(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.
H04N 7/26 (2006.01)

(11) 공개번호 10-2006-0069804
(43) 공개일자 2006년06월22일

(21) 출원번호 10-2006-0043693(분할)
(22) 출원일자 2006년05월16일
(62) 원출원 특허10-1999-0007916
원출원일자 : 1999년03월10일 심사청구일자 2004년03월10일

(30) 우선권주장 JP-P-1998-00058118 1998년03월10일 일본(JP)
JP-P-1998-00157243 1998년06월05일 일본(JP)

(71) 출원인 소니 가부시끼 가이샤
일본국 도쿄도 시나가와쿠 키타시나가와 6쵸메 7반 35고

(72) 발명자 다하라 가츠미
일본 도쿄도 시나가와쿠 기다시나가와 6-7-35 소니(주)내
무라까미 요시히로
일본 도쿄도 시나가와쿠 기다시나가와 6-7-35 소니(주)내
기따무라 다꾸야
일본 도쿄도 시나가와쿠 기다시나가와 6-7-35 소니(주)내
미하라 간지
일본 도쿄도 시나가와쿠 기다시나가와 6-7-35 소니(주)내

(74) 대리인 이병호

심사청구 : 있음

(54) 부호화 히스토리 정보를 이용하는 트랜스코딩 시스템

요약

본 발명은 MPEG 표준을 근거로 부호화 처리의 결과로서 얻어진 부호화된 비트 스트림의 GOP 구조 및 비트 레이트를 변환하는 트랜스코더를 제공한다. 본 발명에 의해 제공된 트랜스코더에 따라 과거 부호화 처리시 생성된 부호화 파라미터들은 히스토리 정보로서 현재 부호화 처리를 수행하는 MPEG 인코더에 전송된다. 상기 현재 부호화 처리에 상응하는 최적 부호화 처리는 전송된 부호화 파라미터로부터 선택되고, 상기 선택된 부호화 파라미터는 현재 부호화 처리시 재사용된다. 결국, 상기 화상의 품질은 부호화 및 복호화 처리가 반복적으로 수행될 때 조차도 일그러지지 않는다.

대표도

도 14

색인어

부호화, 복호화, 트랜스코더

명세서

도면의 간단한 설명

도 1은 고효율성 부호화 처리의 원리를 설명하는데 이용되는 예시적 도면.

도 2는 화상 데이터의 압축에 이용되는 화상 타입을 도시하는 예시적 도면.

도 3은 화상 데이터의 압축에 이용되는 화상 타입을 도시하는 예시적 도면.

도 4는 동화상 비디오 신호를 부호화하는 처리의 원리를 설명하는데 이용되는 예시적 도면.

도 5는 동화상 비디오 신호를 부호화 및 복호화하는데 이용되는 장치의 구성을 도시하는 블록도.

도 6a 내지 도 6c는 포맷 변환을 설명하는데 이용되는 예시적 도면.

도 7은 도 5에 도시된 장치에 이용되는 인코더(18)의 구성을 도시하는 블록도.

도 8은 도 7에 도시된 인코더(18)에 이용되는 예측 모드 스위칭 회로(52)의 동작을 설명하는데 이용되는 예시적 도면.

도 9는 도 7에 도시된 인코더(18)에 이용되는 예측 모드 스위칭 회로(52)의 동작을 설명하는데 이용되는 예시적 도면.

도 10은 도 7에 도시된 인코더(18)에 이용되는 예측 모드 스위칭 회로(52)의 동작을 설명하는데 이용되는 예시적 도면.

도 11은 도 7에 도시된 인코더(18)에 이용되는 예측 모드 스위칭 회로(52)의 동작을 설명하는데 이용되는 예시적 도면.

도 12는 도 5에 도시된 장치에 이용되는 디코더(31)의 구성을 도시하는 블록도.

도 13은 화상 타입에 기초한 SNR 제어를 설명하는데 이용되는 예시적 도면.

도 14는 본 발명에 의해 제공된 트랜스코더(101)의 구성을 도시하는 블록도.

도 15는 도 14에 도시된 트랜스코더(101)의 더욱 상세한 구성을 도시하는 블록도.

도 16은 도 14에 도시된 트랜스코더(101)의 복호화 장치(102)에 이용되는 디코더(111)의 구성을 도시하는 블록도.

도 17은 매크로블록의 픽셀을 도시하는 예시적 도면.

도 18은 부호화 파라미터를 기록하기 위한 영역을 도시하는 예시적 도면.

도 19는 도 14에 도시된 트랜스코더(101)의 부호화 장치(106)에 이용되는 인코더(121)의 구성을 도시하는 블록도.

도 20은 도 15에 도시된 트랜스코더(101)에 이용되는 히스토리 포맷터(211)의 주요 구성을 도시하는 블록도.

도 21은 도 15에 도시된 트랜스코더(101)에 이용되는 히스토리 디코더(203)의 주요 구성을 도시하는 블록도.

도 22는 도 15에 도시된 트랜스코더(101)에 이용되는 변환기(212)의 주요 구성을 도시하는 블록도.

도 23은 도 22에 도시된 변환기(212)에 이용되는 스터프 회로(323)의 주요 구성을 도시하는 블록도.

도 24a 내지 도 24i는 도 22에 도시된 변환기(212)의 동작을 설명하는데 이용되는 타이밍도.

도 25는 도 15에 도시된 트랜스코더(101)에 이용되는 변환기(202)의 주요 구성을 도시하는 블록도.

도 26은 도 25에 도시된 변환기(202)에 이용되는 삭제 회로(343)의 주요 구성을 도시하는 블록도.

도 27은 도 15에 도시된 트랜스코더(101)에 이용되는 변환기(212)의 또다른 주요 구성을 도시하는 블록도.

도 28은 도 15에 도시된 트랜스코더(101)에 이용되는 변환기(202)의 또다른 주요 구성을 도시하는 블록도.

도 29는 도 15에 도시된 트랜스코더(101)에 이용되는 사용자 데이터 포맷터(213)의 주요 구성을 도시하는 블록도.

도 30은 도 14에 도시된 다수의 트랜스코더(101) 각각을 이용하는 실제 시스템의 구성을 도시하는 블록도.

도 31은 부호화 파라미터를 기록하기 위한 영역을 도시하는 도면.

도 32는 변경가능한 화상 타입을 결정하도록 도 14에 도시된 트랜스코더(101)에 이용된 부호화 장치(106)에 의해 실행되는 처리를 설명하는데 이용되는 흐름도.

도 33은 화상 타입을 변경하는 예를 도시하는 도면.

도 34는 화상 타입을 변경하는 또다른 예를 도시하는 도면.

도 35는 도 14에 도시된 트랜스코더(101)에 이용된 부호화 장치(106)에 의해 실행되는 양자화 제어 처리를 설명하는데 이용되는 예시적 도면.

도 36은 도 14에 도시된 트랜스코더(101)에 이용된 부호화 장치(106)에 의해 실행되는 양자화 제어 처리를 설명하는데 이용되는 흐름도.

도 37은 빽빽하게 접속된 트랜스코더(101)의 구성을 도시하는 블록도.

도 38은 MPEG 스트림의 구문을 설명하는데 이용되는 예시적 도면.

도 39는 도 38에 도시된 구문의 구성을 설명하는데 이용되는 예시적 도면.

도 40은 고정 길이로 히스토리 정보를 기록하는 history_stream()의 구문을 설명하는데 이용되는 예시적 도면.

도 41은 고정 길이로 히스토리 정보를 기록하는 history_stream()의 구문을 설명하는데 이용되는 예시적 도면.

도 42는 고정 길이로 히스토리 정보를 기록하는 history_stream()의 구문을 설명하는데 이용되는 예시적 도면.

도 43은 고정 길이로 히스토리 정보를 기록하는 history_stream()의 구문을 설명하는데 이용되는 예시적 도면.

도 44는 고정 길이로 히스토리 정보를 기록하는 history_stream()의 구문을 설명하는데 이용되는 예시적 도면.

도 45는 고정 길이로 히스토리 정보를 기록하는 history_stream()의 구문을 설명하는데 이용되는 예시적 도면.

도 46은 고정 길이로 히스토리 정보를 기록하는 history_stream()의 구문을 설명하는데 이용되는 예시적 도면.

도 47은 가변 길이로 히스토리 정보를 기록하는 history_stream()의 구문을 설명하는데 이용되는 예시적 도면.

도 48은 sequence_header()의 구문을 설명하는데 이용되는 예시적 도면.

도 49는 sequence_extension()의 구문을 설명하는데 이용되는 예시적 도면.

도 50은 extension_and_user_data()의 구문을 설명하는데 이용되는 예시적 도면.

도 51은 user_data()의 구문을 설명하는데 이용되는 예시적 도면.

도 52는 group_of_picture_header()의 구문을 설명하는데 이용되는 예시적 도면.

도 53은 picture_header()의 구문을 설명하는데 이용되는 예시적 도면.

도 54는 picture_coding_extension()의 구문을 설명하는데 이용되는 예시적 도면.

도 55는 extension_data()의 구문을 설명하는데 이용되는 예시적 도면.

도 56은 quant_matrix_extension()의 구문을 설명하는데 이용되는 예시적 도면.

도 57은 copyright_extension()의 구문을 설명하는데 이용되는 예시적 도면.

도 58은 picture_display_extension()의 구문을 설명하는데 이용되는 예시적 도면.

도 59는 picture_data()의 구문을 설명하는데 이용되는 예시적 도면.

도 60은 slice()의 구문을 설명하는데 이용되는 예시적 도면.

도 61은 macroblock()의 구문을 설명하는데 이용되는 예시적 도면.

도 62는 macroblock_modes()의 구문을 설명하는데 이용되는 예시적 도면.

도 63은 motion_vectors(s)의 구문을 설명하는데 이용되는 예시적 도면.

도 64는 motion_vectors(r, s)의 구문을 설명하는데 이용되는 예시적 도면.

도 65는 I-화상에 대한 macroblock-type 의 가변 길이 코드를 설명하는데 이용되는 예시적 도면.

도 66은 P-화상에 대한 macroblock-type 의 가변 길이 코드를 설명하는데 이용되는 예시적 도면.

도 67은 B-화상에 대한 macroblock-type 의 가변 길이 코드를 설명하는데 이용되는 예시적 도면.

@ 도면의 주요 부분에 대한 부호의 설명 @

1 : 부호화 장치 2 : 복호화 장치

3 : 기록 매체 11 : 사전 처리 회로

12,13 : A/D 변환기 14 : 프레임 메모리

15 : 휘도 신호 프레임 메모리 16 :색차 신호 프레임 메모리

17 : 포맷 변환 회로 18 : 인코더

19 : 기록 회로 30 : 재생 회로

31 : 디코더 32 : 포맷 변환 회로

33 : 프레임 메모리 36,37 : D/A 변환기

38 : 사후 처리 회로

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 MPEG (Moving Picture Experts Group) 규격을 근거로 한 부호화 처리 결과로 구해진 부호화 비트스트림(bitstream)의 비트 레이트와 GOP (Group of Pictures) 구조를 변환시키기 위한 트랜스코딩(transcoding) 시스템, 비디오 부호화 장치, 스트림 처리 시스템, 및 비디오 복호화 장치에 관한 것이다.

최근에는 텔레비젼 프로그램을 제작하여 방송하는 방송국이 일반적으로 비디오 데이터의 압축 및 부호화를 위해 MPEG 기술을 사용하고 있다. 특히, MPEG 기술은 비디오 데이터를 테이프나 랜덤 억세스가능한 기록 매체에 기록하고 케이블이나 위성을 통해 비디오 데이터를 전송하기 위한 사실상의 규격이 되고 있다.

다음에는 방송국에서 제작된 비디오 프로그램을 각 가정에 전송할 때까지 방송국에 의해 실행되는 전형적인 처리를 간략히 설명한다. 먼저, 캠코더(비디오 카메라와 VTR을 단일 본체에 집적한 장치)에서 사용되는 인코더는 소스 비디오 데이터를 부호화하고, 부호화된 데이터를 VTR의 자기 테이프상에 기록한다. 그때, 캠코더에서 사용되는 인코더는 소스 비디오 데이터를 VTR의 자기 테이프의 기록 포맷에 적절한 부호화 비트스트림으로 부호화한다. 전형적으로, 자기 테이프상에 기록된 MPEG 비트스트림의 GOP 구조는 하나의 GOP가 2개의 프레임(frame)으로 구성되는 구조이다. GOP의 구조의 예는 I-, B-, I-, B-, I-, B- 등의 형태의 화상 시퀀스를 구비하는 구조이다. 자기 테이프상에 기록되는 MPEG 비트스트림의 비트 레이트는 18 Mbps이다.

이때, 중앙 방송국은 자기 테이프상에 기록된 비디오 비트스트림을 편집하는 편집 처리를 실행한다. 이를 위해, 자기 테이프상에 기록된 비디오 비트스트림의 GOP 구조는 편집 처리에 적절한 GOP 구조로 변환된다. 편집 처리에 적절한 GOP 구조는 하나의 GOP가 하나의 프레임으로 구성되는 구조이다. 보다 특정하게, 편집 처리에 적절한 GOP 구조의 화상은 모두 I-화상이다. 이는 프레임 단위로 편집 처리를 실행하기 위해서는 다른 화상과의 상관관계를 갖지 않는 I-화상이 가장 적절하기 때문이다. GOP 구조를 변환시키는 실제 동작에서, 자기 테이프상에 기록된 비디오 비트스트림은 일단 기저대(base-band) 비디오 데이터로 다시 복호화된다. 이어서, 기저대 비디오 데이터는 모두 I-화상을 구비하도록 재부호화된다. 이 방법으로 복호화 및 재부호화 처리를 실행함으로써, 편집 처리에 적절한 GOP 구조를 갖는 비트스트림을 생성하는 것이 가능하다.

이어서, 편집 처리의 결과로 구해진 편집된 비디오 프로그램을 중앙 방송국에서 지역 방송국으로 전송하기 위해서는 편집된 비디오 프로그램의 비트스트림의 비트 레이트 및 GOP 구조를 전송에 적절한 비트 레이트 및 GOP 구조로 변환시킬 필요가 있다. 방송국간의 전송에 적절한 GOP 구조는 하나의 GOP가 15개의 프레임으로 구성되는 GOP 구조이다. 이러한 GOP 구조의 예는 I-, B-, B-, P-, B-, B-, P- 등의 형태의 시퀀스를 구비하는 구조이다. 방송국간의 전송에 적절한 비트 레이트에 대해, 일반적으로 방송국간에는 광섬유와 같은 고전송용량을 갖는 전용선이 설치되므로, 적어도 50 Mbps의 높은 비트 레이트가 바람직하다. 구체적으로, 편집 처리가 완료된 비디오 프로그램의 비트스트림은 일단 기저대 비디오 데이터로 다시 복호화된다. 이어서, 기저대 비디오 데이터는 상술된 바와 같이 방송국간의 전송에 적절한 비트 레이트 및 GOP 구조가 되도록 재부호화된다.

지역 방송국에서는 전형적으로 중앙 방송국으로부터 수신된 비디오 프로그램에 편집 처리가 행해져 지역 방송국이 위치하는 지방에 특이한 광고를 삽입한다. 중앙 방송국에서 실행되는 편집 처리와 같이, 중앙 방송국으로부터 수신된 비디오 프로그램의 비트스트림은 일단 기저대 비디오 데이터로 다시 복호화된다. 이어서, 기저대 비디오 데이터는 모두 I-화상을 구비하도록 복호화된다. 그 결과로, 편집 처리에 적절한 GOP 구조를 갖는 비트스트림을 생성하는 것이 가능하다.

이어서, 지역 방송국에서의 편집 처리를 완료한 이후에 비디오 프로그램을 케이블이나 위성을 통해 각 가정에 전송하기 위해서는 비트스트림의 비트 레이트 및 GOP 구조가 각 가정으로의 전송에 적절한 비트 레이트 및 GOP 구조로 각각 변환된다. 각 가정으로의 전송에 적절한 GOP 구조는 하나의 GOP가 15개의 프레임으로 구성되는 구조이다. 이러한 GOP 구조의 예는 I-, B-, B-, P-, B-, B-, P- 등의 형태의 화상 시퀀스를 구비하는 구조이다. 각 가정으로의 전송에 적절한 비트 레이트는 전형적으로 약 5 Mbps 정도로 낮은 값을 갖는다. 편집 처리를 완료한 비디오 프로그램의 비트스트림은 때로 기저대 비디오 데이터로 다시 복호화된다. 이어서, 기저대 비디오 데이터는 각 가정으로의 전송에 적절한 비트 레이트 및 GOP 구조로 재부호화된다.

상기의 설명으로부터 명백한 바와 같이, 중앙 방송국에서 각 가정으로 전송되는 비디오 프로그램에는 전송되는 동안 반복적인 복호화 및 부호화 처리가 여러번 행해진다. 실제로, 상술된 신호 처리 이외의 다양한 종류의 신호 처리가 방송국에서 실행되고, 각 종류의 신호 처리를 위해서는 통상적으로 복호화 및 부호화 처리가 실행된다. 그 결과로, 복호화 및 부호화 처리는 반복적으로 실행될 필요가 있다.

그러나, MPEG 규격을 근거로 한 부호화 및 복호화 처리는 일반적으로 공지된 바와 같이 100 퍼센트 서로 역처리되지 않는다. 특정하게, 부호화 처리가 행해진 기저대 비디오 데이터는 이전 생성의 트랜스코딩(transcoding)에서 실행된 복호화 처리의 결과로 구해지는 비디오 데이터와 완전히 똑같지 않다. 그러므로, 복호화 및 부호화 처리에 의해 화상 품질이 열화된다. 그 결과로, 복호화 및 부호화 처리가 실행될 때마다 일어나는 화상 품질의 열화에 관한 문제점이 있다. 다른 말로 하면, 화상 품질의 열화의 효과는 복호화 및 부호화 처리가 반복될 때마다 누적된다.

발명이 이루고자 하는 기술적 과제

이와 같이, 상술된 문제점을 해결하는 본 발명의 목적은 비트스트림의 비트 레이트 및 GOP 구조를 변환시키기 위해 MPEG 규격을 근거로 한 부호화 처리가 완료된 비트스트림에 부호화 및 복호화 처리가 반복적으로 처리되는 경우라도 화상 품질의 열화를 일으키지 않는 트랜스코딩(transcoding) 시스템, 비디오 부호화 장치, 스트림(stream) 처리 시스템, 및 비디오 복호화 장치를 제공하는 것이다. 부가하여, 본 발명의 목적은 복호화 및 부호화 처리가 반복적으로 처리되는 경우라도 화상 품질의 열화를 일으키지 않는 트랜스코딩 시스템, 비디오 부호화 장치, 스트림 처리 시스템, 및 비디오 복호화 장치를 제공하는 것이다.

발명의 구성 및 작용

상기의 목적을 이루기 위해, 본 발명에 의해 제공되는 트랜스코더(transcoder)에 따라, 이전 부호화 처리에서 생성되고 사용된 부호화 파라미터는 현재 부호화 처리에서 사용될 수 있다. 그 결과로, 화상 품질은 복호화 및 부호화 처리가 반복적으로 실행되는 경우라도 열화되지 않는다. 말하자면, 부호화 처리의 반복으로 인한 화상 품질의 누적 열화를 감소시키는 것이 가능하다.

본 발명에 의해 제공되는 트랜스코더에 따라, 이전 부호화 처리에서 생성되고 사용된 부호화 파라미터는 현재 부호화 처리의 결과로 구해지는 복호화 비트스트림의 사용자 데이터 영역에서 설명되고, 부호화된 비트스트림은 MPEG 규격에 따른다. 그래서, 현존하는 디코더로 복호화된 스트림을 복호화하는 것이 가능하다. 부가하여, 이전 부호화 처리에서 생성되고 사용된 부호화 파라미터를 전송하기 위한 전용선을 제공할 필요가 없다. 그 결과로, 현존하는 데이터 스트림 전송 환경을 사용해 이전 부호화 처리에서 생성되고 사용된 부호화 파라미터를 전송하는 것이 가능하다.

본 발명에 의해 제공되는 트랜스코더에 따라, 부호화 처리에서 생성되고 사용된 부호화 파라미터 중 선택된 이전의 부호화 파라미터만은 현재 부호화 처리의 결과로 구해진 부호화 비트스트림의 사용자 데이터 영역에서 설명된다. 그 결과로, 출력 비트스트림의 비트 레이트를 실질적으로 증가시킬 필요없이 과거에 실행된 부호화 처리에서 생성되고 사용된 부호화 파라미터를 전송하는 것이 가능하다.

본 발명에 의해 제공되는 트랜스코더에 따라, 이전 부호화 처리에서 생성되고 사용된 부호화 파라미터로부터 현재 부호화 처리에 대해 최적인 부호화 파라미터만이 선택되어 현재 부호화 처리에서 사용된다. 그 결과로, 이전 복호화 및 부호화 처리가 반복적으로 실행되는 경우라도 화상 품질의 열화가 누적되지 않는다.

본 발명에 의해 제공되는 트랜스코더에 따라, 이전 부호화 처리에서 생성되고 사용된 이전 부호화 파라미터로부터 현재 부호화 처리에 대해 최적인 부호화 파라미터만이 화상 타입에 따라 선택되어 현재 부호화 처리에서 사용된다. 그 결과로, 복호화 및 부호화 처리가 반복적으로 실행되는 경우라도 화상 품질의 열화가 결코 누적되지 않는다.

본 발명에 의해 제공되는 트랜스코더에 따라, 이전 부호화 처리에서 생성되고 사용된 이전 부호화 파라미터를 재사용하는가 여부에 관한 결정은 이전 부호화 파라미터에 포함된 화상 타입을 근거로 이루어진다. 그 결과로, 최적의 부호화 처리가 이루어질 수 있다.

본 발명에 의해 제공되는 트랜스코더를 설명하기 이전에, 동화상 비디오 신호를 압축 및 부호화하는 처리가 설명된다. 본 명세서에서 사용되는 '시스템'이란 말은 다수의 장치 및 수단을 구비하는 전체적인 시스템을 의미함을 주목하여야 한다.

상술된 바와 같이, 동화상 비디오 신호를 원격 착신지에 전송하는 시스템, 예를 들면 텔레비젼 회담 시스템 및 텔레비젼 전화기 시스템에서는 전송선이 높은 효율성으로 사용되는 것을 허용하기 위해 비디오 신호의 선(line) 상관관계 및 프레임내(intra-frame) 상관관계를 사용해 비디오 신호에 압축 및 부호화 처리가 행해진다. 선 상관관계(line correlation)를 사용함으로써, 비디오 신호는 전형적으로 DCT (Discrete Cosine Transform) 처리를 실행하여 압축될 수 있다.

프레임내 상관관계를 사용함으로써, 비디오 신호는 더 압축 및 부호화될 수 있다. 도 1에 도시된 바와 같이, 프레임 화상 (PC1), (PC2), 및 (PC3)은 각각 시간점 (t1), (t2), 및 (t3)에서 생성된다고 가정한다. 이 경우에는 프레임 화상 (PC1)과 (PC2) 사이의 화상 신호차가 계산되어 프레임 화상(PC12)를 생성한다. 같은 방식으로, 프레임 화상 (PC2)와 (PC3) 사이의 화상 신호차가 계산되어 프레임 화상(PC23)을 생성한다. 일반적으로, 시간축을 따라 서로 인접한 프레임 화상 사이의 화상 신호차는 작다. 그래서, 프레임 화상 (PC12) 및 (PC23)에 포함된 정보량은 작고, 이러한 차이를 부호화한 결과로 구해진 차이 신호에 포함된 코드량도 또한 작다.

그러나, 단순히 차이 신호를 전송하는 것으로, 원래 화상이 재저장될 수는 없다. 원래 화상을 구하기 위해서는 프레임 화상이 3가지 종류, 즉 비디오 신호의 압축 및 부호화 처리에서 각각 최소 처리 유닛으로 사용되는 I-, P-, 및 B-화상으로 분류된다.

처리를 위한 비디오 신호의 최소 단위로 각각 처리되는 17개의 프레임, 즉 프레임 (F1) 내지 (F17)을 구비하는 도 2의 GOP (Group of Pictures)를 가정한다. 특정하게, 제 1 프레임(F1), 제 2 프레임(F2), 및 제 3 프레임(F3)은 각각 I-, B-, 및 P-화상으로 처리된다. 이어지는 프레임들, 즉 제 4 내지 제 17 프레임 (F4) 내지 (F17)은 B- 및 P-화상으로 번갈아 처리된다.

I-화상의 경우에서는 전체 프레임의 비디오 신호가 전송된다. 한편, P-화상 또는 B-화상의 경우에서는 비디오 신호의 차이만이 전체 프레임의 비디오 신호에 대한 대안으로 전송될 수 있다. 특정하게, 도 2에 도시된 P-화상의 제 3 프레임(F3)의 경우에서는 P-화상과 선행하는 I- 또는 P-화상 사이의 비디오 신호차만이 비디오 신호로 전송된다. 예를 들어, 도 3에 도시된 B-화상의 제 2 프레임(F2)의 경우에서는 B-화상과, 선행하는 프레임, 이어지는 프레임, 또는 선행하는 프레임과 이어지는 프레임의 평균값 사이의 비디오 신호차가 비디오 신호로 전송된다.

도 4는 상술된 바에 따라 동화상 비디오 신호를 부호화하는 기술의 원리를 도시하는 도면이다. 도 4에 도시된 바와 같이, 제 1 프레임(F1)은 I-화상으로 처리된다. 그래서, 전체 프레임(F1)의 비디오 신호는 데이터(F1Z)로 전송선에 전송된다(화상내 부호화 (intra-picture encoding)). 한편, 제 2 프레임(F2)은 B-화상으로 처리된다. 이 경우에는 제 2 프레임(F2)과, 선행하는 프레임(F1), 이어지는 프레임(F3), 또는 선행하는 프레임(F1)과 이어지는 프레임(F3)의 평균값 사이의 차이가 데이터(F2X)로 전송된다.

상세히 설명하기 위해, B-화상의 처리는 4가지 종류로 분류될 수 있다. 제 1 처리형에서는 도 4에서 표시(SP1)으로 나타내지는 원래 프레임(F2)의 데이터가 I-화상의 경우와 같이 데이터(F2X)로 전송된다. 그래서, 제 1 처리형은 화상내 부호화라 칭하여진다. 제 2 처리형에서는 표시(SP2)로 나타내지는 제 2 프레임(F2)과 이어지는 제 3 프레임(F3) 사이의 차이가 데이터(F2X)로 전송된다. 이어지는 프레임이 기준 또는 예측 화상으로 취해지므로, 이 처리는 후방 예측 부호화 (backward prediction encoding)라 칭하여진다. 제 3 처리형에서는 표시(SP3)으로 나타내지는 제 2 프레임(F2)과 선행하는 제 1 프레임(F1) 사이의 차이가 P-화상의 경우와 같이 데이터(F2X)로 전송된다. 선행하는 프레임이 예측 화상으로 취해지므로, 이 처리는 전방 예측 부호화 (forward prediction encoding)라 칭하여진다. 제 4 처리형에서는 표시(SP4)로 나타내지는 제 2 프레임(F2)과, 이어지는 제 3 프레임(F3) 및 선행하는 제 1 프레임(F1)의 평균 사이의 차이가 데이터 (F2X)로 전송된다. 선행하는 프레임과 이어지는 프레임이 예측 화상으로 취해지므로, 이 처리는 전후방 예측 부호화 (forward & backward prediction encoding)라 칭하여진다. 실제로, 처리 결과로 구해지는 최소량의 전송 데이터를 생성하기 위해 상술된 4가지 처리형 중 하나가 선택된다.

제 2, 제 3, 또는 제 4 처리형 결과로 구해진 차이의 경우에서는 차이의 계산에서 사용되는 프레임의 화상(예측 화상) 사이의 이동 벡터가 또한 차이를 따라 전송됨을 주목하여야 한다. 특정하게, 전방 예측 부호화의 경우에서는 이동 벡터가 프레임 (F1)과 (F2) 사이의 이동 벡터(x1)이다. 후방 예측 부호화의 경우에서는 이동 벡터가 프레임 (F2)와 (F3) 사이의 이동 벡터(x2)이다. 전후방 예측 부호화의 경우에서는 이동 벡터 (x1) 및 (x2)가 모두 전송된다.

상술된 B-화상과 같이, P-화상의 프레임(F3)인 경우에는 처리 결과로 구해지는 최소량의 전송 데이터를 만들기 위해 전방 예측 부호화나 화상내 처리가 선택된다. 전방 예측 부호화가 선택되면, 표시(SP3)로 나타내지는 제 3 프레임(F3)과 선행하는 제 1 프레임(F1) 사이의 차이는 이동 벡터(x3)와 함께 데이터(F3X)로 전송된다. 한편, 화상내 처리가 선택되면, 표시(SP1)으로 나타내지는 원래 프레임(F3)의 데이터(F3X)가 전송된다.

도 5는 상술된 원리를 근거로 동화상 비디오 신호를 부호화하고 부호화된 신호를 전송 및 복호화하는 시스템의 전형적인 구조를 도시하는 블록도이다. 신호 부호화 장치(1)는 입력 비디오 신호를 부호화하고, 부호화된 비디오 신호를 전송선으로 동작되는 기록 매체(3)를 통해 신호 복호화 장치(2)로 전송한다. 신호 복호화 장치(2)는 기록 매체(3)상에 기록된 부호화 신호를 재생하고, 재생 신호를 출력 신호로 복호화한다.

신호 부호화 장치(1)에서, 입력 비디오 신호는 사전 처리 회로(11)로 공급되어 휘도(luminance) 및 색도(chrominance) 신호로 분리된다. 본 실시예의 경우에, 색도 신호는 색차(color-difference) 신호이다. 아날로그 휘도 및 색차 신호는 각각 A/D 변환기 (12) 및 (13)로 공급되어 각기 디지털 비디오 신호로 변환된다. A/D 변환의 결과로 주어지는 디지털 비디오 신호는 프레임-메모리 유닛(14)에 공급되어 그에 저장된다. 프레임-메모리 유닛(14)은 휘도 신호를 저장하기 위한 휘도-신호 프레임 메모리(15)와 색차 신호를 저장하기 위한 색차-신호 프레임 메모리(16)를 구비한다.

포맷 변환 회로(17)는 프레임-메모리 유닛(14)에 저장된 프레임-포맷 신호를 도 6a 내지 도 6c에 도시된 바와 같은 블록(block)-포맷 신호로 변환한다. 상세하게, 비디오 신호는 도 6a에 도시된 프레임 포맷의 데이터로 프레임-메모리 유닛(14)에 저장된다. 도 6a에 도시된 바와 같이, 프레임 포맷은 각각이 H개 도트(dot)을 구비하는 V개 선(line)의 집합이다. 포맷 변환 회로(17)는 한 프레임의 신호를 도 6b에 도시된 바와 같이 각각이 16개 선을 구비하는 N개의 슬라이스(slice)로 나눈다. 이어서, 각 슬라이스는 도 6b에 도시된 바와 같이 M개의 매크로블록(macroblock)으로 나뉜다. 도 6c에 도시된 바와 같이, 매크로블록은 16 x 16 픽셀(도트)에 대응하는 휘도 신호(Y)를 포함한다. 휘도 신호(Y)는 각각이 8 x 8 도트를 구비하는 블록 Y[1] 내지 Y[4]로 더 나뉜다. 16 x 16 -도트 휘도 신호는 8 x 8 -도트 Cb 신호와 8 x 8 -도트 Cr 신호와 연관된다.

상술된 바와 같이 포맷 변환 회로(17)에 의해 실행되는 포맷 변환의 결과로 구해지는 블록 포맷을 갖는 데이터는 데이터를 부호화하는 인코더(18)에 공급된다. 인코더(18)의 구성은 도 7을 참고로 추후 더 상세히 설명된다.

인코더(18)에 의해 실행되는 인코더 결과로 구해진 신호는 전송선에 비트스트림으로 출력된다. 전형적으로, 부호화된 신호는 전송선으로 사용되는 기록 매체(3)에 부호화된 신호를 디지털 신호로 기록하기 위한 기록 회로(19)에 공급된다.

신호 복호화 장치(2)에서 사용되는 재생 회로(30)는 기록 매체(3)로부터 데이터를 재생하고, 그 데이터를 복호화 장치의 디코더(31)에 공급하여 데이터를 복호화한다. 디코더(31)의 구조는 도 12를 참고로 추후 상세히 설명된다.

디코더(31)에 의해 실행되는 복호화 결과로 구해진 데이터는 데이터의 블록 포맷을 다시 프레임 포맷으로 변환하도록 포맷 변환 회로(32)로 공급된다. 이어서, 프레임 포맷을 갖는 휘도 신호는 프레임-메모리 유닛(33)의 휘도-신호 프레임 메모리(34)에 공급되어 그에 저장된다. 한편, 프레임 포맷을 갖는 색차 신호는 프레임 메모리 유닛(33)의 색차-신호 프레임 메모리(35)에 공급되어 그에 저장된다. 휘도 신호는 다시 휘도-신호 프레임 메모리(34)로부터 판독되어 D/A 변환기(36)에 공급된다. 한편, 색차 신호는 다시 색차-신호 프레임 메모리(35)로부터 판독되어 D/A 변환기(37)에 공급된다. D/A 변환기(36) 및 (37)는 그 신호를 아날로그 신호로 변환하여, 휘도 및 색차 신호를 합성하고 합성된 출력을 생성하는 사후 처리 회로(38)에 공급된다.

다음에는 도 7을 참고로 인코더(18)의 구성이 설명된다. 부호화될 화상 데이터는 이동벡터 검출 회로(50)에 매크로블록 단위로 공급된다. 이동벡터 검출 회로(50)는 미리 설정된 소정의 시퀀스에 따라 각 프레임의 화상 데이터를 I-, P-, 또는 B-화상으로 처리한다. 특정하게, 도 2 및 도 3에 도시된 바와 같이 전형적으로 프레임 (F1) 내지 (F17)을 구비하는 GOP의 화상 데이터는 I-, B-, P-, B-, P-, ..., B-, 및 P-화상의 시퀀스로 처리된다.

도 3에 도시된 프레임(F1)과 같이 I-화상으로 이동벡터 검출 회로(50)에 의해 처리된 프레임의 화상 데이터는 프레임 메모리 유닛(51)의 전방 소스 화상 영역(51a)에 공급되어 그에 저장된다. 프레임(F1)과 같이 P-화상으로 이동벡터 검출 회로(50)에 의해 처리되는 프레임의 화상 데이터는 프레임 메모리 유닛(51)의 기준 소스 화상 영역(51b)에 공급되어 그에 저장된다. 프레임(F3)과 같이 P-화상으로 이동벡터 검출 회로(50)에 의해 처리되는 프레임의 화상 데이터는 프레임 메모리 유닛(51)의 후방 소스 화상 영역(51c)에 공급되어 그에 저장된다.

프레임 (F4) 또는 (F5)와 같이 다음 두 프레임의 화상 데이터가 순차적으로 이동벡터 검출 회로(50)에 공급되어 각각 B 및 P-화상으로 처리될 때, 영역 (51a), (51b), 및 (51c)는 다음과 같이 업데이트된다. 프레임(F4)의 화상 데이터가 이동벡터 검출 회로(50)에 의해 처리될 때, 후방 소스 화상 영역(51c)에 저장된 프레임(F3)의 화상 데이터는 전방 소스 화상 영역(51a)에 전달되고, 소스 화상 영역(51b)에 먼저 저장된 프레임(F1)의 화상 데이터를 오버라이트 (overwrite)한다. 처리된 프레임(F4)의 화상 데이터는 기준 소스 화상 영역(51b)에 저장되고, 기준 소스 화상 영역(51b)에 먼저 저장된 프레임(F2) 상에 화상 데이터를 오버라이트한다. 이어서, 처리된 프레임(F5)의 화상 데이터는 후방 소스 화상 영역(51c)에 저장되고, 전방 소스 화상 영역(51a)에 전달되었던 프레임(F3)의 화상 데이터를 오버라이트한다. 상술된 동작은 반복되어 이어지는 GOP의 프레임을 처리한다.

프레임 메모리 유닛(51)에 저장된 각 화상 신호는 예측모드 스위칭 회로(52)에 의해 판독되어, 프레임-예측 모드 또는 필드(field)-예측 모드에 대해, 즉 처리 유닛(53)에 의해 실행되는 처리 종류에 대해 준비된 동작을 행한다.

프레임-예측 모드나 필드 예측 모드에서, 신호에는 화상내 처리/전방/후방/전후방 예측 결정 회로(54)에 의해 실행되는 제어하에서 전방, 후방, 및 전후방 예측과 같은 화상내 예측 부호화를 구하도록 계산이 행해진다. 처리 유닛(53)에서 실행되는 처리 종류는 기준 화상과 기준 화상에 대한 예측 화상 사이의 차이를 나타내는 예측 에러 신호에 따라 결정된다. 기준 화상은 처리가 행해지는 화상이고, 예측 화상은 기준 화상에 선행하거나 이어지는 화상이다. 이러한 이유로, 이동벡터 검출 회로(50) (엄밀히 말하면, 추후 기술될 벡터 검출 회로(50)에서 사용되는 예측모드 스위칭 회로(52))는 처리 유닛(53)에 의해 실행되는 처리 종류의 결정시 사용되는 예측 에러 신호의 절대값의 합을 생성한다. 예측 에러 신호의 절대값의 합 대신에, 예측 에러 신호의 제곱의 합이 또한 이러한 결정을 위해 사용될 수 있다.

예측모드 스위칭 회로(52)는 프레임 예측 모드 및 필드 예측 모드에서 처리 유닛(53)에 의해 실행될 처리에 대한 다음의 준비 동작을 실행한다.

예측모드 스위칭 회로(52)는 이동벡터 검출 회로(50)에 의해 공급되는 4개의 휘도 블록 [Y1] 내지 [Y4]를 수신한다. 각 블록에는 홀수 필드선의 데이터가 도 8에 도시된 바와 같이 짝수 필드선의 데이터와 혼합된다. 데이터는 그대로 처리 유닛(53)으로 전달된다. 처리 유닛(53)에 의해 실행되는 도 8에 도시된 구성을 갖는 데이터의 처리는 프레임-예측 모드에서의 처리라 칭하여지고, 여기서 예측 처리는 4개의 휘도 블록을 구비하는 각 매크로블록에 대해 실행되고 이동벡터는 4개의 휘도 블록에 대응한다.

이어서, 예측모드 스위칭 회로(52)는 이동벡터 검출 회로(50)에 의해 공급된 신호를 재구성한다. 도 8에 도시된 구성을 갖는 신호 대신에, 도 9에 도시된 구성을 갖는 신호가 처리 유닛(53)으로 전달된다. 도 9에 도시된 바와 같이, 두 휘도 블록 [Y1] 및 [Y2]는 각각 전형적으로 홀수 필드선의 도트만으로 구성되는 반면, 다른 두 휘도 블록 [Y3] 및 [Y4]는 각각 전형적으로 짝수 필드선의 도트만으로 구성된다. 처리 유닛(53)에 의해 실행되는 도 9에 도시된 구성을 갖는 데이터의 처리는 필드-예측 모드에서의 처리라 칭하여지고, 여기서 이동벡터는 2개의 휘도 블록 [Y1] 및 [Y2]에 대응하는 반면, 또 다른 이동벡터는 다른 두 휘도 블록 [Y3] 및 [Y4]에 대응한다.

예측모드 스위칭 회로(52)는 도 8 또는 도 9에 도시된 구성을 갖는 데이터를 선택하여, 다음과 같이 처리 유닛(53)에 공급한다. 예측모드 스위칭 회로(52)는 프레임-예측 모드에 대해, 즉 도 8에 도시된 구성을 갖고 이동벡터 검출 회로(50)에 의해 공급되는 데이터에 대해 계산된 예측 에러의 절대값의 합과, 필드-예측 모드에 대해, 즉 도 8에 도시된 구성을 갖는 데이터의 변환 결과로 구해지는 도 9에 도시된 구성을 갖는 데이터에 대해 계산된 예측 에러의 절대값의 합을 계산한다. 예측 에러는 추후 상세히 설명됨을 주목하여야 한다. 이어서, 예측모드 스위칭 회로(52)는 더 작은 합을 산출하는 모드를 결정하기 위해 데이터에 대해 계산된 예측 에러의 절대값의 합을 비교한다. 예측모드 스위칭 회로(52)는 더 작은 합을 산출하는 프레임-예측 모드 또는 필드-예측 모드에 대해 각각 도 8 또는 도 9에 도시된 구성 중 하나를 선택한다. 예측모드 스위칭 회로(52)는 마지막으로 선택된 구성에 대응하는 모드로 데이터를 처리하도록 선택된 구성을 갖는 데이터를 처리 유닛(53)에 출력한다.

실제적으로, 예측모드 스위칭 회로(52)는 이동벡터 검출 회로(50)에 포함됨을 주목하여야 한다. 말하자면, 도 9에 도시된 구성을 갖는 데이터의 준비, 절대값의 계산, 절대값의 비교, 데이터 구성의 선택, 및 선택된 구성을 갖는 데이터를 처리 유닛(53)에 출력하는 동작은 모드 이동벡터 검출 회로(50)에 의해 실행되고, 예측모드 스위칭 회로(52)는 단순히 이동벡터 검출 회로(50)에 의해 공급된 신호를 처리 유닛(53)의 나중 스테이지에 출력한다.

프레임-예측 모드에서, 색차 신호는 도 8에 도시된 바와 같이 짝수 필드선의 데이터와 혼합된 홀수 필드선의 데이터와 함께 처리 유닛(53)으로 공급됨을 주목하여야 한다. 한편, 도 9에 도시된 필드-예측 모드에서는 색차 블록(Cb)의 상단부 반

의 4개 선이 휘도 블록 [Y1] 및 [Y2]에 대응하는 홀수 필드의 색차 신호로 사용되고, 색차 블록(Cb)의 하단부 반의 4개 선이 휘도 블록 [Y3] 및 [Y4]에 대응하는 짝수 필드의 색차 신호로 사용된다. 유사하게, 색차 블록(Cr)의 상단부 반의 4개 선은 휘도 블록 [Y1] 및 [Y2]에 대응하는 홀수 필드의 색차 신호로 사용되고, 색차 블록(Cr)의 하단부 반의 4개 선은 휘도 블록 [Y3] 및 [Y4]에 대응하는 짝수 필드의 색차 신호로 사용된다.

상술된 바와 같이, 이동벡터 검출 회로(50)는 처리 유닛(53)이 화상내 예측, 전방 예측, 후방 예측, 또는 전후방 예측을 실행하여야 하는가 여부를 결정하는데 사용하도록 예측 에러의 절대값의 합을 예측 결정 회로(54)에 출력한다.

상세하게, 화상내 예측에서는 예측 에러의 절대값의 합이 다음과 같이 찾아진다. 화상내 예측에 대해, 이동벡터 검출 회로(50)는 기준 화상의 매크로블록에서 신호(Aij)의 합(∑Aij)의 절대값 ∣∑Aij∣과, 같은 기준 화상의 매크로블록에서 신호(Aij)의 절대값∣Aij∣의 합(∑∣Aij∣) 사이의 차이를 계산한다. 전방 예측에 대해, 예측 에러의 절대값의 합은 기준 화상의 매크로블록에서의 신호(Aij)와 전방 예측 화상이나 선행하는 화상의 매크로블록에서의 신호(Bij) 사이에서의 차이(Aij - Bij)의 절대값 ∣Aij - Bij∣의 합(∑∣Aij - Bij∣)이다. 전방 예측에 대한 예측 에러의 절대값의 합은 후방 예측에서 사용되는 예측 화상이 후방 예측 화상이나 이어지는 화상인 것을 제외하면 전방 예측에 대한 것과 같은 방법으로 찾아진다. 전후방 예측에 대해서는 전방 예측 화상 및 후방 예측 화상 모두의 매크로블록에서의 신호(Bij) 평균이 합의 계산에서 사용된다.

각 예측 기술에 대한 예측 에러의 절대값의 합은 최소합을 갖는 전방 예측, 후방 예측, 또는 전후방 예측을 화상간 예측(inter-picture prediction)에 대한 예측 에러의 절대값의 합으로 선택하는 예측 결정 회로(54)에 공급된다. 예측 결정 회로(54)는 또한 최소합을 화상내 예측에 대한 합과 비교하고, 훨씬 더 작은 합을 갖는 화상내 예측 또는 화상간 예측을 처리 유닛(53)에 의해 실행되는 처리의 예측 모드로 선택한다. 특정하게, 화상내 예측에 대한 합이 화상간 예측에 대한 최소합보다 더 작은 것으로 발견되면, 예측 결정 회로(54)는 화상내 예측을 처리 유닛(53)에 의해 실행되는 처리형으로 선택한다. 한편, 화상간 예측에 대한 최소합이 화상내 예측에 대한 합 보다 더 작은 것으로 발견되면, 예측 결정 회로(54)는 화상간 예측을 처리 유닛(53)에 의해 실행되는 처리형으로 선택한다. 상술된 바와 같이, 화상간 예측은 최소합을 갖는 처리의 예측 모드로 선택된 전방 예측, 후방 예측, 또는 전후방 예측을 나타낸다. 예측 모드는 각 화상 (또는 프레임)에 대해 결정되지만, 프레임-예측이나 필드-예측 모드는 각 화상 그룹에 대해 결정된다.

상술된 바와 같이, 이동벡터 검출 회로(50)는 예측모드 스위칭 회로(52)에 의해 선택된 프레임-예측 모드나 필드-예측 모드에 대한 기준 화상의 매크로블록에서의 신호를 예측모드 스위칭 회로(52)를 통해 처리 유닛(53)에 출력한다. 동시에, 이동벡터 검출 회로(50)는 또한 예측 결정 회로(54)에 의해 선택된 4개의 예측 모드 중 하나에 대해 기준 화상과 예측 화상 사이의 이동 벡터를 검출한다. 이어서, 이동벡터 검출 회로(50)는 그 이동 벡터를 가변길이 부호화 회로(58) 및 이동 보상 회로(64)에 출력한다. 이 방법으로, 이동벡터 검출 회로(50)는 상술된 바와 같이 선택된 예측 모드에 대한 예측 에러의 절대값의 최소합에 대응하는 이동 벡터를 출력한다.

이동벡터 검출 회로(50)가 전방 소스 화상 영역(51a)으로부터 GOP의 제 1 프레임인 I-화상의 화상 데이터를 판독하고 있을 때, 예측 결정 회로(54)는 예측 모드로 화상내 예측, 엄밀히 말하면, 프레임내 또는 필드내 예측을 설정하도록 처리 유닛(53)에서 사용되는 스위치(53d)를 접촉점(a)에 설정한다. 스위치(53d)를 이 위치에 설정하면, I-화상의 데이터는 DCT-모드 스위칭 회로(55)에 공급된다. 추후 기술될 바와 같이, 화상내 예측 모드는 이동 보상이 실행되지 않는 모드이다.

DCT-모드 스위칭 회로(55)는 도 10에 도시된 프레임-DCT 모드나 혼합 상태로 예측모드 스위칭 회로(52)에서 스위치(53d)에 의해 전달된 데이터를 수신한다. 이어서, DCT-모드 스위칭 회로(55)는 그 데이터를 도 11에 도시된 필드-DCT 모드나 분리된 상태로 변환한다. 프레임-DCT 모드에서는 홀수 및 짝수 필드선의 데이터가 4개의 휘도 블록 각각에서 혼합된다. 한편, 필드-DCT 모드에서, 홀수 필드선은 4개의 휘도 블록 중 2개에 놓이고, 짝수 필드선은 다른 두 블록에 놓인다. 혼합 또는 분리 상태에 있는 I-화상의 데이터는 DCT 회로(56)에 공급된다.

데이터를 DCT 회로(56)로 공급하기 이전에, DCT-모드 스위칭 회로(55)는 서로 혼합된 짝수 및 홀수 필드선 데이터의 DCT 처리에 대한 부호화 효율성을 서로 분리된 짝수 및 홀수 필드선 데이터의 DCT 처리에 대한 부호화 효율성과 비교하여, 더 높은 효율성을 갖는 데이터를 선택한다. 선택된 데이터에 대응하는 프레임-DCT 모드나 필드-DCT 모드는 DCT 모드로 결정된다.

부호화 효율성은 다음과 같이 서로 비교된다. 도 10에 도시된 바와 같이 서로 혼합된 짝수 및 홀수 필드선 데이터인 경우에는 짝수 필드선의 신호와 그 짝수 필드에 수직으로 인접한 홀수 필드선의 신호 사이의 차이가 계산된다. 이어서, 그 차이의 절대값의 합이나 제곱값이 찾아진다. 마지막으로, 인접한 두 짝수 및 홀수 필드 사이의 차이의 절대값이나 제곱값의 합이 계산된다.

도 11에 도시된 바와 같이 서로 분리된 짝수 및 홀수 필드선 데이터인 경우에는 수직으로 인접한 짝수 필드선의 신호간 차이와 수직으로 인접한 홀수 필드선의 신호간 차이가 계산된다. 이어서, 각 차이의 절대값의 합이나 제곱값이 찾아진다. 마지막으로, 인접한 두 짝수 필드와 인접한 두 홀수 필드 사이의 모든 차이의 절대값이나 제곱값의 합이 계산된다.

도 10에 도시된 데이터에 대해 계산된 합은 DCT 모드를 선택하도록 도 11에 도시된 데이터에 대해 계산된 합과 비교된다. 특정하게, 전자가 후자 보다 더 작은 것으로 발견되면, 프레임-DCT 모드가 선택된다. 한편, 후자가 전자 보다 더 작은 것으로 발견되면, 필드-DCT 모드가 선택된다.

마지막으로, 선택된 DCT 모드에 대응하는 구성을 갖는 데이터는 DCT 회로(56)에 공급되고, 동시에, 선택된 DCT 모드를 나타내는데 사용되는 DCT 플래그는 가변길이 부호화 회로(58) 및 이동 보상 회로(64)에 공급된다.

DCT-모드 스위칭 회로(55)에 의해 결정된 도 10 및 도 11의 DCT 모드로 예측모드 스위칭 회로(52)에 의해 결정된 도 8 및 도 9의 프레임 예측 및 필드 예측의 비교는 프레임 예측 및 필드 예측의 데이터 구조가 휘도 블록에 대해 각각 프레임-DCT 모드 및 필드-DCT 모드의 데이터 구조와 실질적으로 같음을 명확하게 나타낸다.

홀수 및 짝수선이 서로 혼합된 프레임 예측 모드를 예측모드 스위칭 회로(52)가 선택하면, DCT-모드 스위칭 회로(55)도 또한 홀수 및 짝수선이 혼합된 프레임-DCT 예측 모드를 선택할 가능성이 있다. 홀수 및 짝수 필드가 서로 분리된 필드 예측 모드를 예측모드 스위칭 회로(52)가 선택하면, DCT-모드 스위칭 회로(55)도 또한 홀수 및 짝수 필드의 데이터가 서로 분리된 필드-DCT 모드를 선택할 가능성이 있다.

그러나, 선택된 DCT-모드가 항상 선택된 예측 모드에 대응하지는 않음을 주목하여야 한다. 임의의 환경에서, 예측모드 스위칭 회로(52)는 예측 에러의 절대값의 합이 더 작은 프레임 예측 또는 필드 예측 모드를 선택하고, DCT-모드 스위칭 회로(55)는 더 나은 부호화 효율성을 제공하는 DCT 모드를 선택한다.

상술된 바와 같이, I-화상의 데이터는 DCT-모드 스위칭 회로(55)에 의해 DCT 회로(56)에 출력된다. 이어서, DCT 회로(56)는 양자화 회로(57)에 공급되는 DCT 계수로의 변환을 위해 데이터에 DCT 처리를 실행한다. 양자화 회로(57)는 추후 기술될 바와 같이 양자화 회로(57)로 귀환되는 전송 버퍼(59)에 저장된 데이터의 양에 따라 조정되는 양자화 스케일로 양자화 처리를 실행한다. 양자화 처리를 완료한 I-화상의 데이터는 가변길이 부호화 회로(58)에 공급된다.

가변길이 부호화 회로(58)는 양자화 회로(57)에서 공급된 I-화상의 데이터를 수신하고, 양자화 회로(57)에 의해 또한 공급된 양자화 스케일로 호프만 코드 (Huffman code)와 같은 가변길이 코드로 화상 데이터를 변환시킨다. 이어서, 가변길이 코드는 전송 버퍼(59)에 저장된다.

양자화 회로(57)에 의해 공급되는 화상 데이터 및 양자화 스케일에 부가하여, 가변길이 부호화 회로(58)는 또한 예측 결정 회로(54)로부터 예측 모드, 이동벡터 검출 회로(50)로부터 이동 벡터, 예측모드 스위칭 회로(52)로부터 예측 플래그, 또한 DCT-모드 스위칭 회로(55)로부터 DCT 플래그에 관한 정보를 수신한다. 예측 모드에 관한 정보는 화상내 부호화, 전방 예측 부호화, 후방 예측 부호화, 또는 전후방 예측 부호화 중에서 처리 유닛에 의해 어떤 처리형이 실행되는가를 나타낸다. 예측 플래그는 예측모드 스위칭 회로(52)에서 처리 유닛(53)으로 공급된 데이터가 프레임 예측 모드인가 필드 예측 모드인가를 나타낸다. DCT 플래그는 DCT-모드 스위칭 회로(55)에서 DCT 회로(56)로 공급된 데이터가 프레임-DCT 또는 필드-DCT 모드인가를 나타낸다.

전송 버퍼(59)는 입력 데이터를 일시적으로 저장하고, 그에 저장된 데이터량을 다시 양자화 회로(57)에 공급한다. 전송 버퍼(59)에 저장된 데이터량이 허용가능한 범위의 상단 제한치를 넘으면, 양자화 제어 신호는 양자화 회로(57)의 양자화 스케일을 증가시켜 양자화 결과로 구해지는 데이터의 양을 줄인다. 한편, 전송 버퍼(59)에 저장된 데이터량이 허용가능한 범위의 하단 제한치 보다 작아지면, 양자화 제어 신호는 양자화 회로(57)의 양자화 스케일을 감소시켜 양자화 결과로 구해지는 데이터의 양을 상승시킨다. 이 방법으로, 전송 버퍼(59)에서 오버플로우(overflow) 및 언더플로우(underflow)가 생성되는 것을 방지할 수 있다.

전송 버퍼(59)에 저장된 데이터는 소정의 타이밍으로 다시 판독되어, 전송선으로 동작하는 기록 매체(3)상에 데이터를 기록하도록 기록 회로(19)에 공급된다.

양자화 회로(57)에 의해 출력된 양자화 스케일 및 I-화상의 데이터는 또한 양자화 스케일에 대응하는 역양자화 스케일로 데이터에 역양자화를 실행하도록 역양자화 회로(60)에 공급된다. 이어서, 역양자화 회로(60)에 의해 출력되는 데이터는

역 이산 코싸인 변환을 실행하도록 IDCT (Inverse Discrete Cosine Transform) 회로(61)에 공급된다. 마지막으로, IDCT 회로(61)에 의해 출력되는 데이터는 프레임 메모리 유닛(63)의 전방 예측 화상 영역(63a)에 저장되도록 처리기(62)를 통해 프레임 메모리 유닛(63)에 공급된다.

동화상 검출 회로(50)에 공급되어 처리되는 GOP는 화상의 시퀀스 I-, B-, P-, B-, P-, B- 등을 구비한다. 이 경우에는 상술된 바와 같이 제 1 프레임의 데이터를 I-화상으로 처리한 이후, 제 2 프레임의 데이터를 B-화상으로 처리하기 전에 제 3 프레임의 데이터가 P-화상으로 처리된다. 이는 이어지는 P-화상을 포함하는 후방 예측이 B-화상에 행해지므로, 이어지는 P-화상이 미리 처리되지 않으면 후방 예측이 실행될 수 없기 때문이다. I-화상의 데이터는 상술된 바와 같이 GOP에 대해 예측모드 스위칭 회로(52)에 의해 설정된 프레임 예측 또는 필드 예측 모드를 위한 포맷으로 예측모드 스위칭 회로(52)에서 처리 유닛(53)으로 전달되어, 항상 프레임내 예측 모드로 처리 유닛(53)에 의해 처리됨을 주목하여야 한다.

상술된 이유로, 제 1 프레임의 처리 데이터를 I-화상으로 처리한 이후에, 동화상 검출 회로(50)는 후방 소스 화상 영역(51c)에 저장되었던 P-화상의 처리를 시작한다. 이어서, 예측모드 스위칭 회로(52)는 각 예측 모드에 대해 한 단위로 취해지는 매크로블록으로 이동벡터 검출 회로(50)에 의해 공급되는 P-화상의 데이터에 대한 프레임 또는 예측 에러간 차이의 절대값의 합을 계산하고, 그 합을 상술된 바와 같이 예측 결정 회로(54)에 공급한다. P-화상의 데이터 자체는 GOP 중 제 1 프레임의 I-화상이 입력되었을 때 P-화상의 GOP에 대해 예측모드 스위칭 회로(52)에 의해 설정된 프레임-예측 또는 필드-예측 모드의 포맷으로 처리 유닛(53)에 전달된다. 한편, 예측 결정 회로(54)는 P-화상의 데이터가 처리 유닛(53)에 의해 처리되어야 하는 예측 모드를 결정한다. 즉, 각 예측 모드에 대해 예측에러 스위칭 회로(52)에 의해 계산된 예측 에러의 절대값의 합을 근거로 P-화상의 데이터에 처리 유닛(53)으로 실행되는 처리형으로서 화상내, 전방, 후방, 또는 전후방 예측을 선택한다. 엄밀히 말하면, P-화상의 경우, 처리형은 상술된 바와 같이 화상내 또는 전방 예측 모드가 될 수 있다.

먼저, 화상내 예측 모드에서, 처리 유닛(53)은 스위치(53d)를 접촉점(a)에 설정한다. 그래서, P-화상의 데이터는 I-화상의 경우와 같이 DCT-모드 스위칭 회로(55), DCT 회로(56), 양자화 회로(57), 가변길이 부호화 회로(58), 및 전송 버퍼(59)를 통해 전송선으로 전달된다. P-화상의 데이터는 또한 양자화 회로(57), 역양자화 회로(60), IDCT 회로(61), 및 처리기(62)를 통해 후방 예측 화상 영역(63b)에 저장되도록 프레임 메모리 유닛(63)에 공급된다.

두 번째로, 전방 예측 모드에서, 처리 유닛(53)은 스위치(53d)를 접촉점(b)에 설정하고, 이동 보상 회로(64)는 프레임 메모리 유닛(63)의 전방 예측 화상 영역(63a)으로부터 데이터를 판독하여, 이동벡터 검출 회로(50)에 의해 이동 보상 회로(64)에 공급된 이동 벡터에 따라 데이터에 이동 보상을 실행한다. 이 경우에서는 전방 예측 화상 영역(63a)에 저장된 데이터가 I-화상의 데이터이다. 말하자면, 예측 결정 회로(54)에 의해 전방 예측 모드가 알려지면, 이동 보상 회로(64)는 전방 예측 화상 영역(63a)내의 판독 어드레스로부터 I-화상의 데이터를 판독함으로써 전방 예측 화상 데이터의 데이터를 생성한다. 판독 어드레스는 이동 벡터에 대응하는 거리 만큼 이동벡터 검출 회로(50)에 의해 현재 출력되는 매크로블록의 위치로부터 시프트된 위치이다.

이동 보상 회로(64)에 의해 판독된 전방 예측 화상의 데이터는 기준 화상의 데이터, 즉 P-화상과 연관되고, 처리 유닛(53)에서 사용되는 처리기(53a)에 공급된다. 처리기(53a)는 이동 보상 회로(64)에 의해 공급된 전방 예측 화상의 데이터, 즉 I-화상을 예측모드 스위칭 회로(52)에 의해 공급된 기준 화상의 매크로블록의 데이터로부터 감산하여, 예측에서의 에러 또는 차이를 찾는다. 차이 데이터는 DCT-모드 스위칭 회로(55), DCT 회로(56), 양자화 회로(57), 가변길이 부호화 회로(58), 및 전송 버퍼(59)를 통해 전송선에 전달된다. 차이 데이터는 또한 역양자화 회로(60) 및 IDCT 회로(61)에 의해 국부적으로 복호화되고, 국부적인 복호화로부터 구해진 결과는 처리기(62)에 공급된다.

이동 보상 회로(64)에 의해 처리기(53a)로 공급된 전방 예측 화상의 데이터는 또한 처리기(62)에도 공급된다. 처리기(62)에서는 전방 예측 화상의 데이터가 원래 P-화상의 데이터를 만들기 위해 IDCT 회로(61)에 의해 출력된 차이 데이터에 가산된다. 이어서, 원래 P-화상의 데이터는 프레임 메모리 유닛(63)의 후방 예측 화상 영역(63b)에 저장된다.

I-화상 및 P-화상의 데이터 부분이 상술된 바와 같이 각각 프레임 메모리 유닛(63)의 전방 예측 화상 영역(63a)과 후방 예측 화상 영역(63b)에 저장된 이후에는 B-화상의 제 2 프레임의 처리가 이동벡터 검출 회로(50)에 의해 시작된다. B-화상의 경우에는 예측 결정 회로(54)에 의해 결정된 처리형이 화상내 예측 모드나 전방 예측 모드에 부가하여 후방 예측 모드나 전후방 예측 모드가 될 수 있다는 점을 제외하면, B-화상은 상술된 P-화상의 경우와 같은 방법으로 예측 모드 스위칭 회로(52)에 의해 처리된다.

화상내 예측 모드나 전방 예측 모드의 경우에는 스위치(53d)가 P-화상의 경우에 대해 상술된 바와 같이 각각 접촉점 (a) 또는 (b)에 설정된다. 이 경우에, 예측모드 스위칭 회로(52)에 의해 출력되는 B-화상의 데이터는 상술된 P-화상과 같은 방법으로 처리되어 전송된다.

한편, 후방 예측 모드나 전후방 예측 모드에서는 스위치(53d)가 각각 접촉점 (c) 또는 (d)에 설정된다.

스위치(53d)가 접촉점(c)에 설정되는 후방 예측 모드에서는 이동 보상 회로(64)가 프레임 메모리 유닛(63)의 후방 예측 화상 영역(63b)으로부터 데이터를 판독하고, 이동벡터 검출 회로(50)에 의해 이동 보상 회로(64)로 공급된 이동 벡터에 따라 데이터에 이동 보상을 실행한다. 이 경우에는 후방 예측 화상 영역(63b)에 저장된 데이터가 P-화상의 데이터이다. 말하자면, 예측 결정 회로(54)에 의해 후방 예측 모드가 알려지면, 이동 보상 회로(64)는 후방 예측 화상 영역(63b)내의 판독 어드레스로부터 P-화상의 데이터를 판독함으로써 후방 예측 화상의 데이터를 생성한다. 판독 어드레스는 이동 벡터에 대응하는 거리 만큼 이동벡터 검출 회로(50)에 의해 현재 출력되는 매크로블록의 위치로부터 시프트된 위치이다.

이동 보상 회로(64)에 의해 판독된 후방 예측 화상의 데이터는 기준 화상, 즉 B-화상의 데이터와 연관되고, 처리 유닛(53)에서 사용되는 처리기(53b)에 공급된다. 처리기(53b)는 이동 보상 회로(64)에 의해 공급된 후방 예측 화상, 즉 P-화상의 데이터를 예측모드 스위칭 회로(52)에 의해 공급된 기준 화상의 매크로블록의 데이터로부터 감산하여, 예측에서의 에러 또는 차이를 찾는다. 차이 데이터는 DCT-모드 스위칭 회로(55), DCT 회로(56), 양자화 회로(57), 가변길이 부호화 회로(58), 및 전송 버퍼(59)를 통해 전송선에 전달된다.

한편, 스위치(53d)가 접촉점(d)에 설정된 전후방 예측 모드에서는 이동 보상 회로(64)가 프레임 메모리 유닛(63)의 전방 예측 화상 영역(63a)으로부터, 이 경우에는, I-화상의 데이터를 판독하고 후방 예측 화상 영역(63b)으로부터 P-화상의 데이터를 판독하여, 이동벡터 검출 회로(50)에 의해 이동 보상 회로(64)에 공급된 이동 벡터에 따라 데이터에 이동 보상을 실행한다.

말하자면, 예측 결정 회로(54)에 의해 전후방 예측 모드가 알려지면, 이동 보상 회로(64)는 전방 예측 화상 영역(63a) 및 후방 예측 화상 영역(63b)내의 판독 어드레스로부터 각각 I- 및 P-화상의 데이터를 판독함으로써 전후방 예측 화상의 데이터를 생성한다. 판독 어드레스는 이동 벡터에 대응하는 거리 만큼 이동벡터 검출 회로(50)에 의해 현재 출력되는 매크로블록의 위치로부터 시프트된 위치이다. 이 경우에는 2개의 이동 벡터, 즉 전방 및 후방 예측 화상에 대한 이동 벡터가 있다.

이동 보상 회로(64)에 의해 판독된 전후방 예측 화상의 데이터는 기준 화상의 데이터와 연관되고, 처리 유닛(53)에서 사용되는 처리기(53c)에 공급된다. 이동 보상 회로(64)에 의해 공급된 예측 화상의 데이터를 이동벡터 검출 회로(50)에서 사용되는 예측모드 스위칭 회로(52)에 의해 공급된 기준 화상의 매크로블록의 데이터로부터 감산하여, 예측에서의 에러 또는 차이를 찾는다. 차이 데이터는 DCT-모드 스위칭 회로(55), DCT 회로(56), 양자화 회로(57), 가변길이 부호화 회로(58), 및 전송 버퍼(59)를 통해 전송선에 전달된다.

또 다른 프레임에 대한 예측 화상으로 사용된 적이 없는 B-화상은 프레임 메모리 유닛(63)에 저장되지 않는다.

전형적으로, 프레임 메모리 유닛(63)의 전방 예측 화상 영역(63a)과 후방 예측 화상 영역(63b)은 서로 교환될 수 있는 메모리 뱅크 (memory bank)로 실행됨을 주목하여야 한다. 그래서, 전방 예측 화상을 판독하는 동작에서, 프레임 메모리 유닛(63)은 전방 예측 화상 영역(63a)에 설정된다. 한편, 후방 예측 화상을 판독하는 동작에서는 프레임 메모리 유닛(63)이 후방 예측 화상 영역(63b)에 설정된다.

상기 설명은 휘도 블록에 초점이 맞추어졌지만, 색차 블록도 또한 휘도 블록과 같은 방법으로 도 8 내지 도 11에 도시된 매크로블록 유닛으로 처리되고 전달된다. 색차 블록 처리에서의 이동 벡터로는 수직 및 수평 방향에서 색차 블록과 연관되는 휘도 블록의 이동 벡터 성분이 각각 반으로 잘린 크기로 사용됨을 주목하여야 한다.

도 12는 도 5에 도시된 동화상 부호화/복호화 장치에서 사용되는 디코더(31)의 구성을 도시하는 블록도이다. 기록 매체(3)로 실행되는 전송선을 통해 전송된 부호화 화상 데이터는 동화상 부호화/복호화 장치의 재생 회로(30)를 통해 디코더(31)에 의해 수신되고, 디코더(31)에서 사용되는 수신 버퍼(81)에 일시적으로 저장된다. 이어서, 화상 데이터는 디코더(31)의 복호화 회로(90)에서 사용되는 가변길이 복호화 회로(82)에 공급된다. 가변길이 복호화 회로(82)는 수신 버퍼(81)로부터 판독된 화상 데이터에 가변길이 복호화를 실행하여, 이동 벡터, 예측 모드에 대한 정보, 프레임/필드-예측 플래그, 및 프레임/필드 DCT-플래그를 이동 보상 회로(87)에 출력하고, 양자화 스케일 및 복호화 화상 데이터를 역양자화 회로(83)에 출력한다.

역양자화 회로(83)는 가변길이 복호화 회로(82)로부터 또한 수신된 양자화 스케일로 가변길이 복호화 회로(82)에 의해 공급된 화상 데이터에 역양자화를 실행한다. 역양자화 회로(83)는 역양자화 결과로 구해진 DCT 계수 데이터를 IDCT 회로(84)에 출력하여 IDCT (inverse discrete cosine transformation)를 실행하고, IDCT의 결과를 처리기(85)에 공급한다.

I-화상의 경우, IDCT 회로(84)에 의해 처리기(85)에 공급된 화상 데이터는 처리기(85)에 의해 프레임 메모리 유닛(86)으로 그대로 출력되어, 프레임 메모리 유닛(86)의 전방 예측 화상 영역(86a)에 저장된다. 전방 예측 화상 영역(86a)에 저장된 I-화상의 데이터는 전방 예측 모드에서 I-화상 이후에 처리기(85)로 공급되는 P- 또는 B-화상의 화상 데이터에 대한 전방 예측 화상의 데이터를 생성하는데 사용된다. I-화상의 데이터는 또한 도 5에 도시된 동화상 부호화/복호화 장치에서 사용되는 포맷 변환 회로(32)에 출력된다.

IDCT 회로(84)에 의해 공급되는 화상 데이터가 한 프레임 만큼 앞선 화상 데이터를 갖는 P-화상의 데이터일 때, I-화상의 화상 데이터는 프레임 메모리 유닛(86)의 전방 예측 화상 영역(86a)으로부터 이동 보상 회로(87)에 의해 판독된다. 이동 보상 회로(87)에서, I-화상의 화상 데이터에는 가변길이 복호화 회로(82)에 의해 공급된 이동 벡터에 대한 이동 보상이 행해진다. 이동 보상이 완료된 화상 데이터는 처리기(85)로 공급되어, 실제 차이 데이터인 IDCT 회로(84)에 의해 공급되는 화상 데이터에 가산된다. 가산 결과, 즉 복호화된 P-화상의 데이터는 프레임 메모리 유닛(86)에 공급되어, 상술된 바와 같이 프레임 메모리 유닛(86)의 후방 예측 화상 영역(86b)에 저장된다. 후방 예측 화상 영역(86b)에 저장된 P-화상의 데이터는 후방 예측 모드에서 이후에 처리기(85)로 공급되는 B-화상의 화상 데이터에 대한 후방 예측 화상의 데이터를 생성하는데 사용된다.

한편, 프레임내 예측 모드에서 신호 부호화 장치(1)에 의해 처리된 P-화상의 화상 데이터는 아무런 처리도 행해지지 않고 그대로 처리기(85)에 의해 I-화상의 경우와 같이 후방 예측 화상 영역(86b)으로 출력된다.

P-화상은 이때 P-화상 이후에 처리되는 B-화상 다음에 디스플레이되므로, P-화상은 포맷 변환 회로(32)에 출력되지 않는다. 인코더(18)와 같이, 디코더(31)는 B-화상 이후에 P-화상이 수신되더라도 B-화상 이전에 P-화상을 처리하여 전달한다.

IDCT 회로(84)에 의해 출력된 B-화상의 화상 데이터는 가변길이 복호화 회로(82)에 의해 공급된 예측 모드에 관한 정보에 따라 처리기(85)에 의해 처리된다. 특정하게, 처리기(85)는 I-화상의 경우와 같이 화상내 예측 모드에서 그대로 화상 데이터를 출력하거나, 전방 예측, 후방 예측, 또는 전후방 예측 모드로 화상 데이터를 처리한다. 전방 예측, 후방 예측, 또는 전후방 예측 모드에서는 이동 보상 회로(87)가 전방 예측 화상 영역(86a)에 저장된 I-화상의 데이터, 후방 예측 화상 영역(86b)에 저장된 P-화상의 데이터, 또는 프레임 메모리 유닛(86)의 전방 예측 화상 영역(86a) 및 후방 예측 화상 영역(86b)에 각각 저장된 I- 및 P-화상의 데이터를 판독한다. 이어서, 이동 보상 회로(87)는 가변길이 복호화 회로(82)에 의해 출력된 이동 벡터에 따라 프레임 메모리 유닛(86)으로부터 판독된 화상 데이터에 이동 보상을 실행하여 예측 화상을 만든다. 상술된 화상내 예측 모드의 경우에서는 처리기(85)에 의해 예측 화상이 요구되지 않기 때문에 예측 화상이 생성되지 않는다.

이동 보상 회로(87)에서 이동 보상이 실행된 예측 화상은 IDCT 회로(84)에 의해 출력되는 B-화상의 화상 데이터, 엄밀히 말하면 차이 데이터로 처리기(85)에 의해 가산된다. 이어서, 처리기(85)에 의해 출력된 데이터는 I-화상의 경우와 같이 포맷 변환 회로(32)에 공급된다.

그러나, 처리기(85)에 의해 출력된 데이터는 B-화상의 화상 데이터이므로, 예측 화상의 생성에서는 데이터가 요구되지 않는다. 그래서, 처리기(85)에 의해 출력된 데이터는 프레임 메모리 유닛(86)에 저장되지 않는다.

B-화상의 데이터가 출력된 이후에는 P-화상의 데이터가 이동 보상 회로(87)에 의해 후방 예측 화상 영역(86b)으로부터 판독되어 처리기(85)에 공급된다. 그러나, 이때, 데이터는 후방 예측 화상 영역(86b)에 저장되기 이전에 이동 보상되므로, 데이터에는 이동 보상이 행해지지 않는다.

디코더(31)는 카운터부 처리, 즉 도 9 및 도 11에 도시된 바와 같이 서로 분리된 짝수 및 홀수 필드를 갖는 신호 포맷을 각각 도 8 및 도 10에 도시된 바와 같이 서로 혼합된 짝수 및 홀수 필드를 갖는 신호 포맷으로 다시 변환하는 처리가 이동 보상 회로(87)에 의해 실행되기 때문에 도 5에 도시된 인코더(18)에서 사용되는 DCT-모드 스위칭 회로(55) 및 예측모드 스위칭 회로(52)의 카운터부 회로를 포함하지 않는다.

상기 설명은 휘도 신호에 초점이 맞추어지지만, 색차 신호도 또한 휘도 신호와 같은 방법으로 도 8 및 도 11에 도시된 매크로블록 단위로 처리되고 전달된다. 색차 신호 처리에서의 이동 벡터로는 수직 및 수평 방향에서 색차 블록과 연관되는 휘도 신호의 이동 벡터 성분이 각각 반으로 잘린 크기로 사용됨을 주목하여야 한다.

도 13은 SNR (Signal-to-Noise Ratio)에 대한 부호화 화상의 품질을 도시하는 도면이다. 도면에 도시된 바와 같이, 화상 품질은 화상의 종류에 많이 의존한다. 특정하게, 각기 전송된 I- 및 P- 화상은 높은 품질을 갖지만, B-화상은 불량한 품질을 갖는다. 도 13에 도시된 화상 품질에서의 의도적인 변경은 인간의 시각 특성을 사용하는 기술 일종이다. 말하자면, 품질을 변경시킴으로서, 모든 화상 품질의 평균이 사용되는 경우보다 시각에 더 나은 화상 품질이 나타난다. 화상 품질을 변경시키기 위한 제어는 도 7에 도시된 인코더(18)에서 사용되는 양자화 회로(57)에 의해 실행된다.

도 14 및 도 15는 본 발명에 의해 제공되는 트랜스코더(transcoder)(101)의 구성을 도시하는 도면이다. 도 15는 도 14에 도시된 구성을 보다 상세히 도시한다. 트랜스코더(101)는 비디오 복호화 장치(102)에 공급된 부호화 비디오 비트스트림의 비트 레이트 및 GOP 구조를 각각 호스트 컴퓨터에 의해 지정되거나 작동자가 원하는 비트 레이트 및 GOP 구조로 변환시킨다. 트랜스코더(101)의 기능은 실질적으로 각각 트랜스코더(101)와 같은 기능을 갖는 3개의 다른 트랜스코더가 트랜스코더(101)의 전치 단계에 연결된다고 가정함으로써 설명된다. 비트스트림의 비트 레이트 및 GOP 구조를 각각 다양한 비트 레이트 중 하나와 다양한 GOP 구조 중 하나로 변환시키기 위해서는 제 1, 제 2 , 및 제 3 세대의 트랜스코더가 직렬로 연결되고, 도 15에 도시된 제 4 세대의 트랜스코더(101)가 제 1, 제 2, 및 제 3 세대의 트랜스코더의 직렬 연결 뒤에 연결된다. 제 1, 제 2, 및 제 3 세대의 트랜스코더는 도 15에 도시되지 않음을 주목하여야 한다.

본 발명의 다음 설명에서, 제 1 세대의 트랜스코더에 의해 실행되는 부호화 처리는 제 1 세대의 부호화 처리라 칭하여지고, 제 1 세대의 트랜스코더 이후에 연결된 제 2 세대의 트랜스코더에 의해 실행되는 부호화 처리는 제 2 세대의 부호화 처리라 칭하여진다. 유사하게, 제 2 세대의 트랜스코더 이후에 연결된 제 3 세대의 트랜스코더에 의해 실행되는 부호화 처리는 제 3 세대의 부호화 처리라 칭하여지고, 제 3 세대의 트랜스코더 이후에 연결된 제 4 트랜스코더, 즉 도 15에 도시된 트랜스코더(101)에 의해 실행되는 부호화 처리는 제 4 세대의 부호화 처리라 칭하여진다. 부가하여, 제 1 세대의 부호화 처리의 결과로 구해질 뿐만 아니라 사용되는 부호화 파라미터는 제 1 세대의 부호화 파라미터라 칭하여지고, 제 2 세대의 부호화 처리의 결과로 구해질 뿐만 아니라 사용되는 부호화 파라미터는 제 2 세대의 부호화 파라미터라 칭하여진다. 유사하게, 제 3 세대의 부호화 처리의 결과로 구해질 뿐만 아니라 사용되는 부호화 파라미터는 제 3 세대의 부호화 파라미터라 칭하여지고, 제 4 세대의 부호화 처리의 결과로 구해질 뿐만 아니라 사용되는 부호화 파라미터는 제 4 세대의 부호화 파라미터 또는 현재 부호화 파라미터라 칭하여진다.

먼저, 제 3 세대의 트랜스코더에 의해 생성되어 도 15에 도시된 제 4 세대의 트랜스코더(101)에 공급되는 제 3 세대의 부호화 비디오 비트스트림 ST(3rd)이 설명된다. 제 3 세대의 부호화 비디오 비트스트림 ST(3rd)은 제 4 세대의 트랜스코더(101)에 선행하는 단계에서 제공된 제 3 트랜스코더에 의해 실행되는 제 3 세대의 부호화 처리 결과로 구해지는 부호화 비디오 비트스트림이다. 제 3 세대의 부호화 처리 결과로 주어지는 제 3 세대의 부호화 비디오 비트스트림 ST(3rd)에서, 제 3 세대의 부호화 처리에서 생성된 제 3 세대의 부호화 파라미터는 제 3 세대의 부호화 비디오 비트스트림(ST)의 시퀀스층, GOP층, 화상층, 슬라이스층, 및 매크로블록층에 대해 sequence_header() 함수, sequence_extension() 함수, group_of_pictures_header() 함수, picture_header() 함수, picture_coding_extension() 함수, picture_data() 함수, slice() 함수, 및 macroblock() 함수로 각각 기술된다. 제 3 세대의 부호화 처리에서 사용되는 제 3 세대의 부호화 파라미터가 제 3 세대의 부호화 처리 결과로 주어지는 제 3 세대의 부호화 비디오 비트스트림(ST)에 기술된다는 사실은 MPEG2 규격에 따르고, 새로운 사실은 아니다.

본 발명에 의해 제공되는 트랜스코더(101)에서 유일한 점은 제 3 세대의 부호화 파라미터가 제 3 세대의 부호화 비디오 비트스트림(ST)에 기술된다는 사실이 아니라, 제 1 및 제 2 세대의 부호화 처리 결과로 각각 구해지는 제 1 및 제 2 세대의 부호화 파라미터가 제 3 세대의 부호화 비디오 비트스트림(ST)에 포함된다는 사실이다. 제 1 및 제 2 세대의 부호화 파라미터는 제 3 세대의 부호화 비디오 비트스트림(ST) 중 화상층의 user_data 영역에 history_stream()으로 기술된다. 본 발명에서, 제 3 세대의 부호화 비디오 비트스트림(ST) 중 화상층의 user_data 영역에 기술된 히스토리 스트림(history stream)은 "히스토리 정보(history information)"라 칭하여지고, 히스토리 스트림으로 기술된 파라미터는 "히스토리 파라미터(history parameter)"라 칭하여진다. 파라미터를 칭하는 다른 방법으로, 제 3 세대의 부호화 비디오 스트림(ST)에 기술된 제 3 세대의 부호화 파라미터는 또한 현재 부호화 파라미터라 칭하여질 수 있다. 이 경우에, 제 1 및 제 2 세대의 부호화 처리는 각각 제 3 세대의 부호화 처리에서 관찰할 때 과거에 실행된 처리이므로, 제 3 세대의 부호화 비디오 비트스트림(ST) 중 화상층의 user_data 영역에 history_stream()으로 기술된 제 1 및 제 2 세대의 부호화 파라미터는 "과거 부호화 파라미터 (past encoding parameter)"라 칭하여진다.

제 1 및 제 2 세대의 부호화 처리 결과로 각각 구해지는 제 1 및 제 2 세대의 부호화 파라미터가 상술된 바와 같이 제 3 세대의 부호화 파라미터에 부가하여 제 3 세대의 부호화 비디오 비트스트림 SR(3rd)에 또한 기술되는 이유는 부호화 스트림의 비트 레이트 및 GOP 구조가 트랜스코딩 처리에서 반복적으로 변하더라도 화상 품질의 열화를 방지하기 위한 것이다. 예를 들면, 화상은 제 1 세대의 부호화 처리에서 P-화상으로 부호화되고, 제 1 세대의 부호화 비디오 비트스트림의 GOP

구조를 변환시키기 위해, 제 2 세대의 부호화 처리에서는 화상이 B-화상으로 부호화된다. 제 2 세대의 부호화 비디오 비트스트림의 GOP 구조를 더 변환시키기 위해서는 제 3 세대의 부호화 처리에서 화상이 다시 P-화상으로 부호화된다. MPEG 규격을 근거로 하는 종래 부호화 및 복호화 처리는 100% 역처리되지는 않으므로, 일반적으로 공지된 바와 같이, 부호화 및 복호화 처리가 실행될 때마다 화상 품질이 열화된다. 이러한 경우에 있어, 양자화 스케일, 이동 벡터, 및 예측 모드와 같은 부호화 파라미터가 제 3 세대의 부호화 처리에서 단순히 재계산되지 않는다. 그 대신, 제 1 세대의 부호화 처리에서 생성된 양자화 스케일, 이동 벡터, 및 예측 모드와 같은 부호화 파라미터가 재사용된다. 제 1 세대의 부호화 처리에서 새롭게 생성된 양자화 스케일, 이동 벡터, 및 예측 모드와 같은 부호화 파라미터들 각각은 제 3 세대의 부호화 처리에서 새롭게 생성된 대응 부호화 파라미터 보다 더 높은 정확도를 명백히 갖는다. 그래서, 제 1 세대의 부호화 처리에서 생성된 부호화 파라미터를 재사용함으로써, 부호화 및 복호화 처리가 반복적으로 실행되더라도 화상 품질이 열화되는 정도를 낮추는 것이 가능하다.

상술된 본 발명에 따른 처리는 도 15에 도시된 제 4 세대의 트랜스코더(101)에 의해 실행되는 복호화 및 부호화 처리를 상세히 설명함으로써 예시화된다. 비디오 복호화 장치(102)는 제 3 세대의 부호화 파라미터를 사용함으로써 제 3 세대의 부호화 비디오 비트스트림 ST(3rd)에 포함된 부호화 비디오 신호를 복호화하여, 복호화된 기저대 디지털 비디오 데이터를 생성한다. 부가하여, 비디오 복호화 장치(102)는 또한 제 3 세대의 부호화 비디오 스트림 ST(3rd) 중 화상층의 사용자 데이터 영역에서 히스토리 스트림으로 기술된 제 1 및 제 2 세대의 부호화 파라미터를 복호화한다. 비디오 복호화 장치(102)의 동작 및 구성은 다음과 같이 도 16을 참고로 상세히 설명된다.

도 16은 비디오 복호화 장치(102)의 상세한 구성을 도시하는 도면이다. 도면에 도시된 바와 같이, 비디오 복호화 장치(102)는 공급된 부호화 비트스트림을 버퍼 처리하는 버퍼(81), 부호화 비트스트림에 가변길이 복호화 처리를 실행하는 가변길이 복호화 회로(112), 가변길이 복호화 회로(112)에 의해 공급된 양자화 스케일에 따라 가변길이 복호화 처리가 완료된 데이터에 역양자화를 실행하는 역양자화 회로(83), 역양자화가 완료된 DCT 계수에 역이산 코싸인 변환을 실행하는 IDCT 회로(84), 이동 보상 처리를 실행하는 처리기(85), 프레임 메모리 유닛(86), 및 이동 보상 회로(87)를 구비한다.

제 3 세대의 부호화 비디오 비트스트림 ST(3rd)을 복호화하기 위해, 가변길이 복호화 회로(112)는 제 3 세대의 부호화 비디오 비트스트림 ST(3rd)에서 화상층, 슬라이스층, 및 매크로블록층에 기술된 제 3 세대의 부호화 파라미터를 추출한다. 전형적으로, 가변길이 복호화 회로(112)에 의해 추출된 제 3 세대의 부호화 파라미터는 화상의 종류를 나타내는 picture_coding_type, 양자화 스케일 단계의 크기를 나타내는 quantiser_scale_code, 예측 모드를 나타내는 macrobloc_type, 이동 벡터를 나타내는 motion_vector, 프레임 예측 모드나 필드 예측 모드를 나타내는 frame/field_motion_type, 및 프레임-DCT 모드나 필드-DCT 모드를 나타내는 dct_type을 포함한다. quantiser_scale_code 부호화 파라미터는 역양자화 회로(83)에 공급된다. 한편, picture_coding_type, quantiser_scale_code, macroblock_type, motion_vector, frame/field_motion_type, 및 doc_type과 같은 나머지 부호화 파라미터는 이동 보상 회로(87)에 공급된다.

가변길이 복호화 회로(112)는 제 3 세대의 부호화 비디오 비트스트림 ST(3rd)을 복호화하는데 요구되는 상술된 부호화 파라미터 뿐만 아니라, 제 3 세대의 부호화 비디오 비트스트림 ST(3rd) 중 시퀀스층, GOP층, 화상층, 슬라이스층, 및 매크로블록층으로부터 제 3 세대의 히스토리 정보로 도 15에 도시된 트랜스코더(101) 뒤에 연결된 제 5 세대의 트랜스코더에 전송되는 제 3 세대의 다른 모든 부호화 파라미터도 추출한다. 상술된 바와 같이 제 3 세대의 처리에서 사용되는 picture_coding_type, quantiser_scale_code, macrobloc_type, motion_vector, frame/field_motion_type, 및 dot_type과 같은 제 3 세대의 상기 부호화 파라미터도 또한 제 3 세대의 히스토리 정보에 포함됨은 말할 필요도 없다. 작동자나 호스트 컴퓨터는 전송 용량에 따라 어느 부호화 파라미터가 히스토리 정보로 추출되어야 하는가를 미리 결정한다.

부가하여, 가변길이 복호화 회로(112)는 또한 제 3 세대의 부호화 비디오 비트스트림 ST(3rd) 중 화상층의 user_data 영역에 기술된 사용자 데이터를 추출하고, 그 사용자 데이터를 히스토리 복호화 장치(104)에 공급한다.

히스토리 복호화 장치(104)는 제 3 세대의 부호화 비디오 비트스트림 ST(3rd) 중 화상층에서 추출된 사용자 데이터로부터 히스토리 정보로 기술된 제 1 및 제 2 세대의 부호화 파라미터를 추출한다. 구체적으로, 가변길이 복호화 회로(112)로부터 수신된 사용자 데이터의 구문 (syntax)을 분석함으로써, 히스토리 복호화 장치(104)는 사용자 데이터에 기록된 유일한 History_Data_Id를 검출하고 이를 converted_history_stream()을 추출하는데 사용할 수 있다. 이어서, 소정의 간격으로 converted_history_stream()에 삽입된 1-비트의 마커 비트 (marker_bit)를 인출함으로써, 히스토리 복호화 장치(104)는 history_stream()을 얻을 수 있다. history_stream()의 구문을 분석함으로써, 히스토리 복호화 장치(104)는 history_stream()에 기록된 제 1 및 제 2 세대의 부호화 파라미터를 추출할 수 있다. 히스토리 복호화 장치(104)의 동작 및 구성은 추후 상세히 설명된다.

결국 제 4 세대의 부호화 처리를 실행하도록 제 1, 제 2, 및 제 3 세대의 부호화 파라미터를 비디오 부호화 장치(106)에 공급하기 위해, 히스토리 정보 다중화 장치(103)는 비디오 복호화 장치(102)에 의해 복호화된 기저대 비디오 데이터에서 제 1, 제 2, 및 제 3 세대의 부호화 파라미터를 다중화한다. 히스토리 정보 다중화 장치(103)는 비디오 복호화 장치(102)로부터 기저대 비디오 데이터를 수신하고, 비디오 복호화 장치(102)에서 사용되는 가변길이 복호화 회로(112)로부터 제 3 세대의 부호화 파라미터를 수신하고, 또한 히스토리 복호화 장치(104)로부터 제 1 및 제 2 세대의 부호화 파라미터를 수신하여, 기저대 비디오 데이터에서 제 1, 제 2, 및 제 3 세대의 부호화 파라미터를 다중화한다. 다중화된 제 1, 제 2, 및 제 3 세대의 부호화 파라미터와 함께 기저대 비디오 데이터는 히스토리 정보 분리 장치(105)로 공급된다.

다음에는 도 17 및 도 18을 참고로 기저대 비디오 데이터에서 제 1, 제 2, 및 제 3 세대의 부호화 파라미터를 다중화하는 기술이 설명된다. 도 17은 휘도 신호부와 색차 신호부로 구성되고 각 부분이 MPEG 규격에 따라 정의된 바와 같이 16 픽셀 16 픽셀을 구비하는 매크로블록을 도시하는 도면이다. 16 픽셀 16 픽셀을 구비하는 부분 중 하나는 휘도 신호에 대해 4개의 서브블록 Y[0], Y[1], Y[2], 및 Y[3]으로 구성되는 반면, 다른 부분은 색차 신호에 대해 4개의 서브블록 Cr[0], Cr[1], Cb[0], 및 Cb[1]으로 구성된다. 서브블록 Y[0], Y[1], Y[2], 및 Y[3]과 서브블록 Cr[0], Cr[1], Cb[0], 및 Cb[1]은 각각 8 픽셀 8 픽셀을 구비한다.

도 18은 비디오 데이터의 포맷을 도시하는 도면이다. ITU에 의해 추천된 RDT601에 따라 정의되면, 포맷은 방송 산업에서 사용되는 소위 D1 포맷을 나타낸다. D1 포맷은 비디오 데이터를 전송하기 위한 포맷으로 표준화되므로, 1 픽셀의 비디오 데이터는 10 비트로 나타낸다.

MPEG 규격에 따라 복호화된 기저대 비디오 데이터는 길이가 8 비트이다. 본 발명에 의해 제공되는 트랜스코더에서, MPEG 규격에 따라 복호화된 기저대 비디오 데이터는 도 18에 도시된 바와 같이 10-비트의 D1 포맷 중 8개의 상위비트 (D9) 내지 (D2)를 사용해 전송된다. 그러므로, 8-비트의 복호화 비디오 데이터는 D1 포맷으로 할당되지 않은 2개의 하위 비트 (D1) 및 (D2)를 남긴다. 본 발명에 의해 제공되는 트랜스코더는 히스토리 정보를 전송하기 위해 이 비할당 비트를 구비하는 비할당 영역을 사용한다.

도 18에 도시된 데이터 블록은 매크로블록의 8개 서브블록에서 픽셀을 전송하기 위한 데이터 블록이다. 각 서브블록은 실제로 상술된 바와 같이 64 (= 8 x 8) 픽셀을 구비하므로, 도 18에 각각 도시된 64개의 데이터 블록은 8개 서브블록을 구비하는 매크로블록의 데이터 볼륨을 전송하는데 요구된다. 상술된 바와 같이, 휘도 및 색차 신호에 대한 매크로블록은 각각 64 (= 8 x 8) 픽셀로 구성된 8개의 서브블록을 구비한다. 그래서, 휘도 및 색차 신호에 대한 매크로블록은 8 x 64 픽셀 = 512 픽셀을 구비한다. 상술된 바와 같이, 각 픽셀이 할당되지 않은 2 비트를 남기므로, 휘도 및 색차 신호에 대한 매크로블록은 512 픽셀 x 2 비할당 비트/픽셀 = 1,024 비할당 비트를 갖는다. 한편, 한 세대의 히스토리 정보는 256 비트의 길이이다. 그래서, 4 (= 1,024/256)개의 이전 세대에 대한 히스토리 정보는 휘도 및 색차 신호에 대한 비디오 데이터의 매크로블록상에 포개질 수 있다. 도 18에 도시된 예에서는 제 1, 제 2, 및 제 3 세대의 히스토리 정보가 2개의 하위 비트 (D1) 및 (D0)을 사용함으로써 비디오 데이터의 한 매크로블록상에 포개진다.

히스토리 정보 분리 장치(105)는 (D1) 포맷으로 그에 전송된 데이터 중 8개의 상위비트로부터 기저대 비디오 데이터를 추출하고, 2개의 하위비트로부터는 히스토리 정보를 추출한다. 도 15에 도시된 예에서, 히스토리 정보 분리 장치(105)는 전송된 데이터로부터 기저대 비디오 데이터를 추출하고, 그 기저대 비디오 데이터를 비디오 부호화 장치(106)에 공급한다. 동시에, 히스토리 정보 분리 장치(105)는 전송된 데이터로부터 제 1, 제 2, 및 제 3 세대의 부호화 파라미터를 구비하는 히스토리 정보를 추출하고, 그 히스토리 정보를 신호 부호화 회로(106) 및 히스토리 부호화 장치(107)에 공급한다.

비디오 부호화 장치(106)는 히스토리 정보 분리 장치(105)에 의해 공급된 기저대 비디오 데이터를 작동자나 호스트 컴퓨터에 의해 지정된 비트 레이트 및 GOP 구조를 갖는 비트스트림으로 부호화한다. GOP 구조를 변환시키는 것은 GOP에 포함된 화상의 수를 변경시키는 것, 연속적인 2개의 I-화상 사이에 존재하는 P-화상의 수를 변경시키는 것, 또는 연속되는 2개의 I-화상 사이나 I-화상과 P-화상 사이에 존재하는 B-화상의 수를 변경시키는 것을 의미함을 주목하여야 한다.

도 15에 도시된 실시예에서, 공급된 기저대 비디오 데이터는 그에 포개진 제 1, 제 2, 및 제 3 세대의 부호화 파라미터의 히스토리 정보를 포함한다. 그래서, 비디오 부호화 장치(106)는 화상 품질이 열화되는 정도를 낮추도록 히스토리 정보 일부를 선택적으로 재사용함으로써 제 4 세대의 부호화 처리를 실행할 수 있다.

도 19는 비디오 부호화 장치(106)에서 사용되는 인코더(121)의 구성을 구체적으로 도시하는 도면이다. 도면에 도시된 바와 같이, 인코더(121)는 이동벡터 검출 회로(50), 예측모드 스위치 회로(52), 처리기(53), DCT 스위치 회로(55), DCT 회로(56), 양자화 회로(57), 가변길이 부호화 회로(58), 전송 버퍼(59), 역양자화 회로(60), 역DCT 회로(61), 처리기(62), 프

레임 메모리 유닛(63), 및 이동 보상 회로(64)를 구비한다. 이들 회로의 기능은 도 7에 도시된 인코더(18)에서 사용되는 것과 똑같으므로, 반복적인 설명이 불필요하다. 그래서, 다음의 설명은 도 7에 도시된 인코더(18)와 인코더(121) 사이의 차이에 초점을 맞춘다.

인코더(121)는 또한 인코더(121)를 구성하는 상술된 다른 구성성분의 동작과 기능을 제어하기 위한 제어기(70)를 포함한다. 제어기(70)는 작동자나 호스트 컴퓨터로부터 GOP 구조를 지정하는 지시를 수신하고, GOP 구조를 구성하는 화상의 종류를 결정한다. 부가하여, 제어기(70)는 또한 작동자나 호스트 컴퓨터로부터 목표 비트 레이트에 관한 정보를 수신하여, 인코더(121)에 의해 출력된 비트 레이트를 지정된 목표 비트 레이트로 설정하도록 양자화 회로(57)를 제어한다.

더욱이, 제어기(70)는 또한 히스토리 정보 분리 장치(105)에 의해 출력되는 다수의 세대에 대한 히스토리 정보를 수신하고, 이 히스토리 정보를 재사용함으로써 기준 화상을 부호화한다. 제어기(70)의 기능은 다음과 같이 상세히 설명된다.

먼저, 제어기(70)는 작동자나 호스트 컴퓨터에 의해 지정된 GOP 구조로부터 결정되는 기준 화상의 종류가 히스토리 정보에 포함된 화상 타입과 정합되는가 여부를 판단한다. 말하자면, 제어기(70)는 기준 화상이 지정된 화상 타입과 같은 화상 타입으로 과거에 부호화되었는가 여부를 판단한다.

상술된 판단은 도 15에 도시된 예를 사용해 예시화될 수 있다. 제어기(70)는 제 4 세대의 부호화 처리에서 기준 화상에 지정된 화상 타입이 제 1 세대의 부호화 처리에서의 기준 화상의 종류, 제 2 세대의 부호화 처리에서의 기준 화상의 종류, 또는 제 3 세대의 부호화 처리에서의 기준 화상의 종류와 같은가 여부를 판단한다.

제 4 세대의 부호화 처리에서 기준 화상에 지정된 화상 타입이 이전 세대의 부호화 처리에서의 기준 화상의 종류와 같지 않은 것으로 판단 결과가 나타내면, 제어기(70)는 정상적인 부호화 처리를 실행한다. 이 판단 결과는 제 4 세대의 부호화 처리에서 기준 화상에 지정된 화상 타입으로 이 기준 화상에는 이전에 제 1, 제 2, 및 제 3 세대의 부호화 처리가 결코 행해지지 않았음을 의미한다. 한편, 제 4 세대의 부호화 처리에서 기준 화상에 지정된 화상 타입이 이전 세대의 부호화 처리에서의 기준 화상의 종류와 같은 것으로 판단 결과가 나타내면, 제어기(70)는 이전 세대의 파라미터를 재사용함으로써 파라미터 재사용 부호화 처리를 실행한다. 이 판단 결과는 제 4 세대의 부호화 처리에서 기준 화상에 지정된 화상 타입으로 이 기준 화상에는 이전에 제 1, 제 2, 및 제 3 세대의 부호화 처리가 행해졌음을 의미한다.

먼저, 제어기(70)에 의해 실행되는 정상적인 부호화 처리가 설명된다. 프레임 예측 모드나 필드 예측 모드 중 어느 것이 선택되어야 하는 것에 대해 제어기(70)가 결정하도록 허용하기 위해, 이동벡터 검출 회로(50)는 프레임 예측 모드에서의 예측 에러와 필드 예측 모드에서의 예측 에러를 검출하고, 예측 에러값을 제어기(70)에 공급한다. 제어기(70)는 그 값들을 서로 비교하여, 더 작은 예측 에러를 갖는 예측 모드를 선택한다. 이어서, 예측 모드 스위치 회로(52)는 제어기(70)에 의해 선택된 예측 모드에 대응하는 신호 처리를 실행하고, 처리 결과로 구해지는 신호를 처리 유닛(53)에 공급한다. 프레임 예측 모드가 선택되면, 예측 모드 스위치 회로(52)는 신호를 수신할 때와 같이 휘도 신호를 처리 유닛(53)에 공급하도록 하는 신호 처리를 실행하고, 도 8을 참고로 상술된 바와 같이 홀수 필드선을 짝수 필드선과 혼합하도록 하는 색차 신호의 신호 처리를 실행한다. 한편, 필드 예측 모드가 선택되면, 예측 모드 스위치 회로(52)는 휘도 서브블록 Y[1] 및 Y[2]가 홀수 필드선을 구비하고 휘도 서브블록 Y[3] 및 Y[4]가 짝수 필드선을 구비하도록 하는 휘도 신호의 신호 처리를 실행하고, 도 9를 참고로 상술된 바와 같이 4개의 상단선이 홀수 필드선을 구비하고 4개의 하단선이 짝수 필드선을 구비하게 하는 색차 신호의 신호 처리를 실행한다.

부가하여, 제어기(70)가 화상내 예측 모드, 전방 예측 모드, 후방 예측 모드, 또는 전후방 예측 모드 중 어느 것이 선택되어야 하는가를 결정하도록 허용하기 위해, 이동벡터 검출 회로(50)는 각 예측 모드에 대해 예측 에러를 생성하고, 그 예측 에러를 제어기(70)에 공급한다. 제어기(70)는 전방 예측 모드, 후방 예측 모드, 또는 전후방 예측 모드로부터 화상내 예측 모드와 같이 가장 작은 예측 에러를 갖는 모드를 선택한다. 이어서, 제어기(70)는 선택된 화상간 예측 모드의 가장 작은 예측 에러를 화상내 예측 모드의 예측 에러와 비교하고, 더 작은 예측 에러를 갖는 선택된 화상간 예측 또는 화상내 예측 모드를 예측 모드로 선택한다. 상세하게, 화상내 예측 모드의 예측 에러가 더 작으면, 화상내 예측 모드가 설정된다. 한편, 화상간 예측 모드의 예측 에러가 더 작으면, 가장 작은 예측 에러를 갖는 선택된 전방 예측 모드, 후방 예측 모드 또는 전후방 예측 모드가 설정된다. 이어서, 제어기(70)는 설정된 예측 모드로 동작되도록 처리기(53)와 이동 보상 회로(64)를 제어한다.

더욱이, 제어기(70)가 프레임-DCT 모드 또는 필드-DCT 모드 중 어느 것이 선택되어야 하는가를 결정하도록 허용하기 위해, DCT 모드 스위치 회로(55)는 4개의 휘도 서브블록의 데이터를 혼합된 홀수 및 짝수 필드선을 구비하는 프레임-DCT 모드의 포맷을 갖는 신호와, 분리된 홀수 및 짝수 필드선을 구비하는 필드-DCT 모드의 포맷을 갖는 신호로 변환하고, 변환 결과로 주어지는 신호를 DCT 회로(56)에 공급한다. DCT 회로(56)는 혼합된 홀수 및 짝수 필드선을 구비하는 신호의

DCT 처리의 부호화 효율성과 분리된 홀수 및 짝수 필드선을 구비하는 신호의 DCT 처리의 부호화 효율성을 계산하고, 계산된 부호화 효율성을 제어기(70)에 공급한다. 제어기(70)는 부호화 효율성을 서로 비교하여, 더 높은 효율성을 갖는 DCT 모드를 선택한다. 제어기(70)는 이어서 선택된 DCT 모드로 작업하도록 DCT 모드 스위치 회로(55)를 제어한다.

제어기(70)는 또한 작동자나 호스트 컴퓨터에 의해 지정된 원하는 비트 레이트를 나타내는 목표 비트 레이트와 전송 버퍼(59)에 저장된 데이터의 볼륨이나 버퍼(59)에 남겨진 잔류 자유 영역의 크기를 나타내는 신호를 수신하고, 목표 비트 레이트와 버퍼(59)에 남겨진 잔류 자유 영역의 크기를 근거로 양자화 회로(57)에 의해 사용되는 양자화 단계의 크기를 제어하기 위한 feedback_q_scale_code를 생성한다. feedback_q_scale_code는 버퍼(59)에서 오버플로우나 언더플로우가 일어나는 것을 방지하고 목표 비트 레이트로 전송 버퍼(59)로부터 비트스트림이 출력되게 하도록 전송 버퍼(59)에 남겨진 잔류 자유 영역의 크기에 따라 생성되는 제어 신호이다. 구체적으로, 예를 들어 전송 버퍼(59)에서 버퍼처리된 데이터의 볼륨이 작아지면, 양자화 단계의 크기는 다음에 부호화될 화상의 비트수가 증가되도록 감소된다. 한편, 전송 버퍼(59)에서 버퍼처리된 데이터의 볼륨이 커지면, 양자화 단계의 크기는 다음에 부호화될 화상의 비트수가 감소되도록 증가된다. 양자화 단계의 크기는 feedback_q_scale_code에 비례함을 주목하여야 한다. 말하자면, feedback_q_scale_code가 증가될 때, 양자화 단계의 크기는 상승된다. 한편, feedback_q_scale_code가 감소될 때, 양자화 단계의 크기는 감소된다.

다음에는 트랜스코더(101)를 특징지우는 부호화 파라미터를 재사용하는 파라미터 재사용 부호화 처리가 설명된다. 이해하기 쉽게 설명하기 위해, 기준 화상은 제 1 세대의 부호화 처리에서 I-화상으로, 제 2 세대의 부호화 처리에서 P-화상으로, 또한 제 3 세대의 부호화 처리에서 B-화상으로 부호화되었고, 제 4 세대의 현재 부호화 처리에서는 다시 I-화상으로 부호화되어야 한다고 가정한다. 이 경우에는 기준 화상이 제 4 세대의 부호화 처리로 지정된 I-화상과 같이 요구되는 화상 타입으로 제 1 세대의 부호화 처리에서 앞서 부호화되었으므로, 제어기(70)는 공급된 비디오 데이터로부터 새로운 부호화 파라미터를 생성하는 대신에 제 1 세대의 부호화 파라미터를 사용하여 부호화 처리를 실행한다. 제 4 세대의 부호화 처리에서 재사용되는 이러한 부호화 파라미터 중 대표적인 것은 양자화 스케일 단계의 크기를 나타내는 quantiser_scale_code, 예측 모드를 나타내는 macrobloc_type, 이동 벡터를 나타내는 motion_vector, 프레임 예측 모드나 필드 예측 모드를 나타내는 frame/field_motion_type, 및 프레임-DCT 모드나 필드_DCT 모드를 나타내는 dct_type을 포함한다. 제어기(70)는 정보 히스토리로서 수신된 모든 부호화 파라미터를 재사용하지는 않는다. 대신에, 제어기(70)는 재사용하기에 적절하다고 판단된 부호화 파라미터만을 재사용하고, 이전 부호화 파라미터가 재사용하기에 적합하지 않은 부호화 파라미터를 새롭게 생성한다.

다음에는 상술된 정상적인 부호화 처리와 다른점에 초점을 맞추어 부호화 파라미터를 재사용하는 파라미터 재사용 부호화 처리가 설명된다. 정상적인 부호화 처리에서는 이동벡터 검출 회로(50)가 기준 화상의 이동 벡터를 검출한다. 한편, 부호화 파라미터를 재사용하는 파라미터 재사용 부호화 처리에서는 이동벡터 검출 회로(50)가 기준 화상의 이동 벡터를 검출하지 않는다. 대신에, 이동벡터 검출 회로(50)는 제 1 세대의 히스토리 정보로 전달되는 motion_vector를 재사용한다. 제 1 세대의 motion_vector가 사용되는 이유는 다음과 같이 설명된다. 제 3 세대의 부호화 비트스트림을 복호화한 처리의 결과로 구해지는 기저대 비디오 데이터에는 적어도 3번의 부호화 처리가 행해졌으므로, 원래 비디오 데이터와 비교해 화상 품질이 확실히 양호하지 못하다. 양호하지 못한 화상 품질을 갖는 비디오 데이터로부터 검출된 이동 벡터는 정확하지 않다. 특정하게, 제 1 세대의 히스토리 정보로 제 4 세대의 트랜스코더(101)에 공급된 이동 벡터는 확실히 제 4 세대의 부호화 처리에서 검출된 이동 벡터 보다 더 나은 정확도를 갖는다. 제 1 세대의 부호화 파라미터로 수신된 이동 벡터를 재사용함으로써, 화상 품질은 제 4 세대의 부호화 처리 동안 열화되지 않는다. 제어기(70)는 제 1 세대의 히스토리 정보로 수신된 이동 벡터를 제 4 세대의 부호화 처리에서 부호화되고 있는 기준 화상의 이동 벡터로 사용되도록 이동 보상 회로(64) 및 가변길이 부호화 회로(58)에 공급한다.

정상적인 처리에서, 이동벡터 검출 회로(50)는 프레임 예측 모드나 필드 예측 모드를 선택하기 위해 프레임 예측 모드에서의 예측 에러와 필드 예측 모드에서의 예측 에러를 검출한다. 한편, 부호화 파라미터의 재사용을 근거로 하는 파라미터 재사용 부호화 처리에서는 이동벡터 검출 회로(50)가 프레임 예측 모드에서의 예측 에러나 필드 예측 모드에서의 예측 에러를 검출하지 않는다. 대신에, 프레임 예측 모드나 필드 예측 모드를 나타내는 제 1 세대의 히스토리 정보로 수신된 frame/field_motion_type이 재사용된다. 이는 제 1 세대의 부호화 처리에서 검출된 각 예측 모드의 예측 에러가 제 4 세대의 부호화 처리에서 검출된 각 예측 모드의 예측 에러 보다 더 높은 정확도를 갖기 때문이다. 그래서, 각기 더 높은 정확도를 갖는 예측 에러를 근거로 선택된 예측 모드는 보다 최적의 부호화 처리가 실행될 수 있도록 허용하게 된다. 구체적으로, 제어기(70)는 제 1 세대의 히스토리 정보로서 수신된 frame/field_motion_type을 나타내는 제어 신호를 예측 모드 스위치 회로(52)에 공급한다. 제어 신호는 재사용되는 frame/field_motion_type에 따라 신호 처리를 실행하도록 예측 모드 스위치 회로(52)를 구동시킨다.

정상적인 처리에서, 이동벡터 검출 회로(50)는 또한 예측 모드 중 하나를 선택하기 위해 화상내 예측 모드, 전방 예측 모드, 후방 예측 모드, 및 전후방 예측 모드 각각에서의 예측 에러를 검출한다. 한편, 부호화 파라미터의 재사용을 근거로 하는

처리에서는 이동벡터 검출 회로(50)가 이들 예측 모드의 예측 에러를 검출하지 않는다. 대신에, 제 1 세대의 히스토리 정보로 수신된 macroblock_type으로 나타내지는 화상내 예측 모드, 전방 예측 모드, 후방 예측 모드, 및 전후방 예측 모드 중 하나가 선택된다. 이는 제 1 세대의 처리에서 검출된 각 예측 모드의 예측 에러가 제 4 세대의 처리에서 검출된 각 예측 모드의 예측 에러 보다 더 높은 정확도를 갖기 때문이다. 그래서, 각기 더 높은 정확도를 갖는 예측 에러를 근거로 선택된 예측 모드가 보다 효율적인 부호화 처리가 실행되도록 허용하게 된다. 구체적으로, 제어기(70)는 제 1 세대의 히스토리 정보에 포함된 macroblock_type에 의해 나타내지는 예측 모드를 선택하고, 선택된 예측 모드로 동작되도록 처리기(53) 및 이동 보상 회로(64)를 제어한다.

정상적인 부호화 처리에서, DCT 모드 스위치 회로(55)는 프레임-DCT 모드에서의 부호화 효율성을 필드-DCT 모드에서의 부호화 효율성을 비교하는데 사용되도록 프레임-DCT 모드의 포맷으로 변환된 신호와 필드-DCT 모드의 포맷으로 변환된 신호를 모두 DCT 회로(56)에 공급한다. 한편, 부호화 파라미터의 재사용을 근거로 하는 처리에서는 프레임-DCT 모드의 포맷으로 변환된 신호나 필드-DCT 모드의 포맷으로 변환된 신호가 생성되지 않는다. 대신에, 제 1 세대의 히스토리 정보에 포함된 dct_type에 의해 나타내지는 DCT 모드에서의 처리만이 실행된다. 구체적으로, 제어기(70)는 제 1 세대의 히스토리 정보에 포함된 dct_type을 재사용하여, dct_type에 의해 나타내지는 DCT 모드에 따라 신호 처리를 실행하도록 DCT-모드 스위칭 회로(55)를 제어한다.

정상적인 부호화 처리에서, 제어기(70)는 작동자나 호스트 컴퓨터에 의해 지정된 목표 비트 레이트와 전송 버퍼(59)에 남겨진 잔류 자유 영역의 크기를 근거로 양자화 회로(57)에서 사용되는 양자화 단계의 크기를 제어한다. 한편, 부호화 파라미터의 재사용을 근거로 하는 처리에서는 제어기(70)가 작동자나 호스트 컴퓨터에 의해 지정된 목표 비트 레이트, 전송 버퍼(59)에 남겨진 잔류 자유 영역의 크기, 및 히스토리 정보에 포함된 과거의 양자화 스케일을 근거로 양자화 회로(57)에서 사용되는 양자화 단계의 크기를 제어한다. 다음의 설명에서, 히스토리 정보에 포함된 과거 양자화 스케일은 history_q_scale_code라 칭하여짐을 주목하여야 한다. 추후 설명될 히스토리 스트림에서, 양자화 스케일은 quantiser_scale_code라 칭하여진다.

먼저, 제어기(70)는 정상적인 부호화 처리의 경우와 같이 현재 양자화 스케일을 나타내는 feedback_q_scale_code를 생성한다. feedback_q_scale_code는 오버플로우나 언더플로우가 생성되지 않은 전송 버퍼(59)에 남겨진 잔류 자유 영역의 크기로부터 결정된 값에 설정된다. 이어서, 제 1 세대의 히스토리 스트림에 포함된 이전 양자화 스케일을 나타내는 history_q_scale_code는 어느 양자화 스케일이 더 큰가를 결정하도록 현재 양자화 스케일을 나타내는 feedback_q_scale_code와 비교된다. 큰 양자화 스케일은 큰 양자화 단계를 의미한다. 현재 양자화 스케일을 나타내는 feedback_q_scale_code가 다수의 이전 양자화 스케일 중 가장 큰 것을 나타내는 history_q_scale_code 보다 더 크면, 제어기(70)는 현재 양자화 스케일을 나타내는 feedback_q_scale_code를 양자화 회로(57)에 공급한다. 한편, 가장 큰 이전 양자화 스케일을 나타내는 history_q_scale_code가 현재 양자화 스케일을 나타내는 feedback_q_scale_code 보다 더 크면, 제어기(70)는 가장 큰 이전 양자화 스케일을 나타내는 history_q_scale_code를 양자화 회로(57)에 공급한다. 제어기(70)는 전송 버퍼(59)에 남겨진 잔류 자유 영역의 크기로부터 유도되는 현재 양자화 스케일과 히스토리 정보에 포함된 다수의 이전 양자화 스케일 중 가장 큰 것을 선택한다. 다른 말로 하면, 제어기(70)는 이전 부호화 처리 (또는 제 1, 제 2, 및 제 3 세대의 부호화 처리)와 현재 부호화 처리 (또는 제 4 세대의 부호화 처리)에서 사용되는 양자화 단계 중 가장 큰 양자화 단계를 사용해 양자화를 실행하도록 양자화 회로(57)를 제어한다. 그 이유는 다음과 같이 설명된다.

제 3 세대의 부호화 처리에서 생성된 스트림의 비트 레이트는 4 Mbps이고, 제 4 세대의 부호화 처리를 실행하는 인코더(121)에 대해 설정된 목표 비트 레이트는 15 Mbps라 가정한다. 이전 비트 레이트 보다 높은 이러한 목표 비트 레이트는 실제로 양자화 단계를 단순히 감소시켜서는 이루어질 수 없다. 이는 큰 양자화 단계로 실행된 이전 부호화 처리를 완료한 화상에 작은 양자화 단계로 실행되는 현재 부호화 처리가 결코 화상 품질을 개선시키지 못하기 때문이다. 말하자면, 큰 양자화 단계로 실행된 이전 부호화 처리를 완료한 화상에 작은 양자화 단계로 실행되는 현재 부호화 처리는 단순히 결과 비트의 수를 증가시키고, 화상 품질의 개선시키는데 도움이 되지는 못한다. 그래서, 이전 부호화 처리 (또는 제 1, 제 2, 및 제 3 세대의 부호화 처리)와 현재 부호화 처리 (또는 제 4 세대의 부호화 처리)에서 사용되는 양자화 단계 중 가장 큰 양자화 단계를 사용함으로써, 가장 효율적인 부호화 처리가 실행될 수 있다.

다음에는 도 15를 참고로 히스토리 복호화 장치(104)와 히스토리 부호화 장치(107)가 설명된다. 도면에 도시된 바와 같이, 히스토리 복호화 장치(104)는 사용자 데이터 디코더(201), 변환기(202), 및 히스토리 디코더(203)를 구비한다. 사용자 데이터 디코더(201)는 복호화 장치(102)에 의해 공급된 사용자 데이터를 복호화한다. 변환기(202)는 사용자 데이터 디코더(201)에 의해 출력된 데이터를 변환시키고, 히스토리 디코더(203)는 변환기(202)에 의해 출력된 데이터로부터 히스토리 정보를 재생한다.

한편, 히스토리 부호화 장치(107)는 히스토리 포맷터(formatter)(211), 변환기(212), 및 사용자 데이터 포맷터(213)를 구비한다. 히스토리 포맷터(211)는 히스토리 정보 분리 장치(105)에 의해 공급된 제 3 세대의 부호화 파라미터를 포맷처리한다. 변환기(212)는 히스토리 포맷터(211)에 의해 출력된 데이터를 변환시키고, 사용자 데이터 포맷터(213)는 변환기(212)에 의해 출력된 데이터를 사용자 데이터의 포맷으로 포맷처리한다.

사용자 데이터 디코더(201)는 복호화 장치(102)에 의해 공급된 사용자 데이터를 복호화하고, 복호화 결과를 변환기(202)에 공급한다. 사용자 데이터의 상세한 내용은 추후 기술된다. 임의의 레이트로, user_data()에 의해 나타내지는 사용자 데이터는 user_data_start_code 및 user_data를 구비한다. MPEG 규격에 따라, 사용자 데이터에서 23개의 "0"의 연속 비트 생성은 start_code가 부정확하게 검출되는 것을 방지하기 위해 금지된다. 히스토리 정보는 23개 이상의 "0"의 연속 비트를 포함할 수 있으므로, 도 38을 참고로 추후 기술될 바와 같이, 히스토리 정보를 converted_history_stream()으로 처리 및 변환할 필요가 있다. "1" 비트를 삽입하여 이 변환을 실행하는 구성성분은 히스토리 부호화 장치(107)에서 사용되는 변환기(212)이다. 한편, 히스토리 부호화 장치(104)에서 사용되는 변환기(202)는 히스토리 부호화 장치(107)에서 사용되는 변환기(212)로 실행된 변환에 반대되는 비트 삭제 변환을 실행한다.

히스토리 디코더(203)는 변환기(202)에 의해 출력된 데이터로부터 히스토리 정보를 생성하고, 그 정보를 히스토리 정보 다중화 장치(103)로 출력한다.

한편, 히스토리 부호화 장치(107)에서 사용되는 히스토리 포맷터(211)는 히스토리 정보 분리 장치(105)에 의해 공급된 제 3 세대의 부호화 파라미터 포맷을 히스토리 정보의 포맷으로 변환시킨다. 히스토리 정보의 포맷은 추후 기술될 도 40 내지 도 46에 도시된 고정 길이를 갖거나, 추후 또한 기술될 도 47에 도시된 가변길이를 가질 수 있다.

히스토리 포맷터(211)에 의해 포맷처리된 히스토리 정보는 변환기(212)에 의해 converted_history_stream()으로 변환되어, 상술된 바와 같이 user_data()의 start_code가 부정확하게 검출되는 것을 방지한다. 말하자면, 히스토리 정보는 23개 이상의 "0"의 연속 비트를 포함할 수 있지만, user_data에서 23개의 "0"의 연속 비트를 생성하는 것은 MEPG 규격에 의해 금지된다. 그래서, 변환기(212)는 금지된 바에 따라 "1" 비트의 삽입으로 히스토리 정보를 변환시킨다.

사용자 데이터 포맷터(213)는 video_stream에 삽입될 수 있는 user_data를 생성하도록 추후 기술된 도 38에 도시된 구문(syntax)을 근거로 변환기(212)에 의해 공급된 converted_history_stream()에 Data_ID 및 user_data_stream_code를 부가하고, user_data를 부호화 장치(106)에 출력한다.

도 20은 히스토리 포맷터(211)의 전형적인 구성을 도시하는 블록도이다. 도면에 도시된 바와 같이, 코드어 변환기(code language)(301)와 코드어 변환기(305)는 부호화 파라미터 분리 회로(105)로부터 항목 데이터와 항목 번호 데이터를 수신한다. 항목 데이터는 부호화 파라미터, 즉 이때 히스토리 정보로 전송된 부호화 파라미터이다. 항목 번호 데이터는 부호화 파라미터를 포함하는 스트림을 식별하는데 사용되는 정보이다. 항목 번호 데이터의 예는 추후 기술될 sequence_header의 명칭과 구문 명칭이다. 코드어 변환기(301)는 공급된 부호화 파라미터를 지정된 구문에 따른 코드어로 변환시키고, 그 코드를 배럴 시프터 (barrel shifter)(302)로 출력한다. 배럴 시프터(302)는 어드레스 생성 회로(306)에 의해 공급된 정보에 대응하는 시프트 양 만큼 공급된 코드어를 배럴 시프트 처리하고, 시프트된 코드를 바이트 단위로 스위치(303)로 출력한다. 접촉 위치가 어드레스 생성 회로(306)에 의해 출력된 비트 선택 신호로 변경될 수 있는 스위치(303)는 배럴 시프터(302)에 의해 공급되는 비트 만큼의 접촉폴(contact pole)쌍을 갖는다. 스위치(303)는 배럴 시프터(302)에 의해 공급된 코드를 RAM 유닛(304)에 전달하여, 어드레스 생성 회로(306)에 의해 지정된 기록 어드레스에 기록한다. RAM 유닛(304)에 저장된 코드어는 어드레스 생성 회로(306)에 의해 지정된 판독 어드레스로부터 다시 판독되어, 다음 단계에서 제공되는 변환기(212)에 공급된다. 필요한 경우, RAM 유닛(304)으로부터 판독된 코드는 다시 스위치(303)를 통해 RAM 유닛(304)에 공급되어 다시 저장된다.

코드길이 변환기(305)는 공급된 부호화 파라미터와 구문으로부터의 부호화 파라미터의 코드 길이를 결정하여, 코드 길이에 대한 정보를 어드레스 생성 회로(406)로 출력한다. 어드레스 생성 회로(306)는 코드길이 변환기(305)로부터 수신된 코드 길이에 관한 정보에 따라 상술된 시프트량, 비트 선택 신호, 기록 어드레스, 및 판독 어드레스를 생성한다. 시프트량, 비트 선택 신호, 및 어드레스는 각각 배럴 시프터(302), 스위치(303), 및 RAM 유닛(304)에 공급된다.

상술된 바와 같이, 히스토리 포맷터(211)는 공급된 부호화 파라미터에 가변 길이 부호화 처리를 실행하고 가변 길이 부호화 처리의 결과를 출력하는 가변길이 인코더로 동작된다.

도 22는 변환기(212)의 전형적인 구성을 도시하는 블록도이다. 이 전형적인 구성에서는 히스토리 포맷터(211)와 변환기(212) 사이에 제공된 버퍼 메모리 유닛(320)에서 판독 어드레스로부터 8-비트 데이터가 판독되고, 8-비트 D형 플립플롭(D-FF) 회로(321)에 공급되어 그에 유지된다. 판독 어드레스는 제어기(326)에 의해 생성된다. D형 플립플롭 회로(321)로부터 판독된 8-비트 데이터는 스터프(stuff) 회로(323) 및 8-비트 D형 플립플롭 회로(322)에 공급되어 그에 유지된다. D형 플립플롭 회로(322)로부터 판독된 8-비트 데이터는 또한 스터프 회로(323)에 공급된다. 상세하게, D형 플립플롭 회로(321)로부터 판독된 8-비트 데이터는 D형 플립플롭 회로(322)로부터 판독된 8-비트 데이터와 연관되어 스터프 회로(323)에 공급되는 16-비트의 병렬 데이터를 형성한다.

스터프 회로(323)는 신호에 의해 지정된 스터프 위치에 코드 "1"을 삽입하여, 배럴 시프터(324)에 공급되는 총 17 비트의 데이터를 만든다. 스터프 위치를 나타내는 신호는 제어기(326)에 의해 공급되고, 코드 "1"을 삽입하는 동작은 스터핑(stuffing)이라 칭하여진다.

배럴 시프터(324)는 스터프 회로(323)에 의해 공급된 데이터를 제어기(326)로부터 수신된 신호로 나타내지는 시프트량만큼 배럴 시프트 처리하고, 시프트된 것에서 8-비트 데이터를 추출한다. 이어서, 추출된 데이터는 8-비트 D형 플립플롭 회로(325)로 출력되어 그에 유지된다. D형 플립플롭 회로(325)에 유지되는 데이터는 최종적으로 버퍼 메모리 유닛(327)을 통해 다음의 단계에 제공되는 사용자 데이터 포맷터(213)로 출력된다. 말하자면, 데이터는 제어기(326)에 의해 생성된 기록 어드레스로 변환기(212)와 사용자 데이터 포맷터(213) 사이에 제공되는 버퍼 메모리 유닛(327)에 일시적으로 저장된다.

도 23은 스터프 회로(323)의 전형적인 구성을 도시하는 블록도이다. 이 구성에서는 D형 플립플롭 회로 (321) 및 (322)로부터 수신된 16-비트 데이터가 스위치 (331-16) 내지 (331-1)의 접촉점(a)에 공급된다. 스위치(331-i) (여기서, i = 0 내지 15)의 접촉점(a)에 공급되는 데이터 부분은 또한 스위치(331-i)의 접촉점(c)에 공급된다. 스위치(331-i) (여기서, i = 1 내지 16)는 스위치(331-i) (여기서, i = 0 내지 15)에서 도면에 도시된 MSB측으로 각각 스위치(331-i) (여기서, i = 0 내지 15)에 인접한 스위치이다. 예를 들면, 인접한 스위치(331-12)의 MSB측에서 스위치(331-13)의 접촉점(a)에 공급된 LSB로부터의 제 13 데이터 부분은 또한 스위치(331-12)의 접촉점(c)에 공급된다. 동시에, 인접한 스위치(331-14)의 MSB측에서 스위치(331-14)의 접촉점(a)에 공급된 LSB로부터의 제 14 데이터 부분은 또한 스위치(331-13)의 접촉점(c)에 공급된다.

그러나, 스위치(331-1)의 하단측에 제공되는 스위치(331-0)의 접촉점(a)은 LSB에 대응하는 스위치(331-0)의 하단측에 스위치가 제공되지 않으므로 개방상태이다. 부가하여, 스위치(331-15)의 상단측에 제공되는 스위치(331-16)의 접촉점(c)은 또한 MSB에 대응하는 스위치(331-16)의 상단측에 스위치가 제공되지 않으므로 개방상태이다.

스위치 (331-0) 내지 (331-16)의 접촉점(b)에는 데이터 "1"이 제공된다. 디코더(332)는 데이터 "1"을 스터프 위치에 삽입하기 위해 제어기(326)로부터 접촉점(b)으로 수신된 스터프 위치 신호에 의해 나타내지는 스터프 위치에서 스위치 (331-0) 내지 (331-16) 중 하나를 변경시킨다. 스터프 위치에 있는 스위치의 LSB측에서 스위치 (331-0) 내지 (331-16)은 접촉점(c)으로 변경되는 반면, 스터프 위치에 있는 스위치의 MSB측에서 스위치(331)는 접촉점(a)로 변경된다.

도 23은 LSB측으로부터 제 13 비트에 데이터 "1"이 삽입되는 예를 도시한다. 그래서, 이 경우에는 스위치 (331-0) 내지 (331-12)가 접촉점(c)로 변경되고, 스위치 (331-14) 내지 (331-16)가 접촉점(a)로 변경된다. 스위치(331-13)는 접촉점(b)으로 변경된다.

상술된 구성으로, 도 22에 도시된 변환기(212)는 22-비트 코드를 데이터 "1"을 포함하는 23-비트 코드로 변환하고, 23-비트의 변환 결과를 출력한다.

도 24는 도 22에 도시된 변환기(212)의 다양한 부분에 의해 출력되는 데이터 부분의 타이밍을 도시하는 도면이다. 변환기(212)에서 사용되는 제어기(326)가 한 바이트의 데이터에 대해 클럭 신호와 동기화되어 도 24a에 도시된 판독 어드레스를 생성할 때, 판독 어드레스에 저장된 바이트 데이터는 버퍼 메모리 유닛(320)으로부터 판독되어 D형 플립플롭 회로(321)에 의해 일시적으로 유지된다. 이어서, D형 플립플롭 회로(321)로부터 판독된 도 24b의 데이터는 스터프 회로(323)와 D형 플립플롭 회로(322)에 공급되어 그에 유지된다. D형 플립플롭 회로(322)로부터 판독된 도 24c의 데이터는 D형 플립플롭 회로(321)로부터 판독된 도 24b의 데이터와 연관되고, 도 24d에 도시된 연관시킨 결과로 구해지는 데이터는 스터프 회로(323)에 공급된다.

그 결과로, 판독 어드레스(A1)의 타이밍에, D형 플립플롭 회로(321)로부터 판독된 도 24b의 데이터 중 제 1 바이트(D0)는 도 24d에 도시된 데이터의 제 1 바이트로 스터프 회로(323)에 공급된다. 이어서, 판독 어드레스(A1)의 타이밍에, D형 플립플롭 회로(321)로부터 판독되고 D형 플립플롭 회로(322)로부터 판독된 도 24c의 데이터 중 제 1 바이트(D0)와 연관된 도 24b의 데이터 중 제 2 바이트(D1)는 도 24d에 도시된 데이터의 제 2의 두 바이트로 스터프 회로(322)에 공급된다. 이어서, 판독 어드레스(A3)의 타이밍에, D형 플립플롭 회로(321)로부터 판독되고 D형 플립플롭 회로(322)로부터 판독된 도 24c의 데이터 중 제 2 바이트(D1)와 연관된 도 24b의 데이터 중 제 3 바이트(D2)는 도 24d에 도시된 데이터의 제 3의 두 바이트로 스터프 회로(322)에 공급된다.

스터프 회로(323)는 제어기(326)로부터 데이터 "1"이 삽입되는 스터프 위치를 나타내는 도 24e의 신호를 수신한다. 스터프 회로(323)에서 사용되는 디코더(332)는 스터프 위치에 있는 스위치 (331-0) 내지 (331-16) 중 하나를 접촉점(b)으로 변경시킨다. 스터프 위치에 있는 스위치의 LSB측으로 스위치(331)는 접촉점(c)로 변경되는 반면, 스터프 위치에 있는 스위치의 MSB측으로 스위치(331)는 접촉점(a)로 변경된다. 그 결과로, 스터프 회로(323)는 스터프 위치 신호에 데이터 "1"을 삽입하고, 데이터 "1"이 삽입된 도 24f의 데이터를 출력한다.

배럴 시프터(324)는 제어기(326)로부터 수신된 도 24g의 시프트 신호에 의해 나타내지는 시프트량 만큼 스터프 회로(323)에 의해 공급된 데이터를 배럴 시프트처리하고, 도 24h에 도시되는 시프트된 신호를 출력한다. 시프트된 신호는 도 24i에 도시된 바와 같이 나중 단계로 출력되기 이전에 D형 플립플롭 회로(325)에 일시적으로 유지된다.

D형 플립플롭 회로(325)로부터 출력된 데이터는 22-비트 데이터에 이어지는 위치에 삽입된 데이터 "1"을 포함한다. 그래서, 데이터 "1"과 다음 데이터 "1" 사이의 모든 비트가 "0"이더라도, 연속적인 0 비트의 수는 결코 22를 넘지 않는다.

도 25는 변환기(202)의 전형적인 구성을 도시하는 블록도이다. D형 플립플롭 회로(321)에서 제어기(326) 범위의 구성성분이 도 22에 도시된 변환기(212)에서 사용되는 것과 같이, D형 플립플롭 회로(341)에서 제어기(346) 범위의 구성성분이 변환기(202)에서 사용된다는 사실은 두 구성이 기본적으로 똑같다는 것을 나타낸다. 변환기(202)는 변환기(212)의 스터프 회로(323) 대신에 삭제 회로(343)가 사용된다는 점에서 변환기(212)와 다르다. 그렇지 않은 경우, 변환기(202)의 구성은 도 22에 도시된 변환기(212)와 똑같다.

변환기(202)에서 사용되는 삭제 회로(343)는 제어기(346)에 의해 출력되는 신호로 나타내지는 삭제 위치에서 비트를 삭제한다. 삭제 위치는 도 23에 도시된 스터프 회로(323)가 데이터 "1"을 삽입한 스터프 위치에 대응한다.

변환기(202)의 나머지 동작은 도 22에 도시된 변환기(212)에 의해 실행되는 것과 똑같다.

도 26은 삭제 회로(343)의 전형적인 구성을 도시하는 블록도이다. 이 구성에서는 D형 플립플롭 회로 (341) 및 (342)로부터 수신된 16-비트 데이터의 LSB측으로 15 비트가 스위치 (351-0) 내지 (351-14)의 접촉점(a)에 공급된다. 스위치 (351-i) (여기서, i = 1 내지 14)의 접촉점(a)에 공급되는 데이터 부분은 또한 스위치(351-i) (여기서, i = 0 내지 13)의 접촉점(b)에 공급된다. 스위치(351-i) (여기서, i = 1 내지 14)는 스위치(351-i) (여기서, i = 0 내지 13)의 MSB측으로 (또는 도면에 도시된 상단측으로) 각각 스위치(351-i) (여기서, i = 0 내지 13)에 인접한 스위치이다. 예를 들면, 인접한 스위치(351-12)의 MSB측에서 스위치(351-13)의 접촉점(a)에 공급된 LSB로부터의 제 13 데이터 부분은 또한 스위치(351-12)의 접촉점(b)에 공급된다. 동시에, 인접한 스위치(351-13)의 MSB측에서 스위치(351-14)의 접촉점(a)에 공급된 LSB로부터의 제 14 데이터 부분은 또한 스위치(351-13)의 접촉점(c)에 공급된다. 디코더(352)는 제어기(346)에 의해 출력된 신호로 나타내지는 삭제 위치에서 한 비트를 삭제하고, 삭제된 비트를 제외한 나머지 15-비트 데이터를 출력한다.

도 26은 LSB로부터 제 13 입력 비트 (입력 비트 12)가 삭제된 상태를 도시한다. 그래서, 이 경우에는 스위치 (351-0) 내지 (351-11)가 접촉점(a)으로 변경되어, LSB (비트 0)로부터 제 12 비트 (비트 11)까지의 12 입력 비트를 그대로 출력한다. 한편, 스위치 (351-12) 내지 (351-14)는 접촉점(b)으로 변경되어, 제 14 내지 제 16 입력 비트 (입력 비트 13 내지 15)를 각각 제 13 내지 제 15 출력 비트 (출력 비트 12 내지 14)로 전달한다. 이 상태에서, LSB로부터의 제 13 입력 비트 (입력 비트 12)는 출력선에 연결되지 않는다.

16-비트 데이터는 도 23에 도시된 스터프 회로(323)와 도 26에 도시된 삭제 회로(343)에 공급된다. 이는 스터프 회로(323)에 공급된 데이터가 도 22에 도시된 변환기(212)에서 사용되는 8-비트 D형 플립플롭 회로 (321) 및 (322)에 의해 출력된 데이터 부분을 연관시킨 결과이기 때문이다. 유사하게, 삭제 회로(343)에 공급된 데이터는 도 25에 도시된 변환기(202)에서 사용되는 8-비트 D형 플립플롭 회로 (341) 및 (342)에 의해 출력된 데이터 부분을 연관시킨 결과이다. 도 22에 도시된 변환기(212)에서 사용되는 배럴 시프터(324)는 제어기(326)로부터 수신된 신호에 의해 나타내지는 시프트량 만큼

스터프 회로(323)에 의해 공급된 17-비트 데이터를 배럴 시프트 처리하고, 마지막으로 시프트된 것에서 전형적으로 8 비트의 데이터를 추출한다. 유사하게, 도 25에 도시된 변환기(202)에서 사용되는 배럴 시프터(344)는 제어기(346)로부터 수신된 신호에 의해 나타내지는 시프트량 만큼 스터프 회로(324)에 의해 공급된 15-비트 데이터를 배럴 시프트 처리하고, 마지막으로 시프트된 것에서 8 비트의 데이터를 추출한다.

도 21은 변환기(202)에서 히스토리 포맷팅 처리를 완료한 데이터를 복호화하기 위한 히스토리 디코더(203)의 전형적인 구성을 도시하는 블록도이다. 변환기(202)에 의해 히스토리 디코더(203)로 공급된 부호화 파라미터의 데이터는 RAM 유닛(311)에 공급되어, 어드레스 생성 회로(315)에 의해 지정된 기록 어드레스에 저장된다. 어드레스 생성 회로(315)는 또한 소정의 타이밍으로 판독 어드레스를 RAM 유닛(311)에 출력한다. 이때, RAM 유닛(311)에서 판독 어드레스에 저장된 데이터는 배럴 시프터(312)로 출력된다. 배럴 시프터(312)는 어드레스 생성 회로(315)에 의해 공급된 정보에 대응하는 시프트량 만큼 공급된 데이터를 배럴 시프트처리하고, 시프트된 데이터를 역코드길이 변환기 (313) 및 (314)로 출력한다.

역코드길이 변환기 (313) 및 (314)는 또한 변환기(202)로부터 부호화 파라미터를 포함하는 스트림의 구문명(name of syntax)을 수신한다. 역코드길이 변환기(313)는 구문에 따라 공급된 코드어 또는 데이터로부터 부호화 파라미터의 코드 길이를 결정하고, 코드 길이에 관한 정보를 어드레스 생성 회로(315)로 출력한다.

한편, 역코드길이 변환기(314)는 구문을 근거로 배럴 시프터(312)에 의해 공급된 데이터를 복호화 또는 역부호화하고, 복호화 처리 결과를 부호화 파라미터 다중화 장치(103)로 출력한다.

부가하여, 역코드길이 변환기(314)는 또한 무슨 코드어가 포함되었나를 식별하는데 요구되는 정보, 즉 코드의 구획 문자(delimiter)를 결정하는데 요구되는 정보를 추출하고, 그 정보를 어드레스 생성 회로(315)로 출력한다. 어드레스 생성 회로(315)는 이 정보 및 역코드길이 변환기(313)로부터 수신된 코드 길이를 근거로 기록 및 판독 어드레스와 시프트량을 생성하고, 기록/판독 어드레스를 RAM 유닛(311)으로, 또한 시프트량을 배럴 레지스터(312)로 출력한다.

도 27은 변환기(212)의 또 다른 전형적인 구성을 도시하는 블록도이다. 이 구성에서 사용되는 카운터(361)는 공급된 데이터 중 연속적인 0 비트의 수를 카운트하고, 카운트 결과를 제어기(326)로 출력한다. 카운터(361)에 공급된 데이터 중 연속적인 0 비트의 수가 22에 이를 때, 제어기(326)는 스터프 위치를 나타내는 신호를 스터프 회로(323)로 출력한다. 동시에, 제어기(326)는 카운터(361)를 재설정하여, 카운터(361)가 0으로부터 다시 연속적인 0 비트의 수를 카운트하기 시작하도록 허용한다. 나머지 구성 및 동작은 도 22에 도시된 변환기(212)와 똑같다.

도 28은 변환기(202)의 또 다른 전형적인 구성을 도시하는 블록도이다. 이 구성에서 사용되는 카운터(371)는 공급된 데이터 중 연속적인 0 비트의 수를 카운트하고, 카운트 결과를 제어기(346)로 출력한다. 카운터(371)에 공급된 데이터 중 연속적인 0 비트의 수가 22에 이를 때, 제어기(346)는 삭제 위치를 나타내는 신호를 삭제 회로(343)로 출력한다. 동시에, 제어기(346)는 카운터(371)를 재설정하여, 카운터(371)가 0으로부터 다시 연속적인 0 비트의 수를 카운트하기 시작하도록 허용한다. 나머지 구성과 동작은 도 25에 도시된 변환기(202)와 똑같다.

상술된 바와 같이, 도 27 및 도 28에 도시된 전형적인 구성에서, 데이터 "1"은 소정의 연속적인 0 비트의 수가 카운터에 의해 검출될 때 각각 마커 비트로 삽입되고 삭제된다. 도 27 및 도 28에 도시된 전형적인 구성은 도 22 및 도 25에 각각 도시된 구성 보다 더 높은 효율도를 갖고 처리가 실행되도록 허용한다.

도 29는 사용자 데이터 포맷터(213)의 전형적인 구성을 도시하는 블록도이다. 이 구성에서는 제어기(383)가 변환기(212)와 사용자 데이터 포맷터(213) 사이에 제공되는 도시되지 않은 버퍼 메모리로 판독 어드레스를 출력할 때, 데이터가 판독 어드레스로부터 출력되어, 사용자 데이터 포맷터(213)에서 사용되는 스위치(382)의 접촉점(a)에 공급된다. 도면에서는 버퍼 메모리가 도시되지 않음을 주목하여야 한다. ROM 유닛(381)에서는 사용자 데이터 시작 코드 및 데이터 ID와 같은 user_data()를 생성하는데 요구되는 데이터가 저장된다. 제어기(313)는 스위치(382)가 ROM 유닛(381)에 저장된 데이터나 변환기(212)에 의해 공급된 데이터를 선택하고 선택된 데이터를 전달하도록 허용하기 위해 소정의 타이밍에 스위치(382)를 접촉점(a)나 접촉점(b)로 변경시킨다. 이 방법으로, user_data()의 포맷을 갖는 데이터가 부호화 장치(106)로 출력된다.

사용자 데이터 디코더(201)는 도 29에 도시된 사용자 데이터 포맷터(213)에서 사용되는 ROM 유닛(381)과 같은 ROM 유닛으로부터 판독된 삽입 데이터를 삭제하도록 스위치를 통해 입력 데이터를 출력함으로써 실행될 수 있음을 주목한다. 사용자 데이터 디코더(201)의 구성은 도면에서 도시되지 않는다.

도 30은 다수의 트랜스코더 (101-1) 내지 (101-N)가 비디오 편집 스튜디오 서비tm의 사용을 위해 직렬로 연결된 실행 상태를 도시하는 블록도이다. 트랜스코더(101-i) (여기서, i = 1 내지 N)에서 사용되는 부호화 파라미터 다중화 장치(103-i)는 각각 부호화 파라미터를 기록하는데 사용되는 영역에서 가장 오래된 부호화 파라미터를 기록한 영역에 걸쳐 그에 의해 사용되는 가장 최근의 부호화 파라미터를 기록한다. 그 결과로, 기저대 화상 데이터는 화상 데이터의 매크로블록과 연관된 4개의 가장 최근 세대의 생성 히스토리 정보 또는 부호화 파라미터를 포함한다.

부호화 장치(106-i)에서 사용된 도 19의 인코더(121-i)에서 사용되는 가변길이 부호화 회로(58)는 부호화 파라미터 분리 회로(105-i)로부터 수신된 현재 부호화 파라미터를 근거로 양자화 회로(57)로부터 수신된 비디오 데이터를 부호화한다. 그 결과로, 현재 부호화 파라미터는 가변길이 부호화 회로(58)에 의해 생성된 비트스트림에 포함되는 picture_header()에서 전형적으로 다중화된다.

부가하여, 가변길이 부호화 회로(58)는 또한 생성 히스토리 정보를 포함하고 히스토리 부호화 장치(107-i)로부터 수신되는 사용자 데이터를 출력 비트스트림에 다중화한다. 이 다중화 처리는 도 18에 도시된 것과 같이 실현된 처리가 아니라, 사용자 데이터를 비트스트림에 다중화하는 것이다. 이어서, 부호화 장치(106-i)에 의해 출력된 비트스트림은 SDTI(108-i)를 통해 다음 단계에서 트랜스코더(101-(i+ 1))에 공급된다.

트랜스코더 (101-i) 및 (101-(i+ 1))의 구성은 도 15에 도시된 것과 똑같다. 그래서, 이들에 의해 실행되는 처리는 도 15를 참고로 설명될 수 있다. 실제 부호화 파라미터의 히스토리를 사용한 부호화 동작에서 현재 화상 타입을 I-화상에서 P- 또는 B-화상으로 변환시키기를 원하면, 이전에 사용된 P- 또는 B-화상에 대해 이전 부호화 파라미터의 히스토리가 탐색된다. 히스토리에서 P- 또는 B-화상의 히스토리가 발견되면, 이동 벡터를 포함하는 파라미터는 화상 타입을 변환시키는데 사용된다. 한편, P- 또는 B-화상의 히스토리가 히스토리에서 발견되지 않으면, 이동 벡터 없이 화상 타입을 수정하는 것은 포기된다. 이동 검출이 실행되다고 가정하면, P- 또는 B-화상의 부호화 파라미터가 히스토리에서 발견되지 않더라도, 화상 타입이 변환될 수 있음은 말할 필요도 없다.

도 18에 도시된 포맷에서, 제 4 세대의 부호화 파라미터는 화상 데이터에 포함된다. 다른 방법으로, I-, P-, 및 B-화상 각각에 대한 파라미터가 도 31에 도시된 바와 같은 포맷으로 포함될 수 있다. 도 31에 도시된 예에서, 한 세대의 화상 히스토리 정보 또는 부호화 파라미터는 이전에 생성되는 화상 타입에서의 변환을 수반하여 같은 매크로블록을 부호화하는 동작에서 각 화상 타입에 대해 기록된다. 이 경우, 도 16에 도시된 디코더(111)는 가장 최근의 제 1, 제 2, 및 제 3 선행 세대의 부호화 파라미터 대신에, 도 19에 도시된 인코더(121)에 공급되는 I-, P-, 및 B-화상에 대한 한 세대의 부호화 파라미터를 출력한다.

부가하여, Cb[1][x] 및 Cr[1][x]의 영역이 사용되지 않으므로, 본 발명은 또한 Cb[1][x] 및 Cr[1][x]의 영역을 사용하지 않는 4 : 2 : 0 포맷의 화상 데이터에 적용될 수 있다. 본 예의 경우에는 복호화 장치(102)가 복호화하는 중에 부호화 파라미터를 인출하여 화상 타입을 식별한다. 복호화 장치(102)는 화상 신호의 화상 타입에 대응하는 위치에 부호화 파라미터를 기록 또는 다중화하고, 다중화된 화상 신호를 부호화 파라미터 분리 장치(105)로 출력한다. 부호화 파라미터 분리 장치(105)는 화상 데이터로부터 부호화 파라미터를 분리하고, 분리된 부호화 파라미터를 사용해, 부호화 파라미터 분리 장치(105)는 공급된 이전 부호화 파라미터와 변환되는 화상 타입을 고려함으로써 화상 타입을 변환시키면서 복호화 이후 부호화 (post-decoding-encoding) 처리를 실행할 수 있다.

트랜스코더(101)는 제어기(70)가 이동벡터 검출 회로의 동작을 허용하지 않는 경우에만 변환가능한 화상 타입을 결정하는 파라미터 재사용 부호화 처리와 다른 또 다른 동작을 갖는다.

다른 동작은 도 32에 도시된 흐름도를 참고로 설명된다. 도 32에 도시된 바와 같이, 흐름도는 각 화상 타입에 대해 한 세대의 화상 히스토리 정보 또는 부호화 파라미터가 인코더(121)의 제어기(70)에 공급되는 단계(S1)로 시작된다. 이어서, 처리 흐름은 화상 히스토리 정보가 B-화상으로의 변환에 사용되는 부호화 파라미터를 포함하는가 여부에 대해 부호화 파라미터 분리 장치(105)가 판단하는 단계(S2)로 진행된다. 화상 히스토리 정보가 B-화상으로의 변환에 사용되는 부호화 파라미터를 포함하면, 처리 흐름은 단계(S3)로 진행된다.

단계(S3)에서, 제어기(70)는 화상 히스토리 정보가 P-화상으로의 변환에 사용되는 부호화 파라미터를 포함하는가 여부를 판단한다. 화상 히스토리 정보가 P-화상으로의 변환에 사용되는 부호화 파라미터를 포함하면, 처리 흐름은 단계(S4)로 진행된다.

단계(S4)에서, 제어기(70)는 변환가능한 화상 타입이 I-, P-, 또는 B-화상임을 결정한다. 한편, 단계(S3)의 판단 결과에서 화상 히스토리 정보가 P-화상으로의 변환에 사용되는 부호화 파라미터를 포함하지 않으면, 처리 흐름은 단계(S5)로 진행된다.

단계(S5)에서, 제어기(70)는 변환가능한 화상 타입이 I- 및 B-화상임을 결정한다. 부가하여, 제어기(70)는 B-화상의 히스토리 정보에 포함되는 후방 예측 벡터 이외에 전방 예측 벡터만을 사용해 특수 처리를 실행함으로써 P-화상으로의 의사변환(pseudo-change)이 또한 가능함을 결정한다. 한편, 단계(S2)의 판단 결과에서 화상 히스토리 정보가 B-화상으로의 변환에 사용되는 부호화 파라미터를 포함하지 않는 것으로 나타내지면, 처리 흐름은 단계(S6)로 진행된다.

단계(S6)에서, 제어기(70)는 화상 히스토리 정보가 P-화상으로의 변환에 사용되는 부호화 파라미터를 포함하는가 여부를 판단한다. 화상 히스토리 정보가 P-화상으로의 변환에 사용되는 부호화 파라미터를 포함하면, 처리 흐름은 단계(S7)로 진행된다.

단계(S7)에서, 제어기(70)는 변환가능한 화상 타입이 I- 및 P-화상임을 결정한다. 부가하여, 부호화 파라미터 분리 장치(105)는 P-화상의 히스토리 정보에 포함되는 후방 예측 벡터 이외에 전방 예측 벡터만을 사용해 특수 처리를 실행함으로써 B-화상으로의 변환이 또한 가능함을 결정한다.

한편, 단계(S6)의 판단 결과에서 화상 히스토리 정보가 P-화상으로의 변환에 사용되는 부호화 파라미터를 포함하지 않는 것으로 나타내지면, 처리 흐름은 단계(S8)로 진행된다. 단계(S8)에서, 제어기(70)는 이동 벡터가 없기 때문에 변환가능한 화상 타입은 단지 I-화상임을 결정한다. I-화상은 I-화상 이외의 다른 화상 타입으로 변환될 수 없다.

단계 (S4), (S5), (S7), 또는 (S8)를 완료한 이후에, 처리 흐름은 제어기(70)가 도면에 도시되지 않은 디스플레이 유닛에 변환가능한 화상 타입을 사용자에게 통보하는 단계(S9)로 진행된다.

도 33은 화상 타입에서의 변환예를 도시하는 도면이다. 화상 타입이 변할 때, GOP 구조를 구성하는 프레임의 수가 변한다. 상세하게, 본 예에서는 긴 GOP 구조가 제 2 세대의 짧은 GOP 구조로 변한다. 이어서, 제 2 세대의 GOP 구조는 다시 제 3 세대의 긴 GOP로 변한다. 긴 GOP 구조는 N = 15 및 M = 3을 갖는다. 여기서, N은 GOP를 구성하는 프레임의 수이고, M은 프레임으로 나타낸 P-화상의 출현 주기이다. 한편, 짧은 GOP는 N = 1 및 M = 1을 갖는다. 여기서, M은 프레임으로 나타낸 I-화상의 출현 주기이다. 도면에서 도시되는 점선은 인접한 두 GOP 사이의 경계를 나타낸다.

제 1 세대의 GOP 구조가 제 2 세대의 GOP 구조로 변할 때, 모든 프레임의 화상 타입은 상기 주어진 변환가능한 화상 타입을 결정하는 처리의 설명으로부터 명백한 바와 같이 I-화상으로 변환될 수 있다. 화상 타입이 변하는 경우, 소스 비디오 신호가 제 1 세대에서 부호화되었을 때 처리되었던 모든 이동 벡터는 저장되거나 남겨진다. 이때, 짧은 GOP 구조는 제 3 세대에서 긴 GOP 구조로 다시 변한다. 즉, 화상 타입이 변하더라도, 소스 비디오 신호가 제 1 세대에서 부호화되었을 때 저장되었던 각 종류에 대한 이동 벡터는 재사용되어, 화상 품질의 열화를 방지하면서 긴 GOP 구조로의 재변환이 이루어지도록 허용한다.

도 34는 화상 타입에서의 또 다른 변환예를 도시하는 도면이다. 본 예의 경우에서는 제 2 세대에서 N = 14 및 M = 2를 갖는 긴 GOP 구조에서 N = 2 및 M = 2를 갖는 짧은 GOP 구조로의 변환이 이루어지고, 이어서 N = 1 및 M = 1을 갖는 짧은 GOP 구조로의 변환, 또한 마지막으로 제 4 세대에서 결정되지 않은 프레임 카운트 N을 갖는 랜덤 GOP로의 변환이 이루어진다.

본 예에서는 또한, 제 1 세대에서 소스 비디오 신호가 부호화되었을 때 처리되었던 각 화상 타입에 대한 이동 벡터가 제 4 세대까지 저장된다. 그 결과로, 저장된 부호화 파라미터를 재사용함으로써, 도 34에 도시된 바와 같이 화상 타입이 복잡한 방식으로 변하더라도 화상 품질의 열화가 최소로 줄어들 수 있다. 부가하여, 저장된 부호화 파라미터의 양자화 스케일이 효과적으로 사용되면, 화상 품질의 열화를 거의 수반하지 않는 부호화 처리가 실행될 수 있다.

양자화 스케일의 재사용은 도 35를 참고로 설명된다. 도 35는 특정한 기준 프레임이 항상 제 1 세대에서 제 4 세대까지 I-화상으로 부호화되는 경우를 도시하는 도면이다. 비트 레이트만이 제 1 세대에서의 4 Mbps에서 제 2 세대에서의 18 Mbps로 변하고, 이어서 제 3 세대에서의 50 Mbps로, 마지막으로 제 4 세대에서의 4 Mbps로 다시 변환된다.

제 1 세대에서 생성된 비트스트림의 4 Mbps 비트 레이트가 제 2 세대에서 18 Mbps의 비트 레이트로 변할 때는 비트 레이트의 증가와 수반되는 미세한 양자화 스케일로 복호화 이후 부호화 처리가 실행되더라도, 화상 품질은 개선되지 않는다.

이는 거친 양자화 단계로 이전에 양자화된 데이터가 재저장되지 않기 때문이다. 그래서, 도 35에 도시된 바와 같이 처리중에 비트 레이트의 상승과 수반되는 미세한 양자화 단계로의 양자화는 단순히 정보량을 증가시키고 화상 품질의 개선을 이끌어내지는 못한다. 이러한 이유로, 이전에 사용된 가장 거칠거나 가장 큰 양자화 스케일을 유지하도록 제어가 실행되면, 부호화 처리는 최소의 낭비와 최대의 효율로 실행될 수 있다.

상술된 바와 같이, 비트 레이트가 변할 때는 양자화 스케일의 이전 히스토리를 사용함으로써, 부호화 처리가 가장 효율적으로 실행될 수 있다.

이 양자화 제어 처리는 도 36에 도시된 흐름도를 참고로 설명된다. 도면에서 도시된 바와 같이, 흐름도는 입력 화상 히스토리 정보가 지금으로부터 변환되는 화상 타입의 부호화 파라미터를 포함하는가 여부를 제어기(70)가 판단하는 단계(S11)로 시작된다. 판단 결과에서 입력 화상 히스토리 정보가 변환되는 화상 타입의 부호화 파라미터를 포함하는 것으로 나타내지면, 처리 흐름은 단계(S12)로 진행된다.

단계(S12)에서, 제어기(70)는 화상 히스토리 정보에 포함되는 비교를 위한 문제의 부호화 파라미터로부터 history_q_scale_code를 추출한다.

처리 흐름은 제어기(70)가 전송 버퍼(59)에 데이터가 채워져 있는 상태를 근거로 feedback_q_scale_code의 후보값을 계산하는 단계(S13)로 진행된다.

처리 흐름은 history_q_scale_code가 feedback_q_scale_code 보다 더 크거나 거친가 여부를 제어기(70)가 판단하는 단계(S14)로 진행된다. 판단 결과에서 history_q_scale_code가 feedback_q_scale_code 보다 더 크거나 거친 것으로 나타내지면, 처리 흐름은 단계(S15)로 계속된다.

단계(S15)에서, 제어기(70)는 history_q_scale_code를 양자화 스케일로 양자화 회로(57)에 공급하고, 이어서 양자화 회로(57)는 history_q_scale_code를 사용해 양자화 처리를 실행한다.

처리 흐름은 프레임에 포함된 모든 매크로블록이 양자화되었나 여부를 판단하는 단계(S16)로 진행된다. 판단 결과에서 프레임에 포함된 모든 매크로블록이 아직 양자화되지 않은 것으로 나타내지면, 처리 흐름은 다시 단계(S13)로 진행되어, 프레임에 포함된 모든 매크로블록이 양자화될 때까지 단계(S13) 내지 (S16)의 처리 부분을 반복적으로 실행한다.

한편, 단계(S14)에서 형성된 판단 결과에서 history_q_scale_code가 feedback_q_scale_code 보다 크지 않은 것으로 나타내지면, 즉 history_q_scale_code가 feedback_q_scale_code 보다 더 미세하면, 처리 흐름은 단계(S17)로 계속된다.

단계(S17)에서, 제어기(70)는 feedback_q_scale_code를 양자화 스케일로 양자화 회로(57)에 공급하고, 이어서 양자화 회로(57)는 feedback_q_scale_code를 사용해 양자화 처리를 실행한다.

한편, 단계(S11)에서 형성된 판단 결과에서 입력 화상 히스토리 정보가 변환되는 화상 타입의 부호화 파라미터를 포함하지 않는 것으로 나타내지면, 처리 흐름은 단계(S18)로 진행된다.

단계(S18)에서, 양자화 회로(57)는 제어기(70)로부터 feedback_q_scale_code의 후보값을 수신한다.

이어서, 처리 흐름은 양자화 회로(57)가 Q_feedback을 사용해 양자화 처리를 실행하는 단계(S19)로 진행된다.

처리 흐름은 프레임에 포함된 모든 매크로블록이 양자화되었나 여부를 제어기(70)가 판단하는 단계(S20)로 진행된다. 판단 결과에서 프레임에 포함된 모든 매크로블록이 아직 양자화되지 않은 것으로 나타내지면, 처리 흐름은 단계(S18)로 진행되어, 프레임에 포함된 모든 매크로블록이 양자화될 때까지 단계(S18) 내지 (S20)의 처리 부분을 반복적으로 실행한다.

도 15를 참고로 앞서 설명된 트랜스코더(101)는 기저대 비디오 데이터에서 부호화 파라미터를 다중화함으로써 제 1, 제 2, 및 제 3 세대의 이전 부호화 파라미터를 비디오 부호화 장치(106)에 공급한다. 그러나, 본 발명에서는 기저대 비디오 데이터에서 이전 부호화 파라미터를 다중화하는 기술이 절대적으로 요구되지는 않는다. 예를 들면, 이전 부호화 파라미터는 도 37에 도시된 바와 같이 기저대 비디오 데이터에 대해 그로부터 분리되어 제공되는 데이터 전달 버스와 같은 전송선을 사용해 전달될 수 있다.

도 37에 도시된 비디오 복호화 장치(102), 히스토리 복호화 장치(104), 비디오 부호화 장치(106), 및 히스토리 부호화 장치(107)는 도 15를 참고로 앞서 기술되었던 비디오 복호화 장치(102), 히스토리 복호화 장치(104), 비디오 부호화 장치(106), 및 히스토리 부호화 장치(107)와 각각 완전히 같은 구성 및 기능을 갖는다.

비디오 복호화 장치(102)에서 사용되는 가변길이 복호화 회로(112)는 제 3 세대의 부호화 비디오 비트스트림 ST(3rd) 중 시퀀스층, GOP층, 화상층, 슬라이스층, 및 매크로블록층으로부터 제 3 세대의 부호화 파라미터를 추출하고, 그 파라미터를 비디오 부호화 장치(106)에서 사용되는 제어기(70) 및 히스토리 부호화 장치(107)에 공급한다. 히스토리 부호화 장치(107)는 공급된 제 3 세대의 부호화 파라미터를 화상층의 사용자 데이터 영역에 기술될 수 있는 converted_history_stream()으로 변환하고, converted_history_stream()을 비디오 부호화 장치(106)에서 사용되는 가변길이 부호화 회로(58)에 사용자 데이터로 공급한다.

부가하여, 가변길이 복호화 회로(112)는 또한 제 3 세대의 부호화 비디오 비트스트림 ST(3rd) 중 화상층의 사용자 데이터 영역으로부터 제 1 및 제 2 세대의 이전 부호화 파라미터를 포함하는 사용자 데이터(user_data)를 추출하고, user_data를 비디오 부호화 장치(106)에서 사용되는 가변길이 부호화 회로(58) 및 히스토리 복호화 장치(104)에 공급한다. 히스토리 복호화 장치(104)는 converted_history_stream()으로 기술되는 사용자 데이터의 히스토리 스트림으로부터 제 1 및 제 2 세대의 부호화 파라미터를 추출하고, 그 파라미터를 비디오 부호화 장치(106)에서 사용되는 제어기(70)에 공급한다.

비디오 부호화 장치(106)의 제어기(70)는 비디오 복호화 장치(102)로부터 수신된 제 3 세대의 부호화 파라미터와 히스토리 복호화 장치(104)로부터 수신된 제 1 및 제 2 세대의 부호화 파라미터를 근거로 비디오 부호화 장치(106)에 의해 실행되는 부호화 처리를 제어한다.

그 사이에, 비디오 부호화 장치(106)에서 사용되는 가변길이 부호화 회로(58)는 비디오 복호화 장치(102)로부터 제 1 및 제 2 세대의 부호화 파라미터를 포함하는 사용자 데이터 (user_data)를 수신하고, 히스토리 부호화 장치(107)로부터 제 3 세대의 부호화 파라미터를 포함하는 사용자 데이터 (user_data)를 수신하여, 제 4 세대의 부호화 비디오 비트스트림 중 화상층의 사용자 데이터 영역에서 user_data 부분을 기록 정보로 기술한다.

도 38은 MPEG 비디오 스트림을 복호화하는데 사용되는 구문(syntax)을 도시하는 도면이다. 디코더는 비트스트림으로부터 다수의 의미있는 데이터 항목 또는 의미있는 데이터 소자를 추출하기 위해 이 구문에 따라 MPEG 비트스트림을 복호화한다. 이후 설명되는 구문에서, 함수 및 조건문은 각각 정상적인 문자열로 나타내지는 반면, 데이터 소자는 굵은 문자열로 나타내진다. 데이터 항목은 데이터 항목의 명칭을 나타내는 연상 기호(Mnemonic)로 기술된다. 일부 경우에서는 연상 기호가 또한 데이터 항목을 구성하는 비트길이와 데이터 항목을 종류를 나타낸다.

먼저, 도 38에 도시된 구문에서 사용되는 함수가 설명된다. next_start_code()는 비트스트림에서 기술되는 시작 코드에 대해 비트스트림을 탐색하는데 사용되는 함수이다. 도 38에 도시된 구문에서, next_start_code() 함수는 비트스트림이 sequence_header() 및 sequence_extension() 함수에 의해 정의되는 데이터 소자를 포함함을 나타내도록 순차적으로 놓인 sequence_header() 함수와 sequence_extension() 함수로 이어진다. 그래서, 비트스트림을 복호화하는 동작에서 비트스트림으로부터 sequence_header() 및 sequence_extension()의 시작부에 기술되는 일종의 데이터 소자인 시작 코드가 next_start_code()에 의해 찾아진다. 시작 코드는 sequence_header() 및 sequence_extension() 함수를 더 찾고 sequence_header() 및 sequence_extension() 함수에 의해 정의된 데이터 소자를 복호화하는 기준으로 사용된다.

sequence_header() 함수는 MPEG 비트스트림에서 시퀀스층의 헤더 데이터를 정의하는데 사용되는 함수인 반면, sequence_extension() 함수는 MPEG 비트스트림에서 시퀀스층의 확장 데이터를 정의하는데 사용되는 함수임을 주목하여야 한다.

do {} while 문은 sequence_extension() 함수 다음에 기술된다. do {} while 문은 do 문에 이어지는 {} 블록과 {} 블록에 이어지는 while 문을 구비한다. do 문에 이어지는 {} 블록에서 함수에 의해 기술되는 데이터 소자는 while 문에 의해 정의된 조건이 참(true)인 동안 비트스트림으로부터 추출된다. 말하자면, do {} while 구문은 while 문에 의해 정의된 조건이 참인 동안 비트스트림으로부터 do 문에 이어지는 {} 블록내의 함수에 의해 기술되는 데이터 소자를 추출하는 복호화 처리를 정의한다.

while 문에서 사용되는 nextbits()는 비트스트림에 나타나는 비트 또는 비트열을 다음에 복호화될 데이터 소자와 비교하는데 사용되는 함수이다. 도 38에 도시된 구문의 예에서, nextbits() 함수는 비트스트림에 나타나는 비트열을 비디오 시퀀스의 종료를 나타내는데 사용되는 sequence_end_code와 비교한다. while 문에 의해 정의된 조건은 비트스트림에 나타

내는 비트열이 sequence_end_code와 정합되지 않으면 참이다. 그래서, sequence_extension() 함수 이후에 기술된 do {} while 문은 비디오 시퀀스의 종료를 나타내는데 사용되는 sequence_end_code가 비트스트림에서 나타나지 않는 한 do 문에 이어지는 {} 블록에서 함수로 정의된 데이터 소자가 비트스트림에서 기술됨을 나타낸다.

비트스트림에서 sequence_extension() 함수에 의해 정의된 데이터 소자 이후에는 extension_and_user_data(0) 함수에 의해 정의된 데이터 소자가 기술된다. extension_and_user_data(0) 함수는 MPEG 비트스트림의 시퀀스층에서 확장 데이터와 사용자 데이터를 정의하는데 사용되는 함수이다.

extension_and_user_data(0) 함수에 이어지는 do {} while 문은 while 문에 의해 정의된 조건이 참인 동안 비트스트림으로부터 do 문에 이어지는 {} 블록에서 함수로 기술된 데이터 소자를 추출한다. while 문에서 사용되는 nextbits() 함수는 비트스트림에 나타나는 비트 또는 비트열이 그 비트를 함수에서 정의된 시작 코드와 비교함으로써 각각 picture_start_code나 group_start_code 시작 코드와 정합되는가 여부를 판단하는데 사용되는 함수이다. 비트스트림에 나타나는 비트열이 picture_start_code나 group_start_code와 정합되면, while 문에 의해 정의된 조건은 참이 된다. 그래서, picture_start_code나 group_start_code가 비트스트림에 나타나면, do 문에 이어지는 {} 블록에서 함수로 정의된 데이터 소자의 코드는 이 시작 코드 이후에 기술된다. 따라서, picture_start_code나 group_start_code로 나타내지는 시작 코드를 찾음으로서, 비트스트림으로부터 do 문의 {} 블록에서 함수로 정의된 데이터 소자를 추출하는 것이 가능하다.

do 문의 {} 블록의 시작부에서 기술되는 if 문은 조건 "group_start_code가 비트스트림에 나타나면"을 말한다. if 문에 의해 언급되는 참(만족되는) 조건은 group_of_picture_header(1) 함수와 extension_and_user_data(1) 함수에 의해 정의된 데이터 소자가 group_start_code 이후에 순차적으로 기술됨을 나타낸다.

group_of_picture_header(1) 함수는 MPEG 비트스트림에서 GOP층의 헤더 데이터를 정의하는데 사용되는 함수이고, extension_and_user_data(1) 함수는 MPEG 비트스트림의 GOP층에서 extension_data라 칭하여지는 확장 데이터 및/또는 user_data라 칭하여지는 사용자 데이터를 정의하는데 사용되는 함수이다.

더욱이, 이 비트스트림에서는 picture_header() 함수와 picture_coding_extension() 함수에 의해 정의된 데이터 소자가 group_of_picture_header(1) 함수와 extension_and_user_data(1) 함수에 의해 정의된 데이터 소자 이후에 기술된다. 물론, if 문에 의해 정의된 조건이 참이 아니면, group_of_picture_header(1) 함수와 extension_and_user_data(1) 함수에 의해 정의된 데이터 소자는 기술되지 않는다. 이 경우에는 picture_header() 함수와 picture_coding_extension() 함수에 의해 정의된 데이터 소자가 extension_and_user_data(0) 함수에 의해 정의된 데이터 소자 이후에 기술된다.

picture_header() 함수는 MPEG 스트림의 화상 층에 대한 헤더 데이터를 정의하는데 사용되는 함수이고, picture_coding_extension() 함수는 MPEG 스트림의 화상층에서 먼저 확장 데이터를 정의하는데 사용되는 함수이다.

다음의 while 문은 조건을 정의하는데 사용되는 함수이다. while 문에 의해 정의된 조건에 이어지는 {} 블록에 기술된 각 if 문에 의해 정의된 조건은 while 문에 의해 정의된 조건이 참인 동안 참 또는 거짓임이 판단된다. while 문에서 사용되는 nextbits() 함수는 비트스트림에 나타나는 비트열이 각각 extension_start_code 및 user_start_code와 정합되는가 여부를 판단하기 위한 함수이다. 비트스트림에 나타나는 비트열이 extension_start_code 또는 user_data_start와 정합되면, while 문에 의해 정의된 조건은 참이 된다.

while 문에 이어지는 {} 블록에서 제 1 if 문은 비트스트림에 나타나는 비트열이 extension_start_code와 정합되는가 여부를 판단하기 위한 함수이다. 32-비트 extension_start_code와 정합되는 비트스트림에 나타나는 비트열은 extension_data(2) 함수에 의해 정의된 데이터 소자가 비트스트림에서 extension_start_code 이후에 기술됨을 나타낸다.

제 2 if 문은 비트스트림에 나타나는 비트열이 user_data_start_code와 정합되는가 여부를 판단하기 위한 함수이다. 비트스트림에 나타나는 비트열이 32-비트 user_data_start_code와 정합되면, 제 3 if 문에 의해 정의된 조건이 참 또는 거짓으로 판단된다. user_data_start_code는 MPEG 비트스트림의 화상층에서 사용자 데이터 영역의 시작을 나타내는데 사용되는 시작 코드이다.

while 문에 이어지는 {} 블록에서 제 3 if 문은 비트스트림에 나타나는 비트열이 History_Data_ID와 정합되는가 여부를 판단하기 위한 함수이다. 8-비트 History_Data_ID와 정합되는 비트스트림에 나타나는 비트열은 converted_history_stream() 함수에 의해 정의된 데이터 소자가 MPEG 비트스트림의 화상층 중 사용자 데이터 영역에서 8-비트 History_Data_ID에 의해 나타내지는 코드 이후에 기술됨을 나타낸다.

converted_history_stream() 함수는 MPEG 부호화 처리에서 사용되는 모든 부호화 파라미터를 전송하기 위한 히스토리 정보 및 히스토리 데이터를 기술하는데 사용되는 함수이다. 이 converted_history_stream() 함수에 의해 정의된 데이터 소자의 상세한 내용은 추후 설명된다. History_Data_ID는 MPEG 비트스트림의 화상층 중 사용자 데이터 영역에서 히스토리 정보 및 히스토리 데이터의 기술이 시작됨을 나타내는데 사용되는 시작 코드이다.

else 문은 제 3 if 문에 의해 정의된 거짓(false) 조건의 경우를 나타내는 구문이다. 그래서, converted_history_stream() 함수에 의해 정의된 데이터 소자가 MPEG 비트스트림의 화상층 중 사용자 데이터 영역에 기술되지 않으면, user_data() 함수에 의해 정의된 데이터 소자가 기술된다.

picture_data() 함수는 MPEG 비트스트림에서 화상층의 사용자 데이터 이후에 슬라이스층 및 매크로블록층에 관련된 데이터 소자를 기술하는데 사용되는 함수이다. 일반적으로, 이 picture_data() 함수에 의해 정의된 데이터 소자는 비트스트림의 화상층 중 사용자 데이터 영역에 기술되는 user_data() 함수에 의해 정의된 데이터 소자 또는 converted_history_stream() 함수에 의해 정의된 데이터 소자 이후에 기술된다. 그러나, extension_start_code나 user_data_start_code가 화상층의 데이터 소자를 나타내는 비트스트림에 존재하지 않으면, 이 picture_data() 함수에 의해 정의된 데이터 소자는 picture_coding_extension() 함수에 의해 정의된 데이터 소자 이후에 기술된다.

이 picture_data() 함수에 의해 정의된 데이터 소자 이후에는 sequence_header() 함수 및 sequence_extension() 함수에 의해 정의된 데이터 소자가 순차적으로 기술된다. sequence_header() 함수 및 sequence_extension() 함수에 의해 기술된 데이터 소자는 비디오 스트림 시퀀스의 시작부에서 기술된 sequence_header() 함수 및 sequence_extension() 함수에 의해 정의된 것과 정확히 같은 데이터 소자이다. 스트림에서 같은 데이터 부분이 정의되는 이유는 시퀀스층의 데이터가 더 이상 수신가능하지 않는 것을 방지하기 위한 것이므로, 화상층에 대응한 비트스트림 부분과 같이, 데이터 스트림의 중간에 비트스트림 수신 장치에 의해 비트스트림의 수신이 시작될 때, 스트림이 더 이상 복호가능하지 않는 것을 방지한다.

sequence_header() 함수 및 sequence_extension() 함수에 의해 정의된 데이터 소자 이후, 즉 데이터 스트림의 끝부분에는 시퀀스의 종료를 나타내는데 사용되는 32-비트 sequence_end_code가 기술된다.

도 39는 지금까지 기술된 구문의 기본적인 구성에 대한 개요를 도시하는 도면이다.

다음에는 converted_history_stream() 함수에 의해 정의된 히스토리 스트림이 설명된다.

converted_history_stream() 함수는 히스토리 정보를 나타내는 히스토리 스트림을 MPEG 비트스트림의 화상층 중 사용자 데이터 영역에 삽입하는데 사용되는 함수이다. 'converted'란 말은 스트림이 시작 에뮬레이션(emulation)을 방지하기 위해 사용자 영역에 삽입되는 히스토리 데이터로 구성된 히스토리 스트림에서 적어도 매 22 비트 마다 하나의 마커 비트를 삽입하는 변환 처리를 완료했음을 의미함을 주목하여야 한다.

converted_history_stream() 함수는 추후 기술된 도 40 내지 도 46에 도시된 고정길이 히스토리 스트림 또는 도 47에 도시된 가변길이 히스토리 스트림의 포맷 중 하나로 기술된다. 인코더측에서 고정길이 히스토리 스트림이 선택되면, 히스토리 스트림으로부터 데이터 소자를 복호화하기 위한 디코더에서 사용되는 소프트웨어 및 회로가 간단해지는 이점이 있다. 한편, 인코더측에서 가변길이 히스토리 스트림이 선택되면, 인코더는 필요한 경우 화상층의 사용자 영역에 기술된 데이터 소자 또는 히스토리 정보를 임의적으로 선택할 수 있다. 그래서, 히스토리 스트림의 데이터량이 줄어들 수 있다. 그 결과로, 전체적으로 비트스트림의 데이터 레이트가 또한 더 낮아질 수 있다.

본 발명의 설명에서 언급된 히스토리 정보, 히스토리 데이터, 및 히스토리 파라미터는 관련된 기술의 부호화 처리에서 사용되는 데이터 소자 또는 부호화 파라미터이고, 최종 단계에서 실행되는 부호화 처리 또는 현재 부호화 처리에서 사용되는 부호화 파라미터 데이터가 아니다. 화상이 제 1 세대의 부호화 처리에서는 I-화상으로, 제 2 세대의 부호화 처리에서는 P-화상으로, 또한 제 3 세대의 부호화 처리에서는 B-화상으로 부호화되어 전송되는 경우를 고려해본다. 제 3 세대의 부호화 처리에서 사용되는 부호화 파라미터는 제 3 세대의 부호화 처리 결과로 생성되는 부호화 비트스트림에서 시퀀스, GOP-화상, 슬라이스, 및 매크로블록층의 소정의 위치에 기술된다. 한편, 제 1 및 제 2 세대의 부호화 처리에서 사용되는 부호화 파라미터는 제 3 세대의 부호화 처리에서 사용되는 부호화 파라미터를 기록하는 시퀀스 또는 GOP층에 기록되지 않고, 부호화 파라미터의 히스토리 정보로 화상층의 사용자 데이터 영역에 기록된다.

먼저, 고정길이 히스토리 스트림의 구문이 도 40 내지 도 46을 참고로 설명된다.

우선, 이전 부호화 처리, 즉 전형적으로 제 1 및 제 2 세대의 부호화 처리에서 사용된 시퀀스층의 시퀀스 헤더에 관련된 부호화 파라미터는 최종 단계에서 실행되는 부호화 처리, 즉 전형적으로 제 3 세대의 부호화 처리에서 생성된 비트스트림의 화상층 중 사용자 데이터 영역에 히스토리 스트림으로 삽입된다. 이전 부호화 처리에서 생성된 비트스트림의 시퀀스층 중 시퀀스 헤더에 관련된 히스토리 정보는 최종 단계에서 실행되는 부호화 처리에서 생성된 비트스트림의 시퀀스층 중 시퀀스 헤더로 결코 삽입되지 않음을 주목하여야 한다.

이전 부호화 처리에서 사용된 시퀀스 헤더에 관련되는 데이터 소자는 도 40에 도시된 바와 같이 sequence_header_code, sequence_header_present_flag, horizontal_size_value, vertical_size_value, aspect_ratio_information, frame_rate_code, bit_rate_value, marker_bit, VBV_buffer_size_value, constrained_parameter_flag, load_intra_quantizer_matrix, intra_quantizer_matrix, load_non_intra_quantizer_matrix, 및 non_intra_quantizer_matrix를 포함한다.

상술된 데이터 소자는 다음과 같이 기술된다. sequence_header_code 데이터 소자는 시퀀스층의 시작 동기화 코드이다. sequence_header_present_flag 데이터 소자는 시퀀스 헤더내의 데이터가 유효한가 여부를 나타내는데 사용되는 플래그이다. horizontal_size_value 데이터 소자는 수평 방향으로 화상의 픽셀수 중 하위 12 비트를 포함하는 데이터이다. vertical_size_value 데이터 소자는 수직 방향으로 화상의 픽셀수 중 상위 12 비트를 포함하는 데이터이다. aspect_ratio_information 데이터 소자는 종횡비, 즉 화상의 폭에 대한 높이의 레이트, 또는 디스플레이 스크린의 종횡비이다. frame_rate_code 데이터 소자는 화상 디스플레이 주기를 나타내는 데이터이다.

bit_rate_value 데이터 소자는 생성된 비트의 수를 제한하기 위한 비트 레이트 중 하위 18 비트를 포함하는 데이터이다. 데이터는 400-bsp 단위로 반올림된다. marker_bit 데이터 소자는 시작 코드 에뮬레이션을 방지하기 위해 삽입된 비트 데이터이다. VBV_buffer_size_value 데이터 소자는 생성된 코드의 양을 제어하는데 사용되는 가상 버퍼(비디오 버퍼 검증기)의 크기를 결정하는 값 중 하위 10 비트를 포함하는 데이터이다. constrained_parameter_flag 데이터 소자는 파라미터가 제한되는가 여부를 나타내는데 사용되는 플래그이다. load_intra_quantizer_matrix 데이터 소자는 MR내 양자화 매트릭스 (intra-MB quantization matrix)의 데이터가 존재하는가 여부를 나타내는데 사용되는 플래그이다. intra_quantizer_matrix 데이터 소자는 MB내 양자화 매트릭스의 값이다. load_non_intra_quantizer_matrix 데이터 소자는 비MB내 양자화 매트릭스 (non-intra-MB quantization matrix)의 데이터가 존재하는가 여부를 나타내는데 사용되는 플래그이다. non_intra_quantizer_matrix 데이터 소자는 비MB내 양자화 매트릭스의 값이다.

이어서, 이전 부호화 처리에서 사용된 시퀀스 확장을 나타내는 데이터 소자는 최종 단계에서 실행된 부호화 처리에서 생성되는 비트스트림의 화상층 중 사용자 영역에 히스토리 스트림으로 기술된다.

이전 부호화에서 사용된 시퀀스 확장을 나타내는 데이터 소자는 도 40 및 도 41에 도시된 바와 같이 extension_start_code, extension_start_code_identifier, sequence_extension_present_flag, profile_and_level_identification, progressive_sequence, chroma_format, horizontal_size_extension, vertical_size_extension, bit_rate_extension, vbv_buffer_size_extension, low_delay, frame_rate_extension_n, 및 frame_rate_extension_d를 포함한다.

상술된 데이터 소자는 다음과 같이 기술된다. extension_start_code 데이터 소자는 확장 데이터의 시작 동기화 코드이다. extension_start_code_identifier 데이터 소자는 어느 확장 데이터가 전송되는가를 나타내는데 사용되는 데이터이다. sequence_extension_present_flag 데이터 소자는 시퀀스 확장내의 데이터가 유효한가 여부를 나타내는데 사용되는 플래그이다. profile_and_level_identification 데이터 소자는 비디오 데이터의 레벨과 프로파일을 지정하는 데이터이다. progressive_sequence 데이터 소자는 비디오 데이터가 순차적인 주사로부터 얻어졌음을 나타내는 데이터이다. chroma_format 데이터 소자는 비디오 데이터의 색차 포맷을 지정하는 데이터이다.

horizontal_size_extension 데이터 소자는 시퀀스 헤더의 horizontal_size_value에 부가되는 상위 두 비트의 데이터이다. vertical_size_extension 데이터 소자는 시퀀스 헤더의 vertical_size_value에 부가되는 상위 두 비트의 데이터이다. bit_rate_extension 데이터 소자는 시퀀스 헤드의 bit_rate_value에 부가되는 상위 12 비트의 데이터이다. vbv_buffer_size_extension 데이터 소자는 시퀀스 헤더의 vbv_buffer_size_extension에 부가되는 상위 8 비트의 데이터이다. low_delay 데이터 소자는 B-화상이 포함되지 않음을 나타내는데 사용되는 데이터이다. frame_rate_extension_n 데이터 소자는 시퀀스 헤더의 frame_rate_code와 연관되어 프레임 레이트를 구하는데 사용되는 데이터이다. frame_rate_extension_d 데이터 소자는 시퀀스 헤더의 frame_rate_code와 연관되어 프레임 레이트를 구하는데 사용되는 데이터이다.

이어서, 이전 부호화 처리에서 사용되는 시퀀스층의 시퀀스-디스플레이 확장을 나타내는 데이터 소자는 비트스트림의 화상층 중 사용자 영역에 히스토리 스트림으로 기술된다.

시퀀스-디스플레이 확장으로 기술되는 데이터 소자는 도 41에 도시된 바와 같이 extension_start_code, extension_start_code_identifier, sequence_display_extension_present_flag, video_format, color_decription, color_primaries, transfer_characteristics, matrix_coefficients, display_horizontal_size, 및 display_vertical_size이다.

상술된 데이터 소자는 다음과 같이 기술된다. extension_start_code 데이터 소자는 확장 데이터의 시작 동기화 코드이다. extension_start_code_identifier 데이터 소자는 어느 확장 데이터가 전송되는가를 나타내는데 사용되는 데이터이다. sequence_display_extension_presentation_flag 데이터 소자는 시퀀스 확장내의 데이터 소자가 유효한가 여부를 나타내는데 사용되는 플래그이다. video_format 데이터 소자는 소스 신호의 비디오 포맷을 나타내는 데이터이다. color_description 데이터 소자는 색상 공간의 상세한 데이터가 존재함을 나타내는데 사용되는 데이터이다. color_primaries 데이터 소자는 소스 신호의 색상 특성에 관한 상세한 내용을 도시하는 데이터이다. transfer_characteristics 데이터 소자는 광전기 변환이 실행된 방법에 관한 상세한 내용을 도시하는 데이터이다. matrix_coefficients 데이터 소자는 소스 신호가 빛의 3가지 주요 색상으로부터 변환된 방법에 관한 상세한 내용을 도시하는 데이터이다. display_horizontal_size 데이터 소자는 의도되는 디스플레이의 수평 크기 또는 활성 영역을 나타내는 데이터이다. display_vertical_size 데이터 소자는 의도되는 디스플레이의 수직 크기 또는 활성 영역을 나타내는 데이터이다.

이어서, 이전 부호화 처리에서 생성된 매크로블록의 위상 정보를 도시하는 매크로블록 지정 데이터 (macroblock_assignment_in_user_data라 칭하여지는)는 최종 단계에서 실행된 부호화 처리에서 생성되는 비트스트림의 화상층 중 사용자 영역에 히스토리 스트림으로 기술된다.

매크로블록의 위상 정보를 나타내는 macroblock_assignment_in_user_data는 도 41에 도시된 바와 같이 macroblock_assignment_present_flag, v_phase, 및 h_phase와 같은 데이터 소자를 포함한다.

상술된 데이터 소자는 다음과 같이 기술된다. macroblock_assignment_present_flag 데이터 소자는 macroblock_assignment_in_user_data의 데이터 소자가 유효한가 여부를 나타내는데 사용되는 플래그이다. v_phase 데이터 소자는 매크로블록이 화상 데이터로부터 이탈될 때 구해지는 수직 방향으로의 위상 정보를 도시하는 데이터이다. h_phase 데이터 소자는 매크로블록이 화상 데이터로부터 이탈될 때 구해지는 수평 방향으로의 위상 정보를 도시하는 데이터이다.

이어서, 이전 부호화 처리에서 사용된 GOP층의 GOP 헤더를 나타내는 데이터 소자는 최종 단계에서 실행된 부호화 처리에서 생성되는 비트스트림의 화상층 중 사용자 영역에 히스토리 스트림으로 기술된다.

GOP 헤더를 나타내는 데이터 소자는 도 41에 도시된 바와 같이 group_start_code, group_of_picture_header_present_flag, time_code, closed_gop, 및 broken_link이다.

상술된 데이터 소자는 다음과 같이 기술된다. group_start_code 데이터 소자는 GOP층의 시작 동기화 코드이다. group_of_picture_header_present_flag 데이터 소자는 group_of_picture_header내의 데이터 소자가 유효한가 여부를 나타내는데 사용되는 플래그이다. time_code 데이터 소자는 GOP의 제 1 화상의 시작부로부터 측정된 시간 길이를 나타내는 시간 코드이다. closed_gop 데이터 소자는 또 다른 GOP로부터 한 GOP에서 독립적인 화상의 재생 동작을 실행하는 것이 가능하가 여부를 나타내는데 사용되는 플래그이다. broken_link 데이터 소자는 GOP의 시작부에 있는 B-화상이 편집과 같은 이유 때문에 높은 정확도로 재생될 수 없는가 여부를 나타내는데 사용되는 플래그이다.

이어서, 이전 부호화 처리에서 사용된 화상층의 화상 헤더를 나타내는 데이터 소자는 최종 단계에서 실행된 부호화 처리에서 생성되는 비트스트림의 화상층 중 사용자 영역에 히스토리 스트림으로 기술된다.

화상 헤더에 관련된 데이터 소자는 도 41 및 도 42에 도시된 바와 같이 picture_start_code, temporal_reference, picture_coding_type, vbv_delay, full_pel_forward_vector, forward_f_code, full_pel_backward_vector, 및 backward_f_code이다.

상술된 데이터 소자는 다음과 같이 구체적으로 기술된다. picture_start_code 데이터 소자는 화상층의 시작 동기화 코드이다. temporal_reference 데이터 소자는 화상의 디스플레이 순서를 나타내는데 사용되는 숫자이다. 이 숫자는 GOP의 시작부에서 재설정된다. picture_coding_type 데이터 소자는 화상의 종류를 나타내는데 사용되는 데이터이다. vbv_delay 데이터 소자는 랜덤 억세스에서 가상 버퍼의 초기 상태를 도시하는 데이터이다. full_pel_forward_vector 데이터 소자는 전방 이동 벡터의 정확도가 픽셀 단위 또는 반픽셀 단위(half-pixel units)로 표시되는가 여부를 나타내는데 사용되

는 플래그이다. forward_f_code 데이터 소자는 전방 이동 벡터 탐색 범위를 나타내는 데이터이다. full_pel_backward_vector 데이터 소자는 후방 이동 벡터의 정확도가 픽셀 단위 또는 반픽셀 단위로 표시되는가 여부를 나타내는데 사용되는 플래그이다. backward_f_code 데이터 소자는 후방 이동 벡터 탐색 범위를 나타내는 데이터이다.

이어서, 이전 부호화 처리에서 사용된 화상층의 화상 부호화 확장을 나타내는 데이터 소자는 최종 단계에서 실행된 부호화 처리에서 생성되는 비트스트림의 화상층 중 사용자 영역에 히스토리 스트림으로 기술된다.

화상 부호화 확장에 관련된 데이터 소자는 도 42에 도시된 바와 같이 extension_start_code, extension_start_code_identifier, f_code[0][0], f_code[0][1], f_code[1][0], f_code[1][1], intra_dc_precision, picture_structure, top_field_first, frame_predictive_frame_dct, concealment_motion_vectors, q_scale_type, intra_vlc_format, alternate_scan, repeat_first_field, chroma_420_type, progressive_frame, composite_display_flag, v_axis, field_sequence, sub_carrier, burst_amplitude, 및 sub_carrier_phase이다.

상술된 데이터 소자는 다음과 같이 기술된다. extension_start_code 데이터 소자는 화상층에서 확장 데이터의 시작을 나타내는데 사용되는 시작 코드이다. extension_start_code_identifier 데이터 소자는 어느 확장 데이터가 전송되는가를 나타내는데 사용되는 코드이다. f_code[0][0] 데이터 소자는 전방향에서의 수평적 이동벡터 탐색 범위를 나타내는 데이터이다. f_code[0][1] 데이터 소자는 전방향에서의 수직적 이동벡터 탐색 범위를 나타내는 데이터이다. f_code[1][0] 데이터 소자는 후방향에서의 수평적 이동벡터 탐색 범위를 나타내는 데이터이다. f_code[1][1] 데이터 소자는 후방향에서의 수직적 이동벡터 탐색 범위를 나타내는 데이터이다.

intra_dc_precision 데이터 소자는 DC 계수의 정확도를 나타내는 데이터이다. picture_structure 데이터 소자는 데이터 구조가 프레임 구조 또는 필드 구조인가 여부를 나타내는데 사용되는 데이터이다. 필드 구조의 경우, picture_structure 데이터 소자는 또한 필드 구조가 상위 필드 또는 하위 필드인가 여부를 나타낸다. top_field_first 데이터 소자는 프레임 구조의 제 1 필드가 상위 필드 또는 하위 필드인가 여부를 나타내는데 사용되는 데이터이다. frame_predictive_frame_dct 데이터 소자는 프레임 구조의 경우에 프레임-모드 DCT의 예측이 프레임-DCT 모드에서만 실행됨을 나타내는데 사용되는 데이터이다. concealment_motion_vector 데이터 소자는 매크로블록내 (intra-macroblock)에 전송 에러를 숨기기 위한 이동 벡터가 포함됨을 나타내는데 사용되는 데이터이다.

q_scale_type 데이터 소자는 선형 양자화 스케일 또는 비선형 양자화 스케일을 사용하는가 여부를 나타내는데 사용되는 데이터이다. intra_vlc_format 데이터 소자는 또 다른 2차원 VLC가 매크로블록내에 사용되는가 여부를 나타내는데 사용되는 데이터이다. alternate_scan 데이터 소자는 지그재그 주사 또는 다른 방법의 주사를 사용하기 위한 선택을 나타내는 데이터이다. repeat_first_field 데이터 소자는 2 : 3 풀다운 (pull-down)의 경우에 사용되는 데이터이다. chroma_420_type 데이터 소자는 4 : 2 : 0 신호 포맷의 경우에는 다음 progressive_frame 데이터 소자의 값과 같고 그렇지 않은 경우에는 0과 같은 데이터이다. progressive_frame 데이터 소자는 이 화상이 순차적인 주사로부터 구해졌는가 여부를 나타내는데 사용되는 데이터이다. composite_display_flag 데이터 소자는 소스 신호가 복합 신호인가 여부를 나타내는데 사용되는 플래그이다.

v-axis 데이터 소자는 PAL 소스 신호의 경우에 사용되는 데이터이다. field_sequence 데이터 소자는 PAL 소스 신호의 경우에 사용되는 데이터이다. sub_carrier 데이터 소자는 PAL 소스 신호의 경우에 사용되는 데이터이다. burst_amplitude 데이터 소자는 PAL 소스 신호의 경우에 사용되는 데이터이다. sub_carrier_phase 데이터 소자는 PAL 소스 신호의 경우에 사용되는 데이터이다.

이어서, 이전 부호화 처리에서 사용된 양자화 매트릭스 확장은 최종 단계에서 실행된 부호화 처리에서 생성되는 비트스트림의 화상층 중 사용자 영역에 히스토리 스트림으로 기술된다.

양자화 매트릭스 확장에 관련된 데이터 소자는 도 43에 도시된 바와 같이 extension_start_code, extension_start_code_identifier, quant_matrix__extension_present_flag, load_intra_quantizer_matrix, intra_quantizer_matrix[64], load_non_intra_quantizer_matrix, non_intra_quantizer_matrix[64], load_chroma_intra_quantizer_matrix, chroma_non_intra_quantizer_matrix[64], load_chroma_intra_quantizer_matrix, 및 chroma_non_intra_quantizer_matrix [64]이다.

상술된 데이터 소자는 다음과 같이 기술된다. extension_start_code 데이터 소자는 양자화 매트릭스 확장의 시작을 나타내는데 사용되는 시작 코드이다. extension_start_code_identifier 데이터 소자는 어느 확장 데이터가 전송되는가를 나타내는데 사용되는 코드이다. quant_matrix_extension_present_flag 데이터 소자는 양자화 매트릭스 확장의 데이터 소자

가 유효한가 여부를 나타내는데 사용되는 플래그이다. load_intra_quantizer_matrix 데이터 소자는 매크로블록내 양자화 매트릭스 데이터가 존재하는가 여부를 나타내는데 사용되는 데이터이다. intra_quantizer_matrix 데이터 소자는 매크로블록내 양자화 매트릭스의 값을 나타내는 데이터이다.

load_non_intra_quantizer_matrix 데이터 소자는 비매크로블록내 (non-intra-macroblock) 양자화 매트릭스 데이터가 존재하는가 여부를 나타내는데 사용되는 데이터이다. non_intra_quantizer_matrix 데이터 소자는 비매크로블록내 양자화 매트릭스의 값을 나타내는 데이터이다. load_chroma_intra_quantizer_matrix 데이터 소자는 색차 매크로블록내 양자화 매트릭스 데이터가 존재하는가 여부를 나타내는데 사용되는 데이터이다. chroma_intra_quantizer_matrix 데이터 소자는 색차 매크로블록내 양자화 매트릭스의 값을 나타내는 데이터이다. load_chroma_non_intra_quantizer_matrix 데이터 소자는 색차 비매트로블록내 양자화 매트릭스 데이터가 존재하는가 여부를 나타내는데 사용되는 데이터이다. chroma_non_intra_quantizer_matrix 데이터 소자가 색차 비매크로블록내 양자화 매트릭스의 값을 나타내는 데이터이다.

이어서, 이전 부호화 처리에서 사용된 저작권 확장은 최종 단계에서 실행된 부호화 처리에서 생성되는 비트스트림의 화상층 중 사용자 영역에 히스토리 스트림으로 기술된다.

저작권 확장에 관련된 데이터 소자는 도 43에 도시된 바와 같이 extension_start_code, extension_start_code_identifier, copyright_extension_present_flag, copyright_flag, copyright_identifier, original_or_copy, copyright_number_1, copyright_number_2, 및 copyright_number_3이다.

상술된 데이터 소자는 다음과 같이 기술된다. extension_start_code 데이터 소자는 저작권 확장의 시작을 나타내는데 사용되는 시작 코드이다. extension_start_code_identifier 데이터 소자는 어느 확장 데이터가 전송되는가를 나타내는데 사용되는 코드이다. copyright_extension_present_flag 데이터 소자는 저작권 확장 데이터 소자가 유효한가 여부를 나타내는데 사용되는 플래그이다. copyright_flag 데이터 소자는 저작권이 다음 저작권 확장이나 시퀀스의 종료까지의 범위에서 부호화 비디오 데이터에 주어졌는가 여부를 나타내는데 사용되는 플래그이다.

copyright_identifier 데이터 소자는 ISO/IEC JTC/SC29에 의해 지정된 저작권을 분류하는 제도를 식별하는데 사용되는 데이터이다. original_or_copy 데이터 소자는 비트스트림의 데이터가 원래 데이터인가 복사된 데이터인가 여부를 나타내는데 사용되는 플래그이다. copyright_number_1 데이터 소자는 저작권 번호의 비트 44 내지 63을 나타낸다. copyright_number_2 데이터 소자는 저작권 번호의 비트 22 내지 43을 나타낸다. copyright_number_3 데이터 소자는 저작권 번호의 비트 0 내지 21을 나타낸다.

이어서, 이전 부호화 처리에서 사용된 화상 디스플레이 확장 (picture_display_extension)은 최종 단계에서 실행된 부호화 처리에서 생성되는 비트스트림의 화상층 중 사용자 영역에 히스토리 스트림으로 기술된다.

화상 디스플레이 확장에 관련된 데이터 소자는 도 44에 도시된 바와 같이 extension_start_code, extension_start_code_identifier, picture_display_extension_present_flag, frame_center_horizontal_offset_1, frame_center_vertical_offset_1, frame_center_horizontal_offset_2, frame_center_vertical_offset_2, frame_center_horizontal_offset_3, 및 frame_center_vertical_offset_3이다.

상술된 데이터 소자는 다음과 같이 기술된다. extension_start_code 데이터 소자는 화상 디스플레이 확장의 시작을 나타내는데 사용되는 시작 코드이다. extension_start_code_identifier 데이터 소자는 어느 확장 데이터가 전송되는가를 나타내는데 사용되는 코드이다. picture_display_extension_present_flag 데이터 소자는 화상 디스플레이 확장의 데이터 소자가 유효한가 여부를 나타내는데 사용되는 플래그이다. frame_center_horizontal_offset 데이터 소자는 수평 방향으로 디스플레이 영역의 오프셋이고, frame_center_vertical_offset 데이터 소자는 수직 방향으로 디스플레이 영역의 오프셋이다. 3개까지의 수평 및 수직 오프셋 값이 각각 정의될 수 있다.

사용자 데이터는 도 44에 도시된 바와 같이 최종 단계에서 실행된 부호화 처리에서 생성되는 비트스트림의 화상층 중 사용자 영역에서 이미 설명된 화상 디스플레이 확장을 나타내는 히스토리 정보 이후에 히스토리 스트림으로 기술된다.

사용자 데이터에 이어서, 이전 부호화 처리에서 사용된 매크로블록에 관한 정보가 도 44 내지 도 46에 도시된 바와 같이 히스토리 스트림으로 기술된다.

매크로블록에 관한 정보는 도 44 내지 도 46에 도시된 바와 같이 매크로블록의 위치에 관련된 데이터 소자, 매크로블록의 모드에 관련된 데이터 소자, 양자화 단계의 제어에 관련된 데이터 소자, 이동 보상에 관련된 데이터 소자, 매크로블록의 패

턴에 관련된 데이터 소자, 및 생성된 코드의 양에 관련된 데이터 소자를 포함한다. 매크로블록의 위치에 관련된 데이터 소자는 예를 들면, macroblock_address_h, macroblock_address_v, slice_header_present_flag, 및 skipped_macroblock_flag를 포함한다. 매크로블록의 모드에 관련된 데이터 소자는 예를 들면, macroblock_quant, macroblock_motion_forward, macroblock_motion_backward, macroblock_pattern, macroblock_intra, spatial_temporal_weight_code_flag, frame_motion_type, 및 dct_type을 포함한다. 양자화 단계의 제어에 관련된 데이터 소자는 예를 들면, quantiser_scale_code를 포함한다. 이동 보상에 관련된 데이터 소자는 PMV[0][0][0], PMV[0][0][1], motion_vertical_field_select[0][0], PMV[0][1][0], PMV[0][1][1], motion_vertical_field_select[0][1], PMV[1][0][0], PMV[1][0][1], motion_vertical_field_select[1][0], PMV[1][1][0], PMV[1][1][1], 및 motion_vertical_field_select[1][1]을 포함한다. 매크로블록의 패턴에 관련된 데이터 소자는 예를 들면, coded_block_pattern을 포함하고, 생성된 코드의 양에 관련된 데이터 소자는 num_mv_bits, num_coef_bits, 및 num_other_bits 등이다.

매크로블록에 관련된 데이터 소자는 다음과 같이 상세히 기술된다.

macroblock_address_h 데이터 소자는 수평 방향으로 매크로블록의 현재 절대위치를 정의하는 데이터이다. macroblock_address_v 데이터 소자는 수직 방향으로 매크로블록의 현재 절대위치를 정의하는 데이터이다. slice_header_present_flag 데이터 소자는 슬라이스 헤더와 동반되는가 여부와 이 매크로블록이 슬라이스층의 시작부에 위치함을 나타내는데 사용되는 플래그이다. skipped_macroblock_flag 데이터 소자는 복호화 처리에서 이 매크로블록을 스킵하는가 여부를 나타내는데 사용되는 플래그이다.

macroblock_quant 데이터 소자는 도 65 내지 도 67에 도시된 macroblock_type으로부터 유도되는 데이터이다. 이 데이터 소자는 quantiser_scale_code가 비트스트림에 나타나는가 여부를 나타낸다. macroblock_motion_forward 데이터 소자는 도 65 내지 도 67에 도시된 macroblock_type으로부터 유도되고 복호화 처리에서 사용되는 데이터이다. macroblock_motion_backward 데이터 소자는 도 65 내지 도 67에 도시된 macroblock_type으로부터 유도되고 복호화 처리에서 사용되는 데이터이다. macroblock_pattern 데이터 소자는 도 65 내지 도 67에 도시된 macroblock_type으로부터 유도되고 coded_block_pattern이 비트스트림에 나타나는가 여부를 나타낸다.

macroblock_intra 데이터 소자는 도 65 내지 도 67에 도시된 macroblock_type으로부터 유도되고 복호화 처리에서 사용되는 데이터이다. spatial_temporal_weight_code_flag 데이터 소자는 도 65 내지 도 67에 도시된 macroblock_type으로부터 유도되고 시간 스케일성 (time scalability)을 갖는 하위층 화상의 업 샘플링 (up-sampling) 기술을 도시하는 spatial_temporal_weight_code가 비트스트림에 존재하는가 여부를 나타내는데 사용되는 플래그이다.

frame_motion_type 데이터 소자는 프레임의 매크로블록의 예측 종류를 나타내는데 사용되는 2-비트 코드이다. "00"의 frame_motion_type 값은 2개의 예측 벡터가 있고 예측 종류가 필드 근거의 예측 종류임을 나타낸다. "01"의 frame_motion_type 값은 하나의 예측 벡터가 있고 예측 종류가 필드 근거의 예측 종류임을 나타낸다. "10"의 frame_motion_type 값은 하나의 예측 벡터가 있고 예측 종류가 프레임 근거의 예측 종류임을 나타낸다. "11"의 frame_motion_type 값은 하나의 예측 벡터가 있고 예측 종류가 이중 프라임 (dual-prime) 예측 종류임을 나타낸다. field_motion_type 데이터 소자는 필드의 매크로블록의 이동 예측을 도시하는 2-비트 코드이다. "01"의 field_motion_type 값은 하나의 예측 벡터가 있고 예측 종류가 필드 근거의 예측 종류임을 나타낸다. "10"의 field_motion_type 값은 2개의 예측 벡터가 있고 예측 종류가 18 x 8 매크로블록 근거의 예측 종류임을 나타낸다. "11"의 field_motion_type 값은 하나의 예측 벡터가 있고 예측 종류가 이중 프라임 예측 종류임을 나타낸다. dct_type 데이터 소자는 DCT가 프레임-DCT 모드 또는 필드-DCT 모드에서 실행되는가 여부를 나타내는데 사용되는 데이터이다. quantiser_scale_code 데이터 소자는 매크로블록의 양자화 단계 크기를 나타낸다.

다음은 이동 벡터에 관련된 데이터 요소들을 설명한다. 복호화 처리에 필요한 이동 벡터의 크기를 감소시키기 위하여, 특정 이동 벡터와 이전에 복호화된 이동 벡터 사이의 차이를 실제로 부호화하므로써 특정 이동 벡터가 부호화 처리된다. 이동 벡터를 복호화하기 위한 디코더는 4개의 이동 벡터 예측 값을 유지해야 하는데, 각각의 예측값은 수평 및 수직 성분을 구비한다. 그들 이동 벡터 예측값은 PMV[r][s][v]로 표시된다. 첨자[r]는 매크로블록이 제 1 또는 제 2 벡터 인지의 여부를 나타내기 위해 이용되는 플래그이다. 보다 상세히 설명하면, "0"의 [r] 값은 제 1 벡터를 나타내고, "1"의 [r] 값은 제 2 벡터를 나타낸다. 첨자[s]는 매크로블록 내의 방향이 순방향 또는 역방향 인지의 여부를 나타내기 위해 이용되는 플래그이다. 보다 상세히 설명하면, "0"의 [s] 값은 이동 벡터의 순방향을 나타내고, "1"의 [s] 값은 이동 벡터의 역방향을 나타낸다. 첨자[v]는 매크로블록 내의 이동 벡터의 성분이 수평 또는 수직 방향의 성분 인지의 여부를 나타내기 위해 이용되는 플래그이다. 보다 상세히 설명하면, "0"의 [v] 값은 이동 벡터의 수평 성분을 나타내고, "1"의 [v] 값은 이동 벡터의 수직 성분을 나타낸다.

따라서, PMV[0][0][0]은 제 1 벡터의 순방향 이동 벡터의 수평 성분을 나타내는 데이터이다. PMV[0][0][1]은 제 1 벡터의 순방향 이동 벡터의 수직 성분을 나타내는 데이터이다. PMV[0][1][0]은 제 1 벡터의 역방향 이동 벡터의 수평 성분을 나타내는 데이터이다. PMV[0][1][1]은 제 1 벡터의 역방향 이동 벡터의 수직 성분을 나타내는 데이터이다. PMV[1][0][0]은 제 2 벡터의 순방향 이동 벡터의 수평 성분을 나타내는 데이터이다. PMV[1][0][1]은 제 2 벡터의 순방향 이동 벡터의 수직 성분을 나타내는 데이터이다. PMV[1][1][0]은 제 2 벡터의 역방향 이동 벡터의 수평 성분을 나타내는 데이터이다. PMV[1][1][1]은 제 2 벡터의 역방향 이동 벡터의 수직 성분을 나타내는 데이터이다.

motion_vertical_field_select[r][s]는 예측 포맷의 기준된 필드가 이용됨을 나타내기 위한 데이터이다. 보다 상세히 설명하면, "0"의 motion_vertical_field_select[r][s]-값은 상부 기준된 필드를 나타내고, "10"의 motion_vertical_field_select[r][s]-값은 이용될 하부 기준된 필드를 나타낸다.

motion_vertical_field_select[r][s]에 있어서, 첨자[r]는 매크로블록내의 이동 벡터가 제 1 또는 제 2 벡터 인지의 여부를 나타내기 위해 이용되는 플래그이다. 보다 상세히 설명하면, "0"의 [r] 값은 제 1 벡터를 나타내고, "1"의 [r] 값은 제 2 벡터를 나타낸다. 첨자[s]는 매크로블록내의 이동 벡터의 방향이 순방향 또는 역방향 인지의 여부를 나타내기 위해 이용되는 플래그이다. 보다 상세히 설명하면, "0"의 [s] 값은 이동 벡터의 순방향을 나타내고, "1"의 [s] 값은 이동 벡터의 역방향을 나타낸다. 따라서, motion_vertical_field_select[0][0]은 제 1 벡터의 순방향 이동 벡터의 생성에 이용된 기준 필드를 나타낸다. motion_vertical_field_select[0][1]은 제 1 벡터의 역방향 이동 벡터의 생성에 이용된 기준 필드를 나타낸다. motion_vertical_field_select[1][0]은 제 2 벡터의 순방향 이동 벡터의 생성에 이용된 기준 필드를 나타낸다. motion_vertical_field_select[1][1]은 제 2 벡터의 역방향 이동 벡터의 생성에 이용된 기준 필드를 나타낸다.

coded_block_pattern 데이터 요소는 DCT 계수를 각각 저장하는복수의 DCT 블록 중에 DCT 블록이 의미가 있거나 비-제로 DCT 계수를 포함하는 것을 나타내는데 이용된 가변 길이 데이터이다. num_mv_bits 데이터 요소는 매크로블록내의 이동 벡터의 코드량을 나타내는 데이터이다. num_coef_bits 데이터 요소는 매크로블록내의 DCT 계수의 코드량을 나타내는 데이터이다. 도46에 도시된 num_other_bits 데이터 요소는 이동 벡터 및 DCT 계수 이외의 매크로블록내의 코드량을 나타내는 데이터이다.

다음은 도47 내지 도64를 참조하여 가변 길이를 갖는 히스토리 스트림으로부터 데이터 요소들을 복호화하기 위한 구문을 설명한다.

도47에 도시된 것 처럼, 가변 길이를 갖는 히스토리 스트림은 next_start_code() 함수, sequence_header() 함수, sequence_extension() 함수, extension_and_user_data(0) 함수, group_of_picture_header() 함수, extension_and_user_data(1) 함수, picture_header() 함수, picture_coding_extension() 함수, extension_and_user_data(2) 함수 및, picture_data() 함수로 정의된 데이터 요소들이다.

next_start_code() 함수는 시작 코드에 대한 비트스트림을 검색하기 위해 이용된 함수이기 때문에, sequence_header() 함수에 의해 정의되고 이전의 부호화 처리에 이용된 데이터 소자는 도48에서 도시된 것 처럼 히스토리 스트림의 시작에서 설명된다.

sequence_header() 함수에 의해 정의된 데이터 요소들은, 도48에 도시된 것 처럼, sequence_headers_code, sequence_header_present_flag, horizontal_size_value, vertical_size_value, aspect_ratio_information, frame_rate_code, bit_rate_value, marker_bit, VBV_buffer_size_value, constrained_parameter_flag, load_intra_quantizer_matrix, intra_quantizer_matrix, load_non_intra_quantizer_matrix 및, non_intra_quantizer_matrix를 포함한다.

상기 리스트된 데이터 요소들을 설명한다. sequence_header_code 데이터는 시퀀스층의 시작 동기 코드이다. sequence_header_present_flag 데이터 요소는 sequence_header가 유효 또는 무효인지의 여부를 나타내기 위해 이용된 플래그이다. horizontal_size_value 데이터 요소는 수평 방향으로 화상의 픽셀 수 중 하위차순 12비트를 구비한다. vertical_size_value 데이터 요소는 수직 방향으로 화상의 픽셀 수 중 하위-차순 12비트를 구비한다. aspect_ratio_information 데이터 요소는 한 화상의 종횡비를 나타내는데, 즉 화상폭에 대한 높이의 레이트가나, 디스플레이 스크린의 종횡비를 나타낸다. frame_rate_code 데이터 요소는 화상 표시 주기를 나타내는 데이터이다. bit_rate_value 데이터 요소는 생성된 비트의 수를 제한하는 비트 레이트의 하위-차순 18비트를 구비하는 데이터이다. 이 데이터는 400-bsp 유닛으로 올림이다.

marker_bit 데이터 요소는 시작-코드 에뮬레이션을 방지하기 위해 삽입된 비트 데이터이다. VBV_buffer_size_value 데이터 요소는 생성된 코드량의 제어에 이용된 가상 버터(비디오 버퍼 검공기)의 사이즈를 결정하기 위한 값의 하위-차순 10 비트를 구비하는 데이터이다. constrained_parameter_flag 데이터 요소는 파라미터들이 제한 상태인지의 여부를 나타내기 위해 이용된 플래그이다. load_intra_quantizer_matrix 데이터 요소는 내부-MB 양자화 매트릭스의 데이터가 존재하는지의 여부를 나타내기 위해 이용된 플래그이다. intra_quantizer_matrix 데이터 요소는 내부-MB 양자화 매트릭스의 값이다. load_non_intra_quantizer_matrix 데이터 요소는 비-내부-MB 양자화 매트릭스의 데이터가 존재하는지의 여부를 나타내기 위해 이용되는 플래그이다. non_intra_quantizer_matrix 데이터 요소는 비-내부-MB 양자화 매트릭스의 값이다.

다음은 sequence_header() 함수에 의해 정의된 데이터 요소에 이어서, sequence_extension() 함수에 의해 정의된 데이터 요소들을 도49에 도시된 것과 같은 히스토리 스트림으로서 설명한다.

sequence_extension() 함수에 의해 정의된 데이터 요소들은, 도49에 도시된 것 처럼, extension_start_code, extension_start_code_identifier, sequence_extension_present_flag, profile_and_level_identification, progressive_sequence, chroma_format, horizontal_size_extension, vertical_size_extension, bit_rate_extension, vbv_buffer_size_extension, low_delay, frame_rate_extension_n 및, frame_rate_extension_d를 포함한다.

상기 리스트된 데이터 요소들을 다음에 설명한다. extension_start_code 데이터 요소는 확장 데이터의 시작 동기화 코드이다. extension_start_code_identifier 데이터 요소는 확장 데이터가 전송되는 것을 나타내기 위해 이용된 데이터이다. sequence_extension_present_flag 데이터 요소는 시퀀스 확장의 데이터가 유효 또는 무효인가의 여부를 나타내기 위해 이용된 플래그이다. profile_and_level_identification 데이터 요소는 비디오 데이터의 레벨 및 프로필을 지정하는 데이터이다. progressive_sequence 데이터 요소는 비디오 데이터가 시퀀스 스캐닝으로부터 얻어졌는지를 나타내는 데이터이다. chroma_format 데이터 요소는 비디오 데이터의 색차 포맷을 지정하는 데이터이다. horizontal_size_extension 데이터 요소는 2 상위-차순 비트로서 시퀀스 헤더의 horizontal_size_value에 부가될 데이터이다. vertical_size_extension 데이터 요소는 2 상위-차순 비트로서 시퀀스 헤더의 vertical_size_value에 부가될 데이터이다. bit_rate_extension 데이터 요소는 12 상위-차순 비트로서 시퀀스 헤더의 bit_rate_value에 부가될 데이터이다. vbv_buffer_size_extension 데이터 요소는 8 상위-차순 비트로서 시퀀스 헤더의 vbv_buffer_size_value에 부가될 데이터이다.

low_delay 데이터 요소는 B-화상이 포함되지 않은 것을 나타내기 위해 이용된 데이터이다. frame_rate_extension_n 데이터 요소는 시퀀스 헤더의 frame_rate_code와 일치하여 프레임 레이트를 얻기 위해 이용된 데이터이다. frame_rate_extension_d 데이터 요소는 시퀀스 헤더의 frame_rate_code와 일치하여 프레임 레이트를 얻기 위해 이용된 데이터이다.

sequence_extension() 함수에 의해 정의된 데이터 요소에 이어서, extension_and_user_data(0) 함수에 의해 정의된 데이터 요소를 도50에 도시된 것과 같은 히스토리 스트림으로서 설명한다. 2 이외의 다른 값에 대한 (i)에 대해서, extension_and_user_data(i) 함수는 extension_data() 함수에 의해 정의된 데이터 요소를 설명하는 대신에 히스토리 스트림으로서 user_data()에 의해 정의된 데이터 요소만을 설명한다. 따라서, extension_and_user_data(0) 함수는 히스토리 스트림으로서 user_data() 함수에 의해 정의된 데이터 요소만을 설명한다.

user_data() 함수는 도51에 도시된 것 과 같은 구문에 기초하여 히스토리 스트림으로서 사용자 데이터를 설명한다.

extension_and_user_data(0) 함수에 의해 정의된 데이터 요소에 이어서, 도50에 도시된 extension_and_user_data(1) 함수에 의해 정의된 데이터 요소와 도52에 도시된 group_of_picture_header() 함수에 의해 정의된 데이터 요소를 히스토리 스트림으로서 설명한다. 그러나, extension_and_user_data(1) 함수에 의해 정의된 데이터 요소와 group_of_picture_header() 함수에 의해 정의된 데이터 요소는 GOP 층의 시작 코드가 히스토리 스트림에서 설명된 것을 나타내는 group_stat_code만을 설명하는 것을 주목한다.

도52에 도시된 것 처럼, group_of_picture_header() 함수에 의해 정의된 데이터 요소는 group_start_code, group_of_picture_header_present_flag, time_code, closed_gop 및, broken_link 이다.

상기 리스트된 데이터 요소들을 설명한다. group_start_code 데이터 요소는 GOP 층의 시작 동기화 코드이다. group_of_picture_header_present_flag 데이터 요소는 group_of_picture_header의 데이터 요소가 유효 또는 무효인지의 여부를 나타내기 위해 이용된 플래그이다. time_code 데이터 요소는 GOP의 제 1 화상의 시작으로부터 측정된 시간의 길이를 도

시한 시간 코드이다. closed_gop 데이터 요소는 다른 GOP로부터 GOP의 화상의 독립된 플레이백 동작을 실행할 수 있는지의 여부를 나타내기 위해 이용된 플래그이다. broken_link 데이터 요소는 GOP의 시작에서 B-화상이 편집등과 같은 이유로 고정밀의 정확도로 재생될 수 있는지를 나타내기 위해 이용된 플래그이다.

도50에 도시된 extension_and_user_data(0) 함수와 마찬가지로, extension_and_user_data(1) 함수는 히스토리 스트림으로서 user_data() 함수에 의해 정의된 데이터 요소만을 설명한다.

만일, GOP층의 시작 코드를 표시하는 group_start_code가 히스토리 스트림에 설명되지 않는다면, group_of_picture_header() 함수에 의해 정의된 데이터 요소와 extension_and_user_data(1)에 의해 정의된 데이터 요소는 또한 히스토리 스트림에 설명되지 않는다. 이 경우에, picture_header() 함수에 의해 정의된 데이터 요소는 extension_and_user_data(0) 함수에 의해 정의된 데이터 요소 이후에 설명된다.

picture_header() 함수에 의해 정의된 데이터 요소들은, 도53에 도시된 것 처럼, picture_start_code, temporal_reference, picture_coding_type, vbv_delay, full_pel_forward_vector, forward_f_code, full_pel_backward_vector, backward_f_code, extra_bit_picture 및, extra_information_picture 이다.

상기 리스트된 데이터 요소들을 구체적으로 설명한다. picture_start_code 데이터 요소는 화상층의 시작 동기화 코드이다. temporal_reference 데이터 요소는 화상의 디스플레이 순서를 나타내기 위해 이용된 수 이다. 이 수는 GOP의 시작에서 재설정된다. picture_coding_type 데이터 요소는 화상의 형태를 나타내기 위해 이용된 데이터이다. vbv_delay 데이터 요소는 랜덤 액세스에서 가상 버퍼의 초기 상태를 도시하는 데이터이다. full_pel_forward_vector 데이터 요소는 순방향 이동 벡터의 예측이 절반의 픽셀 유닛 또는 전체의 픽셀 유닛으로 표시되는지의 여부를 나타내는데 이용되는 플래그이다. forward_f_code 데이터 요소는 순방향 이동 벡터 검색 범위를 나타내는 데이터이다. full_pel_backward_vector 데이터 요소는 백워드 이동 벡터의 예측이 절반의 픽셀 유닛 또는 전체의 픽셀 유닛으로 표시되는지의 여부를 나타내는데 이용되는 플래그이다. backward_f_code 데이터 요소는 역방향 이동 벡터 검색 범위를 나타내는 데이터이다. extra_bit_picture 데이터 요소는 다음 부가 정보가 존재하는 지의 여부를 나타내기 위해 이용된 플래그이다. 보다 상세히 설명하면, "0"의 값을 갖는 extra_bit_picture는 다음 부가 정보가 존재하지 않음을 나타내고, "1"의 값을 갖는 extra_bit_picture는 다음 부가 정보가 존재함을 나타낸다. extra_information_picture 데이터 요소는 명세에 의해 보존된 정보이다.

picture_header() 함수에 의해 정의된 데이터 요소들에 이어서, 도54에 도시된 picture_coding_extension() 함수에 의해 정의된 데이터 요소들을 히스토리 스트림으로서 설명한다.

picture_coding_extension() 함수에 의해 정의된 데이터 요소는, 도54에 도시된 것 처럼, extension_start_code, extension_start_code_identifier, f_code[0][0], f_code[0][1], f_code[1][0], f_code[1][1], intra_dc_precision, picture_structure, top_field_first, frame_predictive_frame_dct, concealment_motion_vectors, q_scale_type, intra_vlc_format, alternate_scan, repeat_first_field, chroma_420_type, progressive_frame, composite_display_flag, v_axis, field_sequence, sub_carrier, burst_amplitude 및, sub_carrier_phase 이다.

상기 리스트된 데이터 소자들을 설명한다. extension_start_code 데이터 요소는 화상층의 확장 데이터의 시작을 나타내기 위해 이용된 시작 코드이다. extension_start_code_identifier 데이터 요소는 확장 데이터가 전송되었음을 나타내는데 이용되는 코드이다. f_code[0][0] 데이터 요소는 순방향으로 수평 이동 벡터 검색 범위를 나타내는 데이터이다. f_code[0][1] 데이터 요소는 순방향으로 수직 이동 벡터 검색 범위를 나타내는 데이터이다. f_code[1][0] 데이터 요소는 역방향으로 수평 이동 벡터 검색 범위를 나타내는 데이터이다. f_code[1][1] 데이터 요소는 역방향으로 수직 이동 벡터 검색 범위를 나타내는 데이터이다. intra_dc_precision 데이터 요소는 DC 계수의 예측을 나타내는 데이터이다.

picture_structure 데이터 요소는 데이터 구조가 프레임 구조 또는 필드 구조인가의 여부를 나타내기 위해 이용된 데이터이다. 필드 구조인 경우에, picture_structure 데이터 요소는 또한 필드 구조가 상위-시퀀스 필드 또는 하위-시퀀스 필드인지의 여부를 나타낸다. top_field_first 데이터 요소는 프레임 구조의 제 1 필드가 상위-시퀀스 필드 또는 하위-시퀀스 필드인지의 여부를 나타낸다. frame_predictive_frame_dct 데이터 요소는 프레임-모드 DCT의 예측이 프레임 구조인 경우에 프레임 모드에서만 실행되는 것을 나타내기 위해 이용된 데이터이다. concealment_motion_vectors 데이터 요소는 내부-매크로블록이 전송 에러를 제거하는 이동 벡터를 포함하는 지를 나타내는데 이용되는 데이터이다. q_scale_type 데이터 요소는 선형 양자화 스케일 또는 비선형 양자화 스케일을 이용하는지의 여부를 나타내기 위해 이용된 데이터이다. intra_vlc_format 데이터 요소는 다른 2차원 VLC가 내부-매크로블록에 이용되는지의 여부를 나타내기 위해 이용된 데이터이다.

alternate_scan 데이터 요소는 교호 주사 또는 지그재그 주사를 이용하기 위해 선택을 나타내는 데이터이다. repeat_first_field 데이터 요소는 2:3 풀-다운의 경우에 이용된 데이터이다. chroma_420_type 데이터 요소는 4:2:0 신호 포맷 또는 0 이외의 경우에 next progressive_frame 데이터 요소의 값과 동일한 데이터이다. progressive_frame 데이터 요소는 화상이 시퀀스 스캐닝으로부터 얻어졌는지의 여부를 나타내기 위해 이용된 데이터이다. composite_display_flag 데이터 요소는 소스 신호가 복합 신호인지의 여부를 나타내기 위해 이용된 데이터이다. v_axis 데이터 요소는 PAL 소스 신호의 경우에 이용되는 데이터이다. field_sequence 데이터 요소는 PAL 소스 신호의 경우에 이용되는 데이터이다. sub_carrier 데이터 요소는 PAL 소스 신호의 경우에 이용되는 데이터이다. burst_amplitude 데이터 요소는 PAL 소스 신호의 경우에 이용되는 데이터이다. sub_carrier_phase 데이터 요소는 PAL 소스 신호의 경우에 이용되는 데이터이다.

picture_coding_extension() 함수에 의해 정의된 데이터 요소들에 이어서, 도50에 도시된 extension_and_user_data(2) 함수에 의해 정의된 데이터 요소들을 히스토리 스트림으로서 설명한다. 그러나, extension_data() 함수에 의해 정의된 데이터 요소들은 확장의 시작 코드를 나타내는 extension_start_code가 비트 스트림에 존재하는 경우에만 extension_and_user_data(2)에 의해 설명됨을 주목한다. 부가로, user_data() 함수에 의해 정의된 데이터 요소는 사용자 데이터의 시작 코드를 나타내는 user_data_start_code가 도50에 도시된 것과 같은 비트스트림에 존재하는 경우에만 extension_data() 함수에 의해 정의된 데이터 요소 이후에 extension_and_user_data(2) 함수에 의해 설명된다. 즉, 사용자 데이터의 시작 코드와 확장의 시작 코드 모두가 비트스트림에 존재하지 않은 경우에, extension_data() 함수에 의해 정의된 데이터 요소와 user_data() 함수에 의해 정의된 데이터 요소는 비트스트림에 설명되지 않는다.

extension_data() 함수는 extension_start_code를 나타내는 데이터 요소와, 도55에 도시된 비트스트림의 히스토리 스트림 처럼, quant_matrix_extension() 함수, copyright_extension() 함수 및, picture_display_extension() 함수에 의해 정의된 데이터 요소들을 설명하는데 이용된 함수이다.

quant_matrix_extension() 함수에 의해 정의된 데이터 요소는, 도56에 도시된 것 처럼, extension_start_code, extension_start_code_identifier, quant_matrix_extension_present_flag, load_intra_quantizer_matrix, intra_quantizer_matrix[64], load_non_intra_quantizer_matrix, non_intra_quantizer_matrix[64], load_chroma_intra_quantizer_matrix, chroma_intra_quantizer_matrix[64], load_chroma_non_intra_quantizer_matrix 및, chroma_non_intra_quantizer_matrix[64] 이다

상기 리스트된 데이터 요소들을 설명한다. extension_start_code 데이터 요소는 양자화 매트릭스 확장의 시작을 나타내기 위해 이용된 시작 코드이다. extension_start_code_identifier 데이터 요소는 확장 데이터가 전송되었음을 나타내기 위해 이용된 코드이다. quant_matrix_extension_present_flag 데이터 요소는 양자화 매트릭스 확장의 데이터 요소가 유효 또는 무효인지를 나타내기 위해 이용된 플래그 이다. load_intra_quantizer_matrix 데이터 요소는 내부-매크로블록에 대한 양자화 매트릭스 데이터가 존재하는지의 여부를 나타내기 위해 이용된 데이터이다. intra_quantizer_matrix 데이터 요소는 내부-매크로블록에 대한 양자화 매트릭스의 값을 나타내는 데이터이다.

load_non_intra_quantizer_matrix 데이터 요소는 내부-매크로블록에 대한 양자화 매트릭스 데이터가 존재하는지의 여부를 나타내기 위해 이용된 데이터이다. non_intra_quantizer_matrix 데이터 요소는 내부-매크로블록에 대한 양자화 매트릭스의 값을 나타내는 데이터이다. load_chroma_intra_quantizer_matrix 데이터 요소는 색차 내부-매크로블록에 대한 양자화 매트릭스 데이터가 존재하는지의 여부를 나타내기 위해 이용된 데이터이다. chroma_intra_quantizer_matrix 데이터 요소는 색차 내부-매크로블록에 대한 양자화 매트릭스의 값을 나타내는 데이터이다. load_chroma_non_intra_quantizer_matrix 데이터 요소는 색차 내부-매크로블록에 대한 양자화 매트릭스 데이터가 존재하는지의 여부를 나타내기 위해 이용된 데이터이다. chroma_non_intra_quantizer_matrix 데이터 요소는 색차 내부-매크로블록에 대한 양자화 매트릭스의 값을 나타내는 데이터이다.

copyright_extension() 함수에 의해 정의된 데이터 요소들은, 도57에 도시된 것 처럼, extension_start_code, extension_start_code_identifier, copyright_extension_present_flag, copyright_flag, copyright_identifier, original_or_copy, copyright_number_1, copy_right_number_2 및, copyright_number_3 이다.

상기 리스트된 데이터 요소들을 설명한다. extension_start_code 데이터 요소는 카피라이트 확장의 시작을 나타내기 위해 이용된 시작 코드이다. extension_start_code_identifier 데이터 요소는 확장 데이터가 전송되었는지를 나타내기 위해 이용된 코드이다. copyright_extension_present_flag 데이터 요소는 카피라이트 확장의 데이터 요소가 유효 또는 무효인지의 여부를 나타내기 위해 이용된 플래그이다.

copyright_flag 데이터 요소는 다음 카피라이트 확장까지 또는 시퀀스의 종료까지의 범위에서 부호화된 비디오 데이터에 카피라이트가 제공되었는지의 여부를 나타내기 위해 이용된 플래그이다. copyright_identifier 데이터 요소는 ISO/IEC/JTC/SC29에 의해 지정된 카피라이트를 설정 분류를 나타내기 위해 이용된 데이터이다. original_or_copy 데이터 요소는 비트스트림의 데이터가 원래 또는 카피된 데이터인지의 여부를 나타내기 위해 이용된 플래그이다. copyright_number_1 데이터 요소는 카피라이트 수의 비트 44 내지 63을 나타낸다. copy_right_number_2 데이터 요소는 카피라이트 수의 비트 22 내지 43을 나타낸다. copyright_number_3 데이터 요소는 카피라이트 수의 비트 0 내지 21을 나타낸다.

picture_display_extension() 함수에 의해 정의된 데이터 요소들은, 도58에 도시된 것 처럼, extension_start_code_identifier, frame_center_horizontal_offset 및, frame_center_vertical_offset 이다.

상기 리스트된 데이터 요소들을 설명한다. extension_start_code_identifier 데이터 요소는 확장 데이터가 전송되었는지를 나타내기 위해 이용된 코드이다. frame_center_horizontal_offset 데이터 요소는 수평 방향으로 표시 영역의 옵셋이다. 이와 같은 옵셋의 수는 number_of_frame_center_offset에 의해 정의될 수 있다. frame_center_vertical_offset 데이터 요소는 수직 방향으로 표시 영역의 옵셋이다. 이와 같은 옵셋의 수는 number_of_frame_center_offset에 의해 정의될 수 있다.

도47의 가변 길이 히스토리 스트림에 도시된 것 처럼, picture_data() 함수에 의해 정의된 데이터 소자들을 extension_and_user(2) 함수에 의해 정의된 데이터 소자 이후에 히스토리 스트림으로서 설명한다.

도59에 도시된 것 처럼, picture_data() 함수에 의해 정의된 데이터 요소들은 slice() 함수에 의해 정의된 데이터 요소들이다. slice() 함수에 의해 정의된 데이터 요소들은 slice() 함수의 시작 코드를 나타내는 slice_start_code가 비트스트림에 존재할 수 없는 경우에 비트스트림에 설명되지 않음을 주목한다.

도60에 도시된 것 처럼, slice() 함수는 slice_start_code, slice_quantizer_scale_code, intra_slice_flag, intra_slice, reserved_bits, extra_bit_slice, extra_information_slice 및, extra_bit_slice와 같은 데이터 소자들과 히스토리 스트림으로서 macroblock() 함수에 의해 정의된 데이터 소자들을 설명하기 위해 이용된 함수이다.

상기 리스트된 데이터 소자들을 설명한다. slice_start_code 데이터 요소는 slice() 함수에 의해 정의된 데이터 요소들을 나타내기 위해 이용된 시작 코드이다. slice_quantizer_scale_code 데이터 요소는 슬라이스 층에 존재하는 매크로블록에 정의된 양자화 스텝의 사이즈이다. 그러나, quantizer_scale_code는 quantizer_scale_code가 설정되었을 때 이용되는 것이 바람직하다.

intra_slice_flag 데이터 요소는 intra_slice 및 reserved_bits가 비트 스트림에 존재하는지의 여부를 나타내기 위해 이용된 플래그이다. intra_slice 데이터 요소는 비_내부_매크로블록이 슬라이스층에 존재하는지의 여부를 나타내기 위해 이용된 플래그이다. 보다 상세히 설명하면, 슬라이스층에 매크로블록 중 하나가 비-내부 매크로블록인 경우에, intra_slice 플래그는 "0"의 값을 갖는다. 슬라이스층내의 모든 매크로블록이 비-내부 매크로블록인 경우에, intra_slice 플래그는 "1"의 값을 갖는다, reserved_bits 데이터 요소는 "0"의 값을 갖는 1-비트 데이터이다. extra_bit_slice 데이터 요소는 extra_information_slice 데이터 요소, 즉 히스토리 스트림으로서 부가된 정보가 존재하는지의 여부를 나타내기 위해 이용된 플래그이다. 보다 상세히 설명하면, 다음의 extra_information_slice 데이터 요소가 존재하는 경우에, extra_bit_slice 플래그는 "1"의 값을 갖는다. 반면에, 다음의 extra_information_slice 데이터 요소가 존재하지 않는 경우에, extra_bit_slice 플래그는 "0"의 값을 갖는다.

slice() 함수에 의해 정의된 데이터 요소에 이어서, macroblock() 함수에 의해 정의된 데이터 요소를 히스토리 스트림으로서 설명한다.

도61에 도시된 것 처럼, macroblock() 함수는 macroblock_escape, macroblock_address_increment 및, macroblock_quantizer_scale_code와 같은 데이터 요소와, macroblock_modes() 함수 및 macroblock_vectors(s) 함수에 의해 정의된 데이터 요소들을 정의하기 위해 이용된 함수이다.

상기 리스트된 데이터 요소들을 설명한다. macroblock_escape 데이터 요소는 기준된 매크로블록과 이전의 매크로블록 사이의 수평 방향으로 차이가 적어도 34 또는 보다 크게 되는지의 여부를 나타내기 위해 이용된 일정한 길이를 갖는 비트의 스트링이다. 기준된 매크로블록과 이전의 매크로블록 사이의 수평 방향으로 차이가 적어도 34 또는 보다 크게 된 경우에, macroblock_address_increment 데이터 요소의 값에 33이 부가된다. macroblock_address_increment 데이터 요소

는 기준된 매크로블록과 이전의 매크로블록 사이의 수평 방향으로 차이이다. 하나의 macroblock_escape 데이터 요소가 macroblock_address_increment 데이터 요소 이전에 존재하는 경우에, macroblock_address_increment 데이터 요소의 값에 33을 부가한 결과로서 얻어진 값은 기준된 매크로블록과 이전의 매크로블록 사이의 수평 방향으로 실제 차이를 나타낸다.

macroblock_quantizer_scale_code 데이터 요소는 각각의 매크로블록에서 설정된 양자화 스텝의 사이즈이다. 슬라이스층의 양자화 스텝의 사이즈를 나타내는 slice_quantizer_scale_code 데이터 요소는 또한 각각의 슬라이스층에 설정된다. 그러나, 매크로블록에 설정된 macroblock_scale_code는 slice_quantizer_scale_code 보다 앞선다.

macroblock_address_increment 데이터 요소에 이어서, macroblock_mode() 함수에 의해 정의된 데이터 요소들을 설명한다. 도62에 도시된 것 처럼, macroblock_mode() 함수는 macroblock_type, frame_motion_type, field_motion_type 및 dct_type와 같은 데이터 요소들을 히스토리 스트림으로서 설명하는데 이용된 함수이다.

상기 리스트된 데이터 소자들을 설명한다, macroblock_type 데이터 요소는 매크로블록의 부호화 형태를 나타내는 데이터이다. 구체적으로 설명하면, macroblock_type 데이터 소자는, 도65 내지 도67에 도시된 것 처럼, macroblock_quant, dct_type_flag, macroblock_motion_forward 및 macroblock_motion_backward와 같은 플래그들로부터 생성된 가변 길이를 갖는 데이터이다. macroblock_quant 플래그는 매크로블록에 대한 양자화 스텝의 사이즈를 설정하기 위한 macroblock_quantizer_scale_code가 설정되었는지의 여부를 나타내기 위해 이용된 플래그이다. 만일, macroblock_quantizer_scale_code가 비트스트림에 존재한다면,macroblock_quant 플래그는 "1"의 값을 갖는다.

dct_type 플래그는 기준 매크로블록이 프레임-DCT 모드 또는 필드-DCT 모드에서 부호화되었는지를 나타내는 dct_type이 존재하는지의 여부를 나타내기 위해 이용된 플래그이다. 다시 말하면, dct_type 플래그는 기준 매크로블록이 DCT를 경험했는지의 여부를 나타내기 위해 이용된 플래그이다. dct_type이 비트스트림에 존재하는 경우에, dct_type_flag는 "1"의 값을 갖는다. macroblock_motion_forward는 기준된 매크로블록이 순방향 예측을 경험하는지의 여부를 나타내기 위한 플래그이다. 만일, 기준된 매크로블록이 순방향 예측을 경험한다면, macroblock_motion_forward 플래그는 "1"의 값을 갖는다. 반면에, macroblock_motion_backward 플래그는 기준된 매크로블록이 역방향 예측을 경험하는지의 여부를 나타내기 위한 플래그이다. 만일, 기준된 매크로블록이 역방향 예측을 경험한다면, macroblock_motion_backward 플래그는 "1"의 값을 갖는다.

만일, macroblock_motion_forward 플래그 또는 macroblock_motion_backward 플래그가 "1"의 값을 갖는다면, 화상은 프레임 예측 모드에 전송되고, frame_period_frame_dct는 "0"의 값을 가지며, frame_motion_type을 나타내는 데이터 요소는 macroblock_type을 나타내는 데이터 소자 이후에 설명된다. frame_period_frame_dct는 frame_motion_type가 비트 스트림에 존재하는지 여부를 나타내기 위해 이용된 플래그임을 주목한다.

frame_motion_type 데이터 요소는 프레임의 매크로블록의 예측 형태를 나타내는 2-비트 코드이다. "00"의 frame_motion_type 값은 두 개의 예측 벡터가 존재하고 예측 형태가 필드-기초한 예측 형태임을 나타낸다. "01"의 frame_motion_type 값은 하나의 예측 벡터가 존재하고 예측 형태가 필드-기초한 예측 형태임을 나타낸다. "10"의 frame_motion_type 값은 하나의 예측 벡터가 존재하고 예측 형태가 프레임-기초한 예측 형태임을 나타낸다. "11"의 frame_motion_type 값은 하나의 예측 벡터가 존재하고 예측 형태가 듀얼-프라임 예측 형태임을 나타낸다.

만일, macroblock_motion_forward 플래그 또는 macroblock_motion_backward 플래그가 "1"의 값을 갖고, 화상이 프레임 예측 모드가 아닌 모드에 전송된다면, frame_motion_type을 나타내는 데이터 요소는 macroblock_type을 나타내는 데이터 소자 이후에 설명된다.

field_motion_type 데이터 요소는 필드의 매크로블록의 이동 예측을 나타내는 2-비트 코드이다. "01"의 field_motion_type 값은 하나의 예측 벡터가 존재하고 예측 형태가 필드-기초한 예측 형태임을 나타낸다. "10"의 field_motion_type 값은 두개의 예측 벡터가 존재하고 예측 형태가 18 x 8 매크로블록-기초한 예측 형태임을 나타낸다. "11"의 field_motion_type 값은 하나의 예측 벡터가 존재하고 예측 형태가 듀얼-프라임 예측 형태임을 나타낸다.

만일, 화상이 프레임 예측 모드에서 전송된다면, frame_period_frame_dct는 frame_motion_type이 비트스트림에 존재함을 나타내고, 또한, frame_period_frame_dct는 dct_type이 비트스트림에 존재함을 나타내며, dct_type을 나타내는 데이터 요소는 macroblock_type을 나타내는 데이터 요소 이후에 설명된다. dct_type 데이터 요소는 DCT가 프레임-DCT 모드 또는 필드-DCT 모드에서 실행되는 것을 나타내기 위해 이용된 데이터이다.

도61에 도시된 것 처럼, 만일, 기준된 매크로블록이 순방향 예측 매크로블록 또는 컨실 처리를 완료하는 내부-매크로블록인 경우에, motion_vectors(0) 함수에 의해 정의된 데이터 요소는 설명된다. 만일, 기준된 매크로블록이 역방향 예측 매크로블록인 경우에, motion_vectors(1) 함수에 의해 정의된 데이터 요소는 설명된다. motion_vectors(0) 함수는 제 1 이동 벡터에 관련된 데이터 요소를 설명하기 위해 이용된 함수이고, motion_vectors(1) 함수는 제 2 이동 벡터에 관련된 데이터 요소를 설명하기 위해 이용된 함수이다.

도63에 도시된 것 처럼, motion_vectors 함수는 이동 벡터에 관련된 데이터 요소를 설명하기 위해 이용된 함수이다.

만일, 하나의 이동 벡터가 존재하고 듀얼-프라임 예측 모드가 이용되지 않는다면, motion_vertical_field_select[0][s] 및 motion_vector[0][s]에 의해 정의된 데이터 요소가 설명된다.

motion_vertical_field_select[r][s]는 순방향 예측 또는 역방향 예측 벡터인 제 1 벡터가 하부 필드 또는 상부 필드를 참조하여 형성된 벡터임을 나타내기 위해 이용된 플래그이다. 첨자[r]는 제 1 또는 제 2 벡터를 나타내는 반면에, 첨자[s]는 순방향 예측 또는 역방향 예측 벡터를 나타낸다.

도64에 도시된 것 처럼, motion_vector(r, s) 함수는 motion_code[r][s][t]에 관련된 데이터 어레이, motion_residual[r][s][t]에 관련된 데이터 어레이 및, dmvector[t]를 표시하는 데이터를 설명하기 위해 이용된 함수이다.

motion_code[r][s][t]는 -16 내지 + 16 범위의 값에 의해 이동 벡터의 크기를 나타내기 위해 이용된 가변 길이를 갖는 데이터이다. motion_residual[r][s][t]는 이동 벡터의 나머지를 나타내기 위해 이용된 가변 길이를 갖는 데이터이다. 따라서, motion_code[r][s][t] 및 motion_residual[r][s][t]를 이용하므로써, 상세한 이동 벡터가 설명될 수 있다. dmvector[t]는 듀얼-프라임 예측 모드에서 상부 및 하부 필드 중 한 필드(예를 들어, 상부 필드)내의 이동 벡터를 형성하기 위하여 임의 시간 거리를 갖는 현재의 이동 벡터를 스케일하고, 하부 및 상부 필드의 선들 사이의 수직 방향으로 시프트를 반영하기 위하여 수직 방향으로 보정을 하기 위해 이용된 데이터이다. 첨자[r]는 제 1 또는 제 2 벡터를 나타내는 반면에, 첨자[s]는 순방향 예측 또는 역방향 예측 벡터를 나타낸다. 첨자[t]는 이동 벡터가 수직 또는 수평 방향의 성분임을 나타낸다.

우선, motion_vector(r, s) 함수는 도64에 도시된 것 처럼 히스토리 스트림으로서 수평 방향으로 motion_code[r][s][0]를 나타내는 데이터 어레이를 설명한다. motion_residual[0][s][t] 및 motion_residual[1][s][t] 모두의 비트 수는 f_code[s][t]에 의해 표시된다. 따라서, "1" 이외의 f_code[s][t]의 값은 motion_residual[r][s][t]가 비트스트림에 존재함을 나타낸다. "1"이 아닌 수평 방향 성분의 motion_residual[r][s][0]과 "0"이 아닌 수평 방향 성분의 motion_code[r][s][0]은 motion_residual[r][s][0]를 나타내는 데이터 요소가 비트 스트림에 포함되고, 이동 벡터의 수평 방향 성분은 존재한다. 이 경우에, 수평 성분의 motion_residual[r][s][0]을 나타내는 데이터 요소가 설명된다.

다음에, motion_code[r][s][1]를 나타내는 데이터 어레이는 히스토리 스트림으로서 수직 방향으로 설명된다. 마찬가지로, motion_residual[0][s][t] 및 motion_residual[1][s][t] 모두의 수는 f_code[s][t]에 의해 표시된다. 따라서, "1" 이외의 f_code[s][t]의 값은 motion_residual[r][s][t]가 비트스트림에 존재함을 나타낸다. "1"이 아닌 수직 방향 성분의 motion_residual[r][s][1]과 "0"이 아닌 수직 방향 성분의 motion_code[r][s][1]은 motion_residual[r][s][1]를 나타내는 데이터 요소가 비트 스트림에 포함되고, 이동 벡터의 수직 방향 성분은 존재한다. 이 경우에, 수평 성분의 motion_residual[r][s][1]을 나타내는 데이터 요소가 설명된다.

가변 길이 포맷에 있어서, 히스토리 정보는 전송된 비트의 전송율을 감소시키기 위하여 생략될 수 있음을 주목한다.

예를 들어, macroblock_type 및 motion_vectors()를 전송하지만, quantizer_scale_code를 전송하지 않기 위하여, slice_quantizer_scale_cod는 비트 율을 감소시키기 위해 "00000"으로 설정된다.

또한, 단지 macroblock_type만을 전송하면서 motion_vectors(), quantizer_scale_code 및 dct_type을 전송하지 않기 위해서, "not_coded"는 비트 레이트를 감소시키기 위해 macroblock_type으로 이용된다.

또한, picture_coding_type만을 전송하고 slice() 다음의 모든 정보를 전송하지 않게 하기 위해서는 slice_start_code를 갖는 picture_data()는 비트 레이트를 감소시키기 위해 이동된다.

상술한 것 처럼, user_data에서 0의 23 연속된 비트가 나타나는 것을 방지하기 위하여, "1" 비트는 모든 22 비트에 삽입된다. 그러나, "1" 비트는 또한 22 보다 작은 각각의 비트 수에 삽입될 수 있음을 주목한다. 또한, "1" 비트의 삽입을 대신하여 연속된 0 비트의 수를 카운트함으로써, "1" 비트는 Byte_allign을 검사에 의해 삽입될 수도 있다.

또한, MPEG에 있어서, 0의 비트 23 연속된 비트의 생성은 금지된다. 그러나, 실제로, 한 바이트의 시작으로부터 개시되는 23 비트의 시퀀스가 문제가 된다. 즉, 한 바이트의 시작으로부터 개시하지 않는 그와 같은 23 비트의 시퀀스는 문제가 되지 않는다. 따라서, "1" 비트가 LSB 이외의 다른 위치에서 각각 전형적으로 24 비트에 삽입될 수 있다.

또한, 히스토리 정보가 상기 기술한 것 처럼 비디오 요소에 가까운 포맷으로 형성되었지만, 히스토리 정보는 패킷화된 기본 스트림 또는 전송 스트림에 가까운 포맷으로 형성될 수도 있다. 또한, 심지어 Elementary Stream 의 user_data가 상술한 것에 따라 picture_data의 전면에 배치되었지만, user_data는 또한 다른 위치에 배치될 수 있다.

상술한 처리의 일부를 실행하기 위한 컴퓨터에 의해 실행될 프로그램은 자기 디스크 및 CD-ROM과 같은 정보 기록 매체에 의해 구현되는 프리젠테이션 메체에 더하여 인터넷 또는 디지탈 위성과 같은 네트워크 프리젠테이션 메체를 통해 사용자에 제공될 수 있음을 주목한다.

발명의 효과

본 발명은 비트스트림의 비트 레이트 및 GOP 구조를 변환시키기 위해 MPEG 규격을 근거로 한 부호화 처리가 완료된 비트스트림에 부호화 및 복호화 처리가 반복적으로 처리되는 경우라도 화상 품질의 열화를 일으키지 않는 트랜스코딩 시스템, 비디오 부호화 장치, 스트림 처리 시스템, 및 비디오 복호화 장치를 제공하는 것이다. 본 발명은 복호화 및 부호화 처리가 반복적으로 처리되는 경우라도 화상 품질의 열화를 일으키지 않는 트랜스코딩 시스템, 비디오 부호화 장치, 스트림 처리 시스템, 및 비디오 복호화 장치를 제공하는 것이다.

(57) 청구의 범위

청구항 1.

부호화 스트림을 재부호화 처리하는 재부호화 장치에 있어서,

과거의 부호화 처리에서 이용된 픽처 타입마다의 이력 부호화 파라미터를 상기 부호화 스트림과 함께 입력하는 입력 수단과,

상기 입력 수단에 의해 입력된 상기 부호화 스트림을 복호처리하여 화상 데이터를 생성함과 함께, 상기 입력 수단에 의해 입력된 상기 이력 부호화 파라미터를 상기 화상 데이터와 함께 출력하는 복호 수단과,

과거의 부호화 처리에서 픽처 타입을 변경하였을 때의 부호화 파라미터가 상기 복호 수단에 의해 출력된 상기 이력 부호화 파라미터 중에 존재하는 것을 판정하는 이력 부호화 파라미터 판정 수단과,

상기 이력 부호화 파라미터 판정 수단의 판정 결과에 기초하여, 상기 복호 수단에 의해 생성된 상기 화상 데이터를 재부호화 처리할 때의 픽처 타입을 판정하는 픽처 타입 판정 수단과,

상기 복호 수단에 의해 출력된 상기 화상 데이터를 상기 픽처 타입 판정 수단에 의해 판정된 픽처 타입으로 재부호화 처리하여, 재부호화 스트림을 생성하는 재부호화 수단과,

상기 픽처 타입 판정 수단에 의해 판정된 픽처 타입에 기초하여, 상기 복호 수단에 의해 출력된 상기 이력 부호화 파라미터를 선택적으로 재이용하여 재부호화 처리하도록, 상기 재부호화 수단을 제어하는 제어 수단을 구비하는, 재부호화 장치.

청구항 2.

제 1 항에 있어서, 상기 제어 수단은 상기 픽처 타입 판정 수단에 의해 판정된 픽처 타입이 P 픽처라고 판정된 경우에, 과거의 부호화 처리에서 픽처 타입을 P 픽처로 변경하였을 때의 상기 이력 부호화 파라미터를 재이용하여 재부호화 처리하도록, 상기 재부호화 수단을 제어하는, 재부호화 장치.

청구항 3.

제 2 항에 있어서, 상기 부호화 파라미터는 움직임 벡터이고,

상기 제어 수단은 과거의 부호화 처리에서 픽처 타입을 P 픽처로 변경하였을 때의 상기 움직임 벡터를 선택적으로 재이용함으로써, B 픽처로 재부호화 처리하도록, 상기 재부호화 수단을 제어하는, 재부호화 장치.

청구항 4.

제 1 항에 있어서, 상기 제어 수단은 상기 픽처 타입 판정 수단에 의해 판정된 픽처 타입이 B 픽처라고 판정된 경우에, 과거의 부호화 처리에서 픽처 타입을 B 픽처로 변경하였을 때의 상기 이력 부호화 파라미터를 재이용하여 재부호화 처리하도록, 상기 재부호화 수단을 제어하는, 재부호화 장치.

청구항 5.

제 4 항에 있어서, 상기 부호화 파라미터는 움직임 벡터이고,

상기 제어 수단은 과거의 부호화 처리에서 픽처 타입을 B 픽처로 변경하였을 때의 상기 움직임 벡터를 선택적으로 재이용함으로써, P 픽처로 재부호화 처리하도록, 상기 재부호화 수단을 제어하는, 재부호화 장치.

청구항 6.

제 1 항에 있어서, 상기 재부호화 수단에 의해 생성된 재부호화 스트림을 출력하는 출력 수단을 더 구비하는, 재부호화 장치.

청구항 7.

제 6 항에 있어서, 상기 출력 수단은 상기 제어 수단에 의해 재이용된 상기 이력 부호화 파라미터를 상기 재부호화 수단에 의해 생성된 상기 재부호화 스트림과 함께 더 출력하는, 재부호화 장치.

청구항 8.

제 6 항에 있어서, 상기 출력 수단은 상기 복호 수단에 의해 출력된 상기 이력 부호화 파라미터 중, 상기 제어 수단에 의해 재이용되지 않은 이력 부호화 파라미터를 상기 재부호화 수단에 의해 생성된 상기 재부호화 스트림과 함께 더 출력하는, 재부호화 장치.

청구항 9.

제 1 항에 있어서, 상기 복호 수단은 상기 입력 수단에 의해 입력된 상기 이력 부호화 파라미터를 상기 복호 수단에 의해 생성된 상기 화상 데이터로 다중화하여 출력하고,

상기 제어 수단은 상기 복호 수단에 의해 출력된 상기 화상 데이터로 다중화된 상기 이력 부호화 파라미터를 선택적으로 재이용하여 재부호화 처리하도록, 상기 재부호화 수단을 제어하는, 재부호화 장치.

청구항 10.

제 1 항에 있어서, 상기 재부호화 수단은 상기 입력 수단에 의해 입력된 상기 부호화 스트림의 비트 레이트, GOP 구조 중 적어도 어느 하나를 변경하도록 재부호화 처리를 실행하는, 재부호화 장치.

청구항 11.

제 1 항에 있어서, 상기 재부호화 수단은 MPEG 규격에 준하여 상기 재부호화 스트림을 생성하는, 재부호화 장치.

청구항 12.

부호화 스트림을 재부호화 처리하는 재부호화 장치의 재부호화 방법에 있어서,

과거의 부호화 처리에서 이용된 픽처 타입마다의 이력 부호화 파라미터를, 상기 부호화 스트림과 함께 입력하는 입력 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 부호화 스트림을 복호처리하여 화상 데이터를 생성하는 복호 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 이력 부호화 파라미터를 상기 화상 데이터와 함께 출력하는 출력 단계와,

과거의 부호화 처리에서 픽처 타입을 변경하였을 때의 부호화 파라미터가 상기 출력 단계의 처리에 의해 출력된 상기 이력 부호화 파라미터 중에 존재하는 것을 판정하는 이력 부호화 파라미터 판정 단계와,

상기 이력 부호화 파라미터 판정 단계의 처리에 의한 판정 결과에 기초하여, 상기 복호 단계의 처리에 의해 생성된 상기 화상 데이터를 재부호화 처리할 때의 픽처 타입을 판정하는 픽처 타입 판정 단계와,

상기 출력 단계의 처리에 의해 출력된 상기 화상 데이터를 상기 픽처 타입 판정 단계의 처리에 의해 판정된 픽처 타입으로 재부호화 처리하여, 재부호화 스트림을 생성하는 재부호화 단계와,

상기 픽처 타입 판정 단계의 처리에 의해 판정된 픽처 타입에 기초하여, 상기 출력 단계의 처리에 의해 출력된 상기 이력 부호화 파라미터를 선택적으로 재이용하여 재부호화 처리하도록, 상기 재부호화 단계에서의 처리를 제어하는 제어 단계를 포함하는, 재부호화 방법.

청구항 13.

화상 데이터를 부호화 처리하는 부호화 장치에 있어서,

과거의 부호화 처리에서 이용된 픽처 타입마다의 이력 부호화 파라미터를 상기 화상 데이터와 함께 입력하는 입력 수단과,

과거의 부호화 처리에서 픽처 타입을 변경하였을 때의 부호화 파라미터가 상기 입력 수단에 의해 입력된 상기 이력 부호화 파라미터 중에 존재하는지를 판정하는 이력 부호화 파라미터 판정 수단과,

상기 이력 부호화 파라미터 판정 수단의 판정 결과에 기초하여, 상기 입력 수단에 의해 입력된 상기 화상 데이터를 부호화 처리할 때의 픽처 타입을 판정하는 픽처 타입 판정 수단과,

상기 입력 수단에 의해 입력된 상기 화상 데이터를 상기 픽처 타입 판정 수단에 의해 판정된 픽처 타입으로 부호화 처리하여, 부호화 스트림을 생성하는 부호화 수단과,

상기 픽처 타입 판정 수단에 의해 판정된 픽처 타입에 기초하여, 상기 입력 수단에 의해 입력된 상기 이력 부호화 파라미터를 선택적으로 재이용하여 부호화 처리하도록, 상기 부호화 수단을 제어하는 제어 수단을 구비하는, 부호화 장치.

청구항 14.

화상 데이터를 부호화 처리하는 부호화 장치의 부호화 방법에 있어서,

과거의 부호화 처리에서 이용된 픽처 타입마다의 이력 부호화 파라미터를 상기 화상 데이터와 함께 입력하는 입력 단계와,

과거의 부호화 처리에서 픽처 타입을 변경하였을 때의 부호화 파라미터가 상기 입력 단계의 처리에 의해 입력된 상기 이력 부호화 파라미터 중에 존재하는지를 판정하는 이력 부호화 파라미터 판정 단계와,

상기 이력 부호화 파라미터 판정 단계의 판정 결과에 기초하여, 상기 입력 단계의 처리에 의해 입력된 상기 화상 데이터를 부호화 처리할 때의 픽처 타입을 판정하는 픽처 타입 판정 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 화상 데이터를 상기 픽처 타입 판정 단계의 처리에 의해 판정된 픽처 타입으로 부호화 처리하여, 부호화 스트림을 생성하는 부호화 단계와,

상기 픽처 타입 판정 단계의 처리에 의해 판정된 픽처 타입에 기초하여, 상기 입력 단계의 처리에 의해 입력된 상기 이력 부호화 파라미터를 선택적으로 재이용하여 부호화 처리하도록, 상기 부호화 단계를 제어하는 제어 단계를 포함하는, 부호화 방법.

청구항 15.

부호화 스트림을 재부호화 처리하는 재부호화 장치에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 부호화 스트림과 함께 입력하는 입력 수단과,

상기 입력 수단에 의해 입력된 상기 부호화 스트림을 복호처리하여 화상 데이터를 생성함과 함께, 상기 입력 수단에 의해 입력된 상기 양자화 스케일을, 상기 화상 데이터와 함께 출력하는 복호 수단과,

상기 복호 수단에 의해 출력된 상기 화상 데이터를 재부호화 처리하여 재부호화 스트림을 생성하는 재부호화 수단과,

상기 복호 수단에 의해 출력된 상기 양자화 스케일 중에서 재부호화 처리에서 재이용하는 양자화 스케일을 선택하여 상기 재부호화 스트림을 생성하도록, 상기 재부호화 수단의 재부호화 처리를 제어하는 제어 수단을 구비하는, 재부호화 장치.

청구항 16.

제 15 항에 있어서, 상기 제어 수단은 상기 입력 수단에 의해 입력된 상기 양자화 스케일 중에서, 가장 큰 양자화 스케일을 선택하는, 재부호화 장치.

청구항 17.

제 15 항에 있어서, 상기 재부호화 수단에 의해 생성된 상기 재부호화 스트림을 출력하는 출력 수단을 더 구비하는, 재부호화 장치.

청구항 18.

제 17 항에 있어서, 상기 출력 수단은 상기 제어 수단에 의해 선택된 상기 양자화 스케일을, 상기 재부호화 수단에 의해 생성된 상기 재부호화 스트림과 함께 더 출력하는, 재부호화 장치.

청구항 19.

제 17 항에 있어서, 상기 출력 수단은 상기 복호 수단에 의해 출력된 상기 양자화 스케일 중, 상기 제어 수단에 의해 선택되지 않은 양자화 스케일을, 상기 재부호화 수단에 의해 생성된 상기 재부호화 스트림과 함께 더 출력하는, 재부호화 장치.

청구항 20.

제 15 항에 있어서, 상기 복호 수단은 상기 입력 수단에 의해 입력된 상기 양자화 스케일을, 상기 복호 수단에 의해 생성된 상기 화상 데이터로 다중화하여 출력하고,

상기 제어 수단은 상기 복호 수단에 의해 출력된 상기 화상 데이터로 다중화된 상기 양자화 스케일로부터, 상기 재부호화 처리에서 재이용하는 상기 양자화 스케일을 선택하는, 재부호화 장치.

청구항 21.

제 15 항에 있어서, 상기 재부호화 수단은 상기 입력 수단에 의해 입력된 상기 부호화 스트림의 비트 레이트, GOP 구조 중의 적어도 어느 하나를 변경하도록 재부호화 처리를 실행하는, 재부호화 장치.

청구항 22.

제 15 항에 있어서, 상기 재부호화 수단은 MPEG 규격에 준하여 상기 재부호화 스트림을 생성하는, 재부호화 장치.

청구항 23.

부호화 스트림을 재부호화 처리하는 재부호화 장치의 재부호화 방법에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 부호화 스트림과 함께 입력하는 입력 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 부호화 스트림을 복호처리하여 화상 데이터를 생성하는 복호 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 양자화 스케일을, 상기 화상 데이터와 함께 출력하는 출력 단계와,

상기 출력 단계의 처리에 의해 출력된 상기 화상 데이터를 재부호화 처리하여 재부호화 스트림을 생성하는 재부호화 단계와,

상기 출력 단계에 의해 출력된 상기 양자화 스케일 중에서 재부호화 처리에서 재이용하는 양자화 스케일을 선택하여 상기 재부호화 스트림을 생성하도록, 상기 재부호화 단계에서의 재부호화 처리를 제어하는 제어 단계를 포함하는, 재부호화 방법.

청구항 24.

부호화 스트림을 재부호화 처리하는 재부호화 장치에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 부호화 스트림과 함께 입력하는 입력 수단과,

상기 입력 수단에 의해 입력된 상기 부호화 스트림을 복호처리하여 화상 데이터를 생성함과 함께, 상기 입력 수단에 의해 입력된 상기 양자화 스케일을, 상기 화상 데이터와 함께 출력하는 복호 수단과,

상기 복호 수단에 의해 출력된 상기 화상 데이터를 재부호화 처리하여 재부호화 스트림을 생성하는 재부호화 수단과,

현재의 부호화 처리에서 양자화 스케일을 산출함과 함께, 상기 복호 수단에 의해 출력된 양자화 스케일 및 상기 산출된 양자화 스케일 중에서, 재부호화 처리에서 재이용하는 양자화 스케일을 선택하여, 상기 재부호화 스트림을 생성하도록, 상기 부호화 수단의 부호화 처리를 제어하는 제어 수단을 구비하는, 재부호화 장치.

청구항 25.

제 24 항에 있어서, 상기 제어 수단은 상기 입력 수단에 의해 입력된 상기 양자화 스케일 및 상기 산출된 양자화 스케일 중에서, 가장 큰 양자화 스케일을 선택하는, 재부호화 장치.

청구항 26.

부호화 스트림을 재부호화 처리하는 재부호화 장치의 재부호화 방법에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 부호화 스트림과 함께 입력하는 입력 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 부호화 스트림을 복호처리하여 화상 데이터를 생성하는 복호 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 양자화 스케일을, 상기 화상 데이터와 함께 출력하는 출력 단계와,

상기 출력 단계의 처리에 의해 출력된 상기 화상 데이터를 재부호화 처리하여 재부호화 스트림을 생성하는 재부호화 단계와,

현재의 부호화 처리에서 양자화 스케일을 산출함과 함께, 상기 출력 단계의 처리에 의해 출력된 상기 양자화 스케일 및 상기 산출된 양자화 스케일 중에서, 재부호화 처리에서 재이용하는 양자화 스케일을 선택하여, 상기 재부호화 스트림을 생성하도록, 상기 부호화 단계에서의 부호화 처리를 제어하는 제어 단계를 포함하는, 재부호화 방법.

청구항 27.

화상 데이터를 부호화 처리하는 부호화 장치에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 화상 데이터와 함께 입력하는 입력 수단과,

상기 입력 수단에 의해 입력된 상기 화상 데이터를 부호화 처리하여 부호화 스트림을 생성하는 부호화 수단과,

상기 입력 수단에 의해 입력된 상기 양자화 스케일 중에서, 상기 부호화 처리에서 재이용하는 양자화 스케일을 선택하여, 상기 부호화 스트림을 생성하도록, 상기 부호화 수단의 부호화 처리를 제어하는 제어 수단을 구비하는, 부호화 장치.

청구항 28.

화상 데이터를 부호화 처리하는 부호화 장치의 부호화 방법에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 화상 데이터와 함께 입력하는 입력 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 화상 데이터를 부호화 처리하여 부호화 스트림을 생성하는 부호화 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 양자화 스케일 중에서, 상기 부호화 처리에서 재이용하는 양자화 스케일을 선택하여, 상기 부호화 스트림을 생성하도록, 상기 부호화 단계에서의 부호화 처리를 제어하는 제어 단계를 포함하는, 부호화 방법.

청구항 29.

화상 데이터를 부호화 처리하는 부호화 장치에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 화상 데이터와 함께 입력하는 입력 수단과,

상기 입력 수단에 의해 입력된 상기 화상 데이터를 부호화 처리하여 부호화 스트림을 생성하는 부호화 수단과,

현재의 부호화 처리에서 양자화 스케일을 산출함과 함께, 상기 입력 수단에 의해 입력된 상기 양자화 스케일 및 상기 산출된 양자화 스케일 중에서, 상기 부호화 처리에서 재이용하는 양자화 스케일을 선택하여, 상기 부호화 스트림을 생성하도록, 상기 부호화 수단의 부호화 처리를 제어하는 제어 수단을 구비하는, 부호화 장치.

청구항 30.

화상 데이터를 부호화 처리하는 부호화 장치의 부호화 방법에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 화상 데이터와 함께 입력하는 입력 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 화상 데이터를 부호화 처리하여 부호화 스트림을 생성하는 부호화 단계와,

현재의 부호화 처리에서 양자화 스케일을 산출함과 함께, 상기 입력 단계의 처리에 의해 입력된 상기 양자화 스케일 및 상기 산출된 양자화 스케일 중에서, 상기 부호화 처리에서 재이용하는 양자화 스케일을 선택하여, 상기 부호화 스트림을 생성하도록, 상기 부호화 단계에서의 부호화 처리를 제어하는 제어 단계를 포함하는, 부호화 방법.

청구항 31.

부호화 스트림을 복호처리하는 복호 장치에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 부호화 스트림과 함께 입력하는 입력 수단과,

상기 입력 수단에 의해 입력된 상기 부호화 스트림을 복호처리하여 화상 데이터를 생성함과 함께, 상기 입력 수단에 의해 입력된 상기 양자화 스케일을, 상기 화상 데이터와 함께 출력하는 복호 수단을 구비하는, 복호 장치.

청구항 32.

부호화 스트림을 복호처리하는 복호 장치의 복호 방법에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일을, 상기 부호화 스트림과 함께 입력하는 입력 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 부호화 스트림을 복호처리하여 화상 데이터를 생성하는 복호 단계와,

상기 입력 단계의 처리에 의해 입력된 상기 양자화 스케일을, 상기 화상 데이터와 함께 출력하는 출력 단계를 포함하는, 복호 방법.

청구항 33.

제 1 부호화 스트림을 제 2 부호화 스트림으로 변환하는 부호화 스트림 변환장치에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일로부터, 현재 세대에서 재이용하는 양자화 스케일을, 재이용 양자화 스케일로서 선택하는 선택 수단과,

상기 선택 수단에 의해 선택된 상기 재이용 양자화 스케일을 사용하여, 상기 제 1 부호화 스트림을 상기 제 2 부호화 스트림으로 변환하는 변환 수단을 구비하는, 부호화 스트림 변환 장치.

청구항 34.

제 1 부호화 스트림을 제 2 부호화 스트림으로 변환하는 부호화 스트림 변환장치의 부호화 스트림 변환 방법에 있어서,

과거의 부호화 처리에서 이용된 양자화 스케일로부터, 현재 세대에서 재이용하는 양자화 스케일을, 재이용 양자화 스케일로서 선택하는 선택 단계와,

상기 선택 단계의 처리에 의해 선택된 상기 재이용 양자화 스케일을 사용하여, 상기 제 1 부호화 스트림을 상기 제 2 부호화 스트림으로 변환하는 변환 단계를 포함하는, 부호화 스트림 변환 방법.

청구항 35.

제 1 부호화 스트림을 제 2 부호화 스트림으로 변환하는 부호화 스트림 변환 장치에 있어서,

현재의 부호화 처리에서 양자화 스케일을 산출함과 함께, 과거의 부호화 처리에서 이용된 양자화 스케일 및 상기 산출된 양자화 스케일 중에서, 현재 세대에서 재이용하는 양자화 스케일을, 재이용 양자화 스케일로서 선택하는 선택 수단과,

상기 선택 수단에 의해 선택된 상기 재이용 양자화 스케일을 사용하여, 상기 제 1 부호화 스트림을 상기 제 2 부호화 스트림으로 변환하는 변환 수단을 구비하는, 부호화 스트림 변환 장치.

청구항 36.

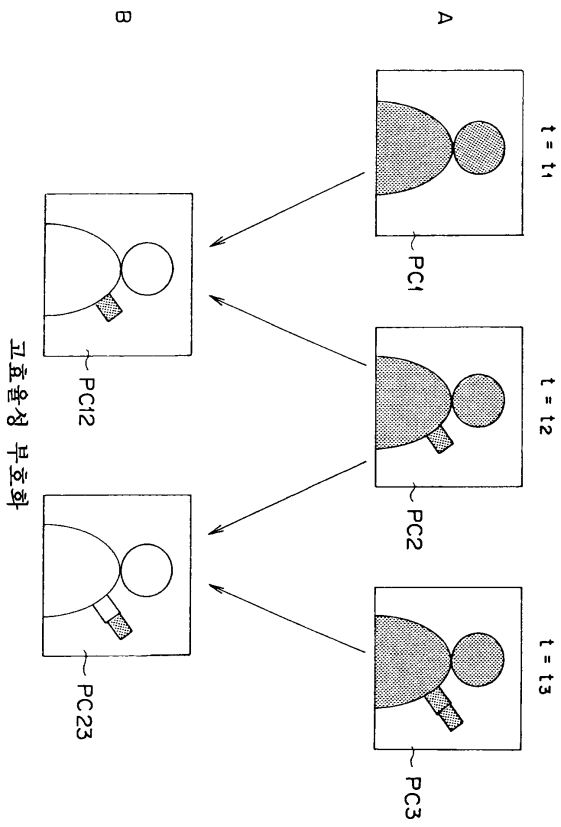
제 1 부호화 스트림을 제 2 부호화 스트림으로 변환하는 부호화 스트림 변환 장치의 부호화 스트림 변환 방법에 있어서,

현재의 부호화 처리에서 양자화 스케일을 산출함과 함께, 과거의 부호화 처리에서 이용된 양자화 스케일 및 상기 산출된 양자화 스케일 중에서, 현재 세대에서 재이용하는 양자화 스케일을, 재이용 양자화 스케일로서 선택하는 선택 단계와,

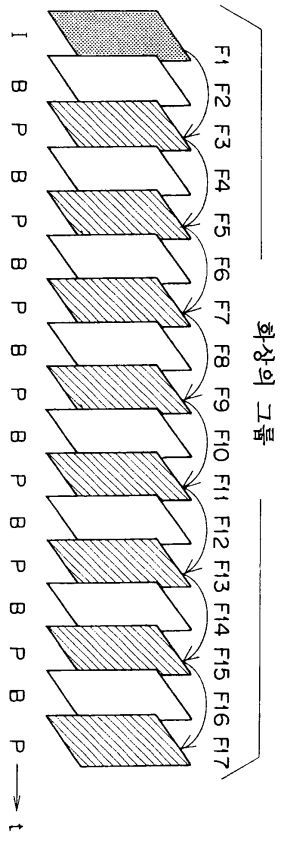
상기 선택 단계의 처리에 의해 선택된 상기 재이용 양자화 스케일을 사용하여, 상기 제 1 부호화 스트림을 상기 제 2 부호화 스트림으로 변환하는 변환 단계를 포함하는, 부호화 스트림 변환 방법.

도면

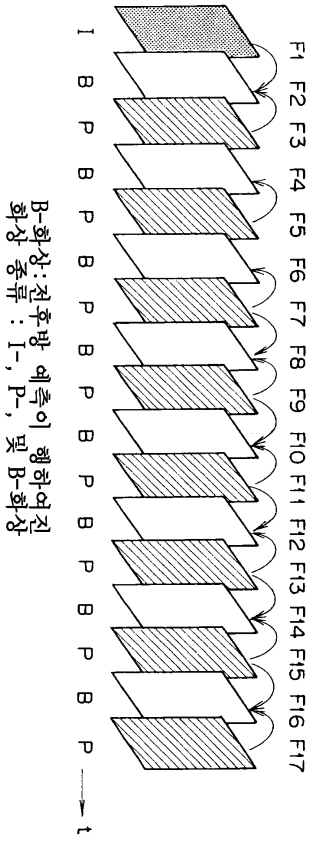
도면1


A
B
t = t1
t = t2
t = t3
PC1
PC2
PC3
PC12
PC23
고효율성 부호화

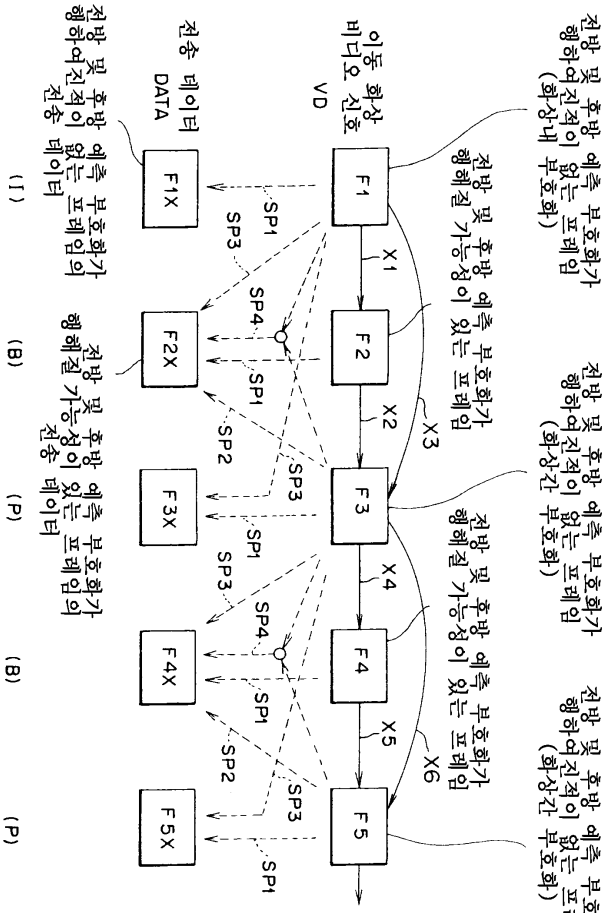
도면2


화상의 그룹
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17
I B P B P B P B P B P B P B P B P
t
P-화상:전방 예측이 행하여진
화상 종류 : I-, P-, 및 B-화상

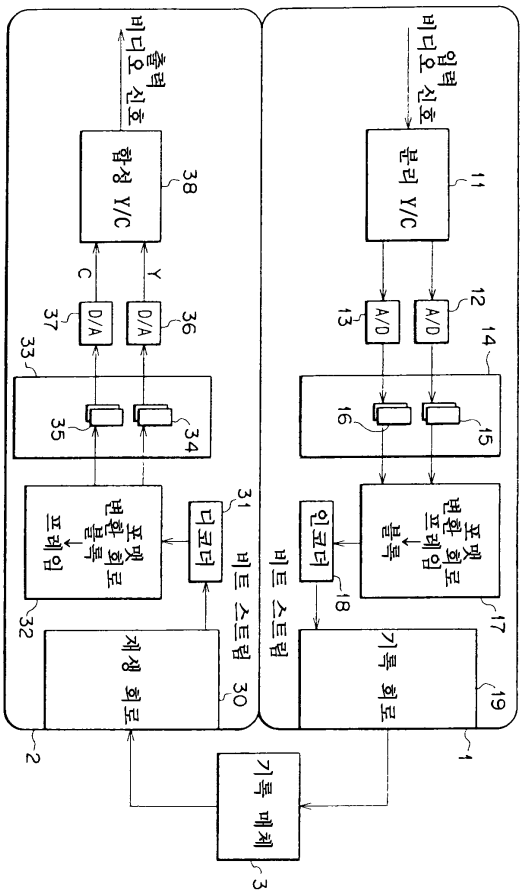
도면3


F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17
I B P B P B P B P B P B P B P B P
t
B-화상:전후방 예측이 행하여진
화상 종류 : I-, P-, 및 B-화상

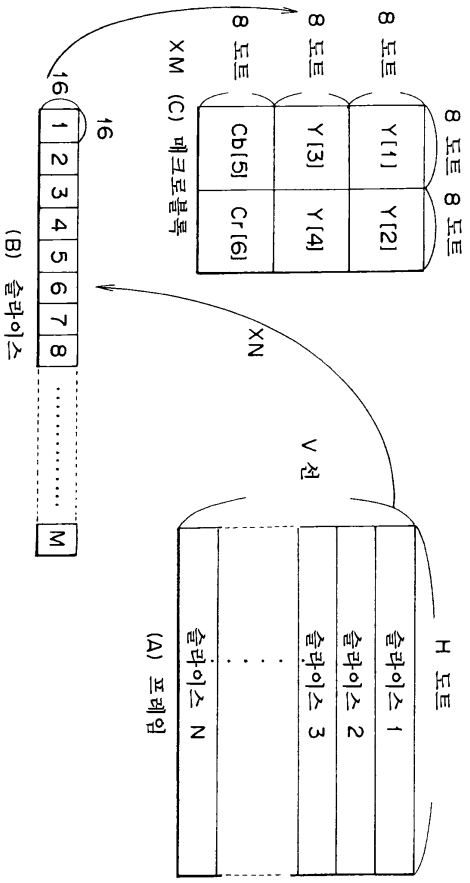
도면4


이동 화상
비디오 신호
VD
전송 데이터
DATA
F1
F2
F3
F4
F5
F1X
F2X
F3X
F4X
F5X
X1
X2
X3
X4
X5
X6
SP1
SP2
SP3
SP4
전방 및 후방 예측 부호화가 행하여진적이 없는 프레임 (화상내 부호화)
전방 및 후방 예측 부호화가 행하여진적이 없는 프레임 (화상간 부호화)
전방 및 후방 예측 부호화가 행하여진적이 있는 프레임 (화상간 부호화)
전방 및 후방 예측 부호화가 행해질 가능성이 있는 프레임
전방 및 후방 예측 부호화가 행해질 가능성이 있는 프레임의 전송 데이터
(I)
(B)
(P)
(B)
(P)

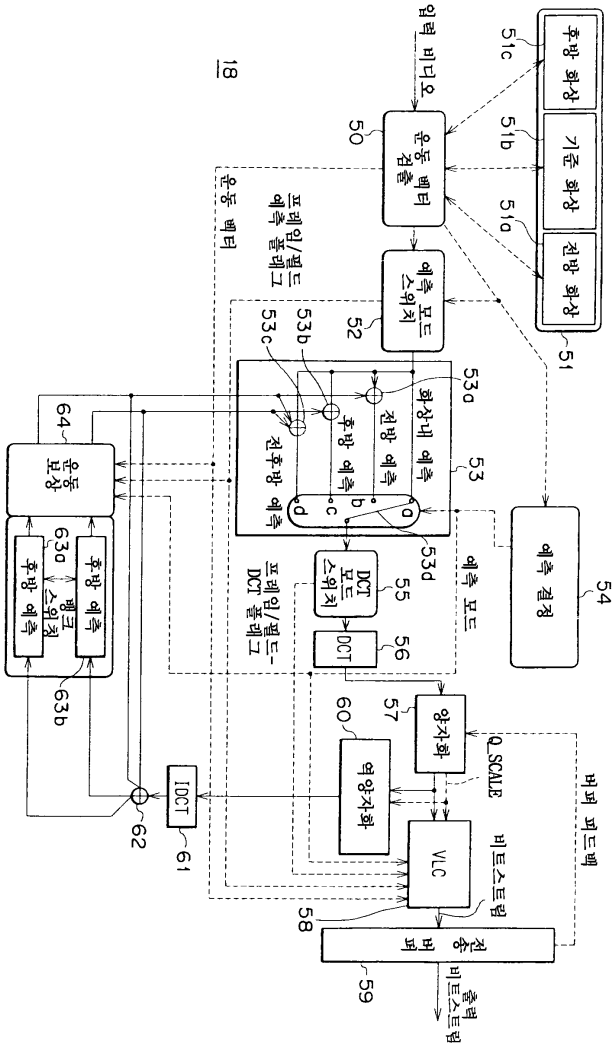
도면5


입력 비디오 신호
분리 Y/C
11
A/D
12
13
14
16
15
포맷 변환 회로 블록 프레임
17
인코더
18
기록 회로
19
1
비트 스트림
출력 비디오 신호
합성 Y/C
38
C
Y
D/A
37
36
33
35
34
31
디코더
포맷 변환 회로 블록 프레임
32
재생 회로
30
2
기록 매체
3

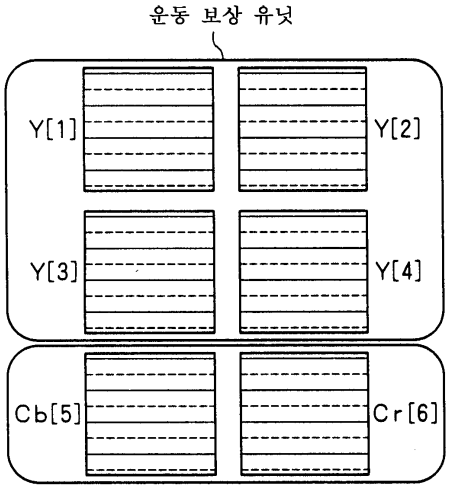
도면6


8 도트
8 도트
8 도트 8 도트
Y[1]
Y[2]
Y[3]
Y[4]
Cb[5]
Cr[6]
8 도트
XM (C) 매크로블록
16
16
1 2 3 4 5 6 7 8
M
(B) 슬라이스
XN
V 선
H 도트
슬라이스 1
슬라이스 2
슬라이스 3
슬라이스 N
(A) 프레임

도면7



도면8

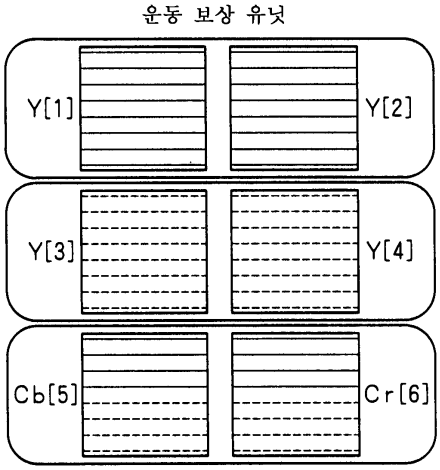


프레임 예측 모드

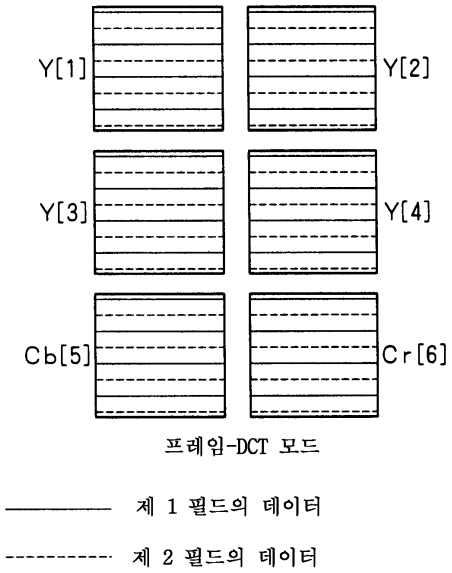
———— 제 1 필드의 데이터

---------- 제 2 필드의 데이터

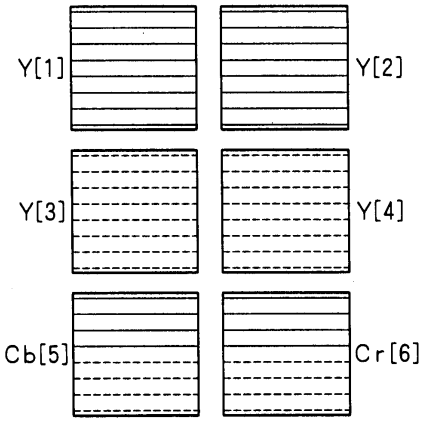
도면9


운동 보상 유닛
Y[1]
Y[2]
Y[3]
Y[4]
Cb[5]
Cr[6]
필드 예측 모드
제 1 필드의 데이터
제 2 필드의 데이터

도면10


Y[1]
Y[2]
Y[3]
Y[4]
Cb[5]
Cr[6]
프레임-DCT 모드
제 1 필드의 데이터
제 2 필드의 데이터

도면11

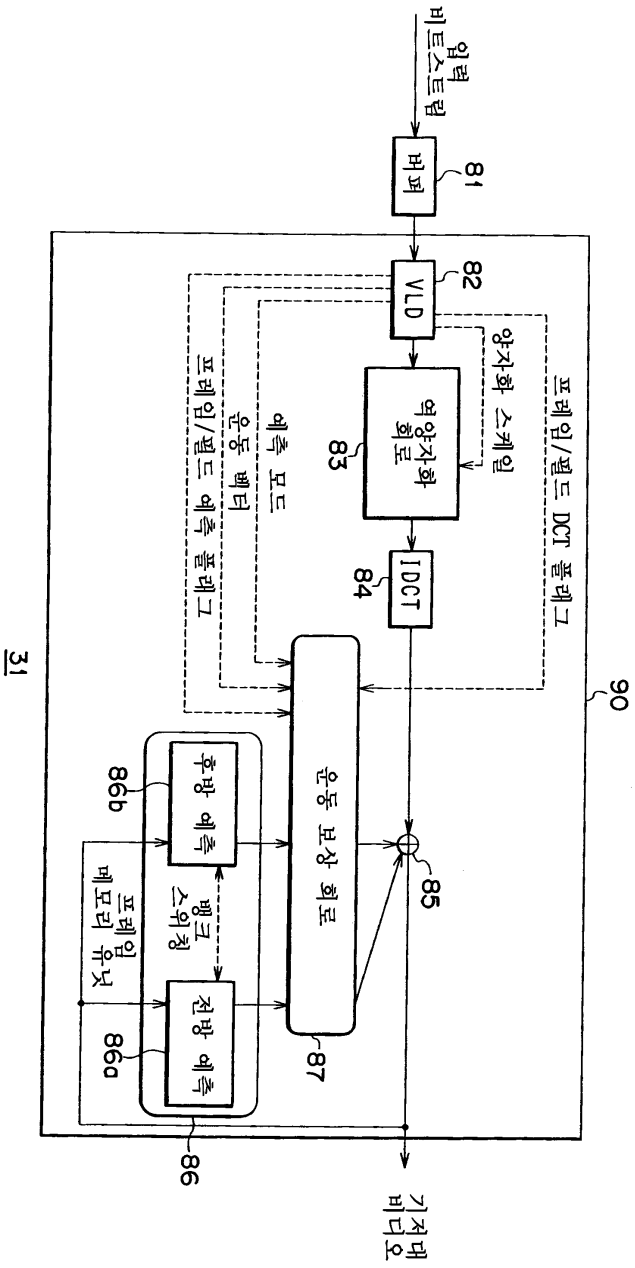


필드 -DCT 모드

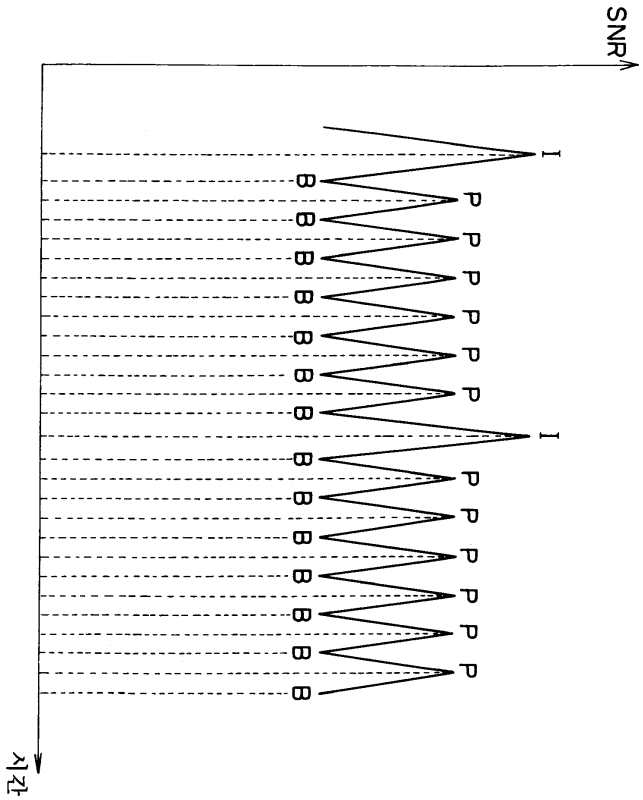
———— 제 1 필드의 데이터

---------- 제 2 필드의 데이터

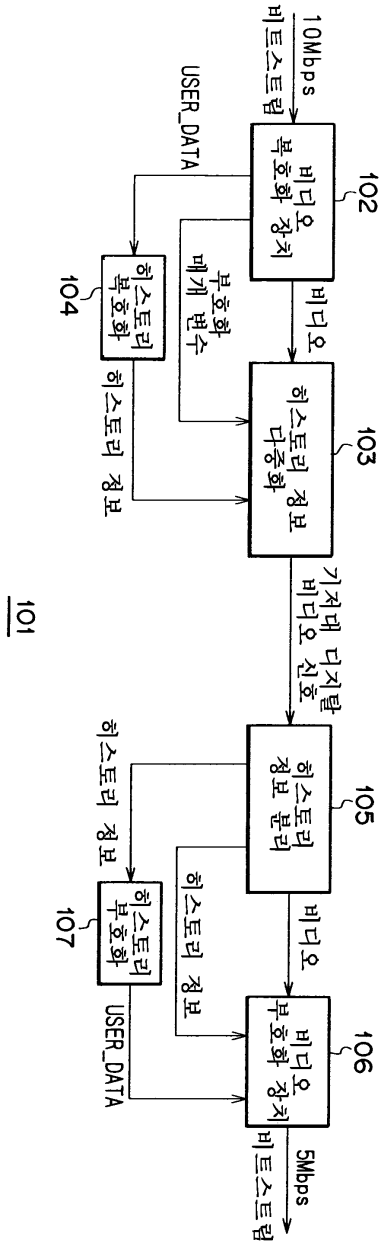
도면12


입력 비트스트림
버퍼
81
VLD
82
역양자화 회로
83
IDCT
84
양자화 스케일
프레임/필드 DCT 플래그
예측 모드
운동 벡터
프레임/필드 예측 플래그
운동 보상 회로
87
85
후방 예측
86b
뱅크 스위칭
프레임 메모리 유닛
전방 예측
86a
86
90
31
기저대 비디오

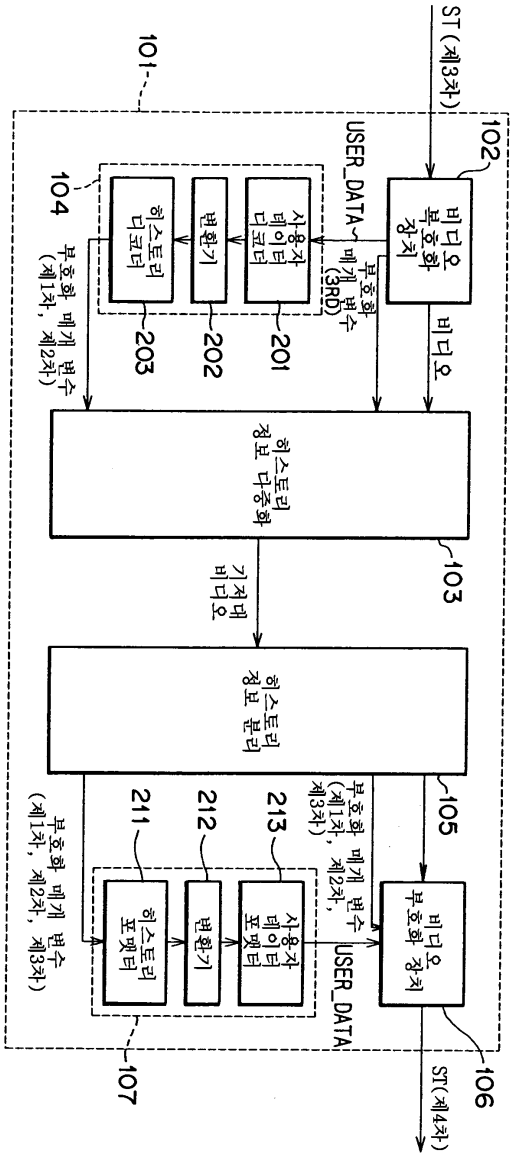
도면13



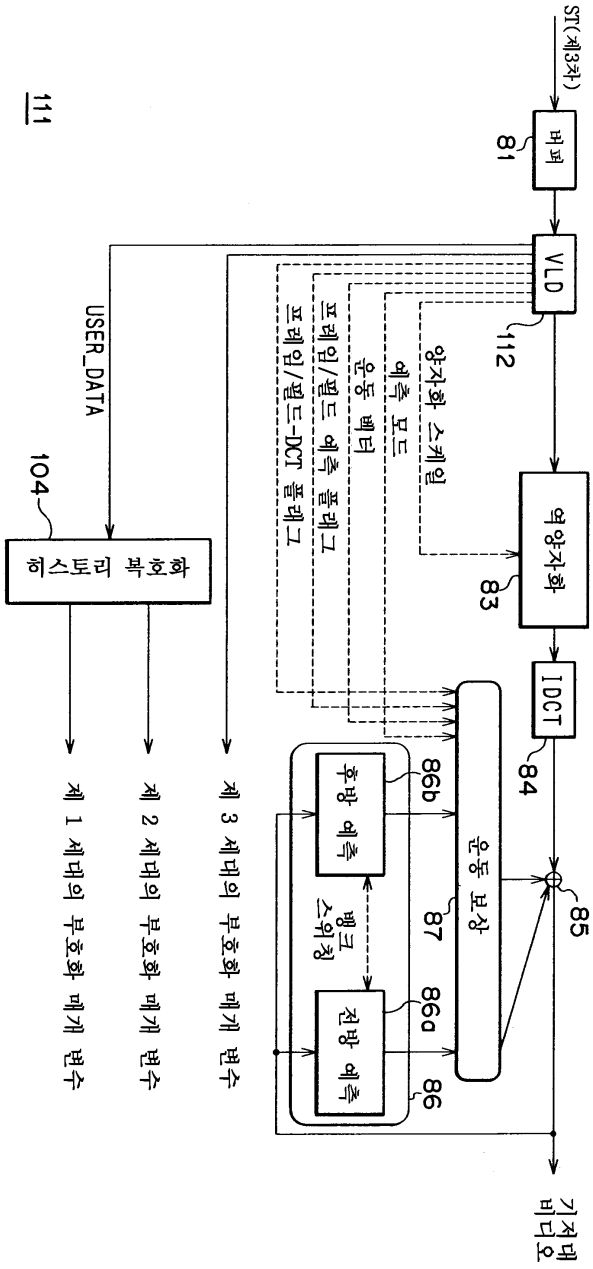
도면14


101
10Mbps
비트스트림
비디오
복호화 장치
102
USER_DATA
부호화 매개 변수
히스토리 복호화
104
히스토리 정보
히스토리 정보 다중화
103
기저대 디지탈 비디오 신호
히스토리 정보
히스토리 정보 분리
105
히스토리 정보
히스토리 부호화
107
USER_DATA
비디오
비디오 부호화 장치
106
5Mbps
비트스트림

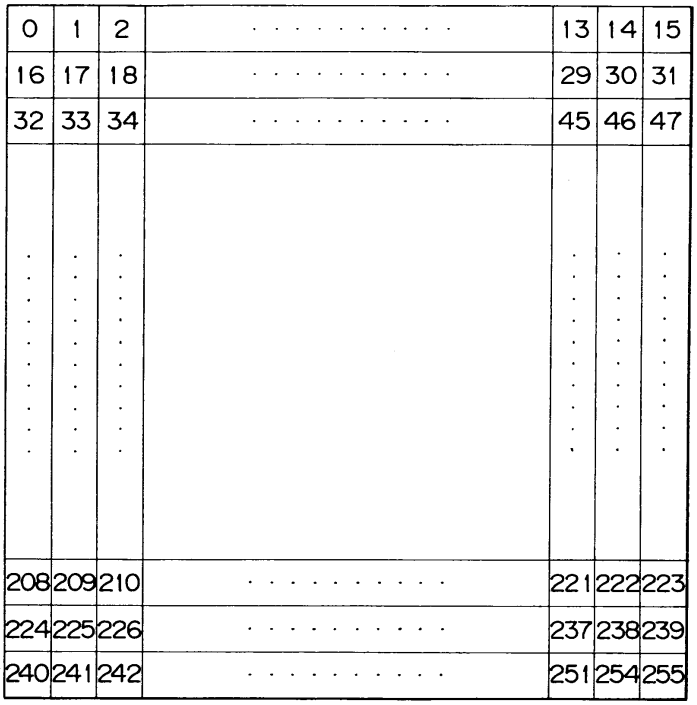
도면15


ST (제3차)
101
102
USER_DATA
비디오 복호화 장치
104
사용자 데이터 디코더
변환기
히스토리 디코더
201
202
203
메개변수 (3RD)
복호화
비디오
부호화 메개 변수 (제1차, 제2차)
히스토리 정보 다중화
103
기저대 비디오
히스토리 정보 분리
105
부호화 메개 변수 (제1차, 제2차, 제3차)
213
212
211
사용자 데이터 포맷터
변환기
히스토리 포맷터
USER_DATA
부호화 메개 변수 (제1차, 제2차, 제3차)
107
비디오 부호화 장치
106
ST(제4차)

도면16

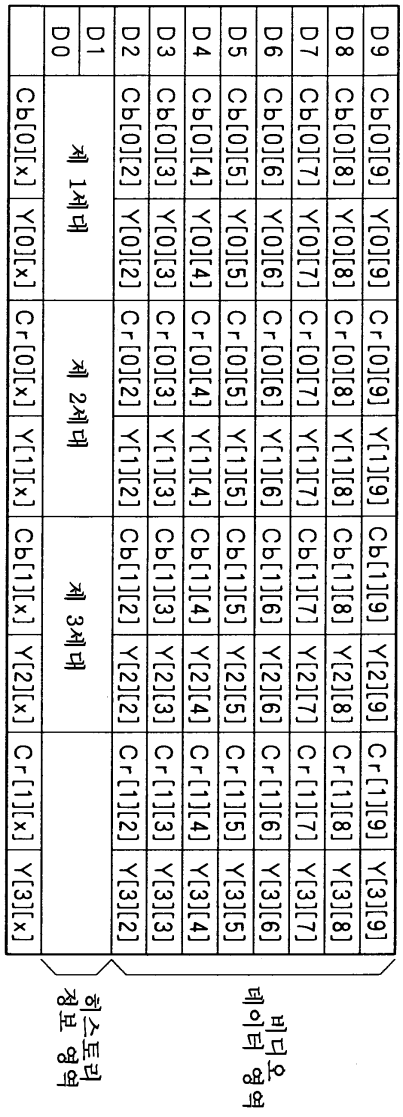

111
ST(제3차)
버퍼
81
VLD
112
양자화 스케일
예측 모드
움동 벡터
프레임/필드 예측 플래그
프레임/필드-DCT 플래그
USER_DATA
104
히스토리 복호화
역양자화
83
IDCT
84
후방 예측
86b
뱅크 스위칭
전방 예측
86a
86
움동 보상
87
85
제 1 세대의 부호화 매개 변수
제 2 세대의 부호화 매개 변수
제 3 세대의 부호화 매개 변수
기저대 비디오

도면17

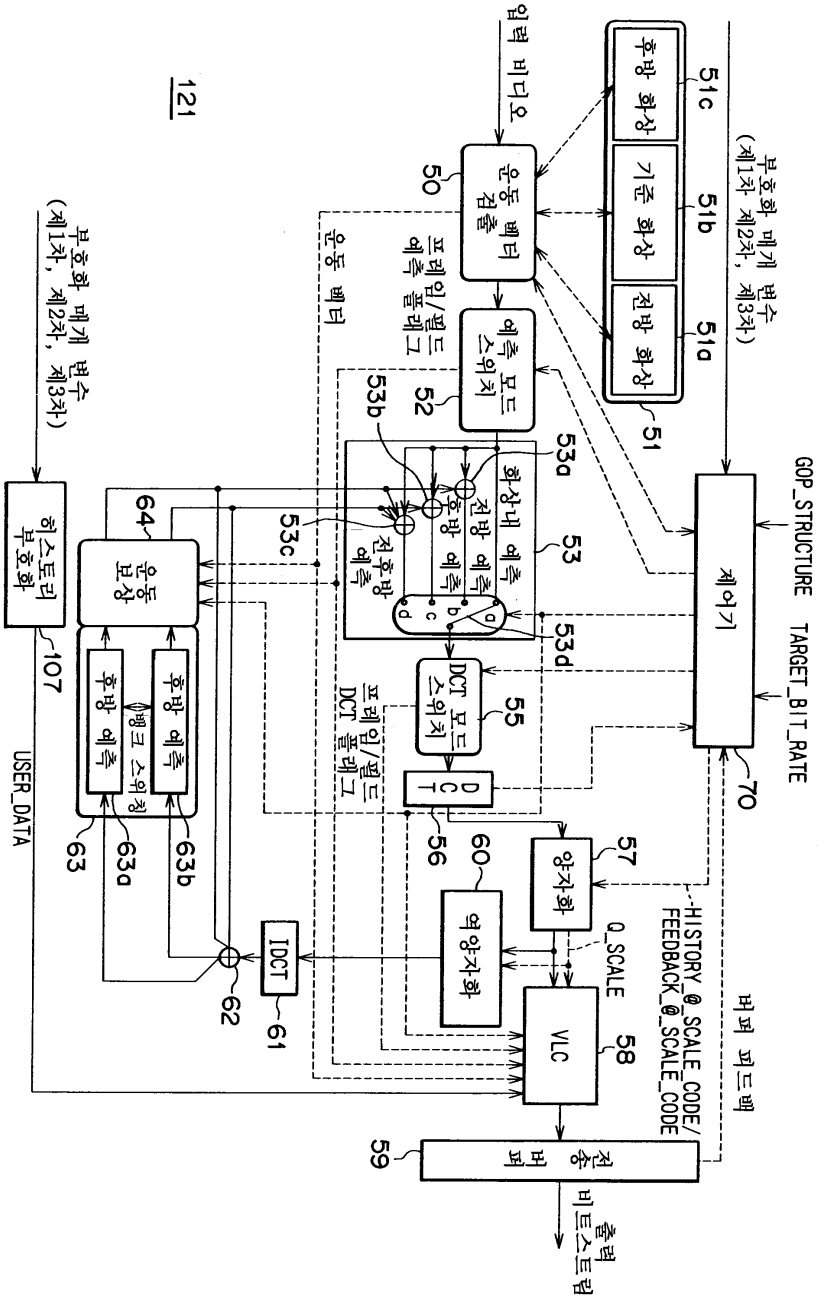

0 1 2 13 14 15
16 17 18 29 30 31
32 33 34 45 46 47
208 209 210 221 222 223
224 225 226 237 238 239
240 241 242 251 254 255

매크로블록

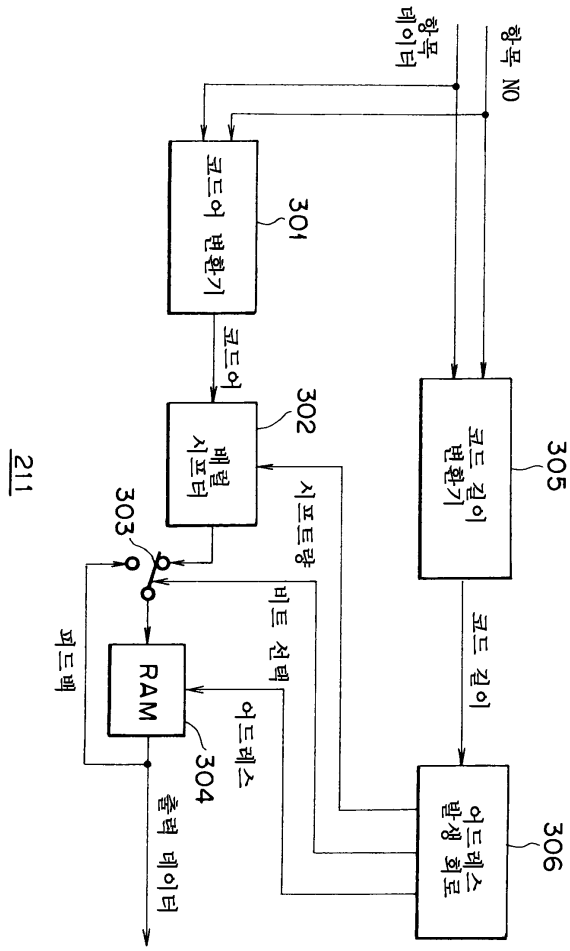
도면18


D9 Cb[0][9] Y[0][9] Cr[0][9] Y[1][9] Cb[1][9] Y[2][9] Cr[1][9] Y[3][9]
D8 Cb[0][8] Y[0][8] Cr[0][8] Y[1][8] Cb[1][8] Y[2][8] Cr[1][8] Y[3][8]
D7 Cb[0][7] Y[0][7] Cr[0][7] Y[1][7] Cb[1][7] Y[2][7] Cr[1][7] Y[3][7]
D6 Cb[0][6] Y[0][6] Cr[0][6] Y[1][6] Cb[1][6] Y[2][6] Cr[1][6] Y[3][6]
D5 Cb[0][5] Y[0][5] Cr[0][5] Y[1][5] Cb[1][5] Y[2][5] Cr[1][5] Y[3][5]
D4 Cb[0][4] Y[0][4] Cr[0][4] Y[1][4] Cb[1][4] Y[2][4] Cr[1][4] Y[3][4]
D3 Cb[0][3] Y[0][3] Cr[0][3] Y[1][3] Cb[1][3] Y[2][3] Cr[1][3] Y[3][3]
D2 Cb[0][2] Y[0][2] Cr[0][2] Y[1][2] Cb[1][2] Y[2][2] Cr[1][2] Y[3][2]
D1 D0
Cb[0][x] Y[0][x] Cr[0][x] Y[1][x] Cb[1][x] Y[2][x] Cr[1][x] Y[3][x]
제 1세대
제 2세대
제 3세대
히스토리 정보 영역
비디오 데이터 영역

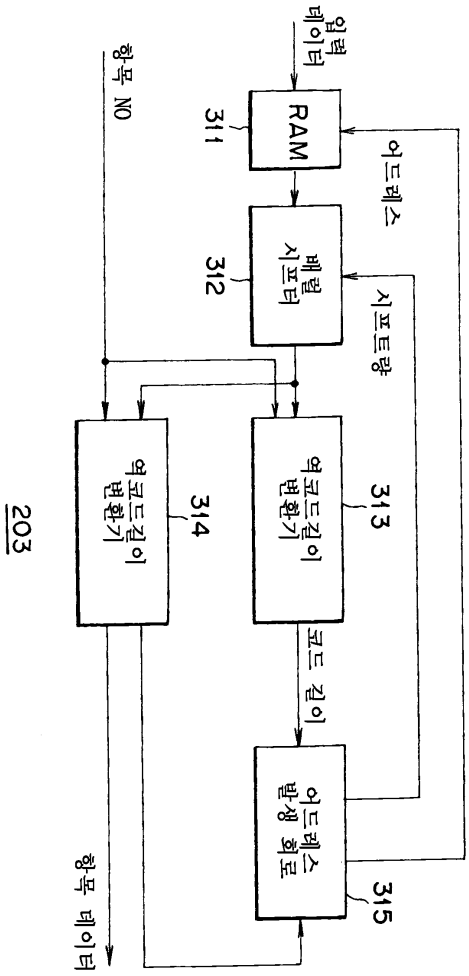
도면19


121
입력 비디오
부호화 매개 변수
(제1차, 제2차, 제3차)
부호화 매개 변수
(제1차 제2차, 제3차)
후방 화상
기준 화상
전방 화상
51c
51b
51a
51
50
움직임 벡터 검출
프레임/필드 예측 플래그
움직임 벡터
예측 모드 스위치
52
53b
53a
화상내 예측
전방 예측
후방 예측
전후방 예측
53
53c
53d
DCT 모드 스위치
55
프레임/필드 DCT 플래그
DCT
56
양자화
57
60
역양자화
IDCT
61
62
VLC
58
전송 버퍼
59
출력 비트스트림
제어기
70
GOP_STRUCTURE
TARGET_BIT_RATE
버퍼 피드백
HISTORY @ SCALE CODE/
FEEDBACK @ SCALE CODE
Q_SCALE
하프도리 부호화
107
USER_DATA
움직임 보상
64
후방 예측
뱅크 스위칭
63
63a
63b

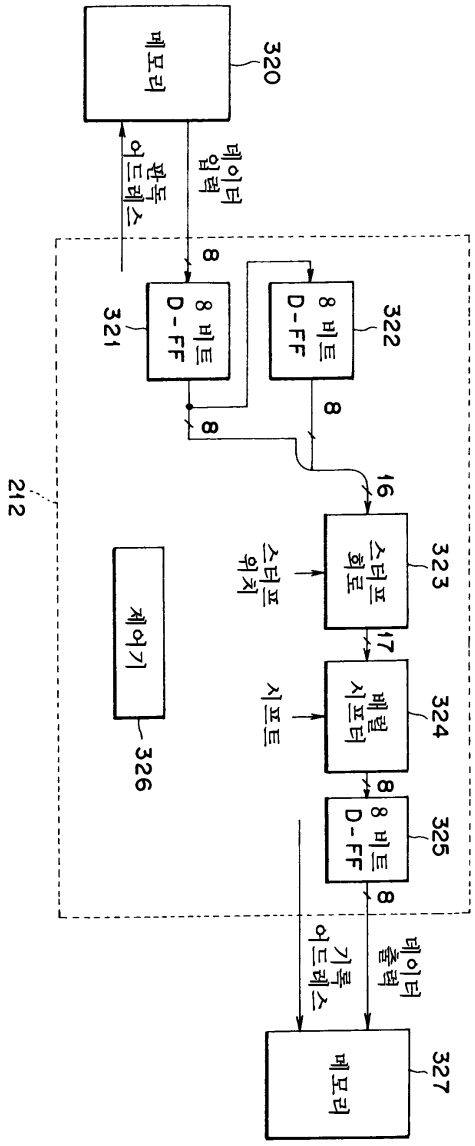
도면20


211
항목 데이터
항목 NO
코드어 변환기
301
코드어
배럴 시프터
302
시프트량
303
RAM
304
출력 데이터
피드백
비트 선택
어드레스
코드 길이 변환기
305
코드 길이
어드레스 발생 회로
306

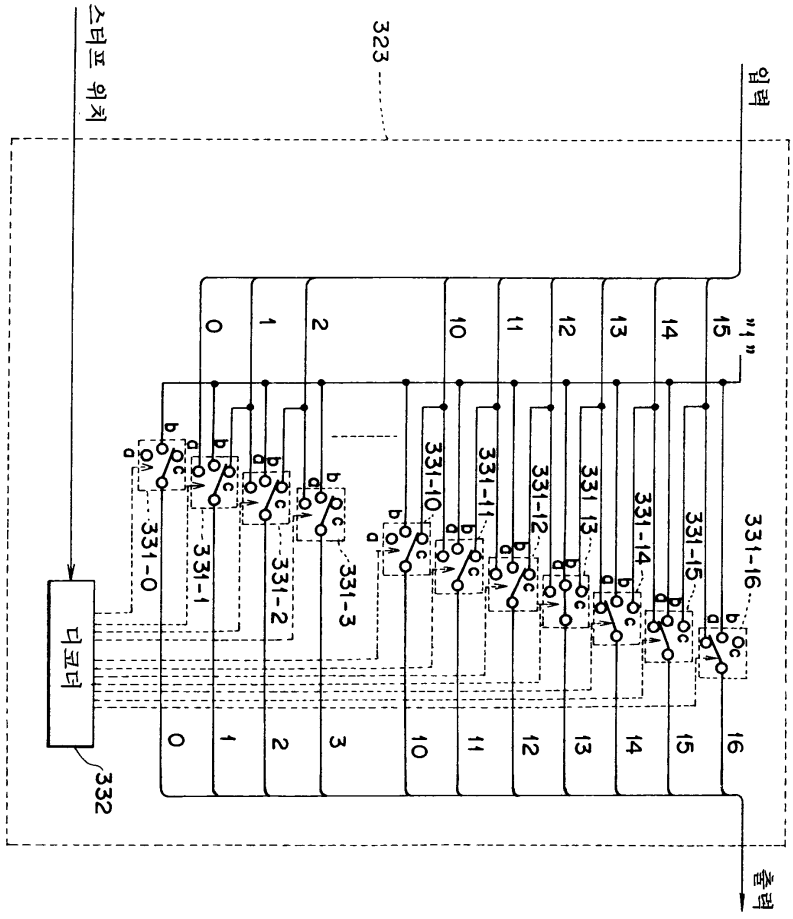
도면21


입력 데이터
RAM
311
어드레스
베럴 시프터
312
시프트량
항목 NO
역코드길이 변환기
314
역코드길이 변환기
313
코드 길이
어드레스 발생 회로
315
항목 데이터
203

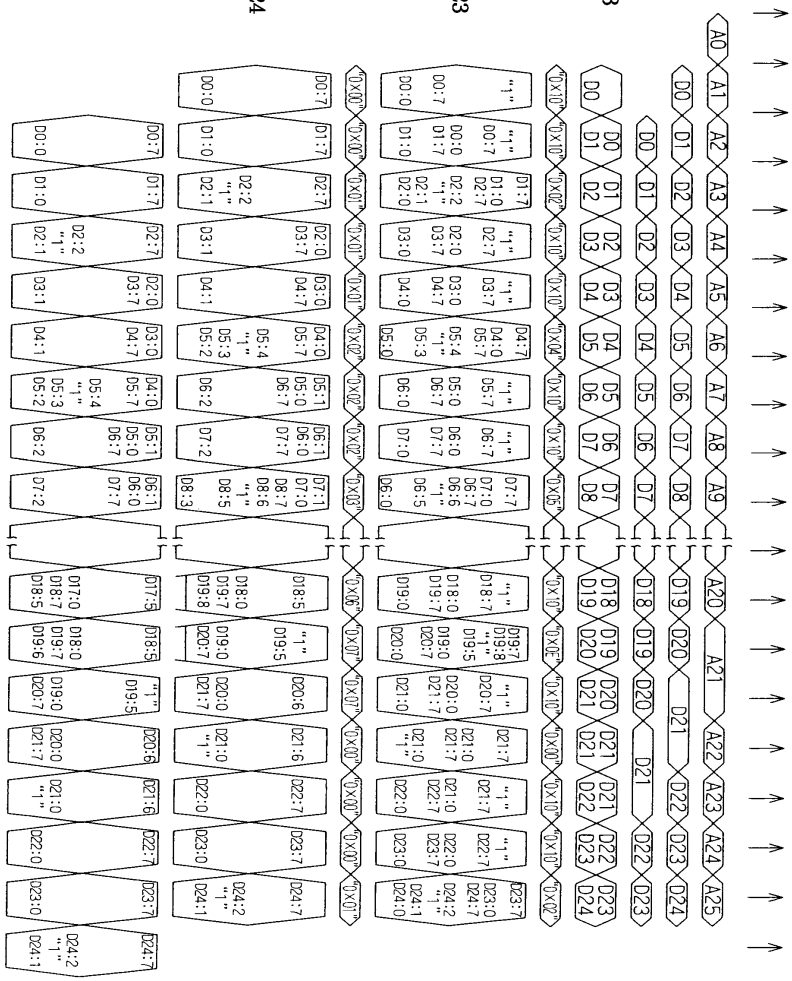
도면22


메모리
320
판독 어드레스
데이터 입력
8
8 비트 D-FF
321
8 비트 D-FF
322
8
8
16
212
스터프 회로
323
스터프 위치
17
배럴 시프터
324
시프트
제어기
326
8
8 비트 D-FF
325
8
기록 어드레스
데이터 출력
메모리
327

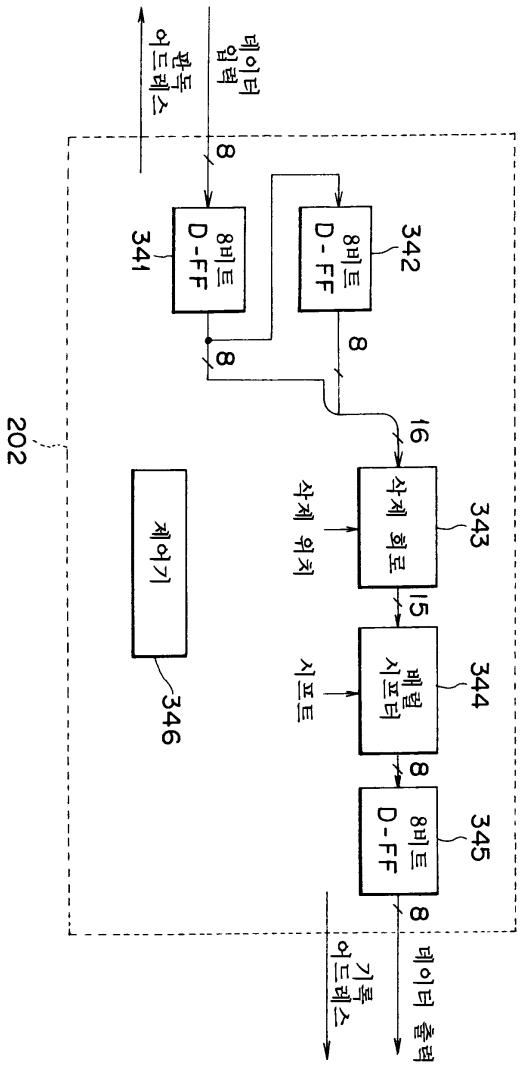
도면23


스티프 위치
323
입력
"1"
331-0
331-1
331-2
331-3
331-10
331-11
331-12
331 13
331-14
331-15
331-16
디코더
332
출력

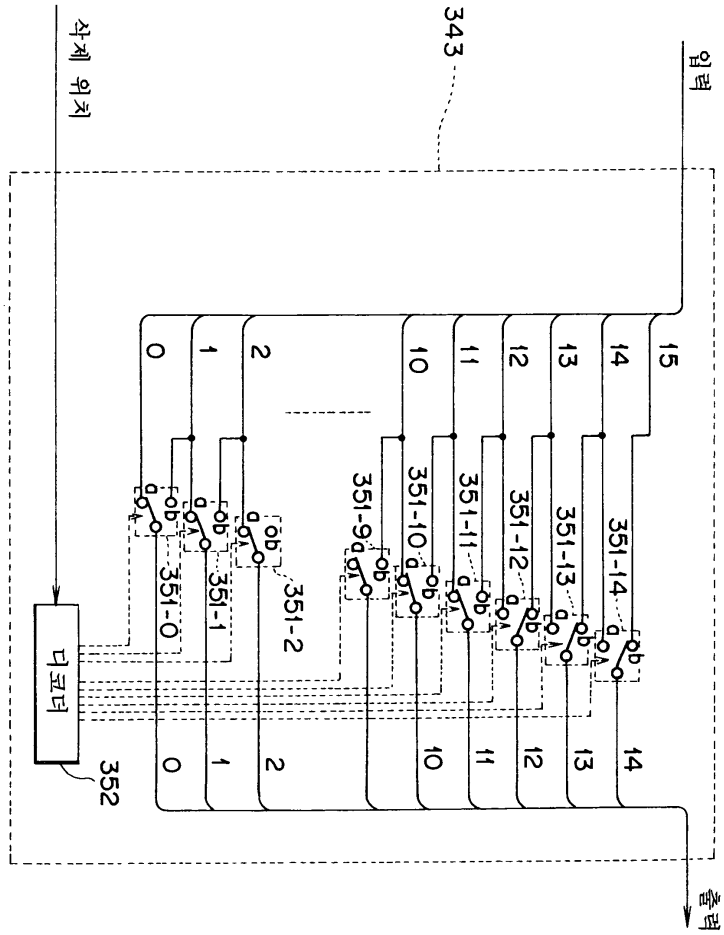
도면24


(A) 판독 어드레스
(B) D-FF 321 출력
(C) D-FF 322 출력
(D) 스터프 회로 323 입력
(E) 스터프 위치
(F) 스터프 회로 323 출력
(G) 시프트
(H) 배럴 시프터 324 출력
(I) D-FF 325 출력

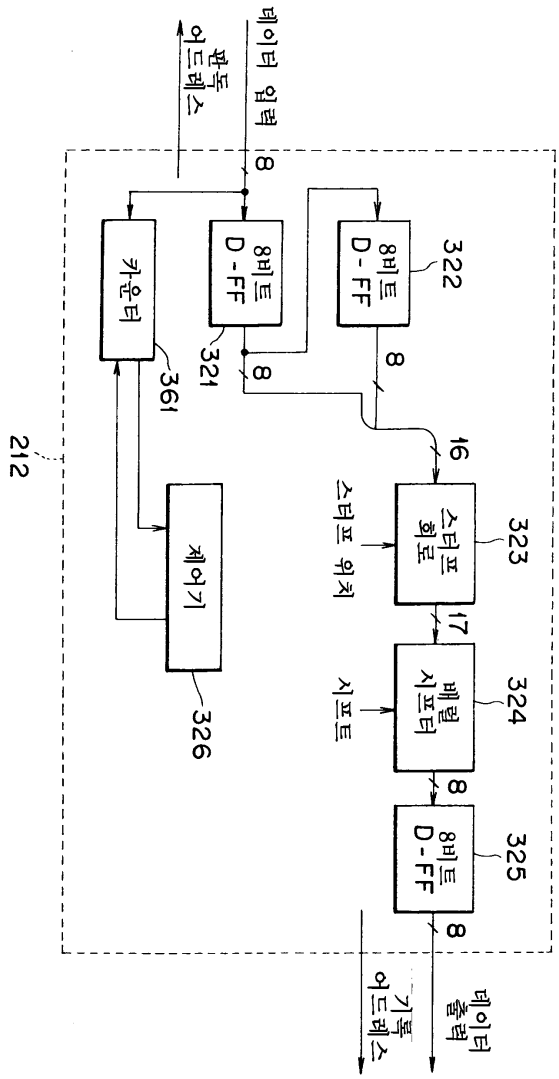
도면25


202
341
342
343
344
345
346
8
8
8
16
15
8
8
8
8비트 D-FF
8비트 D-FF
삭제 회로
베럴 시프터
8비트 D-FF
제어기
데이터 입력
판독 어드레스
삭제 위치
시프트
기록 어드레스
데이터 출력

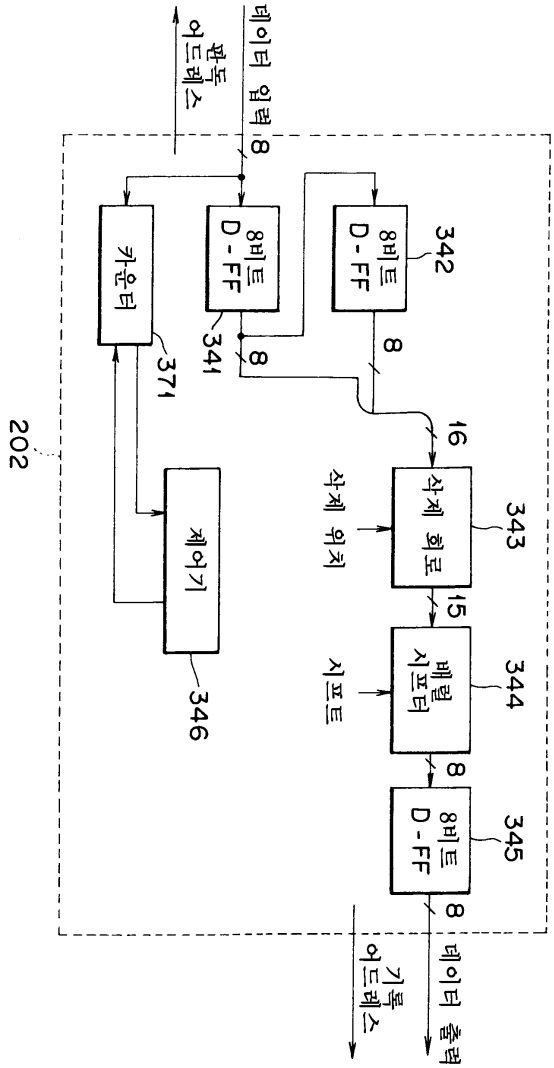
도면26


삭제 위치
343
입력
출력
디코더
352
351-0
351-1
351-2
351-9
351-10
351-11
351-12
351-13
351-14
a
b
0
1
2
10
11
12
13
14
15

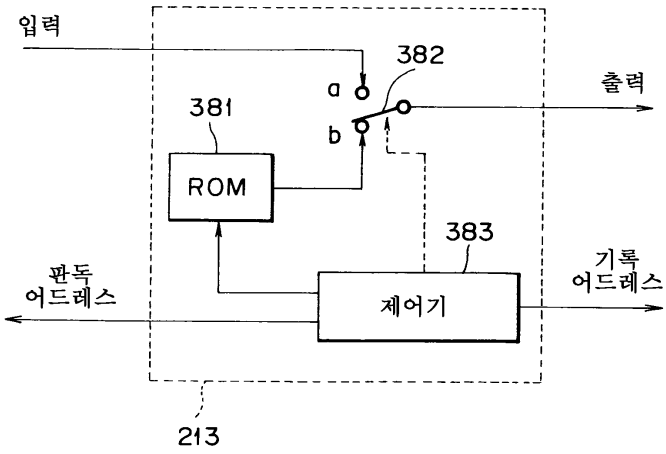
도면27


데이터 입력
8
판독 어드레스
8비트 D-FF
322
8비트 D-FF
321
8
카운터
361
8
16
212
스터프 위치
스터프 회로
323
17
제어기
326
시프트
배럴 시프터
324
8
8비트 D-FF
325
8
기록 어드레스
데이터 출력

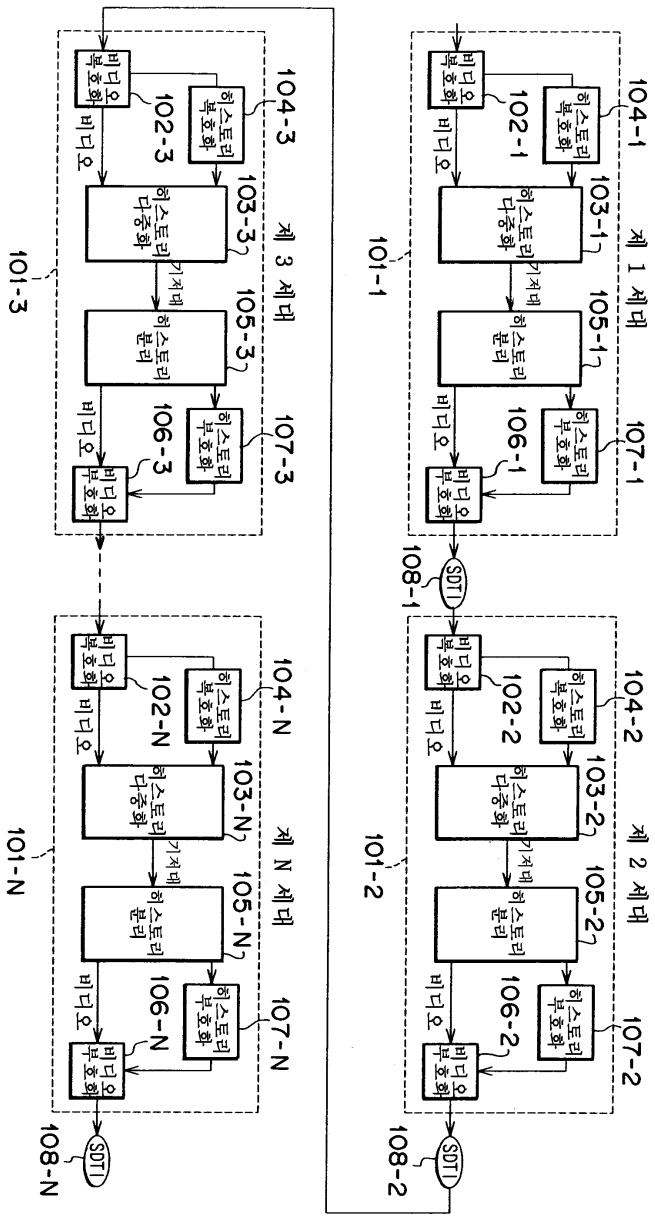
도면28


데이터 입력
8
판독 어드레스
카운터
8비트 D-FF
342
341
371
202
8
16
삭제 위치
삭제 회로
343
15
시프트
배럴 시프터
344
8
8비트 D-FF
345
8
제어기
346
기록 어드레스
데이터 출력

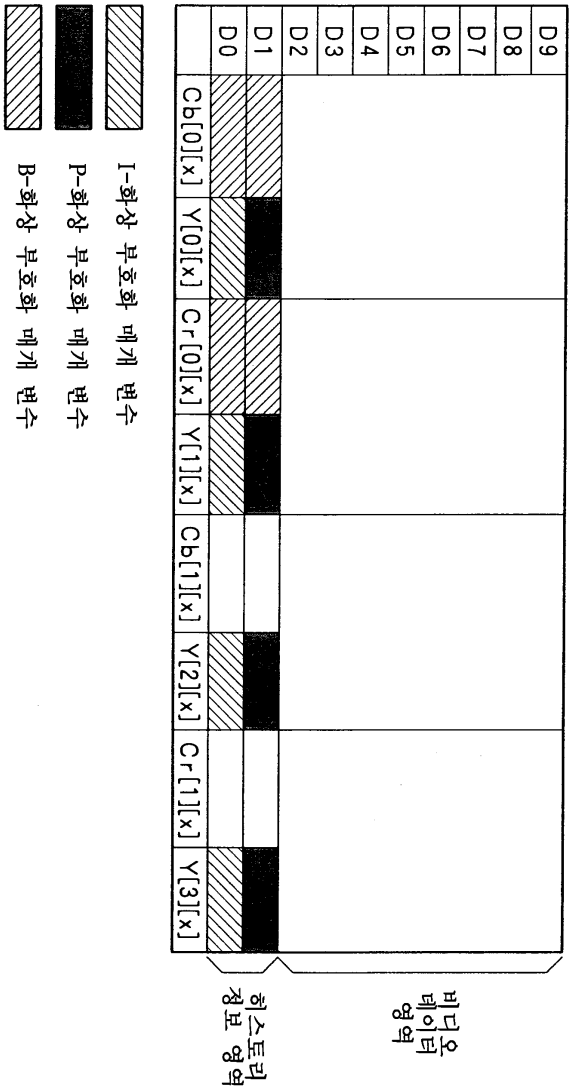
도면29


입력
382
a
b
출력
381
ROM
383
판독 어드레스
제어기
기록 어드레스
213

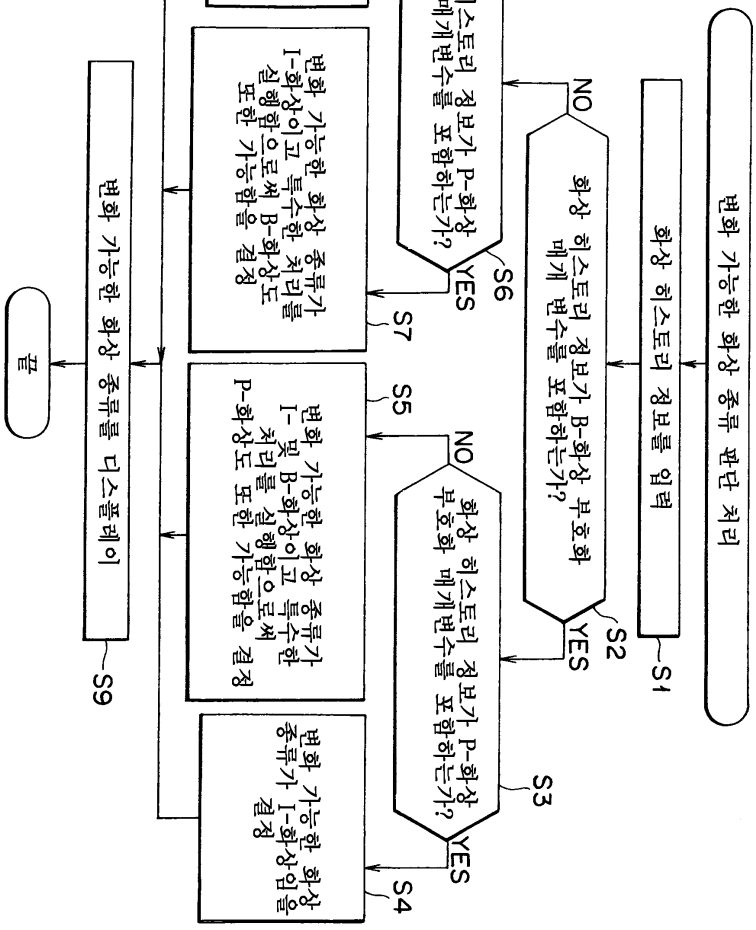
도면30


제 1 세대
제 2 세대
제 3 세대
제 N 세대
101-1
101-2
101-3
101-N
102-1
102-2
102-3
102-N
103-1
103-2
103-3
103-N
104-1
104-2
104-3
104-N
105-1
105-2
105-3
105-N
106-1
106-2
106-3
106-N
107-1
107-2
107-3
107-N
108-1
108-2
108-N
비디오 복호화
히스토리 복호화
히스토리 다중화
히스토리 분리
히스토리 부호화
비디오 부호화
비디오
기저대
SDTI

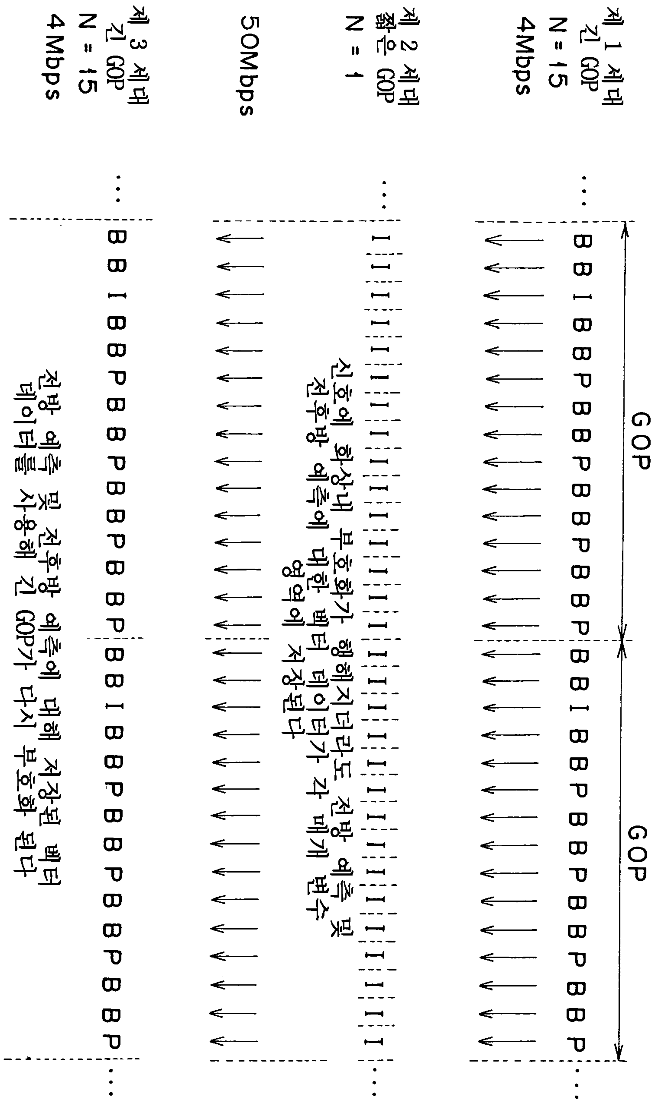
도면31


D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
Cb[0][x]
Y[0][x]
Cr[0][x]
Y[1][x]
Cb[1][x]
Y[2][x]
Cr[1][x]
Y[3][x]
비디오 데이터 영역
히스토리 정보 영역
I-화상 부호화 매개 변수
P-화상 부호화 매개 변수
B-화상 부호화 매개 변수

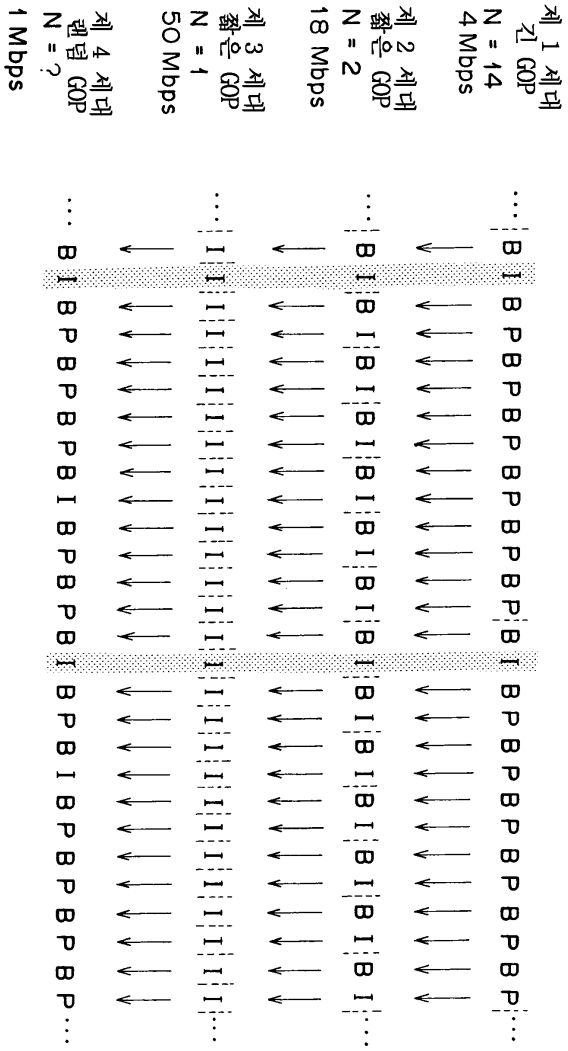
도면32


변화 가능한 화상 종류 판단 처리
화상 히스토리 정보를 입력
S1
화상 히스토리 정보가 B-화상 부호화 매개 변수를 포함하는가?
S2
NO
YES
화상 히스토리 정보가 P-화상 부호화 매개변수를 포함하는가?
S3
NO
YES
변화 가능한 화상 종류가 I-화상임을 결정
S4
변화 가능한 화상 종류가 I- 및 P-화상이고 특수한 처리를 실행함으로써 P-화상도 또한 가능함을 결정
S5
변화 가능한 화상 종류가 I-화상이고 특수한 처리를 실행함으로써 B-화상도 또한 가능함을 결정
S6
S7
화상 히스토리 정보가 P-화상 부호화 매개변수를 포함하는가?
NO
YES
변화 가능한 화상 종류가 I-화상임을 결정
S8
변화 가능한 화상 종류를 디스플레이
S9
끝

도면33


제 1 세대
긴 GOP
N = 15
4Mbps
GOP
GOP
B B I B B P B B P B B P B B P B B I B B P B B P B B P B B P
제 2 세대
짧은 GOP
N = 1
I I I I I I I I I I I I I I I I I I I I I I I I I I I I I I
신호에 화상내 부호화가 행해지더라도 전방 예측 및
전후방 예측에 대한 벡터 데이터가 각 매개 변수
영역에 저장된다
50Mbps
제 3 세대
긴 GOP
N = 15
4Mbps
B B I B B P B B P B B P B B P B B I B B P B B P B B P B B P
전방 예측 및 전후방 예측에 대해 저장된 벡터
데이터를 사용해 긴 GOP가 다시 부호화 된다

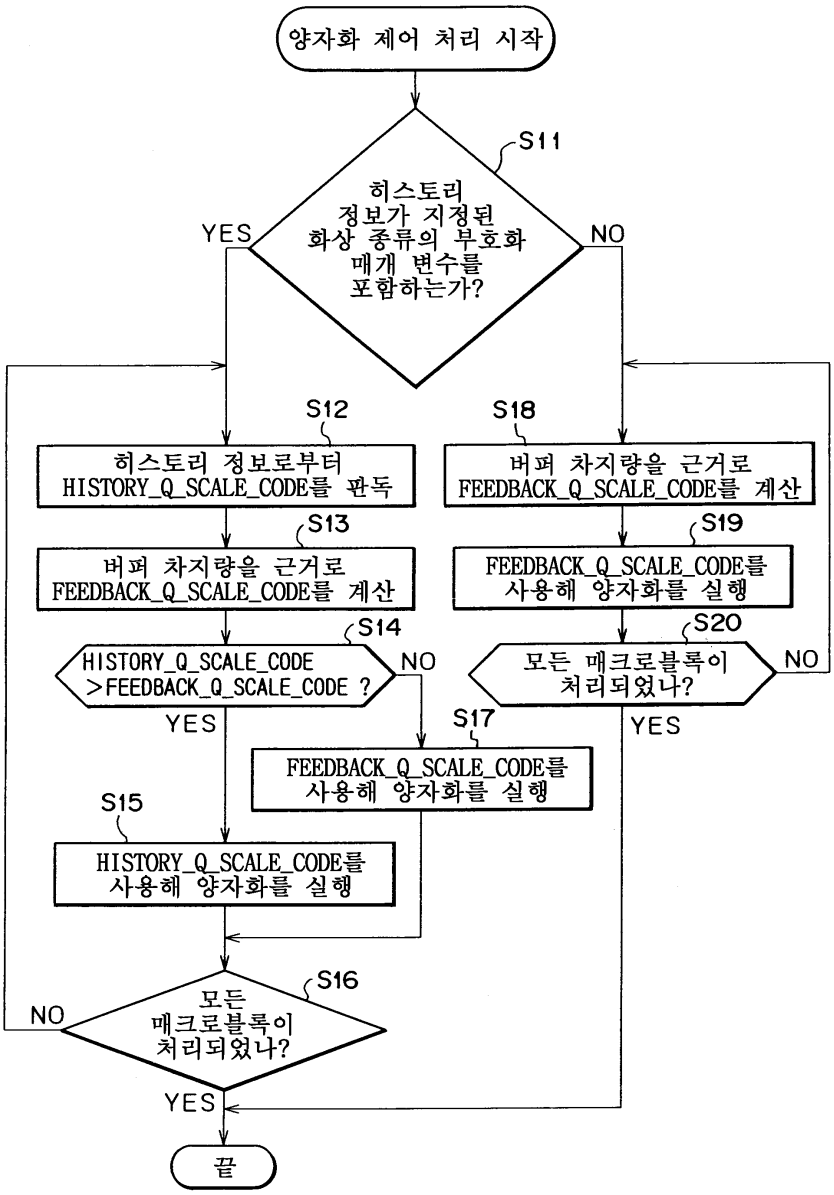
도면34


제 1 세대
긴 GOP
N = 14
4 Mbps
제 2 세대
짧은 GOP
N = 2
18 Mbps
제 3 세대
짧은 GOP
N = 1
50 Mbps
제 4 세대
랜덤 GOP
N = ?
1 Mbps

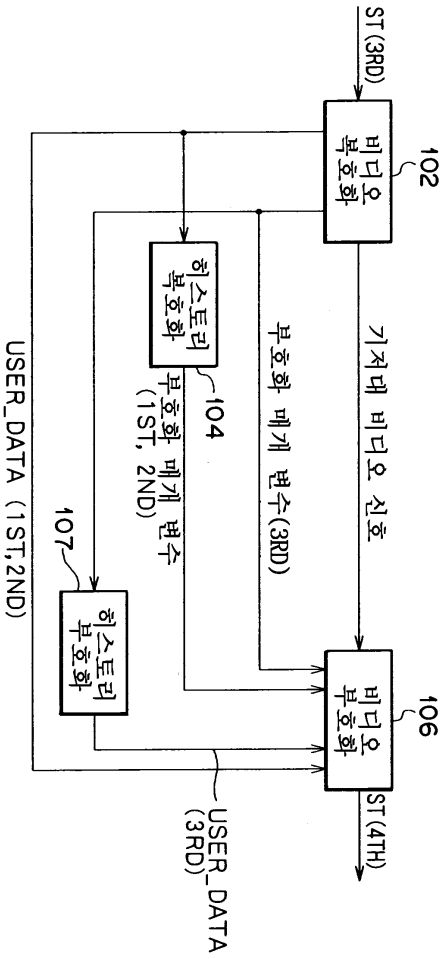
도면35

I-화상
제 1 세대
긴 GOP
N = 14
4Mbps
비트율이
상승되더라도
품질이 개선되지
않는다
가장 거친 이전
양자화 단계
제 1 세대의
양자화 단계값을
유지
I-화상
제 2 세대
짧은 GOP
N = 2
18Mbps
비트율이
상승되더라도
품질이 개선되지
않는다
가장 거친 이전
양자화 단계
제 1 세대의
양자화 단계값을
유지
I-화상
제 3 세대
짧은 GOP
N = 1
50Mbps
비트율이
상승되더라도
품질이 개선되지
않는다
가장 거친 이전
양자화 단계
제 1 세대의
양자화 단계값을
유지
I-화상
제 4 세대
긴 GOP
4Mbps

도면36


양자화 제어 처리 시작
S11
히스토리 정보가 지정된 화상 종류의 부호화 매개 변수를 포함하는가?
YES
NO
S12
히스토리 정보로부터 HISTORY_Q_SCALE_CODE를 판독
S13
버퍼 차지량을 근거로 FEEDBACK_Q_SCALE_CODE를 계산
S14
HISTORY_Q_SCALE_CODE >FEEDBACK_Q_SCALE_CODE ?
NO
YES
S15
HISTORY_Q_SCALE_CODE를 사용해 양자화를 실행
S16
모든 매크로블록이 처리되었나?
NO
YES
끝
S18
버퍼 차지량을 근거로 FEEDBACK_Q_SCALE_CODE를 계산
S19
FEEDBACK_Q_SCALE_CODE를 사용해 양자화를 실행
S20
모든 매크로블록이 처리되었나?
NO
YES
S17
FEEDBACK_Q_SCALE_CODE를 사용해 양자화를 실행

도면37

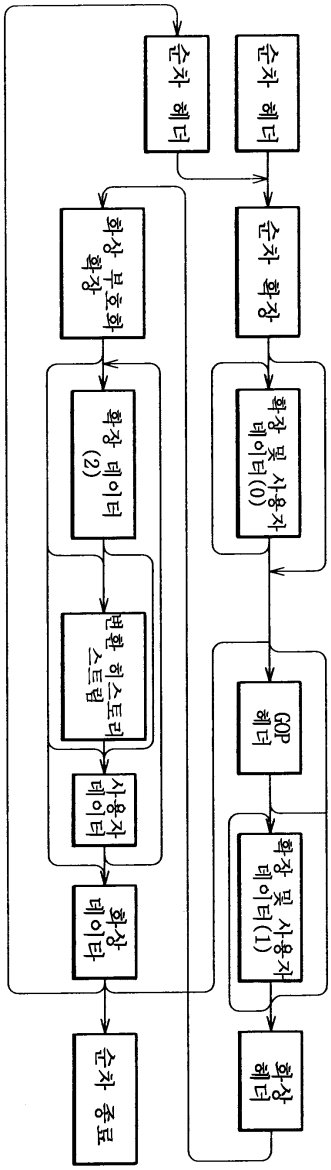

ST(3RD)
비디오 복호화
102
히스토리 복호화
104
부호화 매개 변수 (1ST, 2ND)
107
히스토리 부호화
USER_DATA (1ST,2ND)
부호화 매개 변수(3RD)
기저대 비디오 신호
비디오 부호화
106
ST(4TH)
USER_DATA (3RD)

도면38

히스토리 데이터를 갖춘 스트림

video_sequence( ){	비트수	연상 기호
next_start_code( )		
sequence_header( )		
sequence_extension( )		
do{		
extension_and_user_data(0)		
do{		
if(nextbits( )= = group_start_code){		
group_of_pictures_header(1)		
extension_and_user_data(1)		
}		
picture_header( )		
picture_coding_extension( )		
while((nextbits( )= =extension_start_code)||		
(nextbits( )= =user_data_start_code)){		
if(nextbits( )= =extension_start_code))		
extension_data(2)		
if(nextbits( )= =user_data_start_code){		
user_data_start_code	32	bslbf
if(nextbits( )= =History_Data_ID){		
History_Data_ID	8	bslbf
converted_history_stream( )		
}		
else{		
user_data( )		
}		
}		
}		
picture_data( )		
}while((nextbits( )= =picture_start_code)||		
(nextbits( )= =groupe_start_code))		
if(nextbits( )!=sequence_end_code){		
sequence_header( )		
sequence_extension( )		
}		
}while(nextbits( )!=sequence_end_code)		
sequence_end_code	32	bslbf
}		

도면39


순차 헤더
순차 헤더
화상 부호화 확장
순차 확장
확장 데이터 (2)
확장 및 사용자 데이터(0)
변환 히스토리 스트림
GOP 헤더
사용자 데이터
확장 및 사용자 데이터(1)
화상 데이타
화상 헤더
순차 종료

도면40

히스토리 스트림(40-1)

history_stream( ){	비트	값
sequence_header		
sequence_header_code	32	000001B3
sequence_header_present_flag	1	
horizontal_size_value	12	
marker_bit	1	1
vertical_size_value	12	
aspect_ratio_information	4	
frame_rate_code	4	
maker_bit	1	1
bit_rate_value	18	
marker_bit	1	1
vbv_buffer_size_value	10	
constrained_parameter_flag	1	0
load_intra_quantiser_matrix	1	
load_non_intra_quantiser_matrix	1	
maker_bits	5	1F
intra_quantiser_matrix[64]	8*64	
non_intra_quantiser_matrix[64]	8*64	
sequence_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	1
sequence_extension_present_flag	1	
profile_and_level_indication	8	
progressive_sequence	1	
chroma_format	2	
horizontal_size_extension	2	
vertical_size_extension	2	
marker_bit	1	1
bit_rate_extension	12	
vbv_buffer_size_extension	8	
low_delay	1	
marker_bit	1	1

도면41

히스토리 스트림(40-2)

	비트	값
frame_rate_extension_n	2	
frame_rate_extension_d	5	
marker_bits	6	3F
sequence_display_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	2
sequence_display_extension_present_flag	1	
video_format	3	
colour_description	1	
colour_primaries	8	
transfer_characteristics	8	
marker_bit	1	1
matrix_coefficients	8	
display_horizontal_size	14	
marker_bit	1	1
display_vertical_size	14	
marker_bit	1	1
macroblock_assignment_in_user_data		
macroblock_assignment_present_flag	1	
marker_bits	7	7F
v_phase	8	
h_phase	8	
group_of_picture_header		
group_start_code	32	000001B8
group_of_picture_header_present_flag	1	
time_code	25	
closed_gop	1	
broken_link	1	
marker_bits	4	F
picture_header		
picture_start_code	32	00000100

도면42

히스토리 스트림(40-3)

	비트	값
temporal_reference	10	
picture_coding_type	3	
marker_bit	1	1
vbv_delay	16	
full_pel_forward_vector	1	
forward_f_code	3	
full_pel_backward_vector	1	
marker_bit	1	1
backward_f_code	3	
marker_bit	1	1
picture_coding_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	8
f_code[0][0]	4	
f_code[0][1]	4	
f_code[1][0]	4	
f_code[1][1]	4	
intra_dc_precision	2	
picture_structure	2	
top_filed_first	1	
frame_pred_frame_dct	1	
concealment_motion_vectors	1	
q_scale_type	1	
marker_bit	1	1
intra_vlc_format	1	
alternate_scan	1	
repeat_first_field	1	
chroma_420_type	1	
progressive_frame	1	
composite_display_flag	1	
v_axis	1	
field_sequence	3	
sub_carrier	1	
burst_amplitude	7	

도면43

히스토리 스트림(40-4)

	비트	값
marker_bit	1	1
sub_carrier_phase	8	
quant_matrix_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	3
quant_matrix_extension_present_flag	1	
load_intra_quantiser_matrix	1	
marker_bits	2	3
intra__quantiser_matrix[64]	8*64	
load_non_intra_quantiser_matrix	1	
marker_bits	7	7F
non_intra__quantiser_matrix[64]	8*64	
load_chroma_intra_quantiser_matrix	1	
marker_bits	7	7F
chorma_intra_quantiser_matrix[64]	8*64	
load_chroma_non_intra_quantiser_matrix	1	
marker_bits	7	7F
chroma_non_intra_quantiser_matrix[64]	8*64	
copyright_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	4
copyright_extension_present_flag	1	
copyright_flag	1	
copyright_identifier	8	
original_or_copy	1	
marker_bit	1	
copyright_number_1	20	
marker_bit	1	
copyright_number_2	22	
marker_bit	1	
copyright_number_3	22	3F
marker_bits	6	

도면44

히스토리 스트림(40-5)

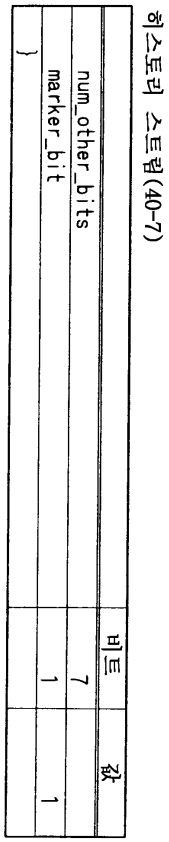
	비트	값
picture_display_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	7
picture_display_extension_present_flag	1	
frame_centre_horizontal_offset_1	16	
marker_bit	1	1
frame_centre_vertical_offset_1	16	
marker_bit	1	1
frame_centre_horizontal_offset_2	16	
marker_bit	1	1
frame_centre_vertical_offset_2	16	
marker_bit	1	1
frame_centre_horizontal_offset_3	16	
marker_bit	1	1
frame_centre_vertical_offset_3	16	
marker_bits	6	3F
user_data		
user_data_start_code	32	000001B2
user_data	2048	
while(macroblock i=macroblock_count){		
macroblock		
macroblock_address_h	8	
macroblock_address_v	8	
slice_header_present_flag	1	
skipped_macroblock_flag	1	
marker_bit	1	1
macroblock_modes( )		
macroblock_quant	1	
macroblock_motion_forward	1	
macroblock_motion_backward	1	
macroblock_pattern	1	
macroblock_intra	1	

도면45

히스토리 스트림(40-6)

	비트	값
spatial_temporal_weight_code_flag	1	
frame_motion_type	2	
field_motion_type	2	
dct_type	1	
marker_bits	2	3
quantiser_scale_code	5	
marker_bits	3	7
PMV[0][0][0]	14	
marker_bits	2	3
PMV[0][0][1]	14	
motion_vertical_field_select[0][0]	1	
marker_bit	1	1
PMV[0][1][0]	14	
marker_bits	2	3
PMV[0][1][1]	14	
motion_vertical_field_select[0][1]	1	
marker_bit	1	1
PMV[1][0][0]	14	
marker_bits	2	3
PMV[1][0][1]	14	
motion_vertical_field_select[1][0]	1	
marker_bit	1	1
PMV[1][1][0]	14	
marker_bits	2	3
PMV[1][1][1]	14	
motion_vertical_field_select[1][1]	1	
marker_bit	1	1
coded_block_pattern	12	
marker_bits	4	F
num_mv_bits	8	
num_coef_bits	14	
marker_bits	2	3

도면46



도면47

history_stream( ){	비트수	연상기호
next_start_code( )		
sequence_header( )		
sequence_extension( )		
extension_and_user_data(0)		
if(nextbits( )== group_start_code){		
group_of_pictures_header( )		
extension_and_user_data(1)		
}		
picture_header( )		
picture_coding_extension( )		
extensions_and_user_data(2)		
picture_data( )		
sequence_end_code	32	bslbf
}		

도면48

sequence_header( ){	비트수	연상기호
sequence_header_code	32	bslbf
horizontal_size_value	12	uimsbf
vertical_size_value	12	uimsbf
aspect_ratio_information	4	uimsbf
frame_rate_code	4	uimsbf
bit_rate_value	18	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_value	10	uimsbf
constrained_parameters_flag	1	bslbf
load_intra_quantiser_matrix	1	uimsbf
if(load_intra_quantiser_matrix)		
intra_quantiser_matrix[64]	8*64	uimsbf
load_non_intra_quantiser_matrix	1	uimsbf
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix[64]	8*64	uimsbf
next_start_code( )		
}		

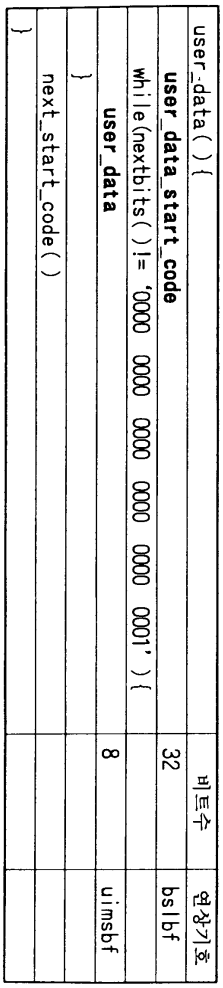
도면49

sequence_extension( ){	비트수	연상기호
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
profile_and_level_indication	8	uimsbf
progressive_sequence	1	uimsbf
chroma_format	2	uimsbf
horizontal_size_extension	2	uimsbf
vertical_size_extension	2	uimsbf
bit_rate_extension	12	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	uimsbf
low_delay	1	uimsbf
frame_rate_extension_n	2	uimsbf
frame_rate_extension_d	5	uimsbf
next_start_code( )		
}		

도면50

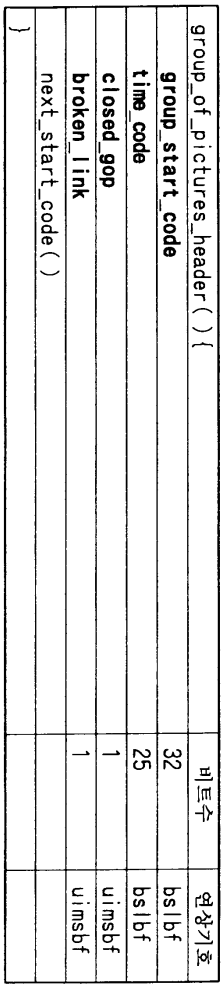
	비트수	연상기호
extension_and_user_data(i) {		
while((nextbits( ) == extension_start_code) ||		
(nextbits( ) == user_data_start_code)) {		
if((i == 2) && (nextbits( ) == extension_start_code))		
extension_data( )		
if(nextbits( ) == user_data_start_code)		
user_data( )		
}		
}		

도면51



user_data( ) {	비트수	연창기호
user_data_start_code	32	bslbf
while(nextbits( )!= '0000 0000 0000 0000 0000 0001'){		
user_data	8	uimsbf
}		
next_start_code( )		
}		

도면52



group_of_pictures_header( ){		비트수	연상기호
group_start_code		32	bslbf
time_code		25	bslbf
closed_gop		1	uimsbf
broken_link		1	uimsbf
next_start_code( )			
}			

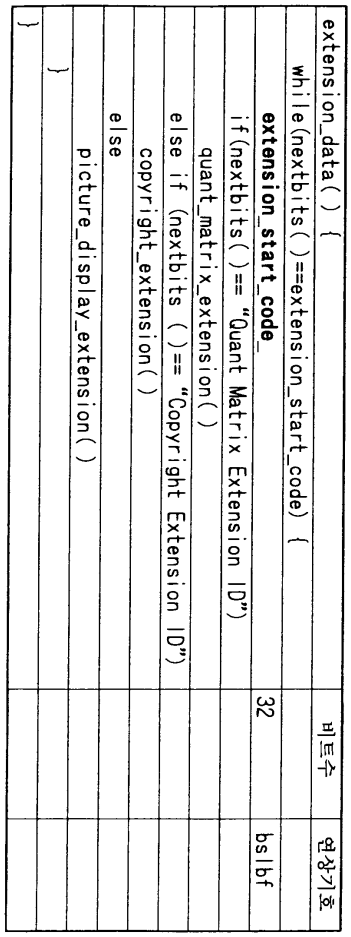
도면53

picture_header( ){	비트수	연상기호
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if(picture_coding_type==2 || picture_coding_type==3{		
full_pel_forward_vector	1	bslbf
forward_f_code	3	bslbf
}		
if(picture_coding_type==3) {		
full_pel_backward_vector	1	bslbf
backward_f_code	3	bslbf
}		
while(nextbits( )=='1') {		
extra_bit_picture/*with the value'1'*/	1	uimsbf
extra_information_picture	8	uimsbf
}		
extra_bit_picture/*with the value'0'*/	1	uimsbf
next_start_code( )		
}		

도면54

picture_coding_extension( ) {	비트수	연상기호
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
f_code [0] [0] /*forward horizontal*/	4	uimsbf
f_code [0] [1] /*forward vertical*/	4	uimsbf
f_code [1] [0] /*backward horizontal*/	4	uimsbf
f_code [1] [1] /*backward vertical*/	4	uimsbf
intra_dc_precision	2	uimsbf
picture_structure	2	uimsbf
top_field_first	1	uimsbf
frame_pred_frame_dct	1	uimsbf
concealment_motion_vectors	1	uimsbf
q_scale_type	1	uimsbf
intra_vlc_format	1	uimsbf
alternate_scan	1	uimsbf
repeat_first_field	1	uimsbf
chroma_420_type	1	uimsbf
progressive_frame	1	uimsbf
composite_display_flag	1	uimsbf
if(composite_display_flag) {		
v_axis	1	uimsbf
field_sequence	3	uimsbf
sub_carrier	1	uimsbf
burst_amplitude	7	uimsbf
sub_carrier_phase	8	uimsbf
}		
next_start_code()		
}		

도면55



extension_data( ) {	비트수	연상기호
while(nextbits( )==extension_start_code) {		
extension_start_code	32	bslbf
if(nextbits( )== "Quant Matrix Extension ID")		
quant_matrix_extension( )		
else if (nextbits ( )== "Copyright Extension ID")		
copyright_extension( )		
else		
picture_display_extension( )		
}		
}		

도면56

quant_matrix_extension( ) {	비트수	연상기호
extension_start_code_identifier	4	uimsbf
load_intra_quantiser_matrix	1	uimsbf
if(load_intra_quantiser_matrix)		
intra_quantiser_matrix [64]	8*64	uimsbf
load_non_intra_quantiser_matrix	1	uimsbf
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix [64]	8*64	uimsbf
load_chroma_intra_quantiser_matrix	1	uimsbf
if(load_chroma_intra_quantiser_matrix)		
chroma_intra_quantiser_matrix [64]	8*64	uimsbf
load_chroma_non_intra_quantiser_matrix	1	uimsbf
if(load_chroma_non_intra_quantiser_matrix)		
chroma_non_intra_quantiser_matrix [64]	8*64	uimsbf
next_start_code( )		
}		

도면57

copyright_extension( ){	비트수	연상기호
extension_start_code_identifier	4	uimsbf
copyright_flag	1	bslbf
copyright_identifier	8	uimsbf
original_or_copy	1	bslbf
reserved	7	uimsbf
marker_bit	1	bslbf
copyright_number_1	20	uimsbf
marker_bit	1	bslbf
copyright_number_2	22	uimsbf
marker_bit	1	bslbf
copyright_number_3	22	uimsbf
next_start_code( )		
}		

도면58

picture_display_extension( ){	비트수	연상기호
extension_start_code_identifier	4	uimsbf
for(i=0;ijnumber_of_frame_centre_offsets;i++){		
frame_centre_horizontal_offset	16	simsbf
marker_bit	1	bslbf
frame_centre_vertical_offset	16	simsbf
marker_bit	1	bslbf
}		
next_start_code( )		
}		

도면59

picture_data(){	비트수	연상기호
while(nextbits()==slice_start_code){		
slice()		
}		
next_start_code()		
}		

도면60

slice( ){		비트수	연상기호
slice_start_code		32	bslbf
slice_quantiser_scale_code		5	uimsbf
if(nextbit( )=='1'){			
intra_slice_flag		1	bslbf
intra_slice		1	uimsbf
reserved_bits		7	uimsbf
while(nextbits( )=='1'){			
extra_bit_slice/*with the value'1'*/		1	uimsbf
extra_information_slice		8	uimsbf
}			
}			
extra_bit_slice/*with the value'0'*/		1	uimsbf
do{			
macroblock( )			
}while(nextbit( )!='000 0000 0000 0000 0000 0000')			
next_start_code( )			
}			

도면61

macroblock(){		비트수	연상기호
while(nextbits()== '0000 0001 000')			
macroblock_escape		11	bslbf
macroblock_address_increment		1-11	vlclbf
macroblock_modes()			
if(macroblock_quant)			
macroblock_quantiser_scale_code		5	uimsbf
if(macroblock_motion_forward ||			
(macroblock_intra **&&** concealment_motion_vectors))			
motion_vectors(0)			
if(macroblock_motion_backward)			
motion_vectors(1)			
if(macroblock_intra **&&** concealment_motion_vectors)			
marker_bit		1	bslbf
}			
}			

도면62

macroblock_modes() {	비트수	연상기호
macroblock_type	1-9	vlclbf
if(macroblock_motion_forward ||		
macroblock_motion_backward) {		
if(picture_structure=='frame') {		
if(frame_pred_frame_dct==0)		
frame_motion_type	2	uimsbf
}else{		
field_motion_type	2	uimsbf
}		
}		
if((picture_structure=="Frame picture")&&		
(frame_pred_frame_dct==0)&&		
(dct_type_flag==1)) {		
dct_type	1	uimsbf
}		
}		

도면63

motion_vectors(s) {	비트수	연상기호
if(motion_vector_count==1) {		
if((mv_format==field)&&(dmv!=1)		
motion_vertical_field_select[0][s]	1	uimsbf
motion_vector(0,s)		
}else{		
motion_vertical_field_select[0][s]	1	uimsbf
motion_vector(0,s)		
motion_vertical_field_select[1][s]	1	uimsbf
motion_vector(1,s)		
}		
}		

도면64

motion_vector(r,s) {	비트수	연상기호
motion_code[r][s][0]	1-11	vlclbf
if((f_code[s][0]!=1)&&(motion_code[r][s][0]!=0))		
motion_residual[r][s][0]	1-8	uimsbf
if(dmv==1)		
dmvector[0]	1-2	vlclbf
motion_code[r][s][1]	1-11	vlclbf
if(f_code[s][1]!=1)&&(motion_code[r][s][1]!=0))		
motion_residual[r][s][1]	1-8	uimsbf
if(dmv==1)		
dmvector[1]	1-2	vlclbf
}		

도면65

macroblock_type VLC code	macroblock_quant	dct_type_flag	macroblock_motion_forward	macroblock_motion_backward	Description
1	0	1	0	0	Intra
01	1	1	0	0	Intra, Quant

도면66

macroblock_type VLC code	macroblock_quant	dct_type_flag	macroblock_motion_forward	macroblock_motion_backward	Description
1	0	1	1	0	MC, Coded
01	0	1	0	0	No MC, Coded
001	0	0	0	0	MC, Not Coded
0001 1	0	1	0	0	Intra
0001 0	1	1	1	0	MC, Coded, Quant
0000 1	1	1	0	0	No MC, Coded, Quant
0000 01	1	1	0	0	Intra, Quant

도면67

macroblock_type VLC code	macroblock_quant	dct_type_flag	macroblock_motion_forward	macroblock_motion_backward	Description
10	0	0	0	0	Interp, Not Coded
11	0	1	1	1	Interp, Coded
010	0	0	0	0	Bwd, Not Coded
011	0	1	0	1	Bwd, Coded
0010	0	0	0	0	Fwd, Not Coded
0011	0	1	1	0	Fwd, Coded
0001 1	0	1	0	0	Intra
0001 0	1	1	1	1	Interp, Coded, Quant
0000 11	1	1	1	0	Fwd, Coded, Quant
0000 10	1	1	0	1	Bwd, Coded, Quant
0000 01	1	1	0	0	Intra, Quant