(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2010 년 7 월 29 일 (29.07.2010)

PCT

(10) 국제공개번호
WO 2010/085064 A2

(51) 국제특허분류:
H04N 7/32 (2006.01) *H04N 7/24* (2006.01)

(21) 국제출원번호: PCT/KR2010/000248

(22) 국제출원일: 2010 년 1 월 15 일 (15.01.2010)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
10-2009-0006308 2009 년 1 월 23 일 (23.01.2009) KR

(71) 출원인 (US 을(를) 제외한 모든 지정국에 대하여): **에스케이텔레콤 주식회사 (SK TELECOM CO., LTD.)** [KR/KR]; 서울 중구 을지로 2 가 11 번지, 100-999 Seoul (KR).

(72) 발명자; 겸

(75) 발명자/출원인 (US 에 한하여): **김수년 (KIM, Sunyeon)** [KR/KR]; 서울 송파구 잠실동 잠실엘스아파트 158 동 2603 호, 138-220 Seoul (KR). **전병우 (JEON, Byeungwoo)** [KR/KR]; 경기도 성남시 분당구 수내동 양지마을 한양아파트 527 동 1302 호, 463-922 Gyeonggi-do (KR). **양 정 엽 (YANG, Jungyoup)** [KR/KR]; 서울시 금천구 독산동 1140 번지 독산동 중앙하이츠빌 105 동 1702 호, 153-010 Seoul (KR). **원광현 (WON, Kwanghyun)** [KR/KR]; 경기도 부천시 원미구 중동 동원아파트 가동 301 호, 420-020 Gyeonggi-do (KR). **문주희 (MOON, Joohee)** [KR/KR]; 서울 강남구 삼성 2 동 14-1 삼성중앙하이츠빌리지 101 동 903 호, 135-092 Seoul (KR). **이영렬 (LEE, Yunglyul)** [KR/KR]; 서울 송파구 잠실 5 동 주공아파트 519 동 510 호, 138-917 Seoul (KR). **김 해 광 (KIM, Haekwang)** [KR/KR]; 서울 광진구 광장동 극동아파트 11 동 1303 호, 143-751 Seoul (KR).

(74) 대리인: **이철희 (LEE, Chulhee)**; 서울 강남구 역삼동 647-13 동궁빌딩 5 층, 135-911 Seoul (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), 유럽 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서 없이 공개하며 보고서 접수 후 이를 별도 공개함 (규칙 48.2(g))

(54) Title: APPARATUS AND METHOD FOR MOTION VECTOR ENCODING/DECODING, AND APPARATUS AND METHOD FOR IMAGE ENCODING/DECODING USING SAME

(54) 발명의 명칭 : 움직임 벡터 부호화/복호화 장치 및 방법과 그를 이용한 영상 부호화/복호화 장치 및 방법

(57) Abstract: The present invention relates to an apparatus and to a method for motion vector encoding/decoding, and to an apparatus and to a method for image encoding/decoding using same. The motion vector encoding apparatus of the present invention comprises: a predicted motion vector selector for selecting an optimum predicted motion vector from among the candidate motion vector group of the current block; a prediction mode determiner for selecting one motion vector prediction mode from among a plurality of motion vector prediction modes on the basis of the determination criteria previously agreed upon by a motion vector decoding apparatus, and determining the selected motion vector prediction mode as a motion vector prediction mode of the current block; and an encoder for determining the optimum predicted motion vector or a default predicted motion vector as a predicted motion vector for the current motion vector in accordance with the determined motion vector prediction mode, encoding a differential vector between the current block and the determined predicted motion vector, and selectively encoding the determined motion vector prediction mode in accordance with the determined motion vector prediction mode and outputting the encoded motion vector prediction mode in accordance with the determined motion vector prediction mode. The present invention reduces bit rates required for encoding a motion vector to thereby improve encoding efficiency, and prevents failures in the encoder even when an error occurs during encoding thereby improving the performance of the encoder.

(57) 요약서: 본 발명은 움직임 벡터 부호화/복호화 장치 및 방법과 그를 이용한 영상 부호화/복호화 장치 및 방법에 관한 것이다. 본 발명은 현재 블록의 후보 움직임 벡터 집합에서 최적 예측 움직임 벡터를 선택하는 예측 움직임 벡터 선택기; 움직임 벡터 복호화 장치와 미리 약속된 판단 기준에 근거하여 복수 개의 움직임 벡터 예측 모드 중 하나의 움직임 벡터 예측 모드를 선택하여 현재 블록의 움직임 벡터 예측 모드로서 결정하는 예측 모드 결정기; 및 결정된 움직임 벡터 예측 모드에 따라 최적 예측 움직임 벡터 또는 디폴트 움직임 예측 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로서 결정하고 현재 블록과 결정된 예측 움직임 벡터 간의 차분 벡터를 부호화하며, 결정된 움직임 벡터 예측 모드에 따라 결정된 움직임 벡터 예측 모드를 선택적으로 부호화하여 출력하는 부호화기를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 장치. 본 발명에 의하면, 움직임 벡터를 부호화하는 데 소요되는 비트량을 줄일 수 있어 부호화 효율을 향상시킬 수 있을 뿐만 아니라, 복호화 과정에서 오류가 발생한 경우에도 복호화기 장애가 발생하는 것을 방지하여 복호화기의 성능을 향상시킬 수 있다.

WO 2010/085064 A2

명세서

발명의 명칭: 움직임 벡터 부호화/복호화 장치 및 방법과 그를 이용한 영상 부호화/복호화 장치 및 방법

기술분야

[1] 본 발명은 움직임 벡터 부호화/복호화 장치 및 방법과 그를 이용한 영상 부호화/복호화 장치 및 방법에 관한 것이다. 더욱 상세하게는, 동영상을 부호화 및 복호화하는 영상 처리 분야에서 예측을 위한 움직임 벡터를 더욱 정확한 움직임 벡터로 선택하고 적은 비트량으로 부호화하여 부호화 효율 및 압축 효율을 향상시키는 방법 및 장치에 관한 것이다.

배경기술

[2] 근래에 멀티미디어 기술은 비약적으로 발전하고 있으며, 이에 따라 오디오, 이미지, 동영상 등을 포함한 고품질의 멀티미디어 데이터들에 대한 수요도 증가하고 있다. 따라서 제한된 네트워크 환경에서 이러한 멀티미디어 데이터들을 전송하거나 저장하고 다시 읽어내는 (Retrieve) 수요를 충족시키기 위해 고효율의 동영상 압축에 관한 국제 표준이 제정되고 있다. 특히 동영상의 경우, 동영상 압축에 관한 국제표준으로 ISO/IEC JTC1/SC29 MPEG 그룹 및 ITU-T VCEG 그룹이 제정한 H.264/AVC MPEG-4 Part.10 표준은 높은 압축 효율을 달성하기 위해 가변블록 크기에서의 움직임예측 및 보상(Variable block size motion estimation and compensation), 화면 내 예측부호화(Intra prediction) 등의 다양한 예측부호화 방법들을 사용하고 있다. 예측부호화는 데이터간에 존재하는 상관도(Correlation)을 줄일 수 있는 효과적인 방법으로 다양한 데이터들의 압축에 널리 사용된다. 특히 움직임 벡터는 주변 블록의 움직임 벡터와 밀접한 상관관계를 가지고 있기 때문에, 주변 블록의 움직임 벡터로부터 현재 블록의 움직임 벡터에 대한 예측값(PMV: Predicted Motion Vector, "예측 움직임 벡터"라고도 함)을 계산한 후 현재 블록의 움직임 벡터의 값 그 자체를 부호화하지 않고 그 예측값에 대한 차분 벡터(DMV: Differential Motion Vector, "잔차신호", "차분값"이라고도 함)만을 부호화함으로써 부호화해야 할 비트량을 상당히 줄여 부호화 효율을 높힐 수 있다.

[3] 일반적으로 이러한 예측 움직임 벡터를 이용한 움직임 벡터 부호화에서는 효율적인 압축을 위하여 예측 움직임 벡터가 정확할수록 부호화 효율이 증대된다. 따라서 단지 공간적 인접블록의 움직임 벡터뿐만 아니라, 시간적, 공간적 또는 시공간적으로 인접한 블록의 움직임 벡터 또는 그들을 조합하여 계산된 또 다른 움직임 벡터들로 이루어진 유한한 개수의 예측 움직임 벡터 대상들을 생성하고 그 중에서 움직임 벡터 예측부호화에 가장 적당한 것을 선택하여 사용하면 예측부호화 효율을 더욱 높일 수 있다. 이 경우, 예측 부호화된 움직임 벡터 데이터로부터 원래의 움직임 벡터를 올바르게 복원하기

위해서는 유한한 개수의 예측 움직임 벡터 대상들 중에 어떤 예측 움직임 벡터가 사용되었는지 알아야 한다. 이를 위한 가장 간단한 움직임 벡터 예측부호화 방법은 움직임 벡터 예측부호화를 위하여 어떤 예측값을 사용하였는지에 대한 정보를 함께 부호화하는 것이다. 또는 이러한 예측 움직임 벡터 선택을 가리키기 위한 추가적인 데이터 부호화에 소요되는 비트량 발생을 제거하기 위해, 현재의 H.264/AVC 표준은 주변 블록(좌측, 상단, 우측상단)이 가지는 움직임 벡터들의 수평 성분과 수직 성분 각각의 중간값(Median)을 움직임 벡터의 예측 부호화를 위한 예측 움직임 벡터(PMV: Predicted Motion Vector)로 사용한다. 이 방법은 중간값이라는 부호화와 복호화에 있어 공히 알고 있는 소정의 디폴트 방법을 정하고, 부호화나 복호화에 있어 이 디폴트 방법을 사용하여 예측값(예측 움직임 벡터)을 산출함으로써 어떤 예측값을 사용하였는지에 대한 정보를 함께 부호화할 필요를 없애는 것이다. 중간값의 소정의 디폴트 방법을 사전에 정의하여 사용하는 기존 방법은, 어떤 움직임 벡터가 예측 움직임 벡터로 사용되었는지에 대한 추가적인 정보의 전송 없이 향상된 부호화 효율을 유지할 수 있다는 장점이 있지만 사용한 중간값인 예측 움직임 벡터가 차분 벡터를 부호화하는데 소요되는 비트량을 항상 최소로 발생시키는 최적의 예측 움직임 벡터가 아니라는 문제점이 있다.

[4] 일반적으로 동영상 압축 부호화 방법에서는 보다 정교한 다양한 부호화 방법을 구비하여 소정의 부호화 기준에 따라 경쟁시킨 후, 그 중에서 소정의 기준에 따라 평가된 최적의 부호화 효율을 보이는 부호화 방법을 선택함으로써 부호화 효율을 증대시키고 있다. 이렇게 압축된 데이터는 부호화기와 복호화기 사이에 기 약속된 소정의 규정(또는 규약)에 따라 비트스트림으로 저장 또는 전송되는데, 이때 비트스트림을 구성하는 각각의 요소를 신택스요소(Syntax Element)라 한다. 예를 들어, 시간적 중복성을 제거하는 움직임 보상 방법을 이용하여 영상을 압축부호화할때 부호화기는 움직임 벡터를 부호화하여야 하는데 탐색범위내에있는 움직임 벡터들을 다양하게 구비해 두었다가 그중 최적의 것을 찾은 후 상기에 설명한 것과 같이 어떤 예측움직임 벡터를 사용했는지 여부를 복호화기에 시그널링 한다. 이 경우 어떤 예측움직임 벡터를 사용했는지 여부를 알리는 정보를 신택스요소(Syntax Element)의 한 예로 볼 수 잇다. 또는, 어떤 예측움직임 벡터를 사용했는지 여부를 그대로 알려주는 대신, 중간값과 같이 소정의 미리 정의된 어떤 예측 움직임 벡터와의 차이를 부호화 하게 할 수도 있다. 또는 적응적으로 어떤 경우에는 소정의 어떤 예측움직임 벡터를 사용하고, 다른 경우에는 또 다른 예측움직임 벡터를 사용하게 할 수 있다. 이 경우는 또한 예측움직임 벡터 선택을 어떻게 하여야 할 것인가를 복호화기에 알려주어야 하는데 이를 알리는 정보 또한 신택스요소(Syntax Element)의 한 예가 될 수 있다.

[5] 보다 다양하고 정교한 부호화 방법을 사용하여 압축한 데이터를 복호화기가 제대로 복호화하려면 비트스트림에 많은 신택스요소(Syntax Element) 정보가

추가되어야 한다. 이 경우, 이러한 신택스요소(Syntax Element) 정보를 전송 또는 저장하기 위해 발생하는 비트량이 증가하여 정지영상 또는 동영상을 부호화하는데 필요한 데이터 량을 증가시킨다. 뿐만 아니라, 보다 다양하고 정교한 부호화 방법을 사용함으로써 부호화효율을 증가시킬 수 있지만, 어떤 부호화 방법을 어떻게 사용하였는지 여부를 복호화기에 알려주어야 적정한 복호화가 가능하다. 따라서 이런한 정보를 전송 또는 저장 하여야만 하는데, 이때 정교한 부호화방법을 사용하여 얻은 부호화 효율증가 보다도 복호화기에 어떤 부호화방법을 사용했는지 여부를 알려주거나 표현하는 오버헤드가 더 많은 비트를 차지 한다면 즉, 이를 시그널링하는 신택스요소(Syntax Element)의 증가 효과가 더 크다면 오히려 압축 성능을 저하시키는 원인이 되기도 한다.

[6] 이러한 문제를 해결하기 위한 하나의 바람직한 해결책은 부호화기가 소정의 부호화 기준에 따라 결정한 신택스요소(Syntax Element)의 값을, 복호화기가 복호화 과정에서 자체적인 소정의 신택스요소(Syntax Element) 추정 과정을 통해 복호화기가 스스로 추정함으로써, 부호화기가 복호화기로 해당 신택스요소(Syntax Element)값을 저장 또는 전송하지 않아도 되도록 하는 것이다. 그러나 복호화기는 다양한 판단 및 결정을 수행할 수 있는 부호화기에 비해 매우 제한적인 추정 과정만을 수행할 수 있으므로, 모든 경우에 이러한 방법을 적용할 수는 없다는 단점이 있다. 따라서 부호화기가 결정한 신택스요소(Syntax Element)를 복호화기가 동일하게 스스로 추정할 수 있는 일부의 경우에만 이러한 방법을 선택적으로 사용할 수 있다. 그러나 일부경우에만 해당 신택스요소를 전송하지않고 나머지 경우에는 전송하는 선택적 신택스전송의 방법은, 기 복호화된 영상의 값이나 복호화 과정을 자체를 이용한 소정의 신택스요소(Syntax Element) 추정 과정을 통해 신택스요소(Syntax Element)의 존재 유무를 복호화기가 판단해야 하므로 신택스요소(Syntax Element) 분석(Parsing) 과정과 복호화(Decoding) 과정이 분리될 수 없다는 문제점이 발생한다. 뿐만 아니라 신택스요소(Syntax Element)의 존재 유무를 판단하는 소정의 신택스요소(Syntax Element) 추정 과정에 필요한 복호화결과나 복호화 과정에 오류가 포함될 경우, 신택스요소(Syntax Element)의 존재 유무를 정상적으로 판단할 수 없으므로, 해당 신택스요소를 보내지 않았는데도 복호화기는 이를 읽으려고(parsing) 하거나, 해당 신택스요소를 보냈는데도 복호화기는 이를 읽지(parsing) 않아도된다고 판단하는 경우가 생길 수 있어 파싱이나 복호화(decoding) 과정에 심각하고 치명적인 장해를 유발할 수 있다. 이런 이유로 선택적으로 신택스요소를 보내거나 안보내는 방식으로 부호화효율을 높이려는 시도는 아직 보편화되고 있지않다. 그러나 본 발명은 종래의 이러한 문제를 근본적으로 해결하는 방법과 장치를 제공하려 한다.

[7] 따라서 선택적인 신택스요소(Syntax Element)의 전송 및 저장의 생략을 통해 보다 높은 부호화 효율을 달성하는 동시에 상기와 같은 문제점을 해결하기 위해서는 이러한 신택스요소(Syntax Element)를 위한 새로운 신택스요소(Syntax

Element) 부호화/복호화 방법이 필요하다. 본 발명의 기술분야는 상기와 같은 목적을 달성하기 위해 사용되는 정지영상 및 동영상의 신택스요소(Syntax Element) 부호화/복호화 방법과 그 장치에 관한 것이다.

발명의 상세한 설명

기술적 과제

[8] 전술한 문제점을 해결하기 위한 본 발명의 목적은, 복수의 움직임 벡터 예측 모드를 이용하여 현재 움직임 벡터에 대한 예측 움직임 벡터의 선택이 가능하도록 하여 더욱 정확하게 예측 움직임 벡터를 선택하고, 이를 통해 차분 벡터의 부호화 비트량을 감소시키고 움직임 예측 및/또는 움직임 보상의 성능을 향상시키는 데 있다.

[9] 또한, 본 발명은, 부호화 장치는 더욱 정확한 예측 움직임 벡터를 선택하여 부호화 효율을 향상시키면서도 부호화에 이용된 예측 움직임 벡터를 직접 복호화 장치로 알리는 대신에, 예측 움직임 벡터를 찾기 위한 정보를 복호화 장치로 주거나 찾기 위한 함수를 복호화 장치와 공유함으로써, 예측 움직임 벡터를 알리기 위해 발생하는 추가적인 정보에 대한 부호화 비트량 증가를 줄여줌으로써, 부호화 효율 및 복호화 효율을 더욱 향상시키는 데 있다.

[10] 또한, 본 발명은 영상의 비트스트림을 구성하는 신택스 요소 중 복호화된 결과에 따라 신택스가 비트스트림에 존재하는지 여부를 알수 있는 경우, 소정의 단위 내에 있는 존재하는(또는 비존재하는) 경우의 개수와 그때의 값을 복호화기에 별도로 알려주도록 하여 신택스 요소(Syntax Element) 파싱(Parsing) 과정과 복호화(Decoding) 과정이 분리될 수 있도록 하는 데 있다.

과제 해결 수단

[11] 전술한 목적을 달성하기 위해 본 발명은, 움직임 벡터를 부호화하는 장치에 있어서, 현재 블록의 후보 움직임 벡터 집합에서 최적 예측 움직임 벡터를 선택하는 예측 움직임 벡터 선택기; 움직임 벡터 복호화 장치와 미리 약속된 판단 기준에 근거하여 복수 개의 움직임 벡터 예측 모드 중 하나의 움직임 벡터 예측 모드를 선택하여 현재 블록의 움직임 벡터 예측 모드로서 결정하는 예측 모드 결정기; 및 결정된 움직임 벡터 예측 모드에 따라 최적 예측 움직임 벡터 또는 디폴트 움직임 예측 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로서 결정하고 현재 블록과 결정된 예측 움직임 벡터 간의 차분 벡터를 부호화하며, 결정된 움직임 벡터 예측 모드에 따라 결정된 움직임 벡터 예측 모드를 선택적으로 부호화하여 출력하는 부호화기를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 장치.

[12] 여기서, 복수 개의 움직임 벡터 예측 모드는 경쟁 예측 모드 및 비경쟁 예측 모드를 포함하며, 예측 모드 결정기는 움직임 벡터 복호화 장치와 미리 약속된 판단 기준에 근거하여 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하여 경쟁 예측 모드 또는 비경쟁 예측 모드를 현재 블록의

움직임 벡터 예측 모드로서 선택할 수 있다.

[13] 또한, 예측 모드 결정기는 최적 예측 움직임 벡터가 디폴트 예측 움직임 벡터와 동일한 경우, 비경쟁 예측 모드를 현재 블록의 움직임 벡터 예측 모드로서 결정할 수 있다. 이때, 최적 예측 움직임 벡터가 디폴트 예측 움직임 벡터와 동일한 경우는 후보 움직임 벡터 집합 내에 현재 블록의 현재 움직임 벡터에 대하여 예측 움직임 벡터로 사용할 수 있는 후보 움직임 벡터가 없는 경우, 후보 움직임 벡터가 한 개인 경우, 후보 움직임 벡터가 두 개 이상이지만 모두 동일한 경우 및 두 개 이상이고 모두 동일하지는 않은 후보 움직임 벡터 중에서 선택되는 최적 예측 움직임 벡터가 디폴트 예측 움직임 벡터와 동일한 경우 중 하나 이상일 수 있다.

[14] 또한, 부호화기는 결정된 움직임 벡터 예측 모드가 비경쟁 예측 모드인 경우, 디폴트 움직임 예측 벡터를 예측 움직임 벡터로서 결정하여 차분 벡터를 생성하고 부호화하는 제 1 부호화기; 결정된 움직임 벡터 예측 모드가 경쟁 예측 모드인 경우, 움직임 벡터 복호화 장치가 최적 예측 움직임 벡터를 예측할 수 있는지 여부에 따라 최적 예측 움직임 벡터 또는 디폴트 예측 움직임 벡터를 예측 움직임 벡터로서 결정하여 차분 벡터를 생성하고 부호화하는 제 2 부호화기; 및 결정된 움직임 벡터 예측 모드가 경쟁 예측 모드인 경우에만 결정된 움직임 벡터 예측 모드를 부호화하는 제 3 부호화기를 포함할 수 있다.

[15] 또한, 결정된 움직임 벡터 예측 모드는 한 개의 비트로 구성되며, 비트값에 따라 경쟁 예측 모드 또는 비경쟁 예측 모드를 나타낼 수 있으며, 부호화된 움직임 벡터 예측 모드는 비트스트림의 슬라이스 헤더 또는 픽처 헤더에 삽입될 수 있다.

[16] 또한, 본 발명의 다른 목적에 의하면, 움직임 벡터를 부호화하는 방법에 있어서, 움직임 벡터 복호화 장치와 미리 약속된 판단 기준에 근거하여 복수 개의 움직임 벡터 예측 모드 중 하나의 움직임 벡터 예측 모드를 선택하여 현재 블록의 움직임 벡터 예측 모드로서 결정하는 단계; 선택된 움직임 벡터 예측 모드에 따라 현재 움직임 벡터에 대한 예측 움직임 벡터를 결정하는 단계; 현재 움직임 벡터와 결정된 예측 움직임 벡터 간의 차분 벡터를 부호화하는 단계; 결정된 움직임 벡터 예측 모드에 따라 결정된 움직임 벡터 예측 모드를 선택적으로 부호화하는 단계; 및 부호화된 차분 벡터 및 부호화된 움직임 벡터 예측 모드를 포함하는 비트스트림을 생성하여 출력하는 단계를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 방법을 제공한다.

[17] 또한, 본 발명의 또 다른 목적에 의하면, 움직임 벡터를 복호화하는 장치에 있어서, 움직임 벡터 부호화 장치와 미리 약속한 판단 기준에 따라 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하고, 부호화된 차분 벡터를 복호화하여 차분 벡터를 복원하고 출력하는 차분 벡터 복호화기; 부호화된 움직임 벡터 예측 모드를 복호화하여 움직임 벡터 예측 모드를 복원하여 출력하는 예측 모드 복호화기; 현재 블록의 움직임 벡터 예측 모드가

비경쟁 예측 모드인 경우, 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 제 1 예측 움직임 벡터 결정기; 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드가 아닌 경우, 복원된 움직임 벡터 예측 모드에 따라 최적 예측 움직임 벡터 또는 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 제 2 예측 움직임 벡터 결정기; 및 결정된 예측 움직임 벡터와 복원된 차분 벡터를 가산하여 현재 움직임 벡터를 복원하는 움직임 벡터 복원기를 포함하는 것을 특징으로 하는 움직임 벡터 복호화 장치를 제공한다.

[18] 또한, 본 발명의 또 다른 목적에 의하면, 움직임 벡터를 복호화하는 방법에 있어서, 부호화된 움직임 벡터 예측 모드 및 부호화된 차분 벡터를 복호화하여 움직임 벡터 예측 모드 및 차분 벡터를 복원하는 단계; 움직임 벡터 부호화 장치와 미리 약속한 판단 기준에 따라 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하는 단계; 판단된 움직임 벡터 예측 모드가 비경쟁 예측 모드인 경우에는 디폴트 예측 움직임 벡터를 현재 블록의 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계; 판단된 움직임 벡터 예측 모드가 비경쟁 예측 모드가 아닌 경쟁 예측 모드인 경우에는 복원된 움직임 벡터 예측 모드에 근거하여 후보 움직임 벡터 집합에서 선택한 최적 예측 움직임 벡터 또는 디폴트 예측 움직임 벡터를 현재 블록의 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계; 및 결정된 예측 움직임 벡터와 차분 벡터를 가산하여 현재 움직임 벡터를 복원하는 단계를 포함하는 것을 특징으로 하는 움직임 벡터 복호화 방법을 제공한다.

[19] 여기서, 부호화된 움직임 벡터 예측 모드는 움직임 벡터 예측 모드의 개수를 나타내는 비트 및 개수만큼의 움직임 벡터 예측 모드를 나타내는 비트를 포함하며, 움직임 벡터 복호화 방법은 현재 슬라이스 또는 현재 픽처 내에서 복원된 모든 블록 중 경쟁 예측 모드에 해당하는 블록의 개수와 움직임 벡터 예측 모드의 개수를 비교하여 모든 움직임 벡터가 정상적으로 복호화되었는지 여부를 판단할 수 있다.

발명의 효과

[20] 이상에서 설명한 바와 같이 본 발명에 의하면, 영상의 블록을 예측할 때 이용되는 예측 움직임 벡터를 실제의 움직임 벡터와 더욱 근접한 움지임 벡터로 선택할 수 있으므로, 움직임 벡터를 부호화하는 데 소요되는 비트량을 줄일 수 있어 부호화 효율을 향상시킬 수 있을 뿐만 아니라, 전송 시의 오류 등 복호화 과정에서 오류가 발생한 경우에도, 복호화 장치가 스스로 오류의 발생을 판단하여 오류를 은닉하거나 오류를 해결하는 후속 조치를 수행할 수 있으므로, 복호화기 장애가 발생하는 것을 방지하여 복호화기의 성능을 향상시킬 수 있다.

도면의 간단한 설명

[21] 도 1은 본 발명의 일 실시예에 따른 움직임 벡터에 대한 부호화를 위한 블록을

나타낸 예시도,

[22] 도 2는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 장치를 간략하게 나타낸 블록 구성도,

[23] 도 3은 본 발명의 일 실시예에 따른 움직임 벡터 복호화 장치를 간략하게 나타낸 블록 구성도,

[24] 도 4는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 방법을 설명하기 위한 순서도,

[25] 도 5는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 방법의 구체화된 구현예를 나타낸 순서도,

[26] 도 6은 본 발명의 일 실시예에 따라 후보 움직임 벡터 집합을 선정하는 과정을 설명하기 위한 순서도,

[27] 도 7은 본 발명의 일 실시예에 따라 예측 움직임 벡터를 선택하는 과정을 설명하기 위한 순서도,

[28] 도 8은 본 발명의 일 실시예에 따라 최적 예측 움직임 벡터의 예측 가능 여부를 판단하는 과정을 설명하기 위한 순서도,

[29] 도 9는 본 발명의 일 실시예에 따라 최적 예측 움직임 벡터를 이용하여 현재 움직임 벡터를 부호화하는 과정을 설명하기 위한 순서도,

[30] 도 10는 본 발명의 일 실시예에 따라 디폴트 예측 움직임 벡터를 이용하여 현재 움직임 벡터를 부호화하는 과정을 설명하기 위한 순서도,

[31] 도 11는 본 발명의 일 실시예에 따라 제 3 부호화를 수행하는 과정을 설명하기 위한 순서도,

[32] 도 12는 본 발명의 일 실시예에 따라 움직임 벡터 예측 모드의 개수 및 움직임 벡터 예측 모드를 모아서 부호화하는 경우의 규정을 설명하기 위한 예시도,

[33] 도 11은 본 발명의 일 실시예에 따른 움직임 벡터 복호화 방법을 설명하기 위한 순서도,

[34] 도 12는 본 발명의 일 실시예에 따른 움직임 벡터 복호화 방법의 구체적인 구현예를 설명하기 위한 순서도,

[35] 도 13은 본 발명의 일 실시예에 따라 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 과정을 설명하기 위한 순서도,

[36] 도 14는 본 발명의 다른 실시예에 따른 움직임 벡터 복호화 방법을 설명하기 위한 순서도,

[37] 도 15는 본 발명의 일 실시예에 따른 영상 부호화 장치를 간략하게 나타낸 블록 구성도,

[38] 도 16은 본 발명의 일 실시예에 따른 영상 부호화 장치의 구체적인 구현 예를 나타낸 예시도,

[39] 도 17은 본 발명의 일 실시예에 따른 영상 복호화 장치를 간략하게 나타낸 블록 구성도,

[40] 도 18은 본 발명의 일 실시예에 따른 영상 복호화 장치의 구체적인 구현 예를

나타낸 예시도이다.

발명의 실시를 위한 형태

[41] 이하, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다. 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 발명을 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.

[42] 또한, 본 발명의 구성 요소를 설명하는 데 있어서, 제 1, 제 2, A, B, (a), (b) 등의 용어를 사용할 수 있다. 이러한 용어는 그 구성 요소를 다른 구성 요소와 구별하기 위한 것일 뿐, 그 용어에 의해 해당 구성 요소의 본질이나 차례 또는 순서 등이 한정되지 않는다. 어떤 구성 요소가 다른 구성요소에 "연결", "결합" 또는 "접속"된다고 기재된 경우, 그 구성 요소는 그 다른 구성요소에 직접적으로 연결되거나 접속될 수 있지만, 각 구성 요소 사이에 또 다른 구성 요소가 "연결", "결합" 또는 "접속"될 수도 있다고 이해되어야 할 것이다.

[43] 이하에서 후술할 영상 부호화 장치와 영상 복호화 장치, 움직임 벡터 부호화 장치와 움직임 벡터 복호화 장치는 개인용 컴퓨터(PC: Personal Computer), 노트북 컴퓨터, 개인 휴대 단말기(PDA: Personal Digital Assistant), 휴대형 멀티미디어 플레이어(PMP: Portable Multimedia Player), 플레이스테이션 포터블(PSP: PlayStation Portable), 이동통신 단말기(Mobile Communication Terminal) 등일 수 있으며, 각종 기기 또는 유무선 통신망과 통신을 수행하기 위한 통신 모뎀 등의 통신 장치, 영상을 부호화하거나 복호화기 위한 각종 프로그램 또는 움직임 벡터를 부호화거나 복호화하기 위한 각종 프로그램과 데이터를 저장하기 위한 메모리, 프로그램을 실행하여 연산 및 제어하기 위한 마이크로프로세서 등을 구비하는 다양한 장치를 의미한다.

[44] 또한, 영상 부호화 장치 또는 움직임 벡터 부호화 장치에 의해 비트스트림으로 부호화된 영상은 실시간 또는 비실시간으로 인터넷, 근거리 무선 통신망, 무선랜망, 와이브로망, 이동통신망 등의 유무선 통신망 등을 통하거나 케이블, 범용 직렬 버스(USB: Universal Serial Bus) 등의 통신 인터페이스를 통해 영상 복호화 장치 또는 움직임 벡터 복호화 장치로 전송될 수 있어 영상 또는 움직임 벡터로 복원되고 재생될 수 있다.

[45] 도 1은 본 발명의 일 실시예에 따른 움직임 벡터에 대한 부호화를 위한 블록을 나타낸 예시도이다.

[46] 도 1에서, 블록 D는 부호화할 움직임 벡터에 해당하는 "현재 블록"이고, 블록 A, 블록 B 및 블록 C는 블록 D에 대한 "주변 블록"로 가정한다. 도 1을 참조하면, MV^A, MV^B, MV^C 및 MV^D는, 블록 A, 블록 B, 블록 C, 블록 D가 갖는 움직임 벡터(Motion Vector: MV)이고, 각각은 수평 성분(MV^A_x, MV^B_x, MV^C_x 및 MV^D_x)과

수직 성분(MV^A_y, MV^B_y, MV^C_y및 MV^D_y)을 가지는 것으로 정의한다. 여기서, 현재 블록인 블록 D의 움직임 벡터 MV^D를 현재 움직임 벡터라 한다.

[47] 수학식 1

$$PMV^D = \begin{bmatrix} PMV^D_x \\ PMV^D_y \end{bmatrix} = \begin{bmatrix} F(MV^A_x, MV^B_x, MV^C_x) \\ F(MV^A_y, MV^B_y, MV^C_y) \end{bmatrix}$$

[48] 수학식 1을 참조하면, 현재 움직임 벡터에 대한 예측 움직임 벡터(PMV^D: Predicted Motion Vector)는 주변 블록(블록 A, 블록 B, 블록 C)의 움직임 벡터(MV^A, MV^B 및 MV^C)를 특정 함수(F())의 변수로 하여 계산됨을 알 수 있다.

[49] H.264/AVC 표준에서, 현재 움직임 벡터에 대한 예측 움직임 벡터(PMV^D: Predicted Motion Vector)는 중간값(Median)을 계산하는 함수를 특정 함수(F())로 하여 계산된다. 즉, 현재 움직임 벡터에 대한 예측 움직임 벡터(PMV^D)는 주변 블록(블록 A, 블록 B, 블록 C)의 움직임 벡터(MV^A, MV^B 및 MV^C)에 대한 중간값(Median)으로 구해지는 것이다. 이러한 방식으로 계산된 현재 움직임 벡터(MV^D)에 대한 예측 움직임 벡터(PMV^D)는 수학식 2와 같이 표현된다.

[50] 수학식 2

$$PMV^D = \begin{bmatrix} PMV^D_x \\ PMV^D_y \end{bmatrix} = \begin{bmatrix} median(MV^A_x, MV^B_x, MV^C_x) \\ median(MV^A_y, MV^B_y, MV^C_y) \end{bmatrix}$$

[51] 수학식 1(또는 수학식 2)를 이용하여 현재 움직임 벡터(MV^D)의 예측 움직임 벡터(PMV^D)가 구해지면, 수학식 3을 사용하여 압축해야 할 현재 움직임 벡터(MV^D)에서 이에 대한 예측 움직임 벡터(PMV^D)를 차분한 움직임 벡터인 차분 벡터(DMV^D: Differential Motion Vector)를 구할 수 있으며, 이 차분 벡터(DMV^D)는 엔트로피 부호화 등의 미리 정의된 소정의 방법에 의해 부호화되어 전송된다.

[52] 수학식 3

$$DMV^D = MV^D - PMV^D = \begin{bmatrix} MV^D_x - PMV^D_x \\ MV^D_y - PMV^D_y \end{bmatrix}$$

[53] 도 1에 예시된 것처럼, 현재 움직임 벡터(MV^D)의 값이 (2,0)인 경우, 중간값을 통해 예측 움직임 벡터(PMV^D)를 계산하는 종래 방식에 따른 수학식 2를 이용하면, 예측 움직임 벡터(PMV^D)는 (2,1)이 된다.

[54] 이와 같이, 중간값을 예측 움직임 벡터로 하는 것은, 부호화 장치와 복호화

장치가 모두 알 수 있도록 함으로써, 어떤 움직임 벡터가 현재 움직임 벡터의 예측 움직임 벡터로 사용되었는지에 대한 추가적인 정보를 부호화하여 전송할 필요가 없으므로 부호화 효율을 향상시킬 수 있다. 여기서, 부호화 장치란 움직임 벡터 부호화 장치와 영상 부호화 장치를 포함하는 개념이며, 복호화 장치란 움직임 벡터 복호화 장치와 영상 복호화 장치를 포함하는 개념이다. 움직임 벡터 부호화 장치, 움직임 벡터 복호화 장치, 영상 부호화 장치 및 영상 복호화 장치에 대해서는 후술하는 과정에서 상세히 설명한다.

[55] 하지만, 전술한 바와 같이, 중간값을 이용하여 계산된 예측 움직임 벡터(PMV^D)는 실제의 현재 움직임 벡터(MV^D)와 다를 수 있다. 도 1의 예시에서 보면, 중간값을 이용하여 계산된 예측 움직임 벡터(PMV^D)인 (2,1)은 현재 움직임 벡터(MV^D)인 (2,0)와 다름을 알 수 있으며, 수학식 3을 이용하여 차분 벡터(DMV^D)를 구하면 부호화할 차분 벡터(DMV^D)는 (0,-1)이 된다.

[56] 만약, 블록 A의 움직임 벡터(MV^A)인 (2,0)을 예측 움직임 벡터(PMV^D)로 사용하면 실제의 현재 움직임 벡터(MV^D)인 (2,0)과 다르지 않고, 수학식 3을 이용하여 차분 벡터(DMV^D)를 구해보면 부호화할 차분 벡터(DMV^D)는 (0,0)이 된다. 즉, 중간값을 이용하여 계산된 예측 움직임 벡터(PMV^D)인 (2,1)를 사용하는 것보다, 블록 A의 움직임 벡터(MV^A)인 (2,0)을 예측 움직임 벡터(PMV^D)로 사용하는 것이 차분 벡터(DMV^D)가 (0,0)이 되어 이를 부호화하는 데 소용되는 비트량을 감소시킬 수 있다.

[57] 하지만, 중간값을 이용하는 방식에서는 현재 움직임 벡터(MV^D)의 예측 움직임 벡터(PMV^D)를 계산하기 위해서 항상 중간값을 사용해야 하므로, 블록 A의 움직임 벡터(MV^A)를 예측 움직임 벡터(PMV^D)로 사용하는 것이 불가능하다. 만약, 블록 A의 움직임 벡터(MV^A)를 예측 움직임 벡터(PMV^D)로 사용한다고 하더라도, MV^A, MV^B 및 MV^C중에서 어떤 움직임 벡터가 예측 움직임 벡터(PMV^D)로 사용되었는지에 대한 추가적인 정보를 함께 전송해야 하므로, 추가적인 정보를 부호화함으로써 압축 효율의 향상 여부를 보장할 수 없는 또 다른 문제점이 있다.

[58] 따라서, 본 발명의 실시예에서는 복수 개의 움직임 벡터 예측 모드를 이용하여 예측 움직임 벡터를 선택할 수 있도록 함으로써 더욱 정확하게 예측 움직임 벡터를 선택할 수 있도록 하고, 이를 통해 중간값을 예측 움직임 벡터로 선택하여 발생하는 문제점을 해결하고자 한다. 또한, 본 발명의 실시예에서는 더욱 정확한 예측 움직임 벡터를 선택하여 부호화 효율을 향상시키면서도, 선택된 예측 움직임 벡터를 복호화 장치에 알리기 위한 효율적인 방법을 제안함으로써, 예측 움직임 벡터를 알리기 위해 발생하는 추가적인 정보의 부호화로 인한 비트량 증가의 문제점도 해결하고자 한다. 또한, 본 발명의 실시예에서는 소정의 경우 비트스트림에서 움직임 벡터 예측 모드를 생략하여 전송 또는 저장하도록 하여 부호화 효율을 향상시키면서도 움직임 벡터 예측 모드가 생략되었는지 여부를 알지 못하여도 복호화 장치에서의 비트스트림의

파싱(Parsing)이 가능하도록 하여 복호화 장치가 안정적으로 동작할 수 있도록 한다. 여기서, 움직임 벡터 예측 모드란 움직임 벡터를 어떠한 방식으로 예측하는지를 식별하기 위한 움직임 벡터의 예측 모드를 말한다.

[59] 도 1을 통해 예시적으로 나타낸, 블록(블록 A, 블록 B, 블록 C 및 블록 D) 및 그에 대한 움직임 벡터인 MV^A, MV^B, MV^C 및 MV^D는 후술하는 과정에서 전반적으로 이용된다. 단, 도 1에서, 움직임 벡터(MV^A, MV^B, MV^C 및 MV^D)를 수직 성분과 수평 성분을 가지는 이차원적인 벡터로 도시하였으나, 이는 설명의 편의를 위한 것이지 이에 한정되지 않고 n차원 움직임 벡터로 확대하여 적용할 수 있다. 또한, 도 1에서는 현재 블록(블록 D)의 주변 블록을 공간적인 인접성에 따라 3개의 블록 A, 블록 B 및 블록 C로만 나타내었으나, 이도 설명의 편의를 위한 것이지 이에 한정되지 않고, 시간적 또는 공간적으로 주변에 있는 하나 이상의 주변 블록이 있을 수 있다.

[60] 본 발명의 일 실시예에 따른 움직임 벡터 예측 모드는 경쟁 예측 모드(Competing Prediction Mode))와 비경쟁 예측 모드(Non-competing Prediction Mode)를 포함한다. 여기서, 경쟁 예측 모드란 현재 움직임 벡터를 예측하는 방식이 여러 가지가 존재하여 서로 경쟁하는 하는 모드를 말하고, 비경쟁 예측 모드란 현재 움직임 벡터를 예측하는 방식이 여러가지가 존재하지 않아서 하나의 방식으로만 예측하는 모드를 말한다.

[61] 현재 움직임 벡터의 예측 움직임 벡터로 사용할 수 있는 후보 움직임 벡터가 없거나, 하나이거나, 하나 이상이라도 모든 후보 움직임 벡터가 모두 동일한 경우, 비경쟁 예측 모드가 움직임 벡터 예측 모드로서 정해진다. 이는, 부호화 장치와 복호화 장치는 상호 간에 어떤 후보 움직임 벡터를 예측 움직임 벡터로서 사용했는지를 식별하기 위한 정보를 추가로 교환하지 않아도 스스로 동일한 예측 움직임 벡터를 찾을 수 있기 때문이다.

[62] 또한, 후보 움직임 벡터가 복수 개이고 서로 다르더라도 소정의 과정을 통해 선택한 최적 예측 움직임 벡터가 수학식 2를 통해 전술한 중간값과 같이 사전에 정해진 디폴트 예측 움직임 벡터와 동일한 경우에도 비경쟁 예측 모드가 움직임 벡터 예측 모드로서 정해진다. 이는 복호화 장치가 스스로 중간값과 같은 사전에 정해진 디폴트 예측 움직임 벡터를 구한 후, 디폴트 예측 움직임 벡터가 소정의 과정을 통해 예측된 최적 예측 움직임 벡터와 동일한지 여부를 스스로 확인할 수 있으며, 디폴트 예측 움직임 벡터와 최적 예측 움직임 벡터가 동일한 경우에는 디폴트 예측 움직임 벡터를 예측 움직임 벡터로 사용하면 된다는 것을 복호화 장치가 추가적인 정보를 부호화 장치와 교환하지 않아도 알 수 있기 때문이다.

[63] 즉, 움직임 벡터 예측 모드가 비경쟁 예측 모드인 경우에는, 복호화 장치는 스스로 움직임 벡터 예측 모드가 비경쟁 예측 모드임을 알 수 있으며, 디폴트 예측 움직임 벡터를 예측 움직임 벡터로 사용하면 된다는 것을 알 수 있다. 따라서, 전술한 비경쟁 예측 모드로 결정되는 경우를 비경쟁 경우라고 하며, 비경쟁 경우에는 복호화 장치가 디폴트 예측 움직임 벡터를 예측 움직임

벡터로서 사용하면 된다는 것과 현재 블록이 비경쟁 예측 모드에 해당하는지 여부를 스스로 알 수 있으므로, 부호화 장치는 현재 블록이 비경쟁 예측 모드에 해당한다는 정보를 추가로 부호화하여 전송하지 않아도 된다. 즉, 이 경우, 움직임 벡터 예측 모드의 부호화 및 전송은 생략될 수 있다.

[64] 한편, 전술한 경우 즉, 비경쟁 경우에 해당하지 않는 경우(이하 '경쟁 경우'라 칭함)에는 복호화 장치가 부호화 장치로부터 추가적인 정보를 받지 않으면 후보 움직임 벡터 중 어떤 움직임 벡터를 예측 움직임 벡터로 사용했는지를 알 수 없으므로, 부호화 장치는 이에 대한 정보를 추가로 부호화하여 전송해야 한다. 이와 같이, 경쟁 경우에 해당하는 움직임 벡터 예측 모드를 경쟁 예측 모드라고 한다. 이 경우, 현재 블록이 경쟁 예측 모드에 해당함을 알리는 정보를 추가로 부호화하여 전송해야 하는데, 추가적인 정보는 현재 움직임 벡터를 복원하기 위해 복호화 장치가 후보 움직임 벡터 중 어떤 움직임 벡터를 예측 움직임 벡터로 사용할 것인지를 알려준다. 예를 들어, 추가적인 정보는 복호화 장치에서 현재 움직임 벡터를 복원하는 데, 후보 움직임 벡터 중 미리 정해진 기준 또는 방법에 따라 선택된 움직임 벡터(최적 예측 움직임 벡터)를 사용할 지 아니면 중간값과 같은 사전에 정의된 디폴트 예측 움직임 벡터를 사용할지를 알려 주는 정보가 될 수 있는데, 이 정보가 움직임 벡터 예측 모드이다.

[65] 여기서, 최적 예측 움직임 벡터라 함은 미리 정해진 기준 또는 방법에 따라 얻어진 현재 움직임 벡터의 예측 움직임 벡터를 의미하는 것일 뿐, 이와 같이 구해진 예측 움직임 벡터가 항상 최적 예측값이라는 것을 의미하는 것은 아니다. 또한, 추가적인 정보가 최적 예측 움직임 벡터를 사용하는 것이라면, 부호화 장치에서 복호화 장치가 최적 예측 움직임 벡터를 예측할 수 있다고 판단한 경우이므로 예측이 가능한 경우(즉, 예측 가능 경우)이며, 움직임 벡터 예측 모드를 1비트의 플래그로 표현하되 플래그를 '1'과 같이 표시할 수 있다. 또한, 추가적인 정보가 디폴트 예측 움직임 벡터를 사용하는 것이라면, 부호화 장치에서 복호화 장치가 최적 예측 움직임 벡터를 예측할 수 없다고 판단한 경우이므로 예측이 불가능한 경우(즉, 예측 불가능 경우)이며, 움직임 벡터 예측 모드를 1비트의 플래그로 표현하되 플래그를 '0'과 같이 표시할 수 있다.

[66] 도 2는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 장치를 간략하게 나타낸 블록 구성도이다.

[67] 도 2를 참조하면, 본 발명의 일 실시예에 따른 움직임 벡터 부호화 장치(200)는 예측 움직임 벡터 선택기(210), 움직임 벡터 예측 모드 선택기(220), 제 1 부호화기(230), 제 2 부호화기(240) 및 제 3 부호화기(250)를 포함하여 구성될 수 있다.

[68] 예측 움직임 벡터 선택기(210)는 현재 블록의 현재 움직임 벡터에 대하여 예측 움직임 벡터로 사용할 수 있는 후보 움직임 벡터의 집합인 후보 움직임 벡터 집합을 선정하고, 선정된 후보 움직임 벡터 집합에서 예측 움직임 벡터를 선택한다. 후보 움직임 벡터 집합에서 예측 움직임 벡터를 선택하는 방법은

다양할 수 있는데, 예를 들어, 후보 움직임 벡터 집합에서 현재 움직임 벡터와의 차이를 최소화하는 후보 움직임 벡터를 예측 움직임 벡터로 선택할 수 있다. 이와 같이, 예측 움직임 벡터 선택기(210)에 의해 후보 움직임 벡터 집합에서 선택된 예측 움직임 벡터를 이하에서는 설명의 편의를 위해 최적 예측 움직임 벡터라고 한다. 다만, 이는 설명의 편의를 위해 구체적인 예로서 최적 예측 움직임 벡터를 정의한 것일 뿐, 본 발명의 실시예에서 최적 예측 움직임 벡터는 이미 전술한 바와 같이 미리 정해진 기준 또는 방법에 따라 얻어지는 현재 움직임 벡터의 예측 움직임 벡터를 의미한다.

[69] 예측 모드 결정기(220)는 예측 움직임 벡터 선택기(210)에서 선택된 최적 예측 움직임 벡터를 움직임 벡터 복호화 장치에서 현재 움직임 벡터를 복원하는 데 사용할 것이지 아니면 디폴트 예측 움직임 벡터를 사용할 것인지 여부를 판단하여, 판단 결과에 따라 경쟁 예측 모드 또는 비경쟁 예측 모드 중 하나를 선택하여 움직임 벡터 예측 모드로서 결정한다. 이때, 움직임 벡터 예측 모드가 비경쟁 예측 모드에 해당하는 경우, 움직임 벡터 복호화 장치에서 복호화하고자 하는 블럭이 비경쟁 예측 모드에 해당하는지 여부를 스스로 알수 있으므로, 움직임 벡터 예측 모드의 부호화 및 전송을 생략하여 압축 효율을 높힐 수 있다.

[70] 여기서, 예측 모드 결정기(220)는 예측 움직임 벡터 선택기(210)에서 선택한 최적 예측 움직임 벡터가 디폴트 예측 움직임 벡터와 동일한 경우, 비경쟁 예측 모드를 현재 블록의 움직임 벡터 예측 모드로서 결정한다. 비경쟁 예측 모드를 현재 블록의 움직임 벡터 예측 모드로서 결정하는 경우는, 복수 개의 서로 다른 후보 움직임 벡터 중에서 선택한 움직임 벡터 즉, 최적 예측 움직임 벡터가 디폴트 예측 움직임 벡터와 동일한 경우뿐만 아니라, 예측 움직임 벡터로 선택할 수 있는 후보 움직임 벡터가 존재하지 않는 경우, 후보 움직임 벡터가 존재하더라도 하나만 있는 경우, 후보 움직임 벡터가 두 개 이상 존재하지만 모두 동일한 경우가 될 수 있다. 이는 예측 움직임 벡터 선택기(210)에서 예측 움직임 벡터로 사용할 수 있는 후보 움직임 벡터가 존재하지 않거나(이 경우, 예측 움직임 벡터로 사용할 수 있는 후보 움직임 벡터가 존재하지 않으므로 사용할 수 있는 후보 움직임 벡터를 모두 (0,0)으로 정의할 수도 있다.) 하나 또는 모두 동일한 경우, 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치는 모두 동일한 최적 예측 움직임 벡터를 찾을 것이며 찾아진 최적 예측 움직임 벡터는 디폴트 예측 움직임 벡터와 동일하게 되기 때문이다.

[71] 따라서, 전술한 비경쟁 경우에는 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치는 어떠한 예측 모드를 통해 최적 예측 움직임 벡터를 선택하더라도 동일한 결과가 발생하는 소정의 판단 과정을 공유함으로써, 동일한 예측 움직임 벡터를 선택할 수 있다. 이와 같은 비경쟁 경우, 현재 움직임 벡터와 예측 움직임 벡터의 차이인 차분 벡터는 제 1 부호화기(230)에 의해 부호화된다.

[72] 제 1 부호화기(230)는 예측 모드 결정기(220)에서 움직임 벡터 예측 모드가 비경쟁 예측 모드로 결정되는 경우 동작하며, 디폴트 움직임 예측 벡터를 현재

움직임 벡터에 대한 예측 움직임 벡터로 결정하고, 현재 움직임 벡터와 예측 움직임 벡터의 차분 벡터를 계산하여 계산된 차분 벡터를 부호화한다.

[73] 제 2 부호화기(230)는 예측 모드 결정기(220)에서 움직임 벡터 예측 모드가 경쟁 예측 모드로 판단되는 경우 동작하며, 예측 모드 결정기(220)에서 결정한 움직임 벡터 예측 모드의 비트 플래그값에 따라 결정되는 예측 움직임 벡터(즉, 최적 예측 움직임 벡터(예를 들어, 움직임 벡터 예측 모드의 비트 플래그값이 '1'인 경우) 또는 중간값과 같이 사전에 정의된 디폴트 예측 움직임 벡터(예를 들어, 움직임 벡터 예측 모드의 비트 플래그값이 '0'인 경우))를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하고, 현재 움직임 벡터와 예측 움직임 벡터의 차분 벡터를 계산하여 계산된 차분 벡터를 부호화한다.

[74] 제 3 부호화기(250)는 소정의 단위(예를 들어, 슬라이스, 픽처 등) 내의 블록들 중 예측 모드 결정기(220)에 의해 움직임 벡터 예측 모드가 경쟁 예측 모드로 결정된 블록들의 움직임 벡터 예측 모드를 부호화한다. 제 3 부호화기(250)가 소정의 단위 내의 블록들 중 움직임 벡터 예측 모드가 경쟁 예측 모드로 결정된 블록들의 움직임 벡터 예측 모드를 별도로 부호화하는 것은, 움직임 벡터 복호화 장치가 비트스트림으로부터 경쟁 예측 모드에 해당하는 움직임 벡터 예측 모드를 움직임 벡터 예측 모드를 파싱할 때, 파싱 과정을 현재 움직임 벡터를 복원하거나 블록의 텍스처(Texture) 정보를 복원하는 복호화(Decoding) 과정과 독립적으로 수행하기 위한 것이다.

[75] 즉, 제 3 부호화기(250)는 현재 움직임 벡터를 부호화는 부호화 단위(예를 들어, 매크로블록 단위 또는 블록 단위)의 상위에 존재하는 소정의 부호화 단위(예를 들어, 슬라이스 단위 또는 픽처 단위, 이하에에서는 상위에 존재하는 소정의 부호화 단위를 '상위 단위'라 칭함)에 존재하는 움직임 벡터 예측 모드를 모아서 부호화한다. 이때, 상위 단위는 슬라이스 단위일 수 있지만, 반드시 이에 한정되지 않고 픽처 단위와 같은 슬라이스 단위보다 더 큰 단위가 될 수도 있다.

[76] 여기서, 제 3 부호화기(250)는 도 12에 도시한 바와 같이, 움직임 벡터 예측 모드를 부호화할 때, 상위 단위 내의 블록들 중에서 경쟁 예측 모드로 움직임 벡터 예측 모드가 결정된 블록의 움직임 벡터 예측 모드의 개수(즉, (즉, 전송 또는 저장하는 움직임 벡터 예측 모드의 개수)를 식별하는 정보(예를 들면, 비트값이 될 수 있으며, 도 12에는 전체 개수보다 하나 작은 값인 "num_mv_mode_flag_minus1"로 표시함)와 그 개수만큼의 움직임 벡터 예측 모드에 대한 정보(즉, 움직임 벡터 예측 모드를 나타내는 비트 플래그)를 식별하는 정보(예를 들면, 비트 플래그의 비트값들이 나열될 수 있으며, 도 12에는 "mv_mode_flag[i], $0 \le i \le$ num_mv_mode_flag_minus1"로 표시함)를 부호화한다. 이때, 제 3 부호화기(250)는 움직임 벡터 예측 모드의 개수를 식별하는 정보와 그 개수만큼의 움직임 벡터 예측 모드에 대한 정보를 엔트로피 부호화와 같은 부호화 기법을 이용하여 부호화하거나 다양한 비손실 압축 방식으로 부호화하여 압축할 수도 있지만, 압축하지 않고 그대로 전송할 수도

있다.

[77] 제 1 부호화기(230) 또는 제 2 부호화기(240)에 의해 부호화된 차분 벡터는 비트스트림에 삽입되며, 제 3 부호화기(250)에 의해 부호화된 움직임 예측 모드도 비트스트림에 삽입된다. 이를 위해, 도 2에는 도시하지 않았지만, 멀티플렉서(Multiplexer)에 의해 부호화된 차분 벡터와 부호화된 움직임 예측 모드가 비트스트림에 삽입될 수 있다. 다만, 제 3 부호화기(250)에 의해 부호화된 움직임 예측 모드는 비트스트림에 삽입될 때, 비트스트림 중에서 차분 벡터가 삽입되기 이전에 삽입될 수 있는데, 상위 단위에 따라 다른 헤더에 삽입될 수 있다. 예를 들어, 상위 단위가 슬라이스 단위라면 슬라이스 헤더에 삽입될 수 있으며, 상위 단위가 픽처 단위라면 픽처 헤더에 삽입될 수 있다.

[78] 이와 같이 부호화된 비트스트림은 유선, 무선 또는 저장매체 등의 전송 채널을 통해 움직임 벡터 복호화 장치로 전송되어 복호화된다.

[79] 도 3은 본 발명의 일 실시예에 따른 움직임 벡터 복호화 장치를 간략하게 나타낸 블록 구성도이다.

[80] 도 3을 참조하면, 본 발명의 일 실시예에 따른 움직임 벡터 복호화 장치(300)는 예측 모드 복호화기(310), 움직임 벡터 예측 모드 판단 및 차분 벡터 복호화기(320), 제 1 예측 움직임 벡터 결정기(330), 제 2 예측 움직임 벡터 결정기(340) 및 움직임 벡터 복원기(350)를 포함하여 구성될 수 있다.

[81] 움직임 벡터 부호화 장치(200)에 의해 부호화되어 비트스트림에 삽입된 부호화된 움직임 벡터 예측 모드와 부호화된 차분 벡터는 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300) 사이에 미리 약속된 서로 다른 소정의 단위에 따라 복호화된다. 즉, 부호화된 움직임 벡터 예측 모드는 상위 단위로 복호화되고, 부호화된 차분 벡터는 그 하위 단위 즉, 블록 단위로 복호화된다.

[82] 이를 위해, 도 3에는 도시하지 않았지만, 디멀티플렉서(Demultiplexer)에 의하여 비트스트림으로부터 부호화된 움직임 벡터 예측 모드와 부호화된 차분 벡터가 구분되며, 상위 단위로 부호화된 움직임 벡터 예측 모드는 예측 모드 복호화기(310)에 입력되고, 각 블록에 대한 부호화된 차분 벡터는 움직임 벡터 예측 모드 판단 및 차분 벡터 복호화기(320)로 입력된다. 만일, 현재 블록이 SKIP 모드로 부호화된 경우는 차분 벡터도 존재하지 않으므로, 움직임 벡터 예측 모드 판단 및 차분 벡터 복호화기(320)로 입력되는 부호화된 차분 벡터는 없다.

[83] 예측 모드 복호화기(310)는 상위 단위 내의 경쟁 예측 모드에 해당하는 블록들의 부호화된 움직임 벡터 예측 모드를 복호화하여 복원된 움직임 벡터 예측 모드를 출력하며, 출력된 움직임 벡터 예측 모드는 메모리와 같은 저장 매체에 저장되어 제 2 예측 움직임 벡터 결정기(340)에서 각 블록의 예측 움직임 벡터를 결정할 때마다 하나씩 순차적으로 읽혀져서, 순차적으로 읽혀지는 움직임 벡터 예측 모드에 의해 결정되는 움직임 벡터를 각 블록의 예측 움직임 벡터로 결정하도록 한다. 이때, 복원된 움직임 벡터 예측 모드는 움직임 벡터 예측 모드의 개수를 나타내는 비트와 그 개수만큼의 움직임 벡터 예측 모드를

나타내는 비트를 포함하는데, 예측 모드 복호화기(310)는 우선 움직임 벡터 예측 모드의 개수를 나타내는 비트를 읽어서 움직임 벡터 예측 모드의 개수를 파악하고, 그 개수만큼의 비트를 차례로 읽어 그 개수만큼의 움직임 벡터 예측 모드를 나타내는 비트를 읽어들임으로써 움직임 벡터 예측 모드를 파싱하며, 파싱한 움직임 벡터 예측 모드를 나타내는 비트만을 출력하여 저장되도록 한다.

[84] 또한, 예측 모드 복호화기(310)는 비트스트림 내에 구분되어 있는 상위 단위에 해당하는 움직임 벡터 예측 모드들이 엔트로피 부호화 등의 추가 압축 과정 없이 도 12에 도시된 것처럼 표현된 경우에는 파싱 과정을 통해 파싱하여 움직임 벡터 예측 모드를 복원하며, 만일 엔트로피 부호화 또는 여타의 무손실 압축 부호화가 수행된 경우에는 각 부호화 방법에 해당하는 역 과정을 통하여 복호화하여 움직임 벡터 예측 모드를 복원한다. 어떤 경우에 해당하던지, 이러한 움직임 벡터 예측 모드의 복호화 과정은 이전 또는 이후의 복호화 과정(예를 들면, 부호화된 차분 벡터의 복호화 과정 또는 부호화된 텍스쳐 정보의 복호화 과정 등)과 독립적으로 수행된다.

[85] 차분 벡터 복호화기(320)는 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)가 서로 공유하고 있는 소정의 판단 과정을 통해 현재 복호화하고자 하는 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하고, 비경쟁 예측 모드가 아니라면 경쟁 예측 모드로 판단한다. 여기서, 차분 벡터 복호화기(320)가 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하는 방법은 도 2를 통해 전술한 예측 모드 결정기(220)에서 판단하는 방법과 동일 또는 유사하므로 상세한 설명은 생략한다. 또한, 차분 벡터 복호화기(320)는 부호화된 차분 벡터를 복호화하여 현재 블록의 차분 벡터를 복원한다. 이때, 차분 벡터 복호화기(320)는 SKIP 모드에 따른 부호화와 같이 부호화된 차분 벡터가 없는 경우에는 (0,0)을 복호화된 차분 벡터로 출력한다.

[86] 또한, 차분 벡터 복호화기(320)는 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인 경우, 제 1 예측 움직임 벡터 결정기(330)로 현재 블록이 비경쟁 예측 모드임을 알려서 디폴트 예측 움직임 벡터를 현재 블록의 예측 움직임 벡터로 결정하도록 하고, 현재 블록의 움직임 벡터 예측 모드가 경쟁 예측 모드인 경우, 제 2 예측 움직임 벡터 결정기(340)로 현재 블록이 경쟁 예측 모드임을 알려서 복원된 움직임 벡터 예측 모드에 따라 결정되는 움직임 벡터를 현재 블록의 예측 움직임 벡터로 결정하도록 한다.

[87] 제 1 예측 움직임 벡터 결정기(330)는 현재 블록이 비경쟁 예측 모드인 경우 동작하며, 움직임 벡터 부호화 장치(200)와 서로 약속된 방식으로 결정되는 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정한다.

[88] 제 2 예측 움직임 벡터 결정기(340)는 현재 블록이 경쟁 예측 모드인 경우에 동작하며, 예측 모드 복호화기(310)에서 복원한 움직임 벡터 예측 모드에 따라

선택 가능한 후보 움직임 벡터 집합 중에 선정된 움직임 벡터 즉, 최적 예측 움직임 벡터 또는 움직임 벡터 부호화 장치(200)와 서로 약속된 방식으로 결정되는 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정한다.

[89] 여기서, 제 2 예측 움직임 벡터 결정기(340)는 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드가 아닌 경우 즉, 경쟁 예측 모드인 경우, 최적 예측 움직임 벡터를 사용하여야 할지(예를 들어, 움직임 벡터 예측 모드의 비트 플래그의 값이 '1'인지) 또는 디폴트 예측 움직임 벡터를 사용해야 할지(예를 들어, 움직임 벡터 예측 모드의 비트 플래그의 값이 '0'인지) 여부를 결정해야 한다. 이를 위해, 제 2 예측 움직임 벡터 결정기(340)는 각 블록의 예측 움직임 벡터를 결정할 때마다 예측 모드 복호화기(310)에 의해 복원되어 저장된 움직임 벡터 예측 모드의 비트를 하나씩 순차적으로 읽어서, 읽혀지는 움직임 벡터 예측 모드에 의해 결정되는 움직임 벡터를 각 블록의 예측 움직임 벡터로 결정한다. 이때, 예측 모드 복호화기(310)에 의해 복원되어 저장되는 움직임 벡터 예측 모드의 비트는 움직임 벡터 예측 모드를 나타내는 비트만으로 구성되어 있으므로, 제 2 예측 움직임 벡터 결정기(340)는 각 블록의 예측 움직임 벡터를 결정하기 위한 움직임 벡터 예측 모드를 얻기 위해 움직임 벡터 예측 모드를 나타내는 비트를 순차적으로 읽고, 읽어들인 비트의 값을 확인하여 최적 예측 움직임 벡터를 사용할지 또는 디폴트 예측 움직임 벡터를 사용할지 여부를 결정한다(예를 들어, 읽어들인 비트의 값이 '1'인 경우 최적 예측 움직임 벡터를 사용하는 것으로 결정하고, 읽어들인 비트의 값이 '0'인 경우 디폴트 예측 움직임 벡터를 사용하는 것으로 결정한다.).

[90] 움직임 벡터 복원기(350)는 제 1 예측 움직임 벡터 결정기(330) 또는 제 2 예측 움직임 벡터 결정기(340)에서 결정된 예측 움직임 벡터(최적 예측 움직임 벡터 또는 디폴트 예측 움직임 벡터)와 차분 벡터 복호화기(320)가 복원한 차분 벡터를 더하여 현재 블록의 현재 움직임 벡터를 복원한다.

[91] 이하에서는 전술한 본 발명의 일 실시예에 따른 움직임 벡터 부호화 장치(200)에 의한 움직임 벡터 부호화 방법을 도 4 내지 도 12를 참조하여 상세하게 설명한다.

[92] 도 4는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 방법을 설명하기 위한 순서도이다.

[93] 움직임 벡터 부호화 장치(200)는 복수 개의 움직임 벡터 예측 모드 중 선택된 움직임 벡터 예측 모드에 따라 현재 움직임 벡터에 대한 예측 움직임 벡터를 선택하고(S410), 현재 움직임 벡터와 선택된 예측 움직임 벡터 간의 차분 벡터를 부호화하며(S420), 선택된 움직임 벡터 예측 모드에 따라 움직임 벡터 예측 모드를 부호화하며(S430), 부호화된 차분 벡터 및 부호화된 움직임 벡터 예측 모드를 포함하는 비트스트림을 생성하여 출력한다(S440).

[94] 단계 S430에서, 움직임 벡터 부호화 장치(200)는 선택된 움직임 벡터 예측

모드에 따라 움직임 벡터 예측 모드를 부호화하는데, 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)가 현재 블록의 움직임 벡터 예측 모드를 결정하는 판단 과정을 공유한다는 가정 한다면, 움직임 벡터 예측 모드가 비경쟁 예측 모드인 경우에는 움직임 벡터 예측 모드를 부호화하지 않고 생략하며, 경쟁 예측 모드인 경우에만 움직임 벡터 예측 모드를 부호화할 수 있다.

[95] 예를 들어, 현재 움직임 벡터에 대한 예측 움직임 벡터로 사용할 수 있는 후보 움직임 벡터가 존재하지 않거나(이 경우, 예측 움직임 벡터로 사용할 수 있는 후보 움직임 벡터가 존재하지 않으므로, 사용할 수 있는 후보 움직임 벡터를 모두 (0,0)으로 정의할 수도 있다.) 하나만 있거나, 사용할 수 있는 복수 개의 후보 움직임 벡터가 존재하더라도 모두 동일하거나, 사용할 수 있는 서로 다른 복수 개의 후보 움직임 벡터가 존재하여 결정한 최적 예측 움직임 벡터가 디폴트 예측 움직임 벡터와 동일하다면, 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)는 어떠한 움직임 벡터 예측 모드를 통해 최적 예측 움직임 벡터를 선택하더라도 동일한 결과가 발생함을 알 수 있다. 따라서, 이러한 경우, 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)는 어떠한 예측 모드를 통해 최적 예측 움직임 벡터를 선택하더라도 동일한 결과가 발생한다는 소정의 판단 과정을 공유함으로써 동일한 최적 예측 움직임 벡터를 선택할 수 있다.

[96] 이와 같이, 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)가 소정의 판단 과정을 공유함으로써, 움직임 벡터 예측 모드의 저장(또는 전송) 여부를 판단할 수 있는 경우, 비경쟁 예측 모드에 의해 결정되는 예측 움직임 벡터를 이용하여 생성되고 부호화된 차분 벡터는 동일할 것임을 알 수 있으므로, 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)는 서로 약속된 방식으로 결정되는 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로서 선택한다.

[97] 단계 S410에서, 움직임 벡터 부호화 장치(200)는 선택된 움직임 벡터 예측 모드가 비경쟁 예측 모드인 경우에는, 기 설정된 방식으로 결정되는 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 선택한다.

[98] 또한, 단계 S410에서, 움직임 벡터 부호화 장치(200)는 선택된 움직임 벡터 예측 모드가 비경쟁 예측 모드가 아닌 경우에는, 예측 움직임 벡터로서 최적 움직임 예측 벡터를 사용할지 또는 디폴트 예측 움직임 벡터를 사용할지 여부를 결정한다. 이를 위해, 움직임 벡터 부호화 장치(200)는 소정의 방법을 이용하여 복수 개의 후보 움직임 벡터 중에서 하나의 후보 움직임 벡터를 현재 움직임 벡터의 예측 움직임 벡터(즉, 최적 예측 움직임 벡터)로 선택한다. 그리고 이와 동일한 과정을 통하여 움직임 벡터 복호화 장치(300)가 움직임 벡터 부호화 장치(200)에서 선택한 최적 예측 움직임 벡터와 동일한 움직임 벡터를 최적 예측 움직임 벡터로서 선택할 수 있는지 여부 즉, 움직임 벡터 부호화 장치(200)에서 선택한 최적 예측 움직임 벡터가 움직임 벡터 복호화 장치(300)에서 예측할 수

있는지 여부를 판단한다.

[99] 만일, 움직임 벡터 복호화 장치(300)가 선택할 최적 예측 움직임 벡터가 움직임 벡터 부호화 장치(200)가 선택한 최적 예측 움직임 벡터와 동일하면, 움직임 벡터 예측 모드를 나타내는 비트를 예측 가능 모드를 나타내는 값(예를 들면, '1')으로 결정하고, 현재 움직임 벡터에 대한 예측 움직임 벡터로서 최적 예측 움직임 벡터를 선택한다. 하지만, 움직임 벡터 복호화 장치(200)가 선택할 최적 예측 움직임 벡터가 움직임 벡터 부호화 장치(200)가 선택한 최적 예측 움직임 벡터와 동일하지 않다면 움직임 벡터 예측 모드를 나타내는 비트를 예측 불가능 모드를 나타내는 값(예를 들면, '0')으로 결정하고, 현재 움직임 벡터에 대한 예측 움직임 벡터로서 디폴트 예측 움직임 벡터를 선택한다.

[100] 도 4를 통해 전술한 본 발명으 일 실시예에 따른 움직임 벡터 부호화 방법은 도 5에 나타낸 바와 같이 구체화되어 구현될 수 있다.

[101] 도 5는 본 발명의 일 실시예에 따른 움직임 벡터 부호화 방법의 구체화된 구현예를 나타낸 순서도이다.

[102] 도 5를 참조하면, 움직임 벡터 부호화 장치(200)는 현재 블록의 현재 움직임 벡터에 대한 후보 움직임 벡터 집합을 선정하고(S500), 선정된 후보 움직임 벡터 집합에서 최적 예측 움직임 벡터를 선택하며(S502), 현재 블록의 움직임 벡터 예측 모드를 선택하여 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하며(S504), 단계 S504의 판단 결과, 움직임 벡터 예측 모드가 비경쟁 예측 모드인 경우에는 움직임 벡터 복호화 장치(300)에서 최적 예측 움직임 벡터를 예측할 수 있는지 여부를 판단하며(S506), 단계 S506의 판단 결과, 예측할 수 있는 경우에는 최적 예측 움직임 벡터를 이용하여 현재 움직임 벡터를 부호화하며(S508), 단계 S506의 판단 결과, 예측할 수 없는 경우 또는 단계 S504의 판단 결과, 움직임 벡터 예측 모드가 비경쟁 예측 모드가 아닌 경우 즉, 경쟁 예측 모드인 경우에는 디폴트 예측 움직임 벡터를 이용하여 현재 움직임 벡터를 부호화한다(S510).

[103] 또한, 움직임 벡터 부호화 장치(200)는 블록의 상위 단위 내의 모든 블록 내에서 단계 S500 내지 단계 S510의 절차가 수행되지 않은 남은 블록이 있는지 여부를 판단하여(S512), 남은 블록이 있는 경우에는 단계 S500으로 진행하여 남은 블록에 대해 단계 S500 내지 단계 S510의 절차를 수행하며, 남은 블록이 없는 경우에는 상위 단위 내의 블록에 대한 움직임 벡터 예측 모드를 부호화하는 제 3 부호화를 수행한다(S514).

[104] 또한, 움직임 벡터 부호화 장치(200)는 단계 S508 또는 단계 S510에서 부호화된 현재 움직임 벡터(즉, 차분 벡터)와 단계 S514에서 부호화된 움직임 벡터 예측 모드를 비트스트림에 삽입하여 출력한다.

[105] 단계 S504에서, 움직임 벡터 부호화 장치(200)는 움직임 벡터 복호화 장치(300)와 사전에 공유하고 있는 소정의 판단 기준에 따라 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부(그에 따라 부호화를 생략할 수 있는지

여부)를 판단한다. 여기서, 움직임 벡터 예측 모드의 부호화를 생략할 수 있는지 여부 즉, 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부는 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)가 소정의 판단 과정을 공유함으로써, 상호 간에 움직임 벡터 예측 모드에 대한 정보를 교환하지 않아도 어떤 움직임 벡터 예측 모드를 사용해야 하는지를 알 수 있는 경우에 해당하는지 여부를 판단함으로써 판단할 수 있다. 즉, 후보 예측 움직임 벡터의 개수가 0 또는 1이거나, 하나 이상 있더라도 모두 동일한 경우 또는 후보 예측 움직임 벡터의 개수가 두 개 이상 복수 개이며 서로 동일하지 않더라도 그를 통해 선택된 최적 예측 움직임 벡터가 디폴트 예측 움직임 벡터와 동일한 경우가 비경쟁 예측 모드에 해당한다. 따라서, 움직임 벡터 예측 모드의 부호화를 생략할 수 있는지 여부 즉, 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부는 디폴트 예측 움직임 벡터에 의해 생성되고 부호화된 차분 벡터와 최적 예측 움직임 벡터에 의해 생성되고 부호화된 차분 벡터가 동일함을 알 수 있다는 조건에 의해 판단된다. 이에 대해서는 도 2를 통해 전술하였으므로 상세한 설명은 생략한다.

[106] 단계 S504에서 움직임 벡터 예측 모드가 비경쟁 예측 모드라고 판단된 경우 또는 단계 S506에서, 움직임 벡터 복호화 장치(300)가 단계 S502에서 선택된 최적 예측 움직임 벡터를 예측할 수 없는 경우, 단계 S514에서 움직임 벡터 부호화 장치(200)는 움직임 벡터 예측 모드를 부호화하지 않는다. 하지만, 단계 S506에서 움직임 벡터 예측 모드가 비경쟁 예측 모드라고 판단되고 단계 506에서, 움직임 벡터 복호화 장치(300)가 단계 S502에서 선택된 최적 예측 움직임 벡터를 예측할 수 있는 경우, 단계 S514에서 움직임 벡터 부호화 장치(200)는 블록의 움직임 벡터 예측 모드를 부호화한다.

[107] 또한, 움직임 벡터 부호화 장치(200)는 단계 S514에서, 움직임 벡터 예측 모드를 부호화할 때, 블록 단위가 아닌 상위 단위의 블록 내에서 경쟁 예측 모드에 해당하는 블록들의 움직임 벡터 예측 모드들을 모아서 부호화한다. 이를 위해, 움직임 벡터 부호화 장치(200)는 상위 단위 내의 블록들 중 경쟁 예측 모드에 해당하는 블록의 움직임 벡터 예측 모드를 움직임 벡터 예측 모드의 개수(예를 들어, 그를 나타내는 비트) 및 그 개수만큼의 움직임 벡터 예측 모드(예를 들어, 그를 나타내는 비트)를 부호화하는데, 압축 없이 부호화하거나 무손실 압축 부호화를 이용하여 부호화할 수 있다. 이때, 움직임 벡터 부호화 장치(200)는 상위 단위 내의 블록들 중 경쟁 예측 모드에 해당하는 블록들의 움직임 벡터 예측 모드들을 모아서 부호화하므로, 단계 S512를 통해 상위 단위 내에 남은 블록이 존재하는지 여부를 판단하여 상위 단위 내의 모든 블록들에 대해 단계 S500 내지 단계 S512를 수행하여 상이 단위 내의 모든 블록 내에서 경쟁 예측 모드에 해당하는 블록의 움직임 벡터 예측 모드를 구한 후 단계 S514에서 부호화한다.

[108] 최종적으로 부호화된 차분 벡터와 S512 단계에 의해 부호화된 움직임 벡터

예측 모드는 각각 비트스트림의 신택스(Syntax)로서 삽입된다. 비트스트림은 유선, 무선 또는 저장매체 등의 전송 채널을 통해 움직임 벡터 복호화 장치(300)로 전송된다.

[109] 이하에서는 도 5를 통해 전술한 각 단계 중 단계 S500, 단계 S502, 단계 S506, 단계 S508, 단계 S510에 대해 도 6, 도 7, 도 8, 도 9 및 도 12를 참조하여 각각 상세하게 설명한다.

[110] 도 6은 본 발명의 일 실시예에 따라 후보 움직임 벡터 집합을 선정하는 과정을 설명하기 위한 순서도이다.

[111] 도 6를 참조하면, 본 발명의 일 실시예에 따라 후보 움직임 벡터 집합을 선정하는 단계(S500)는, 현재 블록에 대한 하나 이상의 주변 블록을 검색하는 단계(S600), 검색된 하나 이상의 주변 블록의 움직임 벡터를 각각 산출하는 단계(S602) 및 각각 산출된 후보 움직임 벡터를 취합하여 후보 움직임 벡터 집합으로 선정하는 단계(S604) 등을 포함한다.

[112] 도 1에 나타낸 예시를 참조하면, 전술한 후보 움직임 벡터 집합 선정 단계(S500)에서 선정된 후보 움직임 벡터 집합은, 현재 블록인 블록 D의 좌측, 상단, 우측 상단에 있는 주변 블록인 블록 A, 블록 B 및 블록 C가 가지는 움직임 벡터들로 이루어진 $\{MV^A, MV^B, MV^C\}$가 된다. 하지만, 더욱 일반적으로는 후보 움직임 벡터 집합은 구현 방법이나 필요에 따라 보다 다양한 움직임 벡터들을 후보 움직임 벡터들의 집합(후보 움직임 벡터 집합)으로 선정할 수 있다.

[113] 예를 들어, 시간축 상으로 이전에 존재하는 픽처의 동일 위치 블록의 움직임 벡터나 공간축 상으로 좌측 상단에 위치한 블록의 움직임 벡터도 후보 움직임 벡터로 사용할 수 있다. 또한 이들 움직임 벡터들을 이용하여 선정된 또 다른 움직임 벡터(예를 들어, 몇 개의 움직임 벡터의 평균값이나 중간값 등)도 포함될 수 있다.

[114] 따라서, 단계 S600는, 시간축 상에서 이전에 존재하는 픽처(Picture) 및 이후에 존재하는 픽처 중 하나 이상의 픽처에서 현재 블록과 동일한 위치에 있는 하나 이상의 제 1 블록; 및 공간축 상으로 현재 블록에 인접한 위치에 있는 하나 이상의 제 2 블록 중 하나 이상을 하나 이상의 주변 블록으로 검색할 수 있다.

[115] 또한 단계 S602는, 하나 이상의 제 1 블록에서의 각각의 움직임 벡터; 하나 이상의 제 2 블록에서의 각각의 움직임 벡터; 및 제 1 블록에서의 각각의 움직임 벡터와 제 2 블록에서의 각각의 움직임 벡터 중 하나 이상에 대한 평균값 또는 중간값; 중 하나를 검색된 하나 이상의 주변 블록의 움직임 벡터로 산출할 수도 있다.

[116] 전술한 후보 움직임 벡터 집합은 그 정의를 부호화 장치(200)와 복호화 장치(300)가 사전에 알고 있다는 전제 하에 다양한 방법으로 정의될 수 있다. 또한 후보 움직임 벡터 집합을 구성하는 원소인 후보 움직임 벡터 중 일부 또는 전부가 동일한 값을 갖는 경우, 서로 다른 값을 갖는 후보 움직임 벡터들만의 집합으로 구성할 수도 있다.

[117] 도 7은 본 발명의 일 실시예에 따라 예측 움직임 벡터를 선택하는 과정을 설명하기 위한 순서도이다.

[118] 도 7을 참조하면, 예측 움직임 벡터 선택 단계(S502)는, 부호화 장치(200) 및 복호화 장치(300) 간에 기 정의된 선택 함수를 이용하여 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각에 대한 선택 함수값을 계산하는 단계(S700); 및 계산된 선택 함수값들에 근거하여 하나의 후보 움직임 벡터를 최적 예측 움직임 벡터로 선택하는 단계(S702) 등을 포함한다.

[119] 일 예로서, 전술한 "선택 함수값"은, 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각에 대한 현재 움직임 벡터와의 차분 벡터를 부호화하는데 소요되는 비트량; 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각에 대한 현재 움직임 벡터와의 차분 벡터의 크기; 및 움직임 벡터 예측 모드를 부호화하는데 소요되는 비트량; 중 하나 이상 또는 하나 이상을 조합한 것을 포함할 수도 있다.

[120] 만약 차분 벡터의 비트량을 선택 함수값으로 이용하는 경우, 전술한 단계 S702에서는, 일 예로, 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각에 대한 차분 벡터를 부호화하는데 소요되는 비트량을 계산하고, 계산된 비트량 중 최소값이 되는 후보 움직임 벡터를 최적 예측 움직임 벡터로 선택할 수 있다.

[121] 또한, 예측 움직임 벡터 선택 단계(S502)는, 전술한 비트량을 선택 함수값에 근거하여 최적 예측 움직임 벡터를 선택하는 방식 이외에도, 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 중의 하나의 움직임 벡터를 선택했을 경우 발생하는 부호화에 소요되는 비트량과 이때 발생할 복원화질을 같이 고려하는 율-왜곡 최적화(Rate-Distortion Optimization) 방식을 이용하여서도 최적 예측 움직임 벡터를 선택할 수도 있다.

[122] 전술한 예측 움직임 벡터 선택 단계(S502)에서는, 수학식 4에서의 라그랑제 코스트(Lagrangian Cost) 함수를 최적 예측 움직임 벡터를 선택하기 위한 선택 함수로 이용할 수 있다.

[123] 수학식 4

$$J(n,k)=D(n,k)+\lambda\times[R_H(n,k)+R_M(n,k)]$$

[124] 여기서, J는 라그랑제 코스트, D는 원 영상과 복원 영상 간의 오차, λ는 라그랑제 승수이다. R_H은 움직임 벡터 예측 모드를 부호화하는데 소요되는 비트량이고, R_M은 현재 움직임 벡터의 차분 벡터를 부호화하는데 소요되는 비트량이다. 수학식 4에서의 J, D, R_H, R_M은 모두 현재 블록이 위치하는 픽처 번호를 표시하는 n과 블록 번호를 표시하는 k에 따라 정의된다. 따라서, 픽처 또는 블록 단위에서 선택적으로 적용될 수 있는 것이다.

[125] 또한, 최적 예측 움직임 벡터 선택하는 과정에서 원 영상과 복원 영상 간의

오차인 D가 변하지 않는 경우 또는 계산상의 편의를 위해, 라그랑제 코스트 J를 산출하는 수학식 4에서 D와 λ를 제거하여 식을 간략화할 수도 있다.

[126] 움직임 벡터 부호화 방법에 의해 발생하는 라그랑제 코스트를 산정하는 과정에서 수학식 4의 R_H는 움직임 벡터 예측 모드 부호화에 소요되는 비트량이고, R_M은 부호화된 현재 움직임 벡터에 대한 차분 벡터에 소요되는 비트량으로서, 그 계산 방법은 움직임 벡터 예측 모드에 따라 달라진다. 즉, 움직임 벡터 예측 모드가 복호화 장치(300)에서 최적 예측 움직임 벡터를 예측할 수 없음을 지시하는 경우, R_M은 중간값(Median) 계산과 같이 사전에 정의된 디폴트 방법에 의해 생성된 예측 움직임 벡터 즉, 디폴트 예측 움직임 벡터와 현재 움직임 벡터 간의 차분 벡터를 부호화하는데 소요되는 비트량이다. 또한, 움직임 벡터 예측 모드가, 복호화 장치(300)에서 최적 예측 움직임 벡터를 예측할 수 있음을 지시하는 경우, R_M은 선택된 최적 예측 움직임 벡터와 현재 움직임 벡터 간의 차분 벡터을 부호화하는데 소요되는 비트량이 되는 것이다.

[127] 또한, 전술한 예측 움직임 벡터 선택 단계(S502)에서는 수학식 4에서의 라그랑제 코스트(Lagrangian Cost) 함수를 이용하여 최적 예측 움직임 벡터를 선택하는 것 이외에도, 좀 더 일반화된 선택 함수의 식이 포함된 수학식 5를 이용하여 최적 예측 움직임 벡터를 선택할 수도 있다. 단, 수학식 5는 부호화할 현재 블록의 현재 움직임 벡터를 도 1에서의 블록 D의 움직임 벡터인 MV^D로 가정하여 표현되었다.

[128] 수학식 5

$$PMV_{enc} = \arg\min_{PMVC \in CS} h(PMVC|MV^D)$$

[129] 수학식 5에서, PMV_{enc}는 선택된 최적 예측 움직임 벡터이고, PMVC는 현재 움직임 벡터 MV^D의 예측 움직임 벡터로 선택 가능한 후보 움직임 벡터들의 집합인 후보 움직임 벡터 집합(CS)에 속하는 하나의 원소(움직임 벡터)를 의미한다. "h()"는 현재 움직임 벡터 MV^D를 그 예측 움직임 벡터와 차분하여 부호화하는데 최적인 예측 움직임 벡터를 선택하기 위한 선택 함수이다.

[130] 전술한 선택 함수 h()의 일례로 현재 움직임 벡터를 차분하여 부호화하는데 소요되는 비트량을 사용하거나, 현재 움직임 벡터를 차분하여 부호화하는데 소요되는 비트량과 움직임 벡터 예측 모드를 부호화하는데 소요되는 비트량의 합을 사용할 수 있다. 또한, 계산을 간단히 하기 위해 실제 소요되는 비트량 대신 현재 움직임 벡터와 예측 움직임 벡터와의 차이인 차분 벡터의 크기를 사용할 수도 있다. 더욱 일반적으로는 선택 함수 h()의 정의를 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)가 사전에 미리 알고 있다는 전제 하에 다양한 방법으로 정의하여 사용할 수 있다. 이러한 선택 함수 h()가 주어지면, 예측 움직임 벡터의 후보인 후보 움직임 벡터를 포함하는 후보 움직임 벡터

집합(CS)으로부터 선택 함수 h()를 최적화하는 하나의 후보 움직임 벡터(PMVC)를 최적 예측 움직임 벡터(PMV_{enc})로 선택할 수 있다.

[131] 도 8은 본 발명의 일 실시예에 따라 최적 예측 움직임 벡터의 예측 가능 여부를 판단하는 과정을 설명하기 위한 순서도이다.

[132] 도 8을 참조하면, 최적 예측 움직임 벡터의 예측 가능 여부를 판단하는 단계(S506)는 부호화 장치(200) 및 복호화 장치(300) 간에 기 정의된 판단 함수를 이용하여 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각에 대한 판단 함수값을 계산하는 단계(S800); 계산된 판단 함수값에 근거하여 하나 이상의 후보 움직임 벡터 중 하나의 후보 움직임 벡터를 현재 움직임 벡터에 대한 추정 예측 움직임 벡터로 선택하는 단계(S802); 선택된 추정 예측 움직임 벡터 및 선택된 최적 예측 움직임 벡터를 비교하는 단계(S804); 및 비교 결과에 따라 선택된 최적 예측 움직임 벡터가 복호화 장치(300)에서 예측 가능한지를 판단하는 단계(S806) 등을 포함한다.

[133] 전술한 예측 움직임 벡터 예측가능 여부를 판단하는 단계(S506)를 수학식 5에서의 표현을 이용하여 다시 설명하면, 앞선 예측 움직임 벡터 선택 단계(S502)에서 선택된 최적 예측 움직임 벡터(PMV_{enc})를 사용하여 계산된 차분 벡터와, 예측 움직임 벡터의 후보가 될 수 있는 유한한 개수의 후보 움직임 벡터와, 움직임 보상에 사용할 참조 픽처와, 이미 복원된 주변 블록의 정보와, 화소값에 대한 움직임 보상에 따른 잔차 신호를 이용하여, 앞선 단계(S502)에서 선택된 최적 예측 움직임 벡터가 움직임 벡터 복호화 장치(300)에서 예측 가능한지 여부를 판단한다.

[134] 이를 위해, 움직임 벡터 복호화 장치(300)는 움직임 벡터 부호화 장치(200)가 계산하여 전송할 현재 움직임 벡터(MV^D)에 대한 차분 벡터인 DMV^D(=MV^D-PMV_{enc})와 이미 복원된 주변 블록의 정보와 움직임 보상에 사용할 참조 픽처를 이용하여, 수학식 6과 같은 소정의 결정 방법을 이용하여 추정 예측 움직임 벡터인 PMV_{dec}를 찾는다.

[135] 수학식 6

$$PMV_{dec} = \arg\min_{PMVC \in CS} g(PMVC|DMV^D) = \arg\min_{PMVC \in CS} g(PMVC|MV^D - PMV_{enc})$$

[136] 수학식 6에서, 판단 함수인 g()는, 움직임 벡터 부호화 장치(200)가 계산하여 전송할 차분 벡터 및 이미 복원된 주변 블록의 정보를 이용하여, 움직임 벡터 복호화 장치(300)가 최적 예측 움직임 벡터(PMV_{enc})를 예측할 수 있는지를 움직임 벡터 부호화 장치(200)에서 판단하기 위한 함수이다. 또한, 이 판단 함수 g()는 움직임 벡터 복호화 장치(300)에서 예측 움직임 벡터를 예측할 때도 사용된다.

[137] 판단 함수 g()는 그 정의를 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)가 사전에 알고있다는 전제 하에 다양한 방법으로 정의될 수 있으며, 이러한 판단 함수 g()에 대한 구체적인 실시예를 후술한다.

[138] 수학식 6에 의해, 움직임 벡터 부호화 장치(200)는 움직임 벡터 복호화 장치(300)가 찾아낼 추정 예측 움직임 벡터인 PMV_{dec}를 미리 계산한 후, 움직임 벡터 복호화 장치(300)가 제공된 차분 벡터인 $DMV^D(=MV^D-PMV_{enc})$ 및 이미 복원된 주변 블록의 정보를 이용하여, 최적 예측 움직임 벡터(PMV_{enc})를 예측하여 올바르게 복원된 영상 데이터를 얻어낼 수 있는지 여부를 검사한다. 즉, 움직임 벡터 부호화 장치(200)는 나중에 움직임 벡터 복호화 장치(300)가 수행할 최적 예측 움직임 벡터를 찾아내는 과정을 미리 수행하여 그 결과를 부호화 시에 이용하는 것이다.

[139] 전술한 단계 S804 및 단계 S806에 대한 몇가지 구현 예를 아래에서 설명한다. 먼저, 움직임 벡터 복호화 장치(300)에 의해 최적 예측 움직임 벡터(PMV_{enc})가 예측 가능한지를 판단하기 위한 방법의 일 예는 다음과 같다.

[140] 단계 S506에서 수학식 6을 통해 계산된 추정 예측 움직임 벡터(PMV_{dec})가 단계 S502에서 선택된 최적 예측 움직임 벡터(PMV_{enc})가 같을 경우, 움직임 벡터 복호화 장치(300)는 움직임 벡터 부호화 장치(200)가 제공한 차분 벡터(DMV^D)에 자신이 추정한 예측 움직임 벡터인 추정 예측 움직임 벡터(PMV_{dec})를 더하여 정확한 현재 움직임 벡터 MV^D를 복원함으로써 올바르게 복원된 영상 데이터를 얻어낼 수 있다. 따라서, 움직임 벡터 부호화 장치(200)는 자신이 직접 선택한 최적 예측 움직임 벡터(PMV_{enc})와 움직임 벡터 복호화 장치(300)가 찾아낼 것으로 추정하는 추정 예측 움직임 벡터(PMV_{dec})가 같은 경우에는, 움직임 벡터 복호화 장치(300)에서의 최적 예측 움직임 벡터(PMV_{enc})에 대한 예측이 가능하다고 판정하고, 그 이외의 경우는 경우에는 예측이 불가능하다고 판정한다.

[141] 또는, 구현 예에 따라, 움직임 벡터 부호화 장치(200)는 자신이 직접 선택한 최적 예측 움직임 벡터(PMV_{enc})와 움직임 벡터 복호화 장치(300)가 찾아낼 것으로 추정하는 추정 예측 움직임 벡터(PMV_{dec}) 간의 차이가 소정의 경계값보다 작은 경우에는, 움직임 벡터 복호화 장치(300)에서의 최적 예측 움직임 벡터(PMV_{enc})에 대한 예측이 가능하다고 판정하고, 그 이외의 경우에는 예측이 불가능하다고 판정할 수도 있다.

[142] 움직임 벡터 복호화 장치(300)에 의해 최적 예측 움직임 벡터(PMV_{enc})가 예측 가능한지를 판단하기 위한 방법의 다른 예는 다음과 같다. 영상의 압축율이 높을 경우, 영상의 화소값의 변화가 크지 않을 경우, 또는 영상의 움직임 벡터의 변화가 그리 크지 않을 경우 등에는 비록 최적 예측 움직임 벡터(PMV_{enc})와 추정 예측 움직임 벡터(PMV_{dec})가 같지 않더라도, 추정 예측 움직임 벡터(PMV_{dec})를 사용하여 복원한 움직임 벡터(즉, $MV'^D = DMV^D + PMV_{dec}$)를 사용하여 움직임 보상된 영상 데이터와, 최적 예측 움직임 벡터(PMV_{enc})를 사용하여 복원한

움직임 벡터(즉, $MV^D = DMV^D + PMV_{enc}$)를 사용하여 움직임 보상된 영상 데이터가 같은 경우가 발생할 수 있는데, 이러한 경우, 최적 예측 움직임 벡터(PMV_{enc})와 추정 예측 움직임 벡터(PMV_{dec})가 같지 않더라도, 움직임 벡터 복호화 장치(300)가 추정 예측 움직임 벡터(PMV_{dec})를 이용하여 최적 예측 움직임 벡터(PMV_{enc})를 예측할 수 있다고 판정하고, 그 이외의 경우에는 예측할 수 없다고 판정한다.

[143] 움직임 벡터 복호화 장치(300)에 의해 최적 예측 움직임 벡터(PMV_{enc})가 예측 가능한지를 판단하기 위한 방법의 또 다른 예는 다음과 같다. 더욱 더 압축율을 높이기 위하여 추정 예측 움직임 벡터(PMV_{dec})를 사용하여 얻어진 복원된 영상 데이터와 최적 예측 움직임 벡터(PMV_{enc})를 사용하여 얻어질 복원된 영상 데이터가 소정의 경계값 이하로 차이가 날 경우에도(예를 들어, 두 복원된 영상데이터 사이의 SAD(Sum of Absolute Difference)가 소정의 경계값 이하로 차이가 날 경우), 움직임 벡터 복호화 장치(300)가 추정 예측 움직임 벡터(PMV_{dec})를 이용하여 최적 예측 움직임 벡터(PMV_{enc})를 예측가능하다고 판정하고, 그 이외의 경우에는 움직임 벡터 복호화 장치(300)가 추정 예측 움직임 벡터(PMV_{dec})를 이용하여 최적 예측 움직임 벡터(PMV_{enc})를 예측할 수 없다고 판정한다.

[144] 이상에서 전술한 예측가능 여부 판단의 몇 가지의 구현 예를 다시 설명하면, 전술한 단계 S804 및 단계 S806에서는, 선택된 추정 예측 움직임 벡터와 선택된 최적 예측 움직임 벡터가 동일하거나 소정의 경계값 이하로 차이가 날 경우, 선택된 최적 예측 움직임 벡터가 움직임 벡터 복호화 장치(300)에서 예측가능하다고 판단할 수 있다.

[145] 또한, 전술한 단계 S804 및 단계 S806에서는, 선택된 추정 예측 움직임 벡터를 이용하여 복원한 영상 데이터가 선택된 최적 예측 움직임 벡터를 이용하여 복원한 영상 데이터와 동일하거나 소정의 경계값 이하로 차이가 날 경우, 선택된 최적 예측 움직임 벡터가 복호화 장치(300)에서 예측 가능하다고 판단할 수도 있다.

[146] 도 8를 참조하면, 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화장치(300) 간에 미리 정의된 판단 함수(수학식 6에서의 g())를 이용하여 움직임 벡터 부호화 장치(200)는 판단 함수값을 계산하고(S800), 계산된 판단 함수값을 토대로 수학식 6과 같이 최소값의 판단 함수값을 갖는 후보 움직임 벡터를 추정 예측 움직임 벡터(PMV_{dec})로 선택한다(S802). 이후, 추정 예측 움직임 벡터(PMV_{dec})와 최적 예측 움직임 벡터(PMV_{enc})를 비교하여(S804), 최종적으로 예측가능 여부를 판단하게 된다(S806).

[147] 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화장치(300) 간에 미리 정의된 판단 함수는 다양한 형태로 실현될 수 있다.

[148] 수학식 6의 판단 함수 g()로서는, 템플릿 매칭(TM: Template Matching)을 이용한 함수, 경계 화소 정합(BM: Boundary Matching)을 사용한 함수를 사용할 수 있다.

[149] 먼저, 템플릿매칭(TM: Template Matching)을 이용한 판단 함수를 아래에서

설명하면, 탬플릿 매칭 화소 인덱스 집합(TMS: Template Matching Set)은 주어진 지정된 블록의 위치를 기준으로, 선택된 화소들의 상대적 위치를 나타내는 인덱스 집합으로 정의할 수 있는데, 예를 들어, 지정된 블록의 좌측, 좌측 상단 및 상단에 인접한 주변의 M개 화소의 위치이다. 필요에 따라 다른 방법도 가능함은 물론이다. 일반적으로 TMS가 지시하는 화소의 수가 많으면 더욱 정확한 매칭이 가능하나 계산량이 많아진다. 템플릿 매칭 방법은, 예측 움직임 벡터로 선택 가능한 모든 후보 예측 움직임 벡터 집합(CS)의 선정 이후, 선정된 후보 예측 움직임 벡터 집합 내의 각각의 후보 움직임 벡터가 지정하는 참조 블록에 대해 TMS가 가리키는 화소들과 현재 블록에 대해 TMS가 지시하는 화소들의 차이를 수학식 7(수학식 6의 일 실시예)을 이용하여 계산하여 각각의 후보 움직임 벡터에 따른 정합 오류를 계산한 후 이중 가장 최소의 정합 오류를 가지는 것을 앞에서 언급한 추정 예측 움직임 벡터(PMV_{dec})로 선정한다.

[150] 수학식 7

$$PMV_{dec} = \arg\min_{PMVC \in CS} g(PMVC|DMV)$$

$$g(PMVC|DMV) = \sum_{i \in TMS} [f(PMVC+DMV, i) - C(i)]^2$$

[151] 수학식 7에서, f(PMVC+DMV, i)에서 (PMVC+DMV)는 인덱스 i(TMS에 포함됨)가 가리키는 참조 픽처 내의 참조 블록 주변의 인덱스 i 가 나타내는 화소 위치를 나타내고, f(PMVC+DMV, i)는 이 위치에서의 화소값을 의미한다. 또한 C(i)는 인덱스 i가 지시하는 현재 블록 주변의 화소값을 의미한다.

[152] 판단 함수의 일 실시예인 g(PMVC|DMV)는 움직임 벡터 부호화 장치(200)가 제공하는 차분 벡터(DMV^D)에 후보 움직임 벡터 집합(CS) 중의 한 원소인 후보 움직임 벡터(PMVC)를 예측 움직임 벡터로 사용하여 해당블록의 움직임 벡터를 PMVC+DMV로 계산한 후, 이 움직임 벡터를 이용하여 해당 블록을 움직임 보상하여 얻어진 복원된 블록이 얼마나 올바른지를 산정한 값을 제공한다. 이를 산정하기 위하여 수학식 7에서는 차이의 제곱합(Sum of Squared Error)을 사용하였지만, 응용에 따라 SAD(sum of Absolute difference) 등 다른 방법도 가능하다. 추정 예측 움직임 벡터(PMV_{dec})는 판단 함수의 일 실시예인 g(PMVC|DMV)를 최소로 하는 후보 움직임 벡터(PMVC)를 의미한다.

[153] 즉, 도 8에서 단계 S800는, 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각이 지정하는 참조 블록에 대해 탬플릿 매칭 화소 인덱스 집합(TMS: Template Matching Set)이 지시하는 화소들과 현재 블록에 대해 탬플릿 매칭 화소 인덱스가 지시하는 화소들의 화소값 차이를 계산하고, 계산된 화소값 차이에 근거하여, 하나 이상의 후보 움직임 벡터 각각에 대한 정합 오류를 판단 함수값으로 계산함으로써, 기정의된 판단 함수를 수행하여 판단

함수값을 계산할 수 있다.

[154] 다음으로, 경계화소정합(BM: Boundary Matching)을 이용한 판단 함수를 설명하면, 전술한 일 실시예의 TMS와 유사하게, 경계 매칭 화소 인덱스 집합 BMS(Boundary Matching Index Set)는 현재 블록 내의 최좌측 및 최상단에 위치한 화소값의 위치를 가리키는 인덱스의 집합으로 정의할 수 있다. 응용에 따라 현재 블록 내의 블록 경계에 위치한 전부 또는 일부의 화소들의 위치로 정의하는 것도 가능함은 물론이다.

[155] 예측 움직임 벡터로 선택 가능한 모든 후보 움직임 벡터 집합(CS)을 선정한 이후, 후보 움직임 벡터 집합(CS) 중의 어느 후보 움직임 벡터(PMVC)가 가장 최적인지를 알기 위해, 움직임 보상으로 부호화된 것을 복원한 현재 블록이 그 주변 블록들과 불록 경계에서 얼마나 부드럽게 연결되는 지를 산정하기 위해 경계 화소 매칭을 수행하여 이 경계 화소 매칭 정합 오류를 최소화하는 후보 움직임 벡터(PMVC)를 후보 움직임 벡터 집합(CS) 중에서 정하여 이를 추정 예측 움직임 벡터(PMV_{dec})로 선택한다. 이를 움직임 벡터 복호화 장치(300)가 선정한 예측 움직임 벡터라 한다.

[156] 이 과정을 좀 더 자세히 설명하면, 후보 움직임 벡터 집합(CS) 중의 한 원소인 후보 움직임 벡터(PMVC)와 움직임 벡터 부호화 장치(200)가 결정한 차분 벡터(DMV)를 사용하여 PMVC+DMV로 계산된 후보 움직임 벡터가 지정하는 참조 블록의 값과 움직임 벡터 부호화 장치(200)가 계산한 차분 벡터를 더하여 현재 블록의 화소값을 복원 한 후, 각각의 후보 움직임 벡터에 따라 움직임 보상 복원된 현재 블록 중에서 현재 블록 주변의 기 복원된 화소들과의 경계 정합도가 가장 높은 복원블록을 생성하는 예측 움직임 벡터를 결정한다. 이를 위해, 수학식 8과 같이 차이값의 제곱합으로 각각의 후보 움직임 벡터가 가지는 정합 오류를 계산한다. 물론 SAD(Sum of Absolute Difference) 등 다른 방법으로 정합오류를 계산할 수 있음은 자명하다.

[157] 수학식 8

$$PMV_{dec} = \arg\min_{PMVC \in CS} g(PMVC|DMV)$$

$$g(PMVC|DMV) = \sum_{i \in BMS} [f(i) - C(i)]^2$$

[158] 수학식 8에서, C(i)는 후보 움직임 벡터 집합(CS) 중의 한 원소인 후보 움직임 벡터(PMVC)와 움직임 벡터 부호화 장치(200)가 결정한 DMV를 사용하여 PMVC+DMV로 계산된 후보 움직임 벡터가 지정하는 참조 블록의 값과 움직임 벡터 부호화 장치(200)가 계산한 움직임 보상에 따른 화소의 잔차 신호를 더하여 복원된 현재 블록의 복원 화소 중 경계 매칭 화소 인덱스 집합 BMS(Boundary

Matching Set) 내의 인덱스 i가 지정하는 화소값을 의미한다. 또한, f(i)는 현재 블록에 인접하는 주변 블록 내의 경계 화소 중 BMS의 인덱스 i가 지정하는 화소와 바로 인접하는 화소값을 의미한다. 수학식 8을 사용하여 후보 움직임 벡터 집합(CS) 내의 각각의 후보 움직임 벡터(PMVC)에 대해 경계 화소 정합 오류가 계산되고, 그 중 최소의 정합 오류를 발생하는 후보 움직임 벡터를 추정 예측 움직임 벡터(PMV_{dec})로 선택한다. 이를 움직임 벡터 복호화 장치(300)가 선정한 예측 움직임 벡터라 한다.

[159] 즉, 도 8에서 단계 S800는 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각에 대하여 움직임 벡터 부호화 장치(200)에서 결정한 차분 벡터와 해당 후보 움직임 벡터를 사용하여 계산된 후보 움직임 벡터가 지정하는 참조 블록의 값과 움직임 벡터 부호화 장치(200)가 계산한 움직임 보상에 따른 잔차 신호를 더하여 복원된 현재 블록의 복원 화소 중 경계 매칭 화소 인덱스 집합(BMS:Boundary Matching Set) 내의 인덱스가 지정하는 화소값; 및 현재 블록에 인접하는 주변 블록 내의 경계화소 중 경계 매칭 화소 인덱스 집합의 인덱스가 지정하는 화소와 인접한 화소값의 차이에 근거하여, 하나 이상의 후보 움직임 벡터 각각에 대한 정합 오류를 판단 함수값으로 계산함으로써, 기 정의된 판단 함수를 수행하여 판단 함수값을 계산할 수도 있다.

[160] 도 8을 참조하여 단계 S506단계를 정리하면, S800 및 S802단계에서, 움직임 벡터 부호화 장치(200)는 템플릿 매칭 또는 경계 화소 매칭 등의 정해진 결정 방법에 따라 결정된 최소 정합 오류를 갖는 추정 예측 움직임 벡터(PMV_{dec})를 선택한다.

[161] 단계 S804에서는, 선택된 추정 예측 움직임 벡터(PMV_{dec})와 단계 S502에서 선택한 최적 예측 움직임 벡터(PMV_{enc})가 동일한지 여부를 비교한다. 단계 S806에서는, 전술한 판단 방식에 따라, 추정 예측 움직임 벡터(PMV_{dec})와 최적 예측 움직임 벡터(PMV_{enc})가 동일하다면 움직임 벡터 복호화 장치(300)가 현재 블록의 현재 움직임 벡터의 최적 예측 움직임 벡터를 예측할 수 있는 것으로 판단하고, 그렇지 않은 경우에는 예측할 수 없다고 판단한다. 또한, 전술한 것처럼, 추정 예측 움직임 벡터(PMV_{dec})와 최적 예측 움직임 벡터(PMV_{enc})가 동일하지 않더라도 각각을 이용하여 복원된 복원 영상이 동일하거나, 그 차이가 소정의 경계치보다 작은 경우는 이 두개의 추정 예측 움직임 벡터(PMV_{dec})와 최적 예측 움직임 벡터(PMV_{enc})가 같다고 간주하여 움직임 벡터 복호화 장치(300)가 현재 블록의 현재 움직임 벡터의 최적 예측 움직임 벡터를 예측할 수 있는 것으로 판단할 수도 있다.

[162] 도 9는 본 발명의 일 실시예에 따라 최적 예측 움직임 벡터를 이용하여 현재 움직임 벡터를 부호화하는 과정을 설명하기 위한 순서도이다.

[163] 도 9를 참조하면, 최적 예측 움직임 벡터를 이용하여 현재 움직임 벡터를 부호화하는 단계(S508)는 선택된 최적 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계(S900); 현재 움직임 벡터와 선택된 최적

예측 움직임 벡터의 차분 벡터를 계산하는 단계(S902); 및 계산된 차분 벡터를 부호화함으로써 현재 움직임 벡터를 부호화하고 예측 가능을 지시하는 움직임 벡터 예측모드를 제 3 부호화기(250)에 전달하는 단계(S904) 등을 포함한다.

[164] 도 10는 본 발명의 일 실시예에 따라 디폴트 예측 움직임 벡터를 이용하여 현재 움직임 벡터를 부호화하는 과정을 설명하기 위한 순서도이다.

[165] 도 10를 참조하면, 디폴트 예측 움직임 벡터를 이용하여 현재 움직임 벡터를 부호화하는 단계(S510)는 기설정된 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계(S1000); 현재 움직임 벡터와 기 설정된 디폴트 예측 움직임 벡터의 차분 벡터를 계산하는 단계(S1002); S504 단계에 의한 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하는 단계(S1004); 움직임 벡터 예측 모드가 비경쟁 예측 모드가 아닌 경우 즉, 움직임 벡터 예측 모드의 생략이 불가능한 경우, 계산된 차분 벡터를 부호화함으로써 현재 움직임 벡터를 부호화하고 예측 불가능을 지시하는 움직임 벡터 예측 모드를 제 3 부호화기(250)에 전달하는 단계(S1006); 및 움직임 벡터 예측 모드가 비경쟁 예측 모드인 경우 즉, 움직임 벡터 예측 모드의 생략이 가능한 경우, 계산된 차분 벡터를 부호화함으로써 현재 움직임 벡터를 부호화하고 움직임벡터 예측 모드의 부호화를 생략하는 단계(S1008) 등을 포함한다.

[166] 단계 S1002에서의 기 설정된 디폴트 예측 움직임 벡터는, 현재 블록에 인접하는 하나 이상의 주변 블록의 움직임 벡터의 중간값(Median)이고, 움직임 벡터 부호화 장치(200) 및 움직임 벡터 복호화 장치(300) 간에 기 설정된다.

[167] 도 11는 본 발명의 일 실시예에 따라 움직임 벡터 예측 모드를 부호화하는 과정을 설명하기 위한 순서도이다.

[168] 도 11를 참조하면, 움직임 벡터 예측 모드를 부호화하는 단계(S514)는 움직임 벡터가 존재하는 소정의 비독립적 부호화 단위의 상위에 해당하는 독립적 부호화 단위 내에 존재하는 움직임 벡터 예측 모드의 개수를 계산하고 부호화하는 단계(S1110); 및 움직임 벡터가 존재하는 소정의 비독립적 부호화 단위의 상위에 해당하는 독립적 부호화 단위 내에 존재하는 움직임 벡터 예측 모드를 모아서 부호화하는 단계(S1120) 등을 포함한다.

[169] 도 12는 본 발명의 일 실시예에 따라 움직임 벡터 예측 모드의 개수 및 움직임 벡터 예측 모드를 모아서 부호화하는 경우의 규정을 설명하기 위한 예시도이다.

[170] 도 12를 참조하면, 움직임 벡터가 부호화되는 블록 단위가 아닌 상위에 해당하는 독립적 부호화 단위인 슬라이스 단위와 같은 상위 단위로 상위 내에 존재하는 움직임 벡터 예측 모드의 개수 및 움직임 벡터 예측 모드를 모아서 부호화하는 것을 알 수 있다.

[171] 이하에서는, 도 3을 참조하여 전술한 본 발명의 일 실시예에 따른 움직임 벡터 복호화 장치(300)에 의한 움직임 벡터 복호화 방법을 도 13 내지 도 15를 참조하여 더욱 상세하게 설명한다.

[172] 도 13은 본 발명의 일 실시예에 따른 움직임 벡터 복호화 방법을 설명하기 위한

순서도이다.

[173] 도 13을 참조하면, 움직임 벡터 복호화 장치(300)는 비트스트림을 복호화하여 현재 움직임 벡터와 예측 움직임 벡터 간의 차분 벡터 및 현재 움직임 벡터의 움직임 벡터 예측 모드를 복원하고(S1300), 움직임 벡터 예측 모드에 근거하여 현재 움직임 벡터의 예측 움직임 벡터를 선택하하며(S1302), 선택된 예측 움직임 벡터를 이용하여 현재 움직임 벡터를 복원한다(S1304).

[174] 단계 S1300에서, 움직임 벡터 부호화 장치(200)는 선택적으로 부호화된 움직임 벡터 예측 모드와 차분 벡터를 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300) 사이에 미리 약속된 서로 다른 소정의 독립적 또는 비독립적 단위에 따라 복호화한다. 즉, 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300) 사이에 기 약속된 소정의 판단 과정을 통해 움직임 벡터 예측 모드의 존재 여부를 판단한다.

[175] 만약, 움직임 벡터 예측 모드가 존재하지 않는 것으로 판단되면 움직임 벡터 예측 모드를 비경쟁 예측 모드로 정의하고, 단계 S1302에서 예측 움직임 벡터를 디폴트 예측 움직임 벡터로 선택한다. 만약, 단계 S1300에서 움직임 벡터 예측 모드가 존재하는 것으로 판단되면, 단계 S1300에서 복원된 상위 단위 내의 경쟁 예측 모드에 해당하는 블록의 움직임 벡터 예측 모드로부터 해당 블록의 움직임 벡터 예측 모드를 나타내는 비트값을 읽고, 이 값이 지시하는 움직임 벡터를 해당 블록의 예측 움직임 벡터로 선택한다.

[176] 단계 S1302에서, 움직임 벡터 복호화 장치(300)는 움직임 벡터 예측 모드를 나타내는 비트값이 움직임 벡터 예측 모드가 경쟁 예측 모드이지만 예측 불가능 모드를 나타내는 경우(예를 들어, 비트값이 '0'인 경우), 기 약속된 방식에 따라 결정되는 디폴트 예측 움직임 벡터를 선택한다. 또한, 단계 S1302에서, 움직임 벡터 복호화 장치(300)는 움직임 벡터 예측 모드를 나타내는 비트값이 움직임 벡터 예측모드가 경쟁 예측 모드이면서 예측 가능 모드를 나타내는 경우(예를 들어, 비트값이 '1'인 경우), 소정의 방법을 이용하여 복수 개의 후보 움직임 벡터 중에서 하나의 후보 움직임 벡터를 현재 움직임 벡터의 예측 움직임 벡터 즉, 최적 예측 움직임 벡터로 선택한다. 여기서, 소정의 방법은 탬플릿 매칭 및 경계 화소 매칭 중 하나에 기초한 방법일 수 있다.

[177] 이하에서는 도 13을 참조하여 전술한 본 발명의 일 실시예에 따른 움직임 벡터 복호화 방법의 구체적인 구현 예를 도 14 및 도 15를 참조하여 더욱 상세하게 설명한다.

[178] 도 14는 본 발명의 일 실시예에 따른 움직임 벡터 복호화 방법의 구체적인 구현 예를 설명하기 위한 순서도이다.

[179] 도 14를 참조하면, 움직임 벡터 복호화 장치(300)는 움직임 벡터 부호화 장치(200)에 의해 부호화된 움직임 벡터 예측 모드와 부호화된 차분 벡터를 복호화하는 단계(S1400); 움직임 벡터 예측 모드의 존재 여부를 판단하는 단계(S1402); 복호화된 움직임 벡터 예측 모드가 예측 가능을 지시하는지 또는

예측 불가능을 지시하는지를 판단하는 단계(S1404); 복호화된 움직임 벡터 예측 모드가 예측 가능을 지시하면, 현재 블록의 현재 움직임 벡터를 복원하기 위한 예측 움직임 벡터로 선택가능한 후보 움직임 벡터 집합을 선정하고, 선정된 후보 움직임 벡터 집합에서 최적 예측 움직임 벡터를 선택하며, 선택된 최적 예측 움직임 벡터를 현재 움직임벡터에 대한 예측 움직임 벡터로 결정하는 단계(S1406); 움직임 벡터 예측 모드가 존재하지 않거나 복호화된 움직임 벡터 예측모드가 예측 불가능을 지시하면, 기 설정된 방식에 따라 결정되는 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계(S1408); 및 S1406 단계 또는 S1408 단계에서 결정된 예측 움직임 벡터와 복호화된 차분 벡터를 더하여 현재 블록의 현재 움직임 벡터를 복원하는 단계(S1410) 등을 포함한다.

[180] S1400 단계에서의 부호화된 차분 벡터는, 현재 움직임 벡터와 기 설정된 디폴트 예측 움직임 벡터의 차분 벡터이거나, 현재 움직임 벡터와 최적 예측 움직임벡터의 차분 벡터일 수 있다.

[181] 도 15는 본 발명의 일 실시예에 따라 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 과정을 설명하기 위한 순서도이다.

[182] 도 15를 참조하면, 본 발명의 일 실시예에 따라 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계(S1406)는 현재 블록의 현재 움직임 벡터를 복원하기 위한 예측 움직임 벡터로 선택가능한 후보 움직임 벡터 집합을 선정하는 단계(S1500); 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)에서 공통적으로 정의된 판단 함수를 이용하여, 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각에 대하여 판단 함수값을 계산하는 단계(S1502); 및 계산된 판단 함수값 중에서 최소값을 갖는 후보 움직임 벡터를 최적 예측 움직임 벡터로 선택하고, 선택된 최적 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 선택하는 단계(S1504) 등을 포함한다.

[183] 단계 S1504에서, 움직임 벡터 복호화 장치(300)가 현재 움직임 벡터에 대한 예측 움직임 벡터로서 결정한 최적 예측 움직임 벡터는 움직임 벡터 부호화 장치(200)에서 움직임 벡터 복호화 장치(300)가 찾을 것으로 추정한 추정 예측 움직임 벡터(PMV_{dec})와 동일하거나 소정 경계값의 차이를 보일 수 있다.

[184] 또한, 단계 S1502에서, 움직임 벡터 복호화 장치(300)는 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각이 지정하는 참조 블록에 대해 탬플릿 매칭 화소 인덱스 집합(TMS: Template Matching Set)이 지시하는 화소들과 현재 블록에 대해 탬플릿 매칭 화소 인덱스가 지시하는 화소들의 화소값 차이를 계산하고, 계산된 화소값 차이에 근거하여, 하나 이상의 후보 움직임 벡터 각각에 대한 정합 오류를 판단 함수값으로 계산한다.

[185] 또한 단계 S1502에서, 움직임 벡터 복호화 장치(300)는 선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각에 대하여 움직임 벡터

부호화 장치(200)에서 결정한 차분 벡터와 해당 후보 움직임 벡터를 사용하여 계산된 후보 움직임 벡터가 지정하는 참조 블록의 값과 움직임 벡터 부호화 장치(200)가 계산한 움직임 보상에 따른 잔차 신호를 더하여 복원된 현재 블록의 복원 화소 중 경계 매칭 화소 인덱스 집합(BMS:Boundary Matching Set) 내의 인덱스가 지정하는 화소값; 및 현재 블록에 인접하는 주변 블록 내의 경계 화소 중 경계 매칭 화소 인덱스 집합의 인덱스가 지정하는 화소와 인접한 화소값의 차이에 근거하여, 하나 이상의 후보 움직임 벡터 각각에 대한 정합 오류를 판단 함수값으로 계산한다.

[186] 도 16은 본 발명의 다른 실시예에 따른 움직임 벡터 복호화 방법을 설명하기 위한 순서도이다.

[187] 도 16을 참조하면, 본 발명의 다른 실시예에 따라 움직임 벡터 복호화 장치(300)에 의해 제공되는 움직임벡터 복호화 방법은, 움직임 벡터 부호화 장치(200)에 의해 부호화된 움직임 벡터 예측 모드와 부호화된 차분 벡터를 복호화하는 단계(S1600); 움직임 벡터 예측 모드의 존재여부를 판단하는 단계(S1602); 복호화된 움직임 벡터 예측 모드가 예측가능을 지시하는지 예측 불가능을 지시하는지를 판단하는 단계(S1604); 복호화된 움직임 벡터 예측 모드가 예측가능을 지시하면, 현재 블록의 현재 움직임 벡터를 복원하기 위한 예측 움직임 벡터로 선택가능한 후보 움직임 벡터 집합을 선정하고, 선정된 후보 움직임 벡터 집합에서 최적 예측 움직임 벡터를 선택하며, 선택된 최적 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계(S1606); 움직임 벡터 예측 모드가 존재하지 않거나 복호화된 움직임 벡터 예측 모드가 예측 불가능을 지시하면, 기 설정된 방식에 따라 결정되는 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계(S1608); 단계 S1606 또는 단계 S1608에서 결정된 예측 움직임 벡터와 복호화된 차분 벡터를 더하여 현재 블록의 현재 움직임 벡터를 복원하는 단계(S1610); 움직임 벡터 예측 모드의 개수를 이용한 정상 복호화 판단 단계(S1612); 및 비정상 복호화를 위한 선택적 추가 처리 단계(S1614) 등을 포함할 수 있다.

[188] 본 발명의 실시예에서는 움직임 벡터 부호화 장치(200)와 움직임 벡터 복호화 장치(300)가 미리 약속된 소정의 판단 과정을 공유한다는 전제 하에서, 스스로 그리고 독립적으로 부호화 또는 복호화하고자 하는 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단할 수 있으므로, 블록의 움직임 벡터 예측 모드가 경쟁 예측 모드인 경우에만 예측 가능 모드와 예측 불가능 모드를 구분하는 비트만을 전송하고 비경쟁 예측 모드인 경우에는 움직임 벡터 예측 모드에 대한 부호화를 생략함으로써, 현재 움직임 벡터를 부호화하는 데 소요되는 비트량을 크게 절감할 수 있다.

[189] 이때, 움직임 벡터 부호화 장치(200)가 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하기 위해서는 이전에 복호화한 주변 블록의 정보가

정확하다는 또 다른 전제가 부합해야 한다. 즉, 이전에 복호화한 주변 블록의 복원된 움직임 벡터에 오류가 발생하여 부호화할 때 사용한 움직임 벡터가 아닌 다른 움직임 벡터로 복원된 경우, 복호화하고자 하는 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단할 때, 오류가 발생하여 그로 인해 부호화 장애(Decoder Crash)가 발생할 수 있다.

[190] 예를 들어, 이전에 복호화한 주변 블록의 복원된 움직임 벡터에 오류가 없는 경우, 복호화하고자 하는 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드라고 가정한 상태에서, 만약 이전에 복호화한 주변 블록의 복원된 움직임 벡터에 오류가 발생하고 그로 인해 복호화하고자 하는 블록의 움직임 벡터 예측 모드가 경쟁 예측 모드로 판단이 되었다면, 움직임 벡터 복호화 장치(300)는 복원한 움직임 벡터 예측 모드의 비트값을 읽어서 읽혀진 비트값에 따라 예측 가능 모드 또는 예측 불가능 모드를 판단하여 최적 예측 움직임 벡터 또는 디폴트 예측 움직임 벡터를 현재 블록에 대한 예측 움직임 벡터로 결정한다. 하지만, 이전에 복호화한 주변 블록의 복원된 움직임 벡터에 오류가 없었다면, 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드라고 판단될 것이므로, 복원한 움직임 벡터 예측 모드의 비트값을 읽지 말아야 하며, 그로 인해 이후 블록에서 이용해야할 움직임 벡터 예측 모드의 비트값이 비정상적으로 사용된 결과를 초래하여, 이후 블록에서 복원된 움직임 벡터 예측 모드의 비트값을 읽으려할 때 비트값이 부족하게 되며, 이 경우 복호화기 장애가 발생하여 영상을 복호화하는 전체 과정이 중단될 수 있다.

[191] 따라서, 이러한 문제점을 해결하기 위해, 본 발명의 실시예에서는 움직임 벡터 부호화 장치(200)에서 움직임 벡터 예측 모드를 부호화할 때, 움직임 벡터 예측 모드를 나타내는 비트(즉, 예측 가능 모드를 나타내는 비트 및/또는 예측 불가능 모드를 나타내는 비트)뿐만 아니라, 움직임 벡터 예측 모드의 개수를 나타내는 비트를 함께 부호화하여 움직임 벡터가 결정되는 블록 단위보다 상위인 슬라이스 단위 또는 픽처 단위 등의 상위 단위로 비트스트림에 삽입한다.

[192] 또한, 움직임 벡터 복호화 장치(300)에서는 후술하는 바와 같이, 비트스트림으로부터 상위 단위로 삽입된 부호화된 움직임 벡터 예측 모드를 파싱하여 블록 단위로 진행되는 복호화 과정과는 독립적으로 복호화하며, 움직임 벡터 예측 모드를 나타내는 비트를 출력할 뿐만 아니라, 움직임 벡터 예측 모드의 개수를 획득하여 획득한 움직임 벡터 예측 모드의 개수와 복호화된 블록 중 움직임 벡터 예측 모드가 존재하는 블록의 개수를 비교하여 그 개수가 다른 경우, 오류가 발생한 것으로 판단하여 후속 조치를 취할 수 있다. 즉, 움직임 벡터 복호화 장치(300)는 움직임 벡터 예측 모드의 개수를 이용하여 오류의 발생 여부를 스스로 판단할 수 있으며, 오류가 발생한 것으로 판단되면 스스로 리셋하거나 스스로 오류를 추적하여 움직임 벡터의 정상적인 복원을 시도할 수 있다.

[193] 이러한 움직임 벡터 복호화 장치(300)의 동작을 도 16을 통해 설명하면, 움직임

벡터 예측 모드의 개수를 이용한 정상 복호화 판단 단계(S1612)는 부호화된 움직임 벡터 예측 모드와 부호화된 차분 벡터를 복호화하는 단계(S1400)에서 획득한 움직임 벡터 예측 모드의 개수와 소정의 독립적 부호화 단위인 슬라이스와 같은 상위 단위 내의 모든 블록이 모두 복호화 될 때까지 움직임 벡터 예측 모드의 존재 여부를 판단하는 단계(S1402)를 통해 계산된 움직임 벡터 예측 모드의 개수가 다를 경우, 해당 비트스트림이 정상적으로 복호화되지 못한 것으로 판단할 수 있다. 만약 해당 비트스트림이 정상적으로 복원되지 못한 경우, 비정상 복호화를 위한 선택적 추가 처리 단계(S1614)를 통해 다양한 방법으로 움직임벡터 복원을 시도할 수 있다.

[194] 예를 들어, 움직임 벡터 부호화 장치(200)가 전달한(또는 단계 S1400를 통해 획득한) 움직임 벡터 예측 모드의 개수가 해당 상위 단위(예를 들면, 슬라이스)에 대한 복호화 과정이 모두 종료된 후에 계산된 움직임 벡터 예측 모드의 개수에 비해 많은 경우, 남은 움직임 벡터 예측 모드의 모임을 무시하고 모두 움직임 벡터 예측 모드가 존재하지 않는 것으로 가정할 수 있다. 반면에, 움직임 벡터 예측 모드의 개수가 상위 단위에 대한 복호화 과정이 모두 종료된 후에 계산된 움직임 벡터 예측 모드의 개수에 비해 적은 경우, 저장(또는 전송)된 비트스트림에 오류가 발생해 정상적으로 복원되지 못했음을 의미하므로 저장(또는 전송)된 비트스트림 자체를 무시하고 오류 은닉(Error Concealment) 방법을 적용해 영상을 복원할 수도 있다.

[195] 도 4 내지 도 16에서 도시한 순서도와 그를 통한 설명에서의 각 단계의 순서는 본 발명의 실시예를 설명하기 위한 구현 예에 불구하다. 따라서, 각 단계의 순서는 구현 방식에 따라 변경될 수 있다. 즉, 일부 또는 전부의 단계에 대한 순서는 선택적으로 변경될 수 있으며, 경우에 따라서는 일부 또는 전부의 단계가 병행적으로 수행될 수도 있을 것이다.

[196] 도 17은 본 발명의 일 실시예에 따른 영상 부호화 장치를 간략하게 나타낸 블록 구성도이다.

[197] 본 발명의 일 실시예에 따른 영상 부호화 장치(1700)는 텍스처 부호화기(1710), 움직임 벡터 부호화기(1720) 및 멀티플렉서(1730)를 포함하여 구성될 수 있다.

[198] 텍스처 부호화기(1710)는 입력 영상에서 부호화하고자 하는 현재 블록의 현재 움직임 벡터에 따라 현재 블록을 예측하여 현재 블록과 예측 블록 간의 차인 잔여 블록을 부호화하여 부호화된 텍스처(Texture)를 출력한다. 텍스처 부호화기(1710)에 대해서는 후술하는 과정에서 도 18을 통해 상세히 설명한다.

[199] 움직임 벡터 부호화기(1720)는 복수 개의 움직임 벡터 예측 모드 중 하나의 움직임 벡터 예측 모드를 선택하고, 선택된 움직임 벡터 예측 모드에 따라 현재 움직임 벡터에 대한 예측 움직임 벡터를 선택하며, 현재 움직임 벡터와 선택된 예측 움직임 벡터 간의 차인 차분 벡터를 부호화하며, 선택된 움직임 벡터 예측 모드에 따라 움직임 벡터 예측 모드를 부호화하여 부호화된 차분 벡터 및 부호화된 움직임 벡터 예측 모드를 출력한다. 여기서, 움직임 벡터

부호화기(1720)는 도 2를 통해 전술한 움직임 벡터 부호화 장치(200)의 기능과 동일 또는 유사한 기능을 수행하므로 상세한 설명을 생략한다.

[200] 멀티플렉서(1730)는 부호화된 텍스처, 부호화된 차분 벡터 및 부호화된 움직임 벡터 예측 모드를 포함하는 비트스트림을 생성하여 출력한다. 여기서, 멀티플렉서(1730)는 움직임 벡터 부호화기(1720)로부터 출력되는 부호화된 움직임 벡터 예측 모드를 비트스트림에 삽입하는 데 있어서, 비트스트림에서 부호화된 텍스처 및 부호화된 차분 벡터가 삽입되는 블록 단위가 아닌 슬라이스 단위 또는 픽처 단위 등의 상위 단위에 삽입함으로써, 영상 복호화 장치에서 부호화된 움직임 벡터 예측 모드를 독립적으로 복호화할 수 있도록 할 수 있다.

[201] 도 18은 본 발명의 일 실시예에 따른 영상 부호화 장치의 구체적인 구현 예를 나타낸 예시도이다.

[202] 도 17을 통해 전술한 영상 부호화 장치(1700)를 텍스처 부호화기(1710)의 구체적인 구성을 포함하여 나타내면, 도 18에 도시한 영상 부호화 장치(1800)로 나타낼 수 있다.

[203] 텍스처 부호화기(1710)는 움직임 추정기(1812)와 움직임 보상기(1814)를 포함하는 예측기(1810), 감산기(1820), 변환기(1830), 양자화기(1840), 부호화기(1850), 역 양자화기(1860), 역 변환기(1870), 가산기(1880) 및 디블로킹 필터(1890)를 포함하여 구성될 수 있다.

[204] 예측기(1810)는 현재 블록을 예측하여 예측 블록을 생성하고 출력한다. 즉, 예측기(1810)는 영상에서 부호화하고자 하는 현재 블록의 각 화소의 화소값(Pixel Value)을 예측하여 예측된 각 화소의 예측 화소값(Predicted Pixel Value)을 갖는 예측 블록(Predicted Block)을 생성한다. 이러한 예측기(1810)는 인터 예측을 수행하는 경우에는 도시한 바와 같이, 움직임 벡터 추정기(1812) 및 움직임 보상부(1814)를 포함하여 구성될 수 있다.

[205] 움직임 벡터 추정기(1812)는 현재 블록에 대한 블록 모드 또는 기 설정된 블록 모드에 대응하는 블록 단위(예를 들어, 16×16 블록, 16×8 블록, 8×16 블록, 8×8 블록, 8×4 블록, 4×8 블록, 4×4 블록)로 하나 이상의 참조 픽처를 참조하여 현재 블록의 움직임 벡터를 추정하여 현재 움직임 벡터를 결정한다. 움직임 벡터 추정기(1812)에 의해 결정된 현재 움직임 벡터는 움직임 벡터 부호화기(1720)로 출력되고, 움직임 벡터 부호화기(1720)에 의해 차분 벡터와 움직임 벡터 예측 모드가 부호화된다.

[206] 여기서, 움직임 벡터 추정기(1812)가 참조 픽처를 이용하여 현재 블록의 움직임 벡터를 추정하여 현재 움직임 벡터를 결정하는 것에 대해서는 당업자에게 자명하므로 이에 대한 상세한 설명은 생략한다.

[207] 움직임 보상기(1814)는 움직임 벡터 추정기(1812)로부터 출력되는 현재 움직임 벡터를 이용하여 현재 블록의 예측 블록을 생성하고 출력한다. 즉, 움직임 보상기(1814)는 참조 픽처에서 현재 움직임 벡터가 지시하는 블록을 예측 블록으로 출력한다.

[208] 감산기(1820)는 현재 블록에서 예측 블록을 감산하여 잔여 블록(Residual Block)을 생성한다. 즉, 감산기(1820)는 부호화하고자 하는 현재 블록의 각 화소의 화소값과 예측기(1810)에서 예측한 예측 블록의 각 화소의 예측 화소값의 차이를 계산하여 블록 형태의 잔여 신호(Residual Signal)를 갖는 잔여 블록을 생성한다.

[209] 변환기(1830) 및 양자화기(1840)는 잔여 블록을 변환 및 양자화하여 양자화된 잔차 블록을 출력한다. 즉, 변환기(1830) 및 양자화기(1840)는 잔여 블록의 잔여 신호를 주파수 영역으로 변환하여 잔여 블록의 각 화소값을 변환 계수(Transform Coefficient)로 변환하고, 주파수 계수를 갖는 잔여 블록을 양자화(Quantization)한다.

[210] 여기서, 변환기(1830)는 하다마드 변환(Hadamard Transform), 이산 코사인 변환 기반 변환(DCT based Transform: Discrete Cosine Transform Based Transform) 등과 같은 공간축의 화상 신호를 주파수축으로 변환하는 다양한 변환 기법을 이용하여 잔여 신호를 주파수 영역으로 변환할 수 있는데, 주파수 영역으로 변환된 잔여 신호가 변환 계수가 된다. 또한, 양자화기(1840)는 변환된 잔여 블록을 데드존 균일 경계 양자화(DZUTQ: Dead Zone Uniform Threshold Quantization, 이하 'DZUTQ'라 칭함), 양자화 가중치 매트릭스(Quantization Weighted Matrix) 또는 이를 개량한 양자화 기법 등을 사용하여 양자화할 수 있다.

[211] 부호화기(1850)는 양자화기(1840)로부터 출력되는 잔여 블록을 부호화하여 부호화된 텍스처를 출력한다. 즉, 부호화기(1850)는 잔여 블록의 양자화 주파수 계수, 변환 계수 또는 잔여 신호를 지그재그 스캔과 같은 다양한 스캔 방식에 따라 스캔하여 양자화 변환 계수열, 변환 계수열 또는 신호열을 생성하고 엔트로피 부호화(Entropy Coding) 기법 등 다양한 부호화 기법을 이용하여 부호화한다.

[212] 부호화기(1850)로부터 출력되는 부호화된 텍스처는 움직임 벡터 부호화기(1720)로부터 출력되는 부호화된 차분 벡터 및 부호화된 움직임 벡터 예측 모드와 함께 멀티플렉서(1730)에서 비트스트림으로 생성되어 출력된다.

[213] 역양자화기(1860) 및 역 변환기(1870)는 양자화된 잔여 블록을 역 양자화(Inverse Quantization)하고 역 변환(Inverse Transform)한다. 즉, 역양자화기(1860) 및 역 변환기(1870)는 앙자화된 잔여 블록의 양자화 주파수 계수들을 역 양자화하여 변환 계수를 갖는 잔여 블록을 생성하고, 역 양자화된 잔여 블록을 역 변환하여 화소값을 갖는 잔여 블록 즉, 복원된 잔여 블록을 생성한다. 여기서, 역양자화기(1860) 및 역 변환기(1870)는 전술한 변환 방식과 양자화 방식을 역으로 사용하여 역 변환 및 역 양자화할 수 있다.

[214] 가산기(1880)는 예측기(1810)에서 예측된 예측 블록과 역 변환기(1870)에 의해 복원된 잔여 블록을 가산하여 현재 블록을 복원한다. 디블록킹 필터기(1890)는는 복원된 현재 블록을 디블로킹 필터링하여 출력한다. 여기서, 디블로킹 필터링이란 영상을 블록 단위로 부호화하면서 발생하는 블록 왜곡을

감소시키는 작업을 말하며, 블록 경계와 매크로블록 경계에 디블로킹 필터를 적용하거나 매크로블록 경계에만 디블로킹 필터를 적용하거나 디블로킹 필터를 사용하지 않는 방법 중 하나를 선택적으로 사용할 수 있다.

[215] 도 18에서, 변환기(1830) 및 양자화기(1840)는 제 1 부호화기로 구현되고 부호화기(1850)를 제 2 부호화기로 구현될 수 있는데, 제 1 부호화기는 텍스처 부호화기(1710)에 포함되어야 하는 것은 아니며, 구현 방식에 따라 선택적으로 포함될 수 있다. 즉, 감산기(1820)로부터 출력되는 잔여 신호는 변환과 양자화가 반드시 수행되어야 하는 것은 아니며 변환 및 양자화 중 어느 하나만을 수행할 수도 있으며, 심지어는 변환 및 양자화가 모두 생략될 수도 있다. 이 경우, 역 양자화기(1860), 역 변환기(1870)도 변환 및/또는 양자화의 수행 여부에 따라 선택적으로 포함될 수 있다. 또한, 디블로킹 필터(1890)도 텍스처 부호화기(1710)에 반드시 포함되어야 하는 것은 아니며, 구현 방식에 따라 선택적으로 포함될 수 있다.

[216] 이상에서 전술한 바와 같이, 영상 부호화 장치(1800)에 의해 비트스트림으로 부호화된 영상은 실시간 또는 비실시간으로 인터넷, 근거리 무선 통신망, 무선랜망, 와이브로망, 이동통신망 등의 유무선 통신망 등을 통하거나 케이블, 범용 직렬 버스(USB: Universal Serial Bus) 등의 통신 인터페이스를 통해 후술할 영상 복호화 장치로 전송되어 후술할 영상 복호화 장치에서 복호화되어 영상으로 복원되고 재생될 수 있다.

[217] 도 19는 본 발명의 일 실시예에 따른 영상 복호화 장치를 간략하게 나타낸 블록 구성도이다.

[218] 본 발명의 일 실시예에 따른 영상 복호화 장치(1900)는 디멀티플렉서(1910), 움직임 벡터 복호화기(1920) 및 텍스처 복호화기(1930)를 포함하여 구성될 수 있다.

[219] 디멀티플렉서(1910)는 비트스트림으로부터 부호화된 텍스처, 부호화된 차분 벡터 및 부호화된 움직임 벡터 예측 모드를 추출하여 출력한다. 여기서, 부호화된 텍스처는 텍스처 복호화기(1930)로 출력되고, 부호화된 차분 벡터 및 부호화된 움직임 벡터 예측 모드는 움직임 벡터 복호화기(1920)로 출력된다. 이때, 디멀티플렉서(1910)는 비트스트림의 슬라이스 헤더 또는 픽처 헤더 등 블록 단위 이상의 상위 단위에서 부호화된 움직임 벡터 예측 모드를 파싱하여 추출할 수 있다.

[220] 움직임 벡터 복호화기(1920)는 부호화된 차분 벡터 및 부호화된 움직임 벡터 예측 모드를 복호화하여 차분 벡터 및 움직임 벡터 예측 모드를 복원하고, 복원된 움직임 벡터 예측 모드에 근거하여 현재 블록의 현재 움직임 벡터에 대한 예측 움직임 벡터를 선택하며, 복원된 차분 벡터와 선택된 예측 움직임 벡터를 더하여 현재 움직임 벡터를 복원한다. 움직임 벡터 복호화기(1920)는 도 3을 통해 전술한 움직임 벡터 복호화 장치(300)와 동일 또는 유사한 기능을 수행하므로 상세한 설명은 생략한다.

[221] 텍스처 복호화기(1930)는 부호화된 텍스처를 복호화하여 잔여 블록을 복원하고, 움직임 벡터 복호화기(1920)에 의해 복원된 현재 움직임 벡터를 이용하여 현재 블록을 예측한 예측 블록과 복원된 잔여 블록을 가산하여 현재 블록을 복원하고 출력한다. 텍스처 복호화기(1930)의 구체적인 구현 에에 대해서는 후술하는 과정에서 도 20을 통해 설명한다.

[222] 도 20은 본 발명의 일 실시예에 따른 영상 복호화 장치의 구체적인 구현 예를 나타낸 예시도이다.

[223] 도 19를 통해 전술한 영상 복호화 장치(1900)를 텍스처 복호화기(1930)의 구체적인 구성을 포함하여 나타내면, 도 20에 도시한 영상 부호화 장치(2000)로 나타낼 수 있다.

[224] 텍스처 복호화기(1930)는 복호화기(2010), 역 양자화기(2020), 역 변환기(2030), 움직임 보상기(2040), 가산기(2050) 및 디블로킹 필터(2060)를 포함하여 구성될 수 있다.

[225] 복호화기(2010)는 디멀티플렉서(1910)로부터 출력되는 부호화된 텍스처를 복호화하여 양자화된 잔여 블록을 복원한다. 역 양자화기(2020)와 역 변환기(2030)는 도 18을 통해 전술한 역 양자화기(1860) 및 역 변환기(1870)와 동일 또는 유사한 기능을 수행하므로 상세한 설명은 생략한다. 다만, 영상 부호화 장치(1800)에서 잔여 블록을 변환 및 양자화 중 하나만을 수행하거나 하나도 수행하지 않은 경우, 역 양자화기(1860) 및 역 변환기(1870)도 그에 따라 하나만을 수행하거나 하나도 수행하지 않을 수 있다.

[226] 움직임 보상기(2040)는 움직임 벡터 복호화기(1930)로부터 출력되는 복원된 현재 움직임 벡터를 이용하여 현재 블록을 예측하여 예측 블록을 출력한다. 즉, 움직임 보상기(2040)는 하나 이상의 참조 픽처에서 복원된 현재 움직임 벡터가 지시하는 블록을 예측 블록으로 생성하여 출력한다.

[227] 가산기(2050)는 역 변환기(2030)로부터 출력되는 복원된 잔여 블록과 움직임 보상기(2040)로부터 출력되는 예측 블록을 가산하여 현재 블록을 복원하고 출력한다. 디블로킹필터(2060)는 복원된 현재 블록을 디블로킹 필터링하여 출력한다.

[228] 전술한 본 발명의 실시예에 따르면, 움직임 벡터 예측 모드는 경쟁 예측 모드와 비경쟁 예측 모드를 포함한다. 경쟁 예측 모드의 일 예로서 최적 예측 움직임 벡터의 예측 불가능을 나타내는 움직임 벡터 예측모드와 최적 예측 움직임벡터의 예측 가능을 나타내는 움직임 벡터 예측 모드가 설명되었다. 여기서, 최적 예측 움직임 벡터라 함은 미리 정해진 기준 또는 방법에 따라 유한한 개수의 후보 움직임벡터 집합 중에서 선택된 예측 움직임 벡터를 의미하는 것으로, 선택된 예측 움직임 벡터의 예측값이 항상 최적이 되는 것을 의미하는 것은 아니다.

[229] 또한, 경쟁 예측 모드는 디폴트 예측 움직임 벡터가 아닌 다른 예측 움직임 벡터를 미리 정해진 소정 방법에 따라 선택한 후 현재 움직임 벡터의 부호화

및/또는 복호화에 이용할 수 있는 모드로 이해될 수 있다. 이 경우, 전술한 바와 같은 복수 개의 후보 움직임 벡터 집합에서 소정 방법에 의해 예측 움직임 벡터가 선택될 수 있다.

[230] 이상에서 전술한 본 발명의 실시예에 따르면, 움직임 벡터 예측 모드를 선택할 수 있도록 함으로써 현재 움직임 벡터와 더욱 근접한 움직임 벡터를 예측 움직임 벡터로 선택할 수 있으므로, 차분 벡터의 부호화에 따른 비트량을 최소화함으로써 부호화 효율을 향상시킬 수 있다.

[231] 또한, 본 발명의 실시예에 따르면, 더욱 정확한 예측 움직임 벡터를 선택하여 부호화 효율을 향상시키면서도, 움직임 벡터 부호화 장치(200)는 선택된 예측 움직임 벡터를 직접 움직임 벡터 복호화 장치(300)로 알리지 않고, 찾을 수 있도록 정보, 예컨대 움직임 벡터 예측 모드를 전송하거나 그를 찾기 위한 함수를 공유함으로써, 예측 움직임 벡터를 알리기 위해 발생하는 추가적인 정보에 대한 부호화 비트량 증가를 줄여줌으로써, 부호화 효율 및 복호화 효율을 더욱 향상시킬 수 있다.

[232] 또한, 본 발명의 실시예에 따르면, 전송 시의 오류 등으로 인해 이전에 복원된 주변 블록의 움직임 벡터에 오류가 발생한 경우에도, 움직임 벡터 복호화 장치(300)가 스스로 오류의 발생을 판단하여 오류를 은닉하거나 오류를 해결하는 후속 조치를 수행할 수 있으므로, 복호화기 장애가 발생하는 것을 방지하여 복호화기의 성능을 향상시킬 수 있다.

[233] 또한, 이러한 본 발명의 실시예를 움직임 벡터 부호화 및 복호화 기술을 영상 서비스 또는 멀티미디어 스트리밍 서비스에 적용하면, 적은 비트량으로 부호화를 가능하게 함으로써, 사용자에게 만족도 높은 서비스를 제공해줄 수 있다. 특히, 유선 환경에 비해 상대적으로 작은 대역폭, 큰 데이터 손실 및 지연 등을 가질 수 있는 무선 이동 환경에서는 더욱 큰 효과를 기대할 수 있다.

[234] 이하에서는 본 발명의 실시예를 도출하는 과정에서의 연구 결과물에 대해 설명한다.

[235] 본 발명의 실시예는 MVOP를 더욱 향상시킨 방법이다. 이 향상된 MVOP 방법은 SKIP 모드를 지원할 수 있으며, 독립적인 파싱(independent parsing)이 가능하다. 베이스라인 프로파일(Baseline profile)을 위한 VCEG 공통실험조건(common test condition) 하에서, 향상된 MVOP 방법은 KTA 2.0 소프트웨어와 비교하여 약 평균 4.46%(1.66% for QCIF, 3.60% for CIF and 6.84% for HD 시퀀스)의 BDBR 성능 향상이 있음을 보여준다.

[236] 이전 MVOP 방법은 더욱 정밀한 예측 움직임 벡터(PMV)를 선택하기 위해서 현재 블록에 대해 사용가능한 모든 후보 PMV를 이용해 후보 집합을 구성한다. 후보 집합은 시/공간적 주변 블록들의 움직임 벡터의 수평 및 수직 성분의 조합으로 구성된다. 부호화기는 수정된 율-왜곡 최적화 방법을 이용하여 후보 집합 중에서 최적의 PMV를 선택한다. 복호화기는 템플릿 정합 방법을 이용하여 스스로 최적의 PMV를 예측한다. 템플릿 정합 오류가 가장 적은 최적의 PMV를

선택하기 위해서, 모든 사용가능한 후보 집합이 테스트된다. 최종적으로, 전송되는 차분 움직임 벡터(DMV)는 이게 예측된 PMV와 더해져 움직임 벡터로 계산된다.

[237] 더욱 향상된 이 MVOP 방법에서는 1 비트의 추가적인 정보 (*mvop_flag*)를 사용하여, 복호화기가 예측된 최적의 PMV를 사용하여야 하는지 아닌지를 신호한다. 이전 MVOP 방법은 SKIP 모드를 지원하지 않았다. 예를 들어, 제안된 방법이 0인 DMV를 갖고 다른 모든 SKIP 조건을 만족한다 하더라도, 해당 매크로블록은 SKIP 모드가 아닌 P16x16 모드로 부호화 된다. 더욱 향상된 MVOP 방법에서는 이런 SKIP 모드를 보다 효과적으로 사용할 수 있도록 MVOP 방법을 확장할 수 있다.

[238] 이전의 MVOP 방법에서 복호화기는 어떤 PMV를 사용했는지를 모호성없이 예측할 수 없고 이 경우 어떤 예측값을 사용하여야 하는지를 플래그가 알려줘야 하므로, 1 비트의 추가적인 정보(mvop_flag)가 필요하다. 이러한 이유로 이전의 MVOP 방법에서는 세 가지의 다른 움직임 벡터 부호화 모드(예외 모드, 대체 모드, 경합 모드)가 고려되었으나, 더욱 향상된 MVOP 방법에서는 이러한 움직임 벡터 부호화 모드를 다음과 같이 간략화하였다.

[239] 비경쟁 예측 모드

[240] 비경합 모드는 최적의 PMV 선택이 별다른 차이를 주지 않음을 나타낸다. 여기에는 다음과 같은 두 가지 경우가 존재한다. 첫 번째는 PMV 선택을 위해 오직 하나의 후보 PMV만이 존재하는 경우이다. 이 경우, 어느 예측치를 사용하여야 하는가는 부호화기/복호화기 모두에 자명하다. 이런 경우는 주변 움직임 벡터들이 모두 동일하거나 모두 유효하지 않은 경우에 발생한다. 부호화기(그리고 복호화기)는 사용 가능한 오직 하나의 PMV를 동일하게 사용하게 된다. 두 번째는 예측 가능한 최적의 PMV가 중간값 PMV와 같은 경우이다 (즉, 예측된 최적의 PMV와 중간값 PMV가 동일하다). 그러므로 이 모드에서는 , 부호화기(그리고 복호화기)는 사용가능한 PMV 나 중간값 PMV를 사용한다 . 그러므로, 복호화기는 이러한 두 경우를 부호화기로부터 추가적인 정보없이 인식할 수 있으므로, 플래그를 별도로 사용하지 않는다.

[241] 경쟁 예측 모드

[242] 만약 하나의 블록이 위에서 설명된 비경쟁 예측 모드가 아닌 경우, 복호화기는 경합 모드임을 인지할 수 있다. 이 모드에서 복호화기는 예측된 최적의 PMV를 사용하여야 하는지 아닌지 여부를 지시받아야 한다. 이러한 목적을 위하여, 1 비트 플래그인 *mvop_flag*가 사용된다. 만약 *mvop_flag* 플래그가 1이라면, DMV는 템플릿 정합을 사용해 예측된 최적의 PMV를 이용하여 복호화되고, 플래그가 0이라면, 복호화기는 DMV를 복호화하기 위하여 중간값 PMV를 사용한다.

[243] 독립적 파싱

[244] 추가적인 정보에 대한 비트량을 줄이기 위해 오직 경합 모드에 한하여 *mvop_flag*가 전송된다. 이전에 제안된 우리의 MVOP 방법에서는 *mvop_flag*를

읽을지 않을지 결정할 때, 주변의 복원된 영상 데이터를 사용해야 한다. 따라서, 이것은 독립파싱(independent parsing) 조건을 만족할 수 없다. 뿐만 아니라 참조 영상에 전송 오류가 발생할 경우, 의도되지 않은 플래그 비트를 잘못 읽을 수 있으므로, decoder crash 문제를 야기하는 원인이 된다.

[245] 파싱과정(parsing 프로세스)을 복호화과정(dedoding 프로세스)에서 완전히 분리시켜 잠재적인 복호화기 크래쉬(crash) 문제를 없애는 문법파티셔닝방법이 새롭게 제안되었다. 표 1에 보이는 바와 같이, 슬라이스 내에 포함된 모든 *mvop_flag* 개수를 나타내는 *num_mvop_flag_minus1*와 그 개수만큼의 *슬라이스내의 mvop_flag*를 슬라이스 헤더 정보로 부호화함으로써, 파싱과정(parsing 프로세스)을 복호화과정(decoding 프로세스)에서 완전히 분리시켰다.

[246] 파싱과정은 슬라이스내의 *mvop_flag* 를 읽어 저장한다. 그리고 복호화(decoding) 과정은 경합 모드인지아닌지의 결정에 따라 저장된 *mvop_flag* 를 각각 선택적으로 읽어온다. 따라서 또 다른 syntax 변화는 없다. 이러한 방법을 사용함으로써, 제안된 MVOP 방법은 독립파싱(independent parsing)의 요구사항을 만족할 수 있고, decoder crash 문제도 피할 수 있다. 또한 제안된 방법은 비트스트림이 전송 오류를 갖고 있는지의 여부를 발견할 수 있다는 추가적인 이점을 가지고 있다. 예를 들어, 슬라이스를 모두 복호화한 후에 항상 알 수 있는 모든 경합 모드의 개수가 *num_mvop_flag_minus1+1* 와 같지 않은 경우에는 비트스트림에 오류가 있음을 의미한다. 따라서 이러한 경우에는 그렇게 설계되었다면오류를 최소화 할 수 있는 또 다른 복호화 방법을 사용하도록 할 수 있다.

[247] 표 1

slice_header ( ) {	C	Descriptor
.....		
num_mvop_flag_minus1	2	u(v)
for(i=0; i<=num_mvop_flag_minus1; i++) {		
mvop_flag[i]	2	u(1)
}		
.....		
}		

[248] 실험 결과

[249]

[250] 향상된 MVOP는 KTA 2.0 소프트웨어를 고쳐 구현되었다. 실험은 baseline 프로파일에 대한 공통실험조건(common test condition) 하에서 실행되었으며, 성능은 BDBR과 BDPSNR 측면에서 측정되었다. 보다 실용성있는 비교를 위하여, MVC(움직임벡터 경쟁) 방법의 성능도 보고 되었다. MVC 방법은 두 개의 PMV 후보(공간 및 시간 MV)를 사용하며, 이는 가장 뛰어난 MVC 성능을 보이는 경우이다. 모든 비교는 KTA 2.0 소프트웨어결과를 앵커(anchor)로

하였다.

[251] CS_{SPTP}를 이용한 실험결과

[252] 표 2

Size	Number of FrameSkip	Sequence	MVC in KTA 2.0		Improved MVOP	
			BDPSNR[dB]	BDBR[%]	BDPSNR[dB]	BDBR[%]
QCIF (176x144)	1	Container	-0.009	0.067	0.040	-0.846
	1	Foreman	0.093	-1.695	0.215	-3.848
	1	Silent Voice	-0.011	0.198	0.015	-0.282
CIF (352x288)	1	Paris	0.043	-0.741	0.100	-1.792
	0	Foreman	0.155	-3.786	0.253	-6.100
	0	Mobile	0.161	-3.073	0.180	-3.449
	0	Tempete	0.115	-2.391	0.145	-3.057
720p60 (1280x720)	0	BigShips	0.130	-3.653	0.296	-8.228
	0	City	0.041	-1.152	0.069	-1.872
	0	Crew	0.002	-4.022	0.210	-7.595
	0	Night	0.149	-5.519	0.227	-8.128
	0	ShuttleStart	0.204	-6.126	0.268	-8.365
Average of QCIF			**0.024**	**-0.477**	**0.090**	**-1.658**
Average of CIF			**0.118**	**-2.498**	**0.170**	**-3.600**
Average of 720p60			**0.127**	**-4.094**	**0.214**	**-6.838**
Average of overall			**0.089**	**-2.658**	**0.168**	**-4.464**

[253] 표 3

Size	Number of FrameSkip	Sequence	Improved MVOP vs. KTA 2.0 with MVC	
			BDPSNR[dB]	BDBR[%]
QCIF (176x144)	1	Container	0.049	-0.915
	1	Foreman	0.121	-2.190
	1	Silent Voice	0.026	-0.476
CIF (352x288)	1	Paris	0.057	-1.059
	0	Foreman	0.100	-2.406
	0	Mobile	0.019	-0.350
	0	Tempete	0.031	-0.674
720p60 (1280x720)	0	BigShips	0.167	-4.808
	0	City	0.039	-1.054
	0	Crew	0.102	-3.748
	0	Night	0.076	-2.745
	0	ShuttleStart	0.069	-2.365
Average of QCIF			**0.065**	**-1.194**
Average of CIF			**0.052**	**-1.122**
Average of 720p60			**0.091**	**-2.944**
Average of overall			**0.071**	**-1.899**

[254] 표 2는 CS_{SPTP}를 사용한 제안된 MVOP과 MVC 방법을 KTA 2.0 소프트웨어를 기준으로 비교한 것이다. 표 2에서의 실험결과는 제안된 MVOP 방법이 KTA 2.0 소프트웨어와 비교하여 평균적으로 1.658 % (QCIF) ~ 6.838% (HDTV)의 성능을 가지고 있음을 보여준다. 제안된 MVOP 방법은 저해상도보다는 고해상도에서 보다 나은 성능을 보인다. 개선된 MVOP 방법은 KTA 2.0 소프트웨어에 비해 항상 향상된 성능을 보여주고 있음에 반해, MVC 방법은 QCIF에서 성능이 감소하는 경우도 있다는 점에 주목해야 한다.

[255] 표 3은 BDBR과 BDPSNR을 이용하여 MVC 방법을 앵커로 향상된 MVOP 방법의 성능을 직접 비교한 것을 보여준다. 제안된 MVOP 방법은 MVC 방법과 비교하여 BDBR에서 대략 1.899% 향상된 성능을 보인다. 성능개선은 고해상도에서 더 크다. MVOP 방법은 최적의 PMV를 가르키는 추가적인 정보

시그널링 없이 최적의 PMV를 사용할 수 있기 때문에 상대적인 이점을 가질 수 있다. MVC 방법은 후보 PMV의 수가 증가하는 경우에 최적의 PMV를 가르키기 위해 보다 많은 추가 정보를 필요로 한다. 그러므로, MVOP 방법의 상대적인 부호화 이득은 PMV의 수가 증가할 수록 더욱 커질것으로 예상된다.

[256] 이상에서, 본 발명의 실시예를 구성하는 모든 구성 요소들이 하나로 결합하거나 결합하여 동작하는 것으로 설명되었다고 해서, 본 발명이 반드시 이러한 실시예에 한정되는 것은 아니다. 즉, 본 발명의 목적 범위 안에서라면, 그 모든 구성 요소들이 하나 이상으로 선택적으로 결합하여 동작할 수도 있다. 또한, 그 모든 구성 요소들이 각각 하나의 독립적인 하드웨어로 구현될 수 있지만, 각 구성 요소들의 그 일부 또는 전부가 선택적으로 조합되어 하나 또는 복수 개의 하드웨어에서 조합된 일부 또는 전부의 기능을 수행하는 프로그램 모듈을 갖는 컴퓨터 프로그램으로서 구현될 수도 있다. 그 컴퓨터 프로그램을 구성하는 코드들 및 코드 세그먼트들은 본 발명의 기술 분야의 당업자에 의해 용이하게 추론될 수 있을 것이다. 이러한 컴퓨터 프로그램은 컴퓨터가 읽을 수 있는 저장매체(Computer Readable Media)에 저장되어 컴퓨터에 의하여 읽혀지고 실행됨으로써, 본 발명의 실시예를 구현할 수 있다. 컴퓨터 프로그램의 저장매체로서는 자기 기록매체, 광 기록매체, 캐리어 웨이브 매체 등이 포함될 수 있다.

[257] 또한, 이상에서 기재된 "포함하다", "구성하다" 또는 "가지다" 등의 용어는, 특별히 반대되는 기재가 없는 한, 해당 구성 요소가 내재할 수 있음을 의미하는 것이므로, 다른 구성 요소를 제외하는 것이 아니라 다른 구성 요소를 더 포함할 수 있는 것으로 해석되어야 한다. 기술적이거나 과학적인 용어를 포함한 모든 용어들은, 다르게 정의되지 않는 한, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미가 있다. 사전에 정의된 용어와 같이 일반적으로 사용되는 용어들은 관련 기술의 문맥상의 의미와 일치하는 것으로 해석되어야 하며, 본 발명에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[258] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명에 개시된 실시예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

산업상 이용가능성

[259] 이상에서 설명한 바와 같이 본 발명은 움직임 벡터를 부호화하거나 복호화

기술과 영상 서비스 또는 멀티미디어 스트리밍 서비스 분야에 적용되어, 적은 비트량으로 영상을 부호화할 수 있도록 함으로써, 사용자에게 만족도 높은 서비스를 제공해줄 수 있으며, 특히, 유선 환경에 비해 상대적으로 작은 대역폭, 큰 데이터 손실 및 지연 등을 가질 수 있는 무선 이동 환경에서는 더욱 큰 효과를 기대할 수 있는 매우 유용한 발명이다.

[260] CROSS-REFERENCE TO RELATED APPLICATION

[261] 본 특허출원은 2009년 1월 23일 한국에 출원한 특허출원번호 제 10-2009-0006308 호에 대해 미국 특허법 119(a)조(35 U.S.C § 119(a))에 따라 우선권을 주장하면, 그 모든 내용은 참고문헌으로 본 특허출원에 병합된다. 아울러, 본 특허출원은 미국 이외에 국가에 대해서도 위와 동일한 동일한 이유로 우선권을 주장하면 그 모든 내용은 참고문헌으로 본 특허출원에 병합된다.

청구범위

[청구항 1] 움직임 벡터를 부호화하는 장치에 있어서,
현재 블록의 후보 움직임 벡터 집합에서 최적 예측 움직임 벡터를 선택하는 예측 움직임 벡터 선택기;
움직임 벡터 복호화 장치와 미리 약속된 판단 기준에 근거하여 복수 개의 움직임 벡터 예측 모드 중 하나의 움직임 벡터 예측 모드를 선택하여 현재 블록의 움직임 벡터 예측 모드로서 결정하는 예측 모드 결정기;
상기 결정된 움직임 벡터 예측 모드에 따라 상기 최적 예측 움직임 벡터 또는 디폴트 움직임 예측 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로서 결정하고 상기 현재 블록과 상기 결정된 예측 움직임 벡터 간의 차분 벡터를 부호화하며, 상기 결정된 움직임 벡터 예측 모드에 따라 상기 결정된 움직임 벡터 예측 모드를 선택적으로 부호화하여 출력하는 부호화기
를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 장치.

[청구항 2] 제 1 항에 있어서,
상기 복수 개의 움직임 벡터 예측 모드는 경쟁 예측 모드 및 비경쟁 예측 모드를 포함하며, 상기 예측 모드 결정기는 상기 움직임 벡터 복호화 장치와 미리 약속된 판단 기준에 근거하여 상기 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하여 상기 경쟁 예측 모드 또는 상기 비경쟁 예측 모드를 상기 현재 블록의 움직임 벡터 예측 모드로서 선택하는 것을 특징으로 하는 움직임 벡터 부호화 장치.

[청구항 3] 제 2 항에 있어서, 상기 예측 모드 결정기는,
상기 최적 예측 움직임 벡터가 상기 디폴트 예측 움직임 벡터와 동일한 경우, 상기 비경쟁 예측 모드를 상기 현재 블록의 움직임 벡터 예측 모드로서 결정하는 것을 특징으로 하는 움직임 벡터 부호화 장치.

[청구항 4] 제 3 항에 있어서,
상기 최적 예측 움직임 벡터가 상기 디폴트 예측 움직임 벡터와 동일한 경우는 상기 후보 움직임 벡터 집합 내에 상기 현재 블록의 현재 움직임 벡터에 대하여 예측 움직임 벡터로 사용할 수 있는 후보 움직임 벡터가 없는 경우, 후보 움직임 벡터가 한 개인 경우, 후보 움직임 벡터가 두 개 이상이지만 모두 동일한 경우 및 두 개 이상이고 모두 동일하지는 않은 후보 움직임 벡터 중에서 선택되는 최적 예측 움직임 벡터가 상기 디폴트 예측 움직임 벡터와 동일한 경우 중 하나 이상인 것을 특징으로 하는 움직임

벡터 부호화 장치.

[청구항 5] 제 1 항에 있어서, 상기 부호화기는,
상기 결정된 움직임 벡터 예측 모드가 비경쟁 예측 모드인 경우,
상기 디폴트 움직임 예측 벡터를 상기 예측 움직임 벡터로서
결정하여 상기 차분 벡터를 생성하고 부호화하는 제 1 부호화기;
상기 결정된 움직임 벡터 예측 모드가 경쟁 예측 모드인 경우, 상기
움직임 벡터 복호화 장치가 상기 최적 예측 움직임 벡터를 예측할
수 있는지 여부에 따라 상기 최적 예측 움직임 벡터 또는 상기
디폴트 예측 움직임 벡터를 상기 예측 움직임 벡터로서 결정하여
상기 차분 벡터를 생성하고 부호화하는 제 2 부호화기; 및
상기 결정된 움직임 벡터 예측 모드가 상기 경쟁 예측 모드인
경우에만 상기 결정된 움직임 벡터 예측 모드를 부호화하는 제 3
부호화기
를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 장치.

[청구항 6] 제 5 항에 있어서, 상기 결정된 움직임 벡터 예측 모드는,
한 개의 비트로 구성되며, 비트값에 따라 상기 경쟁 예측 모드 또는
상기 비경쟁 예측 모드를 나타내는 것을 특징으로 하는 움직임
벡터 부호화 장치.

[청구항 7] 제 1 항에 있어서, 상기 부호화된 움직임 벡터 예측 모드는,
상기 비트스트림의 슬라이스 헤더 또는 픽처 헤더에 삽입되는
것을 특징으로 하는 움직임 벡터 부호화 장치.

[청구항 8] 움직임 벡터를 부호화하는 방법에 있어서,
움직임 벡터 복호화 장치와 미리 약속된 판단 기준에 근거하여
복수 개의 움직임 벡터 예측 모드 중 하나의 움직임 벡터 예측
모드를 선택하여 현재 블록의 움직임 벡터 예측 모드로서
결정하는 단계;
상기 선택된 움직임 벡터 예측 모드에 따라 현재 움직임 벡터에
대한 예측 움직임 벡터를 결정하는 단계;
상기 현재 움직임 벡터와 상기 결정된 예측 움직임 벡터 간의 차분
벡터를 부호화하는 단계;
상기 결정된 움직임 벡터 예측 모드에 따라 상기 결정된 움직임
벡터 예측 모드를 선택적으로 부호화하는 단계; 및
상기 부호화된 차분 벡터 및 상기 부호화된 움직임 벡터 예측
모드를 포함하는 비트스트림을 생성하여 출력하는 단계
를 포함하는 것을 특징으로 하는 움직임 벡터 부호화 방법.

[청구항 9] 움직임 벡터를 복호화하는 장치에 있어서,
움직임 벡터 부호화 장치와 미리 약속한 판단 기준에 따라 현재
블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를

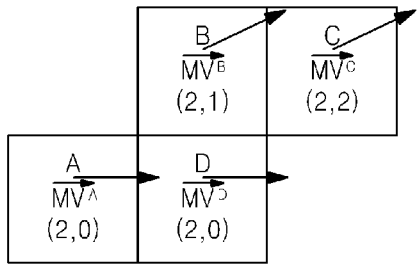
판단하고, 부호화된 차분 벡터를 복호화하여 차분 벡터를 복원하고 출력하는 차분 벡터 복호화기;
부호화된 움직임 벡터 예측 모드를 복호화하여 움직임 벡터 예측 모드를 복원하여 출력하는 예측 모드 복호화기;
상기 현재 블록의 움직임 벡터 예측 모드가 상기 비경쟁 예측 모드인 경우, 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 제 1 예측 움직임 벡터 결정기;
상기 현재 블록의 움직임 벡터 예측 모드가 상기 비경쟁 예측 모드가 아닌 경우, 상기 복원된 움직임 벡터 예측 모드에 따라 최적 예측 움직임 벡터 또는 상기 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 제 2 예측 움직임 벡터 결정기; 및
상기 결정된 예측 움직임 벡터와 상기 복원된 차분 벡터를 가산하여 상기 현재 움직임 벡터를 복원하는 움직임 벡터 복원기
를 포함하는 것을 특징으로 하는 움직임 벡터 복호화 장치.

[청구항 10] 움직임 벡터를 복호화하는 방법에 있어서,
부호화된 움직임 벡터 예측 모드 및 부호화된 차분 벡터를 복호화하여 움직임 벡터 예측 모드 및 차분 벡터를 복원하는 단계;
움직임 벡터 부호화 장치와 미리 약속한 판단 기준에 따라 현재 블록의 움직임 벡터 예측 모드가 비경쟁 예측 모드인지 여부를 판단하는 단계;
상기 판단된 움직임 벡터 예측 모드가 상기 비경쟁 예측 모드인 경우에는 디폴트 예측 움직임 벡터를 현재 블록의 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계;
상기 판단된 움직임 벡터 예측 모드가 상기 비경쟁 예측 모드가 아닌 경쟁 예측 모드인 경우에는 상기 복원된 움직임 벡터 예측 모드에 근거하여 후보 움직임 벡터 집합에서 선택한 최적 예측 움직임 벡터 또는 상기 디폴트 예측 움직임 벡터를 현재 블록의 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정하는 단계; 및
상기 결정된 예측 움직임 벡터와 상기 차분 벡터를 가산하여 상기 현재 움직임 벡터를 복원하는 단계
를 포함하는 것을 특징으로 하는 움직임 벡터 복호화 방법.

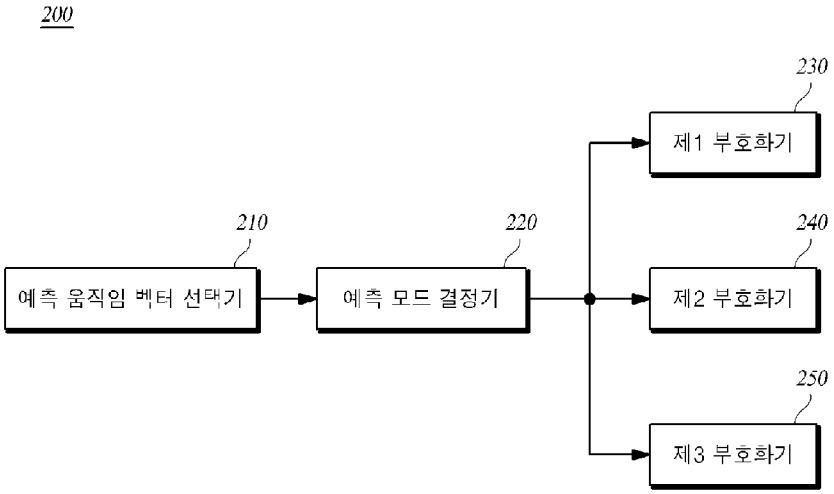
[청구항 11] 제 10 항에 있어서,
상기 복원된 움직임 벡터 예측 모드는 움직임 벡터 예측 모드의 개수를 나타내는 비트 및 상기 개수만큼의 움직임 벡터 예측 모드를 나타내는 비트를 포함하며, 상기 움직임 벡터 복호화 방법은 현재 슬라이스 또는 현재 픽처 내에서 복원된 모든 블록 중 상기 경쟁 예측 모드에 해당하는 블록의 개수와 상기 움직임 벡터

예측 모드의 개수를 비교하여 모든 움직임 벡터가 정상적으로 복호화되었는지 여부를 판단하는 단계를 추가로 포함하는 것을 특징으로 하는 움직임 벡터 복호화 방법.

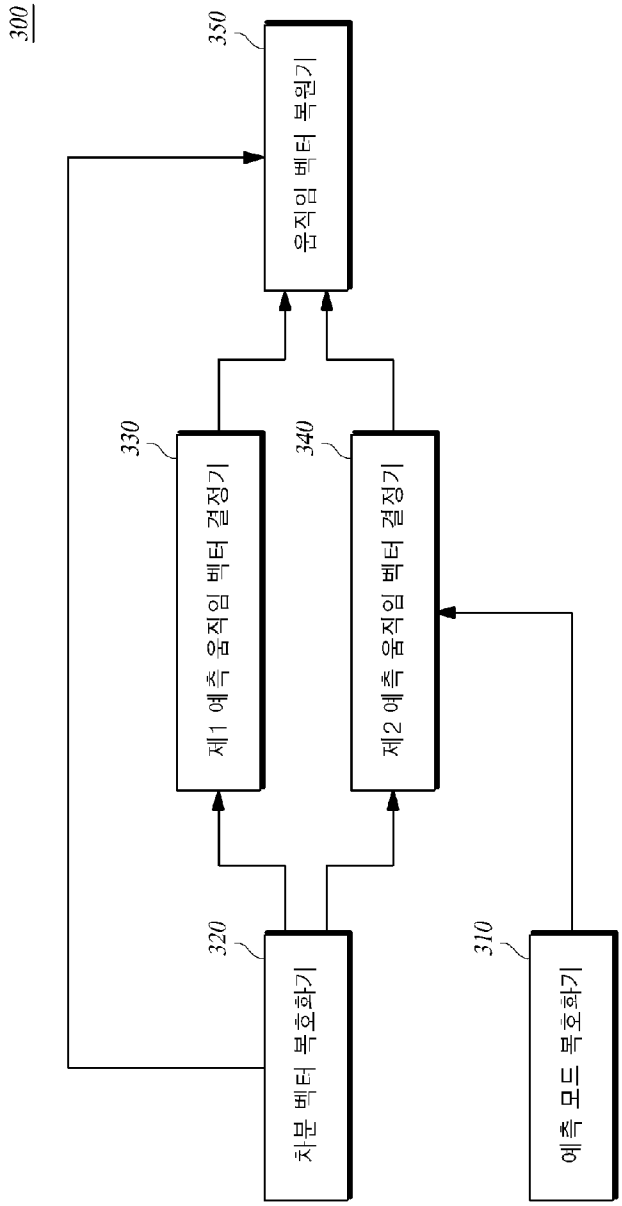
[Fig. 1]


B
MVB
(2,1)
C
MVC
(2,2)
A
MVA
(2,0)
D
MVD
(2,0)

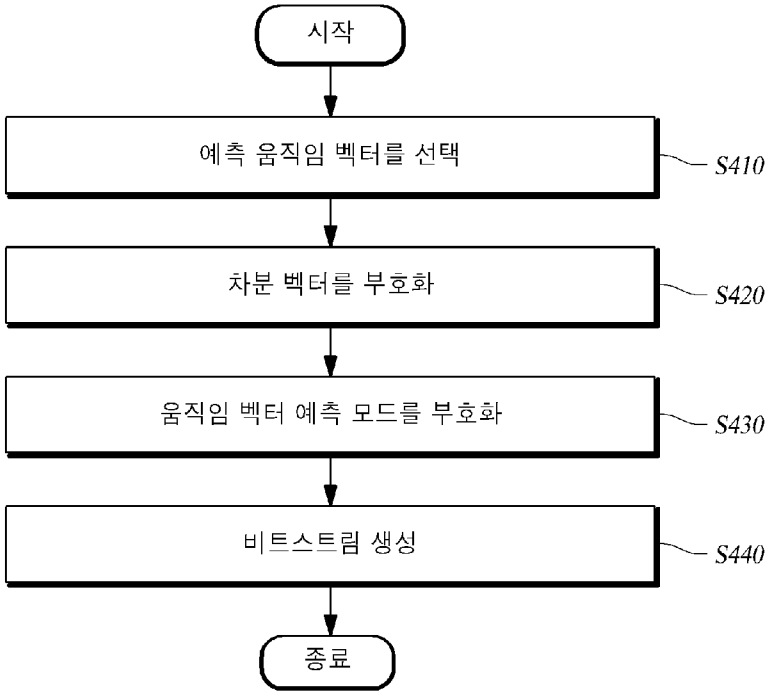
[Fig. 2]


200
230
제1 부호화기
210
220
240
예측 움직임 벡터 선택기
예측 모드 결정기
제2 부호화기
250
제3 부호화기

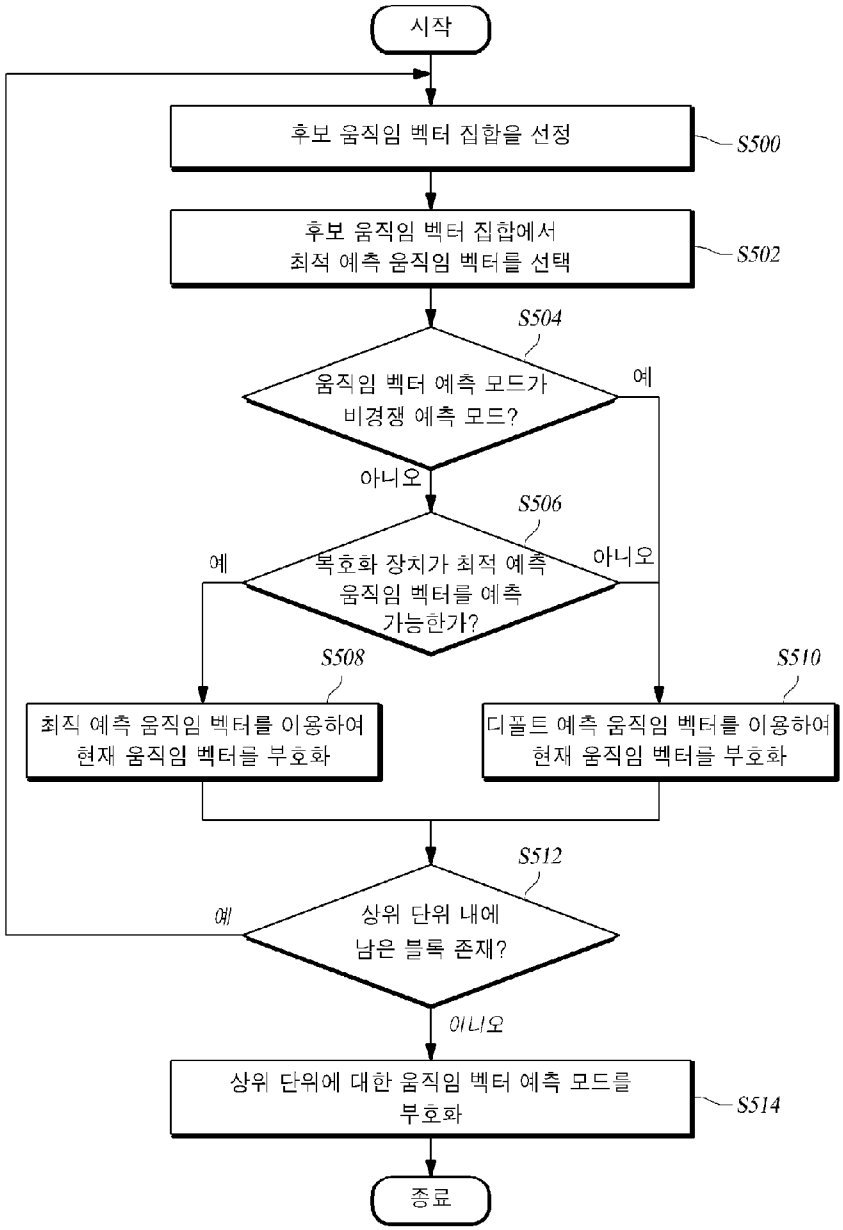
[Fig. 3]


300
350
움직임 벡터 복원기
330
제1 예측 움직임 벡터 결정기
340
제2 예측 움직임 벡터 결정기
320
차분 벡터 복호화기
310
예측 모드 복호화기

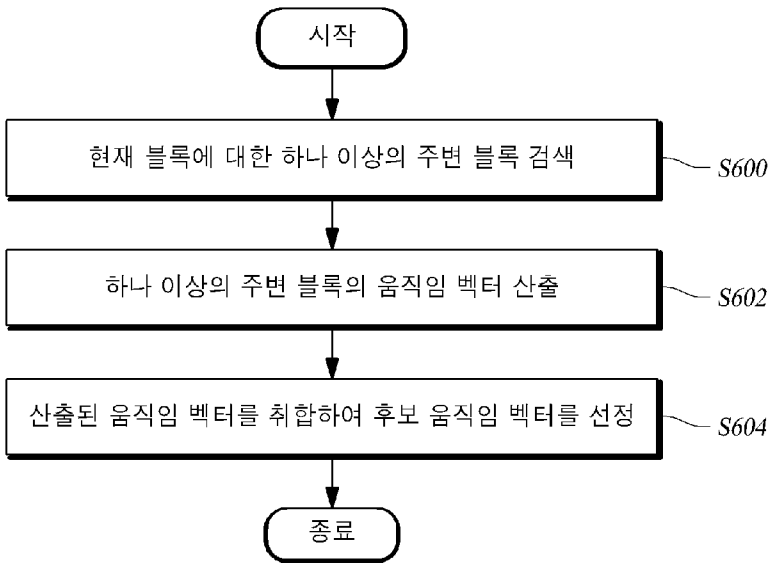
[Fig. 4]


시작
예측 움직임 벡터를 선택
S410
차분 벡터를 부호화
S420
움직임 벡터 예측 모드를 부호화
S430
비트스트림 생성
S440
종료

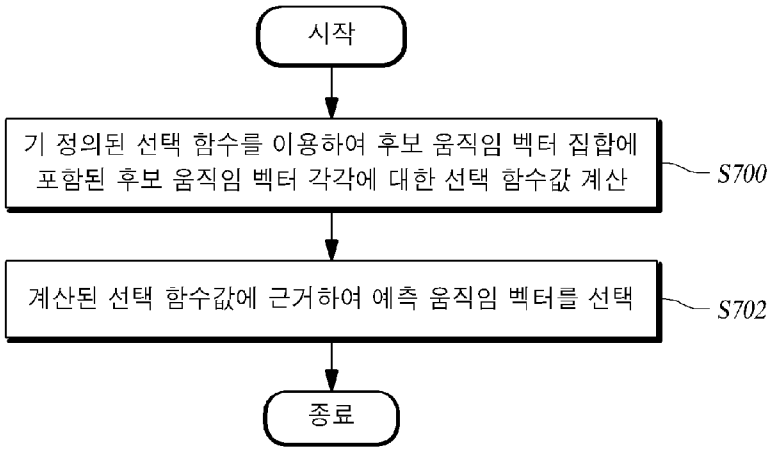
[Fig. 5]


시작
후보 움직임 벡터 집합을 선정
S500
후보 움직임 벡터 집합에서
최적 예측 움직임 벡터를 선택
S502
S504
움직임 벡터 예측 모드가
비경쟁 예측 모드?
예
아니오
S506
복호화 장치가 최적 예측
움직임 벡터를 예측
가능한가?
예
아니오
S508
최적 예측 움직임 벡터를 이용하여
현재 움직임 벡터를 부호화
S510
디폴트 예측 움직임 벡터를 이용하여
현재 움직임 벡터를 부호화
S512
상위 단위 내에
남은 블록 존재?
예
아니오
상위 단위에 대한 움직임 벡터 예측 모드를
부호화
S514
종료

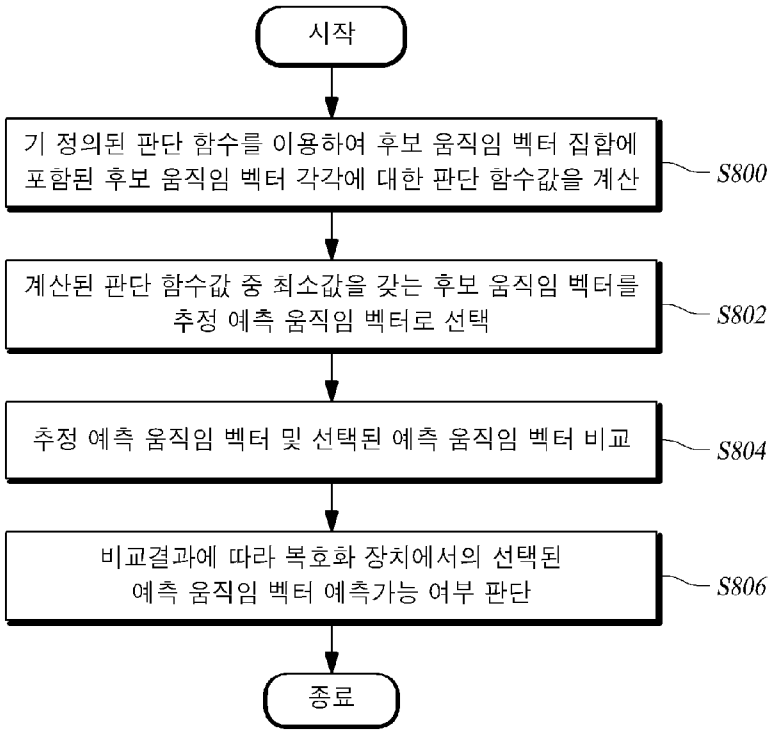
[Fig. 6]


시작
현재 블록에 대한 하나 이상의 주변 블록 검색
S600
하나 이상의 주변 블록의 움직임 벡터 산출
S602
산출된 움직임 벡터를 취합하여 후보 움직임 벡터를 선정
S604
종료

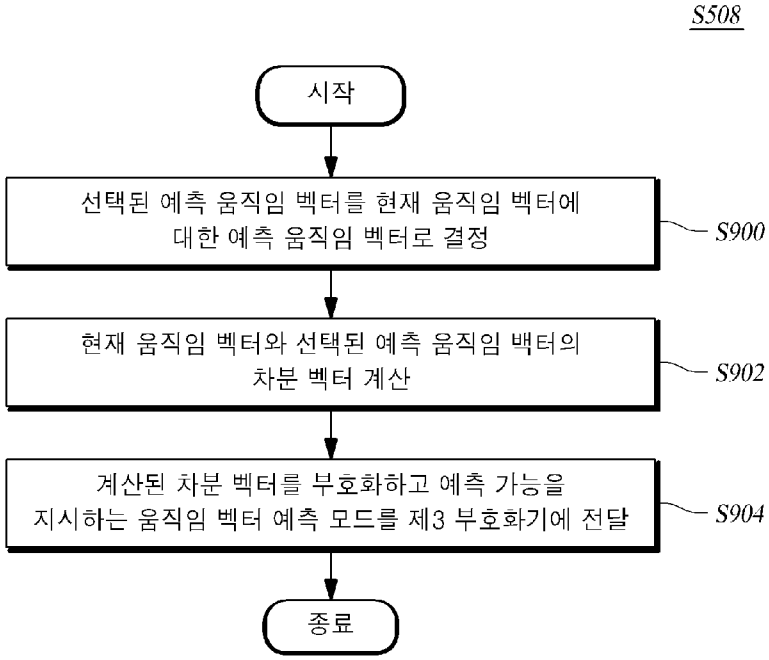
[Fig. 7]


시작
기 정의된 선택 함수를 이용하여 후보 움직임 벡터 집합에 포함된 후보 움직임 벡터 각각에 대한 선택 함수값 계산
S700
계산된 선택 함수값에 근거하여 예측 움직임 벡터를 선택
S702
종료

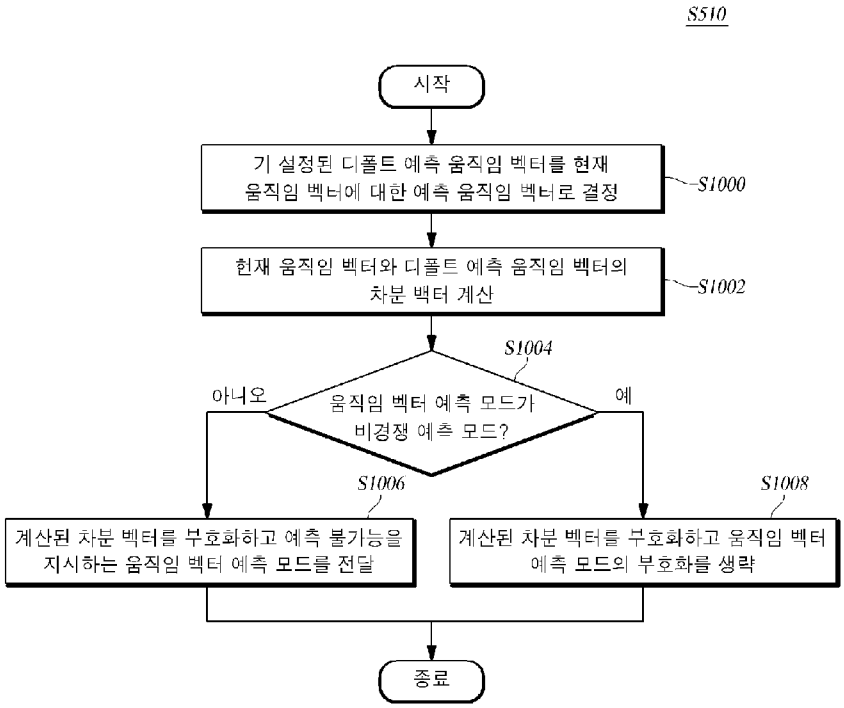
[Fig. 8]


시작
기 정의된 판단 함수를 이용하여 후보 움직임 벡터 집합에 포함된 후보 움직임 벡터 각각에 대한 판단 함수값을 계산
S800
계산된 판단 함수값 중 최소값을 갖는 후보 움직임 벡터를 추정 예측 움직임 벡터로 선택
S802
추정 예측 움직임 벡터 및 선택된 예측 움직임 벡터 비교
S804
비교결과에 따라 복호화 장치에서의 선택된 예측 움직임 벡터 예측가능 여부 판단
S806
종료

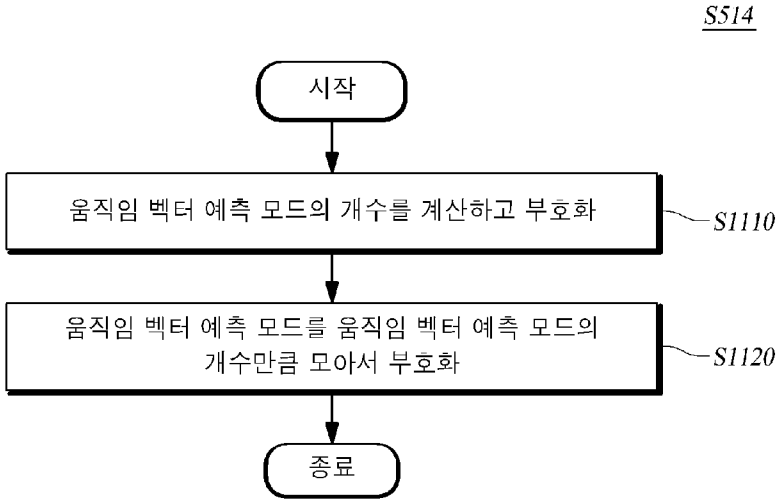
[Fig. 9]


S508
시작
선택된 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정
S900
현재 움직임 벡터와 선택된 예측 움직임 백터의 차분 벡터 계산
S902
계산된 차분 벡터를 부호화하고 예측 가능을 지시하는 움직임 벡터 예측 모드를 제3 부호화기에 전달
S904
종료

[Fig. 10]


S510
시작
기 설정된 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정
S1000
현재 움직임 벡터와 디폴트 예측 움직임 벡터의 차분 벡터 계산
S1002
S1004
움직임 벡터 예측 모드가 비경쟁 예측 모드?
아니오
예
S1006
계산된 차분 벡터를 부호화하고 예측 불가능을 지시하는 움직임 벡터 예측 모드를 전달
S1008
계산된 차분 벡터를 부호화하고 움직임 벡터 예측 모드의 부호화를 생략
종료

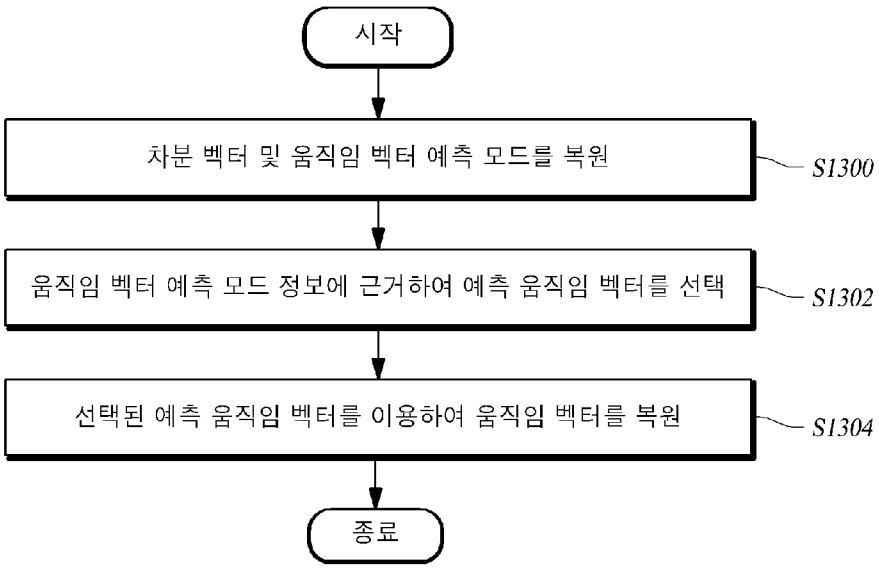
[Fig. 11]



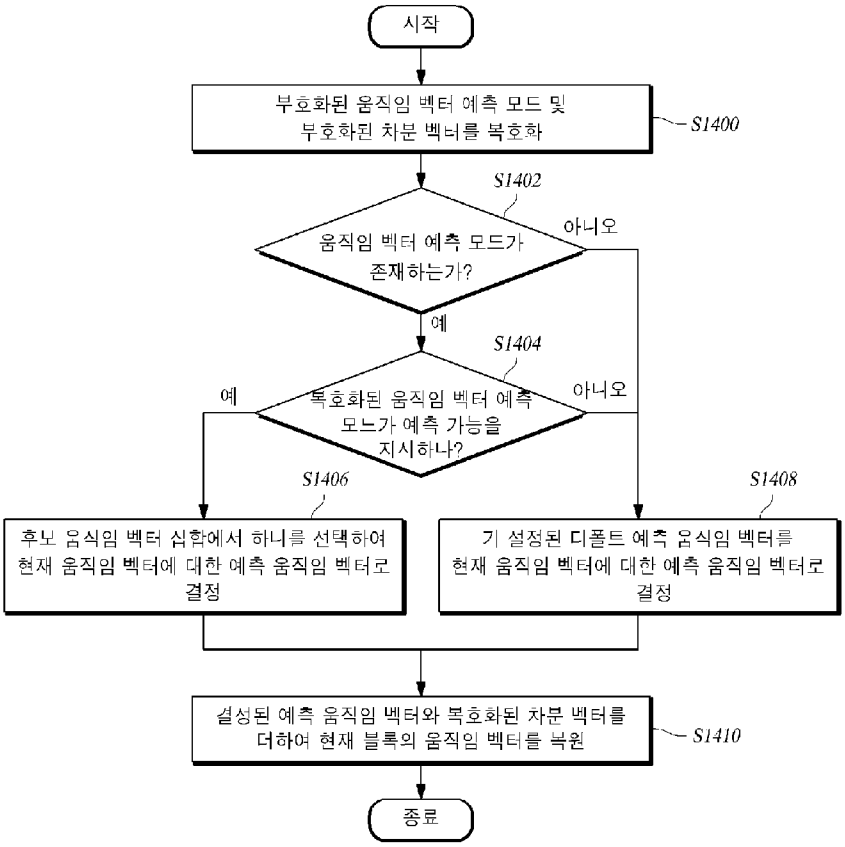
[Fig. 12]

slice_header ( ) {	C	Descriptor
.....		
num_mvop_flag_minus1 for(i=0; i<=num_mvop_flag_minus1; i++) {	2	u(v)
mvop_flag[i]	2	u(1)
}		
.....		
}		

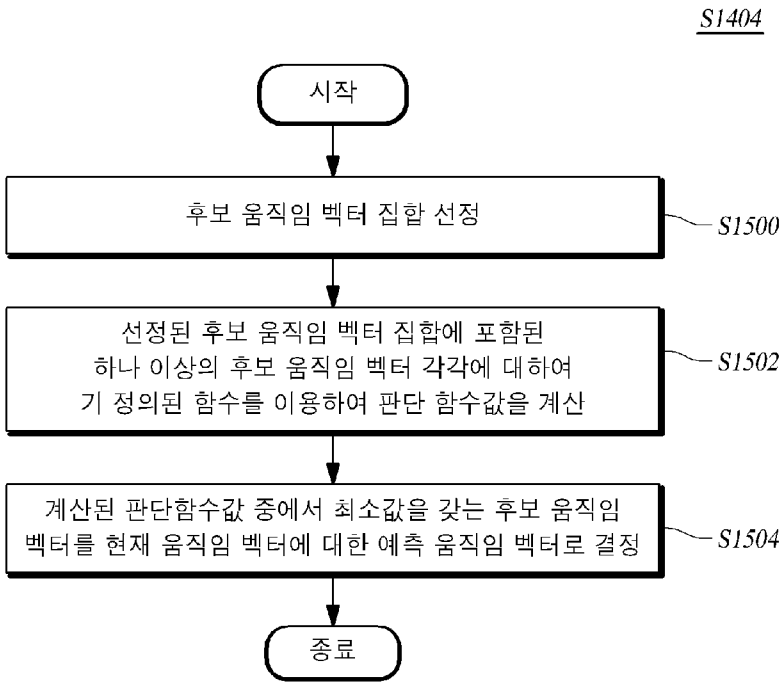
[Fig. 13]



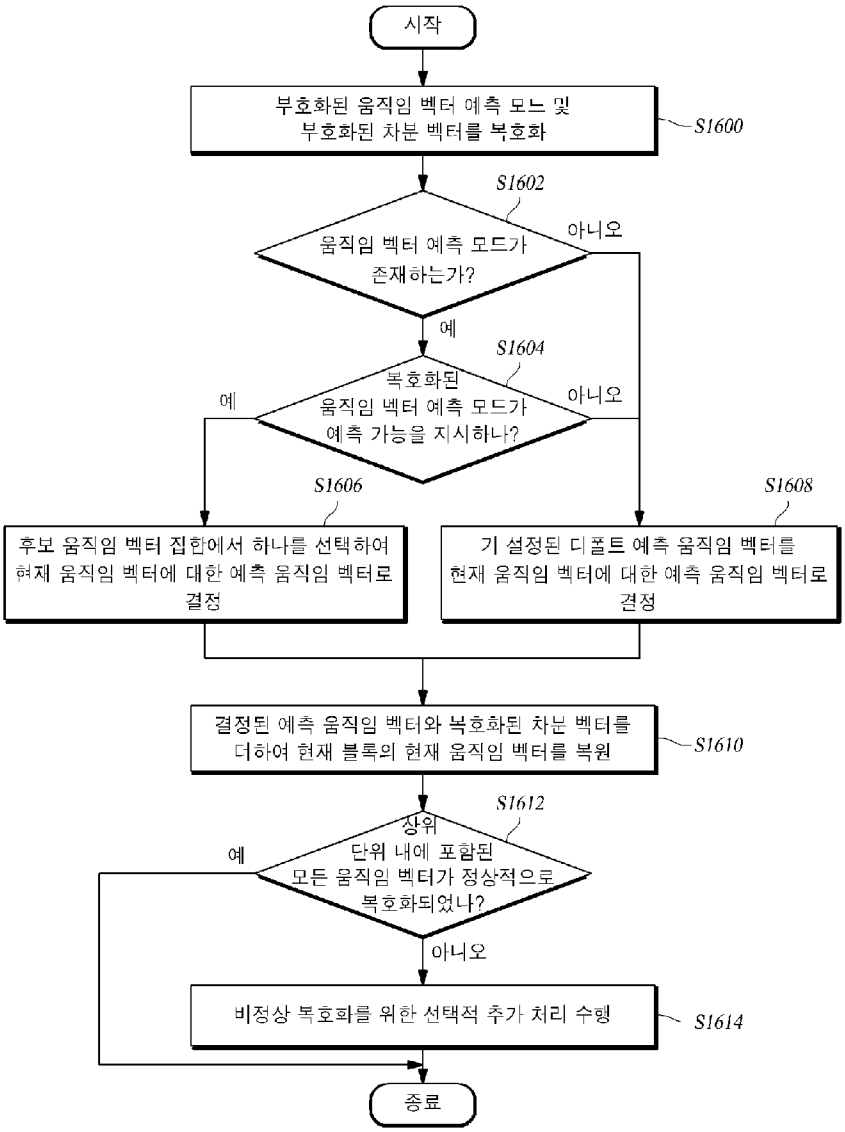
[Fig. 14]


시작
부호화된 움직임 벡터 예측 모드 및 부호화된 차분 벡터를 복호화
S1400
S1402
움직임 벡터 예측 모드가 존재하는가?
아니오
예
S1404
복호화된 움직임 벡터 예측 모드가 예측 가능을 지시하나?
예
아니오
S1406
후보 움직임 벡터 집합에서 하나를 선택하여 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정
S1408
기 설정된 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정
설정된 예측 움직임 벡터와 복호화된 차분 벡터를 더하여 현재 블록의 움직임 벡터를 복원
S1410
종료

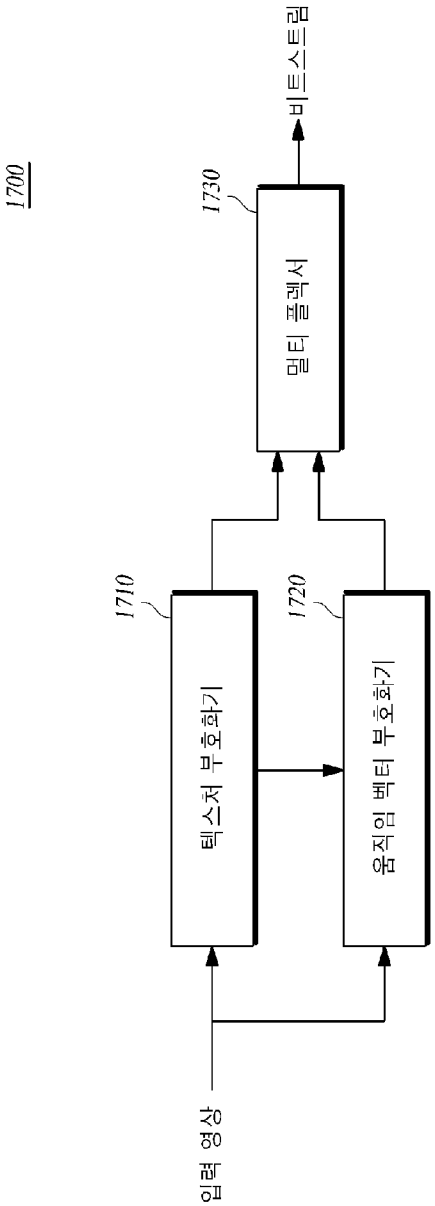
[Fig. 15]


S1404
시작
후보 움직임 벡터 집합 선정
S1500
선정된 후보 움직임 벡터 집합에 포함된 하나 이상의 후보 움직임 벡터 각각에 대하여 기 정의된 함수를 이용하여 판단 함수값을 계산
S1502
계산된 판단함수값 중에서 최소값을 갖는 후보 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정
S1504
종료

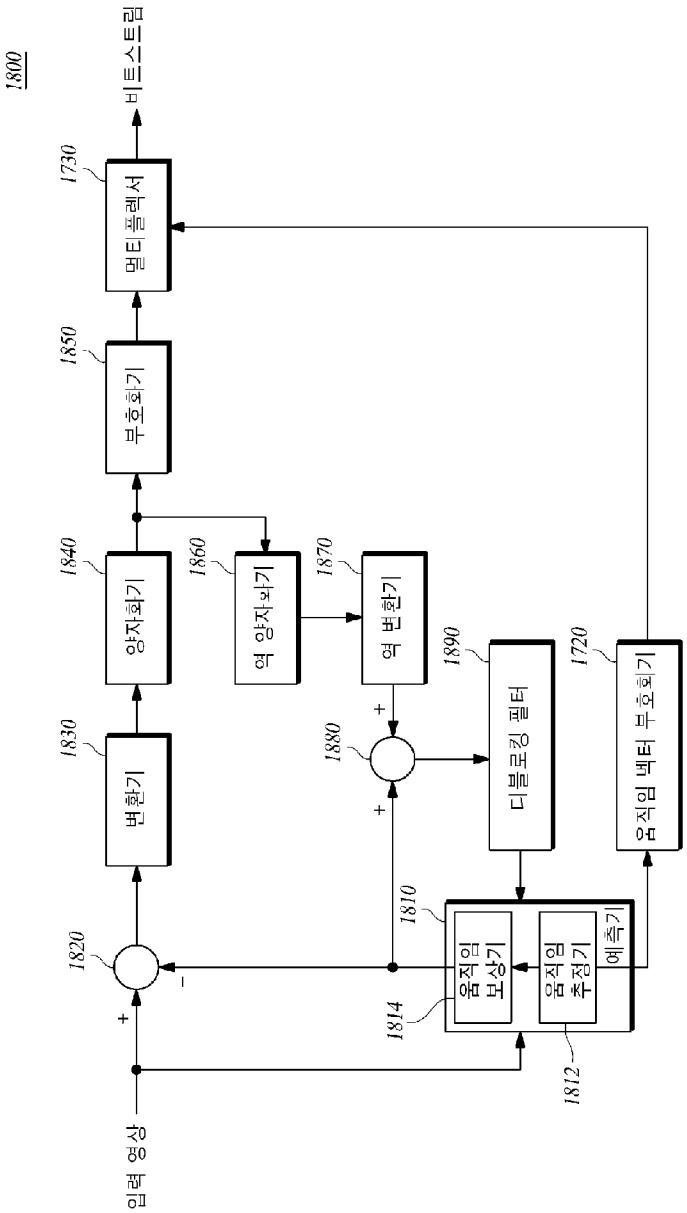
[Fig. 16]


시작
부호화된 움직임 벡터 예측 모드 및 부호화된 차분 벡터를 복호화
S1600
S1602
움직임 벡터 예측 모드가 존재하는가?
아니오
예
S1604
복호화된 움직임 벡터 예측 모드가 예측 가능을 지시하나?
예
아니오
S1606
후보 움직임 벡터 집합에서 하나를 선택하여 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정
S1608
기 설정된 디폴트 예측 움직임 벡터를 현재 움직임 벡터에 대한 예측 움직임 벡터로 결정
결정된 예측 움직임 벡터와 복호화된 차분 벡터를 더하여 현재 블록의 현재 움직임 벡터를 복원
S1610
S1612
상위 단위 내에 포함된 모든 움직임 벡터가 정상적으로 복호화되었나?
예
아니오
비정상 복호화를 위한 선택적 추가 처리 수행
S1614
종료

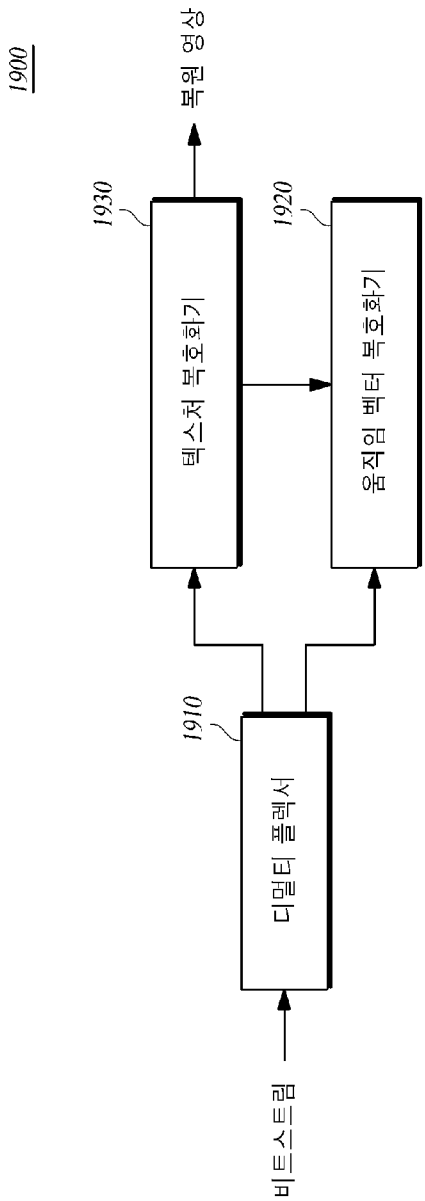
[Fig. 17]



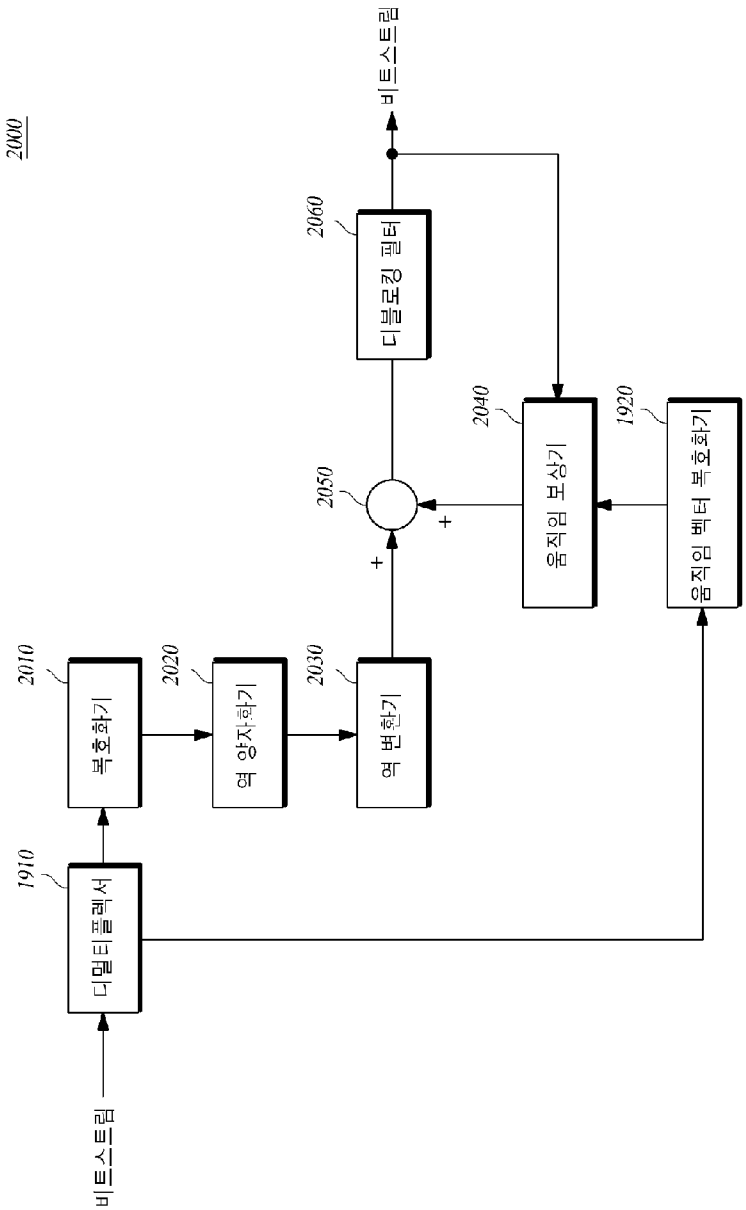
[Fig. 18]


1800
비트스트림
1730
멀티플렉서
1850
부호화기
1840
양자화기
1860
역 양자화기
1870
역 변환기
1890
디블로킹 필터
1720
움직임 벡터 부호화기
1830
변환기
1880
1820
1810
움직임 보상기
움직임 추정기
예측기
1814
1812
입력 영상

[Fig. 19]



[Fig. 20]


2000
비트스트림
2060
디블로킹 필터
2040
움직임 보상기
1920
움직임 벡터 복호화기
2050
+
+
2010
복호화기
2020
역 양자화기
2030
역 변환기
1910
디멀티플렉서
비트스트림