과 같은 멀티미디어 정보를 유지하는 외부 파일에 대한 참조를 포함할 수 있다. 이들 트랙에 대한 참조(references)는 'hdrl' 청크 또는 'junk' 청크(41)에 포함될 수 있다. 어느 경우에서나, 로컬 파일이나 원격에 저장된 파일을 참조하는, 'strl' 청크(90)의 'strh' 청크(92)에 대한 참조가 포함될 수 있다. 참조된 파일은 추가의 트랙을 포함하는 본 발명의 실시예에 따라 멀티미디어 파일 또는 표준 AVI 파일이 될 수 있다.

[0148] 다른 실시예에서, 참조된 파일은 ''DMNU' 청크, 'DXDT' 청크 및 멀티미디어 표현(multimedia presentation)에 대한 오디오, 비디오 및/또는 서브타이틀 트랙과 관련된 청크를 포함하는 참조용 파일에 존재할 수 있는 임의의 청크를 포함할 수 있다. 예를 들어, 제1 멀티미디어 파일은, 제1 멀티미디어 파일의 'movi' 리스트 청크 내에 위치한 제1 멀티미디어 표현과, 제2 멀티미디어 파일의 'movi' 리스트 청크 내에 위치한 제2 멀티미디어 표현을 참조하는 'DMNU' 청크(이하 상세하게 설명함)를 포함할 수 있다. 이와 달리, 이들 2개의 'movi' 리스트 청크는 'DMNU' 청크가 위치해 있는 파일과 동일한 파일일 필요가 없는 동일한 멀티미디어 파일 내에 포함될 수 있다.

[0149] 2.4 'DXDT' 청크

[0150] 'DXDT' 청크(38)는, 소위 "메타데이터"라고도 불린다. "메타데이터"는 파일, 문서 또는 방송의 컨텐츠에 관련된 정보를 제공하는 데이터를 기술하기 위해 이용되는 용어이다. 본 발명의 일실시예에 따라 멀티미디어 파일의 'DXDT' 청크 내에 저장된 메타데이터는, 제목, 저자, 저작권 소유자 및 배역(cast)과 같은 컨텐츠 특정 정보를 저장하는데 이용될 수 있다. 또한, 각각의 과정 이후 이용되는 CLI 옵션 및 등화기 분배와 같은, 멀티미디어 파일을 부호화하는데 이용되는 코덱(codec)에 대한 기술적 설명이 제공될 수 있다.

[0151] 일실시예에서, 메타데이터는 주어부(subject), 술어부(predicate), 객체부(object) 및 출처부(authority)를 포함하는 각각의 문(statement)으로 이루어진 일련의 문으로서 'DXDT' 청크 내에 표현된다. 주어부는 기술되는 대상을 나타낸다. 주어부는 파일, 아이템, 개인 또는 조직을 나타낸다. 주어부는 서술 가능한 특징을 갖는 것이면 어느 것이나 가능하다. 술어부는 서술의 대상이 되는 주어부의 특징을 식별한다. 객체부는 주어부의 식별된 특징에 대한 설명이며, 출처부는 정보의 소스를 식별한다.

[0152] 이하는, 다양한 "메타데이터"가 객체부, 술어부, 주어부 및 출처부로 어떻게 표현될 수 있는지를 나타내는 표이다.

표 1

[0153] "메타데이터"의 개념적 표현

주어부	술어부	객체부	출처부
_:file281	http://purl.org/dc/elements/1.1/title	The Matrix	_:auth42
_:file281	http://xmlns.divxnetworks.com/2004/11/cast#Person	_:cast871	_:auth42
_:file281	http://xmlns.divxnetworks.com/2004/11/cast#Person	_:cast872	_:auth42
_:file281	http://xmlns.divxnetworks.com/2004/11/cast#Person	_:cast873	_:auth42
_:cast871	http://xmlns.divxnetworks.com/2004/11cast#name	Keanu Reeves	_:auth42
_:cast871	http://xmlns.divxnetworks.com/2004/11/cast#role	Actor	_:auth42
_:cast871	http://xmlns.divxnetworks.com/2004/11/cast#character	Neo	_:auth42

_:cast282	http://xmlns.divxnetworks.com/2004/11/cast#name	Andy Wachowski (앤디 위쇼스키)	_:auth42
_:cast282	http://xmlns.divxnetworks.com/2004/11/cast#role	Director(감독)	_:auth42
_:cast283	http://xmlns.divxnetworks.com/2004/11/cast#name	Larry Wachowski(래리 위쇼스키)	_:auth42
_:cast283	http://xmlns.divxnetworks.com/2004/11/cast#role	Director (감독_	_:auth42
_:file281	http://purl.org/dc/elements/1.1/rights	Copyright 1998 Warner Brothers. All Rights Reserved.	_:auth42
_:file281	Series	_:file321	_:auth42
_:file321	Episode	2	_:auth42
_:file321	http://purl.org/dc/elements/1.1/title	The Matrix Reloaded	_:auth42
_:file321	Series	_:file122	_:auth42
_:file22	Episde	3	_:auth42
_:auth42	http://xmlns.com/foaf/0.1/Organization	_:foaf92	_:auth42
_:foaf92	http://xmlns.com/foaf/0.1/name	Warner Brothers	_:auth42
_:file281	http://xmlns.divxnetworks.com/2004/11/track#track	_:track#dc00	_:auth42
_:track#dc00	http://xmlns.divxnetworks.com/2004/11/track#resolution	1024x768	_:auth42
_:file281	http://xmlns.divxnetworks.com/2004/11/content#certificationLevel	HT	_:auth42
_:track#dc0	http://xmlns.divxnetworks.com/2004/11/track#frameTypeDist	32, 1, 3, 5	_:auth42
_:track#dc00	http://xmlns.divxnetworks.com/2004/11/track#codecSettings	bv1 276 -psy 0 -key 300 -b 1 -sc 50 -pq 5 -vbv 6951200, 3145728, 2359296 - profile 3 -nf	_:auth42

[0154] 일실시예에서, 주어부, 술어부, 객체부 및 출처부의 표현은, 방향성 라벨부착 그래프(DLG: Directed-Labeled Graphs)를 형성하는 것으로 고려될 수 있는 데이터의 이진 표현을 이용하여 구현된다. DLG는 리소스(resource) 또는 리터럴(literal)에 해당하는 노드로 구성된다. 리소스는 인터넷 엔지니어링 태스크포스(Internet Engineering Taskforce)(http://www.ietf.org/rfc/rfc2396.txt)에 의해 RFC 2396에서 정의된 바와 같은 "URI"(Universal Resource Identifier) 등의 이름 부여 방식(naming convention)에 부합되거나, 시스템 전용의 데이터를 의미하는 식별자이다. 리터럴은 참조용이 아니라 실제 값을 나타낸다.

[0155] DLG의 장점은, 영화의 배역과 같이, 동일한 타입의 데이터로서 탄력적인 개수의 데이터 아이템을 포함할 수 있다는 것이다. 표 1에 나타낸 예에서는 3개의 배역이 포함되어 있다. 그러나, 임의의 수의 배역이 포함될 수 있다. DLG에 의하면 다른 데이터 타입에도 상관적인 연결 관계를 가질 수 있다. 표1 에는, 주어부 "_:file281", 술어부 "Series", 객체부 "_:file321"을 갖는 "메타데이터" 아이템이 있다. 주제 "_:file321"은 "메타데이터"가 "_:file321"로서 참조된 파일의 내용(이 경우에는 영화 "The Matrix")을 참조한다는 것을 나타낸다. 술어부는, 객체부가 "The Matrix"가 속하는 시리즈에서 다른 영화와 관련된 정보를 가진다는 것을 의미한다. 그러나, "_:file321"은 "The Matrix"를 포함하는 시리즈에 관한 임의의 다른 특정한 정보가 아니라, "_:file321"에 관한 더 많은 정보를 제공하는 다른 엔트리에 대한 참조이다. 주어부 "_:file321"을 갖는 다음 "메타데이터"엔트리는, "_:file321"에 대한 데이터, 즉 이 결과의 "http://purl.org/dc/elements/1.1/ title"

로 표시된 것과 같이 듀블린 코어 어휘(Dublin Core Vocabulary)에서 특정하는 타이틀이 "The Matrix Reloaded"라는 것을 포함한다.

[0156] 표 1에서 추가의 '메타데이터' 문은 키아누 리브스(Keanu Reeves)가 네오(Neo)의 역할을 수행하는 배역이었으며, 래리 위쇼스키(Larry Wachowski)와 앤디 위쇼스키(Andy Wachowski)가 감독이었다는 것을 특정한다. 기술적 정보도 또한 '메타데이터'로 표현된다. 이 '메타데이터' 문은 "_:file281"이 트랙 "_:track#dc00"을 포함한다는 것을 식별한다. '메타데이터'는 비디오 트랙의 해상도, 비디오 트랙 및 코덱 설정의 인증 수준을 포함하는 정보를 제공한다. 표 1에는 나타내지 않았지만, '메타데이터'는 부호화할 때 트랙에 할당되는 고유 식별자를 포함할 수도 있다. 고유 식별자가 사용되는 경우, 동일한 컨텐츠를 여러 번 부호화하면, 그 컨텐츠의 부호화된 각각의 버전에 대해 상이한 식별자가 사용될 것이다. 그러나, 부호화된 비디오 트랙의 복제는 복제된 때부터 트랙의 식별자를 유지할 것이다.

[0157] 표 1에 나타낸 엔트리는, UPnP 포럼(http://www.upnpforum.org 참조)에서 정한, UPnp 어휘 등의 다른 어휘로 대체시킬 수 있다. 다른 대안은, MPEG-21 표준에 대한 작업의 일부로서 국제 표준화 기구(ISO)에서 개발한 Digital Item Declaration Language(DIDL) 또는 DIDL-Lite 어휘가 될 수 있다. 이하, UPnP 어휘 내에서의 술어부의 예이다.

[0158] urn:schemas-upnp-org:metadata-1-0/upnp/artist

[0159] urn:schemas-upnp-org:metadata-1-0/upnp/actor

[0160] urn:schemas-upnp-org:metadata-1-0/upnp/author

[0161] urn:schemas-upnp-org:metadata-1-0/upnp/producer

[0162] urn:schemas-upnp-org:metadata-1-0/upnp/director

[0163] urn:schemas-upnp-org:metadata-1-0/upnp/genre

[0164] urn:schemas-upnp-org:metadata-1-0/upnp/album

[0165] urn:schemas-upnp-org:metadata-1-0/upnp/playlist

[0166] urn:schemas-upnp-org:metadata-1-0/upnp/originalTrackNumber

[0167] urn:schemas-upnp-org:metadata-1-0/upnp/userAnnotation

[0168] 모든 '메타데이터'에 대한 출처부는 '_:auth42'이다. '메타데이터' 문은 '_:auth42'가 'Warner Brothers'이다. 출처부에 의하여 파일의 품질과 이 파일에 관련된 '메타데이터' 문의 품질을 모두 평가할 수 있다.

[0169] 그래프 내부의 노드는 명명된 리소스 노드를 통해 접속된다. '메타데이터'의 문은 주어부 노드, 서술부 노드 및 객체부 노드로 구성된다. 선택적으로, 출처부 노드는 '메타데이터' 문의 일부로서 DLG에 접속될 수 있다.

[0170] 각각의 노드에 대하여, 노드의 기능을 설명하는데 도움이 되는 소정의 특징이 있다. ANSI C 프로그래밍 언어를 이용하여 가능한 타입을 다음과 같이 나타낸다.

[0171] /**Invalid Type*/

[0172] #define RDF_IDENTIFIER_TYPE_UNKNOWN 0X00

[0173] /**Resource URI rdf:about*/

[0174] #define RDF_IDENTIFIER_TYPE_UNKNOWN 0X01

[0175] /**rdf:NodeId, _:file or generated N-Triples*/

[0176] #define RDF_IDENTIFIER_TYPE_UNKNOWN 0X02

[0177] /**Predicate URI*/

[0178] #define RDF_IDENTIFIER_TYPE_UNKNOWN 0X03

[0179] /**rdf:li, rdf:_<n>*/

[0180] #define RDF_IDENTIFIER_TYPE_UNKNOWN 0X04

[0181] /**Authority URI*/

[0182] #define RDF_IDENTIFIER_TYPE_UNKNOWN 0X05

[0183] /**EUC-KR formatted literal*/

[0184] #define RDF_IDENTIFIER_TYPE_UNKNOWN 0X06

[0185] /**Literal Integer*/

[0186] #define RDF_IDENTIFIER_TYPE_UNKNOWN 0X07

[0187] /**Literal XML data*/

[0188] #define RDF_IDENTIFIER_TYPE_UNKNOWN 0X08

[0189] 'DXDT' 청크 내에 포함되는 '메타데이터' 청크를 나타내는 데이터 구조(ANSI C 프로그래밍 언어로 표현됨)의 예는 다음과 같다.

[0190] typedef struct RDFDataStruct

[0191] {

[0192] RDFHeader Header;

[0193] uint32_t numOfStatements;

[0194] RDFStatement statements[RDF_MAX_STATEMENTS];

[0195] } RDFData

[0196] 'RDFData' 청크는 'RDFHeader' 청크, 'numOfStatements' 값 및 'RDFStatement' 청크의 리스트로서 불리는 청크를 포함한다.

[0197] 'RDFHeader' 청크는 '메타데이터'가 청크로 포맷화되는 방식에 대한 정보를 포함한다. 일실시예에서, 'RDFHeader' 청크 내의 데이터는 다음과 같이 나타낼 수 있다(ANSI C 프로그래밍 언어로 표현).

[0198] typedef struct RDFHeaderStruct

[0199] {

[0200] unit16_t versionMajor;

[0201] unit16_t versionMinor;

[0202] unit16_t versionFix;

[0203] unit16_t numOfSchemas;

[0204] RDFSchema schemas[RDF_MAX_SCHEMA];

[0205] } RDFHeader;

[0206] 'RDFHeader' 청크는 포워드 호환성을 가능하게 하도록 리소스 서술 포맷의 버전을 나타내는 '버전'의 수를 포함한다. 헤더는 'RDFHeader' 청크의 일부를 형성하기도 하는, 리스트 'schemas' 에서의 'RDFSchema' 청크의 수를 나타내는 'numOfSchemas'의 다른 수를 포함한다. 몇몇 실시예에서, 'RDFSchema' 청크는 복합적인 리소스를 더 효과적으로 나타내는데 이용된다. 일실시예에서, 'RDFSchema' 청크에 포함되는 데이터는 다음과 같이 나타낼 수 있다(ANSI C 프로그래밍 언어로 표현되어 있음).

[0207] typedef struct RDFSchemaStruct

[0208] {

[0209] wchar_t* prefix;

[0210] wchar_t* uri;

[0211] } RDFSchema;

[0212] 'RDFSchema' 청크는, '프리픽스'로 식별된 'dc' 등의 텍스트로 이루어진 제1 문자열과, 'uri'로 식별된 "http://purl.org/dc/elements/1.1/'과 같은 텍스트로 이루어진 제2 문자열을 포함한다. '프리픽스'는 'uri' 대신에 '메타데이터'에서 사용될 수 있는 용어를 정의한다. 'uri'는, 범용 리소스 식별자(Universal Resource Identifier)로서, 특정의 표준화된 어휘에 부합하거나 특정의 시스템에 대한 특정의 어휘에 부합할 수 있다.

[0213] 'RDFData' 청크에 대하여 다시 설명한다. 'RDFHeader' 청크에 추가하여, 'RDFData' 청크는 'RDFStatement' 청크의 'numOfStatements' 값 및 'statement' 리스트를 포함한다. 'numOfStatements' 값은 정보를 포함하는 리스트 'statement'에서의 'RDFStatement' 청크의 실제 개수를 나타낸다. 일실시예에서, 'RDFStatement' 청크에 포함된 데이터는 다음과 같이 표현될 수 있다(ANSI C 프로그래밍 언어로 표현되어 있음).

[0214] typedef struct RDFStatementStruct

[0215] {

[0216] RDFSubject subject;

[0217] RDFPredicate predicate;

[0218] RDFObject object

[0219] RDFAuthority authority;

[0220] } RDFStatemtnt;

[0221] 각각의 'RDFStatement' 청크는 멀티미디어 파일과 관련된 '메타데이터'의 일부를 포함한다. 청크인 '주어부', '술어부', 객체부' 및 '출처부'는 상술한 '메타데이터'의 다양한 성분을 저장하는데 이용된다. '주어부'는 'RDFSubject' 청크이며, 앞서 설명한 '메타데이터'의 다양한 구성 요소를 유지하는데 이용된다.

[0222] '주어'는 'RDFStubject' 청크이며, 이 청크는 앞서 설명한 '메타데이터'의 제어부분을 나타낸다. 일실시예에서, 'RDFSubject' 청크 내에 포함되는 데이터는 다음과 같이 표현될 수 있는 'RDFSubject'내에 포함된다(ANSI C 프로그래밍 언어로 표현되어 있음).

[0223] *typedef struct RDFSSubjectStruct

[0224] {

[0225] unit16_t type;

[0226] wchar_t* value;

[0227] } RDFSubject;

[0228] 위에 나타낸 'RDFSubject' 청크는, 데이터가 리소스(Resource) 또는 '메타데이터'와 유니코드 텍스트 문자열 '값'의 일부의 익명의 노드(anonymous node)라는 것을 나타내는 'type'(타입) 값을 포함한다. 이 유니코드 텍스트 문자열 '값'(value)은 '메타데이터'의 일부의 주어부를 나타내는 데이터를 포함한다. 'RDFSchema' 청크가 정의된 실시예에서, 그 값은 리소스에 대한 직접적인 참조 대신에 정의된 용어가 도리 수 있다.

[0229] 'RDFStatement' 청크에서의 '서술부'(predicate)는 'RDFPredicate' 청크이며, '메타데이터'의 서술부의 일부를 나타낸다. 일실시예에서, 'RDFPredicate' 청크 내에 포함된 데이터는 다음과 같이 나타낼 수 있다(ANSI C 프로그래밍 언어로 표현되어 있음).

[0230] *typedef struct RDFPredicateStruct

[0231] {

[0232] unit16_t type;

[0233] wchar_t* value;

[0234] } RDFPredicate;

[0235] 위에 나타낸 'RDFPredicate' 청크는, 데이터가 서술형 URI 또는 '메타데이터'의 일부 정규의 리스트 엔트리와 텍스트 문자열 '값을 나타내는, '타입'(type) 값을 포함하며, 여기서 텍스트 문자열 '값'은 '메타데이터'의 일부의 서술부를 나타내는 데이터를 포함한다. 'RDFSchema' 청크가 정해진 실시예에서, 그 값은 리소스에 대한 직접 참조 대신에 정의된 용어가 될 수 있다.

[0236] 'RDFStatement' 청크에서 '객체'(object)는 'RDFObject' 청크이며, '메타데이터'의 일부의 객체 부분을 나타낸다. 일실시예에서, 'RDFObject' 청크에 포함된 데이터는 다음과 같이 표현될 수 있다(ANSI C 프로그래밍 언어로 표현되어 있음).

[0237] typedef struct RDFObjectStruct

[0238] {

[0239] unit16_t type;

[0240] wchar_t* value;

[0241] wchar_t* dataTypeURI;

[0242] wchar_t* value;

[0243] } RDFObject;

[0244] 위에 나타낸 'RDFObject' 청크는, 데이터의 일부가 EUC-KR 문자열, 문자 정수또는 '메타데이터'의 일부의 문자 XML 데이터라는 것을 나타내는 'type' 값을 포함한다. 이 청크는 또한, 3개의 값을 포함한다. 첫 번째 값 'language'는 '메타데이터' 부분이 표현되는 언어를 나타내는데 이용된다(예컨대, 영화의 타이틀이 언어에 따라 달라질 수 있다). 몇몇 실시예에서, 언어를 식별하기 위하여 표준 표현이 이용될 수 있다(예컨대, RFC3066-IETF에서 지정한 언어의 식별을 위한 태그, http://www.ietf.org/rfc/rfc3066.txt 참조). 두 번째 값인 'dataTypeURI'는, 'type'(타입) 필드에 의해 명시적으로 표시되지 않는 경우, 'value'(값) 필드 내에 포함되는 데이터의 타입을 나타내는데 이용된다. 특정 타입의 데이터를 기술하는데 이용되는 일반적인 RDF URI Vocabulary에 대한 dataTypeURI에서 지정된 URI 포인트가 이용된다. URI가 표현될 수 있는 여러 포맷에 대해서는 http://www.w3.org/TR/rdf-concepts/#section-Datatypes에서 설명하고 있다. 일실시예에서, 'value'는 'wide character'(확장 문자)이다. 다른 실시예에서, 이 'value'는 단일 비트에서 이미지 또는 비디오 시퀀스까지의 다양한 타입의 데이터 중 임의의 것이 될 수 있다. 이 'value'는 '메타데이터'의 객체 부분을 포함한다.

[0245] 'RDFStatement' 청크에서의 'authority'(출처부)는 'RDFAuthority' 청크이며, '메타데이터'의 일부 중 출처부 부분을 나타낸다. 일실시예에서, 'RDFAuthority' 청크 내에 포함되는 데이터는 다음과 같이 나타낼 수 있다(ANSI C 프로그래밍 언어로 표현되어 있음).

[0246] typedef struct RDFAuthorityStruct

[0247] {

[0248] unit16_t type;

[0249] wchar_t* value;

[0250] } RDFAuthority;

[0251] 위에 나타낸 'RDFAuthority' 데이터는 이 데이터가 리소스 또는 '메타데이터'의 일부의 익명의 노드라는 것을 나타내는 'type' 값을 포함한다. 'value'는 '메타데이터'에 대한 출처부를 나타내는 데이터를 포함한다. 'RDFSchema' 청크가 정의된 실시예에서, 그 값은 리소스에 대한 직접 참조 대시에 정의된 용어가 될 수 있다.

[0252] 도 2d에, 본 발명의 실시예에 따른 멀티미디어 파링의 'DXDT' 청크의 기억장치를 개념적으로 나타낸다. 'DXDT' 청크(38)는, 'RDFHeader' 청크(110), 'numOfStatements' 값(112) 및 RDFStatement 청크(114)의 리스트를 포함한다. 'RDFHeader' 청크(110)는, '버전' 값(116), 'numOfSchemas' 값(118), 및 'Schema' 청크(120)의 리스트를 포함한다. 각각의 'RDFStatement' 청크(114)는 'RDFSubject' 청크(122), 'RDFPredicate' 청크(124), 'RDFObject' 청크(126) 및 'RDFAuthority' 청크(128)를 포함한다. 'RDFSubject' 청크는, 'type' 값(130) 및

'value' 값(136)을 포함한다. 'RDFObject' 청크(126)는, 'type' 값(138), 'language' 값(140)(도면에서는 'lang'로 표시함), 'dataTypeURI' 값(142)(도면에서는 'dataT'로 표시함) 및 'value' 값(144)을 포함한다. 'RDFAuthority' 청크(128)는, 'type' 값(146) 및 'value' 값(148)을 포함한다. 'DXDT' 청크는 단일의 'Schema' 청크와 단일의 'RDFStatement' 청크를 포함하는 것으로 도시하고 있지만, 본 기술분야의 당업자라면 여러 개수의 'Schema' 청크와 'RDFStatement' 청크가 '메타데이터'를 기술하는 청크에 사용될 수 있다는 것을 용이하게 이해할 수 있을 것이다.

[0253] 이하 설명하는 바와 같이, 본 발명의 실시예에 따른 멀티미디어 파일은 계속해서 변경 및 갱신될 수 있다. '메타데이터'가 파일 자체와 관련되도록 미리 판정하는 것과 '메타데이터'가 원격으로 액세스(예컨대, 인터넷을 통해)되도록 미리 판정하는 것은 어려울 수 있다. 통상적으로, 파일의 컨텐츠를 기술하기 위하여, 본 발명의 실시예에 따른 멀티미디어 파일 내에는 충분한 '메타데이터'가 포함된다. 파일을 검토하는 장치가 네트워크를 통해 파일 내에서부터 참조된 '메타데이터'를 포함하는 다른 장치를 액세스할 수 있는 경우 추가의 정보가 획득될 수 있다.

[0254] 상기 설명한 '메타데이터'를 나타내는 방법은 확장이 가능하며, 시간의 경과에 따라 변경하는데 필요한 파일 내에 저장된 상이한 '메타데이터' 필드를 추가 및 제거하는 능력을 제공한다. 또한, '메타데이터'의 표현이 수정 표현들 사이에서 포워드 호환 가능하다.

[0255] '메타데이터'가 본 발명의 실시예에 따라 표현되는 구조화 방식에 의하여, 장치는 멀티미디어 파일에 질의를 행하여 그 파일 내의 컨텐츠를 더 효과적으로 판정할 수 있게 된다. 이러한 질의(query)는 멀티미디어 파일의 컨텐츠를 갱신하고, 멀티미디어 파일에 관한 추가의 '메타데이터'를 획득하며, 파일의 컨텐츠에 관련된 메뉴를 생성하고, 표준 포맷으로 표현된 데이터를 자동 처리를 포함하는 임의의 다른 기능을 수행하는데 이용될 수 있다. 또한, '메타데이터'의 파서 가능한 소자의 길이를 한정함으로써, 가정용 전자 제품 등과 같이 기억용량이 제한된 장치를 이용하여 '메타데이터'를 액세스하는 것이 더 용이하게 된다.

[0256] 다른 실시예에서, '메타데이터'는 '메타데이터'의 각 부분에 대한 개별 청크를 이용하여 표현된다. 본 발명에 따른 몇 개의 'DXDT' 청크는 상술한 바와 같이 '메타데이터'와 상술한 바와 같이 또는 다른 형식으로 포맷화된 '메타데이터'의 각각의 부분을 포함하는 추가의 청크를 포함하는 이진 청크를 포함한다. 이진 '메타데이터'가 DXDT' 청크에 포함되는 실시예에서, 이진 '메타데이터'는 64 비트로 부호화된 ASCII 코드를 이용하여 표현될 수 있다. 다른 실시예에서는, 다른 이진 표현도 가능하다.

[0257] 본 ㅂ라명에 따라 'DXDT' 청크에 포함될 수 있는 개별 청크의 예가 도 2da에 도시되어 있다. '메타데이터'는, 'PixelAspectRatioMetaData' 청크(152a), 'EncoderURIMetaData' 청크(152b), 'CodexSettingsMetaData' 청크(152c), 'FrameTypeMetaData' 청크(152d), 'VideoResolutionMetaData' 청크(152e), 'PublisherMetaData' 청크(152f), 'CreatorMetaData' 청크(152g), 'GenreMetaData' 청크(152h), 'CreatorToolMetaData' 청크(152i), 'RightsMetaData' 청크(152j), 'RunTimeMetaData' 청크(152k), 'QuantizerMetaData' 청크(152l), 'CodecInfoMetaData' 청크(152m), 'EncoderNameMetaData' 청크(152n), 'FrameRateMetaData' 청크(152o), 'InputSourceMetaData' 청크(152p), 'FileIDMetaData' 청크(152q), 'TypeMetaData' 청크(152r), 'TitleMetaData' 청크(152s) 및/또는 'CertLevelMetaData' 청크(152t)를 포함할 수 있는 'MetaData' 청크(150)를 구비한다.

[0258] 'PixelAspectRatioMetaData' 청크(152a)는 부호화된 비디오의 픽셀 종횡비(어스팩트비)에 관한 정보를 포함한다. 'EncoderURIMetaData' 청크(152b)는 인코더에 관한 정보를 포함한다. 'CodexSettingsMetaData' 청크(152c)는 비디오를 부호화하는데 이용되는 코덱의 설정에 관한 정보를 포함한다. 'FrameTypeMetaData' 청크(152d)는 비디오 프레임에 관한 정보를 포함한다. 'VideoResolutionMetaData' 청크(152e)는 부호화된 비디오의 영상 해상도에 관한 정보를 포함한다. 'PublisherMetaData' 청크(152f)는 미디어를 발행하는 개인이나 기관에 관한 정보를 포함한다. 'CreatorMetaData' 청크(152g)는 컨텐츠의 작성자에 관한 정보를 포함한다. 'GenreMetaData' 청크(152h)는 미디어의 장르에 관한 정보를 포함한다. 'CreatorToolMetaData' 청크(152i)는 파일을 작성하는데 이용되는 도구에 관한 정보를 포함한다. 'RightsMetaData' 청크(152j)는 DRM 관련 정보를 포함한다. 'RunTimeMetaData' 청크(152k)는 미디어의 실행 시간에 관련된 정보를 포함한다. 'QuantizerMetaData' 청크(152l)는 비디오를 부호화하는데 이용하는 양자화기에 관한 정보를 포함한다. 'CodecInfoMetaData' 청크(152m)은 코덱 관련 정보를 포함한다. 'EncoderNameMetaData' 청크(152n)은 인코더의 이름 관련 정보를 포함한다. 'FrameRateMetaData' 청크(152o)는 미디어의 프레임 레이트(frame rate)에 관련된 정보를 포함한다. 'InputSourceMetaData' 청크(152p)는 입력 소스에 관한 정보를 포함한다.

'FileIDMetaData' 청크(152q)는 파일에 대해 고유한 식별자를 포함한다. 'TypeMetaData' 청크(152r)은 멀티미디어 파일의 타입에 관한 정보를 포함한다. 'TitleMetaData' 청크(152s)는 미디어의 타이틀을 포함하며, 'CertLevelMetaData' 청크(152t)는 미디어의 인증 수준에 관한 정보를 포함한다. 다른 실시예에서는, 추가의 '메타데이터'를 포함하는 추가의 청크가 포함될 수 있다. 몇몇 실시예에서, 상술한 바와 같이 '메타데이터'를 이진 형태로 포함하는 청크는 '메타데이터' 청크 내에 포함될 수 있다. 일실시예에서, 이진 '메타데이터'의 청크는 64 비트의 ASCII 코드로서 부호화된다.

[0259] 2.5 'DMNU' 청크

[0260] 도 2a 내지 2.0.1을 참조하면, 제1 'DMNU' 청크 (40; 40')와 제2 'DMNU' 청크(46; 46')가 도시되어 있다. 도 2a에서, 제2 'DMNU' 청크(46)는 멀티미디어 파일(30)의 일부를 형성한다. 도 2aa에 나타낸 실시예에서, 'DMNU' 청크(46')는 개별의 RIFF 청크 내에 포함된다. 위 2가지 경우 모두, 제1 및 제2 'DMNU' 청크가 조정가능한 메뉴를 표시하는데 이용될 수 있는 데이터를 포함한다. 일실시예에서, 제1 'DMNU' 청크(40; 40')는 확장된 애니메이션 배경과 같이 진보된 특징을 포함하지 않는 단순한 메뉴를 작성하는데 이용될 수 있는 데이터를 포함한다. 또한, 제2 'DMNU' 청크(46; 46')는 확장된 애니메이션 배경과 같이 진보된 특징을 포함하는 보다 복잡한 메뉴를 작성하는데 이용될 수 있는 데이터를 포함한다.

[0261] 몇몇 실시예에서, 단순하고 복잡한 메뉴를 제공함으로써, 장치는 원하는 메뉴를 선택하여 표시할 수 있게 된다. 'movi' 리스트 청크(42)의 앞에 더 작은 2개의 매뉴를 배치함으로써, 메뉴를 표시할 수 없는 본 발명의 실시예에 따른 장치가 표시될 수 없는 정보를 신속하게 스킵할 수 있도록 한다.

[0262] 다른 실시예에서, 단일의 메뉴를 작성하는데 필요한 데이터는 제1 'DMNU' 청크와 제2 'DMNU' 청크로 분할된다. 대안으로서, 'DNMU' 청크는 단일 세트의 메뉴 또는 다중 세트의 메뉴에 대한 데이터를 포함하는 'movi' 청크의 앞에 놓이는 단일의 청크가 될 수 있다. 다른 실시예에서, 'DMNU' 청크는 멀티미디어 파일 전체를 통해 다른 위치에 배치되는 단일의 청크 또는 다중의 청크가 될 수 있다.

[0263] 본 ㅂ라명의 실시예에 따른 몇몇 멀티미디어 파일에 있어서, 제1 'DMNU' 청크(40; 40')는 제2 'DMNU' 청크(46; 46') 내의 '더 풍부한' 메뉴에 기초하여 자동으로 생성될 수 있다. 메뉴의 자동 생성에 대해서는 나중에 더 상세하게 설명한다.

[0264] 본 발명의 실시예에 따른 'DMNU' 청크의 구조에 대하여 도 2e에 나타낸다. 'DMNU' 청크(158)는 메뉴 청크(160)와 'MRIF' 청크(162)를 포함하는 리스트 청크이다. 메뉴 청크는 메뉴를 통해 구성 및 네비게이트하는데 필요한 정보를 포함한다. 'MRIF' 청크는 서브타이틀, 배경 비디오 및 배경 오디오를 메뉴에 제공하는데 이용될 수 있는 미디어 정보를 포함한다. 몇몇 실시예에서, 'DMNU' 청크는 메뉴를 여러 가지 언어로 표시할 수 있는 메뉴 정보를 포함한다.

[0265] 일실시예에서, 'WowMenu' 청크(160)는 도 2f에 개념적으로 도시한 메뉴 정크 객체의 계층을 포함한다. 이 계층의 최상부는 'WowNenuManager' 청크(170)이다. 'WowNenuManager' 청크는 하나 이상의 'LanguageMenu' 청크(172)와 하나의 'Media' 청크(174)를 포함할 수 있다.

[0266] 'LanguageMenus' 청크(172)를 이용함으로써, 'DMNU' 청크(158)는 상이한 언어로 된 메뉴 정보를 포함할 수 있다. 각각의 'LanguageMenus' 청크(172)는 지정된 언어로 된 완전한 세트의 메뉴를 생성하는데 이용되는 정보를 포함한다. 따라서, 'LanguageMenus' 청크는 'LanguageMenus' 청크와 관련된 정보의 언어를 식별하는 식별자를 포함한다. 'LanguageMenus' 청크는 또한 'WowMenu' 청크(175)의 리스트를 포함한다.

[0267] 각각의 'WowMenu' 청크(175)는 특정의 메뉴에 대해 스크린상에 표시될 모든 정보를 포함한다. 이 정보는 배경 비디오 및 오디오를 포함할 수 있다. 이 정보는 또한 다른 메뉴에 액세스하거나, 메뉴에서 빠져나가고 멀티미디어 파일의 일부의 표시를 개시하는데 이용될 수 있는 버튼 동작과 관련된 데이터를 포함할 수 있다. 일실시예에서, 'WowMenu' 청크(175)는 미디어에 대한 참조 리스트를 포함한다. 이들 참조는, 이하 더 상세하게 설명하는 'Media' 청크(174)에 포함된 정보를 참조한다. 이러한 미디어에 대한 참조는 메뉴에 대한 배경 비디오 및 배경 오디로를 정의할 수 있다. 'WowMenu' 청크(175)는 또한, 메뉴가 첫 번째로 액세스될 때, 특정의 버튼을 강조(밝게)하는데 이용될 수 있는 오버레이(overlay)를 정의한다.

[0268] 또한, 각각의 'WowMenu' 청크(175)는 많은 수의 'ButtonMenu' 청크(176)를 포함한다. 각각의 'ButtonMenu' 청크는 스크린상의 버튼의 특성을 정의한다. 'ButtonMenu' 청크는, 사용자에 의해 버튼이 강조될 때 이용되는 오버레이, 버튼의 이름, 및 메뉴를 통한 사용자의 네비게이션에 의해 수행되는 다양한 동작에 어떻게 응답하는 지에 관한 것들을 기술할 수 있다. 동작에 따른 응답은 'Action' 청크(178)를 참조함으로써 정의된다. 단일의

동작, 예컨대 버튼을 선택하는 동작에 의해 여러 개의 'Action' 청크가 액세스될 수 있다. 사용자가 디스플레이 상에서 온스크린 포인터를 제한 없이 이동시킬 수 있는 마우스 등의 장치를 이용하여 메뉴와 상호동작을 행할 수 있는 환경에서, 버튼의 온스크린 위치는 'MenuRectangle' 청크(180)를 이용하여 정의될 수 있다. 버튼의 온스크린 위치를 알면, 시스템은 자유 범위의 입력 장치를 이용하면서, 사용자가 버튼을 선택하고 있는지를 판정할 수 있다.

[0269] 각각의 'Action' 청크는 동작 관련 청크의 많은 상이한 변형 중 하나 이상을 식별하며, 이러한 동작 관련 청크에는, 'PlayAction' 청크(182), 'MenuTransitionAction' 청크(184), 'ReturnToPlayAction' 청크(186), 'AudioSelectAcion' 청크(188), 'SubtitleSelectAction' 청크(190) 및 'ButtonTransitionAction' 청크(191)가 포함된다. 'PlayAction' 청크(182)는 멀티미디어 파일 내의 비디오, 오디오 및 서브타이틀 트랙의 각각의 부분을 식별한다. 'PlayAction' 청크는 'MediaTrack' 청크에 대한 참조를 이용하여 비디오 트랙의 일부를 참조한다(이하 참조). 'PlayAction' 청크는 'SubtitleTrack' 청크(192)와 'AudioTrack' 청크(194)를 이용하여 오디오 및 서브타이틀 트랙을 식별한다. 'SubtitleTrack' 청크와 'AudioTrack' 청크는 'MediaTrack' 청크(198)에 대한 참조를 포함한다. 'PlayAction' 청크가 본 발명의 실시예에 따라 동작을 기본을 형성하면, 선택된 오디오 및 서브타이틀 트랙이 디폴트로서 초기에 설정된 변수 값으로서 결정되고, 사용자가 메뉴를 사용하여 변경시킬 수 있다.

[0270] 각 'MenuTransitionAction' 청크(184)는 'WowMenu' 청크(175)에 관한 참조를 포함하고 있다. 상기 참조는 다른 메뉴를 표시하거나 전환하기 위한 정보를 얻을 수 있다.

[0271] 각 'ReturnToPlayAction' 청크(186)는 사용자가 메뉴를 불러오기 전에 액세스했었던 멀티미디어 파일 부분으로 조작자가 되돌아갈 수 있도록 하는 정보를 포함하고 있다.

[0272] 각 'AudioSelectAction' 청크(188)는 특정의 오디오 트랙을 선택하는데 이용될 수 있는 정보를 포함하고 있다. 일 실시예에서, 오디오 트랙은 본 발명의 실시예에 따른 멀티미디어 파일의 내에 포함된 오디오 트랙으로부터 선택된다. 다른 실시예에서, 오디오 트랙은 외부에서 참조하는 파일에 위치할 수 있다.

[0273] 각 'SubtitleSelectAction' 청크(190)는 특정의 서브타이틀을 선택하는데 이용될 수 있는 정보를 포함하고 있다. 일 실시예에서, 서브타이틀 트랙은 본 발명의 실시예에 따른 멀티미디어 파일의 내에 포함된 서브타이틀로부터 선택된다. 다른 실시예에서, 서브타이틀 트랙은 외부에서 참조하는 파일에 위치할 수 있다.

[0274] 각 'ButtonTransitionAction' 청크(191)는 동일 메뉴 내에서의 다른 버튼으로 전환하는데 이용될 수 있는 정보를 포함하고 있다. 다른 버튼으로의 전환은, 하나의 버튼과 관련된 다른 동작이 수행된 후에 행하여진다.

[0275] 'Media' 청크(174)는 많은 수의 'MediaSource' 청크(166)와 'MediaTrack' 청크(198)를 포함한다. 'Media' 청크는 자질문자(feature system)와 메뉴 시스템에 이용되는 모든 멀티미디어 트랙(예컨대, 오디오, 비디오, 서브타이틀)을 규정한다. 각 'MediaSource' 청크(196)는 본 발명의 실시예에 따른 멀티미디어 파일 내의 RIFF 청크를 정의함으로써, 다수의 RIFF 청크를 포함할 수 있다. 각 'MediaTrack' 청크(198)는 'MediaSource' 청크에 의해 특정되는 RIFF 청크 내의 멀티미디어 트랙의 일부를 식별한다.

[0276] 'MRIF' 청크(162)는 그 자신의 RIFF 포맷에 따르는 소형 멀티미디어 파일이다. 'MRIF' 청크는 메뉴용의 배경 오디오, 비디오 및 오버레이를 제공하는데 이용되는 오디오, 비디오 및 서브타이틀 트랙을 포함한다. 'MRIF' 청크는 강조된 메뉴 버튼을 지시하기 위한 오버레이로 이용될 수 있는 비디오를 포함할 수도 있다. 일 실시예에서, 메뉴 데이터가 덜 요구되는 실시예의 경우, 배경 비디오는 스틸 프레임(avi 포맷의 변경예)이 되거나 시퀀스가 작은 동일한 프레임이 될 수 있다. 다른 실시예에서, 더 정교한 비디오 시퀀스는 배경 비디오를 제공하는데에 이용될 수 있다.

[0277] 상술한 바와 같이, 'WowMenu' 청크(175)와 'WowMenu' 청크 부분을 형성하는 다양한 청크는 실제 미디어 트랙에 대한 참조를 자체적으로 포함하고 있다. 이러한 각 참조는 전형적으로 RIFF 청크의 'hdrl' 리스트 청크에서 규정한 미디어 트랙에 대한 참조가다.

[0278] 도 2g에는 본 발명에 따라 'DMNU' 청크를 작성하는데 이용될 수 있는 다른 청크들이 도시되어 있다. 'DMNU' 청크는 'WowMenuManager' 청크(170')를 포함한다. 'WowMenuManager' 청크(170')는 적어도 하나의 "LanguageMenus' 청크(172'), 적어도 하나의 'Media' 청크(174'), 및 적어도 하나의 'TranslationTable' 청크(200)를 포함한다.

[0279] 'LanguageMenus' 청크(172')의 컨텐츠는, 도 2f에 도시된 'LanguageMenus' 청크(172)의 컨텐츠와 대략적으로 유사하다. 주요한 차이는 'PlayAction' 청크(182')가 'SubtitleTrack' 청크(192)와 'AudioTrack' 청크(194)를 포함하고 있지 않다는 것이다.

[0280] 'Media' 청크(174')는 도 2f에 도시된 'Media' 청크(174)와 명백히 다르다. 'Media' 청크(174')는 적어도 하나의 'Title' 청크(202)와 적어도 하나의 'MenuTracks' 청크(204)를 포함한다. 'Title' 청크는 멀티미디어 파일 내의 타이틀을 의미한다. 상기 언급한 바와 같이, 본 발명의 실시예에 따른 멀티미디어 파일은 하나 이상의 타이틀(예컨대, TV 시리즈에서의 여러 개의 에피소드, 장편 영화 중에서 연관된 시리즈, 또는 단순한 다른 영화의 선택)을 포함할 수 있다. 'MenuTracks' 청크(204)는 메뉴 디스플레이, 오디오 사운드 트랙 및 디스플레이에 동반되는 서브타이틀을 작성하는데 이용되는 미디어 정보에 대한 정보를 포함한다.

[0281] 'Title' 청크는 적어도 하나의 'Chapter' 청크(206)를 포함할 수 있다. 'Chapter'청크(206)는 특정한 타이틀 내에서의 장면(scene)을 참조한다. 'Chapter' 청크(206)는 'Chapter' 청크에 의해 나타나는 장면에 대응하는 비디오 트랙, 각 오디오 트랙 및 서브타이틀 트랙의 일부에 대한 참조를 포함한다. 일 실시예에서, 참조는 도 2f와 관련해서 언급된 것과 유사한 'MediaSource' 청크(196')와 'MediaTrack' 청크(198')를 이용하여 구현된다. 몇몇 실시예에서, 'MediaTrack' 청크는 비디오 트랙의 적절한 부분과 다수의 추가적인 'MediaTrack' 청크를 참조하며, 이들 추가적인 'MediaTrack' 청크의 각각은 오디오 트랙이나 서브타이틀 트랙 중 하나를 참조한다. 일 실시예에서, 특정의 비디오 트랙에 대응하는 모든 오디오 트랙과 서브타이틀 트랙은 별도의 'MediaTrack' 청크를 이용하여 참조된다.

[0282] 상술한 바와 같이, 'MenuTracks' 청크(204)는 메뉴의 오디오, 비디오 및 오버레이 미디어를 생성하는데 이용되는 미디어에 대한 참조를 포함한다. 일 실시예에서, 미디오 정보에 대한 참조는 'MenuTracks' 청크에 있는 'MediaSource'청크(196')와 'MediaTrack' 청크(198')를 이용하여 이루어진다. 일 실시예에서, 'MediaSource' 청크(196')와 'MediaTrack' 청크(198')는, 도 2f와 관련하여 설명한 부분에서의 방식으로 구현된다.

[0283] 'TranslationTable' 청크(200)는 각 타이틀과 챕터를 다양한 언어로 기술하는 문자열(text string)을 포함하는데 이용될 수 있다. 일 실시예에서, 'TranslationTable' 청크(200)는 적어도 하나의 'TranslationLookup' 청크(208)를 포함한다. 각 'TranslationLookup' 청크(208)는 'Title' 청크(202), 'Chapter' 청크(206) 또는 'MediaTrack' 청크(196')와 관련이 있고, 복수의 'Translation' 청크(210)를 포함한다. 'TranslationLookup' 청크의 각 'Translation' 청크는, 'TranslationLookup' 청크와 관련된 청크를, 'Translation' 청크에 의해 지시되는 언어로 기술하는 문자열을 포함한다.

[0284] 도 2ga에는, 'DMNU' 청크 안에 포함된 다양한 청크 사이의 관계를 개념적으로 나타내고 있다. 도 2ga은 굵은 실선의 화살표시를 이용하여, 하나의 청크가 다른 청크에 포함되는 것을 보여준다. 화살표시가 가리키는 방향은 하나의 청크가 화살표가 시작되는 부분의 다른 청크에 포함되는 것을 나타낸다. 하나의 청크가 다른 청크를 참조하는 것은 점선으로 나타내고 있으며, 참조된 청크는 점선 화살표시로 나타내고 있다.

[0285] 2.6. 'junk' 청크

[0286] 'junk' 청크(41)는 본 발명의 실시예에 따른 멀티미디어 파일에 포함될 수 있는 임의 선택적인 청크이다. 'junk' 청크의 속성은 AVI 포맷에서 지정된다.

[0287] 2.7. 'movi' 리스트 청크

[0288] 'movi' 리스트 청크(42)는 다수의 'data' 청크를 포함한다. 'data' 청크가 포함할 수 있는 정보의 예는, 오디오, 비디오 또는 서브타이틀 데이터이다. 일 실시예에서, 'movi' 리스트 청크는 적어도 하나의 비디오 트랙, 다수의 오디오 트랙및 다수의 서브타이틀 트랙용 데이터를 포함한다.

[0289] 도 2h에는, 하나의 비디오 트랙, 3개의 오디오 트랙 및 3개의 서브타이틀 트랙을 포함하는 멀티미디어 파일의 'movi' 리스트 청크(42)에 'data' 청크를 인터리브(삽입) 배치하는 것(인터리빙)이 나타나 있다. 편의상, 비디오를 포함하는 'data' 청크를 'video' 청크로 하고, 오디오를 포함하는 'data' 청크를 'audio' 청크라고 한다. 또한, 서브타이틀을 포함하는 'data' 청크를 'subtitle' 청크로 할 것이다. 도면에 표시된 'movi' 리스트 청크(42)에 있어서, 각 'video' 청크(262)는 각 오디오 트랙으로부터의 'audio' 청크(264)에 의해 다음 'video' 청크와 분리되어 있다. 여러 실시예에서, 'audio' 청크는 'audio' 청크에 후속하는 'video' 청크에 포함된 비디오의 일부에 대응하는 오디오 트랙의 일부를 포함한다.

[0290] 인접한 'video' 청크는, 하나 이상의 'subtitle'청크(266)에 의해, 하나의 서브타이틀 트랙으로부터 분리될 수

있다. 일 실시예에서, 'subtitle' 청크(266)는 서브타이틀, 개시 시간, 정지 시간을 포함한다. 여러 실시예에서, 'subtitle' 청크는, 'subtitle' 청크에 후속하는 'video' 청크가 서브타이틀의 개시 시간에 동작하는 비디오 부분을 포함하도록, 'movi' 리스트 청크에 인터리브된다. 다른 실시예에서, 모든 'subtitle'과 'audio' 청크의 개시 시간은 이에 상응하는 비디오의 개시 시간에 앞선다. 일 실시예에서, 'audio'와 'subtitle' 청크는 대응되는 'video' 청크의 개시 시간부터 5초 범위 이내에 배치될 수 있고, 다른 실시예에서는, 파일 내의 오디오와 비디오를 표시할 수 있는 장치에 의해 버퍼링될 수 있는 비디오의 분량과 관련된 시간 내에 배치될 수 있다.

[0291] 일 실시예에서, 'data' 청크는 'data' 청크가 속하는 스트림을 식별하기 위해 'FOURCC' 코드를 포함한다. 'FOURCC' 코드는 청크에서 정보의 타입을 규정하는 2자리 문자 코드가 뒤따르는 2자리 숫자 스트림 번호로 구성된다. 다른 'FOURCC' 코드는 2자리 숫자 스트림 번호가 뒤따르는 청크에서 정보의 타입을 정의하는 2자리 문자 코드를 포함하여 이루어진다.

[0292] 2자리 문자 코드의 예를 다음의 표에 나타낸다.

표 2

[0293] FOURCC 코드에서 사용되는 선택된 2자리 문자 코드

2자리 문자 코드	설명
db	압축해제된 비디오 프레임
dc	압축된 비디오 프레임
dd	비디오 프레임용 DRM 키 정보
pc	팔레트 변경
wb	오디오 데이터
st	서브타이틀(텍스트 모드)
sb	서브타이틀(비트맵 모드)
ch	챕터

[0294] 일 실시예에서, 'video' 청크(262)와 'audio' 청크(264)의 구조는 AVI 파일 포맷에 따른다. 다른 실시예에서, 비디오의 속성을 특정하거나 부호화된 미디어를 포함하는 청크용의 다른 포맷이 이용될 수 있다.

[0295] 여러 실시예에서, 'subtitle' 청크(266) 내에 포함된 데이터는 다음과 같이 표현할 수 있다.

[0296] typedef struct_subtitlechunk {

[0297] FOURCC fcc ;

[0298] DWORD cb;

[0299] STR duration;

[0300] STR subtitle;

[0301] } SUBTITLECHUNK;

[0302] 'fcc'의 값은 서브타이틀 트랙과 서브타이틀 트랙의 속성(텍스트 또는 비트맵 모드)을 나타내는 FOURCC 코드이다. 'cb'의 값은 구조의 크기를 특정한다. 'duration'의 값은 서브타이틀의 시작 지점과 끝 지점에서의 시간을 나타내며, 일 실시예에서는 hh:mm:ss.xxx-hh:mm:ss.xxx의 형태로 될 수 있다. hh는 시간을 나타내고, mm은 분을, ss는 초를, xxx는 밀리초를 각각 나타낸다. 'subtitle'의 값은 텍스트 모드에서의 서브타이틀의 유니코드 텍스트 또는 비트맵 모드에서의 서브타이틀의 비트맵 이미지를 포함한다. 본 발명의 여러 실시예에서는 서브타이틀 정보를 나타내기 위한 압축된 비트맵 이미지를 이용한다.

[0303] 일 실시예에서, 'subtitle' 필드는 서브타이틀의 폭, 높이 및 온스크린 위치에 관한 정보를 포함한다. 또한, 'subtitle' 필드는 컬러(색채) 정보와 실제 비트맵의 화소를 포함할 수 있다. 여러 실시예에서, 런 랭스 부호화(run length encoding)는 비트맵을 표현하는데 필요한 화소 정보의 양을 감소시키는 데 이용할 수 있다.

[0304] 본 발명에 따른 멀티미디어 파일은 디지털 저작권 관리 정보(digital rights management)를 포함할 수 있다. 이 정보는 주문형 비디오 애플리케이션에 이용될 수 있다. 디지털 저작권 관리기능에 의해 보호되는 멀티미디어 파일은, 지정된 재생 권한을 부여받은 사용자에 의해서만 바르게 재생될 수 있다. 일 실시예에서, 디지털

저작권 관리기능에 의해 트랙이 보호되고 있다는 것은, 'hdrl' 리스트 청크(상술한 내용 참조)에서의 트랙에 관한 정보에 표시될 수 있다. 디지털 저작권 관리기능에 의해 보호되는 트랙을 포함하는, 본 발명에 따른 멀티미디어 파일은, 'movi' 리스트 청크에서의 디지털 저작권 관리기능에 관한 정보를 포함할 수 있다.

[0305] 도 2i에는, 비디오 트랙, 다수의 오디오 트랙, 적어도 하나의 서브타이틀 트랙, 및 디지털 저작권 관리기능을 가능하게 하는 정보를 포함하는 본 발명에 따른 멀티미디어 파일의 'movi'리스트 청크가 도시되어 있다. 'movi' 리스트 청크(42')는 각 비디오 청크(262') 이전에 'DRM' 청크(270)가 추가된, 도 2h에 도시된 'movi' 리스트 청크와 유사하다. 'DRM' 청크(270)는 디지털 저작권 관리기능 관련 정보를 포함하는 'data' 청크이고, FOURCC 코드 'nndd'에 의해 식별될 수 있다. 처음 2개의 문자인 'nn'은 트랙 번호를 나타내고, 다음 2개의 문자인 'dd'는 청크가 디지털 저작권 관리기능 관련 정보를 포함한다는 것을 의미한다. 일 실시예에서, 'DRM' 청크(270)는 'DRM' 청크와 동반하는'video' 청크(262')용의 디지털 저작권 관리기능 관련 정보를 제공한다. 디지털 저작권 관리기능에 의해 보호되는 비디오 트랙을 재생시키는 장치는, 'DRM' 청크의 정보를 이용하여, 'video' 청크의 비디오 정보를 복호화한다. 통상, 'video' 청크의 앞에 'DRM' 청크가 없으면, 'video' 청크가 보호되지 않는다는 의미로서 해석된다.

[0306] 본 발명의 실시예에 따른 암호화 시스템에서, 비디오 청크는 부분적으로만 암호화된다. 부분적 암호화가 이용되는 'DRM' 청크는 'video' 청크의 암호화된 부분에 대한 참조와 암호화된 부분을 해독하는 데 이용될 수 있는 키(key)에 대한 참조를 포함하고 있다. 해독 키는 'strd' 청크(상술한 내용 참조)의 일부분에 해당하는 'DRM' 헤더에 위치할 수 있다. 해독 키는 스크램블처리되고 마스터 키로 암호화된다. 'DRM' 헤더는 마스터 키를 식별하는 정보를 포함한다.

[0307] 도 2j에서는, 'DRM' 청크의 정보를 개념적으로 나타내고 있다. 'DRM' 청크(270)는 'frame' 값(280), 'status' 값(282), 'offset' 값(284), 'number' 값(286), 및 'key' 값(288)을 포함할 수 있다. 'frame' 값은 비디오의 암호화된 프레임을 참조하는데 이용될 수 있다. 'status' 값은 프레임이 암호화되었는지 여부를 나타내는데에 이용되고, 'offset' 값(284)은 프레임 내의 암호화된 블록의 개시 부분을 나타내고, 'number' 값(286)은 블록 내의 암호화된 바이트의 수를 나타낸다. 'key' 값(288)은 블록을 해독하는데 이용될 수 있는 해독 키를 참조한다.

[0308] 2.8. 'idxl' 청크

[0309] 'idxl'청크(44)는 'movi'리스트 청크(42)에서 'data' 청크를 인덱스할 수 있는 임의 선택적 청크이다. 일 실시예에서, 'idxl' 청크는 AVI 포맷에서 특정되는 것으로 실행된다. 다른 실시예에서, 'idxl' 청크는 'movi' 리스트 청크에서 각 데이터 청크의 파일 내의 위치를 참조하는 데이터 구조를 이용하여 구현된다. 여러 실시예에서, 'idxl' 청크는 데이터의 트랙 번호와 데이터의 타입에 의해 각 'data' 청크를 식별한다. 이를 위하여, 상기 언급한 FOURCC 코드가 이용될 수 있다.

[0310] 3. 멀티미디어 파일의 부호화

[0311] 본 발명의 실시예는 많은 방법으로 멀티미디어 파일을 생성할 수 있다. 예를 들어, 본 발명에 따른 시스템은 별개의 비디오 트랙, 오디오 트랙 및 서브타이틀 트랙을 포함하는 파일로부터 멀티미디어 파일을 생성할 수 있다. 또한, 메뉴 정보나 "메타데이터"와 같은 다른 정보를 작성하여, 파일에 삽입할 수 있다.

[0312] 본 발명에 따른 다른 시스템도 디지털 비디오 디스크(digital video disk: DVD) 상의 다양한 파일로부터 정보를 추출할 수 있으며, 단일의 멀티미디어 파일을 작성할 수 있다. DVD가 정보의 초기 소스인 경우, 본 발명의 시스템은 더 많은 압축을 얻기 위해 코덱(codec)을 이용할 수 있고, 오디오를 리청크(re-chunk)함으로써, 오디오 청크가 새롭게 작성한 멀티미디어 파일 내의 비디오 청크에 대응하도록 할 수 있다. 또한, DVD 메뉴 시스템에서의 메뉴 정보는 파싱(parsing) 처리되어, 멀티 미디어 파일에 포함된 메뉴 정보를 생성할 수 있다.

[0313] 다른 실시예에서는 본 발명의 실시예에 따라 기존의 멀티미디어 파일에 부가 컨텐츠를 추가함으로써 새로운 멀티미디어 파일을 생성할 수 있다. 부가 컨텐츠를 추가하는 예는, 해설(예컨대, 감독의 논평, 작성한 이후의 휴가 비디오의 이야기 설명)을 포함하는 오디오 트랙 등의 부가 오디오 트랙을 파일에 추가할 수 있을 것이다. 멀티미디어 파일에 인터리브된 부가 오디오 트랙 정보는, 새로운 오디오 트랙의 재생을 가능하게 하기 위하여 멀티미디어 파일에서의 메뉴 정보를 수정함으로써 가능하다.

[0314] 3.1. 저장된 데이터 트랙을 이용한 생성

[0315] 본 발명의 실시예에 따른 멀티미디어 파일을 생성하기 위한 시스템은 도 3a에 나타나 있다. 시스템(350)의 주

요 구성요소는 인터리버(interleaver)(352)이다. 인터리버는, 정보의 청크를 수신하여 인터리브함으로써, 본 발명의 실시예에 따른 멀티 미디어 파일을 상기 설명한 포맷으로 생성할 수 있다. 인터리버는 메타데이터 매니저(354)로부터 "메타데이터"에 관한 정보를 수신한다. 인터리버는 본 발명의 실시예에 따른 멀티미디어 파일을 기억 장치(356)에 출력한다.

[0316] 통상, 인터리버에 제공되는 청크는 기억 장치에 저장된다. 여러 실시예에서, 모든 청크는 동일한 기억 장치에 저장된다. 다른 실시예에서, 청크는 다양한 기억 장치로부터 인터리브에 제공되거나, 생성한다음 인터리브에 실시간으로 제공될 수 있다.

[0317] 도 3a에 도시된 실시예에서는, 'DMNU' 청크(358)와 'DXDT' 청크(360)가 이미 생성되어 기억 장치에 저장되어 있다. 비디오 소스(362)는 기억 장치에 저장되고, 비디오 디코드(364)를 이용하여 복호화된 후, 비디오 인코더(366)를 이용하여 암호화된 다음, 'video' 청크를 생성하게 된다. 오디오 소스(368)도 역시 기억 장치에 저장된다. 오디오 청크는, 오디오 디코더(370)를 이용한 오디오 소스를 복호화하고, 이 복호화된 오디오를 오디오 인코더(372)를 이용하여 부호화하여 생성된다. 'subtitle' 청크는 기억 장치에 저장된 텍스트 서브타이틀(374)로부터 생성된다. 서브타이틀은 제1 트랜스코더(376)에 제공되고, 이 트랜스코더에 의해 복수의 서브타이틀 포맷이 초기 비트맵 포맷으로 전환된다. 일 실시예에서, 저장된 서브타이틀 포맷은 SRT, SUB, 또는 SSA와 같은 포맷이 될 수 있다. 또한, 비트맵 포맷은 컬러 팔레트(palate) 조사 테이블을 포함하는 4비트 비트맵의 포맷이 될 수 있다. 컬러 팔레트 조사 테이블은, 16개로 이루어진 4비트 컬러 코드의 각각에 대해 24비트 컬러 깊이를 갖는 식별자를 포함한다. 단일의 멀티미디어 파일은 하나 이상의 컬러 팔레트 조사 테이블(표 2에서의 "pc" 팔레트 FOURCC 코드 참조)을 포함한다. 4비트 비트맵은 각 메뉴가 1천6백만 컬러의 팔레트에서 동시에 취한 16개의 다른 컬러를 가질 수 있다. 다른 실시예에서, 픽셀 당 비트의 수와 컬러 깊이를 다르게 할 수 있다. 제1 트랜스코더(376)의 출력은 제2 트랜스코더(378)에 제공되고, 제2 트랜스코더(378)에서는 비트맵을 압축한다. 일 실시예에서, 비트맵을 압축하기 위하여 런 랭스 부호화가 이용된다. 다른 실시예에서, 다른 적절한 압축 포맷이 이용된다.

[0318] 일 실시예에서, 다양한 인코더, 디코더, 트랜스코더 사이의 인터페이스(interface)는 마이크로소프트사에서 지정한 Direct Show 표준을 따른다. 다른 실시예에서, 인코딩, 디코딩, 트랜스코딩을 수행하기 위한 소프트웨어는 이러한 표준을 따를 필요는 없다.

[0319] 도시된 실시예에서, 각각의 미디어 소스에 대해 개별의 처리용 구성요소가 도시되어 있다. 다른 실시예에서 리소스(resources)는 공유 가능하다. 예컨대, 단일 오디오 디코더와 오디오 인코더는 모든 소스로부터의 오디오 청크를 생성하는 데 이용될 수 있다. 전형적으로, 전체 시스템은 소프트웨어를 이용하고 하드 디스크 드라이버와 같은 기억 장치에 연결된 컴퓨터에서 수행될 수 있다.

[0320] 상술한 방법으로 인터리버를 사용하기 위하여, 본 발명의 실시예에 따라, 'DMNU' 청크, 'DXDT' 청크, 'video' 청크, 'audio' 청크 및 'subtitle' 청크를 생성하여, 인터리버에 제공되어야 한다. 본 발명의 실시예에 따른 멀티미디어 파일에서의 다양한 청크 각각을 생성하기 위한 과정은 이하 더 구체적으로 설명한다.

[0321] 3.2. 'DXDT' 청크의 생성

[0322] 'DXDT' 청크는 많은 방법으로 생성될 수 있다. 일 실시예에서, 'meta data'는 그래픽 사용자 인터페이스를 통해 데이터 구조로 입력되어, 'DXDT' 청크로 파싱 처리된다. 일 실시예에서, 'meta data'는 일련의 주어부(subject), 술어부(predicate), 객체부(object) 및 출처(authority) 서술문으로 표현된다. 다른 실시예에서, 'meta data' 문은 다양한 포맷으로 표현된다. 여러 실시예에서, 각 'meta data' 문은 파싱 처리되어 개별 청크로 된다. 다른 실시예에서, 제1 포맷(주어부, 술어부, 객체부, 및 출처 표현과 같은)에서의 몇몇 'meta data' 문은 파싱 처리되어 제1 청크로 되고, 다른 포맷으로 된 다른 'meta data' 문은 파싱 처리되어 개별 청크로 된다. 일 실시예에서, 'meta data' 문은 XML 형식의 파일로 기록되고, XML 형식의 파일은 'DXDT' 청크 내에 청크를 작성하도록 파싱 처리된다.

[0323] 도 3b에는, XML 형식의 파일 내에 포함된 일련의 'meta data' 문으로부터 'DXDT' 청크를 생성하기 위한 시스템의 예를 도시하고 있다. 시스템(380)은 파서(parser)(384)에 제공될 수 있는 XML 형식의 파일(382)을 포함한다. XML 형식의 파일은 XML로 암호화된 'meta data'를 포함한다. 파서는 XML을 분석하고 'meta data' 문을 상술한 'meta data' 청크 포맷에 따라 'DXDT' 청크로 기록된 청크로 변환함으로써, 'DXDT' 청크(386)를 생성한다.

[0324] 3.3. 'DMNU' 청크의 생성

[0325] 도 3c에는, 본 발명의 실시예에 따른 'DMNU' 청크를 생성하기 위해 이용될 수 있는 시스템을 도시하고 있다. 시스템(420)을 생성하는 메뉴 청크는 미디어 모델(422)과 미디어 정보를 입력해야 한다. 미디어 정보는, 비디오 소스(424), 오디오 소스(426), 및 오버레이 소스(428)의 형식을 취할 수 있다.

[0326] 시스템을 생성하는 메뉴 청크를 입력으로 이용하는 'DMNU' 청크의 생성은, 복수의 중간 파일(intermediate file)의 생성을 포함한다. 미디어 모델(422)은 XML 형식의 파일(430)을 생성하는 데 이용되고, 미디어 정보는 복수의 AVI 파일(432)을 생성하는 데 이용된다. XML 형식의 파일은 모델 트랜스코더(434)에 의해 작성된다. AVI 파일(432)은 인터리버(436)를 이용하는 비디오, 오디오, 오버레이 정보를 인터리빙함으로써 작성된다. 비디오 정보는, 비디오 소스(424)를 복호화하고 아래 설명될 방법으로 재부호화하기 위해, 비디오 디코더(438)와 비디오 인코더(440)를 이용하여 얻을 수 있다. 오디오 정보는 오디오를 복호화하고 아래 설명될 방법으로 부호화하기 위하여, 오디오 디코더(442)와 오디오 인코더(444)를 이용하여 얻을 수 있다. 오버레이 정보는 제1 트랜스코더(446)와 제2 트랜스코더(448)를 이용하여 생성된다. 제1 트랜스코더(446)는 오버레이를 표준 비트맵과 같은 그래픽 표현으로 전환하고, 제2 트랜스코더(448)는 그래픽 정보를 취하여, 멀티미디어 파일에 포함시켜야 할 때 포맷화한다. 메뉴를 만들기 위해 요구되는 정보를 포함하는 XML 파일 및 AVI 파일이 생성되면, 메뉴 생성기(menu generator)(450)는 'DMNU' 청크(358)를 생성하기 위한 정보를 이용할 수 있다.

[0327] 3.3.1. 메뉴 모델

[0328] 일 실시예에서, 미디어 모델은 모든 메뉴와 보조 구성요소(subcomponents)를 나타내는 객체 지향(object-oriented)의 모델이다. 미디어 모델은 메뉴를 계층구조로 조직화하고, 메뉴가 언어 선택(language selection)에 의해 조직화되도록 한다. 도 3d에, 본 발명의 실시예에 따른 미디어 모델이 도시되어 있다. 미디어 모델(460)은 최상단의 'MediaManager' 객체(462)를 포함하고, 'MediaManager' 객체(462)는, 복수의 'LanguageMenu' 객체(463), 'Media' 객체(464) 및 'TranslationTable' 객체(465)와 관련되어 있다. 'MenuManager' 역시 디폴트(default) 메뉴 언어를 포함한다. 일 실시예에서, 디폴트 언어는 ISO 639 2문자 언어 코드로 표시될 수도 있다.

[0329] 'LanguageMenu' 객체는 언어 선택에 의해 다양한 메뉴용 정보를 조직한다. 소정 언어에 대한 모든 'Menu' 객체(466)는 그 언어에 대한 'LanguageMenu'객체(463)와 연관되어 있다. 각 'Menu' 객체는 복수의 'Button' 객체(468)와 관련이 있고, 복수의 'MediaTrack' 객체(488)를 참조한다. 참조된 'MediaTrack' 객체(488)는 'Menu' 객체에 대한 배경 비디오와 배경 오디오를 나타낸다.

[0330] 각 'Button' 객체(468)는 'Action' 객체(470)와 'Rectangle' 객체(484)와 연관되어 있다. 또한, 'Button' 객체(468)는, 버튼이 디스플레이상에 강조되었을 때 이용되는 오버레이를 나타내는 'MediaTrack' 객체(488)에 대한 참조를 포함한다. 각 'Action' 객체(470)는 'MenuTransition' 객체(472), 'ButtonTransition' 객체(474), 'ReturnToPlay' 객체(476), 'Subtitle Selection' 객체(478), 'AudioSelection' 객체(480) 및 'PlayAction' 객체(482)와 같은 복수의 객체와 연관되어 있다. 이러한 각객체는 사용자로부터의 다양한 입력에 대한 메뉴 시스템의 응답을 규정한다. 'MenuTransition' 객체는 동작에 따라 변환되어야 할 메뉴를 나타내는 'Menu' 객체에 대한 참조를 포함하고 있다. 'ButtonTransition' 객체는 동작에 따라 강조되어야 하는 버튼을 나타낸다. 'ReturnToPlay' 객체는 플레이어로 하여금 피처(feature)의 재생을 재개하도록 할 수 있다. 'SubtitleSelection' 및 'AudioSelection' 객체는 'Title' 객체(487)(이하 참조)에 대한 참조를 포함한다. 'PlayAction' 객체는 'Chapter' 객체(492)(이하 참조)에 대한 참조를 포함한다. 'Rantangle' 객체(484)는 스크린상의 버튼이 차지하는 부분을 나타낸다.

[0331] 'Media' 객체(464)는 메뉴 시스템에서 참조된 미디어 정보를 나타낸다. 'Media' 객체는 하나의 'MenuTrack' 객체(486) 및 이와 관련된 다수의 'Title' 객체(487)을 포함한다. 'MenuTracks' 객체(486)는 메뉴(즉, 배경 오디오, 배경 비디오 및 오버레이)를 구성하기 위해 이용되는 미디어를 나타내는 'MediaTrack' 객체(488)를 참조한다.

[0332] 'Title' 객체(487)는 멀티미디어 재현을 나타내며, 다수의 'Chapter' 객체(492) 및 이와 관련된 'MediaSource' 객체(490)를 구비한다. 'Title' 객체는 'TranslationLookup' 객체(494)에 대한 참조를 포함한다. 'Chapter' 객체는 멀티미디어 재현에서의 소정의 포인트를 나타내며, 이와 관련된 다수의 'MediaTrack' 객체(488)를 구비한다.

[0333] 'Chapter' 객체는 'TranslationLookup' 객체(494)에 대한 참조를 포함한다. 'Chapter' 객체와 관련된 각각의 'MediaTrack' 객체는 멀티미디어 재현의 오디오, 비디오 또는 서브타이틀 트랙의 포인트를 나타내고,

'MediaSource' 객체(490) 및 'TranslationLookup' 객체(494)를 참조한다(이하 설명함).

[0334] 'TranslationTable' 객체(465)는 'Title' 객체, 'Chapter' 객체 및 'Mediatrack' 객체에 의해 지시되는 멀티미디어 재현의 다양한 부분을 기술하는 다수의 문자열을 그룹화한다. 'TranslationTable' 객체(465)는 이와 관련된 다수의 'TranslationLookup' 객체(494)를 구비한다. 각각의 'TranslationLookup' 객체는 특정의 객체를 나타내며, 이와 관련된 다수의 'Translation' 객체(496)를 구비한다. 'Translation' 객체는 'TranslationLookup' 객체에 의해 지시되는 객체를 특정의 언어로 기술하는 문자열을 각각 나타낸다.

[0335] 상기 설명한 다양한 객체를 생성하고, 필요한 연관성 및 객체간의 참조를 설정하도록 구성된 소프트웨어를 이용하여 미디어 객체 모델이 구성될 수 있다.

[0336] 3.3.2

[0337] XML 파일의 생성

[0338] XML 구성 파일은 메뉴와 이들의 부분 구성요소 모두를 나타내는 메뉴 모델로부터 생성된다. XML 구성 파일은 메뉴에 의해 사용되는 모든 미디어 파일을 식별한다. XML은 객체 모델을 파싱 처리하여 XML 코드화하는 적절한 파서 애플리케이션을 수행시킴으로써 생성될 수 있다.

[0339] 다른 실시예에서, 비디오 편집 애플리케이션에 의해 제공되는 사용자 인터페이스에 의해, 사용자는 메뉴 모델을 작성하지 않고 XML 구성 파일을 직접 생성할 수 있다.

[0340] 다른 메뉴 시스템이 DVD 메뉴 등의 메뉴 모델의 기본이 되는 실시예에서, 메뉴는 본 발명의 실행에 따라 생성된 멀티미디어 파일에 포함되지 않는 컨텐츠에 관련된 메뉴 옵션을 제거하기 위해 사용자에 의해 제거될 수 있다. 일실시예에서, 이러한 제거는 메뉴 모델로부터 객체를 제거할 수 있도록 해주는 그래픽 사용자 인터페이스를 제공함으로써 이루어질 수 있다. 다른 실시예에서, 메뉴를 제거하는 것은, XML 구성 파일을 편집할 수 있는 텍스트 인터페이스 또는 그래픽 사용자 인터페이스를 제공함으로써 달성될 수 있다.

[0341] 3.3.3 미디어 정보

[0342] 'DMNU' 청크가 생성되면, 메뉴 생성기(450)에 제공된 미디어 정보는, 메뉴 모델에서 특정된 버튼에 대한 배경 비디오, 배경 오디오 및 전경 오버레이를 제공하도록 요구된 데이터를 포함한다(이하의 설명 참조). 일실시예에서, 캘리포니아 산타클라라에 소재하는 Roxio, Inc.에서 만든 Video Wave 등의 비디오 편집 애플리케이션은, 각각의 메뉴에 대해 비디오, 오디오 및 버튼 선택 오버레이를 표현하는 소스 미디어 트랙을 제공하는데 이용된다.

[0343] 3.3.4 중간 AVI 파일의 생성

[0344] 상술한 바와 같이, 배경 비디오, 배경 오디오 및 전경 버튼 오버레이로서 이용되는 미디어 트랙은 하나 이상의 메뉴에 대해 단일의 AVI 파일에 저장된다. 메뉴 AVI 파일에 미디어 트랙을 포함하는 청크는 비디오, 오디오 및 버튼 오버레이 트랙을 인터리브하도록 설계된 소프트웨어를 이용하여 작성될 수 있다. 'audio', 'video' 및 'overlay' 청크(즉, 오버레이 정보를 포함하는 'subtitle' 청크)는 인터리버(interleaver)를 이용하여 AVI 포맷에 호환가능한 파일로 인터리브된다.

[0345] 상술한 바와 같이, 각각의 메뉴에 대해 개별 AVI 파일이 작성될 수 있다. 다른 실시예에서, 배경 오디오, 배경 비디오 및 전경 오버레이 정보를 제공하기 위해 이용되는 미디어 정보를 포함하기 위하여, 다른 파일 포맷이나 단일의 파일이 이용될 수 있다.

[0346] 3.3.5 XML 구성 파일 및 AVI 파일의 조합

[0347] 일실시예에서, 컴퓨터는 XML 구성 파일로부터의 정보를 파싱 처리하여 'WowMenu' 청크(이하 설명됨)를 작성하도록 구성된다. 또한, 컴퓨터는 각각의 메뉴에 대해 미디어를 포함하는 AVI 파일을 이용하여 'MRIF' 청크(이하 설명됨)를 작성할 수 있다. 컴퓨터는 'MRIF' 청크에서의 미디어 청크와 'WowMenu' 청크 사이에서 필요한 참조를 작성함으로써 'DMNU' 청크의 생성을 완료할 수 있다. 몇몇 실시예에서, 메뉴 정보는 암호화될 수 있다. 암호화는 'video' 청크와 관련하여 아래 설명하는 것과 유사한 방식으로 'MRIF' 청크에 포함된 미디어 정보를 암호화함으로써 달성될 수 있다. 다른 실시예에서는 다른 암호화 기법이 이용될 수 있다.

[0348] 3.3.6 객체 모델로부터 메뉴의 자동 생성

[0349] 도 3d을 보면, 풀 메뉴보다 컨텐츠가 적은 메뉴는 'Media' 객체(464)와 관련된 'Title' 객체(487)를 단순히 조

사함으로써 메뉴 모델로부터 자동으로 생성될 수 있다. 본 발명의 실시예에 따라 메뉴를 자동을 생성하는데 이용되는 객체가 도 3da에 도시되어 있다. 소프트웨어는 이용할 오디오 및 서브타이틀 트랙의 선택 및 멀티미디어 제현의 특정의 선택을 가능하게 하는 단순한 메뉴에 대해 XML 구성 파일을 생성할 수 있다.

[0350] 3.3.7 단일의 구성 파일을 이용한 'DXDT' 및 'DMNU' 청크의 생성

[0351] 본 발명의 몇몇 실시예에 따른 시스템은 'DXDT' 및 'DMNU' 청크를 생성하기 위해 XML 파일을 이용하고 'meta data' 및 메뉴 정보를 포함하는 단일의 XML 구성 파일을 생성할 수 있다. 이들 시스템은 메뉴 객체 모델과 'meta data' 정보를 이용하여 XML 구성 파일을 유도한다. 다른 실시예에서, 구성 파일은 XML이 아니어도 된다.

[0352] 3.4 'audio' 청크의 생성

[0353] 본 발명의 실시예에 따른 멀티미디어 파일의 'movi' 리스트 청크에서의 'audio' 청크는 오디오 소스를 복호화하고 그 소스를 본 발명에 따라 'audio' 청크로 부호화함으로써 생성될 수 있다. 일실시예에서, 'audio' 청크는 mp3 코덱을 이용하여 부호화될 수도 있다.

[0354] 3.4.1 오디오 리청크

[0355] 대응하는 'video' 청크의 컨텐츠에 대응하는 오디오 정보를 포함하지 않는 청크에 오디오 소스가 제공되는 경우, 본 발명의 실시예는 오디오를 리청크(re-chunk)할 수 있다. 오디오를 리청크하는데 이용될 수 있는 프로세스를 도 3e에 나타낸다. 이 프로세스(480)는 'video' 청크의 식별(482), 'video' 청크와 동반하는 오디오 정보의 식별(484) 및 새로운 'audio' 청크를 작성(488)하기 위하여 기존의 오디오 청크로부터 오디오 정보를 추출(486)하는 과정을 포함한다. 이 프로세스는 모든 오디오 소스가 리청크되었다는 판정(490)이 이루어질 때까지 반복된다. 이 시점에서, 오디오는 리청크 과정이 완료(492)된다.

[0356] 3.5 'video' 청크의 생성

[0357] 상술한 바와 같이, 비디오 청크를 작성하는 프로세서는 비디오 소스를 복호화하고, 복호화된 비디오를 'video' 청크로 부호화하는 과정을 포함할 수 있다. 일실시예에서, 각각의 'video' 청크는 단일의 비디오 프레임에 대한 정보를 포함한다. 복호화 프로세스는 비디오를 특정의 포맷으로 포맷화하고 압축해제될 수 있는 표준 비디오 포맷으로 포맷화되는 비디오를 복호화하는 과정을 포함한다. 부호화 과정은 표준 비디오를 수신하고, 이 비디오를 부호화하며, 부호화된 비디오를 이용하여 'video' 청크를 생성하는 과정을 포함한다.

[0358] 본 발명의 실시예에 따른 비디오 인코더는 도 3f에 개념적으로 도시되어 있다. 비디오 인코더(500)는 표준 비디오 정보(504)를 전처리(502)한다. 전처리된 비디오에 움직임 보상(508)을 제공하기 위하여 전처리된 비디오에 대해 움직임 추정(506)이 수행된다. 움직임 보상된 비디오에 대해 이산 코사인 변환(DCT 변환)(510)이 수행된다. DCT 변환 이후에, 비디오는 양자화되고(512), 예측(514)이 수행된다. 움직임 추정의 결과를 이용하여 생성된 움직임 코딩(520)과 비디오의 텍스트 코딩된(518) 버전을 조합함으로써, 압축된 비트스트림(516)이 생성된다. 압축된 비트스트림은 'video' 청크를 생성하는데 이용된다.

[0359] 움직임 추정(506)을 수행하기 위하여, 시스템은 복호화 장치에 의해 복호화될 비디오의 사전에 어떻게 프레임을 처리했는지를 알고 있어야 한다(예컨대, 압축된 비디오를 플레이어가 보려고 압축해제한 경우). 이 정보는 양자화기(512)의 출력을 역양자화(522)함으로써 얻어질 수 있다. 역 DCT(542)는 역 양자화기의 출력에 대해 수행될 수 있다. 움직임 추정 프로세서 동안의 액세스를 위해 그 결과가 프레임 단위로 저장될 수 있다.

[0360] 본 발명의 실시예에 따른 멀티미디어 파일은 다수의 사이코비주얼 인핸스먼트(528)를 포함할 수 있다. 사이코비주얼 인핸스먼트는 인간의 지각에 기초하여 비디오를 압축하는 방식이다. 이러한 기술은, 이하 설명하며, 비디오의 다양한 특징을 표현하기 위해 양자화기에 의해 이용되는 비트의 수를 변경하는 과정을 포함하는 것이 일반적이다. 부호화 프로세스의 다른 특징도 사이코비주얼 인핸스먼트를 포함할 수 있다.

[0361] 일실시예에서, 모든 부호화 시스템(500)은 상기 설명한 다양한 기능을 수행하기 위하여 구성된 컴퓨터를 이용하여 구현될 수 있다. 이들 기능들의 상세한 구현의 예를 이하 설명한다.

[0362] 3.5.1 전처리

[0363] 본 발명의 실시예에 따라 인코더(500)에 의해 임의 선택적으로 수행될 수 있는 전처리 동작(502)은 부호화된 비디오의 품질을 향상시키기 위하여 다수의 신호 처리 기술을 포함할 수 있다. 일실시예에서, 전처리(502)는 디인터레이싱(deinterlacing), 시간/공간 노이즈 감소 및 리사이징(resizing) 중 하나 또는 모두를 포함할 수 있다. 이들 전처리 기술 중 3가지 모두가 사용되는 실시예에 있어서, 디인터레이싱은 시간/공간 노이즈 감소 및

리사이징에 후속해서 수행되는 것이 일반적이다.

[0364] 3.5.2 움직임 추청 및 보상

[0365] 본 발명의 실시예에 따른 비디오 인코더는 다수의 프레임에서 반복되는 픽셀에 대한 검색에 의해 비디오 트랙을 표현하는데 요구되는 픽셀의 수를 감소시킬 수 있다. 필수적으로, 비디오에서의 각각의 프레임은 이전에 가졌던 것과 동일한 수의 픽셀을 포함하는 것이 일반적이다. 인코더는 각 프레임 사이에서 픽셀의 매칭(예컨대, 매크로블록, 픽셀, 하프픽셀, 쿼터픽셀)에 대한 몇 가지 타입의 검색을 수행할 수 있으며, 이미지 품질의 감소 없이 이러한 용장성을 제거할 수 있다. 움직임 추청을 이용하여, 인코더는 모든 프레임에 대해 전체 화면을 저장하지 않고 최후의 프레임 이후에 발생된 변화를 기록함으로써 대부분의 화면을 표현할 수 있다. 움직임 추정 동안, 인코더는 프레임을 분할하고 해석하여 균일한 그리드의 블록(매크로블록이라고도 함)을 생성한다. 프레임 내의 각각의 '매크로블록'에 대해, 인코더는 이전 프레임에서의 매칭 블록을 발견하고자 할 수 있다. 매칭 블록을 찾는 과정을 '움직임 검색'이라고 한다. '매크로블록'의 움직임은 2차원 벡터, 즉 (x, y) 표현으로 표시될 수 있다. 움직임 검색 알고리즘은 다양한 정확도로 수행될 수 있다. 홀플(whole-ple) 검색은 인코더가 일정 시간에 1차원 픽셀에서의 참조 프레임을 통해 스테핑함으로써 매칭 블록을 찾는 것이다. 하프픽셀 검색에 있어서, 인코더는 참조 프레임을 통해 일정 시간에 픽셀의 절반에 의한 차원을 스테핑함으로써 매칭 블록을 찾는다. 인코더는 픽셀보다 큰 입도를 포함하는 쿼터픽셀, 다른 픽셀 프랙션이나 검색을 이용할 수 있다.

[0366] 도 3f에 도시된 인코더 실시예는 본 발명의 실시예에 따라 움직임 추정을 수행한다. 움직임 추정 동안, 인코더는 프레임 기억 장치(526)에 저장되는, 이전 프레임과 전처리된 비디오(502)에 대한 액세스를 갖는다. 이전 프레임은 양자화하기의 출력을 수신하고, 역 양자화(522)와 역 DCT 변환(524)을 수행함으로써 생성된다. 역 기능을 수행하는 이유는 프레임 기억 장치 내의 프레임이 본 발명의 실시예에 따라 플레이어에 의해 복호화될 때 나타나는 것처럼 하기 위해서이다.

[0367] 움직임 보상은 움직임 추정의 결과로서 생성된 블록과 벡터를 가짐으로써 수행된다. 그 결과는 추가의 텍스트 정보를 제공함으로써 실제의 이미지에 대해 매칭될 수 있는 부호화된 이미지의 근사값이다.

[0368] 3.5.3 이산 코사인 변환

[0369] 도 3f에 도시된 인토더에 의해 수행되는 DCT 및 역DCT는 ISO/IEC 14496-2:2001(E), 부록 A.1(코딩 변환)에서 특정한 표준에 따라 이루어진다.

[0370] 3.5.3.1 변환의 설명

[0371] DCT는 주파수 영역 표현에 대한 공간 영역 데이터 포인트의 세트를 변환하는 방법이다. 비디오 압축의 경우, 2차원 DCT는 이미지 블록을 용장성이 용이하게 이용되는 형태로 변환한다. 주파수 영역 블록은 엔트로피 코딩에 의해 용이하게 압축되는 희소 행렬(sparse matrix)이 될 수 있다.

[0372] 3.5.3.2 변환에 대한 사이코비주얼 인핸스먼트

[0373] DCT 계수는, 보는 사람에 대해 용이하게 보이는 영역에서의 양자화 노이즈를 감소시킴으로써 양자화된 이미지의 화질을 개선하기 위해 변경될 수 있다. 또한, 파일 사이즈는 보는 사람에 의해 용이하게 분간이 가능하지 않은 이미지 부분에서의 양자화 노이즈를 증가시킴으로써 감소될 수 있다.

[0374] 본 발명의 실시예에 따른 인코더는 'slow'(슬로우) 사이코비주얼 인핸스먼트로 불리는 기능을 수행할 수 있다. '슬로우' 사이코비주얼 인핸스먼트는 비디오 이미지의 블록을 분석하고 비디오의 외양을 열화시키지 않고 몇몇 비트를 세이브시킬 수 있는 몇몇 노이즈를 결정한다. 이 프로세스는 블록당 하나의 메트릭스를 이용한다. 이 프로세스를 '슬로우' 프로세스라고 하고, 이는 블로킹 또는 링잉 현상(ringing artifacts)를 피하기 위해 상당한 양의 연산을 수행하기 때문이다.

[0375] 본 발명의 실시에에 따른 인코더의 다른 예는, 'fast'(패스트) 사이코비주얼 인핸스먼트를 수행한다. '패스트' 사이코비주얼 인핸스먼트는 블록 내에 보이는 노이즈를 제어할 수 있으며 양자화 노이즈를 정형화할 수 있다.

[0376] '슬로우' 및 '패스트' 사이코비주얼 인핸스먼트에 대해서는 이하 더 상세히 설명한다. 다른 사이코비주얼 인핸스먼트는 이미지 에지에서의 노이즈를 제거하는 것, 사람의 눈에 명백하게 보이지 않는 이미지 영역에서의 양자화 노이즈의 더 높은 레벨을 집중시키도록 하는 본 발명의 실시예에 따라 수행될 수 있다.

[0377] 3.5.3.3 '슬로우' 사이코비주얼 인핸스먼트

[0378] '슬로우' 사이코비주얼 인핸스먼트는, 비디오 이미지의 블록을 분석하고, 비디오의 외양을 열화시키지 않는 몇몇 노이즈가 비트를 세이브시킬 수 있는지 여부를 판정한다. 일실시예에서, 이 알고리즘은 2가지 단계를 포함한다. 첫 번째 단계는 입력 휘도 픽셀에 대해 식별된 이미지의 생성을 포함한다. 식별된 이미지는 이하의 방식으로 생성된다. 두 번째 단계는 양자화 이전에 DCT 계수를 변경하는 단계를 포함한다.

[0379] 3.5.3.3.1 식별된 이미지의 생성

[0380] 식별된 이미지의 각각의 픽셀 P'_{xy}는 이하에 따라 압축해제된 소스 픽셀 Pxy로부터 연산된다.

[0381] $$p'_{xy} = \max\left(\left|p_{x+1y} - p_{xy}\right|, \left|p_{x-1y} - p_{xy}\right|, \left|p_{xy+1} - p_{xy}\right|, \left|p_{xy-1} - p_{xy}\right|\right)$$

[0382] 여기서, P'_{xy}는 0 에서 255까지의 범위(8 비트 비디오라고 가정함)를 가질 것이다.

[0383] 3.5.3.2 DCT 계수의 변경

[0384] DCT 계수의 변경은 블록 링잉 팩터의 연산, 블록 에너지의 연산 및 계수 값의 실질적인 변경을 포함할 수 있다.

[0385] 3.5.3.3.3 블록 링잉 팩터(block ringing factor)

[0386] 이미지의 각각의 블록에 대해, 식별된 이미지의 로컬 영역에 기초하여 "링잉 팩터"(ringing factor)가 계산된다. 블록이 8x8 블록으로 규정된 실시예인 경우, 링잉 팩터는 다음과 같은 방법을 이용하여 결정될 수 있다.

[0387] 처음에, 8x8 블록 이내의 최대 휘도 및 최소 휘도 픽셀에 기초하여 임계값이 결정된다.

[0388] $$threshold_{block} = floor((\max_{block} - \min_{block})/8) + 2$$

[0389] 식별된 이미지와 임계값은 블록의 이웃에 있는 "플랫"(flat) 픽셀의 맵을 생성하는데 이용될 수 있다. 상이한 임계값을 갖는 각 블록의 가능성 때문에 전체 프레임에 대한 플랫 픽셀의 맵의 작성을 방지한다. 이 맵은 다음과 같이 생성된다.

[0390] $P'_{xy} < threshold_{block}$ 이면, $flat_{xy} = 1$

[0391] 그외의 경우에는 $flat_{xy} = 0$

[0392] 플랫 픽셀의 맵은 간단한 논리 연산에 따라 필터링된다.

[0393] $flat_{xy}=1$, $flat_{x-1y}=1$, $flat_{xy-1}=1$, $flat_{x-1y-1}=1$ 이면, $flat'_{xy} = 1$

[0394] 그외의 경우에는 $flat'_{xy}$

[0395] 필터링된 맵에서의 플랫 픽셀은 8x8 블록에 차지하는 9x9 블록에 대해 카운트된다.

[0396] $$flatcount_{block} = \sum flat'_{xy} \quad \text{for} \quad 0 = x = 8 \ \text{and} \ 0 = y = 8$$

[0397] 가시적 링잉 현상의 위험은 다음 식을 이용하여 평가될 수 있다.

[0398] $$ringingbrisk_{block} = ((flatcount_{block} - 10) \times 256 + 20)/40$$

[0399] 8x8 블록의 링잉 팩터는 다음 식을 이용하여 유도될 수 있다.

[0400] ringingrisk > 255 이면 Ringingfactor =0

[0401] ringingrisk < 0 이면 Ringingfactor = 255

[0402] 그외의 경우에는 Ringingfactor = 255 - ringingrisk

[0403] 3.5.3.3.4 블록 에너지의 연산

[0404] 이미지의 블록에 대한 에너지는 다음의 과정에 의해 산출될 수 있다. 몇몇 실시예에서, 이미지의 8x8 블록이 이용된다.

[0405] 소스 이미지에 대해 순방향 DCT가 수행된다.

[0406] $$T = fDCT(S)$$

[0407] 여기서, S는 관련된 8x8 블록의 64개의 소스 이미지 휘도 값이며, T는 소스 이미지의 동일 부분의 변환된 버전이다.

[0408] 특정한 계수 위치에서의 에너지는 계수 값을 제곱계산한 것으로 규정된다.

[0409] $0 = k = 63$인 경우 $e_k = t_k^2$

[0410] 여기서, t_k는 변환된 블록 T의 k번째 계수

[0411] 3.5.3.3.5 계수 수정

[0412] DCT 계수의 수정은 이하의 과정에 따라 본 발명의 실시예에서 수행될 수 있다. 몇몇 실시예에서, 이 과정은 양자화 이전에 논제로(non-zero) AC DCT 계수에 대해 수행된다. 각각의 계수의 크기는 작은 델타에 의해 변경되고, 이 델타의 값은 사이코비주얼 기술에 따라 결정된다.

[0413] 각 논제로 AC 계수 c_k의 DCT 계수 수정은 다음 공식을 이용하여 로컬 및 블록 에너지에 기초하여 에너지를 계산함으로써 수행된다.

[0414] $$energy_k = \max(a_k \times e_k, 0.12 \times totalenergy)$$

[0415] 여기서, a_k는 이하의 표에 기재되어 있는 것과 같이 계수 위치에 기초하는 값을 갖는 상수이다.

[0416] 표 3

계수 테이블

0.0	1.0	1.5	2.0	2.0	2.0	2.0	2.0
1.0	1.5	2.0	2.0	2.0	2.0	2.0	2.0
1.5	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0

[0417] 에너지는 다음 관계식을 이용하여 블록의 링잉 팩터에 따라 수정될 수 있다.

[0418] $$energy'_k = ringingfactor \times energy_k$$

[0419] 결과 값은 조사 테이블(LUT)에 입력으로 사용되기 전에 시프트 및 클립 처리된다.

[0420] $$e_k = \min(1023, 4 \times energy'_k)$$
$$d_k = LUT_i \quad \text{where} \quad i = e_k$$

[0421] '오프셋'(offset) 값은 이하 표에 개시된 바와 같이 양자화기 Q_p에 의존한다.

표 4

[0422] Q_p 값의 함수로서의 offset

Qp	offset	Qp	offset
1	0.5	16	8.5
2	1.5	17	7.5
3	1.0	18	9.5
4	2.5	19	8.5
5	1.5	20	10.5
6	3.5	21	9.5
7	2.5	22	11.5
8	4.5	23	10.5
9	3.5	24	12.5
10	5.5	25	11.5
11	4.5	26	13.5
12	6.5	27	12.5
13	5.5	28	14.5
14	7.5	29	13.5
15	6.5	30	15.5
		31	14.5

[0423] 변수 $k_{texture}$ 및 k_{flat}는 플랫하고 텍스처가 있는 영역에서의 사이코비주얼 효과에 대한 강도를 제어한다. 일실시예에서, 0과 1 사이의 범위(0은 효과가 없으며, 1은 풀 효과를 의미함)를 갖는다. 일실시예에서, 변수 $k_{texture}$ 및 k_{flat}는 다음과 같이 설정된다.

[0424] 휘도

$$k_{texture} = 1.0$$
$$k_{flat} = 1.0$$

[0425]

[0426] 크로미넌스

$$k_{texture} = 1.0$$
$$k_{flat} = 0.0$$

[0427]

[0428] 조사 테이블로부터의 출력(d_k)은 부가적인 처리에 의해 DCT 계수의 크기를 수정하는데 이용된다.

$$c'_k = c_k - \min(d_k, |c_k|) \times \mathrm{sgn}(c_k)$$

[0429]

[0430] 마지막으로, DCT 계수 c_k는 수정된 계수 c'_k로 대체되고 양자화를 위한 진행을 계속한다.

[0431] 3.5.3.4 '패스트' 사이코비주얼 인핸스먼트

[0432] '패스트' 사이코비주얼 인핸스먼트는 입력 휘도 픽셀에 대해 '중요한' 맵을 계산하고 DCT 계수를 수정함으로써 DCT 계수에 대해 수행된다.

[0433] 3.5.3.4.1 '중요한' 맵의 계산

[0434] '중요한' 맵은 입력 비디오 프레임의 휘도 공간에서의 각각의 픽셀에 대한 '중요한' 값을 계산함으로써 생성될 수 있다. 몇몇 실시예에서, '중요한' 값은 특정의 픽셀에 위치하는 임의의 왜곡에 대한 인간 눈의 감도를 근사한다. 이 '중요한' 맵은 픽셀의 '중요한' 값의 어레이이다.

[0435] 이 '중요한' 픽셀은 첫 번째로 픽셀(d_{xy}) 주위의 픽셀의 블록의 동적 범위를 계산함으로써 결정된다. 몇몇 실시예에서, 픽셀 위치(x, y)에 대해 중심에 있는 픽셀의 3x3 블록의 동적 범위는 그 영역에서 가장 밝은 값에서 가장 어두운 픽셀의 값을 감산함으로써 산출된다.

[0436] '중요한' 픽셀(m_{xy})은 다음과 같이 픽셀의 동적 범위로부터 유도될 수 있다.

[0437] $$m_{xy} = 0.08 / \max(d_{xy}, 3) + 0.001$$

[0438] 3.5.3.4.2 DCT 계수의 수정

[0439] 일실시예에서, DCT 계수의 수정은 기본 기능 에너지 매트릭스와 델타 조사 테이블의 생성을 포함한다.

[0440] 3.5.3.4.3 기본 기능 에너지 매트릭스의 생성

[0441] 기본 기능 에너지 메트릭스의 세트는 DCT 계수를 수정하는데 이용될 수 있다. 이들 매트릭스는 부호화 이전에 산출될 수 있는 상수 값을 포함한다. 8x8 매트릭스는 64개의 DCT 기본 기능의 각각에 이용된다. 각의 매트릭스는, 8x8 블록에서의 모든 픽셀이 대응하는 계수의 수정에 의해 어떻게 영향을 받는지를 개시한다. k번째 기본 기능 에너지 매트릭스는 8x8 매트릭스 A_k를 취함으로써 유도되는데, 대응하는 계수는 100으로 설정되고 다른 계수는 0으로 설정된다.

[0442] n = k 이면 a_{kn} = 100 이고,

[0443] 그외의 경우는 a_{kn} = 0 이다.

[0444] 여기서, n은 8x8 매트릭스 내의 계수 위치를 나타낸다: 0 = n = 63.

[0445] 역 DCT는 추가의 8x8 매트릭스 A'_k를 제공하기 위하여 매트릭스에 대해 수행된다. 매트릭스(a'_{kn})의 요소는 k번째 기본 기능을 나타낸다.

[0446] $$A'_k = iDCT(A_k)$$

[0447] 변환된 매트릭스에서의 각각의 값은 제곱계산된다.

[0448] 0 = n = 63 인 경우에 대해 $b_{kn} = a'^{2}_{kn}$

[0449] 이 과정은 기본 기능 에너지 매트릭스 B_k를 생성하기 위해 64번 수행되며(0 = n = 63), 각각 64번의 자연 값을 구비한다. 각각의 매트릭스 값은 8x8 블록에서의 n번째 위치에서의 픽셀이 임의의 에러 또는 계수 k의 수정에 의해 얼마나 많이 영향을 받는지에 대한 측정값이다.

[0450] 3.5.3.4.4 델타 조사 테이블의 생성

[0451] 조사 테이블(LUT)은 계수 수정 델타의 계산을 촉진하는데 이용될 수 있다. 테이블의 내용은 '패스트' 사이코비주얼 인핸스먼트와 양자화 파라미터(Q_p)의 소망하는 강도에 의존하는 방식으로 생성될 수 있다.

[0452] 조사 테이블의 값은 다음 관계식에 따라 생성될 수 있다.

[0453] $$LUT_i = \min\left(floor\left(128 \times k_{texture} \times strength / (i + 0.5) + k_{flat} \times offset + 0.5\right), 2 \times Q_p\right)$$

[0454] 여기서, i는 테이블 내의 위치, 0 = i = 1023.

[0455] *strength*와 *offset*은 이하의 표에서 개시하는 바와 같이 양자화기 Q_p에 의존한다.

표 5

[0456] *strength*및 *offset*의 값과 Q_p의 값 간의 관계

Q_p	*strength*	*offset*	Q_p	*Strength*	*offset*
1	0.2	-0.5	16	2.0	8.5
2	0.6	1.5	17	2.0	7.5
3	1.0	1.0	18	2.0	9.5
4	1.2	2.5	19	2.0	8.5
5	1.3	1.5	20	2.0	10.5
6	1.4	3.5	21	2.0	9.5
7	1.6	2.5	22	2.0	11.5
8	1.8	4.5	23	2.0	10.5
9	2.0	3.5	24	2.0	12.5
10	2.0	5.5	25	2.0	11.5
11	2.0	4.5	26	2.0	13.5
12	2.0	6.5	27	2.0	12.5
13	2.0	5.5	28	2.0	14.5
14	2.0	7.5	29	2.0	13.5
15	2.0	6.5	30	2.0	15.5
			31	2.0	14.5

[0457] 변수 $k_{texture}$ 및 k_{flat}는 플랫하고 텍스처가 있는 영역에서의 사이코비주얼 효과에 대한 강도를 제어한다. 일실시예에서, 0과 1 사이의 범위(0은 효과가 없으며, 1은 풀 효과를 의미함)를 갖는다. 일실시예에서, 변수 $k_{texture}$ 및 k_{flat}는 다음과 같이 설정된다.

[0458] 휘도

[0459]
$$k_{texture} = 1.0$$
$$k_{flat} = 1.0$$

[0460] 크로미넌스

[0461]
$$k_{texture} = 1.0$$
$$k_{flat} = 0.0$$

[0462] 3.5.3.4.5 DCT 계수의 수정

[0463] DCT 계수는 상기 산출한 값을 이용하여 수정될 수 있다. 일실시예에서, 각각의 논제로 AC DCT 계수는 양자화 이전에 이하의 과정에 따라 수정된다.

[0464] 처음에, 'energy'(에너지) 값(e_k)은 대응하는 기본 기능 에너지 매트릭스와 중요한 맵으로부터의 적절한 8x8 블록의 도토 곱(dot product)을 취함으로써 산출된다. 이 '에너지'는 특정 계수에서의 양자화 에러가 인간에 의해 얼마나 인지될 수 있는지를 나타내는 측정값이다. 픽셀 중요도와 픽셀 기본 기능 에너지의 곱의 합이다.

[0465]
$$e_k = M \cdot B_k$$

[0466] 여기서, M은 8x8 블록의 중요 맵의 값을 포함하며,

[0467] B_k는 k번째 기본 기능 에너지 매트릭스이다.

[0468] 그 결과로서의 '에너지' 값은 인덱스(d_k)로서 이용되기 전에 델타 조사 테이블에 시프트 및 클립 처리된다.

[0469]
$$e'_k = \min[1023, floor(e_k / 32768)]$$
$$d_k = LUT_i$$

[0470] 여기서, $i=e'_k$

[0471] 델타 조사 테이블의 출력은 추가의 처리과정에 의해 DCT 계수의 크기를 수정하는데 이용된다.

[0472] $$c'_k = c_k - \min(d_k, |c_k|) \times sign(c_k)$$

[0473] DCT 계수 c_k는 수정된 c'_k로 대체되고, 양자화를 위해 진행된다.

[0474] 3.5.4 양자화

[0475] 본 발명의 실시예에 따른 인코더는 1996년 International Telecommunication Union as Video Coding for Low Bitrate Communication, ITU-T Rcommendation H.263에 의해 규정된 양자화기와 같은 표준 양자화기를 이용할 수 있다.

[0476] 3.5.4.1 양자화에 대한 사이코비주얼 인핸스먼트

[0477] 본 발명의 실시예에 따른 인코더의 일부는 더 효과적인 압축을 얻기 위하여 인간 시각의 심리적 효과를 이용하는 사이코비주얼 인핸스먼트 기능을 이용한다. 심리 시각적 효과는 프레임 단위와 매크로블록 단위에 적용될 수 있다.

[0478] 3.5.4.2 프레임 단위의 사이코비주얼 인핸스먼트

[0479] 프레임 단위로 적용될 때, 사이코비주얼 인핸스먼트는 레이트 제어(rate control) 알고리즘의 일부이며 그 목적은 부호화를 조정하는 것이고 주어진 양의 비트 레이트는 인간 시각이 인지할 수 있는 최대의 가시적 질을 보장하는데 최적으로 이용된다. 프레임 레이트 사이코비주얼 인핸스먼트는, 액션이 클 때 인간 시각이 미세 부분을 무시한다는 원리와, 이미지가 정적일 때 인간 시각이 미세 부분을 주목한다는 원리에 의해 촉진된다. 일실시예에서, 움직임의 양은 프레임에 대해 절대 차(absolute difference: SAD)의 합을 조사함으로써 결정된다. 일실시예에서, SAD 값은 2개의 블록의 함께 있는 휘도 픽셀의 절대 차를 가산함으로써 결정된다. 몇몇 실시예에서, 16x16 픽셀 블록의 절대 차가 이용된다. 일부분의 픽셀 오프셋을 취급하는 실시예에서, 절대 차의 합이 계산되기 전에, MPEG-4 표준(ISO/IEC의 동화상 전문가 그룹에서 개발한 ISO/IEC 표준)에서 규정한 보간(interpolation)이 수행된다.

[0480] 프레임 단위의 사이코비주얼 인핸스먼트는 비디오 트랙의 P 프레임에만 적용되며 그 프레임의 SAD 값에 기초한다. 부호화 동안, 사이코비주얼 모듈은 비디오 트랙의 모든 P 프레임의 평균 SAD(즉, $\overline{SAD}$)의 기록을 유지하며, 그 전체 SAD(즉, $\overline{DSAD}$)로부터 각 프레임의 SAD의 평균 거리의 기록을 유지한다. 평균화는 지수 이동 평균 알고리즘을 이용하여 이루어질 수 있다. 일실시예에서, 상술한 단일 패스 레이트 제어 알고리즘은 여기서 평균화 기간으로서 이용된다(상기 설명 참조).

[0481] 부호화된 비디오 트랙의 각각의 P 프레임에 대해, 프레임 양자화기 Q(레이즈 제어 모듈로부터 획득됨)는 적용되는 사이코비주얼 보정을 갖는다. 일실시예에서, 이 과정은 이하의 공식을 이용하여 레이트 R을 연산하는 과정을 포함한다.

[0482] $$R = \frac{SAD - \overline{SAD}}{\overline{DSAD}} - I$$

[0483] 여기서, I는 상수로서 계속 0.5로 설정되어 있다. R은 [1, -1]의 바운드 내로 클립 처리된다.

[0484] 양자화기는 이하 나타낸 수식을 통해 레이트 R에 따라 조정된다.

[0485] $$Q_{adj} = Q\lfloor Q.(1 + R.S_{frame})\rfloor$$

[0486] 여기서, Sframe는 프레임 단위의 사이코비주얼 인핸스먼트에 대한 강도 상수이다.

[0487] Sframe 상수는 프레임 단위의 사이코비주얼에 대해 조정이 얼마나 강하게 이루어질 수 있는지를 결정한다. 코

텍의 일실시예에서는, Sframe을 0.2, 0.3 또는 0.4로 설정하는 옵션을 이용할 수 있다.

[0488] 3.5.4.3 매크로블록 단위의 사이코비주얼 인핸스먼트

[0489] 매크로블록 단위로 사이코비주얼 인핸스먼트를 이용하는 본 발명의 실시예에 따른 인코더는 사용자에 대한 비디오의 가시적 품질에 대해 현저하고 이들 매크로블록을 더 높은 품질로 코드화하는 매크로블록을 식별한다. 매크로블록 단위의 사이코비주얼 인핸스먼트의 효과는 프레임의 덜 중요한 부분으로부터 비트를 제거하고 이들을 프레임의 중요한 부분에 적용하는 것이다. 몇몇 실시예에서, 인핸스먼트는 평활화, 밝기 및 매크로블록 SAD에 기초하는 기술을 이용하여 달성된다. 이들 기술을 단독으로 사용하거나 또는 다른 기술을 조합하거나 다른 기술만을 사용하는 다른 실시예가 가능하다.

[0490] 일실시예에서, 상기 설명한 이들 3가지 매크로블록 단위의 사이코비주얼 인핸스먼트는 공통의 파라미터 S_{MB}를 공유하는데, 이 파라미터는 매크로블록 단위의 사이코비주얼 인핸스먼트의 강도를 제어한다. 매크로블록에 대한 최대 및 최소 양자화기는 이하 나타낸 수식을 이용하여 프레임 양자화기 Q_{frame}와 강도 파라미터로부터 유도된다.

$$Q_{MBMax} = \frac{Q_{frame}}{(1 - S_{MB})}, \text{ and}$$
$$Q_{MBMin} = Q_{frame} \cdot (1 - S_{MB})$$

[0491]

[0492] 여기서, Q_{MBMax}는 최대 양자화기이며,

[0493] Q_{MBMin}은 최소 양자화기이다.

[0494] Q_{MBMax}와 Q_{MBMin}은 전체 프레임에 대한 매크로블록 양자화기의 상한 및 하한을 규정한다. 일실시예에서, 값 S_{MB}를 값 0.2, 0.3 및 0.4 중 어느 것으로 설정하는 옵션이 제공된다. 다른 실시예에서, S_{MB}의 다른 값을 이용할 수도 있다.

[0495] 3.5.4.3.1 밝기 인핸스먼트

[0496] 사이코비주얼 인핸스먼트가 매크로블록의 밝기에 기초하여 수행되는 실시예에서, 인코더의 더 밝은 매크로블록의 화질을 더 높이도록 부호화한다. 이러한 인핸스먼트의 이론적 원리는, 사용자가 프레임의 상대적으로 어두운 부분을 더 많이 또는 더 적게 무시한다는 것에 기초한다. 이러한 매크로블록 사이코비주얼 인핸스먼트는 비디오 트랙의 I 프레임과 P 프레임에 적용된다. 각 프레임에 대해, 인코더는 먼저 전체 프레임을 전체적으로 본다. 평균 밝기($\overline{BR}$)이 계산되고, 평균값($\overline{DBR}$)으로부터 밝기의 평균차가 계산된다. 이들 값은, 사이코비주얼 인핸스먼트가 적용되어야 하는지에 대한 지시기로서 사용될 수 있는 2개의 임계값($T_{BRLower}$, $T_{BRUpper}$)을 만드는데 이용된다.

$$T_{BRLower} = \overline{BR} - \overline{DBR}$$
$$T_{BRUpper} = \overline{BR} + \left(\overline{BR} - T_{BRLower}\right)$$

[0497]

[0498] 밝기 인핸스먼트는 매크로블록에 대한 의도된 양자화기(Q_{MB})를 생성하기 위해 아래 설명한 조건을 이용하여 2개의 임계값에 기초하여 적용된다.

$$Q_{MB} = Q_{MBMin} \text{ when } BR > T_{BRUpper}$$
$$Q_{MB} = Q_{frame} \text{ when } T_{BRLower} \le BR \le T_{BRUpper}, \text{ and}$$
$$Q_{MB} = Q_{MBMax} \text{ when } BR < T_{BRLower}$$

[0499]

[0500] 여기서, BR은 특정 매크로블록에 대한 밝기 값이다.

[0501] 인코더가 MPEG-4 표준에서 사용되는 실시예에서, 매크로블록 단위의 사이코비주얼 밝기 인핸스먼트 기술은 하나의 매크로블록에서 다음 매크로블록까지 ±2 이상 양자화기를 변경하지 않는다. 따라서, 계산된 Q_{MB}는 이전 매크로블록에서 이용된 양자화기에 기초한 변경을 요구할 수 있다.

[0502] 3.5.4.3.2 평활화 인핸스먼트

[0503] 평활화 사이코비주얼 인핸스먼트를 포함하는 본 발명의 실시예에 따른 인코더는, 부호화되는 이미지의 공간 편차에 기초하여 양자화기를 수정한다. 인간의 시각은 이미지의 평탄한 부분에서의 양자화 아티팩트(quantization artifacts)에 대해 민감도가 증가한다는 원리에 따라, 평활화 사이코비주얼 인핸스먼트의 이용이 촉진된다. 따라서, 평활화 사이코비주얼 인핸스먼트는 이미지의 더 평활한 부분을 표현하기 위해 비트의 수를 증가시키는 기술과, 이미지에서의 공간 편차가 높은 곳에서의 비트의 수를 감소시키는 기술을 포함한다.

[0504] 일실시예에서, 이미지의 일부의 평활화는 매크로블록($\overline{DR}$)의 밝기에 대한 매크로블록에서의 픽셀의 휘도의 평균 차로서 측정된다. 본 발명의 실시예에 따라 I 프레임에 대해 평활화 사이코비주얼 인핸스먼트를 수행하는 방법이 도 3g에 도시되어 있다. 프로세스(540)는 $\overline{DR}$을 계산(542)하기 위해 전체 프레임을 조사하는 과정을 포함한다. 평활화 인핸스먼트를 적용하기 위한 임계값 T_{DR}은 이하의 식을 이용하여 유도(544)될 수 있다.

[0505] $$T_{DR} = \frac{\overline{DR}}{2}$$

[0506] 이하의 평활화 인핸스먼트는 이 임계값에 기초하여 수행된다(546).

[0507] DR ≥ T_{DR}인 경우, $Q_{MB} = Q_{frame}$,

[0508] DR < T_{DR} 인 경우, $Q_{MB} = Q_{MBMin}$

[0509] 여기서, Q_{MB}는 매크로블록에 대한 의도된 양자화기이며, DR은 매크로블록에 대한 편차 값(즉, 평균 휘도 - 평균 밝기)이다.

[0510] MPEG-4 표준에 따라 파일을 부호화는 실시예는, 매크로블록 단위의 양자화기 변화가 하나의 매크로블록에서 다른 매크로블록으로 최대 ±2가 된다는 것에 제한된다.

[0511] 3.5.4.3.3 매크로블록 SAD 앤핸스먼트

[0512] 본 발명의 실시예에 따른 인코더는 매크로블록 SAD 사이코비주얼 인핸스먼트를 이용할 수 있다. 매크로블록 SAD 사이코비주얼 인핸스먼트는 정적 매크로블록에 대한 상세를 증가시키는데 이용될 수 있으며, 높은 동작 장면이 이용되는 프레임의 부분에서의 상세를 감소시키는데 이용될 수 있다.

[0513] 본 발명의 실시예에 따라 매크로블록 SAD 사이코비주얼 인핸스먼트를 수행하는 프로세스가 도 3h에 도시되어 있다. 이 프로세스(570)는 전체 프레임에서의 모든 매크로블록에 대하여 평균 SAD(즉, $\overline{MBSAD}$)를 결정하기 위해 I 프레임 전체를 조사(572)하는 과정을 포함하고, 평균(즉, $\overline{MBSAD}$)로부터 매크로의 SAD의 평균 편차를 얻는다. 일실시예에서, 이들 2개의 매크로블록은 프레임간 부호화 매크로블록(움직임 보상 또는 이전 부호화된 비디오 프레임에서의 다른 종속성을 이용하여 부호화된 매크로블록)에 대해 평균화된다. 매크로블록 SAD를 적용하기 위한 2개의 임계값은 다음 공식을 이용하여 이들 평균값으로부터 유도(574)된다.

[0514] $$T_{MBSADLower} = \overline{MBSAD} - \overline{DMBSAD},$$
$$T_{MBSADUpper} = \overline{MBSAD} + \overline{DMBSAD}$$

[0515] 여기서, $T_{MBSADLower}$은 하한 임계값이며, $T_{MBSADUpper}$은 상한 임계값으로서, 필요한 경우 1024만큼 제한될 수 있다.

[0516] 매크로블록 SAD 인핸스먼트는 다음 조건에 따라 이들 2개의 임계값에 기초하여 적용(576)된다.

[0517]
$$Q_{MB} = Q_{MBMax} \text{ when } MBSAD > T_{MBSADUpper},$$
$$Q_{MB} = Q_{frame} \text{ when } T_{MADLower} \le MBSAD \le T_{MBSADUpper}$$
$$Q_{MB} = Q_{MBMin} \text{ when } MBSAD < T_{MBSADLower}$$

[0518] 여기서, QMB는 매크로블록에 대한 의도된 양자화기이며, MBSAD는 특정 매크로블록에 대한 SAD 값이다.

[0519] MPEG-4 사양에 따라 파일을 부호화하는 실시예는, 상기 설명한 바와 같이, 매크로블록 단위의 양자화기 변경이 하나의 매크로블록으로부터 다음 매크로블록까지 최대 ±2가 된다는 것에 제한된다.

[0520] 3.5.5 레이트 제어(rate control)

[0521] 본 발명의 실시예에 따라 부호화기에 의해 이용되는 레이트 제어 기술은, 인코더가 비디오 시퀀스를 부호화하기 위해 할당된 비트 레이트를 어떻게 이용하는지 결정할 수 있다. 인코더는 미리 정해진 비트 레이트로 부호화할 것이며, 레이트 제어 기술은 미리 정해진 비트 레이트에 가능한 가깝게 인코더에 의해 생성되는 비트 레이트를 매칭시키게 된다. 이러한 레이트 제어 기술은 비디오 시퀀스의 가장 높은 가시적 품질을 복호화할 때 보장하는 방식으로 비트 레이트를 할당할 수 있다. 많은 레이트 제어는 양자화기를 조정함으로써 수행된다. 양자화기는 인코더가 비디오 시퀀스를 얼마나 정교하게 부호화하는지 결정한다. 더 작은 양자화기는 더 높은 품질과 더 높은 비트 소비를 초래한다. 따라서, 레이트 제어 알고리즘은 비디오 품질과 비트 소비의 상충되는 관점에서의 균형을 맞추는 방식으로 양자화기를 수정하게 된다.

[0522] 본 발명의 실시예에 따른 인코더는 여러 가지 다양한 레이트 제어 기술을 이용할 수 있다. 일실시예에서, 단일 패스 레이트 제어 기술이 이용된다. 다른 실시예에서는, 이중(또는 다중) 패스 레이트 제어 기술이 이용된다. 또한, 필요에 따라, '인증된 비디오 버퍼' 레이트 제어가 수행될 수 있다. 이들 기술에 대한 특정의 예를 이하 설명한다. 그러나, 본 발명의 적용에 따라 인코더에 임의의 레이트 제어 기술이 이용될 수 있다.

[0523] 3.5.5.1 단일 패스 레이트 제어

[0524] 본 발명의 실시예에 따른 단일 패스 레이트 제어 기술의 실시예는, 움직임이 큰 장면에 대해 높은 비트 레이크 피크를 허용하게 된다. 몇몇 실시예에서, 단일 패스 레이트 제어 기술은 소정 장면에서의 움직임의 양의 증가에 따라 비트 레이트를 완만하게 증가시키고, 소정 장면에서의 움직임의 양의 감소에 따라 비트 레이트를 급격하게 감소시키게 된다.

[0525] 일실시예에서, 단일 패스 레이트 제어 알고리즘은 비트 레이트를 추종하기 위해 2개의 평균 기간을 이용한다. 이들 2개의 평균 기간은, 전체 비트 레이트의 수렴(convergence)을 보장하기 위한 장기 평균화와, 소정 장면에서의 동작의 양의 편차에 따라 반응하도록 하는 단기 평균화이다.

[0526] 본 발명의 실시예에 따른 단일 패스 레이트 제어 기술은 도 3i에 도시되어 있다. 단일 패스 레이트 제어 기술(580)은 인코더를 소망하는 비트 레이트, 비디오 프레임 레이트 및 다양한 다른 파라미터(이하 설명함)로 초기화(584)함으로써 개시(582)한다. 양자화기를 나타내는 부동소수점 변수가 저장된다. 프레임을 양자화하는 경우(586), 부동소수점 변수가 검색되며(588), 양자화기에 의해 부동소수점 변수를 가장 인접한 정수로 라운딩 처리함으로써 획득된다. 다음으로, 프레임이 부호화된다(590). 새로운 양자화 값의 결정(592)을 가능하게 하는 프레임의 부호화 동안 관측(observation)이 이루어진다. 프로세스는 더 이상 프레임이 없는 경우까지 반복을 결정한다(594). 이 시점에서, 부호화가 완료된다(596).

[0527] 상술한 바와 같이, 인코더는 다양한 파라미터에 의해 초기화된다(584). 이들 파라미터는, '비트 레이트', '프레임 레이트', 'Max Key Frame Interval'(최대 키 프레임 구간), 'Maximum Quantizer'(최대 양자화기), 'Minimum Quantizer'(최소 양자화기), '평균화 기간', '반응 기간', '다운 비율/업 비율'이다. 이하 이들 파라미터의 각각에 대해 설명한다.

[0528] 3.5.5.1.1. "비트 레이트"

[0529] "비트 레이트" 파라미터는 부호화의 목표 비트 레이트를 설정한다.

[0530] 3.5.5.1.2. "프레임 레이트"

[0531] "프레임 레이트"는 비디오 프레임 사이의 기간을 정의한다.

[0532] 3.5.5.1.3. "최대 키 프레임 구간"(Max Key Frame Interval)

[0533] "최대 키 프레임 구간"은 키 프레임 사이의 최대 구간을 특정한다. 키 프레임은 코덱이 장면 전환을 검출한 때에 부호화된 비디오에 자동으로 삽입된다. 단일 컷없이 긴 구간동안 장명이 연속하는 환경에서, 키 프레임은 그 키 프레임 사이의 구간이 항상 "최대 키 프레임 구간" 미만이거나 동일하다는 것을 보장하면서 삽입될 수 있다. 일실시예에서, "최대 키 프레임 구간" 파라미터는 300 프레임의 값으로 설정될 수 있다. 다른 실시예에서는 다른 값이 사용될 수 있다.

[0534] 3.5.5.1.4. "최대 양자화기" 및 "최소 양자화기"

[0535] "최대 양자화기" 및 "최소 양자화기" 파라미터는 부호화에 사용된 양자화기의 상위 경계와 하위 경계를 설정한다. 일실시예에서, 양자화기 경계는 1과 31 사이의 값으로 설정된다.

[0536] 3.5.5.1.5. "평균 기간"

[0537] "평균 기간" 파라미터는 양자화기를 수정할 때에 간주되는 비디오의 양을 제어한다. 긴 평균 기간은 통상적으로 더욱 정확한 전체 레이트를 갖는 부호화된 비디오의 결과로 나타날 것이다. 일실시예에서는 2000의 "평균 기간"이 이용된다. 그러나, 다른 실시예에서는 다른 값이 이용될 수도 있다.

[0538] 3.5.5.1.6. "반응 기간"

[0539] "반응 기간" 파라미터는 최근의 장면에서의 움직임의 변화를 부호화기가 얼마나 고속으로 적응시키는지를 결정한다. "반응 기간"의 값이 더 길면, 고속 움직임 장면은 더 우수한 화질이 되고, 저속 움직임 장면은 화질이 더 열화될 수 있다. 일실시예에서는 10의 "반응 기간"이 이용된다. 그러나, 다른 실시예에서는 다른 값이 이용될 수도 있다.

[0540] 3.5.5.1.7. "다운/업 비율"

[0541] "다운/업 비율" 파라미터는 움직임이 큰 장면 또는 움직임이 작은 장면에 반응하여 양자화기 조정을 위한 상대적인 민감도를 제어한다. 이 값이 더 크면, 움직임이 큰 장면은 화질이 더 우수하게 되고, 비트 소모를 증가시킨다. 일실시예에서는 20의 "다운/업 비율"이 이용된다. 그러나, 다른 실시예에서는 다른 값이 이용될 수도 있다.

[0542] 3.5.5.1.8. 양자화기값을 계산

[0543] 전술한 바와 같이, 단일 패스 레이트 제어 기술은 각각의 프레임의 부호화 후에 양자화기값의 계산을 수반한다. 이하는 양자화기값을 갱신하기 위해 사용될 수 있는 본 발명의 실시예에 따른 기술에 대한 설명이다.

[0544] 부호화기는 "평균 기간"($P_{average}$) 및 "반응 기간"($P_{reaction}$)과 동일한 기간을 갖는 2개의 지수함수적 이동 평균을 비트 레이트의 이동 평균으로서 유지한다. 2개의 지수함수적 이동 평균은 다음의 관계에 따라 계산될 수 있다:

[0545] $$A_t = A_{t-1} \cdot \frac{P-T}{P} + B \cdot \frac{T}{P}$$

[0546] 여기서,

[0547] A_t는 순간 t에서의 평균이고,

[0548] A_{t-1}은 순간 t-T에서의 평균(통상, 이전 프레임에서의 평균)이며,

[0549] T는 간격 기간(통상, 프레임 시간)을 나타내고,

[0550] P는 $P_{average}$ 및/또는 $P_{reaction}$ 중의 하나가 될 수 있는 평균 기간이다.

[0551] 위에서 계산된 이동 평균은 다음의 계산을 이용하여 현재 순간과 과거 순간 사이의 시간 구간을 분할함으로써

비트 레이트로 조정된다:

$$R_t = A_t \frac{1}{T}$$

[0552]

[0553] 여기서,

[0554] Rt는 비트 레이트이고,

[0555] At는 이동 평균 중의 하나이며,

[0556] T는 현재 순간과 과거 순간 사이의 시간 구간이다(이것은 통상적으로 프레임 레이트의 역이다).

[0557] 부호화기는 다음 프레임의 목표 비트 레이트(R_{target})를 다음과 같이 계산할 수 있다:

$$R_{target} = R_{overall} + (R_{overall} - R_{average})$$

[0558]

[0559] 여기서,

[0560] $R_{overall}$은 전체 비디오에 대해 설정된 전체적인 비트 레이트이고,

[0561] $R_{average}$는 긴 평균 기간을 이용하는 평균 비트 레이트이다.

[0562] 여러 실시예에서, 목표 비트 레이트는 하위 경계가 전체 비트 레이트의 75%이다. 목표 비트 레이트가 이 경계 아래로 떨어지면, 목표 비트 레이트는 비디오의 화질을 보장하기 위해 경계의 위로 강제로 상향될 것이다.

[0563] 그 후, 부호화기는 R_{targer}과 $R_{reaction}$ 간의 차이에 기초하여 내부 양자화기를 갱신한다. $R_{reaction}$이 R_{target} 미만이라면, 이전의 프레임이 비교적 낮은 복잡도의 것일 가능성이 있다. 따라서, 양자화기는 다음의 계산을 수행함으로써 감소될 수 있다:

$$Q'_{internal} = Q_{internal} \cdot \left(1 - \frac{1}{P_{reaction}}\right)$$

[0564]

[0565] 여기서, $R_{reaction}$이 R_{target} 보다 클 때, 처리된 이전의 프레임이 비교적 높은 레벨의 복잡도를 가질 상당한 가능성이 존재한다. 따라서, 양자화기는 다음의 계산을 수행함으로써 증가될 수 있다:

$$Q'_{internal} = Q_{internal} \cdot \left(1 + \frac{1}{SP_{reaction}}\right)$$

[0566]

[0567] 여기서,

[0568] s는 "업/다운 비율"이다.

[0569] 3.5.5.1.9. B-VOP 부호화

[0570] 전술한 알고리즘은 B-VCO는 부호화에 적용될 수도 있다. B-VOP가 부호화시에 인에이블될 때, B-VOP(Qb)에 대한 양자화기는 B-VOP 다음의 P-VOR(Qp)의 양자화기에 기초하여 선택된다. 그 값은 다음의 관계에 따라 획득될 수 있다:

$$Q_B = 2.Q_P \quad \text{for} \quad Q_P \le 4$$

$$Q_B = 5 + \frac{3}{4}.Q_P \quad \text{for} \quad 4 < Q_P \le 20$$

$$Q_B = Q_P \quad \text{for} \quad Q_P \ge 20$$

[0571]

[0572] 3.5.5.2. 이중 패스 레이트 제어

[0573] 이중(또는 다중) 패스 레이트 제어 기술을 이용하는 본 발명의 실시예에 따른 부호화기는 제1 패스에서 비디오 시퀀스의 속성을 결정할 수 있고, 그 후 전체 시퀀스의 속성에 대한 지식을 가지고 비디오 시퀀스를 부호화한다. 따라서, 부호화기는 비디오 시퀀스에서의 다른 프레임과 비교한 상대적인 복잡도에 기초하여 각각의 프레임에 대한 양자화 레벨을 조정할 수 있다.

[0574] 본 발명의 실시예에 따른 이중 패스 레이트 제어 기술에서, 부호화기는 비디오가 전술한 단일 패스 레이트 제어 기술에 따라 부호화되고 각각의 프레임의 복잡도가 기록되는(복잡도를 측정하기 위한 광범위의 상이한 측정법 중의 어떠한 것도 이용될 수 있음) 제1 패스를 수행한다. 평균 복잡도 및 그에 따라 평균 양자화기(Q_{ref})가 제1 패스에 기초하여 결정될 수 있다. 제2 패스에서, 비트 스트림은 제1 패스 동안에 계산된 복잡도값에 기초하여 결정된 양자화기로 부호화된다.

[0575] 3.5.5.2.1. I-VOP에 대한 양자화기

[0576] I-VOP에 대한 양자화기 Q는 다음 프레임이 I-VOP가 아닌 경우에는 $0.75 \times Q_{ref}$로 설정된다. 다음 프레임 또한 I-VOP 이라면, Q(현재 프레임에 대한)는 $1.25 \times Q_{ref}$ 로 설정된다.

[0577] 3.5.5.2.2. P-VOP에 대한 양자화기

[0578] P-VOP에 대한 양자화기는 다음의 표현식을 이용하여 계산될 수 있다:

$$Q = F^{-1}\left\{ F(Q_{ref}) \left(\overline{C_{complexity}} / C_{complexity} \right)^{k} \right\}$$

[0579]

[0580] 여기서,

[0581] $C_{complexity}$는 프레임의 복잡도이고,

[0582] $\overline{C_{complexity}}$ 는 비디오 시퀀스의 평균 복잡도이다.

[0583] F(x)는 양자화값 x를 갖는 양자화기를 이용하여 프레임을 부호화하기 위해 요구된 비트의 수를 제공하도록 프레임의 복잡도가 승산되어야 하는 횟수를 제공하는 함수이며,

[0584] F'(x)는 F(x)의 역함수이며,

[0585] k는 강도 파라미터이다.

[0586] 다음의 표는 양자화기 Q와 함께 부호화기를 이용하여 프레임을 부호화하도록 요구된 비트의 수를 계산하기 위해 프레임의 복잡도가 승산되어야 하는 계수를 생성하기 위해 사용될 수 있는 함수 F(Q)의 실시예를 정의한다.

표 6

[0587]

Q	F(Q)	Q	F(Q)
1	1	9	0.013
2	0.4	10	0.01
3	0.15	11	0.008
4	0.08	12	0.0065
5	0.05	13	0.005
6	0.032	14	0.0038
7	0.022	15	0.0028
8	0.017	16	0.002

[0588] Q에 대한 F(Q)의 값

[0589] 강도 파라미터 k가 0로 되도록 선택되면, 그 결과는 일정 양자화기가 된다. 강도 파라미터가 1이 되도록 선택될 때, 양자화기는 $C_{complexity}$에 비례한다. 본 발명의 실시예에 따른 여러 부호화기는 0.5와 동일한 강도 파라미터 k를 갖는다.

[0590] 3.5.5.2.3. B-VOP에 대한 양자화기

[0591] B-VOP에 대한 양자화기는 전술한 단일 패스 기술에서의 B-VOP에 대한 양자화기를 선택하는 기술을 이용하여 선택될 수 있다.

[0592] 3.5.5.3. 비디오 버퍼 검증 레이트 제어

[0593] 프레임을 표현하기 위해 요구된 비트의 수는 비디오 시퀀스의 특성에 따라 좌우되어 변화할 수 있다. 대부분의 통신 시스템은 일정한 비트 레이트로 동작한다. 가변 비트 레이트 통신이 가질 수도 있는 문제점은 리소스 사용에서의 피크를 처리하기 위해 충분한 리소스를 할당하는 것이다. 본 발명의 실시예에 따른 여러 부호화기는, 가변 비트 레이트 통신의 비트 레이트가 급감할 때, 복호화기 비디오 버퍼의 오버플로우를 방지한다는 관점에서 비디오를 부호화한다.

[0594] 비디오 버퍼 검증기(VBV) 레이트 제어의 목적은 전송시에 복호화기의 버퍼를 초과하지 않을 비디오를 생성하는 것을 포함한다. 또한, 부호화된 비디오가 목표 비트 레이트에 부합하고 레이트 제어가 높은 화질의 비디오를 생성하는 것이 바람직하다.

[0595] 본 발명의 여러 실시예에 따른 부호화기는 적어도 2개의 VBV 레이트 제어 기술의 선택을 제공한다. VBV 레이트 제어 기술 중의 하나는 코절 레이트(causal rate) 제어로서 지칭되며, 다른 한 가지 기술은 N번째 패스 레이트 제어로서 지칭된다.

[0596] 3.5.5.3.1 코절 레이트 제어

[0597] 코절 VBV 레이트 제어는 단일 패스 레이트 제어 기술과 함께 사용되고, 현재의 양자화기값 및 이전의 양자화기값에 기초하여 간략히 출력을 생성할 수 있다. 본 발명의 실시예에 따른 부호화기는 다음의 관계에 따라 프레임 n에 대한 양자화기(즉, Q_n)를 설정하는 과정을 포함하는 코절 레이트 제어를 포함한다.

[0598]
$$\frac{1}{Q'_n} = \frac{1}{Q'_{n-1}} + X_{bitrate} + X_{velocity} + X_{size}$$
$$\frac{1}{Q_n} = \frac{1}{Q'_n} + X_{drift}$$

[0599] 여기에서, Q'n은 단일 패스 레이트 제어에 의해 평가된 양자화기이며, $X_{bitrate}$는 요구된 비트 레이트로부터의 드리프트에 기초하여 목표 비트 레이트를 결정함으로써 계산되며, $X_{velocity}$는 VBV 버퍼가 오버플로우하거나 또는 언더플로우할 때까지의 평가 시간에 기초하여 계산되며, X_{size}는 P-VOP의 결과에만 적용되고, 압축된 P-VOP의 크기

가 시간 경과에 따라 변화하는 레이트에 기초하여 계산되며, X_{drift}는 요구된 비트 레이트로부터의 드리프트량이다.

[0600] 3.5.5.3.2. N번째 패스 VBV 레이트 제어

[0601] N번째 패스 VBV 레이트 제어는 복수의 패스 레이트 제어 기술과 관련하여 사용될 수 있으며, 비디오 시퀀스의 이전의 분석 동안에 축적된 정보를 이용한다. 본 발명의 여러 실시예에 따른 부호화기는 도 3.9에 예시된 프로세스에 따라 N번째 VBV 레이트 제어를 수행한다. 프로세스(600)는 제1 패스가 개시되고, 그 동안 분석(602)이 수행된다. 맵 생성이 수행되고(604), 전략(strategy)이 생성된다(606). 그 후, n번째 패스 레이트 제어가 수행된다(608).

[0602] 3.5.5.3.3. 분석

[0603] 일실시예에서, 제1 패스는 일부 형태의 코절 레이트 제어를 이용하며, 프레임의 지속시간, 프레임의 코딩 유형, 사용된 양자화기, 생성된 움직임 비트, 및 생성된 텍스쳐 비트 등에 관하여 각각의 프레임에 대해 데이터가 기록된다. 또한, 타임스케일, 레졸루션 및 코덱 설정 등의 전체적인 정보 또한 기록될 수 있다.

[0604] 3.5.5.3.4. 맵 생성

[0605] 분석으로부터의 정보는 비디오 시퀀스의 맵을 생성하기 위해 사용된다. 맵은 각각의 프레임(I/B/P)에 대한 코딩 유형을 특정할 수 있고, 프레임의 지속시간, 움직임 복잡도 및 텍스쳐 복잡도에 관하여 각각의 프레임에 대한 데이터를 포함할 수 있다. 다른 실시예에서, 맵은 또한 압축 프레임 크기 및 지각 변화(perceptual distortion)에 관한 양자화기 및 다른 파라미터의 영향의 더 우수한 예측을 가능하게 하는 정보를 포함할 수도 있다. 여러 실시예에서, 맵 생성은 N번째 패스가 완료된 후에 수행된다.

[0606] 3.5.5.3.5. 전략 생성

[0607] 맵은 N번째 패스 레이트 제어가 어떻게 작용할지에 관해 전략을 세우기 위해 사용될 수 있다. 모든 프레임이 부호화된 후 이상적인 레벨의 VBV 버퍼가 계획될 수 있다. 일실시예에서, 전략 생성은 요구된 압축 프레임 크기를 포함한 각각의 프레임에 대한 정보, 즉 요구된 프레임 양자화기의 결과로 나타난다. 여러 실시예에서, 전략 생성은 맵 생성 후와 N번째 패스 이전에 수행된다.

[0608] 일실시예에서, 전략 생성 프로세스는 반복 프로세스를 이용하여, 부호화기를 시뮬레이션하고, 양자화기를 가능한 한 중간 양자화기값에 근접하게 유지하도록 함으로써 각각의 프레임에 대한 요구된 양자화기값을 결정한다. 전체 비디오 시퀀스에 대한 베이스 양자화기를 생성하기 위해 이진수 탐색이 사용될 수 있다. 베이스 양자화기는 시뮬레이터로 하여금 요구된 목표 비트 레이트를 달성하도록 하는 정수값이다. 베이스 양자화기가 발견된 후, 전략 생성 프로세스는 VBV 제한사항을 고려한다. 일실시예에서, 일정 양자화기가 VBV 제한사항을 수정하지 않을 것이라면, 일정 양자화기가 사용된다. 다른 실시예에서, 양자화기는 비디오 프레임에서의 움직임의 복잡도에 기초하여 변조된다. 이 과정은 장면 전환 및 기타 임시 효과의 차단을 통합하도록 추가로 확장될 수 있다.

[0609] 3.5.5.3.6. 인루프 N번째 패스 레이트 제어

[0610] 일실시예에서, 인루프 N번째 패스 레이트 제어는 전략을 사용하며, 압축된 프레임 크기 및 지각 변화에 관한 양자화기 및 다른 파라미터의 영향에 대한 최상의 가능한 예측을 행하기 위해 맵을 사용한다. 그 전략에서 탈피하여 단기간 보정 전략(short-term corrective strategy)을 취하기 위한 제한된 재량권이 존재할 수 있다. 통상적으로, 전략에 후속하여서는 VBV 모델의 위반을 방지할 것이다. 일실시예에서, 인루프 N번째 패스 레이트 제어는 PID 제어 루프를 이용한다. 제어 루프에서의 피드백은 이상적인 비트 레이트로부터의 누적된 드리프트양이다.

[0611] 전략 생성이 프레임을 저하시키는 단계를 포함하지는 않지만, 인루프 N번째 레이트 제어는, N번째 레이트 제어가 프레임을 저하시키지 않을 경우에 VBV 버퍼가 언더플로우할 것이라면, 프레임을 저하시킬 수도 있다. 마찬가지로, 인루프 N번째 패스 레이트 제어는 VBV 오버플로우를 방지하기 위해 삽입될 비디오 스터핑(video stuffing)을 요청할 수 있다.

[0612] 3.5.6. 예측

[0613] 일실시예에서, AD/DC 예측은 7.4.3.절(DC 및 AC 예측) 및 7.7.1.절(필드 DC 및 AC 예측)의 ISO/IEC 1449-2:2001(E)로 지칭된 표준에 따르는 방식으로 수행된다.

[0614] 3.5.7. 문자 코딩

[0615] 본 발명의 실시예에 따른 부호화기는 부록 B(가변장 코드) 및 7.4.1.절(가변장 복호화)의 ISO/IEC 14496-2:2001(E)로 지칭된 표준에 따르는 방식으로 문자 코딩을 수행할 수 있다.

[0616] 3.5.8. 움직임 코딩

[0617] 본 발명의 실시예에 따른 부호화기는 부록 B(가변장 코드) 및 7.6.3.절(움직임 벡터 복호화)의 ISO/IEC 14496-2:2001(E)로 지칭된 표준에 따르는 방식으로 문자 코딩을 수행할 수 있다.

[0618] 3.5.9. "비디오" 청크를 생성

[0619] 비디오 트랙은 1 내지 N의 프레임 시퀀스로 고려될 수 있다. 본 발명의 실시예에 따른 시스템은 압축 비트스트림을 생성하기 위해 시퀀스를 부호화할 수 있다. 압축 비트스트림은 압축 비트스트림을 청크 1 내지 N으로 단편화함으로써 포맷화된다. 각각의 비디오 프레임 n은 대응하는 청크 n을 갖는다.

[0620] 청크는 청크 n이 청크 1 내지 n-1과 함께 비디오 프레임 n을 복호화하기 위해 본 발명의 실시예에 따른 복호화기에 대한 충분한 정보를 포함할때까지 비트스트림으로부터의 비트를 청크 n에 추가함으로써 생성된다. 비디오 프레임 n을 생성하기 위해 충분한 정보가 청크 1 내지 n-1에 포함되는 경우, 본 발명의 실시예에 따른 부호화기는 마커 청크를 포함할 수 있다. 일실시예에서, 마커 청크는 이전의 프레임만큼의 동일한 프레임 정보를 갖는 코딩되지 않은 P-프레임이다.

[0621] 3.6. "서브타이틀" 청크를 생성

[0622] 본 발명의 실시예에 따른 부호화기는 일련의 표준 포맷 중의 한 포맷의 서브타이틀을 취하고, 그 후 이 서브타이틀을 비트 맵으로 변환할 수 있다. 비트 맵에서의 정보는 그 후 실행 길이 부호화를 이용하여 압축된다. 실행 길이 부호화된 비트 맵은 청크로 포맷화되며, 이 청크 또한 청크 내에 포함된 특정의 서브타이틀에 대한 개시 시각 및 중지 시간에 관련한 정보를 포함한다. 여러 실시예에서, 화면 상의 서브타이틀의 컬러, 크기 및 위치에 관련한 정보 또한 청크 내에 포함될 수 있다. 청크는 서브타이틀에 대한 팰릿(palette)을 설정하고 그 팰릿이 변경되었다는 것을 나타내는 서브타이틀 트랙에 포함될 수 있다. 서브타이틀을 표준 서브타이틀 포맷으로 생성할 수 있는 임의의 어플리케이션이 사용되어 서브타이틀의 텍스트를 생성한다. 이와 달리, 사용자가 직접 입력한 텍스트를 서브타이틀 정보로 변환하기 위해 소프트웨어가 사용될 수도 있다.

[0623] 3.7. 인터리빙

[0624] 인터리버(interleaver)가 전술한 청크 전부를 수신한 후, 인터리버는 멀티미디어 파일을 구성한다. 멀티미디어 파일을 구성하는 것은 "CSET" 청크, "INFO" 리스트 청크, "hdrl" 청크, "movi" 리스트 청크 및 idxl 청크를 작성하는 단계를 포함할 수 있다. 이하에서는 이들 청크를 작성하고 멀티미디어 파일을 생성하기 위한 본 발명의 실시례에 따른 방법이 설명된다.

[0625] 3.7.1. "CSET" 청크를 생성

[0626] 전술한 바와 같이, "CSET" 청크는 옵션이며, AVI 컨테이너 포맷 사양에 따른 인터리버에 의해 생성될 수 있다.

[0627] 3.7.2. "INFO" 리스트 청크를 생성

[0628] 전술한 바와 같이, "CSET" 청크는 옵션이며, AVI 컨테이너 포맷 사양에 따른 인터리버에 의해 생성될 수 있다.

[0629] 3.7.3. "hdrl" 리스트 청크를 생성

[0630] "hdrl" 리스트 청크는 인터리버에 제공된 다양한 청크에서의 정보에 기초하여 인터리버에 의해 생성된다. "hdrl" 리스트 청크는 참조된 청크의 파일 내의 위치(location)를 참조한다. 일실시예에서, "hdrl" 리스트 청크는 기준을 확립하기 위해 파일 오프셋을 사용한다.

[0631] 3.7.4. "movi" 리스트 청크를 생성

[0632] 전술한 바와 같이, "movi" 리스트 청크는 "오디오", "비디오" 및 "서브타이틀" 청크를 작성하기 위해 오디오, 비디오 및 서비타이틀 트랙을 부호화하고 그 후 이들 청크를 인터리브함으로써 작성된다. 여러 실시예에서,

"movi" 리스트 청크 또한 디지털 권리 관리 정보를 포함할 수 있다.

[0633] 3.7.4.1. 비디오/오디오/서브타이틀을 인터리브

[0634] 오디오, 비디오 및 서브타이틀 청크를 인터리브하기 위해 다양한 규칙이 사용될 수 있다. 대표적으로, 인터리버는 비디오 트랙 및 오디오 트랙의 각각에 대해 다수의 대기열을 구성한다. 인터리버는 어느 대기열이 출력 파일에 기입되어야 하는지를 결정한다. 대기열 선택은 최저의 인터리브 기간이 기입되는 대기열에서부터 기록함으로써 인터리브 기간에 기초하여 이루어질 수 있다. 인터리버는 청크가 파일에 기입될 수 있기 전에 대기열에 제공될 전체 인터리브 기간을 대기하여야만 할 것이다.

[0635] 일실시예에서, 생성된 "오디오", "비디오" 및 "서브타이틀" 청크는, 이들이 대응하는 비디오 프레임에 관련한 정보를 포함하고 있는 "비디오" 청크에 앞서 "오디오" 및 "서브타이틀" 청크가 파일 내에 위치되도록 인터리브된다. 다른 실시예에서, "오디오" 및 "서브타이틀" 청크는 이들이 대응하는 "비디오" 청크 뒤에 위치될 수도 있다. "오디오", "비디오" 및 "서브타이틀" 청크의 위치 간의 시간차는 디바이스를 플레이하기 위해 사용되는 플레이어의 버퍼링 성능에 크게 좌우된다. 버퍼링이 제한되거나 알려져 있지 않은 실시예에서, 인터리버는 "오디오" 및 "서브타이틀" 청크가 "비디오" 청크 사이에 위치되도록 "오디오", "비디오" 및 "서브타이틀" 청크를 인터리브하며, "오디오" 및 "서브타이틀" 청크를 바로 후속하는 "비디오" 청크는 오디오 또는 서브타이틀에 대응하는 제1 비디오 프레임을 포함한다.

[0636] 3.7.4.2. DRM 정보를 생성

[0637] 멀티미디어 파일의 비디오 컨텐츠를 보호하기 위해 DRM이 사용되는 실시예에서, DRM 정보는 비디오 청크의 부호화와 동시에 생성될 수 있다. 각각의 청크가 생성될 때, 청크는 암호화되어, 비디오 청크의 암호화에 관련한 정보를 포함하여 생성된 DRM 청크가 될 수 있다.

[0638] 3.7.4.3. DRM 정보를 인터리브

[0639] 본 발명의 실시예에 따른 인터리버는 비디오 청크의 암호화에 관련한 정보를 포함하는 DRM 청크를 비디오 청크의 앞에 인터리브한다. 일실시예에서, 비디오 청크 n에 대한 DRM 청크는 비디오 청크 n-1과 비디오 청크 n 사이에 위치된다. 다른 실시예에서, 비디오 청크 n의 전과 후의 간격은 멀티미디어 파일을 복호화하는 디바이스 내에 제공된 버퍼링의 양에 좌우된다.

[0640] 3.7.5. "idxl" 청크의 생성

[0641] "movi" 리스트 청크가 생성된 후에는, "idxl" 청크의 생성은 간단한 프로세스가 된다. "idxl" 청크는 각각의 "data" 청크의 "movi" 리스트 청크 내에서의 위치를 판독함으로써 작성된다. 이 정보는 "data" 청크가 속하는 트랙에 관련한 "data" 청크로부터 판독된 정보 및 "data" 청크의 컨텐츠와 조합된다. 이 정보의 전부는 그 후 전술한 포맷의 어느 것이라도 정보를 표현하기 위해 사용될 수 있는 적합한 방식으로 "idxl" 청크에 삽입된다.

[0642] 4. 멀티미디어 파일의 전송 및 분배

[0643] 멀티미디어 파일이 생성된 후, 생성된 멀티미디어 파일은 다양한 네트워크를 통해 분배될 수 있다. 다수의 실시예에서 다른 것 중에서도 멀티미디어 재현 및 메뉴를 생성하는데 요구된 요소가 단일 파일 내에 포함된다는 사실은 정보의 전달을 간략화한다. 여러 실시예에서, 멀티미디어 파일은 멀티미디어 파일의 컨텐츠를 해독하기 위해 요구된 정보와는 별개로 분배될 수 있다.

[0644] 일실시예에서, 멀티미디어 컨텐츠는 제1 서버에 제공되고, 본 발명에 따라 멀티미디어 파일을 작성하기 위해 부호화된다. 그 후, 멀티미디어 파일은 제1 서버 또는 제2 서버 중의 하나에 위치될 수 있다. 다른 실시예에서, DRM 정보가 제1 서버, 제2 서버 또는 제3 서버에 위치될 수 있다. 일실시예에서, 제1 서버는 부호화된 멀티미디어 파일의 위치를 규명하고 및/또는 DRM 정보의 위치를 규명하기 위해 질의될 수 있다.

[0645] 5. 멀티미디어 파일을 복호화

[0646] 본 발명의 실시예에 따른 멀티미디어 파일로부터의 정보는 적합한 소프트웨어를 사용하여 구성된 컴퓨터, 멀티미디어 파일로부터의 정보를 액세스하기 위해 배선으로 연결되는 전용 플레이어, 또는 AVI 파일을 구문분석할 수 있는 임의의 다른 디바이스에 의해 액세스될 수 있다. 여러 실시예에서, 디바이스는 멀티미디어 파일의 모

든 정보를 액세스할 수 있다. 다른 실시예에서, 디바이스는 본 발명의 실시예에 따른 멀티미디어 파일 내의 모든 정보를 액세스할 수도 있을 것이다. 특정 실시예에서, 디바이스는 AVI 파일 포맷으로 특정되지 않은 청크 내에 저장되는 전술한 정보 중의 어떠한 정보를 액세스할 수 없다. 정보의 전부가 액세스될 수 있지 않은 실시예에서, 통상적으로, 디바이스는 이 디바이스에 의해 인식되지 않는 청크들을 폐기할 것이다.

[0647] 대표적으로, 본 발명의 실시예에 따른 멀티미디어 파일에 포함된 정보를 액세스할 수 있는 디바이스는 다수의 기능을 수행할 수 있다. 디바이스는 가시 디스플레이 상의 비디오의 디스플레이를 수반하는 멀티미디어 재현을 디스플레이하고, 오디오 시스템 상의 다수의 오디오 트랙 중의 하나로부터 오디오를 생성하고, 다수의 서브타이틀 트랙 중의 하나로부터 서브타이틀을 디스플레이한다. 약간의 실시예는 또한 동반하는 오디오 및/또는 비디오를 디스플레이하면서 가시 디스플레이 상의 메뉴를 디스플레이할 수 있다. 이들 디스플레이 메뉴는 선택 가능한 버튼, 풀다운 메뉴 및 서브-메뉴 등의 특징을 갖는 대화식이이다. 일부 실시예에서, 메뉴 아이템은 현재 액세스되고 있는 멀티미디어 파일 이외의 오디오/비디오 컨텐츠를 가리킬 수 있다. 외부의 컨텐츠는 멀티미디어 파일을 액세스하는 디바이스에 대해 근거리에 위치될 수도 있고, 근거리 통신망, 광역 통신망 또는 공중 통신망을 통해서와 같이 원거리에 위치될 수도 이다. 다수의 실시예는 멀티미디어 파일에 포함된 "메타 데이터" 또는 하나 이상의 멀티미디어 파일에 의해 참조된 "메타 데이터"에 따르 하나 이상의 멀티미디어 파일을 선택할 수 있다.

[0648] 5.1. 멀티미디어 재현의 디스플레이

[0649] 본 발명의 실시예에 따른 멀티미디어 파일의 성능이 복수의 오디오 트랙, 복수의 비디오 트랙 및 복수의 서브타이틀 트랙을 지원하도록 제공되면, 비디오, 오디오 및/또는 서브타이틀을 합성하는 이러한 멀티미디어 파일을 이용한 멀티미디어 재현의 디스플레이는 시각적 메뉴 시스템 또는 풀다운 메뉴 시스템(그 동작은 아래에 설명됨) 중의 하나, 또는 멀티미디어 재현을 생성하기 위해 사용된 디바이스의 디폴트 세팅을 통한 특정 오디오 트랙, 비디오 트랙 및/또는 서브타이틀 트랙의 선택을 필요로 할 수 있다. 오디오 트랙, 비디오 트랙 및 서브타이틀 트랙이 선택된 후, 멀티미디어 재현의 디스플레이가 진행할 수 있다.

[0650] DRM을 포함하는 멀티미디어 파일로부터의 요구된 멀티미디어 정보를 위치시키고 본 발명의 실시예에 따라 멀티미디어 정보를 디스플레이하기 위한 프로세스가 도 4.0에 예시되어 있다. 프로세스(620)는 DRM 헤더를 해독하기 위해 요구된 암호화 키를 획득하는 단계를 포함한다(622). 암호화 키는 그 후 DRM 헤더를 해독하기 위해 사용되며(624), 제1 DRM 청크는 멀티미디어 파일의 "movi" 리스트 청크 내에 위치된다(626). "DRM" 청크를 해독하기 위해 요구된 암호화 키는 "DRM" 헤더의 테이블로부터 획득되며(628), 암호화 키는 암호화된 비디오 프레임을 해독하기 위해 사용된다. 비디오 정크를 수반하는 요구된 오디오 정크 및 임의의 요구된 서브타이틀 정크는 그 후 복호화되며(630), 오디오, 비디오 및 임의의 적합한 정보가 디스플레이 및 사운드 시스템을 통해 표시된다(632).

[0651] 여러 실시예에서, 선택된 오디오 트랙은 스테레오 또는 서라운드 사운드 오디오를 제공하기 위해 복수의 채널을 포함할 수 있다. 적합한 트랙이 디스플레이되도록 선택될 때, 이전의 비디오 프레임이 서브타이틀을 포함하였는지에 관한 판정이 이루어질 수 있다(이러한 판정은 현재 복호화된 비디오 프레임을 통해 디스플레이되어야 하는 서브타이틀 정보를 포함하고 있던 이전의 "서브타이틀" 정크를 식별하는 성과를 달성하는 다양한 방식 중의 하나로 이루어질 수도 있다. 이전의 서브타이틀이 서브타이틀을 포함하고, 서브타이틀에 대한 타이밍 정보가 그 서브타이틀이 현재 프레임으로 디스플레이되어야 한다는 것을 나타낸다면, 그 서브타이틀은 복호화된 비디오 프레임에 중첩된다. 이전의 프레임이 서브타이틀을 포함하지 않거나, 이전의 프레임 상의 서브타이틀에 대한 타이밍 정보가 그 서브타이틀이 현재 복호화된 프레임과 함께 디스플레이되지는 않아야 한다면, 선택된 서브타이틀 트랙에 대한 "서브타이틀" 청크가 탐색된다. "서브타이틀" 정크가 위치된다면, 그 서브타이틀은 복호화된 비디오 상에 중첩된다. 그 후, 비디오(임의의 중첩된 서브타이틀을 포함)가 오디오를 동반하여 디스플레이된다.

[0652] 도 4.0의 설명을 참조하면, 프로세스는 임의의 추가의 DRM 정크가 존재하는지의 여부를 판정한다(634). 존재한다면, 다음 DRM 청크가 위치되고(626), 프로세스는 추가의 DRM 정크가 남지 않을 때까지 지속한다. 이 때에, 오디오, 비디오 및/또는 서브타이틀 트랙의 재현이 완료된다(636).

[0653] 여러 실시예에서, 디바이스는 본 발명에 따른 멀티미디어 파일의 "hdrl" 청크 내에 포함된 정보를 이용하여 멀티미디어 정보의 특정 부분(예컨대, 특정의 동반 오디오 트랙 및 옵션으로 특정의 동반 서브타이틀 트랙을 갖는 영화의 특정 장면)을 탐색할 수 있다. 다수의 실시예에서, "비디오" 정크, "오디오" 정크 및/또는 "서브타이틀" 정크의 복호화는 다른 태스크와 병렬로 수행될 수 있다.

[0654] 멀티미디어 파일로부터 정보를 액세스할 수 있고 특정의 오드오 트랙 및/또는 특정의 서브타이틀 트랙과 함께 비디오를 디스플레이할 수 있는 디바이스의 예로는 소프트웨어를 이용하는 전술한 방식으로 구성되는 컴퓨터가 있다. 또 다른 예로는 이들 성능을 포함하는 코덱이 설치된 DVD 플레이어가 있다. 다른 실시예에서, 특정 미디어 트랙에 대응하는 "데이터" 청크를 위치확인하거나 선택(의도적으로이던 임의적으로던)하고 프레젠테이션을 위해 이들 코덱을 복호화하도록 구성된 임의의 디바이스는 본 발명의 실시예에 따른 멀티미디어 파일을 이용하여 멀티미디어 재현을 생성할 수 있다.

[0655] 여러 실시예에서, 디바이스는 외부 파일로부터의 멀티미디어 정보와 조합하여 멀티미디어 파일로부터 멀티미디어 정보를 플레이할 수 있다. 통상적으로, 이러한 디바이스는 전술한 유형의 멀티미디어 파일에 참조된 로컬 파일로부터 오디오 트랙 또는 서브타이틀 트랙이 공급되게 함으로써 상기와 같이 할 것이다. 참조된 파일이 근거리에 저장되지 않거나 디바이스가 저장되지 않은 위치에 디바이스가 네트워킹된다면, 디바이스는 파일의 로컬 카피를 획득할 수 있다. 그 후, 디바이스는 멀티미디어의 다양한 트랙이 상이한 파일 소스로부터 입력되는 비디오, 오디오 및 서브타이틀(필요시) 파이프라인을 구성하는 양측 파일을 액세스할 것이다.

[0656] 5.2. 메뉴의 생성

[0657] 본 발명의 실시예에 따른 복호화기가 도 4.1에 예시되어 있다. 복호화기(650)는 파일을 디멀티플렉서(654)에 제공함으로써 본 발명의 실시예에 따른 멀티미디어 파일(652)을 처리한다. 디멀티플렉서는 멀티미디어 파일로부터 "DMNU" 청크를 추출하고, "DMNU" 청크로부터 "LanguageMenus" 청크의 전부를 추출하여 메뉴 마서(menu parser)(656)에 제공한다. 디멀티플렉서는 또한 "DMNU" 청크로부터 "Media" 청크 전부를 추출하여 미디어 렌더러(media renderer)(658)에 제공한다. 메뉴 파서(656)는 "LanguageMenu" 청크에 정의된 메뉴 구조를 표현하는 스테이트 머신을 구축하기 위해 "LanguageMenu" 청크로부터의 정보를 구문분석한다. 메뉴 구조를 표현하는 스테이트 머신은 사용자에게 디스플레이를 제공하고 사용자 명령에 응답하도록 사용될 수 있다. 스테이트 머신은 메뉴 스테이트 컨트롤러(660)에 제공된다. 메뉴 스테이트 컨트롤러는 메뉴 스테이트 머신의 현재 상태의 트랙을 유지하고, 사용자로부터 명령을 수신한다. 사용자로부터의 명령은 상태 전이(state transtion)를 야기할 수 있다. 사용자에게 제공된 초기 디스플레이 및 메뉴 상태 전이를 동반하는 디스플레이에 대한 임의의 업데이트는 메뉴 플레이어 인터페이스(662)를 이용하여 제어될 수 있다. 메뉴 플레이어 인터페이스(662)는 메뉴 스테이트 컨트롤러 및 미디어 렌더(media render)에 접속될 수 있다. 메뉴 플레이어 인터페이스는 어느 미디어가 미디어 청크로부터 추출되어 미디어 렌더러에 접속된 플레이어(664)를 통해 사용자에게 제공되어야 하는지에 관해 미디어 렌더러에게 지시한다. 사용자는 키보드, 마우스 또는 원격 제어기 등의 입력 장치를 이용하여 플레이어에게 지시를 제공할 수 있다. 일반적으로, 멀티미디어 파일은 사용자에 대해 초기에 디스플레이되는 메뉴를 지정하고, 사용자의 지시는 초기 메뉴의 생성에 후속하여 디스플레이된 비디오 및 오디오를 지정한다. 도 4.1에 예시된 시스템은 컴퓨터 및 소프트웨어를 사용하여 구현될 수 있다. 다른 예에서, 시스템은 특정 기능의 집적 회로를 사용하거나 또는 소프트웨어와 펌웨어의 조합을 이용하여 구현될 수 있다.

[0658] 본 발명의 실시예에 따른 메뉴의 예가 도 4.2에 예시되어 이다. 메뉴 디스플레이(670)는 4개의 버튼 영역(672)과, 타이틀(676)을 포함하는 배경 비디오(674)와, 포인터(678)를 포함한다. 메뉴는 또한 배경 오디오(도시하지 않음)를 포함한다. 디스플레이에 의해 형성된 시각적 효과는 믿을 수 없는 것일 수도 있다. 버튼의 시각적 외형은 통상적으로 배경 비디오의 일부이고, 버튼 자체는 배경 비디오의 간략하게 정의된 영역으로, 이 영역은 이 영역이 포인터에 의해 활성화될 때에 이 영역에 연관된 특정의 동작을 갖는다. 포인터로는 대표적으로 오버레이(overlay)가 있다.

[0659] 도 4.3은 도 4.2에 도시된 디스플레이에서의 모든 정보의 소스를 개념적으로 예시한다. 배경 비디오(674)는 메뉴 타이틀, 버튼의 시각적 외형 및 디스플레이의 배경을 포함할 수 있다. 이들 요소 및 추가의 요소의 전부는 정지 화상으로 보여지거나 또는 애니메이션화되어 보여질 수 있다. 배경 비디오는 비디오 트랙(702) 내의 배경 비디오의 위치를 나타내는 "MediaTrack" 청크(708)에 포함된 정보를 이용함으로써 추출된다. 메뉴와 동반될 수 있는 배경 오디오(706)는 오디오 트랙(710) 내의 배경 오디오의 위치를 나타내는 "MediaTrack" 청크(708)를 이용하여 위치될 수 있다. 전술한 바와 같이, 포인터(678)는 오버레이(713)의 일부이다. 오버레이(713)는 또한 버튼으로서 보여지는 배경 비디오의 일부분을 강조 표시하도록 보여지는 그래픽을 포함할 수 있다. 일실시예에서, 오버레이(713)는 오버레이 트랙(714) 내에서의 오버레이의 위치를 나타내는 "MediaTrack" 청크(712)를 이용하여 획득된다. 메뉴가 사용자와 대화하는 방식은 버튼의 각각에 관련된 "Action" 청크(도시하지 않음)에 의해 정의된다. 예시된 실시예에서는 "PlayAction" 청크(716)가 예시되어 있다. "PlayAction" 청크는 멀티미디어 파일(즉, 오디오, 비디오 및 가능하게는 서브타이틀 트랙) 내에 포함된 멀티미디어 재현 내의 장면을 간접적으로 참조한다("PlayTrack" 청크에 의해 참조된 다른 청크는 도시되지 않음). "PlayAction" 청크(716)는 궁극적

으로는 "MediaTrack" 청크(718)를 이용하여 장면을 참조하며, "MediaTrack" 청크는 특징 트랙 내의 장면을 나타낸다. 선택되거나 디폴트된 오디오 트랙 및 가능하게는 서브타이틀 트랙에서의 포인트 또한 참조된다.

[0660] 사용자가 입력 장치를 이용하여 명령을 입력할 때, 디스플레이는 버튼의 선택에 응답하여서 뿐만 아니라 버튼 영역 내에 위치되는 포인터에 의해서도 갱신될 수 있을 것이다. 전술한 바와 같이, 메뉴를 생성하기 위해 사용된 미디어 정보의 전부는 멀티미디어 파일 내에 위치되고, 보다 구체적으로는 "DMNU" 청크 내에 위치된다. 다른 실시예에서 이루어지는 것이기는 하지만, 정보는 그 파일 내의 다른 곳에 위치되거나 및/또는 다른 파일에 위치될 수 있다.

[0661] 5.3. 메타 데이터를 액세스

[0662] "메타 데이터"는 정보를 표현하는 표준 방법이다. "메타 데이터"의 표준 성질은 데이터가 자동 프로세스에 의해 액세스되어 이해될 수 있도록 하는 것이다. 일실시예에서, "메타 데이터"는 추출되어, 시청을 위해 사용자에게 제공된다. 여러 실시예는 서버 상의 멀티미디어 파일이 검사되어 사용자의 시청 습관 및 시청 선호도에 관한 정보를 제공하도록 한다. 이러한 정보는 소프트웨어 애플리케이션에 의해 사용되어 사용자가 시청을 즐길 수도 있는 다른 멀티미디어 파일을 권유하도록 할 수 있다. 일실시예에서, 이러한 권유는 다른 사용자의 서버에 포함된 멀티미디어 파일에 기초하여 이루어질 수 있다. 다른 실시예에서, 사용자는 멀티미디어 파일을 요청할 수 있고, 그 파일은 다양한 위치에 있는 멀티미디어 파일의 "메타 데이터"를 검사하는 검색 엔진 및/또는 지능형 에이전트에 의해 위치확인될 수 있다. 또한, 사용자는 상이한 버젼의 재현의 각각이 부호화되는 방식에 관한 "메타 데이터"에 기초하여 특정의 멀티미디어 재현을 포함하고 있는 다양한 멀티미디어 파일 간에 선택이 가능하게 된다.

[0663] 여러 실시예에서, 본 발명의 실시예에 따른 멀티미디어 파일의 "메타 데이터"는 카다로그화하고자 하는 목적(purpose of cataloging)으로 액세스되거나 또는 파일의 컨텐츠를 액세스하기 위한 간단한 메뉴를 작성하기 위해 액세스될 수 있다.

[0664] 전술한 설명이 본 발명의 다수의 특정 실시예를 포함하고 있지만, 이러한 실시예는 본 발명의 사상을 제한하려는 것이 아니라, 단지 본 발명을 예시하기 위한 것이다. 예컨대, 본 발명의 실시예에 따른 멀티미디어 파일은 단일의 멀티미디어 재현을 포함할 수도 있고 복수의 멀티미디어 재현을 포함할 수도 있다. 또한, 이러한 파일은 하나 이상의 메뉴와 다양한 상이한 유형의 "메타 데이터"를 포함할 수 있다. 따라서, 본 발명의 사상은 예시된 실시예에 의해서가 아니라 첨부된 청구범위 및 그 등가물에 의해 정해져야 한다.

부호의 설명

[0665] 10: 시스템

12: 컴퓨터

14: 네트워크

16: 서버

18: 랩탑 컴퓨터

20: 개인 휴대정보 단말기

30: 멀티미디어 파일

38: 메타데이터 청크

50: 제1 'RIFF' 청크

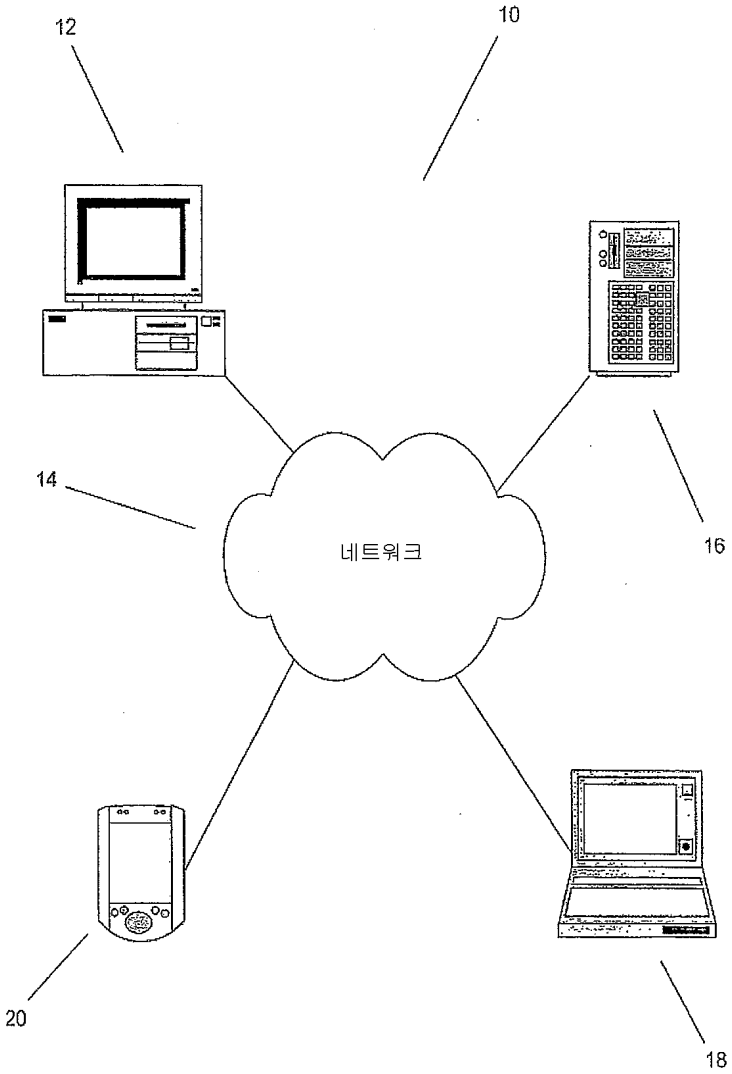
52: 제2 'RIFF' 청크

262: 'video' 청크

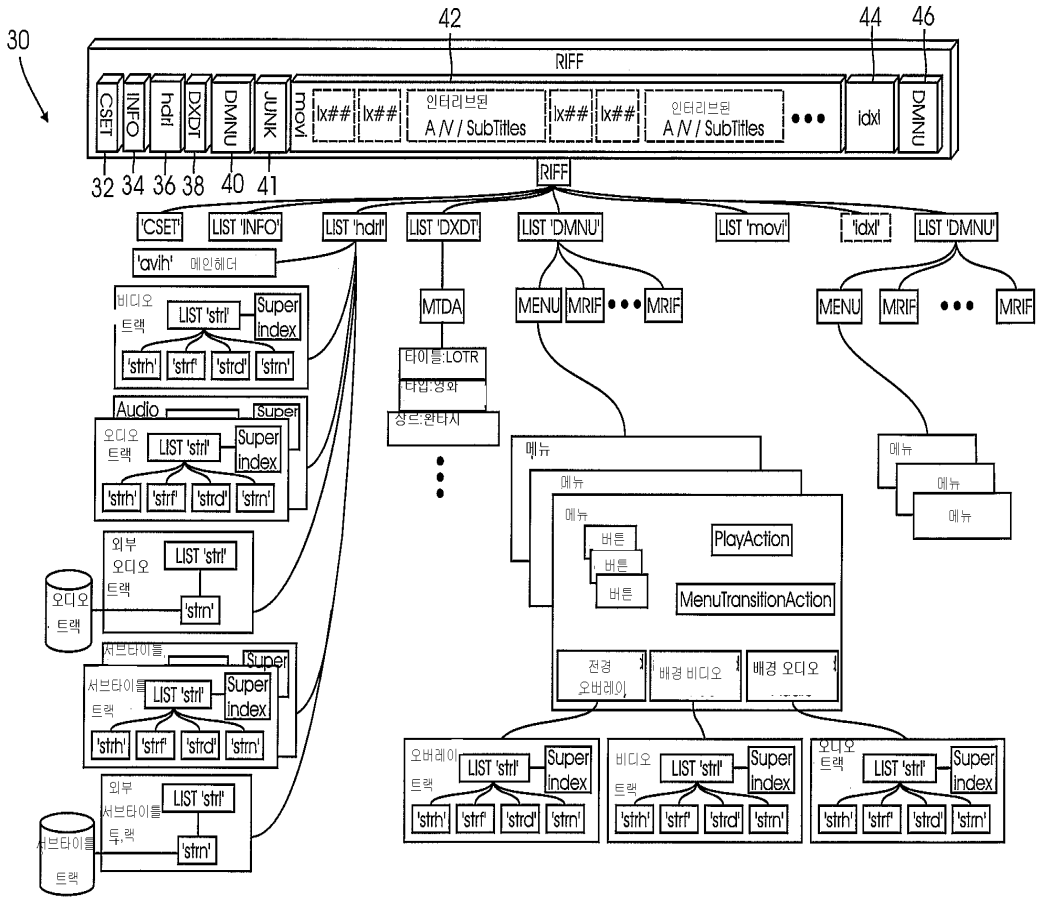
264: 'audio' 청크

도면

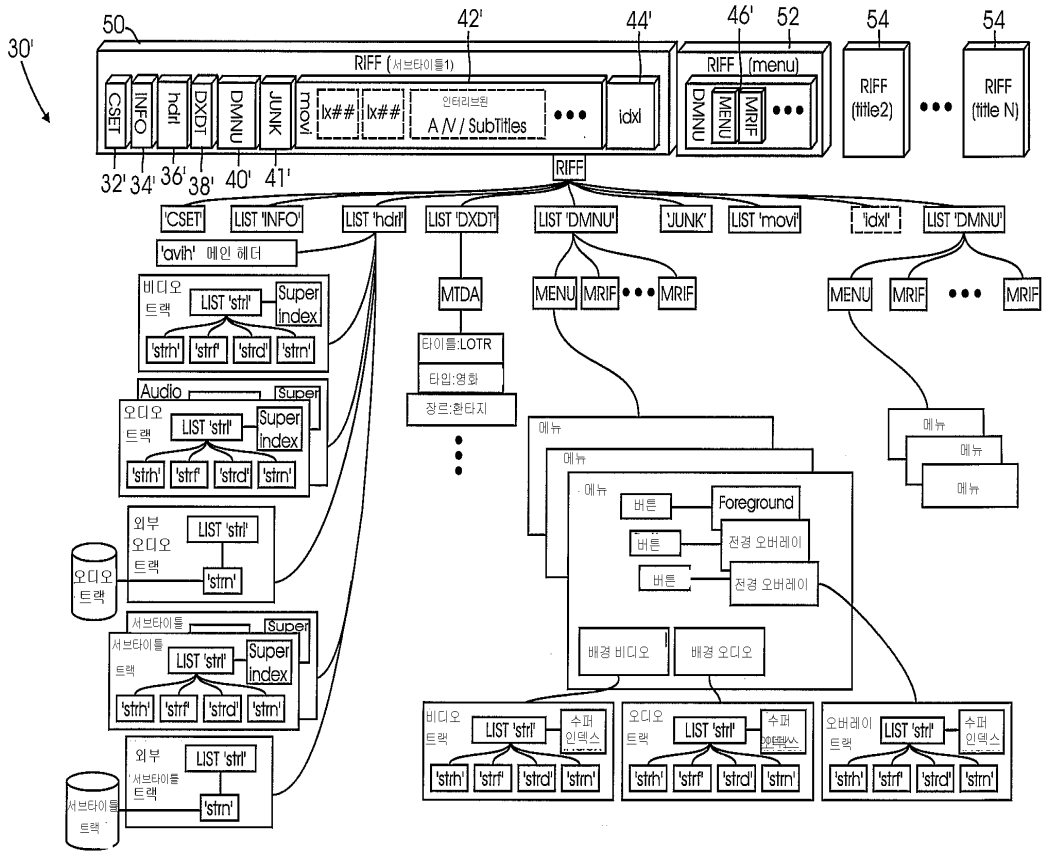
도면1



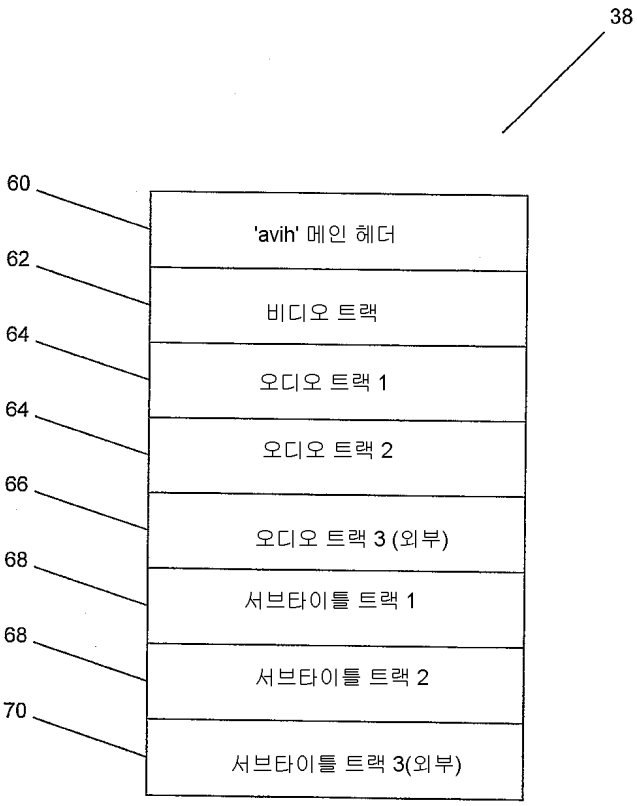
도면2a


30
42
44
46
RIFF
CSET
INFO
hdrl
DXDT
DMNU
JUNK
movi
Ix##
인터리브된 A/V / SubTitles
idxl
32 34 36 38 40 41
'CSET'
LIST 'INFO'
LIST 'hdrl'
LIST 'DXDT'
LIST 'DMNU'
LIST 'movi'
'idxl'
'avih' 메인헤더
비디오 트랙
LIST 'strl'
Super index
'strh' 'strf' 'strd' 'strn'
MTDA
타이틀:LOTR
타입:영화
장르:판타지
MENU
MRIF
Audio
오디오 트랙
외부 오디오 트랙
서브타이틀 트랙
외부 서브타이틀 트랙
메뉴
버튼
PlayAction
MenuTransitionAction
전경 오버레이
배경 비디오
배경 오디오
오버레이 트랙

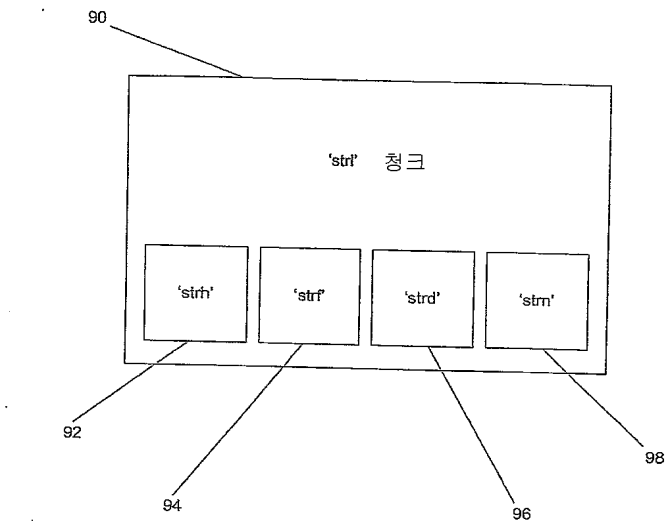
도면2aa


30'
50
42'
44'
46'
52
54
54
RIFF (서브타이틀1)
RIFF (menu)
RIFF (title2)
RIFF (title N)
CSET
INFO
hdrl
DXDT
DMNU
JUNK
movi
ix##
ix##
인터리브된 A/V / SubTitles
idxl
DMNU
MENU
MRIF
32'
34'
36'
38'
40'
41'
RIFF
'CSET'
LIST 'INFO'
LIST 'hdrl'
LIST 'DXDT'
LIST 'DMNU'
'JUNK'
LIST 'movi'
'idxl'
LIST 'DMNU'
'avih' 메인 헤더
비디오 트랙
LIST 'strl'
Super index
'strh'
'strf'
'strd'
'strn'
Audio
오디오 트랙
외부 오디오 트랙
오디오 트랙
서브타이틀 트랙
외부 서브타이틀 트랙
서브타이틀 트랙
MTDA
타이틀:LOTR
타입:영화
장르:환타지
MENU
MRIF
MRIF
메뉴
버튼
Foreground
전경 오버레이
배경 비디오
배경 오디오
비디오 트랙
오디오 트랙
오버레이 트랙
수퍼 인덱스

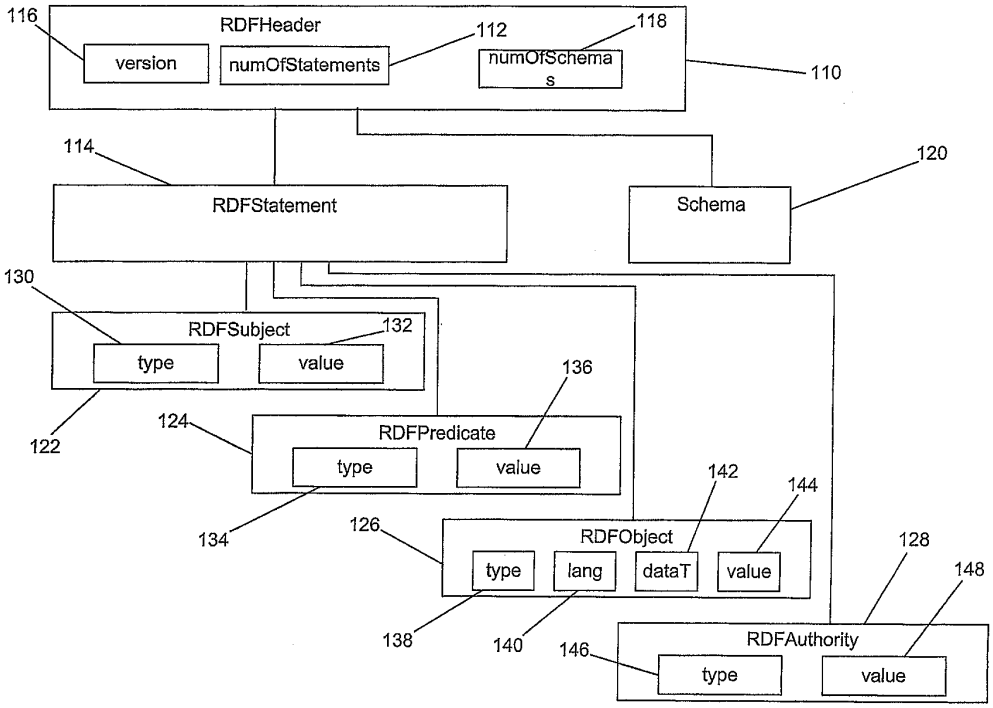
도면2b


38
60
'avih' 메인 헤더
62
비디오 트랙
64
오디오 트랙 1
64
오디오 트랙 2
66
오디오 트랙 3 (외부)
68
서브타이틀 트랙 1
68
서브타이틀 트랙 2
70
서브타이틀 트랙 3(외부)

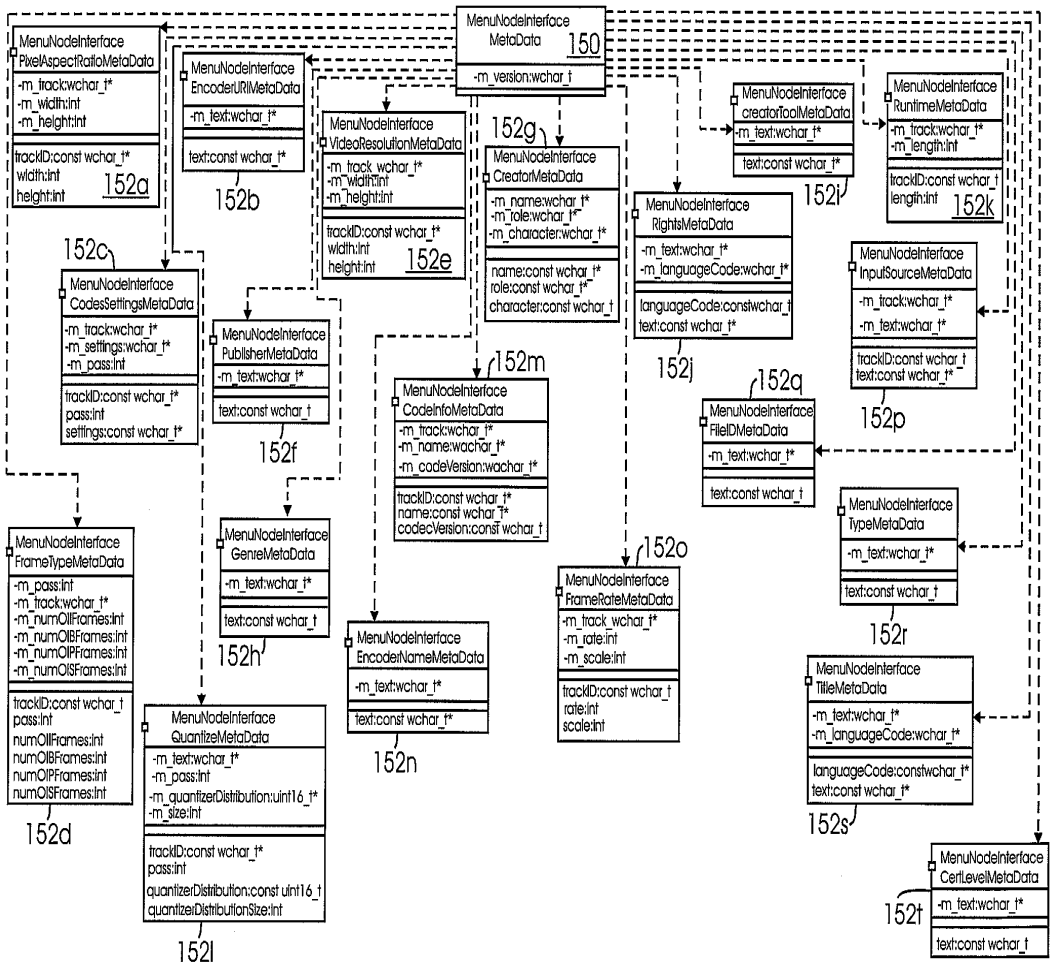
도면2c


90
'strl' 청크
'strh'
'strf'
'strd'
'strn'
92
94
96
98

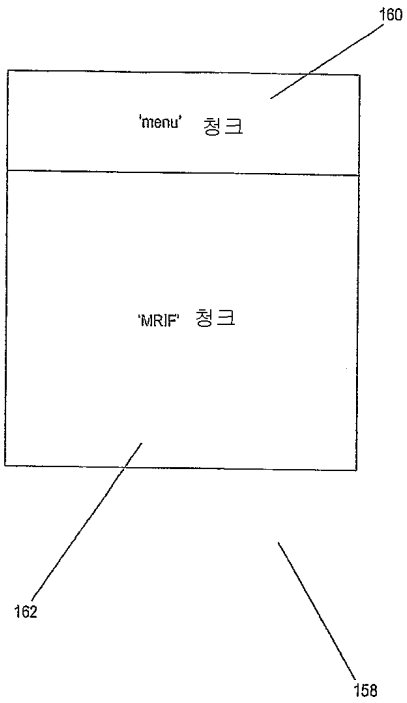
도면2d


116
RDFHeader
version
numOfStatements
112
118
numOfSchema
s
110
114
RDFStatement
Schema
120
130
RDFSubject
type
value
132
122
124
RDFPredicate
type
value
136
134
126
RDFObject
type
lang
dataT
value
142
144
128
148
138
140
146
RDFAuthority
type
value

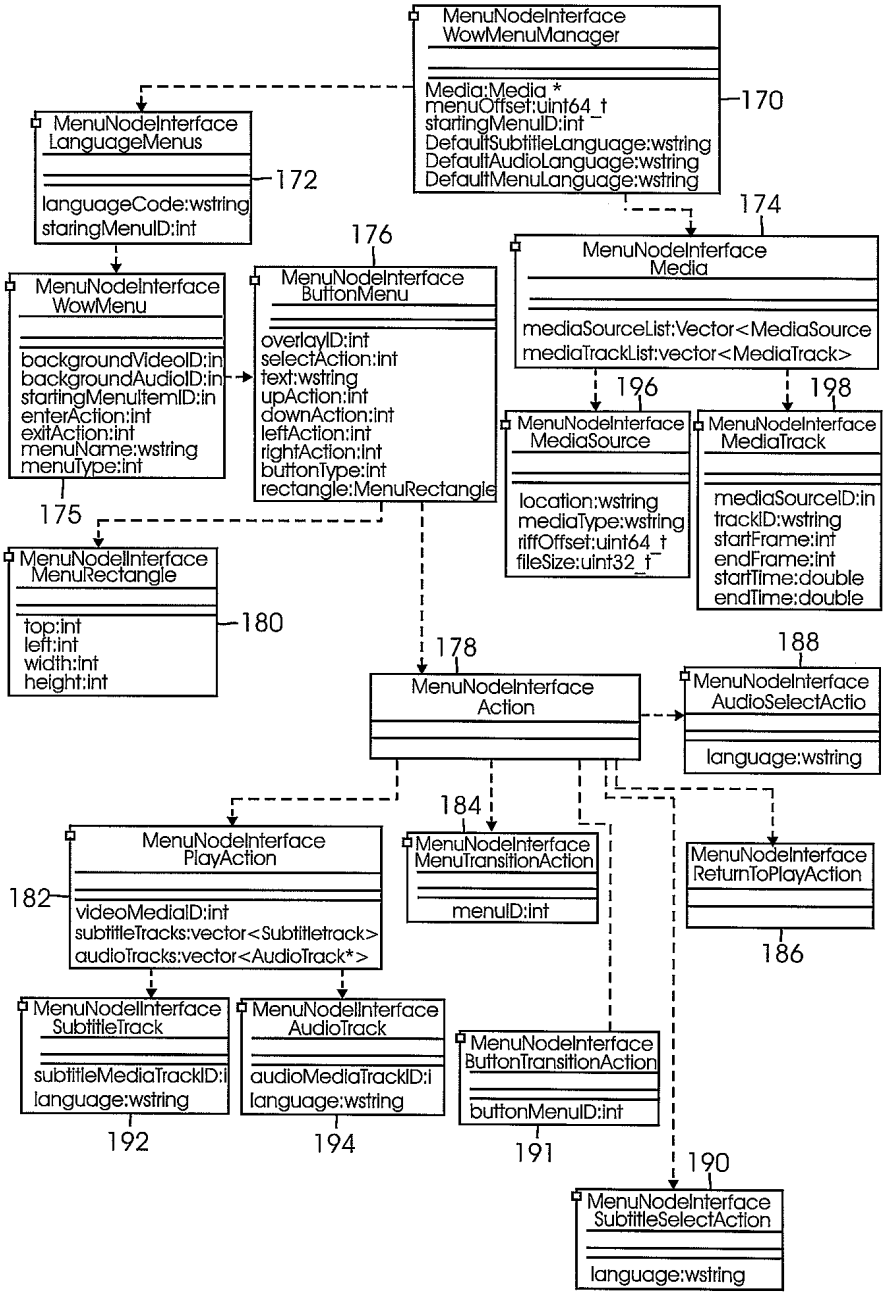
도면2da


MenuNodeInterface
MetaData 150
-m_version:wchar_t
MenuNodeInterface
PixelAspectRatioMetaData
-m_track:wchar_t*
-m_width:int
-m_height:int
trackID:const wchar_t*
width:int
height:int
152a
MenuNodeInterface
EncoderURIMetaData
-m_text:wchar_t*
text:const wchar_t*
152b
MenuNodeInterface
VideoResolutionMetaData
-m_track wchar_t*
-m_width:int
-m_height:int
trackID:const wchar_t*
width:int
height:int
152e
152g
MenuNodeInterface
CreatorMetaData
-m_name:wchar_t*
-m_role:wchar_t*
-m_character:wchar_t*
name:const wchar_t*
role:const wchar_t*
character:const wchar_t
MenuNodeInterface
creatorToolMetaData
-m_text:wchar_t*
text:const wchar_t*
152i
MenuNodeInterface
RuntimeMetaData
-m_track:wchar_t*
-m_length:int
trackID:const wchar_t
length:int
152k
MenuNodeInterface
RightsMetaData
-m_text:wchar_t*
-m_languageCode:wchar_t*
languageCode:constwchar_t
text:const wchar_t*
152j
MenuNodeInterface
InputSourceMetaData
-m_track:wchar_t*
-m_text:wchar_t*
trackID:const wchar_t
text:const wchar_t*
152p
152c
MenuNodeInterface
CodecSettingsMetaData
-m_track:wchar_t*
-m_settings:wchar_t*
-m_pass:int
trackID:const wchar_t*
pass:int
settings:const wchar_t*
MenuNodeInterface
PublisherMetaData
-m_text:wchar_t*
text:const wchar_t
152f
152m
MenuNodeInterface
CodeInfoMetaData
-m_track:wchar_t*
-m_name:wachar_t*
-m_codeVersion:wachar_t*
trackID:const wchar_t*
name:const wchar_t*
codecVersion:const wchar_t
152q
MenuNodeInterface
FileIDMetaData
-m_text:wchar_t*
text:const wchar_t
MenuNodeInterface
FrameTypeMetaData
-m_pass:int
-m_track:wchar_t*
-m_numOfIFrames:int
-m_numOfBFrames:int
-m_numOfPFrames:int
-m_numOfSFrames:int
trackID:const wchar_t
pass:int
numOfIFrames:int
numOfBFrames:int
numOfPFrames:int
numOfSFrames:int
152d
MenuNodeInterface
GenreMetaData
-m_text:wchar_t*
text:const wchar_t
152h
152o
MenuNodeInterface
FrameRateMetaData
-m_track wchar_t*
-m_rate:int
-m_scale:int
trackID:const wchar_t
rate:int
scale:int
MenuNodeInterface
TypeMetaData
-m_text:wchar_t*
text:const wchar_t
152r
MenuNodeInterface
EncoderNameMetaData
-m_text:wchar_t*
text:const wchar_t*
152n
MenuNodeInterface
QuantizeMetaData
-m_text:wchar_t*
-m_pass:int
-m_quantizerDistribution:uint16_t*
-m_size:int
trackID:const wchar_t*
pass:int
quantizerDistribution:const uint16_t
quantizerDistributionSize:int
152l
MenuNodeInterface
TitleMetaData
-m_text:wchar_t*
-m_languageCode:wchar_t*
languageCode:constwchar_t*
text:const wchar_t*
152s
MenuNodeInterface
CertLevelMetaData
-m_text:wchar_t*
text:const wchar_t
152t

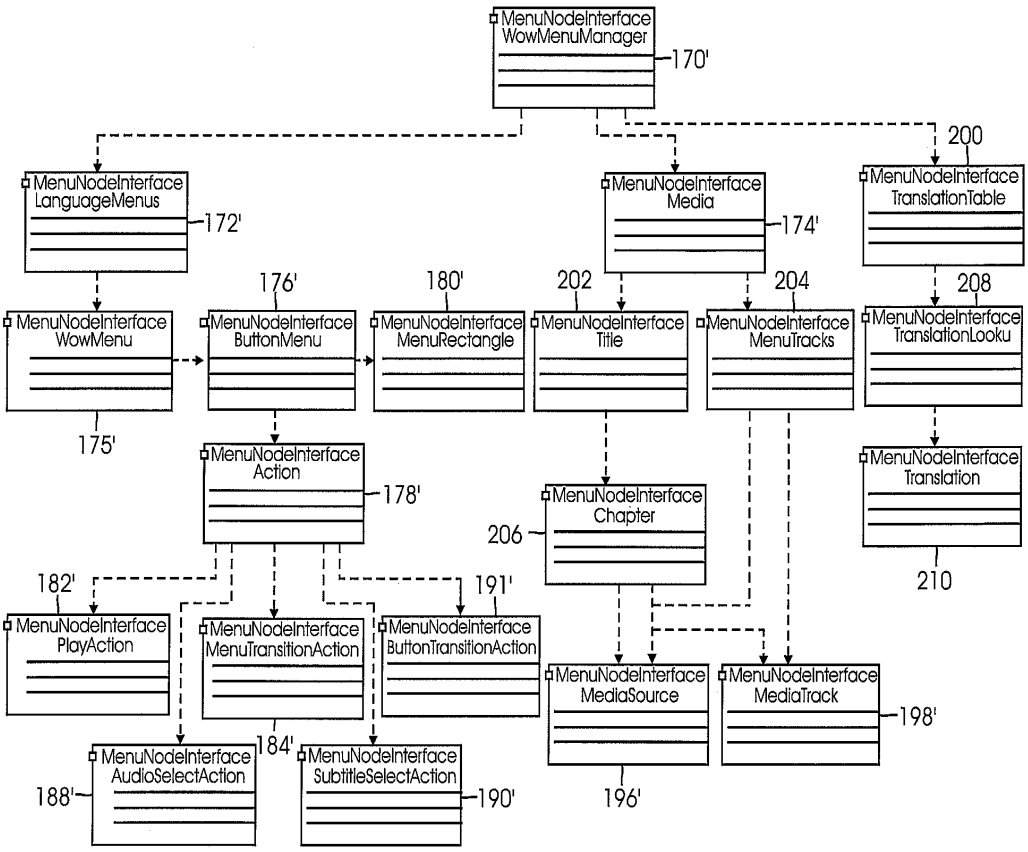
도면2e



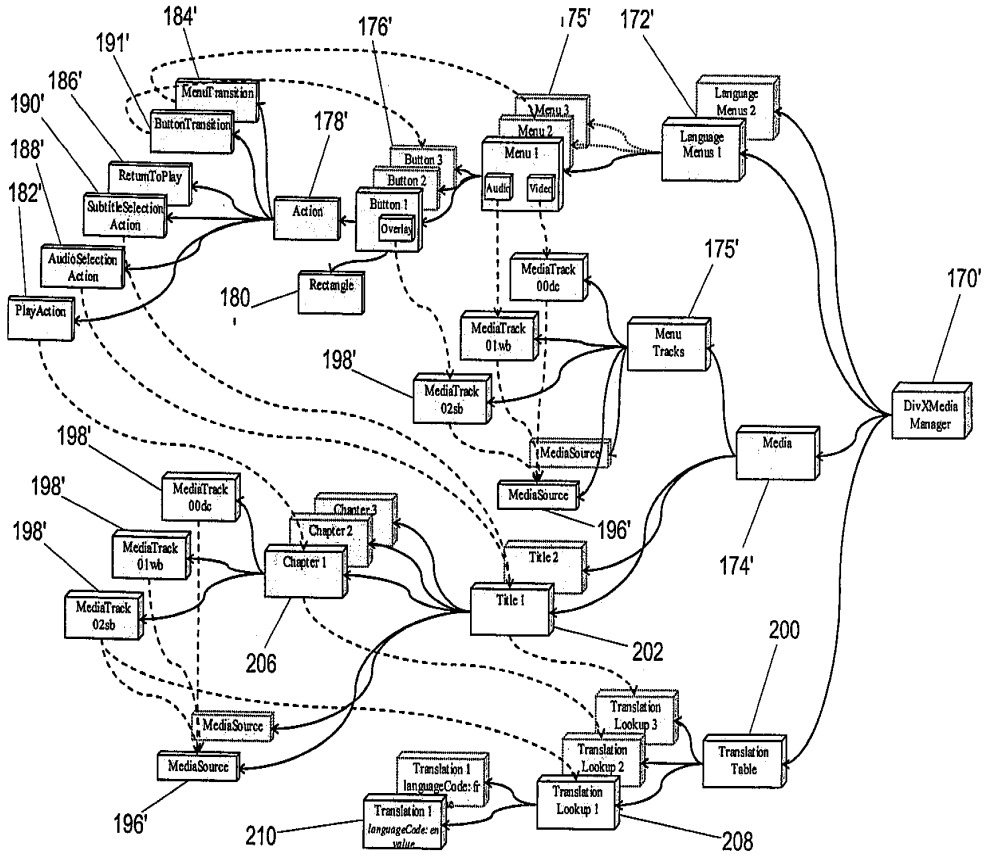
도면2f


MenuNodeInterface
WowMenuManager
Media:Media *
menuOffset:uint64_t
startingMenuID:int
DefaultSubtitleLanguage:wstring
DefaultAudioLanguage:wstring
DefaultMenuLanguage:wstring
170
MenuNodeInterface
LanguageMenus
languageCode:wstring
staringMenuID:int
172
MenuNodeInterface
WowMenu
backgroundVideoID:in
backgroundAudioID:in
startingMenuItemID:in
enterAction:int
exitAction:int
menuName:wstring
menuType:int
175
176
MenuNodeInterface
ButtonMenu
overlayID:int
selectAction:int
text:wstring
upAction:int
downAction:int
leftAction:int
rightAction:int
buttonType:int
rectangle:MenuRectangle
174
MenuNodeInterface
Media
mediaSourceList:Vector<MediaSource
mediaTrackList:vector<MediaTrack>
196
198
MenuNodeInterface
MediaSource
location:wstring
mediaType:wstring
riffOffset:uint64_t
fileSize:uint32_t
MenuNodeInterface
MediaTrack
mediaSourceID:in
trackID:wstring
startFrame:int
endFrame:int
startTime:double
endTime:double
MenuNodeInterface
MenuRectangle
top:int
left:int
width:int
height:int
180
178
MenuNodeInterface
Action
188
MenuNodeInterface
AudioSelectActio
language:wstring
184
182
MenuNodeInterface
PlayAction
videoMediaID:int
subtitleTracks:vector<Subtitletrack>
audioTracks:vector<AudioTrack*>
MenuNodeInterface
MenuTransitionAction
menuID:int
MenuNodeInterface
ReturnToPlayAction
186
MenuNodeInterface
SubtitleTrack
subtitleMediaTrackID:i
language:wstring
192
MenuNodeInterface
AudioTrack
audioMediaTrackID:i
language:wstring
194
MenuNodeInterface
ButtonTransitionAction
buttonMenuID:int
191
190
MenuNodeInterface
SubtitleSelectAction
language:wstring

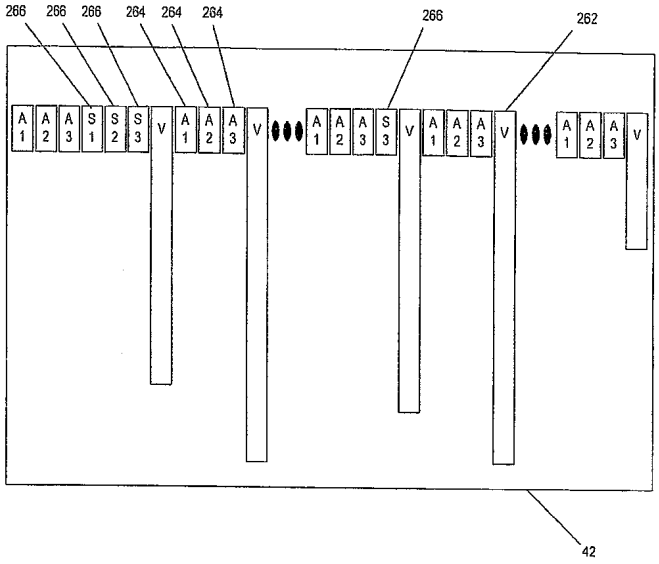
도면2g


MenuNodeInterface WowMenuManager
170'
200
MenuNodeInterface LanguageMenus
172'
MenuNodeInterface Media
174'
MenuNodeInterface TranslationTable
176'
180'
202
204
208
MenuNodeInterface WowMenu
MenuNodeInterface ButtonMenu
MenuNodeInterface MenuRectangle
MenuNodeInterface Title
MenuNodeInterface MenuTracks
MenuNodeInterface TranslationLooku
175'
MenuNodeInterface Action
178'
MenuNodeInterface Chapter
206
MenuNodeInterface Translation
210
182'
191'
MenuNodeInterface PlayAction
MenuNodeInterface MenuTransitionAction
MenuNodeInterface ButtonTransitionAction
MenuNodeInterface MediaSource
MenuNodeInterface MediaTrack
198'
184'
MenuNodeInterface AudioSelectAction
MenuNodeInterface SubtitleSelectAction
188'
190'
196'

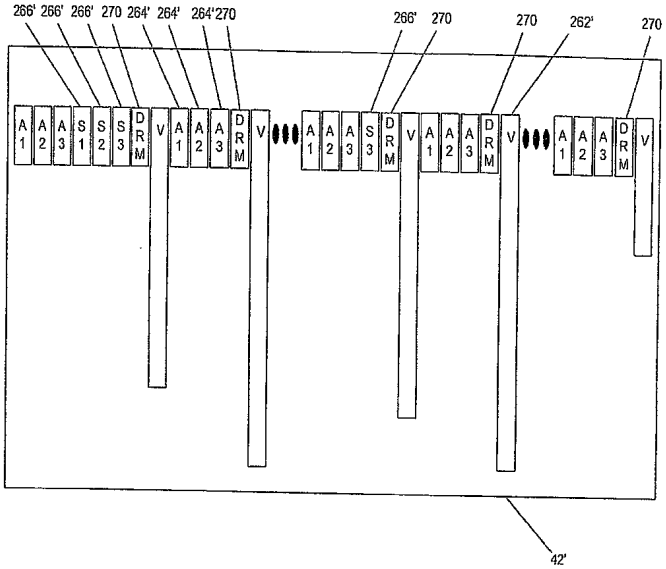
도면2ga


184'
191'
176'
175'
172'
186'
190'
178'
188'
182'
MenuTransition
ButtonTransition
ReturnToPlay
SubtitleSelection Action
Action
Button 3
Button 2
Button 1
Overlay
Menu 3
Menu 2
Menu 1
Audio
Video
Language Menus 2
Language Menus 1
AudioSelection Action
PlayAction
180
Rectangle
MediaTrack 00dc
MediaTrack 01wb
MediaTrack 02sb
175'
Menu Tracks
170'
DivXMedia Manager
198'
MediaSource
MediaSource
196'
Media
174'
MediaTrack 00dc
MediaTrack 01wb
MediaTrack 02sb
Chapter 3
Chapter 2
Chapter 1
Title 2
Title 1
206
202
200
MediaSource
MediaSource
196'
Translation 1 languageCode: fr
Translation 1 languageCode: en value
210
Translation Lookup 3
Translation Lookup 2
Translation Lookup 1
Translation Table
208

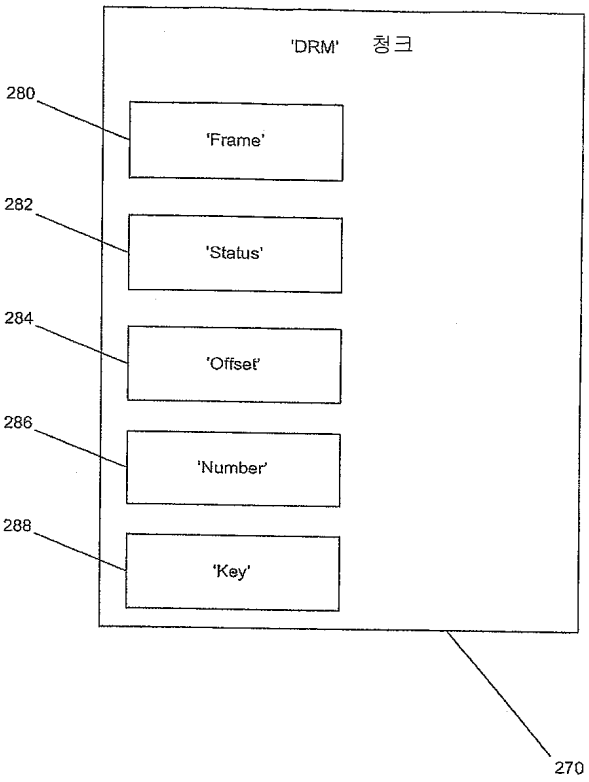
도면2h


266
266
266
264
264
264
266
262
A1
A2
A3
S1
S2
S3
V
42

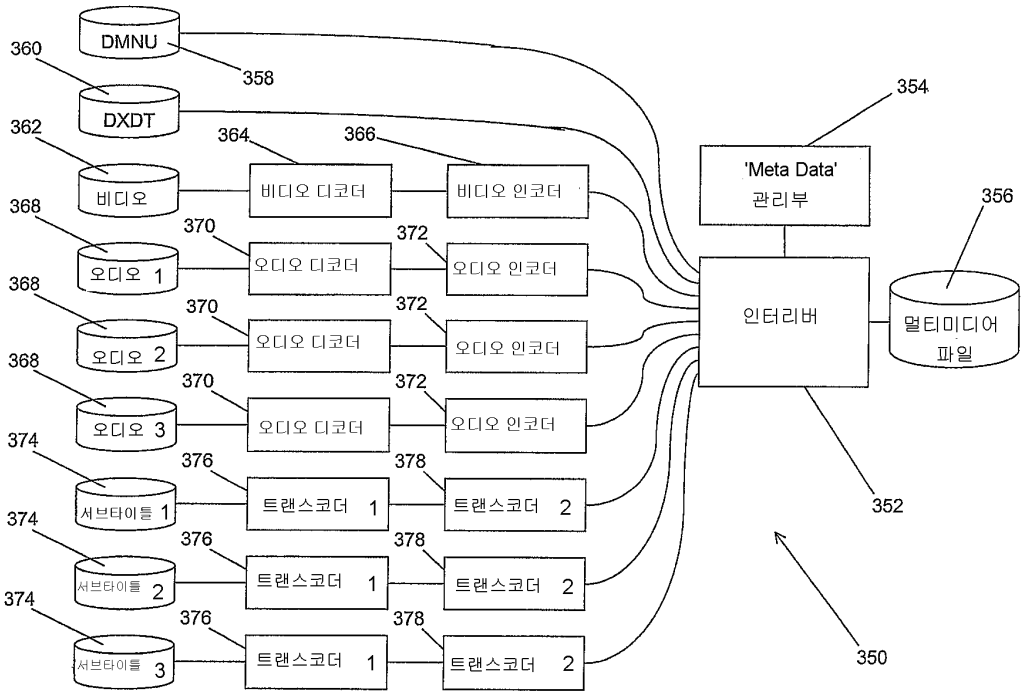
도면2i


266' 266' 266' 270 264' 264' 264' 270
266' 270
270
262'
270
A1 A2 A3 S1 S2 S3 DRM V
A1 A2 A3 DRM V
A1 A2 A3 S3 DRM V
A1 A2 A3 DRM V
A1 A2 A3 DRM V
42'

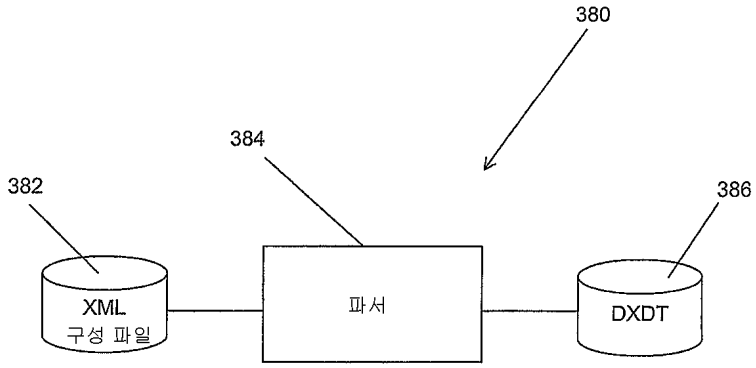
도면2j


'DRM' 청크
280
'Frame'
282
'Status'
284
'Offset'
286
'Number'
288
'Key'
270

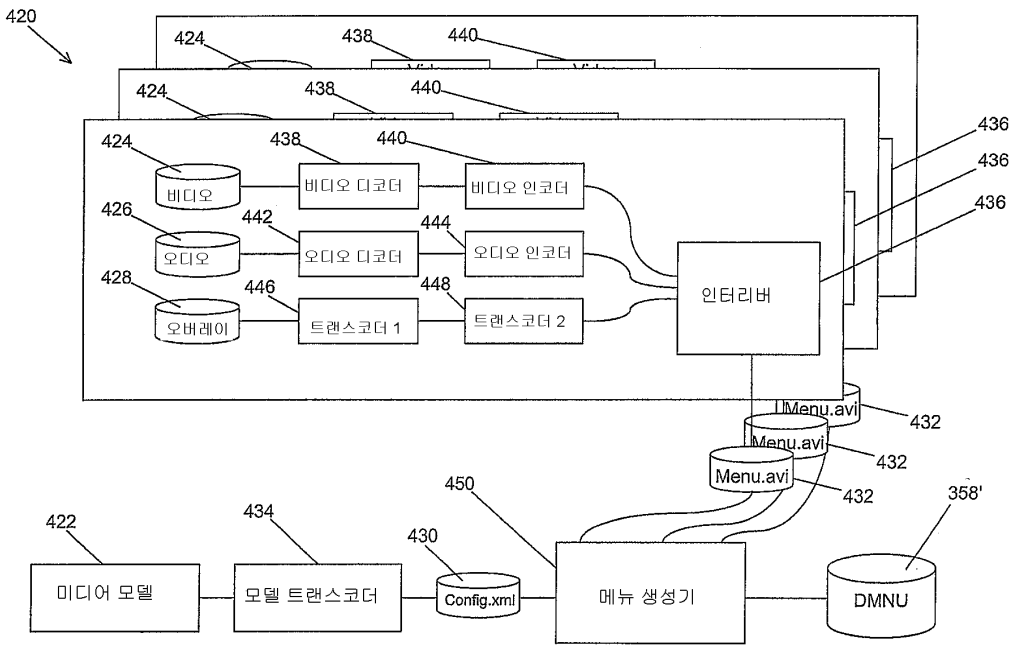
도면3a


DMNU
360
358
DXDT
362
364
366
354
'Meta Data'
관리부
비디오
비디오 디코더
비디오 인코더
356
368
370
372
오디오 1
오디오 디코더
오디오 인코더
인터리버
멀티미디어
파일
오디오 2
오디오 3
374
376
378
서브타이틀 1
트랜스코더 1
트랜스코더 2
352
서브타이틀 2
서브타이틀 3
350

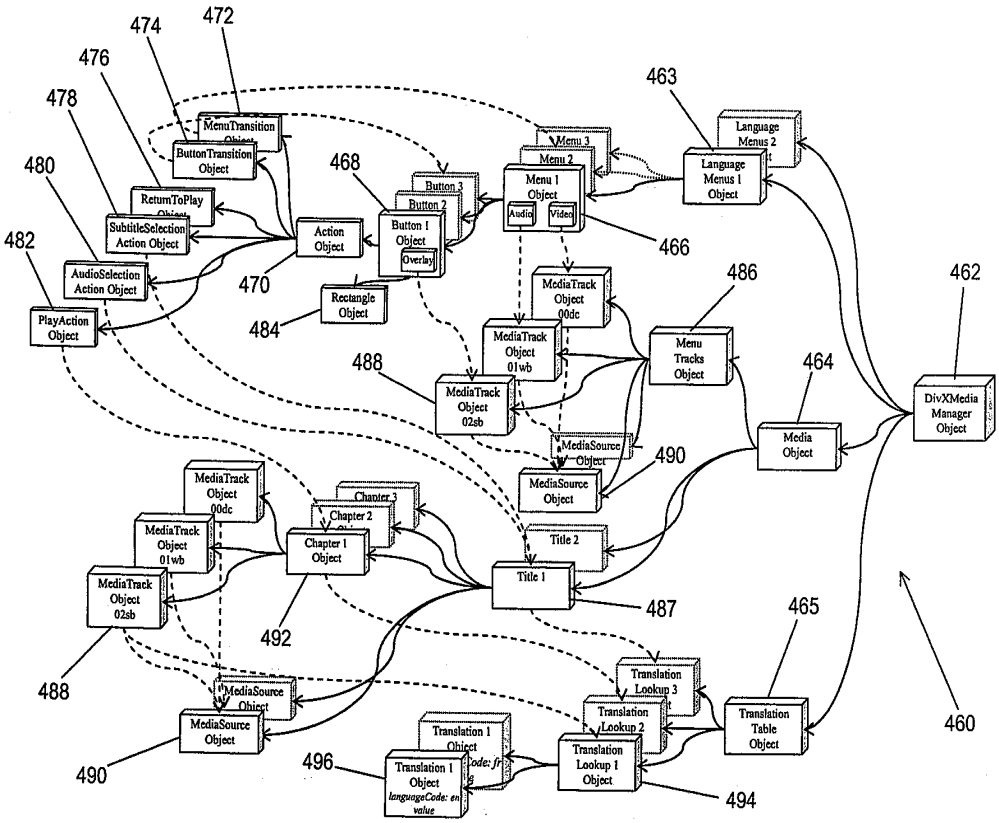
도면3b


380
384
382
386
XML
구성 파일
파서
DXDT

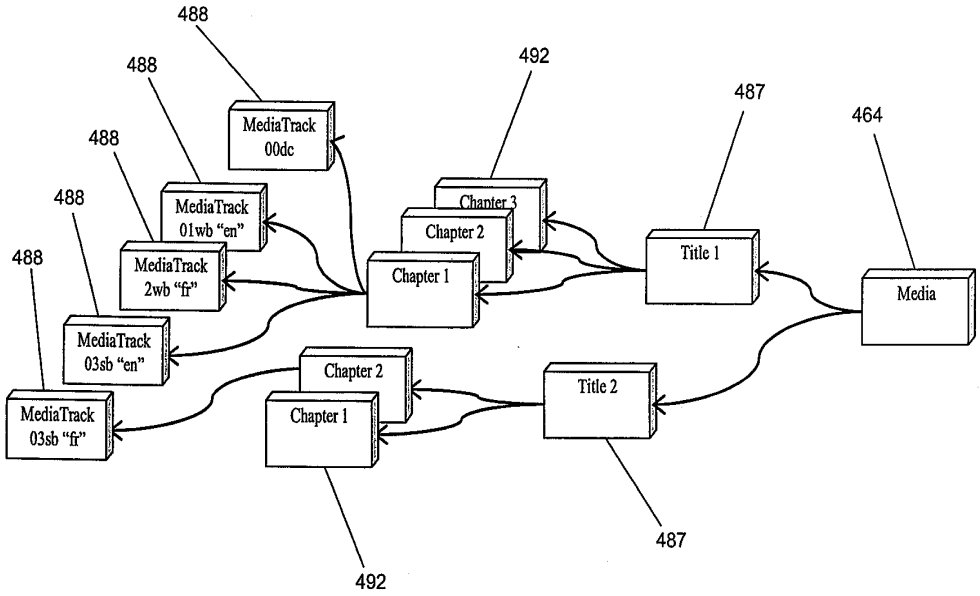
도면3c


420
424
438
440
436
비디오
비디오 디코더
비디오 인코더
426
442
444
오디오
오디오 디코더
오디오 인코더
428
446
448
오버레이
트랜스코더 1
트랜스코더 2
인터리버
Menu.avi
432
450
358'
422
434
430
미디어 모델
모델 트랜스코더
Config.xml
메뉴 생성기
DMNU

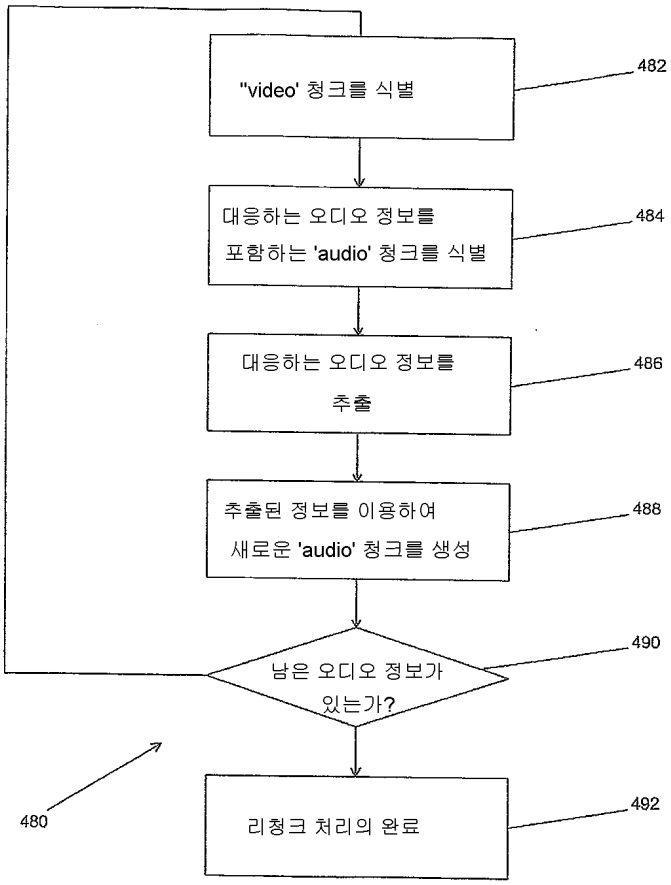
도면3d


474
472
476
463
478
480
468
482
470
484
486
462
488
464
490
487
465
492
460
494
496
MenuTransition Object
ButtonTransition Object
ReturnToPlay Object
SubtitleSelection Action Object
AudioSelection Action Object
PlayAction Object
Action Object
Button 1 Object
Rectangle Object
Menu 1 Object
Language Menus 1 Object
Language Menus 2
MediaTrack Object 00dc
MediaTrack Object 01wb
MediaTrack Object 02sb
Menu Tracks Object
Media Object
DivXMedia Manager Object
MediaSource Object
Chapter 1 Object
Title 1
Title 2
Translation Lookup 1 Object
Translation Lookup 2
Translation Lookup 3
Translation Table Object
Translation 1 Object

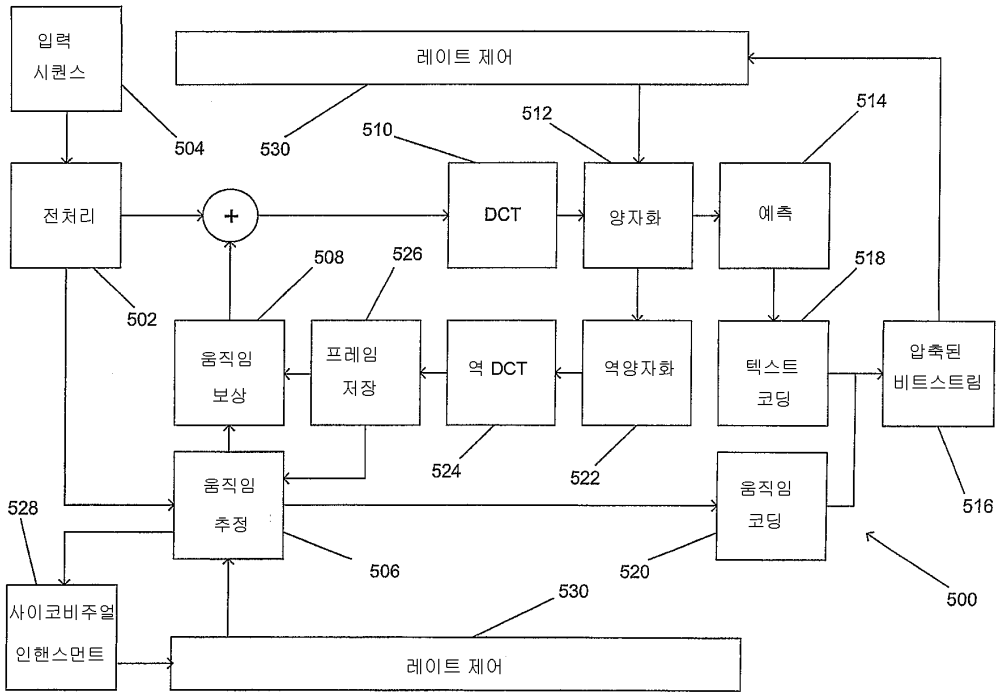
도면3da


488
488
488
488
488
MediaTrack 00dc
MediaTrack 01wb "en"
MediaTrack 2wb "fr"
MediaTrack 03sb "en"
MediaTrack 03sb "fr"
492
Chapter 3
Chapter 2
Chapter 1
Chapter 2
Chapter 1
487
Title 1
Title 2
464
Media
487
492

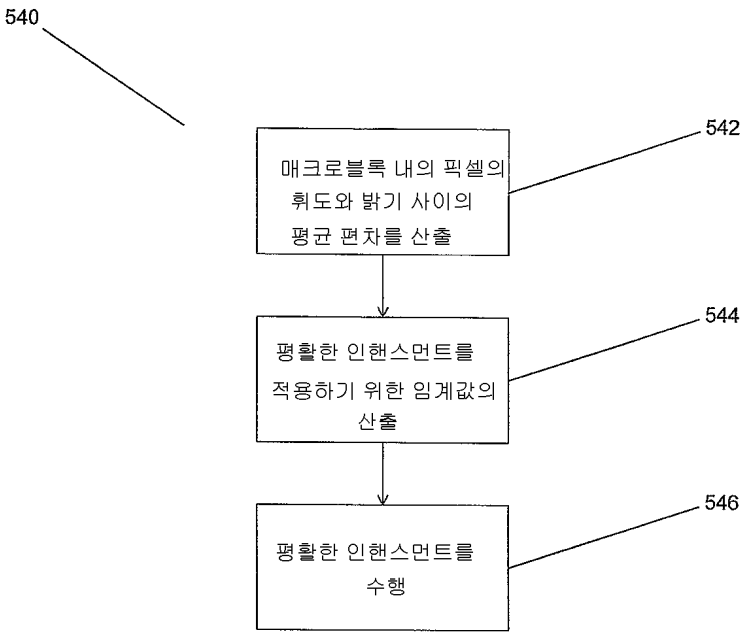
도면3e


"video' 청크를 식별
482
대응하는 오디오 정보를 포함하는 'audio' 청크를 식별
484
대응하는 오디오 정보를 추출
486
추출된 정보를 이용하여 새로운 'audio' 청크를 생성
488
남은 오디오 정보가 있는가?
490
480
리청크 처리의 완료
492

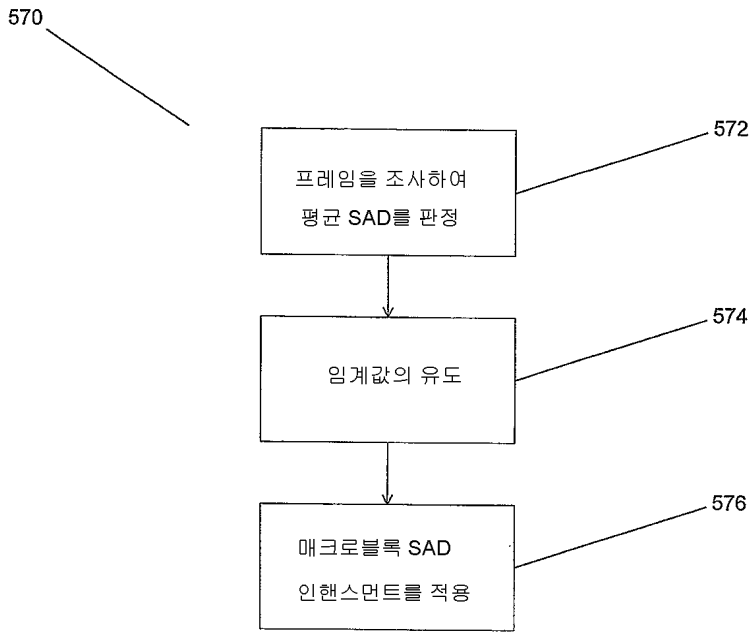
도면3f


입력
시퀀스
레이트 제어
504
530
510
512
514
전처리
+
DCT
양자화
예측
508
526
518
502
움직임
보상
프레임
저장
역 DCT
역양자화
텍스트
코딩
압축된
비트스트림
움직임
추정
524
522
움직임
코딩
528
516
506
520
500
530
사이코비주얼
인핸스먼트
레이트 제어

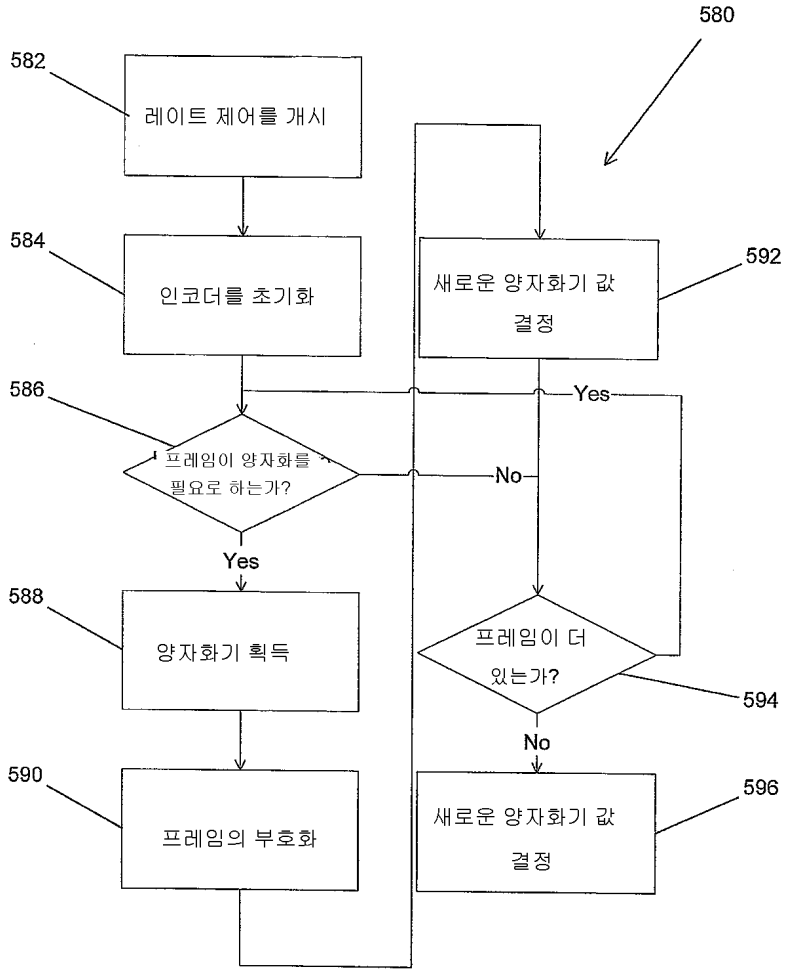
도면3g


540
매크로블록 내의 픽셀의
휘도와 밝기 사이의
평균 편차를 산출
542
평활한 인핸스먼트를
적용하기 위한 임계값의
산출
544
평활한 인핸스먼트를
수행
546

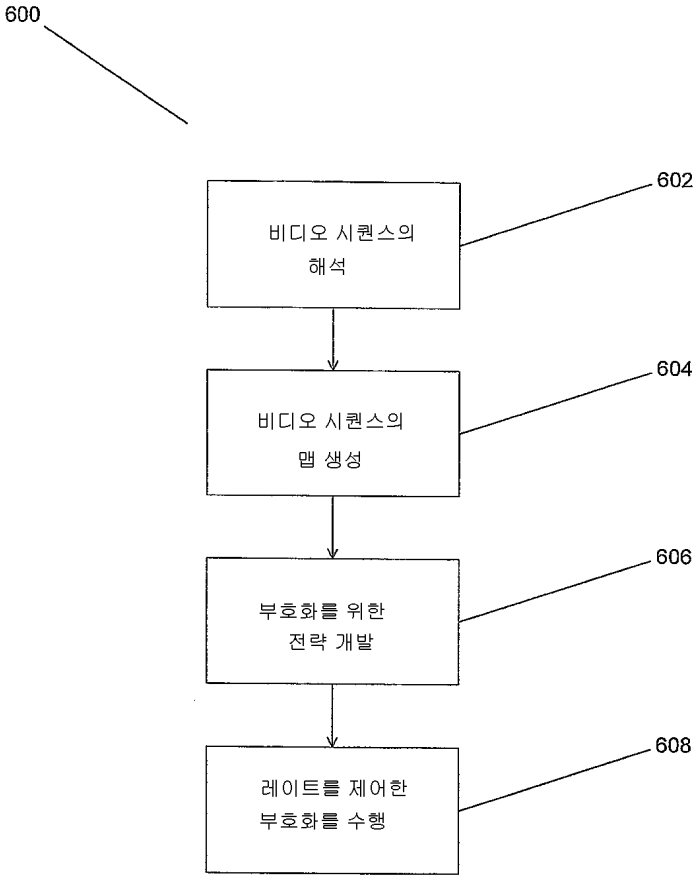
도면3h


570
572
프레임을 조사하여
평균 SAD를 판정
574
임계값의 유도
576
매크로블록 SAD
인핸스먼트를 적용

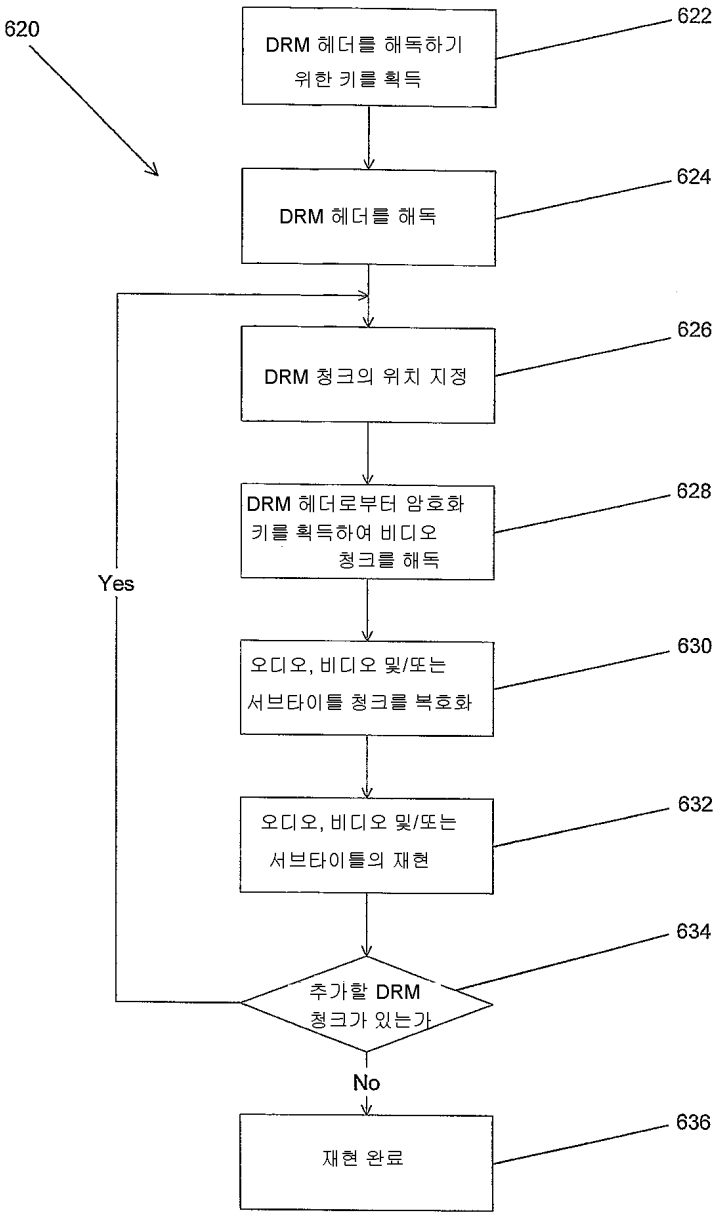
도면3i


580
582
레이트 제어를 개시
584
인코더를 초기화
586
프레임이 양자화를
필요로 하는가?
Yes
No
588
양자화기 획득
590
프레임의 부호화
592
새로운 양자화기 값
결정
Yes
프레임이 더
있는가?
594
No
596
새로운 양자화기 값
결정

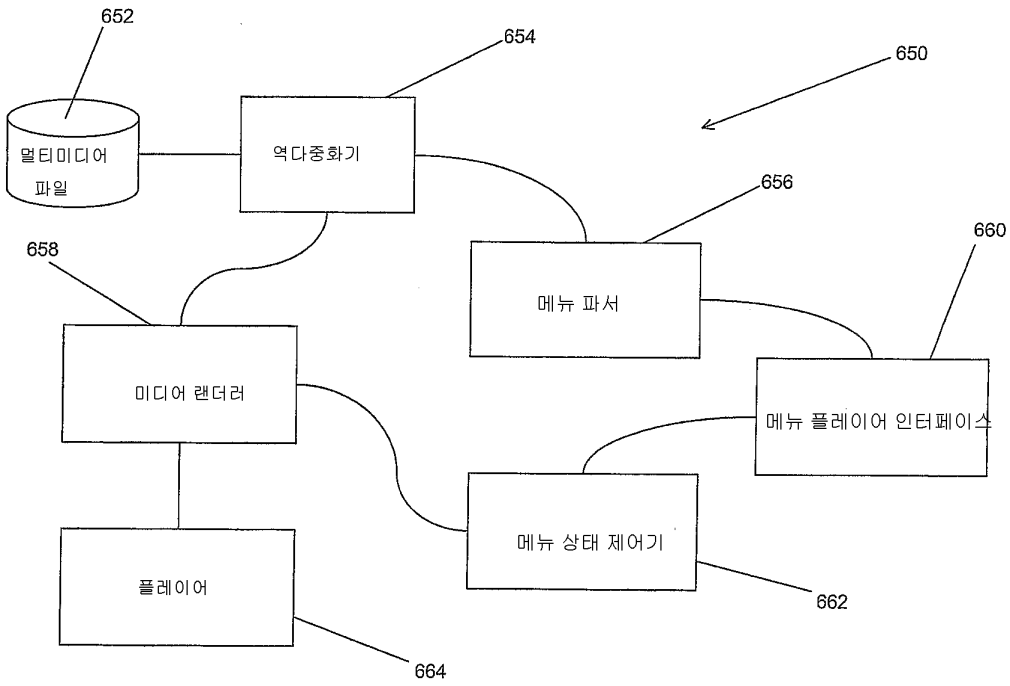
도면3j


600
602
비디오 시퀀스의
해석
604
비디오 시퀀스의
맵 생성
606
부호화를 위한
전략 개발
608
레이트를 제어한
부호화를 수행

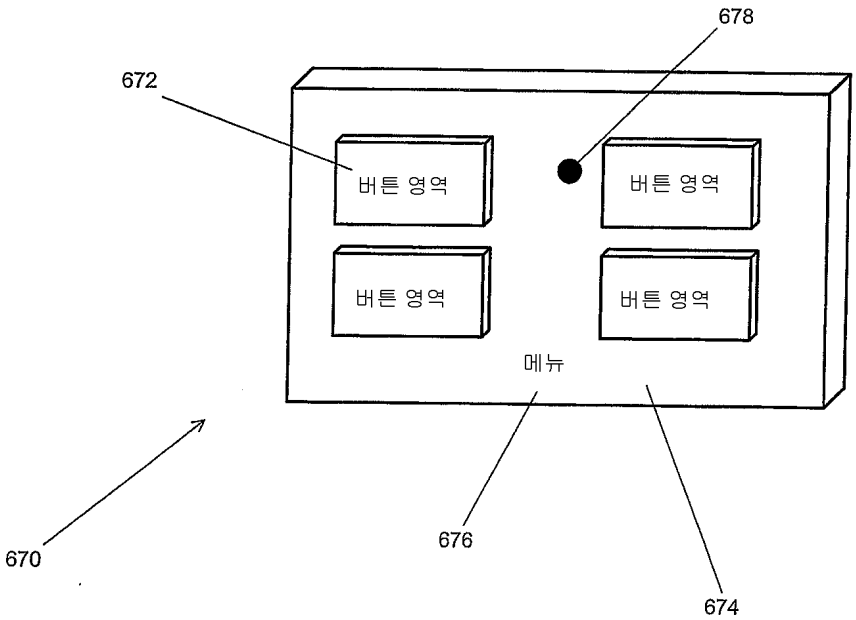
도면4a


620
DRM 헤더를 해독하기 위한 키를 획득
622
DRM 헤더를 해독
624
DRM 청크의 위치 지정
626
DRM 헤더로부터 암호화 키를 획득하여 비디오 청크를 해독
628
Yes
오디오, 비디오 및/또는 서브타이틀 청크를 복호화
630
오디오, 비디오 및/또는 서브타이틀의 재현
632
추가할 DRM 청크가 있는가
634
No
재현 완료
636

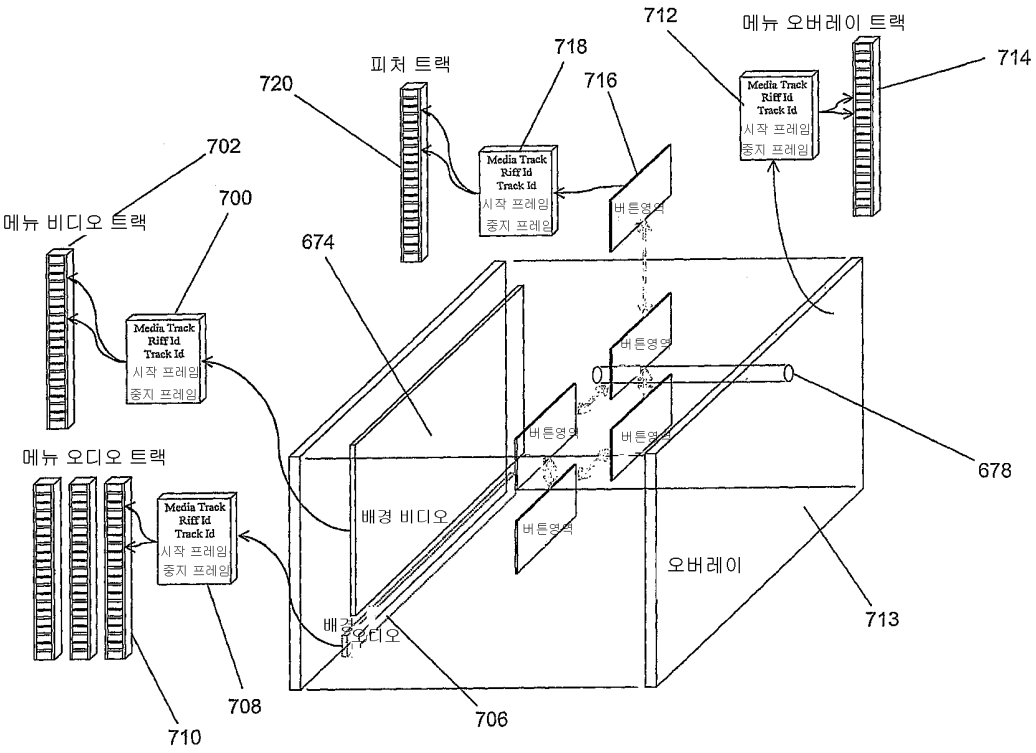
도면4b


652
654
650
멀티미디어
파일
역다중화기
656
658
660
메뉴 파서
미디어 랜더러
메뉴 플레이어 인터페이스
메뉴 상태 제어기
662
플레이어
664

도면4c


678
672
버튼 영역
버튼 영역
버튼 영역
버튼 영역
메뉴
670
676
674

도면4d


712
메뉴 오버레이 트랙
714
718
716
720
피처 트랙
Media Track
Riff Id
Track Id
시작 프레임
중지 프레임
702
700
메뉴 비디오 트랙
674
버튼영역
678
메뉴 오디오 트랙
배경 비디오
오버레이
713
배경 오디오
708
706
710