



(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。 Int. Cl.

H04N 7/24 (2006.01)

H04N 13/00 (2006.01)

(11) 공개번호 10-2007-0011340

(43) 공개일자 2007년01월24일

(21) 출원번호 10-2006-7020050

(22) 출원일자 2006년09월27일

심사청구일자 없음

번역문 제출일자 2006년09월27일

(86) 국제출원번호 PCT/MX2004/000011

(87) 국제공개번호 WO 2005/083636

국제출원일자 2004년02월27일

국제공개일자 2005년09월09일

(71) 출원인 티디비전 코포레이션 에스.에이. 데 씨.브이.
멕시코 디.에프. 02800 멕시코 씨오엘. 누에바 산타 마리아 피나 201-에이

(72) 발명자 구티에레스 노벨로, 마누엘 라파엘
멕시코 디.에프. 02800 멕시코 씨오엘. 누에바 산타 마리아 피나201-에이

(74) 대리인 윤석운

전체 청구항 수 : 총 12 항

(54) 입체적 3D-비디오 이미지 디지털 코딩 시스템 및 방법

(57) 요약

디지털 비디오 스트림으로부터 3차원 이미지들을 얻기 위해서, 현재의 MPEG2 코더들에 특정 변형들이 만들어지고; 코딩 프로세스의 상이한 부분에서의 소프트웨어와 하드웨어 변화들이 이루어진다.

실제로, 비디오 데이터 스트림의 구조 및 비디오 시퀀스(video_sequence)는 비트 레벨에서 TDVision® 기술 이미지 형태를 식별하기 위해 필요한 플래그를 포함하도록 변형된다.

소프트웨어에 관해서, 변형들이 이하의 프로세스 즉, 비디오 시퀀스 헤더(video_sequence header), 식별 플래그 변형, 데이터 필드 변형, 및 이미지 필드 변형에서 이루어진다.

하드웨어에 관해서, 전자적 비교가 좌우측 채널들의 이미지들 간에 이루어지고, 그 차이가 B형 이미지로서 처리되고, 즉 에러 교정이 수행되고, 그 결과들은 TDVision® 기술 식별자를 갖는 임시 버퍼에 저장된다.

정보가 코딩될 때, DSP가 MPEG2-호환성 압축 비디오 스트림을 만들어 내기 위해서 예측, 비교, 양자화, 및 DCT 기능 적용 프로세스들을 실행하기 위해 사용된다.

대표도

도 5

특허청구의 범위

청구항 1.

소프트웨어 알고리즘 및 이와 관련된 하드웨어 내에서의 변경을 포함하는 MPEG형 코딩 프로세스로 구성된 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템에 있어서, 코딩 프로세스에서의 소프트웨어 변형 단계, 즉

비디오 구조들을 변형하는 단계;

비디오 데이터 스트림의 비디오 시퀀스 헤더(video_sequence header)를 변형하는 단계;

비트-레벨 식별 플래그(bit-level identification flag)들을 변형하는 단계;

데이터 필드들을 변형하는 단계; 및

이미지 필드들을 변형하는 단계를 포함하는 것을 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 2.

제1항에 있어서, 비디오 시퀀스 비디오 데이터 스트림(video_sequence video data stream)이 MPEG1와 MPEG2에 적용되고 다음의 구조, 즉

시퀀스 헤더(sequence_header);

시퀀스 확장(sequence_extension);

확장 및 사용자 데이터 (0)(extension_and_user_data(0));

픽처 헤더의 그룹(group_of_pictures_header);

확장 및 사용자 데이터(1)(extension_and_user_data(1));

픽처 헤더(picture header);

픽처 코딩 확장(picture_coding_extension);

확장 및 사용자 데이터(2)(extensions_and_user_data(2));

픽처 데이터(picture_data);

슬라이스(slice);

매크로블록(macroblock);

모션 벡터(motion_vectors);

코딩 블록 패턴(coded_block_pattern);

블록(block);

최종 시퀀스 코드(sequence_end_code)로 구성되는 것을 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템..

청구항 3.

제1항에 있어서, 2D-시스템과의 역 호환성을 얻기 위해 논리 "and"가 0111로 실행되는, 종횡비 정보 필드(aspect_ratio_information field)에서 비디오 스트림에 대한 더 높은 정보 레벨; 및

2D-시스템과의 역 호환성을 얻기 위해 논리 "and"가 0111로 실행되는 프레임 비율 코드(frame_rate_code)를 제공하는 시퀀스 헤더 구조(sequence_header structure)를 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 4.

제2항에 있어서, 확장 및 사용자 데이터(i) 구조(extension_and_user_data(i) structure)가 다른 구조의 저장을 위한 컨테이너가 되고 몇몇 경우에는 빈 구조가 되며;

i의 값은 0 또는 2일 수 있는데, 만일 그것이 0와 같다면, 확장 데이터(extension_data)는 시퀀스 확장(sequence_extension)을 따르고 확장 데이터(i)(extension_data(i))는 시퀀스 디스플레이 확장(sequence_display_extension) 또는 시퀀스 스케일러블 확장(sequence_scalable_extension)을 포함하고;

i = 2이면, 다음 구조는 양자화 매트릭스 확장() (quant_matrix_extension()), 저작권 확장()(copyright_extension()), 픽처 공간 스케일러블 확장()(picture_spatial_scalable_extension()), 또는 픽처 임시 스케일러블 확장(picture_temporal_scalable_extension)을 포함하는 픽처 코딩 확장(picture_coding_extension)인 것을 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 5.

제1항에 있어서, 비디오를 바르게 표시하는데 유용하고;

비디오 포맷 필드(video_format field)에서는 111로 식별되는 코딩된 내용에 관한 정보를 제공하는 시퀀스 디스플레이 확장() 구조 (sequence_display_extension() structure)를 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 6.

제2항에 있어서, 공간 스케일러빌리티 모드 01과 임시 스케일러빌리티 11;

층 아이디(layer_id);

하부층 예측 수직 사이즈(lower_layer_prediction_vertical_size);

마커 비트(marker_bit);

하부층 예측 수직 사이즈(lower_layer_prediction_vertical_size);

수평 서브샘플링 계수 m(horizontal_subsampling_factor_m);

수평 서브샘플링 계수 n(horizontal_subsampling_factor_n);

수직 서브샘플링 계수 m(vertical_subsampling_factor_m);

수직 서브샘플링 계수 n(vertical_subsampling_factor_n);

픽처 멀티플렉스 인에이블(picture_mux_enable);

멀티플렉스 단계적 시퀀스(mux_to_progressive_sequence);

픽처 멀티플렉스 명령(picture_mux_order);

픽처 멀티플렉스 계수(picture_mux_factor)를 갖는 보완계층들 및 기본계층을 포함할 때, 정의를 증가시키는 추가 데이터를 포함하는 시퀀스 스케일러블 확장 구조(sequence_scalable_extension structure)를 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 7.

제2항에 있어서, 이미지 코딩 형태 필드(image coding type field)(픽처 코딩 형태(picture_coding_type)), 예측 코딩(P)을 위한 010, 양방향 예측 코딩(B)을 위한 011;

비디오 임시 메모리(비디오 버퍼) 검증 메커니즘으로 정의되는 픽처 헤더 구조(picture_header structure)를 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 8.

제1항에 있어서, 픽처 구조 필드(picture_structure field)는 이미지가 필드 즉, 00 예약 형태 TDVision 포맷 이미지, 01 상부 필드, 10 하부 필드, 11 바이-프레임 이미지로 나뉘는지, 또는 그것이 전체 프레임인지를 특정하고;

또한 다음의 필드, 즉

복합 디스플레이 플래그(composite_display_flag);

V축(V_axis);

필드 시퀀스(field_sequence);

서브 캐리어(sub_carrier);

버스트 진폭(burst_amplitude);

서브 캐리어 위상(sub_carrier_phase)을 정의하는 것을 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 9.

관련된 하드웨어 내에서의 변경 및 소프트웨어 알고리즘을 포함하는 MPEG형 코딩 프로세스로 구성된 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템에 있어서, 코딩 프로세스에서의 하드웨어 변형 단계는,

2개의 독립적인 비디오 입력 채널들은 좌우의 채널들을 전자적으로 비교할 때 인에이블되는 단계;

좌우의 채널들 간의 비교 차이 처리 단계;

입체적인 이미지의 이전 및 동시적 디스플레이를 제공하기 위해 메모리가 두배로 되는 단계;

DSP는 좌우측 이미지 신호들 둘다를 위한 동시적 엔트리 버퍼(entry buffer)들을 비교하기 위해 인에이블되는 단계를 포함하는 것을 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 10.

제9항에 있어서, TDVision® 카메라의 기존의 입체적인 좌우측 신호에 대응하는, 2개의 독립적인 이미지 신호들의 동시적 입력을 허용하는 2개의 독립적인 비디오 채널들을 인에이블하는 것을 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 11.

제9항에 있어서, DSP 버퍼의 입력 메모리를 두 배로 하는 것을 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

청구항 12.

제9항에 있어서, 하드웨어 변형이,

싱글 비디오 입력 채널의 함수로서 정규 형태 (MPEG)로의 프레임 코딩 단계;

둘 다의 신호들을 획득하는 단계;

좌우측 신호들을 전자적으로 비교하는 단계;

좌우측 신호들 간의 에러 차이를 얻는 단계;

상기 차이를 임시 버퍼에 저장하는 단계;

좌측 신호와 관련하여 루마와 크로마에서의 에러 정정을 계산하는 단계;

DCT를 적용하는 단계; 및

픽처 데이터3D 구조(picture_data3D() structure) 내에서, 정보를 B형 블록에 저장하는 단계로 구성되는 것을 특징으로 하는 입체적 3D-비디오 이미지 디지털 코딩 방법 및 시스템.

명세서

기술분야

본 발명은 3DVisor™ 장치에서 입체적 비디오 이미지 디스플레이, 특히 표준화된 압축 기술들을 사용하여 3차원의 정보 저장을 허용하는 디지털 데이터 압축 시스템에 의한 비디오 이미지 코딩 방법에 관한 것이다.

배경기술

현재, 데이터 압축 기술들은 이미지 또는 일련의 이미지들의 표현에서 비트 소모를 감소시키기 위해 사용된다. 표준화 작업들은 국제 표준 기구(International Standardization Organization)의 전문가들의 그룹에 의해 수행되었다. 현재, 이 방법들은 보통 JPEG (Joint Photographic Expert Group)와 MPEG(Moving Pictures Expert Group)로서 알려져 있다.

이 기술들의 공통적인 특징은 이미지 블록들이 블록들에 적절한 변환을 적용함으로써 처리된다는 것인데, 이것은 보통 이산 여현 변환(Discrete Cosine Transform:DCT)으로서 알려져 있다. 형성된 블록들은 양자화 처리에 제공되고, 그 때 가변 길이 코드로 코딩된다.

가변 길이 코드는 가역 처리인데, 이는 가변 길이 코드로 코딩된 것의 정확한 재구성을 허용한다.

디지털 비디오 디스플레이는 30 내지 75 Hz의 주파수로 연속해서 표시되거나 표현되는 특정 수의 이미지 프레임(30 내지 96 fps)들을 포함한다. 각 이미지 프레임은 특별한 시스템의 디스플레이 해상도에 따라, 픽셀 어레이에 의해 형성된 이미지이다. 예로서, VHS 시스템은 320개의 컬럼들과 480개의 로우들의 디스플레이 해상도를 갖고, NTSC 시스템은 720개의 컬럼들과 486개의 로우들의 디스플레이 해상도를 갖고, 그리고 고해상도 텔레비전 시스템(HDTV)은 1360개의 컬럼들과 1020개의 로우들의 디스플레이 해상도를 갖는다. 저해상도, 즉 320개의 컬럼 X 480개의 로우 VHS 포맷의 디지털화 형태를 기준으로, 2시간의 긴 영화는 100 기가바이트의 디지털 비디오 정보와 같을 수 있다. 비교시에, 종래의 콤팩트 광 디스크는 대략 0.6 기가바이트의 용량을 갖고, 자기 하드 디스크는 1 내지 2 기가바이트 용량을 가지며, 현재의 콤팩트 광 디스크들은 8 기가바이트 내지 그 이상의 용량을 갖는다.

이처럼 큰 정보량의 저장과 전달 한계들에 대응하여, 몇몇 표준 압축 프로세스들이 확립되었다. 이들 비디오 압축 기술들은 프레임간 픽셀들의 표현에 근거하는 프레임 단위(frame-by-frame) 압축을 제공하기 위해 일시적 공간 상관 관계로 이용되는 연속 이미지 프레임들 간의 유사한 특징들을 사용한다.

우리들이 영화관과 TV 스크린들에서 보는 모든 이미지들은 완전한 이미지(정적인 이미지, 사진과 같은)들을 고속으로 제공하는 원리에 근거한다. 이들이 30 프레임/초의 속도 (30 fps)로 빠르고 연속적인 방법으로 제공될 때 그들을 우리는 인간의 눈의 잔상 때문에 활동성 있는 이미지로서 지각한다.

연속적인 방법으로 표현될 이미지들을 분류하고 비디오 신호들을 형성하기 위해서, 각 이미지는 로우들로 나뉘질 필요가 있는데, 여기서 각 라인인 차례대로 화소 또는 픽셀로 나뉘지고, 각 픽셀은 2개의 관련된 값, 즉 루마(luma) 및 크로마(chroma)를 갖는다. 루마는 각 포인트에서 광 세기를 나타내고, 반면에 3 바이트에 의해 표현될 수 있는 규정된 컬러 공간(RGB)의 기능으로서 컬러를 나타낸다.

이미지들은 수평-수직 래스터, 상하 및 좌우로 그리고 주기적으로 스크린 상에 표시된다. 디스플레이의 라인 수와 주파수는 NTSC, PAL 또는 SECAM 같이, 포맷의 기능으로서 바뀔 수 있다.

이론적으로 각 루마, 크로마 U와 크로마 V 픽셀에 대해 값을 할당하는 것이 가능하나, 이것은 4 바이트 (하나의 크로마를 위한 것과 3개는 컬러를 위한 것)를 나타내고, NTSC 480 로우 x 720 컬럼 포맷 및 초당 대략 30 프레임으로, 4 x 480 x 720 x 30, 대략 초당 40 메가바이트 메모리가 생기는데, 이는 가용 대역폭 때문에 저장과 전송이 어렵다. 현재는, 1:4 픽셀로 크로마 데이터들을 줄이고, 즉 각 4개의 픽셀 당 컬러 샘플을 취하고, 분실된 3개에 대해 동일한 정보를 복제하는데, 인간은 그 차이를 인지하지 못하며; 그 포맷은 다음과 같다:

4: 4 : 4 (4 x 4 = 16 픽셀 그룹에서 4개의 루마 샘플과 4개의 크로마 샘플).

4: 2 : 2 (4 x 2 = 8 픽셀 그룹에서 4개의 루마 샘플과 2개의 크로마 샘플).

4: 1 : 1 (4 x 1 = 4 픽셀 그룹에서 4개의 루마 샘플과 1개의 크로마 샘플).

4: 2 : 0 MPEG1의 (4 x 2 = 8 픽셀 그룹에서 수평 픽셀들 간의 2개의 크로마 샘플과, 8개의 루마 샘플).

4: 2 : 0 MPEG1의 (4 x 2 = 8 픽셀 그룹에서 수직 픽셀들 간의 2개의 크로마 샘플과, 8개의 루마 샘플).

정보가 이러한 방식으로 감소될 때에도, 4: 2: 0 품질에서 NTSC 포맷으로 정보의 1/2을 저장하는데 필요한 필수 디지털 정보량은 15 메가바이트이거나, 2시간짜리 파일을 위해서는 108 기가바이트가 된다.

몇가지 방법들이 2차원 비디오 시퀀스로부터 3차원 화면 재구성성을 위해 존재한다. 최근의 기술 발전에 비추어서, 그리고 장래에 관해서, MPEG4 표준은 매체를 코딩하는 시간-공간 관련 그래픽스를 제공하려고 하는데, 이는 입체적인 이미지, 디자인과 공학 응용 제조에 중요한 도구가 된다. 화면의 기하학 모델이 재구성되는 경우에 가상의 공간이 만들어진다. 예를 들면, 세실 듀포(Cecile Dufour)에게 2003년 12월 9일에 특허된 USP No. 6,661,914호가 있는데, 여기서 새로운 3차원의 재구성 방법이 기술되는데, 화면 연속이 간단한 카메라로 일어나고, 이미지의 윤곽들이 재구성되며, 각 뷰(view)에서의 숨겨진 부분의 깊이가 나중에 투영되고 정제된 프로세스에 제출된다.

이미지 처리를 위해, 많은 가치있는 공헌들이 있는데, 예를 들면, 2003년 10월 21일에 이토가와(Itokawa)에게 특허된 USP No. 6,636,644호는 MPEG4를 사용하는 이미징 프로세스를 인용하고 있는데, 여기서는 이미지 경계에 걸쳐 확장되는 크로마 값 이미지가 추출되고, 더 큰 효율이 코딩시에 이뤄지며, 자연적 컬러 생성이 이미지의 윤곽에서 이뤄질 수 있다.

2003년 10월 14일에 클레이호스트(Kleihorst)에게 특허된 USP No. 6,633,676호 같은 몇몇 방법 및 장치들이 비디오 신호를 코딩하기 위해 존재하는데, 이 방법은 카메라 시스템에서 코더 검출기에 적용되고, 비디오 신호는 보상된 모션(I.B.P.)으로 코딩되며, 고해상도 이미지가 생성되고, 이 이미지는 이전의 것들의 보간(interpolation)이며, 요약하면 더 큰 해당 영역들이 비디오 신호에서 결정되는데, 이는 더 작은 메모리를 함께 점유하게 된다.

이미지 압축 코딩은 효율적인 방식으로 디지털 이미지들을 저장하거나 전송하기 위해 본질적으로 사용되는데, 압축된 디지털 이미지들을 코딩하는 방법은 JPEG와 MPEG 같은 공통의 표준들의 지배적인 기술에서와 같이 DCT를 사용한다. 2002년 2월 5일에 분(Boon)에게 특허된 USP No. 6,345,123호는 통상의 DCT 방법으로 계수들을 변형함으로써 디지털 이미지 코딩의 방법을 기술하고 있고, 상기 계수의 양자화 처리를 적용하여 미리 쓰여진 양자화 스케일에서 그들을 변형하고, 그리고 마침내 가변 길이 코딩 프로세스가 양자화되고 변형된 계수들에 적용되어, 이들을 가변 길이 코드 테이블에 비교하게 된다.

이미지는 코딩되기 위해 복수의 작은 영역들로 나뉘지고, 이 작은 영역들은 서로 인접하며, 샘플은 영역으로부터 취해지고 다음 이미지 영역의 환경이 예측된다. 이 예측적인 코딩 방법은 2000년 11월 4일에 분(Boon) 등에게 특허된 USP No. 6,148,109호에서 사용되는데, 작은 영역들 간의 차이의 생성된 이미지 데이터가 코딩되고 추출된다.

2000년 8월 1일에 무라카미(Murakami) 등에게 특허된 USP No. 6,097,759호는 필드 코딩 이미지를 위한 바이-블록(by-block) 코딩 시스템을 기술한다. 블록 패턴들은 1개의 개별 블록 필드와 1개의 논-인터레이스 블록을 포함하고; 또한 코딩 시스템은 보상된 이동 예측 신호를 생성하기 위해 홀수 및 짝수 필드 이동을 조사하고, 따라서 고효율 코딩이 제공된다.

카타다(Katata)에게 특허된 USP No. 5,978,515; 5,963,257; 5,815,601호는 기술하기 위해 사용된 데이터량을 늘리지도 않고, 다른 영역에 비해 선택된 영역의 이미지 품질을 개량하는 방식으로 이미지 데이터를 코딩하기 위한 이미지 코더를 인용하고 있다.

1996년 11월 26일에 기슬(Gisle)에게 특허된 USP No. 5,579,413호는 양자화된 이미지 블록에서 데이터 신호를 변환하고 이를 가변 길이 코딩 데이터 신호에서 변형하기 위한 방법을 기술하는데, 여기서 각 이벤트는 3차원의 양으로서 표현된다.

더 적은 공간에서 같은 내용의 저장을 허용하는 데이터 압축 시스템을 사용해야 하는 필요성이 대두되어, 전문가 그룹은 정보를 압축하고 이미지를 표시하는 방법을 만드는데 전념하였지만; 구현 세부들을 참조하지 않고, 모든 소프트웨어와 하드웨어 개발자들의 목적은 MPEG와 호환가능하게 주어진 프로세스를 수행하는 새로운 형태를 만드는 것이다. 현재, MPEG2는 세계적인 표준이고, 텔레비전과 비디오 및 오디오 관련 회사들에 의해 널리 사용된다.

오디오와 비디오는 요소의 패키지로 패키징되고 (PES), 상기 오디오와 비디오 패키지는 MPEG2 데이터 스트림을 만들기 위해 함께 혼용된다. 각 패키지에는 재생 시간에 오디오 및 비디오 동기화를 위한 시간 식별 (시간 스탬프)이 주어지는데, 예를 들면, 3개의 비디오 프레임마다 1개의 오디오 프레임이 관련된다.

MPEG는 시스템의 스트림들에서 비디오와 오디오를 혼용하는 2개의 상이한 방법들을 가진다:

위성 시스템들 같은, 전송 스트림은 더 큰 에러 가능성이 있는 시스템들에서 사용되는데, 이는 간섭에 민감하다. 각 패킷지는 길이가 188 바이트이고, 갭들을 인지하고, 가능한 에러들을 수리하는 식별 헤더에서 시작한다. 여러 가지 오디오와 비디오 프로그램들은 한 개의 전송 스트림에서 전송 스트림을 통해 전송될 수 있고; 헤더 때문에, 그들은 독립적으로 그리고 개별적으로 해독될 수 있고 많은 프로그램들로 통합될 수 있다.

프로그램 스트림은 DVD 플레이에서와 같이, 더욱 작은 에러 가능성을 갖는 시스템들에서 사용된다. 이 경우, 패키지들은 가변 길이와 실질적으로 전송 스트림에서 사용되는 패키지들보다 큰 사이즈를 갖는다. 주된 특징으로서, 프로그램 스트림은 한 개의 프로그램 내용만을 허용한다.

MPEG2 표준 하의 비디오 시스템은 혼용된 형태와 단계적 형태의 비디오 이미지들의 코딩을 허용한다.

즉, 단계적 비디오 체계는 전체 프레임(프레임 픽처, fp)내에 저장되고, 혼용된 비디오 체계 내에서, 이는 2가지 방식, 즉 전체 프레임 이미지(프레임 픽처)에 의해 또는 필드 이미지(필드 픽처)에 의해 저장될 수 있다.

압축 체계에는 3개의 MPEG2 포맷 이미지들이 존재한다:

인트라코딩된(intracoded) (I), 그들의 정보는 이미지 자신의 내부 데이터들의 함수로 코딩된다.

예측-코딩된 (P), 그 정보는 다른 미래의 시간 포인트에 있는 데이터에만 의존한다.

양방향 예측 코딩된 (B), 그 정보는 과거와 미래에 위치하는 데이터에 의존한다.

순번대로, 3개의 압축 형태들이 있는데, 이들은 상기 패키지들, 예를 들면 시간 예측, 압축과 공간 압축에 적용된다.

제시간의 예측 압축은 제시간의 2개의 다른 프레임을 인용하지만, 이는 모션이 관련되어 있고, 프레임들 간의 이미지가 거의 변하지 않는다는 장점을 갖는다.

공간의 압축은, 예를 들어, 루마를 위한 1 바이트 그리고 컬러를 위한 3 바이트를 갖는 100 x 100의 픽셀 이미지 내에, 1개의 동일한 프레임 내에 있는 (인트라코딩된) 정보를 짝 채우는데, 만일 이러한 정보를 저장하는 것이 요구된다면 프레임당 40 킬로바이트가 필요하고; 반면에 이러한 이미지가 완전히 화이트이면, 그것은 컬러:

255R, 255G, 255B, Xstart=0, Ystart=0, Xend=99, Yend=99로서 표현되고, 이것은 이 전체의 영역이 화이트임을 나타내고; 40 킬로바이트를 사용하는 대신에, 단지 7 또는 8이 사용된다. 따라서, MPEG 압축이 이뤄지고; 처리 스텝들은 복잡하게 되며, 현재의 발명의 범위를 벗어난다.

(I)형 이미지들은 완비되어 있고, 이들은 이전 또는 이후의 이미지 어느 것도 인용하지 않으므로, 압축은 시간 예측에서 사용되지 않고, 그 대신 자신의 공간 기능만으로서의 역할을 한다.

(P)형 이미지들은 그들 자신을 코딩하기 위해 참조 이미지들에 기초하므로, 그들은 시간 예측 압축, 그리고 공간 압축도 사용한다. 이들 이미지들은 (I)형 이미지 또는 다른 (P)형 이미지들을 인용할 수 있지만, 1개의 이미지 참조 이미지만을 사용한다.

(B)형 이미지들은 재구성되기 위해서 2개의 이전 및 이후 참조들을 요구하는데, 이러한 형태의 이미지들은 최고의 압축 색인을 가진다. (B)형 이미지를 얻기 위한 참조는 (P) 또는 (I)형만 있을 뿐, 결코 (B)형은 아니다.

코딩과 디코딩 시퀀스들은 다르다.

정보 볼륨을 감소시키기 위해서, 완전한 이미지는 소위 매크로블록(macroblock)으로 불리는 유닛의 전체 프레임으로 나뉘고; 각 매크로블록은 16 픽셀 x 16 픽셀의 분할에 의해 포함되며; 이는 상하 및 좌우로 순서가 정해지고 명명되어, 스크린 상에 매크로블록 매트릭스 어레이를 만들고, 매크로블록들은 순서가 정해진 연속적 방법, 즉 0, 1, 2, 3,...n으로 정보 스트림 내에 보내어진다.

(I)형 이미지를 가지고 있는 매크로블록들은 공간 압축에서만 완비되어 있고; (P)형 이미지들은, 제한없이 인트라코딩된 매크로블록 (인터레이스된 매크로블록)일 가능성이므로, 이전의 이미지들을 인용하기 위해서 (P)형 매크로블록들을 포함할 수 있다.

(B)형 이미지들은 또한 인트라코딩된 (인터레이스된) 형태의 매크로블록들에 의해 형성될 수 있는데, 이들은 이전 이미지, 이후 이미지 또는 둘다를 인용한다.

순번대로, 매크로블록들은 블록들로 나뉘지고, 1개의 블록은 8 x 8 데이터 또는 샘플 매트릭스이고; 크로마 포맷이 분류되는 형태 때문에, 4 : 4 : 4 포맷은 1개의 루마 샘플 Y, 1개의 크로마 샘플 Cr, 1개의 크로마 샘플 Cb를 필요로 하고, 그러므로 하나의 4 : 4 : 4 포맷 매크로블록은 매크로블록당 12개의 블록을 필요로 하며, 4 : 2 : 0 포맷에서는 매크로블록당 6개의 블록들이 필요하다.

연속적인 매크로블록들의 세트는 슬라이스(slice)를 표현하고; 슬라이스 내에는 소정 수의 매크로블록들이 있으며, 이들은 매크로블록과 동일한 방식으로 싱글 로우에 속해야 하며, 슬라이스들은 좌우 및 상하로 명명된다. 슬라이스들은 코딩된 이미지가 각 픽셀마다 샘플들을 필요로 하지는 않는 것처럼, 모든 이미지를 커버할 필요는 없다.

몇몇 MPEG 프로파일은 견고한 슬라이스 구조를 필요로 하는데, 여기서 이미지는 완전히 충족되어야 한다. 적절한 하드웨어와 소프트웨어 알고리즘 조합의 사용은 MPEG 이미지 압축을 허용한다.

코딩된 데이터들은 블록-특정 정보, 매크로블록, 필드, 프레임, 이미지 및 MPEG2 포맷 비디오를 가진 바이트들이다.

정보는 블록에 의해 그룹화되어야 하고, 정보 코딩, 예를 들면 (VLC)로부터 얻어진 결과들은 선형 비트-바이트 스트림이다.

VLC (가변 길이 디코더)가 압축 알고리즘인 경우에, 가장 빈번한 패턴들은 더 짧은 코드들로 대체되고, 그다지 빈번하게 일어나지 않는 것들은 더 긴 코드들로 대체된다. 이 정보의 압축된 버전은 더 작은 공간을 점유하고, 네트워크에 의해 더 빨리 전송될 수 있다. 그러나, 그것은 쉽게 편집 가능한 포맷이 아니고, 록-업 테이블을 사용하여 감압(decompression)을 필요로 한다.

역 스캔, 정보는 블록들에 의해 그룹화되어야 하고, VLC에 의해 정보를 코딩할 때 얻어진 것들은 선형 스트림이다. 블록들은 8 x 8 데이터 매트릭스들이고, 따라서 정방형의 8 x 8 매트릭스들에서 선형 정보를 변환하는 것이 필요하다. 이것은 단계적 이미지 또는 인터레이스된 이미지인 지에 따라, 내림 순서 지그재그 방식, 즉 둘다 시퀀스 형태로 상하 및 좌우로 만들어진다.

역 양자화는 단지 각 데이터 값에 인수를 곱하여 간단히 구성된다. 코딩될 때, 블록들의 대부분의 데이터는 인간의 눈이 지각할 수 없는 정보를 제거하기 위해 양자화되고, 양자화는 더 큰 MPEG2 스트림 변환을 얻기 위해 허용하며, 그것은 또한 디코딩 프로세스에서 역 프로세스(역 양자화)를 실행하는 것이 요구된다.

MPEG 비디오 시퀀스 구조 : 이것은 MPEG2 포맷에서 사용되는 최대 구조이고, 이하의 포맷을 가진다:

비디오 시퀀스 (Video_Sequence)

시퀀스 헤더 (Sequence_Header)

시퀀스 확장 (Sequence_Extension)

사용자 데이터(0) 및 확장 (Extension_and_User_Data (0))

이미지 그룹 헤더 (Group_of_Picture_Header)

사용자 데이터 (1) 및 확장 (Extension_and_User_Data (1))

이미지 헤더 (Picture_Header)

코딩된 이미지 확장 (Picture_Coding_Extension)

사용자 데이터 (2) 및 확장 (Extension_and_User_Data (2))

이미지 데이터 (Picture_Data)

슬라이스 (Slice)

매크로블록 (Macroblock)

모션 벡터 (Motion_Vectors)

코딩된 블록 패턴 (Coded_Block_Pattern)

블록 (Block)

최종 시퀀스 코드 (Sequence_end_Code)

비디오 시퀀스는 이들 구조들로 이루어지고, 비디오 시퀀스는 MPEG1와 MPEG2 포맷에 적용되는데, 각 버전을 차등화하기 위해서는 시퀀스 헤더 직후 시퀀스 확장이 존재하고, 시퀀스 확장이 헤더를 따르지 않으면 스트림이 MPEG1 포맷 비디오 스트림임이 검증되어야 한다.

발명의 상세한 설명

본 발명의 목적은 입체적인 3D-이미지 디지털 코딩 방법 및 시스템을 제공하는 것인데, 이 시스템은 3Dvisors®에서 전송, 수신 및 디스플레이를 위해 코딩된 데이터들을 제공한다.

본 발명의 다른 목적은 비디오 데이터 스트림 비디오 시퀀스(video_sequence) 구조들이 변형되고 식별 플래그들이 비트 레벨에서 포함되는 코딩 체계를 제공하는 것이다.

본 발명의 또 다른 목적은 비디오 시퀀스(video_sequence), 식별 플래그, 데이터 필드 및 이미지 필드가 변형되는 방식으로 3D-이미지 디지털 코딩 소프트웨어 프로세스를 제공하는 것이다.

본 발명의 또 다른 목적은 좌우 채널들 간의 전자 비교가 이루어지고, 이미지들 간의 차이의 에러 교정이 이루어지며, 처리된 이미지가 TDVision® 기술 식별자를 갖는 비디오 시퀀스(video_sequence) 내에 저장되는 방식으로 3D-이미지 디지털 코딩 하드웨어 프로세스를 제공하는 것이다.

본 발명의 또 다른 목적은 DSP의 입력 버퍼 메모리가 2배로 되고, 2개의 독립적인 비디오 신호들의 동시 입력이 가능하며, DSP가 2개의 비디오 신호들의 입력 버퍼들을 비교할 수 있게 되는 방식으로 3D-이미지 디지털 코딩 하드웨어 프로세스를 제공하는 것이다.

실시예

디지털 비디오 스트림으로부터 3차원 이미지를 얻는 목적에 따르면, 코딩 프로세스의 서로 다른 부분들에서 하드웨어와 소프트웨어를 변경함으로써 변형이 현재의 MPEG2 코더들에 행해진다. 도 1에 도시된 바와 같이, MPEG2-4 호환가능 TDVision® 코더(1)는 소프트웨어(3) 및 하드웨어(4) 변경을 통하여 달성된 자신의 코딩 프로세스(2)를 가진다.

도 2에서, 본 발명의 코더 오브젝트의 컴파일 프로세스가 도시되고, 실제로는, 이미지(10)가 취해져 모션 보상과 에러 검출 프로세스(11)에 제출되고; 이산 역현 변환 기능(discrete cosine transform function)은 주파수 매개 변수(12)를 바꾸기 위해 적용되고, 그 다음 양자화 매트릭스(13)는 표준화 프로세스를 수행하기 위해 적용되며, 로우 변환 프로세스(14)에 대한 매트릭스가 적용되는데, 여기에서는 가변 길이 코딩(15)을 수행하기 위한 가능성이 존재하고, 그리고 마침내, 코딩된 데이터(16)를 가진 비디오 시퀀스가 얻어진다. 이러한 컴파일 프로세스를 수행하기 위해, 포맷(30, 도 3) 또

는, MPEG2 호환가능 3D-이미지 컴파일 방법이 수반되어야 하며, 실제로는, 도 3에서 도시된 바와 같이, 비디오 시퀀스(video_sequence(31))는 시퀀스 헤더(sequence_header(32)), 사용자 데이터(user_data(33)), 시퀀스 스케일러블 확장(sequence_scalable_extension(34)), 픽처 헤더(picture_header(35)), 픽처 코딩 확장(picture_coding_extension(36)) 및 픽처 임시 스케일러블 확장(picture_temporal_scalable_extension(37)) 구조 내에서 변형되어야 하고, 따라서 TDVision® 입체 카메라로 취해진 입체적 3D-디지털 이미지에 적합한 컴파일 포맷을 얻는다.

비디오 데이터 스트림의 구조 및 비디오 시퀀스(video_sequence)는 비트 레벨에서 TDVision® 기술 코딩 이미지 형태를 식별하기 위해 필요한 플래그들을 포함하도록 변형되어야 한다.

변형은 다음 코딩 스테이지, 즉 MPEG2 (소프트웨어)에서 이중 이미지를 코딩할 때; 하드웨어에 의해 이미지를 코딩할 때, 이루어진다.

소프트웨어:

비디오 시퀀스 헤더(video_sequence headers)들이 변형된다.

식별 플래그들이 식별된다.

데이터 필드들이 변형된다.

이미지 필드들이 변형된다.

하드웨어:

전자 비교는 좌우의 채널들 사이에서 이루어진다.

차이는 B형 이미지 (에러 정정)로서 처리된다.

그 다음, TDVision® 식별자와 함께 저장된다.

변경이 상보적 버퍼에 적용된다.

결과들은 절약되고, 2차 버퍼에 저장된다.

실제로, DSP 버퍼의 입력 메모리는 2배로 되고; 입체적 TDVision® 카메라로부터의 입체적인 좌-우 기존 신호에 대응하는 2개의 독립적 비디오 신호들의 동시 입력이 허용되며; DSP는 양쪽 비디오 신호들을 위한 입력 버퍼들을 비교하는 것이 가능하게 된다.

하드웨어 코딩 프로세스는 단일 비디오 입력 채널의 함수로서 보통의 MPEG2 방식으로 수행되고, 양쪽 신호 (좌우)들이 취해지고 전자적으로 비교되며, 좌우의 신호들 간의 비교의 차이가 얻어지고, 상기 차이는 임시 버퍼에 저장되며, 에러 정정은 왼쪽 신호에 관해서 루마(LUMA)와 크로마(CHROMA)에서 계산되고; DCT (이산 역현 변환) 함수가 적용되며, 정보는 B형 블록에 저장된다:

a) 사용자 데이터 식별 구조(USER_DATA()(SW) identification structure) 내부

b) 픽처 데이터 3D 구조(PICTURE_DATA3D() structure) 내부

다음 프레임에서 계속된다.

하드웨어는 도 4에서 블록도로 도시되고, 실제로는, 왼쪽 신호(41) 및 오른쪽 신호(42)가 취해지고, 둘 다의 신호들은 임시 버퍼(43)에 저장되며, 좌우 신호들 간의 차이가 비교되고, 차이의 에러가 계산되고, 그리고 정보는 저장되며(45), 정확한 이미지는 코딩되고(46), 코딩은 "I", "B" 또는 "P"형 이미지로서 수행되고(47), 마침내 비디오 시퀀스(video_sequence) (48) 내에 저장된다.

DSP에 의해 취급될 메모리를 이중으로 하고, 최대 8개의 출력 버퍼들을 배치할 수 있는 가능성을 가지는 것이 필수적인 데, 그것은 TDVision®의 3Dvisors® 같은 장치에 입체적인 이미지의 이전 및 동시 표현을 허용한다.

실제로, 텍사스 인스트루먼트 TMS320C62X DSP의 프로그래밍 API를 호출할 때 2개의 채널들이 초기화되어야 한다.

MPEG2VDEC_create (const IMPEG2VDEC_fxns*fxns, const MPEG2VDEC_Params* params).

여기서, IMPEG2VDEC_fxns y MPEG2VDEC_Params은 예를 들면 다음과 같은 각 비디오 채널을 위해 동작 매개 변수들을 정의하고 있는 포인터 구조체이다:

3DLhandle = MPEG2VDEC_create(fxns3DLEFT,Params3DLEFT).

3DRhandle = MPEG2VDEC_create(fxns3DRIGHT,Params3DRIGHT).

이에 의해, 2개의 비디오 채널들이 디코딩되게 하고, 2개의 비디오 핸들러를 얻게 되는데, 하나는 좌우 입체적 채널을 위한 것이다.

2중 디스플레이 출력 버퍼들이 필요하고, 그리고 소프트웨어에 의해, 2개의 버퍼들 중 어느 쪽이 AP 기능을 호출하는 것에 의해 출력을 표시해야 하느냐가 정의될 것이다:

즉, MPEG2VDEC_APPLY(3DRhandle, inputR1, inputR2, inputR3, 3doutright_pb, 3doutright_fb).

MPEG2VDEC_APPLY(3DLhandle, inputL1, inputL2, inputL3, 3doutleft_pb, 3doutleft_fb).

여기서, 3DLhandle은 DSP의 생성 기능에 의해 복귀된 핸들에 대한 포인터이고, input1 매개 변수는 FUNC_DECODE_FRAME 또는 FUNC_START_PARA 주소이며, input2는 외부 입력 버퍼 주소에 대한 포인터이고, 그리고 input3는 외부 입력 버퍼 크기의 치수다.

3doutleft_pb는 매개 변수 버퍼의 주소이고, 3doutleft_fb는 디코딩된 이미지가 저장될 곳인 출력 버퍼의 처음이다.

타임코드(timecode)와 타임스탬프(timestamp)는 연속적이고, 동기화된 방식으로 최종적인 장치에 출력하기 위해 사용된다.

소프트웨어와 하드웨어 프로세스들의 통합은 DSP로서 알려진 장치들에 의해 수행되는데, 이는 대부분의 하드웨어 프로세스들을 실행한다. 이들 DSP는 제조업자들에 의해 제공된 C 및 복합형의 어셈블리 언어에 의해 프로그램된다. 각 DSP는 자신의 API를 가지는데, 이는 DSP에 위치하고 소프트웨어에 의해 호출된 기능 리스트 또는 절차 호출들로 이루어져 있다.

이러한 참조 정보로 인해, MPEG2 포맷-호환가능 3D-이미지 코딩을 위한 본 출원이 만들어졌다.

실제로, 비디오 시퀀스의 처음에, 시퀀스 헤더와 시퀀스 확장이 항상 나타난다. 시퀀스 확장의 반복은 첫 번째와 동일해야 한다. 이에 반하여, 시퀀스 헤더 반복들은 첫 번째 발생에 비해 약간만 변하고, 양자화 매트릭스를 정의하는 부분만이 변경되어야 한다. 시퀀스 반복은 비디오 스트림에 랜덤 액세스를 허용하고, 즉, 디코더가 비디오 스트림의 중간에서 재생을 시작하면, 이는 작업을 가능케 하고, 그것은 비디오 스트림에서 다음의 이미지들을 디코딩할 수 있게 이전의 시퀀스 헤더와 시퀀스 확장을 검색할 뿐이어야 한다. 이는 또한, 프로그램이 이미 시작되었을 때에 작동된 위성 디코더 같이, 처음부터 시작될 수 없는 비디오 스트림들에 대해 일어난다.

즉, 시퀀스 헤더가 비디오 스트림 위에 더 높은 정보 레벨을 제공하고, 명확성을 위해 각각에 대응하고 있는 비트들의 수가 또한 지시되고, 최상위 비트들은 시퀀스 확장 (Sequence_Extension) 구조들 내에 위치하며, 이는 이하의 구조들에 의해 형성된다:

Sequence_Header

필드	# 비트	설명
Sequence_Header_Code	32	시퀀스 헤더 개시(Sequence_Header Start) 0x00001B3
Horizontal_Size_Value	12	폭에 대해서 12개의 하위 비트
Vertical Size Value	12	높이에 대해서 12개의 하위 비트
Aspect Ratio Information	4	이미지 종횡비 0000 금지 0001 n/a TDVision® 0010 4:3 TDVision® 0011 16:9 TDVision® 0100 2.21:1 TDVision® 0111은 2D 시스템과 역방향 호환성을 얻기 위해 논리 "and" 명령을 실행한다. 0101...1111 예약

Frame rate code	4	0000 금지 TDVision® 포맷 내에서 0001 24,000/1,001 (23.976) TDVision® 포맷 내에서 0010 24 TDVision® 포맷 내에서 0011 25 TDVision® 포맷 내에서 0100 30,000/1001 (29.97) TDVision® 포맷 내에서 0101 30 TDVision® 포맷 내에서 0110 50 TDVision® 포맷 내에서 0111 60,000/1001 (59.94) (논리 "and"는 2D 시스템과 역방향 호환성을 얻기 위해 실행된다) 1000 60 1111 예약
Bit_rate_value	18	비디오 스트림 비트 비율 (video_stream bit rate)의 18개의 하위 비트 (bit_rate = 400 x bit_rate_value + bit_rate_extension << 18) 최상위 비트는 시퀀스 확장 구조 (sequence_extension structure) 내에 위치한다.

Marker_bit	1	항상 1 (개시 코드(start_code) 실패를 방지한다).
Vbv_buffer_size_value	10	비디오 버퍼 검증기(VBV)의 사이즈를 결정하는 vbv_buffer_size의 10개의 하위 비트, 데이터 스트림이 버퍼에서 너무 초과 하거나 너무 많은 자유 공간을 남기는 것 없이 제한된 사이즈 버퍼를 디코딩 하는데 사용될 수 있는 것을 보장하기 위해 사용되는 구조.
Constrained_parameters_flag	1	항상 0, MPEG2에 사용되지 않음.
Load_intra_quantizer_matrix	1	intra-coded quantization matrix가 사용가능한 지를 나타낸다.
If load_intra_quantizer_matrix		
Non_intra_quantizer_matrix(64)		8x64 양자화 매트릭스가 나타나면, 이는 여기에 특정되어야 하고, 8x64 매트릭스이다.
Load_non_intra_quantizer_matrix	1	만일 non-intra-quantized matrix가 이용가능하면, 이 플래그는 활성화되어야 한다.
If load_non_intra_quantizer_matrix		
Non_intra_quantizer_matrix (64)	8x64	만일 이전의 플래그가 활성화되 면, 양자화된 매트릭스를 형성 하는 8x64 데이터들은 여기에서 저장된다.

Sequence_extension

필드	#비트	설명
Extension_Start_Code	32	개시, sequence_extension, 항상 0x000001B5
Extension_Start_code_Identifier	4	확장형 0x1에 의해 식별된다.
Profile_and_level_indication	8	프로필과 비디오 스트림 레벨을 정의한다
Progressive_sequence	1	1= 프레임, 0 = 프레임과 필드
Chroma_format	2	00 예약 01 4:2:0 10 4:2:2 11 4:4:4
Horizontal_size_extension	2	sequence_header 확장
Vertical_size_extension	2	sequence_header 확장
Bit_rate_extension	12	sequence_header 확장
Marker_bit	1	항상 1
Vbv_buffer_size_extension	8	sequence_header 확장
Low_delay	1	1 = B형 이미지를 갖지 않고, 정상 재생 (BIG 이미지로 알려 진) 동안 VBV 버퍼의 under- utilization를 일으킬 수 있다. 0 = B형 이미지들을 포함하나; BIG 이미지를 가질 수 없고, VBV 버퍼의 under-utilization 를 일으킬 수 없다.
Frame_rate_extension_n	2	
Frame_rate_extension_d	5	
Next_start_code()		

확장 및 사용자 데이터 (i)

이것은 다른 구조들을 저장하기 위한 컨테이너이고, 자신의 데이터들을 가지지 않으며, 본질적으로 일련의 extension_data(1) 및 user_data () 구조이고, 몇몇 경우에는 이 구조들은 완전하게 비어 있을 수 있다.

Extension_data (i)

이러한 구조는 단순한 구조 확장을 포함한다. 내포된 확장 구조 형태는 (i)의 값에 의존하는데, 이는 1 또는 2의 값을 가질 수 있다. 만일 이것이 "0"과 같으면, data_extension은 sequence_extension을 따르게 되고, extension_data(i)는 1개의 sequence_display_extension 또는 1개의 sequence_scalable_extension 양쪽을 포함할 수 있다: .

만일, i = 2이면, 구조는 picture_coding_extension을 따르는데, 이는 quant_matrix_extension(), copyright_extension(), picture_display_extension(), picture_spatial_scalable_extension() 또는 1개의 picture_temporal_scalable_extension를 포함할 수 있다. 이러한 구조는 항상 0x000001B5에서 시작한다.

User_data

user_data 구조는 비디오 시퀀스 (video_sequence) 내에 저장될 어플리케이션에 특정한 데이터들을 허용한다. MPEG2 규정은 이러한 함수의 포맷 또는 사용자 데이터의 포맷을 정의하지 않는다. 구조는 user_data_start_code = 0x000001B5에서 시작하고, 데이터 스트림 (스트림) 내에서 다음 개시 코드까지 이어지는 데이터 (user_data)의 임의의 수를 포함한다. 유일한 조건은 그것이 개시 코드로서 해석될 수 있었던 것처럼, 23개 이상의 연속 제로들을 갖지 않는다는 것이다.

Sequence_display_extension()

이 구조는 디코딩 프로세스에 사용되지 않는 정보, 코딩된 내용을 인용하는 정보를 제공하는데, 이는 디코딩된 비디오를 빠르게 디스플레이하는 데 유용하다.

Sequence_display_extension()

필드	비트#	설명
Extension_start_code_identifier	4	2이어야 하고, 개시를 의미한다.
Video_format	3	000 구성 요소 001 PAL 010 NTSC 011 SECAM 100 MAC 101 미지정 110 예약, TDVision® 111 예약, TDVision®
Color_description	1	0 = 컬러 매개 변수들을 지정하지 않는다. 1 = 다음의 3개의 컬러 매개 변수들을 포함한다.
Color_rprimaries	8	0 금지 1ITU-R-BT.709 권고안 2 미지정 비디오 3 예약 4ITU-R-BT.470-2 SYSTEM M 권고안

		5ITU-R-BT.470-2
		권고안
		SYSTEM B,G
		6 SMPTE 170M
		7 SMPTE 240M
		8 - 255 예약
Transfer_characteristics	8	0 금지
		1 ITU-RBT.709
		권고안
		2 미지정 비디오
		3 예약
		4 ITU-R-BT.470-2
		SYSTEM M
		권고안
		5 ITU-R-BT.470-2
		SYSTEM B,G
		권고안
		6 SMPTE 170M
		7 SMPTE 240M
		8 실제 전송 특성
		255 예약

Matrix_coefficients	8	0 금지 1 ITU-RBT8709 권고안 2 미지정 비디오 3 예약 4 FCC 5 ITU-R-BT.470-2 SYSTEM B,G 권고안 6 SMPTE 170M 7 SMPTE 240M 8 - 255 예약
Display_horizontal_style	14	MPEG2에서 지정되지 않는다.
Marker_bit	1	항상 1
Display_vertical_size	14	MPEG2에서 지정되지 않는다.
Next_start_code		

Sequence_scalable_extension

이 구조는 모든 스케일러블 비디오 스트림에 존재해야 하는데, 이는 기본계층과 하나 이상의 보완계층들을 포함하고 있다. 다른 형태의 MPEG2 스케일러빌리티(scalability)가 존재하되, 주된 층을 위한 스케일러빌리티의 예는 확장 층이 추가 데이터를 갖는 동안 그것이 비디오 내용의 표준의 정의를 포함한다는 것인데, 이는 한계를 증가시킨다.

Sequence_scalable_extension

설명	비트#	설명
Extension_start_code_identifier	4	항상 5
Scalable_mode	2	00 분할된 데이터 01 공간 스케일러빌리티 10 SNR 스케일러블 11 임시 스케일러빌리티
Layer_id	4	층 번호 (0)
Lower_layer_prediction_horizontal_size	14	
Marker_bit	1	
Lower_layer_prediction_vertical_size	14	
Horizontal_subsampling_factor_m	5	
Horizontal_subsampling_factor_n	5	
Vertical_subsampling_factor_m	5	
Vertical_subsampling_factor_n	5	
Picture_mux_enable	1	
Mux_to_progressive_sequence	1	
Picture_mux_order	3	
Picture_mux_factor	3	
Group_of_pictures_header()		

Group_of_pictures_header()

이 구조는 이미지 그룹의 개시를 마크한다.

Group_of_pictures_header()

필드	비트#	설명
Group_start_code	32	0x000001B8
Time_code	25	group_of_picture_header 이전의 제1 이미지의 타임스탬프 Dromp_frame_flag-1 Time_code_hours-5 Time_code_minutes-6 Marker_bit-1 Time_code_seconds-6 Time_code_pictures-6
Closed_gop	1	만일 1= B 이미지이면, 이전의 이미지를 인용하지 않는다.
Broken_link	1	1 = 더 이상 존재하지 않는 분실 I형 프레임을 가리킨다. 0 = 링크는 부서지지 않는다.

Next_start_code()

Picture_header

필드	비트#	설명
Picture_start_code	32	0x00000100
Temporal_reference	10	이미지 디스플레이 순서
Picture_coding_type	3	0000 금지 001 intra-coded (I) 010 predictive-coded (P) 011(B) bidirectional_predictive_coded 100 MPEG1를 위해 예약 101 예약 110 예약 111 예약

Vbv_delay	16	비디오 버퍼 검증 기계 장치 (임시 메모리)
Full_pel_forward_vector	1	MPEG1에서 사용된다. MPEG2 = 0
Forward_f_code	3	MPEG1에서 사용된다. MPEG2 = 111
Full_pel_backward_vector	1	MPEG1에서 사용된다. MPEG2 = 0
Backward_f_code	3	MPEG1에서 사용된다. MPEG2 = 111
Extra_bit_picture	1	무시될 수 있다.
Extra_information_picture	8	무시될 수 있다.
Extra_bit_picture	1	무시될 수 있다.
Extra_start_code		

Picture_coding_extension

필드	비트#	설명
Extension_start_code	32	항상 0x000001B5
Extension_start_code_identifier	4	항상 1000
F_code(0)(0)	4	모션 벡터를 디코딩하는데 사용되고; 이것이 I형 이미지일 때, 이러한 테이터는 1111로 가득 차 있다.
F_code(0)(1)	4	
F_code(1)(0)	4	모션 벡터(B)에서 뒤로 정보를 디코딩하고, 그것이 (P)형 이미지일 때, 이는 뒤쪽의 이동이 없기 때문에 그것은 1111에 설정되어야 한다.
F_code(1)(1)	4	모션 벡터에서 뒤로 정보를 디코딩하고, 그것이 P형 이미지일 때, 이는 뒤쪽의 이동이 없기 때문에 그것은 1111에 설정되어야 한다.
Intra_dc_precision	2	DC 이산 여현 변환의 역양자화

		시에 사용된 정밀도.
		00 8개의 비트 정밀도
		01 9개의 비트 정밀도
		10 10개의 비트 정밀도
		11 11개의 비트 정밀도
Picture_structure	2	이미지가 필드에서 또는 전체 프레임에서 나뉘지는 지를 지정한다.
		00 예약 (TDVision®포맷 내의 이미지)
		01 상부 필드
		10 하부 필드
		11 바이-프레임 이미지(by-frame image)
Top_field_first	1	0 = 첫번째로 하부 필드를 디코딩한다.
		1 = 첫번째로 상부 필드를 디코딩한다.
Frame_pred_frame_dct	1	
Concealment_motion_vectors	1	

Q_scale_type	1	
Intra_vic_format	1	
Alternate_scan	1	
Repeat_first_field	1	0 = 단계적 프레임을 표시한다. 1 = 2개의 일치하는 단계적 프레임을 표시한다.
Chroma_420_type	1	만일 크로마 포맷이 4 : 2 : 0이라면, 그것은 progressive_frame와 같고, 그렇지 않다면 그것은 제로와 같은 것임에 틀림없다.
Progressive_frame	1	0 = 인터레이스 1 = 단계적
Composite_display_flag	1	본래 코딩된 정보에 관해 경고한다.
V_axis	1	
Field_sequence	3	
Sub_carrier	1	
Burst_amplitude	7	
Sub_carrier_phase	8	
Next_start_code()		

Picture_temporal_scalable_extension()

2개의 공간 해상도 스트림은 임시 스케일러빌리티를 갖는 경우에 존재하고, 하부층은 비디오 프레임들의 더 작은 인덱스 버전을 제공하며, 반면에 상부층은 동일한 비디오의 프레임들의 더 큰 인덱스 버전을 끌어내기 위해 사용될 수 있다. 임시 스케일러빌리티는 낮은 품질, 저비용 또는 자유 디코더들에 의해 사용될 수 있는 반면, 초당 더 큰 프레임들은 비용을 위해 사용된다.

Picture_temporal_scalable_extension()

필드	비트#	정의
Extension_start_code_identifier	4	항상 1010
Reference_select_code	2	참조 이미지가 intra_coded 이미지를 디 코딩하는데 사용되는 것 을 나타내기 위해 사용된다. 0형 이미지를 위해 00은 가장 최근 이미지를 강화한다. 01은 디스플레이 순서 내 의 하부 및 가장 최근 프레임층 10은 디스플레이 순서 내 의 프레임 아래의 다음층 11 금지 B형 이미지를 위해 00 금지 01은 강화된 모드에서 가 장 최근 디코딩된 이미지 10은 강화된 모드에서 가 장 최근 디코딩된 이미지 11은 디스플레이 순서 내 의 하부층에서 가장 최근 이미지
Forward_temporal_reference	10	임시 참조
Marker_bit	1	
Backward_temporal_reference	10	임시 참조
Next_star_code()		

Picture_spatial_scalable_extension()

이미지 공간 스케일러빌리티의 경우, 보완계층은 데이터를 포함하는데, 이는 기본계층의 양호한 해상도를 허용하여, 재구성될 수 있게 한다. 보완계층이 모션 보상을 위한 참조로서 기본계층의 함수로서 사용될 때, 하부층은 보완계층의 더 큰 해상도를 얻기 위해 확대되고 오프셋되어야 한다.

Picture_spatial_scalable_extension()

필드	비트#	정의
Extension_start_code_identifier	4	항상 1001
Lower_layer_temporal_reference	10	하부층의 임시 이미지의 참조
Marker_bit	1	1
Lower_layer_horizontal_offset	15	수평 보상 (오프셋)
Marker_bit	1	1
Lower_layer_veretical_offset	15	수직 보상 (오프셋)
Spatial_temporal_weight_code_table_index	2	예측 세부
Lower_layer_progressive_frame	1	1 = 단계적 0 = 인터레이스
Lower_layer_desinterlaced_field_select	1	0 = 상부 필드 사용 1 = 하부 필드 사용

Next_start_code()

Copyright_extension()

Extension_start_code_identifier	4	항상 010
Copyright_flag	1	만일 그것이 1과 같다면 저작권 을 사용한다. 만일 그것이 제로 (0)라면, 추 가의 저작권 정보는 필요하다.

Copyright_identifier	8	1 = 원본 0 = 사본
Original_or_copy	1	
Reserved	7	
Marker_bit	1	
Copyright_number_1	20	저작권 예에 의해 허여된 번호
Marker_bit	1	
Copyright_number_2	22	저작권 예에 의해 허여된 번호
Marker_bit	1	
Copyright_number_3	22	저작권 예에 의해 허여된 번호

Next_start_code()

Picture_data()

이는 단순한 구조이고, 자체 내에 필드를 갖지 않는다.

Slice()

동일한 수직 위치의 하나 이상의 매크로블록 상의 정보를 포함.

Slice_start_code	32
Slice_vertical_position_extension	3
Priority_breakpoint	7
Quantizer_scale_code	5
Intra_slice_flag	1
Intra_slice	1
Reserved_bits	7
Extra_bit_slice	1
Extra_information_slice	8
Extra_bit_slice	1
Macroblock()	
Macroblock_modes()	
Motion_vectors()	
Motion_vector()	
Coded_block_pattern()	
Block()	
EXTENSION_AND_USER_DATA(2)	

이러한 MPEG2 호환가능 코딩 프로세스는 현재 도 5의 입체적인 카메라 (52)로 취해진 3D-디지털 이미지들을 코딩하기 위해 사용되는데, 이는 컴파일러(51)을 통해 패스되고, 그 다음에 신호가 PC(50), 즉 DVD(53)에서 그것을 표시하기 위해 얻어질 수 있고; 디코더(54)의 신호가 코딩될 때, 이미지가 케이블(56), 위성(57), 고해상도 텔레비전(59) (HDTV)을 통해, 또는 3Dvisors® 디바이스(59) 등 내에서, 표시될 이미지를 위해 디코더(55) 내에 보내질 수 있다. 따라서, 이미지는 다음의 상부에 표시될 수 있다:

DVD (Digital Versatile Disks)

DTV (Digital Television)

HDTV (High Definition Television)

CABLE (DVB Digital Video Broadcast)

SATELLITE (DSS Digital Satellite Systems); 그리고 그것은 소프트웨어와 하드웨어 프로세스 통합이다.

하드웨어에 관해서, 대부분의 프로세스는 DSP (디지털 신호 프로세서)로서 알려진 장치들로 실행된다. 즉, 1개의 모토로라 모델과 1개의 텍사스 인스트루먼트 (TMS320C62X) 모델이다.

이들 DSP는 해당 제조업자에 의해 제공된 C 언어 및 어셈블리 언어들로부터의 복합형의 언어들에 의해 프로그램된다. 각 DSP는 자신의 API를 가지는데, 이는 소프트웨어에 의해 호출될 DSP에 위치하는 함수 리스트 또는 절차 호출들로 구성된

다. 이러한 참조 정보로부터, 3D-이미지들은 코딩되는데, 이는 MPEG2 포맷과 그리고 그들 자신의 코딩 알고리즘과 호환성을 가진다. 정보가 코딩될 때, DSP는 MPEG2 압축 비디오 스트림을 형성하기 위해 예측, 비교, 양자화 및 DCT 합수 응용 처리할 수 있게 된다.

발명의 특별한 구체화들이 도시되고 기술하게 되었지만, 본 발명의 범위를 벗어나지 않는 한도에서 본 분야의 숙련자들에게는 다양하게 변형 또는 변화될 수 있는 것들은 명백할 것이다. 모든 그러한 변형들과 변화들은 본 발명의 범위내에서 다음의 청구범위에 의해 커버된다.

도면의 간단한 설명

도 1은 입체적 3D-비디오 이미지 코딩을 위한 하드웨어와 소프트웨어 변경들을 도시한 도.

도 2는 MPEG2-4 호환가능 입체적 3D-비디오 이미지를 위한 컴파일 프로세스를 도시한 도.

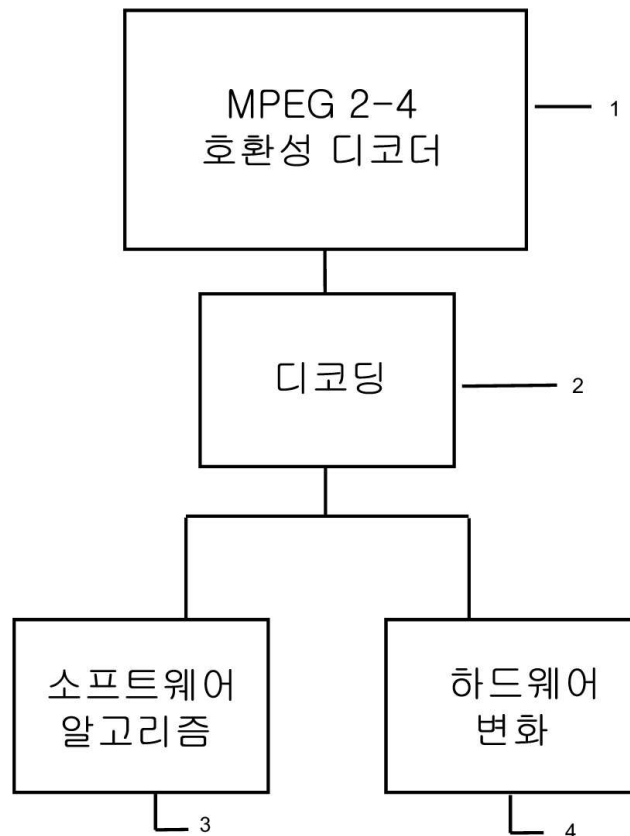
도 3은 MPEG2-4 호환가능 입체적 3D-비디오 이미지의 컴파일을 위한 소프트웨어 포맷을 도시한 도.

도 4는 MPEG2-4 호환가능 입체적 3D-비디오 이미지의 컴파일을 위한 하드웨어 포맷을 도시한 도.

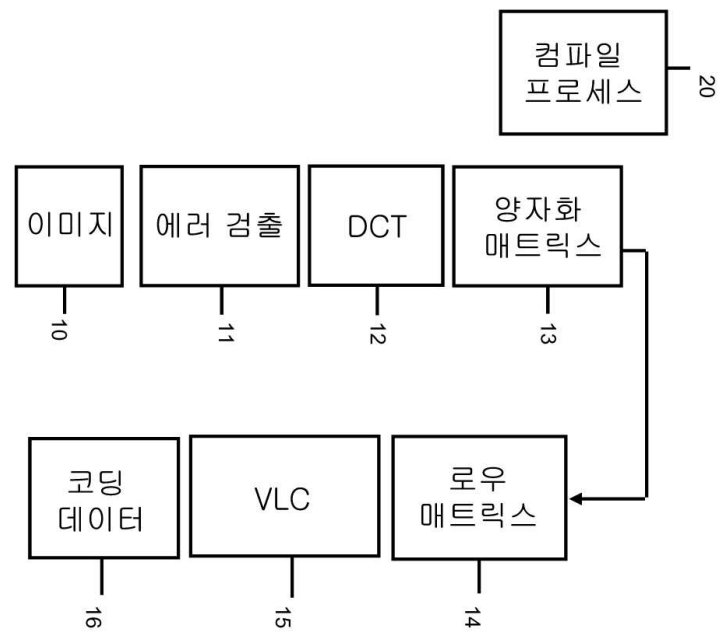
도 5는 본 발명의 코더가 속하는 기술 가지의 맵을 도시하는데; 즉, 입체적 3D-이미지 처리, 코딩, 디코딩, 케이블 및 위성을 통한 전송, 그리고 DVD, HDTV 및 3DVisors® 디스플레이를 도시한 도.

도면

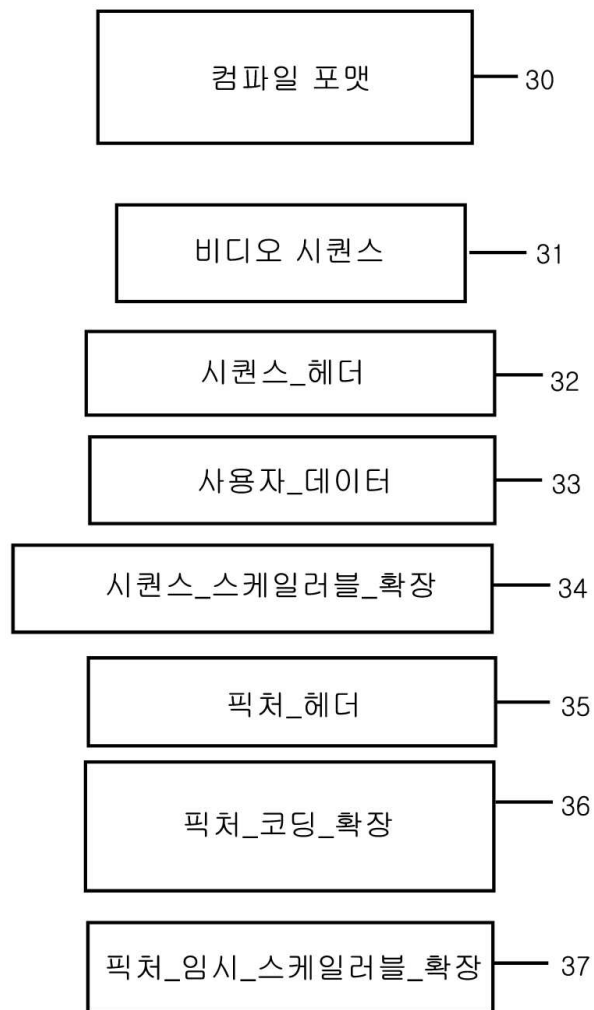
도면1



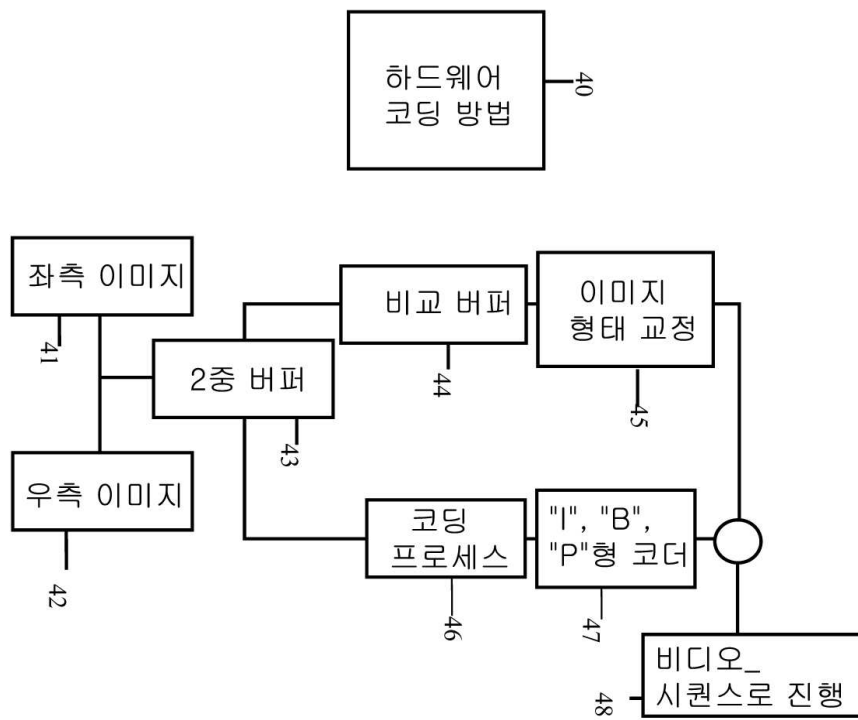
도면2



도면3



도면4



도면5

