



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0081207  
(43) 공개일자 2020년07월07일

- |  |  |
|--|--|
| <p>(51) 국제특허분류(Int. Cl.)<br/>H04N 19/70 (2014.01) H04N 19/109 (2014.01)<br/>H04N 19/423 (2014.01) H04N 19/51 (2014.01)</p> <p>(52) CPC특허분류<br/>H04N 19/70 (2015.01)<br/>H04N 19/109 (2015.01)</p> <p>(21) 출원번호 10-2019-0115621</p> <p>(22) 출원일자 2019년09월19일<br/>심사청구일자 없음</p> <p>(30) 우선권주장<br/>1020180171277 2018년12월27일 대한민국(KR)</p> | <p>(71) 출원인<br/>에스케이텔레콤 주식회사<br/>서울특별시 중구 을지로 65 (을지로2가)</p> <p>(72) 발명자<br/>나태영<br/>서울특별시 중구 을지로 65 SKT-타워<br/>손세훈<br/>서울특별시 마포구 월드컵로1길 14(합정동 마포한강푸르지오)<br/>(뒷면에 계속)</p> <p>(74) 대리인<br/>이철희</p> |
|--|--|

전체 청구항 수 : 총 12 항

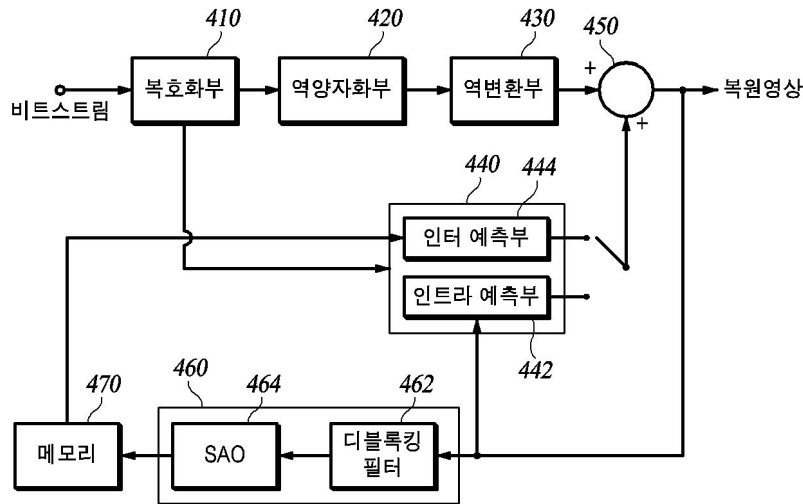
(54) 발명의 명칭 움직임정보 표현 방법 및 이를 이용한 영상 복호화 장치

(57) 요약

움직임정보 표현 방법 및 이를 이용한 영상 복호화 장치를 개시한다.

본 발명의 일 실시예에 의하면, 미리 설정된 크기의 이미지 영역 단위로 저장되는 움직임정보를 표현하는 방법으로서, 상기 이미지 영역에 대응되는 움직임정보를 결정하는 단계; 상기 움직임정보를 하나 이상의 해시함수에 적용하여, 상기 움직임정보에 대한 해시 값을 도출하는 단계; 및 상기 해시 값을 상기 이미지 영역에 대응하여 저장하는 단계를 포함하는 것을 특징으로 하는 움직임정보 표현 방법을 제공한다.

대표도



- (52) CPC특허분류  
*HO4N 19/423* (2015.01)  
*HO4N 19/51* (2015.01)

**김재일**  
서울 중구 을지로 65 SKT타워

- (72) 발명자  
**이선영**  
서울특별시 은평구 진관2로 90(진관동 은평뉴타운  
마고정아파트) 312동 201호

**고경환**  
서울시 강동구 천중로 16길 37 강동리버스타운 10  
3동 402호

---

## 명세서

### 청구범위

#### 청구항 1

미리 설정된 크기의 이미지 영역 단위로 저장되는 움직임정보를 표현하는 방법으로서,  
 상기 이미지 영역에 대응되는 움직임정보를 결정하는 단계;  
 상기 움직임정보를 하나 이상의 해시함수에 적용하여, 상기 움직임정보에 대한 해시 값을 도출하는 단계; 및  
 상기 해시 값을 상기 이미지 영역에 대응하여 저장하는 단계를 포함하는 것을 특징으로 하는 움직임정보 표현 방법.

#### 청구항 2

제1항에 있어서,  
 상기 도출하는 단계는,  
 상기 움직임정보에 포함된 코딩모드 정보, 참조픽처 정보 및 움직임벡터 중 하나 이상을 상기 해시함수에 적용하여, 상기 해시 값을 도출하는 것을 특징으로 하는 움직임정보 표현 방법.

#### 청구항 3

제1항에 있어서,  
 상기 해시함수는,  
 서로 다른 복수 개의 해시함수들을 포함하고,  
 상기 도출하는 단계는,  
 상기 움직임정보에 포함된 코딩모드 정보, 참조픽처 정보 및 움직임벡터로 구성되는 두 개 이상의 그룹을 상기 서로 다른 해시함수에 적용하여 상기 해시 값을 도출하는 것을 특징으로 하는 움직임정보 표현 방법.

#### 청구항 4

제1항에 있어서,  
 상기 도출하는 단계는,  
 상기 움직임정보에 포함된 움직임벡터의 MSB 부분(most significant bits) 및 LSB 부분(least significant bits)을 상기 해시함수에 적용하여 상기 해시 값을 도출하되, 상기 MSB 부분 및 상기 LSB 부분을 구분하여 상기 해시함수에 적용하는 것을 특징으로 하는 움직임정보 표현 방법.

#### 청구항 5

제1항에 있어서,  
 상기 도출하는 단계는,  
 상기 움직임정보에 포함된 참조픽처정보 및 움직임벡터 중 하나 이상의 일부 비트들을 상기 해시함수에 적용하여 상기 해시 값을 도출하는 것을 특징으로 하는 움직임정보 표현 방법.

#### 청구항 6

제5항에 있어서,  
 상기 일부 비트들은,  
 전체 비트들 중 MSB 부분(most significant bits)을 제외한 나머지 비트들인 것을 특징으로 하는 움직임정보 표현 방법.

현 방법.

**청구항 7**

미리 설정된 크기의 이미지 영역 단위로 저장되는 움직임정보를 표현하는 영상 복호화 장치로서,  
 상기 이미지 영역에 대응되는 움직임정보를 결정하고, 상기 움직임정보를 하나 이상의 해시함수에 적용하여 상기 움직임정보에 대한 해시 값을 도출하는 제어수단; 및  
 상기 해시 값을 상기 이미지 영역에 대응하여 저장하는 메모리를 포함하는 것을 특징으로 하는 영상 복호화 장치.

**청구항 8**

제7항에 있어서,  
 상기 제어수단은,  
 상기 움직임정보에 포함된 코딩모드 정보, 참조픽처 정보 및 움직임벡터 중 하나 이상을 상기 해시함수에 적용하여, 상기 해시 값을 도출하는 것을 특징으로 하는 영상 복호화 장치.

**청구항 9**

제7항에 있어서,  
 상기 해시함수는,  
 서로 다른 복수 개의 해시함수들을 포함하고,  
 상기 제어수단은,  
 상기 움직임정보에 포함된 코딩모드 정보, 참조픽처 정보 및 움직임벡터로 구성되는 두 개 이상의 그룹을 상기 서로 다른 해시함수에 적용하여 상기 해시 값을 도출하는 것을 특징으로 하는 영상 복호화 장치.

**청구항 10**

제7항에 있어서,  
 상기 제어수단은,  
 상기 움직임정보에 포함된 움직임벡터의 MSB 부분(most significant bits) 및 LSB 부분(least significant bits)을 상기 해시함수에 적용하여 상기 해시 값을 도출하되, 상기 MSB 부분 및 상기 LSB 부분을 구분하여 상기 해시함수에 적용하는 것을 특징으로 하는 영상 복호화 장치.

**청구항 11**

제7항에 있어서,  
 상기 제어수단은,  
 상기 움직임정보에 포함된 참조픽처정보 및 움직임벡터 중 하나 이상의 일부 비트들을 상기 해시함수에 적용하여 상기 해시 값을 도출하는 것을 특징으로 하는 영상 복호화 장치.

**청구항 12**

제11항에 있어서,  
 상기 일부 비트들은,  
 전체 비트들 중 MSB 부분(most significant bits)을 제외한 나머지 비트들인 것을 특징으로 하는 영상 복호화 장치.

**발명의 설명**

**기술 분야**

[0001] 본 발명은 영상의 부호화 및 복호화에 관한 것으로서, 움직임정보를 효율적으로 표현하여 부호화 및 복호화의 효율을 향상시킨 움직임정보 표현 방법 및 이를 이용한 영상 복호화 장치에 관한 것이다.

**배경 기술**

[0002] 동영상 데이터는 음성 데이터나 정지 영상 데이터 등에 비하여 많은 데이터량을 가지기 때문에, 압축을 위한 처리 없이 그 자체를 저장하거나 전송하기 위해서는 메모리를 포함하여 많은 하드웨어 자원을 필요로 한다.

[0003] 따라서, 통상적으로 동영상 데이터를 저장하거나 전송할 때에는 부호화기를 사용하여 동영상 데이터를 압축하여 저장하거나 전송하며, 복호화기에서는 압축된 동영상 데이터를 수신하여 압축을 해제하고 재생한다. 이러한 동영상 압축 기술로는 H.264/AVC를 비롯하여, H.264/AVC에 비해 약 40% 정도의 부호화 효율을 향상시킨 HEVC(High Efficiency Video Coding)가 존재한다.

[0004] 그러나, 영상의 크기 및 해상도, 프레임율이 점차 증가하고 있고, 이에 따라 부호화해야 하는 데이터량도 증가하고 있으므로 기존의 압축 기술보다 더 부호화 효율이 좋고 화질 개선 효과도 높은 새로운 압축 기술이 요구된다.

**발명의 내용**

**해결하려는 과제**

[0005] 이러한 요구에 부응하기 위해 본 발명은 개선된 영상 부호화 및 복호화 기술을 제공하는 것을 목적으로 하며, 특히, 본 발명의 일 측면은 해시함수를 이용하여 움직임정보를 표현함으로써 부호화 및 복호화의 효율을 향상시키는 기술과 관련된다.

**과제의 해결 수단**

[0006] 본 발명의 일 측면은, 미리 설정된 크기의 이미지 영역 단위로 저장되는 움직임정보를 표현하는 방법으로서, 상기 이미지 영역에 대응되는 움직임정보를 결정하는 단계; 상기 움직임정보를 하나 이상의 해시함수에 적용하여, 상기 움직임정보에 대한 해시 값을 도출하는 단계; 및 상기 해시 값을 상기 이미지 영역에 대응하여 저장하는 단계를 포함하는 것을 특징으로 하는 움직임정보 표현 방법을 제공한다.

[0007] 본 발명의 다른 일 측면은, 미리 설정된 크기의 이미지 영역 단위로 저장되는 움직임정보를 표현하는 영상 복호화 장치로서, 상기 이미지 영역에 대응되는 움직임정보를 결정하고, 상기 움직임정보를 하나 이상의 해시함수에 적용하여 상기 움직임정보에 대한 해시 값을 도출하는 제어수단; 및 상기 해시 값을 상기 이미지 영역에 대응하여 저장하는 메모리를 포함하는 것을 특징으로 하는 영상 복호화 장치를 제공한다.

**발명의 효과**

[0008] 이상에서 설명한 바와 같이 본 발명의 일 실시예에 의하면, 움직임정보의 해시 값을 이용하여 움직임정보를 표현함으로써 움직임정보의 표현에 대한 비트 효율성을 향상시킬 수 있다.

**도면의 간단한 설명**

- [0009] 도 1은 본 개시의 기술들을 구현할 수 있는 영상 부호화 장치에 대한 예시적인 블록도이다.
- 도 2는 QTBITT 구조를 이용하여 블록을 분할하는 방법을 설명하기 위한 도면이다.
- 도 3은 복수의 인트라 예측 모드들을 설명하기 위한 도면이다.
- 도 4는 본 개시의 기술들을 구현할 수 있는 영상 복호화 장치의 예시적인 블록도이다.
- 도 5는 공간적 인접블록과 시간적 인접블록을 설명하기 위한 도면이다.
- 도 6은 MSB 및 LSB를 설명하기 위한 도면이다.
- 도 7 내지 도 9는 움직임정보를 표현하는 본 발명의 일 실시예를 설명하기 위한 도면이다.
- 도 10 및 도 11은 움직임정보를 표현하는 본 발명의 다른 실시예를 설명하기 위한 도면이다.

도 12 및 도 13은 움직임정보를 표현하는 본 발명의 또 다른 실시예를 설명하기 위한 도면이다.

**발명을 실시하기 위한 구체적인 내용**

- [0010] 이하, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다. 각 도면의 구성 요소들에 식별 부호를 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.
- [0012] 도 1은 본 개시의 기술들을 구현할 수 있는 영상 부호화 장치에 대한 예시적인 블록도이다. 이하에서는 도 1을 참조하여 영상 부호화 장치와 이 장치의 하위 구성들에 대하여 설명하도록 한다.
- [0013] 영상 부호화 장치는 블록 분할부(110), 예측부(120), 감산기(130), 변환부(140), 양자화부(145), 부호화부(150), 역양자화부(160), 역변환부(165), 가산기(170), 필터부(180) 및 메모리(190)를 포함하여 구성될 수 있다.
- [0014] 영상 부호화 장치의 각 구성요소는 하드웨어 또는 소프트웨어로 구현되거나, 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다. 또한, 각 구성요소의 기능이 소프트웨어로 구현되고 마이크로프로세서가 각 구성요소에 대응하는 소프트웨어의 기능을 실행하도록 구현될 수도 있다.
- [0015] 하나의 영상(비디오)은 복수의 픽처들로 구성된다. 각 픽처들은 복수의 영역으로 분할되고 각 영역마다 부호화가 수행된다. 예를 들어, 하나의 픽처는 하나 이상의 타일(Tile) 또는/및 슬라이스(Slice)로 분할된다. 여기서, 하나 이상의 타일을 타일 그룹(Tile Group)으로 정의할 수 있다. 각 타일 또는/슬라이스는 하나 이상의 CTU(Coding Tree Unit)로 분할된다. 그리고 각 CTU는 트리 구조에 의해 하나 이상의 CU(Coding Unit)들로 분할된다. 각 CU에 적용되는 정보들은 CU의 선택소로서 부호화되고, 하나의 CTU에 포함된 CU들에 공통적으로 적용되는 정보는 CTU의 선택소로서 부호화된다. 또한, 하나의 타일 내의 모든 블록들에 공통적으로 적용되는 정보는 타일의 선택소로서 부호화되거나 다수 개의 타일을 모아 놓은 타일 그룹의 선택소로서 부호화되며, 하나의 픽처들을 구성하는 모든 블록들에 적용되는 정보는 픽처 파라미터 셋(PPS, Picture Parameter Set) 혹은 픽처 헤더에 부호화된다. 나아가, 복수의 픽처가 공통으로 참조하는 정보들은 시퀀스 파라미터 셋(SPS, Sequence Parameter Set)에 부호화된다. 그리고, 하나 이상의 SPS가 공통으로 참조하는 정보들은 비디오 파라미터 셋(VPS, Video Parameter Set)에 부호화된다.
- [0016] 블록 분할부(110)는 CTU(Coding Tree Unit)의 크기를 결정한다. CTU의 크기에 대한 정보(CTU size)는 SPS 또는 PPS의 선택소로서 부호화되어 영상 복호화 장치로 전달된다.
- [0017] 블록 분할부(110)는 영상을 구성하는 각 픽처(picture)를 미리 결정된 크기를 가지는 복수의 CTU(Coding Tree Unit)들로 분할한 이후에, 트리 구조(tree structure)를 이용하여 CTU를 반복적으로(recursively) 분할한다. 트리 구조에서의 리프 노드(leaf node)가 부호화의 기본 단위인 CU(coding unit)가 된다.
- [0018] 트리 구조로는 상위 노드(혹은 부모 노드)가 동일한 크기의 네 개의 하위 노드(혹은 자식 노드)로 분할되는 쿼드트리(QuadTree, QT), 또는 상위 노드가 두 개의 하위 노드로 분할되는 바이너리트리(BinaryTree, BT), 또는 상위 노드가 1:2:1 비율로 세 개의 하위 노드로 분할되는 터너리트리(TernaryTree, TT), 또는 이러한 QT 구조, BT 구조 및 TT 구조 중 둘 이상을 혼용한 구조일 수 있다. 예컨대, QTBT(QuadTree plus BinaryTree) 구조가 사용될 수 있고, 또는 QTBT(TernaryTree plus BinaryTree) 구조가 사용될 수 있다. 여기서, QTBT를 합쳐서 MTT(Multiple-Type Tree)라 칭할 수 있다.
- [0019] 도 2는 QTBT 분할 트리 구조를 보인다. 도 2에서 보는 바와 같이, CTU는 먼저 QT 구조로 분할될 수 있다. 쿼드트리 분할은 분할 블록(splitting block)의 크기가 QT에서 허용되는 리프 노드의 최소 블록 크기(MinQTSize)에 도달할 때까지 반복될 수 있다. QT 구조의 각 노드가 하위 레이어의 4개의 노드들로 분할되는지 여부를 지시하는 제1 플래그(QT\_split\_flag)는 부호화부(150)에 의해 부호화되어 영상 복호화 장치로 시그널링된다. QT의 리프 노드가 BT에서 허용되는 루트 노드의 최대 블록 크기(MaxBTSize)보다 크지 않은 경우, BT 구조 또는 TT 구조 중 어느 하나 이상으로 더 분할될 수 있다. BT 구조 및/또는 TT 구조에서는 복수의 분할 방향이 존재할 수 있다. 예컨대, 해당 노드의 블록이 가로로 분할되는 방향과 세로로 분할되는 방향 두 가지가 존재할 수 있다. 도 2와 같이, MTT 분할이 시작되면, 노드들이 분할되었는지 여부를 지시하는 제2 플래그(mtt\_split\_flag)와, 분할이 되었다면 추가적으로 분할 방향(vertical 혹은 horizontal)을 나타내는 플래그 및/또는 분할 타입(Binary

혹은 Ternary)을 나타내는 플래그가 부호화부(150)에 의해 부호화되어 영상 복호화 장치로 시그널링된다.

- [0020] 트리 구조의 다른 예로서, QTBT 구조를 사용하여 블록을 분할하는 경우, 먼저 분할 되었음을 나타내는 CU 분할 플래그(split\_cu\_flag) 및 분할 타입이 QT 분할인지를 지시하는 QT 분할 플래그(split\_qt\_flag) 정보가 부호화부(150)에 의해 부호화되어 영상 복호화 장치로 시그널링된다. CU 분할 플래그(split\_cu\_flag) 값이 분할되지 않았음을 지시하지 않는 경우, 해당 노드의 블록이 분할 트리 구조에서의 리프 노드(leaf node)가 되어 부호화의 기본 단위인 CU(coding unit)가 된다. CU 분할 플래그(split\_cu\_flag) 값이 분할되었음을 지시하지 않는 경우, QT 분할 플래그(split\_qt\_flag) 값을 통해 분할 타입이 QT 혹은 MTT인지를 구분한다. 분할 타입이 QT인 경우에는 더 이상의 추가 정보가 없으며, 분할 타입이 MTT인 경우에는 추가적으로 MTT 분할 방향(vertical 혹은 horizontal)을 나타내는 플래그(mtt\_split\_cu\_vertical\_flag) 및/또는 MTT 분할 타입(Binary 혹은 Ternary)을 나타내는 플래그(mtt\_split\_cu\_binary\_flag)가 부호화부(150)에 의해 부호화되어 영상 복호화 장치로 시그널링된다.
- [0021] 트리 구조의 다른 예시로서 QTBT가 사용되는 경우, 해당 노드의 블록을 동일 크기의 두 개 블록으로 가로로 분할하는 타입(즉, symmetric horizontal splitting)과 세로로 분할하는 타입(즉, symmetric vertical splitting) 두 가지가 존재할 수 있다. BT 구조의 각 노드가 하위 레이어의 블록으로 분할되는지 여부를 지시하는 분할 플래그(split\_flag) 및 분할되는 타입을 지시하는 분할 타입 정보가 부호화부(150)에 의해 부호화되어 영상 복호화 장치로 전달된다. 한편, 해당 노드의 블록을 서로 비대칭 형태의 두 개의 블록으로 분할하는 타입이 추가로 더 존재할 수도 있다. 비대칭 형태에는 해당 노드의 블록을 1:3의 크기 비율을 가지는 두 개의 직사각형 블록으로 분할하는 형태가 포함될 수 있고, 혹은 해당 노드의 블록을 대각선 방향으로 분할하는 형태가 포함될 수도 있다.
- [0022] CU는 CTU로부터의 QTBT 또는 QTBT 구조에 따라 다양한 크기를 가질 수 있다. 이하에서는, 부호화 또는 복호화하고자 하는 CU(즉, QTBT의 리프 노드)에 해당하는 블록을 '현재블록'이라 칭한다.
- [0023] 예측부(120)는 현재블록을 예측하여 예측블록을 생성한다. 예측부(120)는 인트라 예측부(122)와 인터 예측부(124)를 포함한다.
- [0024] 일반적으로, 픽처 내 현재블록들은 각각 예측적으로 코딩될 수 있다. 일반적으로 현재블록의 예측은 (현재블록을 포함하는 픽처로부터의 데이터를 사용하는) 인트라 예측 기술 또는 (현재블록을 포함하는 픽처 이전에 코딩된 픽처로부터의 데이터를 사용하는) 인터 예측 기술을 사용하여 수행될 수 있다. 인터 예측은 단방향 예측과 양방향 예측 모두를 포함한다.
- [0025] 인트라 예측부(122)는 현재블록이 포함된 현재 픽처 내에서 현재블록의 주변에 위치한 픽셀(참조 픽셀)들을 이용하여 현재블록 내의 픽셀들을 예측한다. 예측 방향에 따라 복수의 인트라 예측모드가 존재한다. 예컨대, 도 3에서 보는 바와 같이, 복수의 인트라 예측모드는 planar 모드와 DC 모드를 포함하는 비방향성 모드와 65개의 방향성 모드를 포함할 수 있다. 각 예측모드에 따라 사용할 주변 픽셀과 연산식이 다르게 정의된다.
- [0026] 인트라 예측부(122)는 현재블록을 부호화하는데 사용할 인트라 예측 모드를 결정할 수 있다. 일부 예들에서, 인트라 예측부(122)는 여러 인트라 예측 모드들을 사용하여 현재블록을 인코딩하고, 테스트된 모드들로부터 사용할 적절한 인트라 예측 모드를 선택할 수도 있다. 예를 들어, 인트라 예측부(122)는 여러 테스트된 인트라 예측 모드들에 대한 레이트 왜곡(rate-distortion) 분석을 사용하여 레이트 왜곡 값들을 계산하고, 테스트된 모드들 중 최선의 레이트 왜곡 특징들을 갖는 인트라 예측 모드를 선택할 수도 있다.
- [0027] 인트라 예측부(122)는 복수의 인트라 예측 모드 중에서 하나의 인트라 예측 모드를 선택하고, 선택된 인트라 예측 모드에 따라 결정되는 주변 픽셀(참조 픽셀)과 연산식을 사용하여 현재블록을 예측한다. 선택된 인트라 예측 모드에 대한 정보가 부호화부(150)에 의해 부호화되어 영상 복호화 장치로 전달된다.
- [0028] 인터 예측부(124)는 움직임 보상 과정을 통해 현재블록에 대한 예측블록을 생성한다. 현재 픽처보다 먼저 부호화 및 복호화된 참조픽처 내에서 현재블록과 가장 유사한 블록을 탐색하고, 그 탐색된 블록을 이용하여 현재블록에 대한 예측블록을 생성한다. 그리고, 현재 픽처 내의 현재블록과 참조픽처 내의 예측블록 간의 변위(displacement)에 해당하는 움직임벡터(motion vector)를 생성한다. 일반적으로, 움직임 추정은 루마(luma) 성분에 대해 수행되고, 루마 성분에 기초하여 계산된 모션 벡터는 루마 성분 및 크로마 성분 모두에 대해 사용된다. 현재블록을 예측하기 위해 사용된 참조픽처에 대한 정보 및 움직임벡터에 대한 정보를 포함하는 움직임 정보는 부호화부(150)에 의해 부호화되어 영상 복호화 장치로 전달된다.
- [0029] 감산기(130)는 현재블록으로부터 인트라 예측부(122) 또는 인터 예측부(124)에 의해 생성된 예측블록을 감산하



여 잔차블록을 생성한다.

- [0030] 변환부(140)는 공간 영역의 픽셀 값들을 가지는 잔차블록 내의 잔차 신호를 주파수 도메인의 변환 계수로 변환한다. 변환부(140)는 잔차블록의 전체 크기를 변환 단위로 사용하여 잔차블록 내의 잔차 신호들을 변환할 수 있으며, 또는 잔차블록을 변환 영역 및 비변환 영역인 두 개의 서브블록으로 구분하여, 변환 영역 서브블록만 변환 단위로 사용하여 잔차 신호들을 변환할 수 있다. 여기서, 변환 영역 서브블록은 가로축 (혹은 세로축) 기준 1:1의 크기 비율을 가지는 두 개의 직사각형 블록 중 하나일 수 있다. 이런 경우, 서브블록 만을 변환하였음을 지시하는 플래그(cu\_sbt\_flag), 방향성(vertical/horizontal) 정보(cu\_sbt\_horizontal\_flag) 및/또는 위치 정보(cu\_sbt\_pos\_flag)가 부호화부(150)에 의해 부호화되어 영상 복호화 장치로 시그널링된다. 또한, 변환 영역 서브블록의 크기는 가로축 (혹은 세로축) 기준 1:3의 크기 비율을 가질 수 있으며, 이런 경우 해당 분할을 구분하는 플래그(cu\_sbt\_quad\_flag)가 추가적으로 부호화부(150)에 의해 부호화되어 영상 복호화 장치로 시그널링된다.
- [0031] 양자화부(145)는 변환부(140)로부터 출력되는 변환 계수들을 양자화하고, 양자화된 변환 계수들을 부호화부(150)로 출력한다.
- [0032] 부호화부(150)는 양자화된 변환 계수들을 CABAC(Context-based Adaptive Binary Arithmetic Code) 등의 부호화 방식을 사용하여 부호화함으로써 비트스트림을 생성한다. 부호화부(150)는 블록 분할과 관련된 CTU size, CU 분할 플래그, QT 분할 플래그, MTT 분할 방향, MTT 분할 타입 등의 정보를 부호화하여, 영상 복호화 장치가 영상 부호화 장치와 동일하게 블록을 분할할 수 있도록 한다.
- [0033] 또한, 부호화부(150)는 현재블록이 인트라 예측에 의해 부호화되었는지 아니면 인터 예측에 의해 부호화되었는지 여부를 지시하는 예측 타입에 대한 정보를 부호화하고, 예측 타입에 따라 인트라 예측정보(즉, 인트라 예측 모드에 대한 정보) 또는 인터 예측정보(참조픽처 및 움직임벡터에 대한 정보)를 부호화한다.
- [0034] 역양자화부(160)는 양자화부(145)로부터 출력되는 양자화된 변환 계수들을 역양자화하여 변환 계수들을 생성한다. 역변환부(165)는 역양자화부(160)로부터 출력되는 변환 계수들을 주파수 도메인으로부터 공간 도메인으로 변환하여 잔차블록을 복원한다.
- [0035] 가산부(170)는 복원된 잔차블록과 예측부(120)에 의해 생성된 예측블록을 가산하여 현재블록을 복원한다. 복원된 현재블록 내의 픽셀들은 다음 순서의 블록을 인트라 예측할 때 참조 픽셀로서 사용된다.
- [0036] 필터부(180)는 블록 기반의 예측 및 변환/양자화로 인해 발생하는 블로킹 아티팩트(blocking artifacts), 링잉 아티팩트(ringing artifacts), 블러링 아티팩트(blurring artifacts) 등을 줄이기 위해 복원된 픽셀들에 대한 필터링을 수행한다. 필터부(180)는 디블로킹 필터(182)와 SAO(Sample Adaptive Offset) 필터(184)를 포함할 수 있다.
- [0037] 디블로킹 필터(180)는 블록 단위의 부호화/복호화로 인해 발생하는 블로킹 현상(blocking artifact)을 제거하기 위해 복원된 블록 간의 경계를 필터링하고, SAO 필터(184)는 디블로킹 필터링된 영상에 대해 추가적인 필터링을 수행한다. SAO 필터(184)는 손실 부호화(lossy coding)로 인해 발생하는 복원된 픽셀과 원본 픽셀 간의 차이를 보상하기 위해 사용되는 필터이다.
- [0038] 디블로킹 필터(182) 및 SAO 필터(184)를 통해 필터링된 복원블록은 메모리(190)에 저장된다. 한 픽처 내의 모든 블록들이 복원되면, 복원된 픽처는 이후에 부호화하고자 하는 픽처 내의 블록을 인터 예측하기 위한 참조 픽처로 사용된다.
- [0039] 도 1에는 표현하지 않았으나, 영상 부호화 장치는 움직임정보에 대한 헤시 값을 미리 설정된 크기의 이미지 영역 단위로 도출하는 제어수단을 더 포함하여 구성될 수 있다. 이 제어수단은 도 1에 표현된 하위 구성들과 함께 동일한 물리적 구성(프로세서 등)에서 구현되거나, 도 1에 표현된 하위 구성들과는 다른 물리적 구성에서 구현될 수 있다. 제어수단에 대한 상세한 내용은 후술하도록 한다.
- [0041] 도 4는 본 개시의 기술들을 구현할 수 있는 영상 복호화 장치의 예시적인 블록도이다. 이하에서는 도 4를 참조하여 영상 복호화 장치와 이 장치의 하위 구성들에 대하여 설명하도록 한다.
- [0042] 영상 복호화 장치는 복호화부(410), 역양자화부(420), 역변환부(430), 예측부(440), 가산기(450), 필터부(460) 및 메모리(470)를 포함하여 구성될 수 있다.



- [0043] 도 1의 영상 부호화 장치와 마찬가지로, 영상 복호화 장치의 각 구성요소는 하드웨어 또는 소프트웨어로 구현되거나, 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다. 또한, 각 구성요소의 기능이 소프트웨어로 구현되고 마이크로프로세서가 각 구성요소에 대응하는 소프트웨어의 기능을 실행하도록 구현될 수도 있다.
- [0044] 복호화부(410)는 영상 부호화 장치로부터 수신한 비트스트림을 복호화하여 블록 분할과 관련된 정보를 추출함으로써 복호화하고자 하는 현재블록을 결정하고, 현재블록을 복원하기 위해 필요한 예측정보와 잔차신호에 대한 정보 등을 추출한다.
- [0045] 복호화부(410)는 SPS(Sequence Parameter Set) 또는 PPS(Picture Parameter Set)로부터 CTU size에 대한 정보를 추출하여 CTU의 크기를 결정하고, 픽처를 결정된 크기의 CTU로 분할한다. 그리고, CTU를 트리 구조의 최상위 레이어, 즉, 루트 노드로 결정하고, CTU에 대한 분할정보를 추출함으로써 트리 구조를 이용하여 CTU를 분할한다.
- [0046] 예컨대, QTBT 구조를 사용하여 CTU를 분할하는 경우, 먼저 QT의 분할과 관련된 제1 플래그(QT\_split\_flag)를 추출하여 각 노드를 하위 레이어의 네 개의 노드로 분할한다. 그리고, QT의 리프 노드에 해당하는 노드에 대해서는 MTT의 분할과 관련된 제2 플래그(MTT\_split\_flag) 및 분할 방향(vertical / horizontal) 및/또는 분할 타입(binary / ternary) 정보를 추출하여 해당 리프 노드를 MTT 구조로 분할한다. 이를 통해 QT의 리프 노드 이하의 각 노드들을 BT 또는 TT 구조로 반복적으로(recursively) 분할한다.
- [0047] 또 다른 예로서, QTBT 구조를 사용하여 CTU를 분할하는 경우, 먼저 CU의 분할 여부를 지시하는 CU 분할 플래그(split\_cu\_flag)를 추출하고, 해당 블록이 분할된 경우, QT 분할 플래그(split\_qt\_flag)를 추출한다. 분할 타입이 QT가 아니고 MTT인 경우, MTT 분할 방향(vertical 혹은 horizontal)을 나타내는 플래그(mtt\_split\_cu\_vertical\_flag) 및/또는 MTT 분할 타입(Binary 혹은 Ternary)을 나타내는 플래그(mtt\_split\_cu\_binary\_flag)를 추가적으로 추출한다. 분할 과정에서 각 노드는 0번 이상의 반복적인 QT 분할 후에 0번 이상의 반복적인 MTT 분할이 발생할 수 있다. 예컨대, CTU는 바로 MTT 분할이 발생하거나, 반대로 다수 번의 QT 분할만 발생할 수도 있다.
- [0048] 다른 예로서, QTBT 구조를 사용하여 CTU를 분할하는 경우, QT의 분할과 관련된 제1 플래그(QT\_split\_flag)를 추출하여 각 노드를 하위 레이어의 네 개의 노드로 분할한다. 그리고, QT의 리프 노드에 해당하는 노드에 대해서는 BT로 더 분할되는지 여부를 지시하는 분할 플래그(split\_flag) 및 분할 방향 정보를 추출한다.
- [0049] 한편, 복호화부(410)는 트리 구조의 분할을 통해 복호화하고자 하는 현재블록을 결정하게 되면, 현재블록이 인트라 예측되었는지 아니면 인터 예측되었는지를 지시하는 예측 타입에 대한 정보를 추출한다. 예측 타입 정보가 인트라 예측을 지시하는 경우, 복호화부(410)는 현재블록의 인트라 예측정보(인트라 예측 모드)에 대한 선택스 요소를 추출한다. 예측 타입 정보가 인터 예측을 지시하는 경우, 복호화부(410)는 인터 예측정보에 대한 선택스 요소, 즉, 움직임벡터 및 그 움직임벡터가 참조하는 참조픽처를 나타내는 정보를 추출한다.
- [0050] 한편, 복호화부(410)는 잔차신호에 대한 정보로서 현재블록의 양자화된 변환계수들에 대한 정보를 추출한다.
- [0051] 역양자화부(420)는 양자화된 변환계수들을 역양자화하고, 역변환부(430)는 역양자화된 변환계수들을 주파수 도메인으로부터 공간 도메인으로 역변환하여 잔차신호들을 복원함으로써 현재블록에 대한 잔차블록을 생성한다.
- [0052] 또한, 역변환부(430)는 변환블록의 일부 영역(서브블록)만 역변환하는 경우, 변환블록의 서브블록만을 변환하였음을 지시하는 플래그(cu\_sbt\_flag), 서브블록의 방향성(vertical/horizontal) 정보(cu\_sbt\_horizontal\_flag) 및/또는 서브블록의 위치 정보(cu\_sbt\_pos\_flag)를 추출하여, 해당 서브블록의 변환계수들을 주파수 도메인으로부터 공간 도메인으로 역변환함으로써 잔차신호들을 복원하고, 역변환되지 않은 영역에 대해서는 잔차신호로 "0" 값을 채움으로써 현재블록에 대한 최종 잔차블록을 생성한다.
- [0053] 예측부(440)는 인트라 예측부(442) 및 인터 예측부(444)를 포함할 수 있다. 인트라 예측부(442)는 현재블록의 예측 타입이 인트라 예측일 때 활성화되고, 인터 예측부(444)는 현재블록의 예측 타입이 인터 예측일 때 활성화된다.
- [0054] 인트라 예측부(442)는 복호화부(410)로부터 추출된 인트라 예측 모드에 대한 선택스 요소로부터 복수의 인트라 예측 모드 중 현재블록의 인트라 예측 모드를 결정하고, 인트라 예측 모드에 따라 현재블록 주변의 참조 픽셀들을 이용하여 현재블록을 예측한다.
- [0055] 인터 예측부(444)는 복호화부(410)로부터 추출된 인트라 예측 모드에 대한 선택스 요소를 이용하여 현재블록의 움직임벡터와 그 움직임벡터가 참조하는 참조픽처를 결정하고, 움직임벡터와 참조픽처를 이용하여 현재블록을

예측한다.

- [0056] 가산기(450)는 역변환부로부터 출력되는 잔차블록과 인터 예측부 또는 인트라 예측부로부터 출력되는 예측블록을 가산하여 현재블록을 복원한다. 복원된 현재블록 내의 픽셀들은 이후에 복호화할 블록을 인트라 예측할 때의 참조픽셀로서 활용된다.
- [0057] 필터부(460)는 디블록킹 필터(462) 및 SAO 필터(464)를 포함할 수 있다. 디블록킹 필터(462)는 블록 단위의 복호화로 인해 발생하는 블록킹 현상(blocking artifact)를 제거하기 위해, 복원된 블록 간의 경계를 디블록킹 필터링한다. SAO 필터(464)는 손실 부호화(lossy coding)으로 인해 발생하는 복원된 픽셀과 원본 픽셀 간의 차이를 보상하기 위해, 디블록킹 필터링 이후의 복원된 블록에 대해 추가적인 필터링을 수행한다. 디블록킹 필터(462) 및 SAO 필터(464)를 통해 필터링된 복원블록은 메모리(470)에 저장된다. 한 픽처 내의 모든 블록들이 복원되면, 복원된 픽처는 이후에 부호화하고자 하는 픽처 내의 블록을 인터 예측하기 위한 참조 픽처로 사용된다.
- [0058] 도 4에는 표현하지 않았으나, 영상 복호화 장치는 움직임정보에 대한 해시 값을 미리 설정된 크기의 이미지 영역 단위로 도출하는 제어수단을 더 포함하여 구성될 수 있다. 이 제어수단은 도 4에 표현된 하위 구성들과 함께 동일한 물리적 구성(프로세서 등)에서 구현되거나, 도 4에 표현된 하위 구성들과는 다른 물리적 구성에서 구현될 수 있다. 제어수단에 대한 상세한 내용은 후술하도록 한다.
- [0059] 한편, 영상 복호화 장치에 포함되는 제어수단은 영상 부호화 장치에 포함되는 전송된 제어수단과 그 기능 측면에서 대응되는 구성에 해당한다. 이하에서는, 영상 복호화 장치에 포함되는 제어수단과 영상 부호화 장치에 포함되는 제어수단을 용이하게 구별하기 위하여, 영상 복호화 장치에 포함되는 제어수단을 복호화 제어수단으로 지칭하며, 영상 부호화 장치에 포함되는 제어수단을 부호화 제어수단으로 지칭하도록 한다.
- [0061] 화면 간 예측 부호화/복호화 방법(인터 예측 방법)은 크게 skip 모드, merge 모드 및 AMVP(adaptive(or advanced) motion vector predictor) 모드와 같은 세 가지 모드로 구분될 수 있다.
- [0062] 현재블록이 skip 모드 또는 merge 모드로 예측되는 경우, 영상 부호화/복호화 장치는 현재블록의 공간적 주변블록(또는 인접블록) 및 시간적 주변블록의 움직임정보(움직임벡터)를 검색하고, 검색된 움직임정보들을 이용하여(머지) 후보 리스트를 구성한다. 현재블록이 AMVP 모드로 예측되는 경우, 영상 부호화/복호화 장치는 현재블록의 공간적 주변블록 및 시간적 주변블록의 움직임벡터를 검색하고, 검색된 움직임벡터들을 이용하여 MVP(MV predictor) 후보 리스트를 구성한다.
- [0063] 인터 예측을 위해 이용되는 공간적 주변블록의 움직임정보(또는 움직임벡터)는 현재블록이 위치한 픽처(현재픽처) 내에서 현재블록과 공간적으로 인접한 블록들의 움직임정보로부터 유도된다. 영상 부호화/복호화 장치는 도 5 (A)에 표현된 바와 같이 현재블록(current block)과 공간적으로 인접한 블록들(A0, A1, B0, B1, B2)을 미리 설정된 순서에 따라 탐색하고, 탐색된 공간적 주변블록들 중 전부 또는 일부의 움직임정보를 포함시켜 후보 리스트를 구성한다.
- [0064] 인터 예측을 위해 이용되는 시간적 주변블록의 움직임정보는 현재블록의 참조픽처(또는 co-located 픽처) 내에 위치하는 주변블록들(또는 co-located 블록)들의 움직임정보로부터 유도된다. 영상 부호화/복호화 장치는 도 5 (B)에 표현된 바와 같이 co-located 블록의 주변블록들(BR, CT)을 미리 설정된 순서에 따라 탐색하고, 탐색된 시간적 주변블록들 중 전부 또는 일부의 움직임정보를 현재블록의 참조픽처에 맞게 MV scaling 한 후, 이를 포함시켜 후보 리스트를 구성한다. 여기서, co-located 블록은 co-located 픽처 내에 자리하되, 현재블록의 현재 픽처 내 위치와 동일한 위치에 자리하는 블록을 의미한다. HEVC 표준에서, 예측 방향 및 co-located 픽처 정보는 slice 단위로 시그널링되는 정보이다.
- [0065] 후보 리스트를 구성하기 위해서는 공간적/시간적 주변블록의 움직임정보가 메모리(190, 470)에 미리 저장되어야 한다. HEVC 표준에서는 16x16의 이미지 영역 단위로 움직임정보가 저장된다. 영상 부호화/복호화 장치는 이미지 영역 단위 내 최좌상단 픽셀을 포함하는 부호화/복호화 대상 블록의 움직임정보를 해당하는 영역의 대표 움직임정보로 하여 메모리(190, 470)에 저장한다. 하나의 영역 단위에 저장되는 움직임정보는 아래 표 1과 같이 총 74bits로 표현될 수 있다.

표 1

움직임정보	비트 수
코딩모드(coding mode)정보	2bits (bi-prediction, uni-prediction(L0), uni-prediction(L1))
참조픽처정보(reference picture index)	4bits x 2 (L0 and L1) = 8bits
움직임벡터(MV)	16bits x 2 (x component and y component) x 2 (L0 and L1) = 64bits
총 비트 수(Total)	2+8+64=74bits

[0066]

[0067]

이와 같이, 종래 방법은 후보 리스트 구성에 이용되는 공간적/시간적 주변블록의 움직임정보 자체를 메모리 (190, 470)에 저장하므로, 비트 효율성 측면에서 비효율적이라 할 수 있다. 또한, 후보 리스트에 움직임정보가 추가될 때마다 움직임정보의 중복 여부를 확인(움직임정보들을 비교)하는 pruning 과정이 수행되는 데, 종래 방법은 움직임정보들 자체를 비교하여 중복 여부를 확인하므로, pruning 과정의 복잡도가 높아질 수 있다.

[0068]

본 발명은 움직임정보를 더욱 작은 비트 수로 표현하여 비트 효율성을 향상시키고 pruning 과정의 복잡도를 감소시킴으로써, 종래 방법의 문제점 또는 한계를 극복할 수 있는 발명에 해당한다.

[0069]

본 발명에서 제안하는 다양한 실시예들을 설명하기에 앞서, 본 명세서에서는 아래 표 2 및 표 3과 같은 두 가지 방법 중 어느 하나의 방법을 통해 움직임정보를 표현하는 것으로 가정한다. 표 2는 두 가지 방법 중 첫 번째 방법(제1방법)을 나타내며, 표 3은 두 번째 방법(제2방법)을 나타낸다.

표 2

움직임정보	비트 수
코딩모드(coding mode)정보	2bits (bi-prediction, uni-prediction(L0), uni-prediction(L1))
참조픽처정보(reference picture index)	4bits x 2(L0 and L1) = 8bits
움직임벡터(MV)	18bits x 2 (x component and y component) x 2 (L0 and L1) = 72bits
총 비트 수(Total)	2+8+72=82bits

[0070]

[0071]

위 표 2에서, 코딩모드정보는 양방향(bi-prediction) 인터 예측, L0 단방향(uni-prediction, L0) 인터 예측, L1 단방향(uni-prediction, L1) 인터 예측 및 인트라 예측 중 현재블록의 예측에 이용되는 어느 하나를 지시할 수 있다.

표 3

움직임정보	비트 수
코딩모드(coding mode)정보	1bits (intra or inter)
예측방향정보(prediction direction)	2bits (bi-prediction, uni-prediction(L0), uni-prediction(L1))
참조픽처정보(reference picture index)	4bits x 2 (L0 and L1) = 8bits
움직임벡터(MV)	18bits x 2 (x component and y component) x 2 (L0 and L1) = 72bits
총 비트 수(Total)	1+2+8+72=83bits

[0072]

[0073]

위 표 3에서, 코딩모드정보는 인트라 예측 및 인터 예측 중 어느 하나를 지시할 수 있으며, 예측방향정보는 양방향 예측, L0 단방향 예측 및 L1 단방향 예측 중 어느 하나를 지시할 수 있다.

[0075]

실시예 1

[0076]

실시예 1은 움직임정보에 포함된 정보들(코딩모드정보, 참조픽처정보, 움직임벡터 및/또는 예측방향정보) 중 하나 이상을 일부 비트들로만 표현하여 비트 효율성을 향상시키는 실시예에 해당한다.

- [0077] 부호화/복호화 제어수단은 움직임정보에 포함된 정보들의 전체 비트들 중 일부를 생략하고 나머지 비트들로 해당 정보를 표현할 수 있다. 여기서, 전체 비트들이란, 해당하는 정보를 표현하도록 미리 설정된 비트들을 의미하거나, 종래 방법에서 해당 정보들을 표현하는데 소요되는 비트들을 의미할 수 있다.
- [0078] 생략되는 비트들은 MSB 부분(most significant bits) 또는 LSB 부분(least significant bits)에 해당할 수 있다. 또한, 생략되는 비트들은 MSB 부분의 일부 또는 LSB 부분의 일부에 해당할 수 있다. 부호화/복호화 제어수단은 전체 비트들 중 MSB 부분에 해당하는 비트들을 생략하고 LSB 부분에 해당하는 비트들(나머지 비트들)로만 정보를 표현하거나, 전체 비트들 중 LSB 부분에 해당하는 비트들을 생략하고 MSB 부분에 해당하는 비트들(나머지 비트들)로만 정보를 표현할 수 있다. 또한, 부호화/복호화 제어수단은 전체 비트의 MSB 부분 중 일부를 생략하고 MSB 부분의 나머지 비트들로만 해당 정보를 표현할 수 있다.
- [0079] MSB 부분 및 LSB 부분의 정의와 관련하여, MSB 및 LSB에 대한 예가 도 6에 표현되어 있다. 도 6에 표현된 바와 같이, 움직임정보에 포함되는 특정 정보가 8bit로 표현됨을 가정하면, 가장 큰 값에 해당하는 비트(최좌측에 위치하는 비트,  $2^7$ )가 MSB가 되며, 가장 작은 값에 해당하는 비트(최우측에 위치하는 비트,  $2^0$ )가 LSB가 된다.
- [0080] 또한, MSB를 포함하면서 MSB에 가깝게 위치하는 비트들이 MSB 부분(most significant bits)에 해당할 수 있으며, LSB를 포함하면서 LSB에 가깝게 위치하는 비트들이 LSB 부분(least significant bits)에 해당할 수 있다. 나아가, MSB 자체가 MSB 부분에 해당할 수 있으며, LSB 자체가 LSB 부분에 해당할 수 있다.
- [0081] MSB 부분을 구성하는 비트들의 개수와 LSB 부분을 구성하는 비트들의 개수는 가변적으로 설정될 수 있다. 예를 들어, MSB 부분은  $2^7$  비트 및  $2^6$  비트로 구성되거나,  $2^7$  비트 내지  $2^5$  비트로 구성될 수 있다. 또한, LSB 부분은  $2^0$  비트 및  $2^1$  비트로 구성되거나,  $2^0$  비트 내지  $2^2$  비트로 구성될 수 있다. 나아가, MSB 부분은  $2^7$  비트만으로 구성될 수 있으며, LSB 부분은  $2^0$  비트만으로 구성될 수 있다.
- [0082] 부호화/복호화 제어수단은 움직임정보에 포함된 정보들 중 하나 이상의 MSB 부분 또는 LSB 부분을 생략하고 나머지 비트들로 해당 정보를 표현할 수 있다. 예를 들어, 움직임벡터의 경우, 부호화/복호화 제어수단은 움직임벡터를 구성하는 MSB 부분 또는 LSB 부분을 생략하고 나머지 비트들로 움직임벡터를 표현할 수 있다. 또 다른 예로, 참조픽처정보의 경우, 부호화/복호화 제어수단은 참조픽처정보를 구성하는 MSB 부분 또는 LSB 부분을 생략하고 나머지 비트들로 참조픽처정보를 표현할 수 있다.
- [0083] 미세한 움직임은 자주 발생하는 반면에 큰 움직임은 자주 발생하지 않는 점을 고려하면, 큰 움직임에 해당하는 MSB 부분을 생략하고 작은 움직임에 해당하는 나머지 비트들로만 움직임벡터를 표현하면 비트 효율성이 향상될 수 있다. 또한, 적은 수(2장 내지 4장)의 참조픽처가 설정되는 것이 일반적인 점을 고려하면, 큰 수의 인덱스 값에 해당하는 MSB 부분을 생략하고 작은 수의 인덱스 값에 해당하는 비트들로만 참조픽처정보를 표현하면 비트 효율성이 향상될 수 있다.
- [0084] 움직임벡터의 MSB 부분에 해당하는 2비트와 참조픽처정보의 MSB 부분에 해당하는 2비트를 생략한 경우에 움직임 정보를 표현하는데 소요되는 비트 수를 표현하면 아래 표 4 및 표 5와 같다. 여기서, 표 4는 전술된 제1방법과 결합된 예이며, 표 5는 전술된 제2방법과 결합된 예이다.

**표 4**

움직임정보	비트 수
코딩모드(coding mode)정보	2bits (bi-prediction, uni-prediction(L0), uni-prediction(L1))
참조픽처정보(reference picture index)	2bits x 2(L0 and L1) = 4bits
움직임벡터(MV)	16bits x 2 (x component and y component) x 2 (L0 and L1) = 64bits
총 비트 수(Total)	2+4+64=70bits

[0085]



표 5

움직임정보	비트 수
코딩모드(coding mode)정보	1bits (intra or inter)
예측방향정보(prediction direction)	2bits (bi-prediction, uni-prediction(L0), uni-prediction(L1))
참조픽처정보(reference picture index)	2bits x 2 (L0 and L1) = 4bits
움직임벡터(MV)	16bits x 2 (x component and y component) x 2 (L0 and L1) = 64bits
총 비트 수(Total)	1+2+4+64=71bits

[0086]

[0087]

표 4에 표현된 바와 같이, 움직임벡터의 MSB 부분에 해당하는 2비트와 참조픽처정보의 MSB 부분에 해당하는 2비트를 생략하면, 82bits로 표현되던 움직임정보가 70bits의 적은 비트 수로 표현될 수 있다. 또한, 표 5에 표현된 바와 같이, 움직임벡터의 MSB 부분에 해당하는 2비트와 참조픽처정보의 MSB 부분에 해당하는 2비트를 생략하면, 83bits로 표현되던 움직임정보가 71bits의 적은 비트 수로 표현될 수 있다.

[0089]

**실시예 2**

[0090]

실시예 2는 움직임정보에 포함된 정보들 중 하나 이상을 해시함수에 적용하고, 그 결과(해시 값)를 움직임정보로 하여 저장하는 방법에 해당한다. 이 실시예에서, 움직임정보는 해시함수의 입력으로 사용되며, 해시함수의 출력인 해시값이 각 이미지 영역 단위로 각 이미지 영역에 대응하여 저장된다. 따라서, 본 명세서에서, 움직임정보를 해시함수에 적용하는 것은 움직임정보를 해시함수의 입력으로 사용하는 것으로 이해될 수 있으며, 해시값을 도출하는 것은 해시 값을 출력하는 것으로 이해될 수 있다.

[0091]

부호화/복호화 제어수단은 MD5(message digest algorithm 5), CRC(cyclic redundancy check), checksum 등과 같은 해시함수들을 이용하여 해시 값을 도출할 수 있다. 입력되는 정보에 비해 상대적으로 적은 비트 수를 가지는 해시 값을 출력할 수 있다면, 위에서 언급된 해시함수들 이외에도 다양한 해시함수들이 이용될 수 있다.

[0092]

**실시예 2-1**

[0093]

부호화/복호화 제어수단은 미리 설정된 크기의 이미지 영역에 대응되는 움직임정보를 결정할 수 있다(S710). 여기서, 미리 설정된 크기의 이미지 영역은 픽처를 가상적으로 분할한 가상의 영역으로서, 종래 방법과 동일하게 16x16 단위이거나 8x8 단위 또는 다른 단위일 수 있다. 이미지 영역에 대응되는 움직임정보는 이미지 영역 단위 내 최좌상단 픽셀을 포함하는 부호화/복호화 대상블록의 움직임정보이거나, 다른 위치의 픽셀을 포함하는 부호화/복호화 대상블록의 움직임정보일 수 있다.

[0094]

부호화/복호화 제어수단은 결정된 움직임정보의 전체 또는 일부를 해시함수에 적용하여 해시 값을 도출할 수 있다(S720). 결정된 움직임정보 전체가 해시함수에 적용되거나, 이 움직임정보에 포함된 정보들 중 하나 이상이 해시함수에 적용되어 해시 값이 도출될 수 있다.

[0095]

해시 값을 도출하는 일 예가 도 8에 표현되어 있다. 앞서 설명된 제1방법을 전제로 할 때, 움직임정보에는 코딩모드정보(coding mode), 참조픽처정보(reference picture index) 및 움직임벡터(MV)가 포함될 수 있다. 코딩모드정보에는 인트라 예측, 양방향 예측(bi-p), 단방향 예측(uni-p(L0)) 및 단방향 예측(uni-p(L1)) 중 어느 하나가 포함될 수 있다. 참조픽처정보와 움직임벡터는 코딩모드정보가 지시하는 예측 방향에 따라, L0 방향에 대한 정보(ref\_idx(L0), MV(L0)) 및/또는 L1 방향에 대한 정보(ref\_idx(L1), MV(L1))로 구성될 수 있다. 이러한 움직임정보에 포함된 정보들 전부 또는 일부가 해시함수(hash function)에 적용되어 움직임정보에 대한 해시 값(hash value)이 도출될 수 있다.

[0096]

움직임정보에 대한 해시 값이 도출되면, 도출된 해시 값은 메모리(190, 470)에 저장될 수 있다(S730). 저장되는 해시 값은 대응되는 이미지 영역의 움직임정보로 취급될 수 있다.

[0097]

한편, 각 이미지 영역에 대응되는 해시 값들 및 움직임정보가 테이블화되어 메모리(190, 470)에 더 저장될 수 있다. 이 테이블(해시 테이블)은 후보 리스트로부터 선택된 후보의 움직임정보를 파악하거나, pruning 과정에서 움직임정보들의 중복 여부를 판단하기 위해 이용될 수 있다. 아래 표 6은 해시 테이블에 대한 일 예를 나타낸다.

표 6

Index (hash value)	Coding mode	Ref_idx (L0)	MV (L0)	Ref_idx (L1)	MV (L1)
C9AEBFF7	Uni (L0)	0	(1, -1)	-	-
DAEB32F5	Uni (L1)	-	-	1	(0, 2)
94EAA7B3	Bi	1	(2, 0)	0	(1, 1)
...	...	...	...	...	...

[0098]

[0099]

표 6에서, 해시 값들은 움직임정보를 CRC-32 해시함수에 적용하여 도출된 32bits의 해시 값들을 16진수로 변환한 일 예이다. uni(L0)의 코딩모드정보와 참조픽처정보(0) 및 움직임벡터(1, -1)를 CRC-32 해시함수에 적용하여 C9AEBFF7의 해시 값이 도출될 수 있다. uni(L1)의 코딩모드정보와 참조픽처정보(1) 및 움직임벡터(0, 2)를 CRC-32 해시함수에 적용하여 DAEB32F5의 해시 값이 도출될 수 있다. Bi의 코딩모드정보와 참조픽처정보(1), 움직임벡터(2, 0), 참조픽처정보(0) 및 움직임벡터(1, 1)를 CRC-32 해시함수에 적용하여 94EAA7B3의 해시 값이 도출될 수 있다.

[0100]

이와 같이, 해시함수를 이용하여 움직임정보(해시 값)을 표현하면, 제1방법(82bits)과 제2방법(83bits)에 비해 상대적으로 적은 비트 수(32bits)만으로 움직임정보를 표현할 수 있다. 움직임정보가 동일한 경우에는 이에 대한 해시 값도 동일하므로, 동일 움직임정보는 해시 테이블에 한 번만 저장된다.

[0101]

이미지 영역 단위로 저장되는 해시 값과 해시 테이블에 대한 일 예가 도 9에 표현되어 있다. 도 9 (A)는 영상 resolution 내 8x8의 이미지 영역 단위로 저장되는 해시 값을 나타내며, 도 9 (B)는 해시 테이블을 나타낸다.

[0102]

도 9에 표현된 바와 같이, 이미지 영역 단위에 대한 해시 값이 도출되면, 도출된 해시 값은 해당 영역 단위 별로 저장되며, 해시 값 각각에 상응하는 움직임정보들이 해시 테이블에 저장되게 된다.

[0103]

구체적으로, 이미지 영역 a에 대한 해시 값(C9AEBFF7)과 이에 상응하는 움직임정보들(Uni(L0), Ref\_idx(L0)=0, MV(L0)=(1, -1))이 해시 테이블에 저장될 수 있으며, 이미지 영역 b에 대한 해시 값(DAEB32F5)과 이에 상응하는 움직임정보들(Uni(L1), Ref\_idx(L1)=1, MV(L1)=(0, 2))이 해시 테이블에 저장될 수 있고, 이미지 영역 c에 대한 해시 값(94EAA7B3)과 이에 상응하는 움직임정보들(Bi, Ref\_idx(L0)=1, MV(L0)=(2, 0), Ref\_idx(L1)=0, MV(L1)=(1, 1))이 해시 테이블에 저장될 수 있다.

[0104]

움직임정보를 해시 테이블에 저장하기 위해 요구되는 비트 수는 움직임정보들의 중복 여부에 따라 그 크기가 달라질 수 있다. 움직임정보들이 2:1 또는 3:1로 중복된다면, 중복되는 움직임정보들을 각 이미지 영역마다 저장하는 종래 방법에 비해 1/2 또는 1/3의 비트 수가 요구될 수 있다.

[0105]

예를 들어, 움직임정보가 3번 중복되는 경우, 종래 방법(제1방법)에서는 움직임정보를 각각 표현하기 위해 총 246bits(82bits x 3)가 요구되나, 본 발명에서는 총 210bits가 요구될 수 있다. 본 발명에서 요구되는 210bits는 세 개의 이미지 영역에 해시 값을 저장하기 위한 96bits(32bits x 3), 해시 테이블에 해시 값을 저장하기 위한 32bits 및 해당 움직임정보를 해시 테이블에 저장하기 위한 82bits를 합한 결과이다.

[0106]

결과적으로, 본 실시예를 이용하면, 종래 방법에 비해 감소된 비트만으로 움직임정보를 표현할 수 있으므로, 비트 효율성을 향상시킬 수 있다. 움직임정보들의 중복 횟수가 증가할수록, 움직임정보를 해시 테이블에 저장하기 위해 요구되는 비트 수가 더욱 감소될 수 있다.

[0107]

B 픽처(bi-predictive picture) 또는 P 픽처(predictive picture)의 경우에, 현재블록의 움직임정보가 주변블록의 움직임정보와 같음을 의미하는 skip 모드 및 merge 모드의 사용 비율이 상대적으로 높다는 점을 감안하면, 본 실시예를 통해 비트 효율성을 향상시킬 수 있음을 알 수 있다.

[0108]

한편, 본 실시예를 통해 움직임정보를 표현하는 경우, pruning 과정은 두 움직임정보에 대한 참조픽처정보 및 움직임벡터를 비교하는 것이 아닌, 두 움직임정보의 해시 값들에 대한 비트 연산으로 대체될 수 있다. 예를 들어, 움직임정보를 더욱 작은 비트 수인 32bits의 해시 값으로 표현하면, pruning 과정은 두 32bits 해시 값들(예를 들어, 표 6의 C9AEBFF7 및 DAEB32F5)에 대한 exclusive or(xor) 비트 연산으로 대체될 수 있다. 따라서, 제1방법(82bits) 또는 제2방법(83bits)을 이용하여 표현된 움직임정보들을 종래 방법에 비해 pruning 과정의 복잡도가 감소될 수 있다.

[0109]

**실시예 2-2**



- [0110] 실시예 2-2는 움직임정보에 서로 다른 복수 개의 해시함수들을 적용하여 해시 값을 도출하는 방법에 해당한다. 즉, 실시예 2-2는 움직임정보에 포함된 정보들을 나누어서 각각 별도의 서로 다른 해시함수들에 적용하는 방법으로 이해될 수 있다.
- [0111] 부호화/복호화 제어수단은 미리 설정된 크기의 이미지 영역에 대응되는 움직임정보를 결정할 수 있다(S1010). 또한, 부호화/복호화 제어수단은 움직임정보를 서로 다른 해시함수들에 적용하여 해시 값을 도출할 수 있는 데(S1020), 서로 다른 해시함수들 각각에 적용되는 움직임정보들은 실시형태에 따라 달라질 수 있다.
- [0112] 예를 들어, 움직임정보에 포함된 코딩모드정보, 참조픽처정보 및 움직임벡터 각각이 서로 다른 해시함수들에 적용되거나, 코딩모드정보 및 참조픽처정보가 동일한 해시함수에 적용되고 움직임벡터가 다른 해시함수에 적용될 수 있다. 다른 예로, 참조픽처정보 및 움직임벡터가 동일한 해시함수에 적용되고 코딩모드정보가 다른 해시함수에 적용되거나, 코딩모드정보 및 움직임벡터가 동일한 해시함수에 적용되고 참조픽처정보가 다른 해시함수에 적용될 수 있다.
- [0113] 여기서, 특정 해시함수에 적용되는 정보들을 하나의 그룹으로 가정하면, 움직임정보에 포함된 정보들은 두 개 이상의 그룹으로 구성될 수 있다. 이와 같은 경우, 실시예 2-2는 움직임정보에 포함된 정보들로 구성되는 두 개 이상의 그룹을 (그룹 별로) 서로 다른 해시함수에 적용하여 해시 값을 도출하는 방법으로 이해될 수 있다.
- [0114] 코딩모드정보 및 참조픽처정보가 동일한 해시함수에 적용되고 움직임벡터가 다른 해시함수에 적용되는 일 예가 도 11에 표현되어 있다.
- [0115] 도 11에 표현된 바와 같이, 코딩모드정보(intra, bi-p, uni-p(L0) 및 uni-p(L1)) 및 참조픽처정보(ref\_idx(L0 & L1))가 해시함수(제1해시함수, Hash Function 1)에 적용되어 제1해시 값(Hash value 1)이 도출될 수 있다. 또한, 움직임벡터(MV(L0 & L1))가 해시함수(제2해시함수, Hash Function 2)에 적용되어 제2해시 값(Hash value 2)이 도출될 수 있다. 여기서, 제1해시함수와 제2해시함수는 서로 다른 해시함수에 해당한다.
- [0116] 움직임정보에 대한 해시 값(제1해시 값 및 제2해시 값)이 도출되면, 도출된 해시 값은 메모리(190, 470)에 저장될 수 있다(S1030). 저장되는 해시 값은 대응되는 이미지 영역의 움직임정보로 취급될 수 있다.
- [0117] 이와 같이, 움직임정보를 서로 다른 해시함수들에 적용하면, 해시함수 각각에 적용되는 움직임정보를 단순화시킬 수 있으므로 각각의 해시함수들로부터 출력되는 해시 값 자체의 비트 수를 감소시킬 수 있다. 또한, 입력되는 움직임정보에 따라 해시함수를 차별화시켜 적용할 수 있다.
- [0118] **실시예 2-3**
- [0119] 실시예 2-3은 움직임벡터 중, MSB 부분의 값은 변화가 드물게 발생하고 LSB 부분의 값은 빈번하게 발생하는 점을 고려하여, 움직임벡터의 MSB 부분 및 LSB 부분을 별도로 해시함수에 적용함으로써 해시 값을 도출하는 방법에 해당한다.
- [0120] 부호화/복호화 제어수단은 미리 설정된 크기의 이미지 영역에 대응되는 움직임정보를 결정할 수 있다(S1210). 또한, 부호화/복호화 제어수단은 움직임정보를 해시함수에 적용하여 해시 값을 도출할 수 있는 데(S1220), 움직임정보에 포함된 움직임벡터는 MSB 부분과 LSB 부분으로 구분되어 동일하거나 서로 다른 해시함수에 적용될 수 있다.
- [0121] MSB 부분과 LSB 부분이 동일한 해시함수에 적용되는 경우에 이 해시함수를 제1해시함수로 지칭하면, 해시함수들에 적용되는 움직임정보가 다음과 같이 구성될 수 있다.
- [0122] 코딩모드정보, 참조픽처정보 및 움직임벡터(MSB 부분 및 LSB 부분) 각각이 서로 다른 해시함수들에 적용될 수 있다. 여기서, 움직임벡터(MSB 부분 및 LSB 부분)가 적용되는 해시함수는 제1해시함수에 해당한다. 코딩모드정보 및 참조픽처정보가 동일한 해시함수에 적용되고, 움직임벡터(MSB 부분 및 LSB 부분)가 다른 해시함수(제1해시함수)에 적용될 수 있다. 다른 예로, 참조픽처정보 및 움직임벡터(MSB 부분 및 LSB 부분)가 동일한 해시함수(제1해시함수)에 적용되고 코딩모드정보가 다른 해시함수에 적용되거나, 코딩모드정보 및 움직임벡터(MSB 부분 및 LSB 부분)가 동일한 해시함수(제1해시함수)에 적용되고 참조픽처정보가 다른 해시함수에 적용될 수 있다.
- [0123] MSB 부분과 LSB 부분이 서로 다른 해시함수에 적용되는 경우에 MSB 부분이 적용되는 해시함수를 제1해시함수로 지칭하고, LSB 부분이 적용되는 해시함수를 제2해시함수로 지칭하면, 해시함수들에 적용되는 움직임정보가 다음과 같이 구성될 수 있다.
- [0124] 코딩모드정보, 참조픽처정보, MSB 부분 및 LSB 부분 각각이 서로 다른 해시함수들에 적용될 수 있다. 여기서,

MSB 부분이 적용되는 해시함수는 제1해시함수에 해당하며, LSB 부분이 적용되는 해시함수는 제2해시함수에 해당한다. 코딩모드정보 및 참조픽처정보가 동일한 해시함수에 적용되고, MSB 부분 및 LSB 부분이 서로 다른 해시함수들(제1해시함수 및 제2해시함수)에 적용될 수 있다. 여기서, 제1해시함수 및 제2해시함수 중 어느 하나는 코딩모드정보 및 참조픽처정보가 적용되는 해시함수와 동일하거나 다를 수 있다. 다른 예로, MSB 부분 및 LSB 부분 중 어느 하나와 참조픽처정보가 동일한 해시함수(제1해시함수 또는 제2해시함수)에 적용되고 코딩모드정보가 다른 해시함수에 적용될 수 있다. 여기서, MSB 부분 및 LSB 부분 중 다른 하나가 적용되는 해시함수는 코딩모드정보가 적용되는 해시함수와 동일하거나 다를 수 있다.

[0125] 움직임정보에 대한 해시 값이 도출되면, 도출된 해시 값은 메모리(190, 470)에 저장될 수 있다(S1230). 저장되는 해시 값은 대응되는 이미지 영역의 움직임정보로 취급될 수 있다. 본 실시예에서는 움직임벡터의 MSB 부분과 LSB 부분을 구분하여 해시함수에 적용하므로, 도출되는 해시 값은 MSB 부분에 대한 해시 값과 LSB 부분에 대한 해시 값을 포함할 수 있다.

[0126] 본 실시예를 통해 해시 값을 도출하는 일 예가 도 13에 표현되어 있다.

[0127] 도 13에 표현된 바와 같이, 코딩모드정보(coding mode)와 참조픽처정보(ref\_idx (L0 & L1))를 동일한 해시함수에 적용하여 m bits의 해시 값을 도출할 수 있다. L0 방향의 움직임벡터 중 MSB 부분(MSBs of MV (L0))을 제1해시함수에 적용하여 k bits의 해시 값을 도출할 수 있으며, L1 방향의 움직임벡터 중 MSB 부분(MSBs of MV (L1))을 제1해시함수에 적용하여 k bits의 해시 값을 도출할 수 있다. L0 방향의 움직임벡터 중 LSB 부분(LSBs of MV (L0))을 제1해시함수 또는 제2해시함수에 적용하여 h bits의 해시 값을 도출할 수 있으며, L1 방향의 움직임벡터 중 LSB 부분(LSBs of MV (L1))을 제1해시함수 또는 제2해시함수에 적용하여 h bits의 해시 값을 도출할 수 있다. 결과적으로, 메모리(190, 470)에는 총  $m + (2 \times k) + (2 \times h)$  bits의 해시 값(움직임정보)이 (고정 길이로) 저장될 수 있다.

[0129] **실시예 3**

[0130] 실시예 3은 실시예 1과 실시예 2를 결합한 방법에 해당한다. 실시예 3에서, 움직임정보에 포함된 정보들 중 하나 이상은 자체의 일부 비트들만으로 해시함수에 적용될 수 있다.

[0131] 예를 들어, 부호화/복호화 제어수단은 참조픽처정보 및/또는 움직임벡터를 구성하는 전체 비트들 중에서 일부 비트를 생략하고, 나머지 비트들만을 해시함수에 적용하여 해시 값을 도출할 수 있다. 생략되는 비트들은 MSB 부분의 전부 또는 일부이거나, LSB 부분의 전부 또는 일부일 수 있다.

[0132] 일부 비트들만으로 표현된 정보들은 단일의 해시함수에 적용되거나, 구분되어 서로 다른 해시함수들에 적용될 수 있다. 또한, 일부 비트들만으로 표현된 움직임벡터는 MSB 부분과 LSB 부분으로 구분되어 동일한 해시함수에 적용되거나 서로 다른 해시함수에 적용될 수 있다.

[0133] 이와 같이, 실시예 3에서는 일부 비트들만으로 움직임정보를 표현하는 실시예 1과 해시 값으로 움직임정보를 표현하는 실시예 2가 모두 구현되므로, 비트 효율성이 더욱 향상될 수 있다.

[0135] 한편, 본 발명은 움직임정보(해시 값) 이외에 움직임벡터의 resolution 정보(MV resolution)를 더 포함하여 메모리(190, 470)에 저장하는 방법을 제안한다. MV resolution은 코딩모드정보, 참조픽처정보 및 움직임벡터를 포함하는 움직임정보에 추가적으로 포함될 수 있으며, 이 경우 움직임정보에 포함되는 정보는 코딩모드정보, 참조픽처정보, MV resolution, 움직임벡터의 순서가 될 수 있다.

[0136] MV resolution은 sub-pixel 또는 integer-pixel 중 어느 하나로 지정될 수 있으며, MV resolution의 지정에 따라 움직임벡터가 표기된다. 이 때, 움직임벡터의 값이 integer-pixel 값을 가지면, 부호화/복호화 제어수단은 MV resolution을 integer-pixel로 지정하고 integer-pixel 움직임벡터로 표현할 수 있다. 이와 달리, 움직임벡터의 값이 sub-pixel 값을 가지면, 부호화/복호화 제어수단은 MV resolution을 sub-pixel로 지정하고 sub-pixel 움직임벡터로 표현할 수 있다.

[0137] MV resolution을 더 표현하는 방법에 대한 일 예가 아래 표 7에 나타나 있다.

표 7

MV value	MV resolution (sub-pixel)	MV (sub-pixel)	MV resolution (integer-pixel)	MV (integer-pixel)
1	1(1/4)	4	0(1)	1
2	1(1/4)	8	0(1)	2
3	1(1/4)	12	0(1)	3
4	1(1/4)	16	0(1)	4

[0138]

[0139]

표 7에 표현된 바와 같이, 움직임벡터의 값이 큰 경우(MSB 부분까지 정보를 표현해야 하는 경우), MV resolution을 integer-pixel로 지정하고 이를 통해 integer-pixel 움직임벡터로 설정함으로써 해당 움직임벡터의 값을 integer로 표현할 수 있게 된다.

[0140]

예를 들어, 실제 움직임벡터의 값(MV value)이 “4” 인 경우, MV resolution이 sub-pixel(1/4)로 지정되면 해당 움직임벡터를 “16” 으로 표현해야 하지만, MV resolution이 integer-pixel(1)로 지정되면 해당 움직임벡터를 “4” 로 표현할 수 있게 된다. 따라서, 움직임벡터를 표현하기 위한 비트 효율성이 향상될 수 있다.

[0142]

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

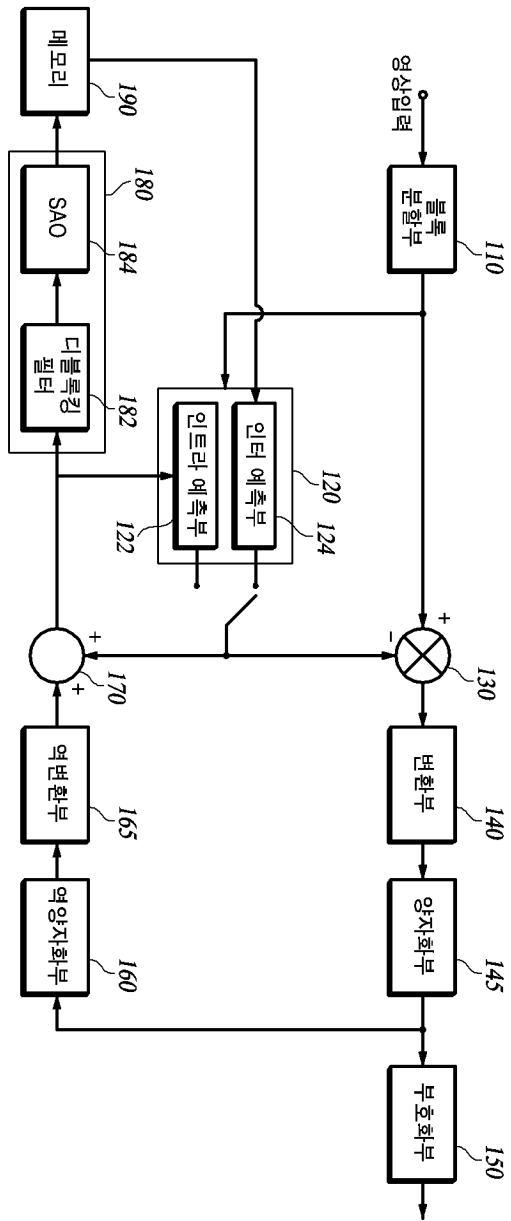
**부호의 설명**

[0143]

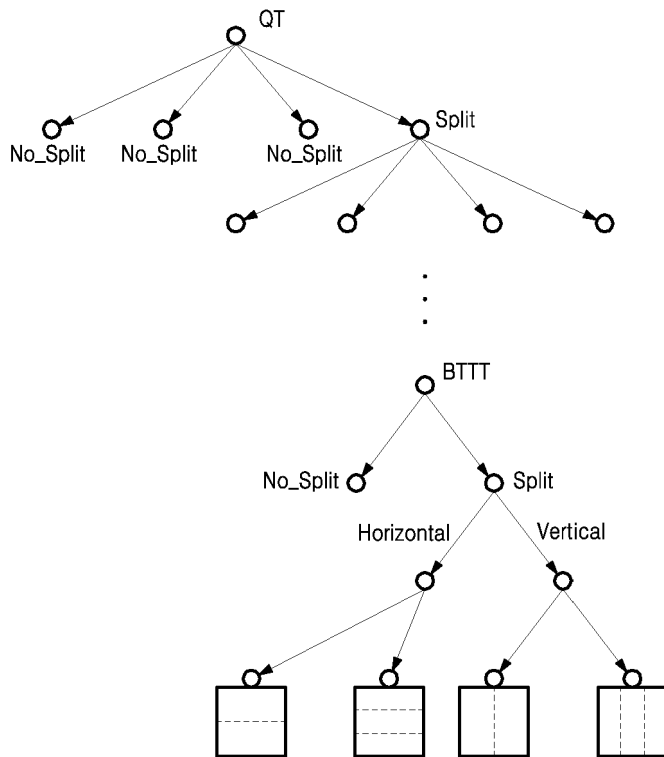
120, 440: 예측부    130: 감산기  
 170, 450: 가산기    180, 450: 필터부

도면

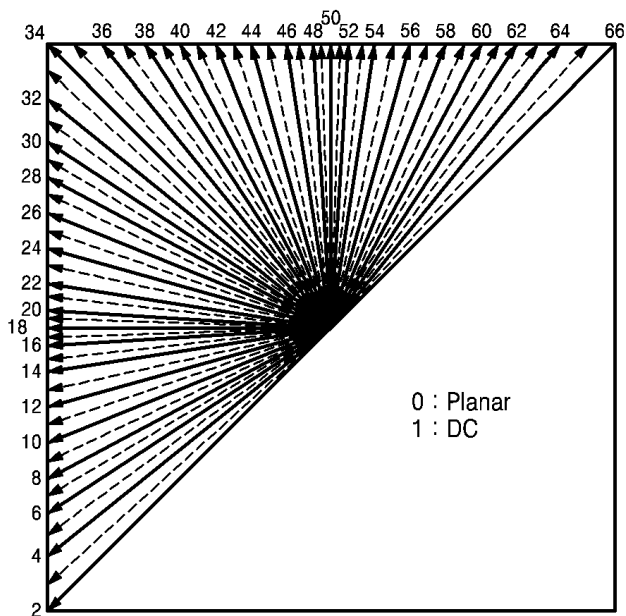
도면1



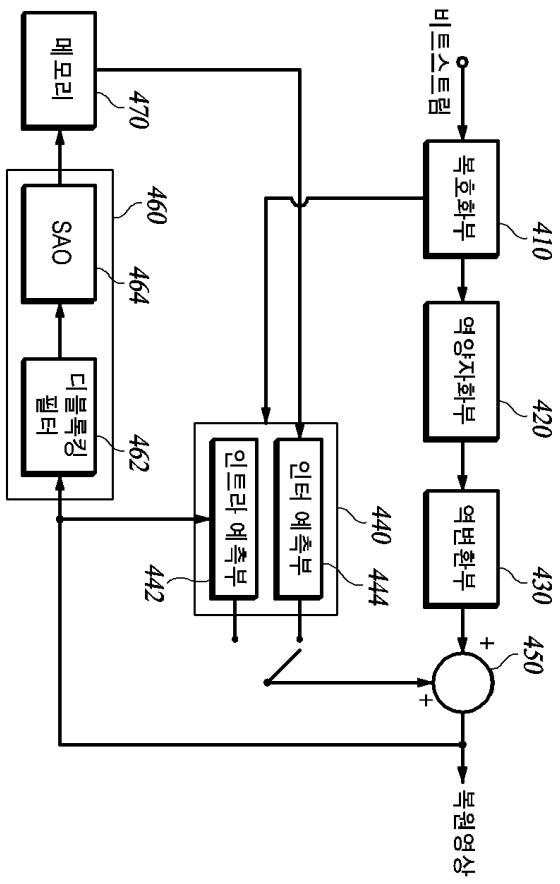
도면2



도면3

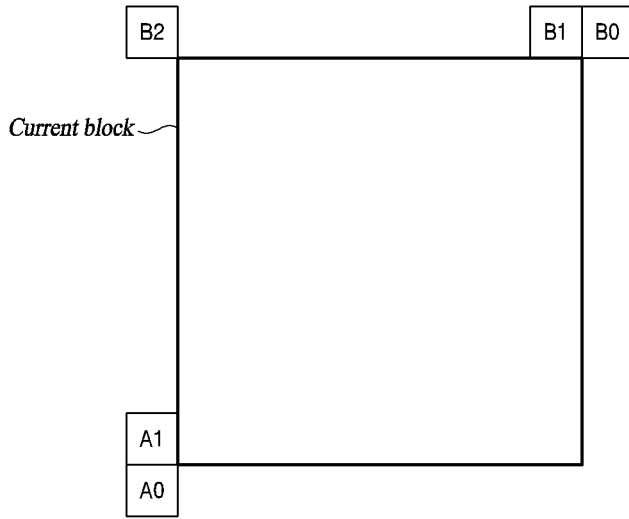


도면4

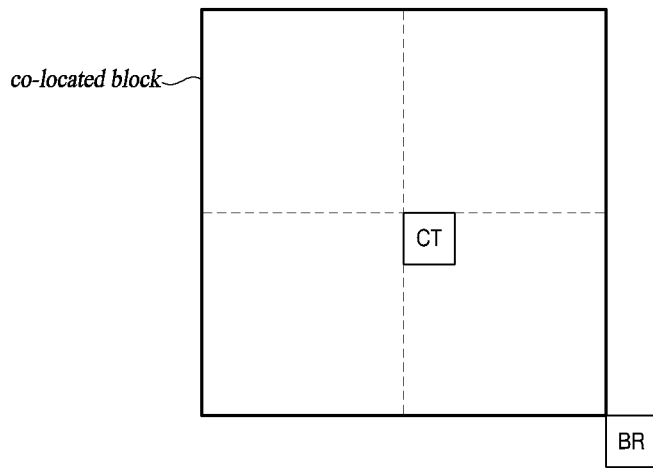




도면5

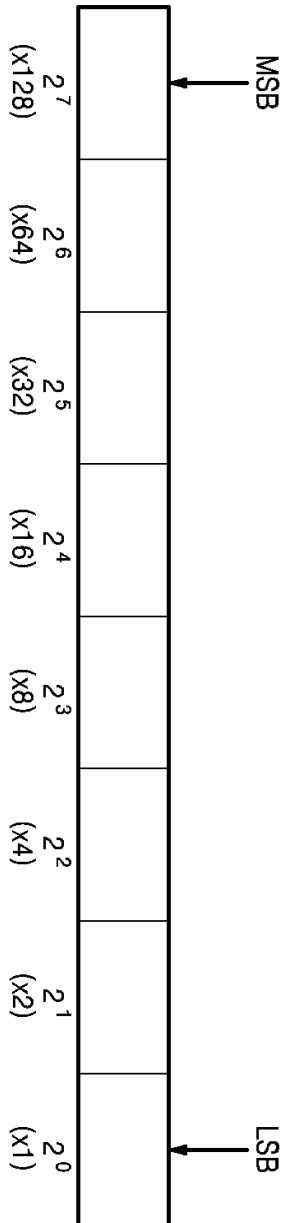


(A)

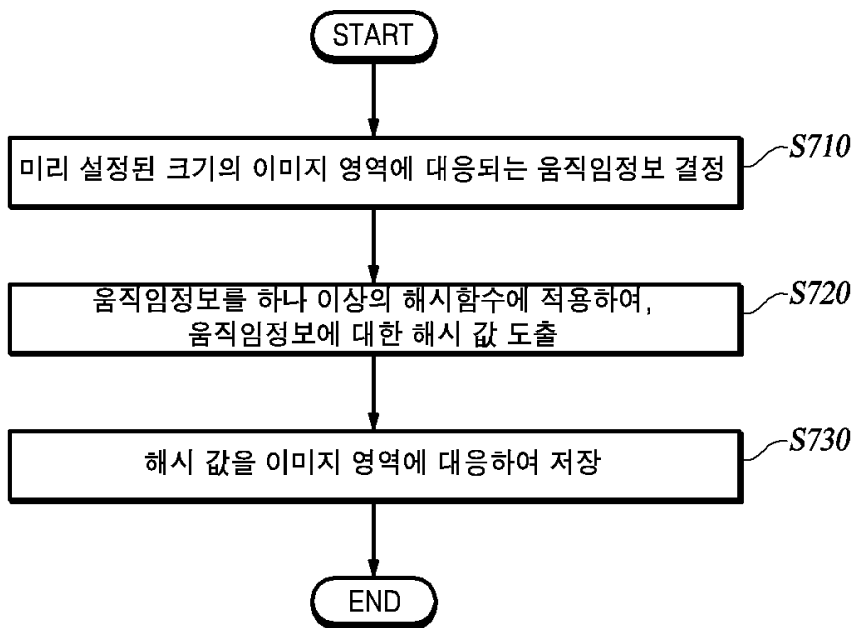


(B)

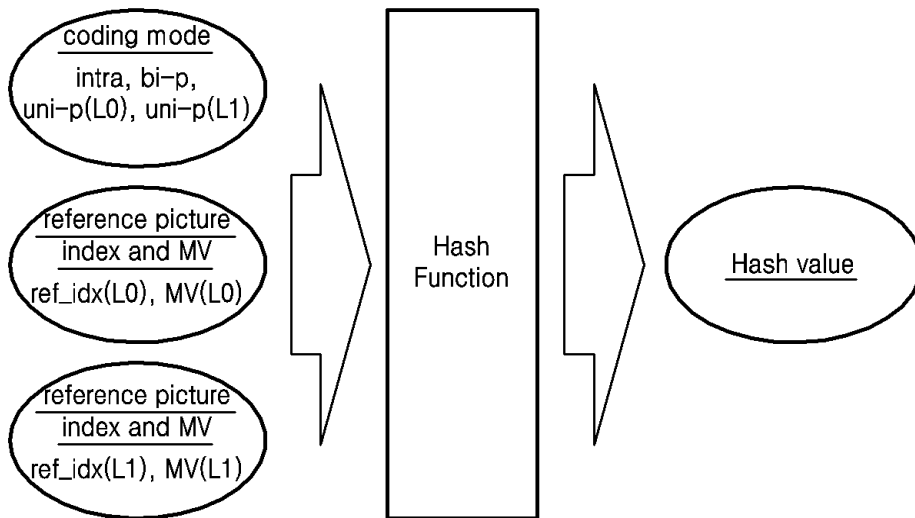
도면6



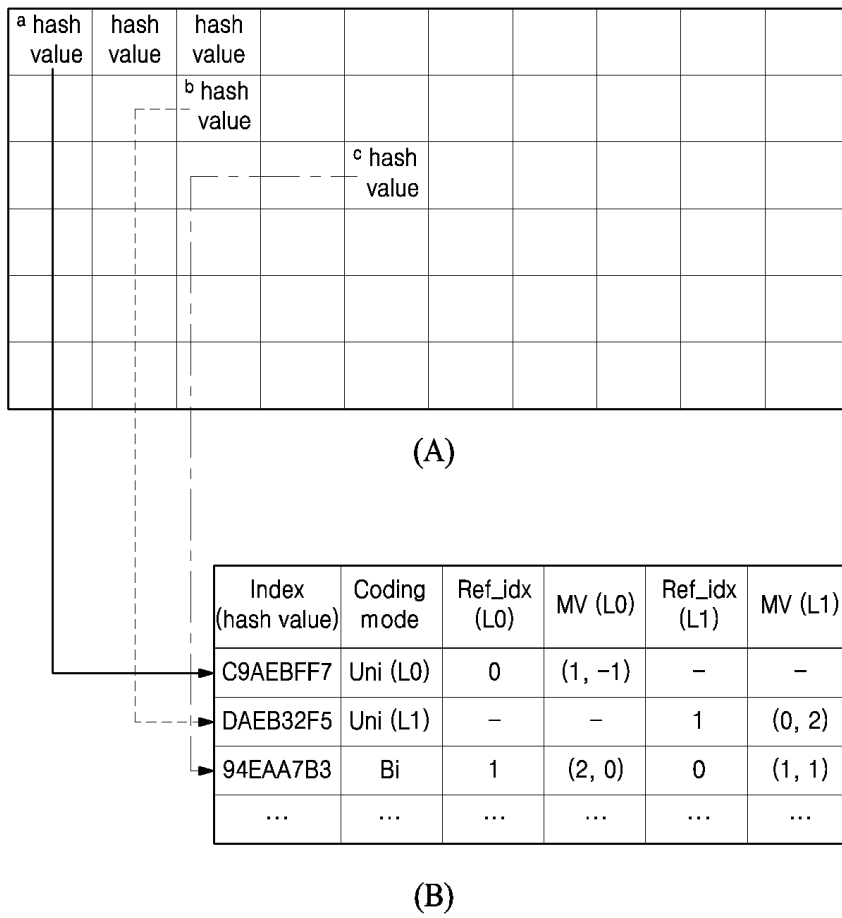
도면7



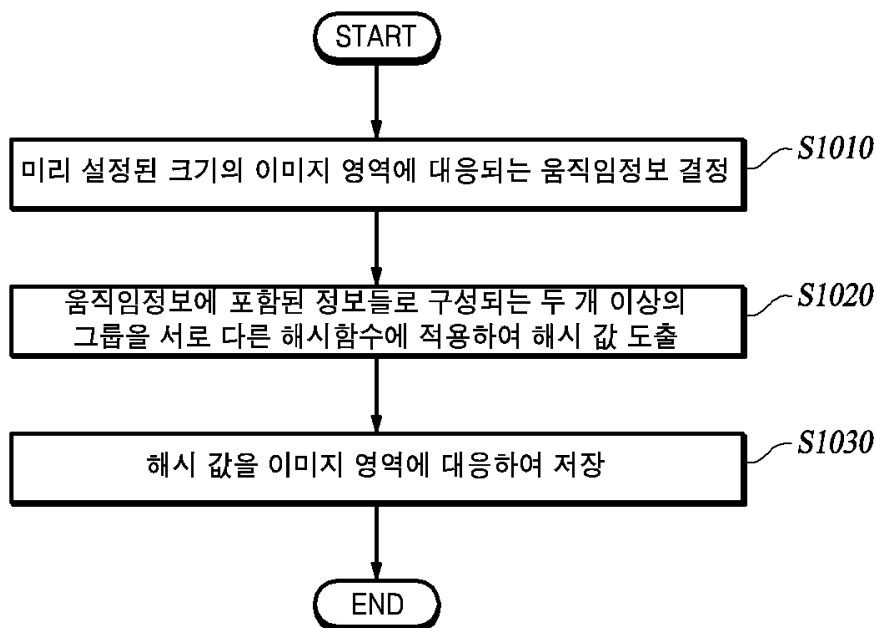
도면8



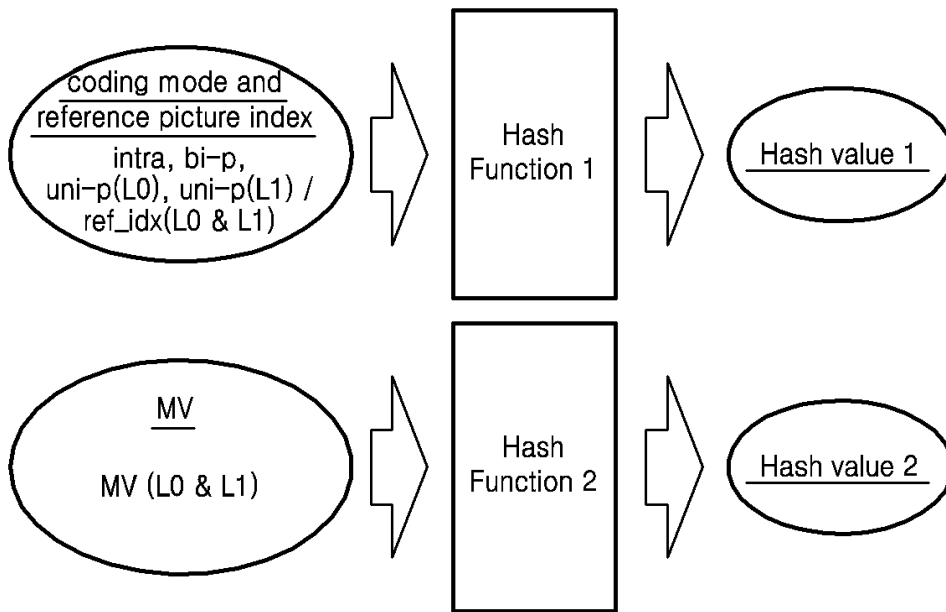
도면9



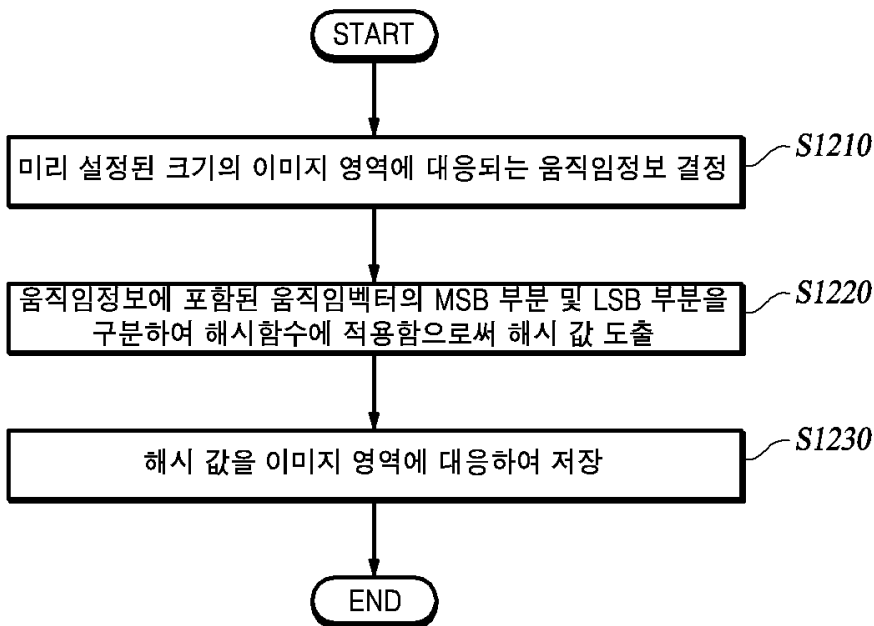
도면10



도면11



도면12



도면13

