



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2018년08월27일  
(11) 등록번호 10-1892165  
(24) 등록일자 2018년08월21일

- (51) 국제특허분류(Int. Cl.)  
*HO4N 19/50* (2014.01)
- (21) 출원번호 10-2013-7024856
- (22) 출원일자(국제) 2012년03월21일  
심사청구일자 2017년03월21일
- (85) 번역문제출일자 2013년09월23일
- (65) 공개번호 10-2014-0022009
- (43) 공개일자 2014년02월21일
- (86) 국제출원번호 PCT/KR2012/001999
- (87) 국제공개번호 WO 2012/128540  
국제공개일자 2012년09월27일
- (30) 우선권주장  
61/454,995 2011년03월21일 미국(US)  
61/466,446 2011년03월22일 미국(US)
- (56) 선행기술조사문헌  
Akira Fujibayashi ET AL., Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 4th Meeting: Daegu, Korea, 20-28 January, 2011, JCTVC-D231\*  
Jian-Liang Lin ET AL., Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 4th Meeting: Daegu, KR, 20-28 January, 2011, JCTVC-D125\*  
\*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
엘지전자 주식회사  
서울특별시 영등포구 여의대로 128 (여의도동)

(72) 발명자  
박준영  
서울 서초구 양재대로11길 19, 엘지전자 CONVERGENCE R&D 연구소 (양재동)  
박승옥  
서울 서초구 양재대로11길 19, 엘지전자 CONVERGENCE R&D 연구소 (양재동)  
(뒷면에 계속)

(74) 대리인  
인비전 특허법인

전체 청구항 수 : 총 8 항

심사관 : 박상철

(54) 발명의 명칭 움직임 벡터 예측자 선택 방법 및 이를 이용하는 장치

**(57) 요약**

본 발명은 움직임 벡터 예측자 선택 방법에 관한 것으로서, 현재 블록에 대한 움직임 벡터 예측자 후보들을 선택하는 단계 및 상기 움직임 벡터 예측자 후보들 중에서 현재 블록에 대한 움직임 벡터 예측자를 선택하는 단계를 포함하며, 상기 현재 블록에 대한 움직임 벡터 예측자 후보들은, 상기 현재 블록의 좌측 주변 블록들 중에서 이용 가능한 블록으로 가장 먼저 검색되는 제1 후보 블록의 움직임 벡터 및 상기 현재 블록의 상측 주변 블록들 중에서 이용 가능한 블록으로 가장 먼저 검색되는 제2 후보 블록의 움직임 벡터를 포함할 수 있다.

(72) 발명자

**임재현**

서울 서초구 양재대로11길 19, 엘지전자  
CONVERGENCE R&D 연구소 (양재동)

**김정선**

서울 서초구 양재대로11길 19, 엘지전자  
CONVERGENCE R&D 연구소 (양재동)

**최영희**

서울 서초구 양재대로11길 19, 엘지전자  
CONVERGENCE R&D 연구소 (양재동)

---

**전병문**

서울 서초구 양재대로11길 19, 엘지전자  
CONVERGENCE R&D 연구소 (양재동)

**전용준**

서울 서초구 양재대로11길 19, 엘지전자  
CONVERGENCE R&D 연구소 (양재동)

명세서

청구범위

청구항 1

삭제

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

삭제

청구항 7

삭제

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

**청구항 16**

디코딩 장치에 의해 수행되는 인터 예측 방법에 있어서,

제1 주변 블록 그룹으로부터 제1 탐색 순서를 기반으로 제1 움직임 벡터 예측자(motion vector predictor, MVP) 후보(candidate)를 도출하되, 상기 제1 주변 블록 그룹은 현재 블록의 좌하측 코너 주변 블록 및 좌측 주변 블록으로 구성되는(consist of) 단계;

제2 주변 블록 그룹으로부터 제2 탐색 순서를 기반으로 제2 MVP 후보를 도출하되, 상기 제2 주변 블록 그룹은 상기 현재 블록의 우상측 코너 주변 블록, 상측 주변 블록, 좌상측 코너 주변 블록으로 구성되는 단계;

상기 제1 MVP 후보 및 상기 제2 MVP 후보를 기반으로 MVP 후보 리스트를 구성하는 단계;

상기 MVP 후보 리스트를 기반으로 상기 현재 블록의 MVP를 결정하는 단계;

상기 결정된 MVP를 기반으로 상기 현재 블록의 움직임 벡터(motion vector, MV)를 도출하는 단계; 및

상기 도출된 MV를 기반으로 상기 현재 블록 내의 예측 픽셀들을 도출하는 단계를 포함하고,

상기 제1 탐색 순서는 상기 좌하측 코너 주변 블록에서 상기 좌측 주변 블록 순이고,

상기 제2 탐색 순서는 상기 우상측 코너 주변 블록에서, 상기 상측 주변 블록, 상기 좌상측 코너 주변 블록 순이며,

상기 제1 MVP 후보 도출 및 상기 제2 MVP 후보 도출은 제1 조건 및 제2 조건을 기반으로 수행되고,

상기 제1 조건은 상기 제1 주변 블록 또는 상기 제2 주변 블록 내의 대상 블록의 참조 픽처 및 참조 픽처 리스트가 상기 현재 블록의 참조 픽처 및 참조 픽처 리스트와 동일한지 여부이고,

상기 제2 조건은 상기 제1 주변 블록 또는 상기 제2 주변 블록 내의 상기 대상 블록의 상기 참조 픽처가 상기 현재 블록의 참조 픽처와 동일하고 상기 대상 블록의 상기 참조 픽처 리스트가 상기 현재 블록의 상기 참조 픽처 리스트와 다른지 여부이며,

상기 제1 MVP 후보 도출을 수행함에 있어, 상기 좌하측 코너 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 좌측 주변 블록에 대하여 상기 제1 조건 또는 상기 제2 조건의 만족 여부가 검토되고,

상기 제2 MVP 후보 도출을 수행함에 있어, 상기 우상측 코너 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 상측 주변 블록에 대하여 상기 제1 조건 또는 상기 제2 조건의 만족 여부가 검토됨을 특징으로 하는 인터 예측 방법.

**청구항 17**

삭제

**청구항 18**

삭제

**청구항 19**

삭제

**청구항 20**

제16항에 있어서,

상기 제2 MVP 후보 도출을 수행함에 있어서,

상기 상측 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 좌상측 코너 주변 블록에 대하여 상기 제1 조건 또는 상기 제2 조건의 만족 여부가 검토됨을 특징으로 하는, 인터 예측 방법.

**청구항 21**

제20항에 있어서,

상기 제1 MVP 후보 도출 및 상기 제2 MVP 후보 도출은 제3 조건을 더 기반으로 수행되고,

상기 제3 조건은 상기 제1 주변 블록 그룹 또는 상기 제2 주변 블록 그룹 내의 상기 대상 블록의 상기 참조 픽처가 상기 현재 블록의 상기 참조 픽처와 다른지 여부인 것을 특징으로 하는, 인터 예측 방법.

### 청구항 22

제21항에 있어서,

상기 제1 MVP 후보 도출을 수행함에 있어, 상기 좌하측 코너 주변 블록 및 상기 좌측 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 좌하측 코너 주변 블록에 대하여 상기 제3 조건의 만족 여부가 검토되고,

상기 제2 MVP 후보 도출을 수행함에 있어, 상기 우상측 코너 주변 블록, 상기 상측 주변 블록 및 상기 좌상측 코너 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 우상측 코너 주변 블록에 대하여 상기 제3 조건의 만족 여부가 검토됨을 특징으로 하는, 인터 예측 방법.

### 청구항 23

인터 예측을 수행하는 디코딩 장치에 있어서,

제1 주변 블록 그룹으로부터 제1 탐색 순서를 기반으로 제1 움직임 벡터 예측자(motion vector predictor, MVP) 후보(candidate)를 도출하고, 제2 주변 블록 그룹으로부터 제2 탐색 순서를 기반으로 제2 MVP 후보를 도출하고, 상기 제1 MVP 후보 및 상기 제2 MVP 후보를 기반으로 MVP 후보 리스트를 구성하고, 상기 MVP 후보 리스트를 기반으로 현재 블록의 MVP를 결정하고, 상기 결정된 MVP를 기반으로 상기 현재 블록의 움직임 벡터(motion vector, MV)를 도출하고, 상기 도출된 MV를 기반으로 상기 현재 블록 내의 예측 픽셀들을 도출하는 예측부를 포함하고,

상기 제1 주변 블록 그룹은 상기 현재 블록의 좌하측 코너 주변 블록 및 좌측 주변 블록으로 구성되고(consist of), 상기 제2 주변 블록 그룹은 상기 현재 블록의 우상측 코너 주변 블록, 상측 주변 블록, 좌상측 코너 주변 블록으로 구성되며,

상기 제1 탐색 순서는 상기 좌하측 코너 주변 블록에서 상기 좌측 주변 블록 순이고,

상기 제2 탐색 순서는 상기 우상측 코너 주변 블록에서, 상기 상측 주변 블록, 상기 좌상측 코너 주변 블록 순이며,

상기 예측부는 상기 제1 MVP 후보 도출 및 상기 제2 MVP 후보 도출을 제1 조건 및 제2 조건을 기반으로 수행하고,

상기 제1 조건은 상기 제1 주변 블록 그룹 또는 상기 제2 주변 블록 그룹 내의 대상 블록의 참조 픽처 및 참조 픽처 리스트가 상기 현재 블록의 참조 픽처 및 참조 픽처 리스트와 동일한지 여부이고,

상기 제2 조건은 상기 제1 주변 블록 그룹 또는 상기 제2 주변 블록 그룹 내의 상기 대상 블록의 상기 참조 픽처가 상기 현재 블록의 참조 픽처와 동일하고 상기 대상 블록의 상기 참조 픽처 리스트가 상기 현재 블록의 상기 참조 픽처 리스트와 다른지 여부이며,

상기 예측부는 상기 제1 MVP 후보 도출을 수행함에 있어, 상기 좌하측 코너 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 좌측 주변 블록에 대하여 상기 제1 조건 또는 상기 제2 조건의 만족 여부를 검토하고,

상기 예측부는 상기 제2 MVP 후보 도출을 수행함에 있어, 상기 우상측 코너 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 상측 주변 블록에 대하여 상기 제1 조건 또는 상기 제2 조건의 만족 여부를 검토함을 특징으로 하는, 디코딩 장치.

### 청구항 24

삭제

### 청구항 25

삭제

청구항 26

삭제

청구항 27

제23항에 있어서,

상기 예측부는 상기 제2 MVP 후보 도출을 수행함에 있어서,

상기 상측 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 좌상측 코너 주변 블록에 대하여 상기 제1 조건 또는 상기 제2 조건의 만족 여부를 검토함을 특징으로 하는, 디코딩 장치

청구항 28

제27항에 있어서,

상기 예측부는 상기 제1 MVP 후보 도출 및 상기 제2 MVP 후보 도출을 제3 조건을 더 기반으로 수행하고,

상기 제3 조건은 상기 제1 주변 블록 그룹 또는 상기 제2 주변 블록 그룹 내의 상기 대상 블록의 상기 참조 픽처가 상기 현재 블록의 상기 참조 픽처와 다른지 여부인 것을 특징으로 하는, 디코딩 장치.

청구항 29

제28항에 있어서,

상기 제1 MVP 후보 도출을 수행함에 있어, 상기 좌하측 코너 주변 블록 및 상기 좌측 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 좌하측 코너 주변 블록에 대하여 상기 제3 조건의 만족 여부가 검토되고,

상기 제2 MVP 후보 도출을 수행함에 있어, 상기 우상측 코너 주변 블록, 상기 상측 주변 블록 및 상기 좌상측 코너 주변 블록이 상기 제1 조건 및 상기 제2 조건을 만족하지 않는 경우, 상기 우상측 코너 주변 블록에 대하여 상기 제3 조건의 만족 여부가 검토됨을 특징으로 하는, 디코딩 장치.

## 발명의 설명

### 기술 분야

[0001]

본 발명은 동영상 압축 기술에 관한 것으로서, 더 구체적으로는 주변 블록의 정보를 이용하여 현재 블록의 정보를 예측하는 방법에 관한 것이다.

### 배경 기술

[0002]

최근, 고해상도, 고품질의 영상에 대한 요구가 다양한 응용 분야에서 증가하고 있다. 하지만, 영상의 고해상도, 고품질이 될수록 해당 영상에 관한 정보량도 함께 증가한다. 따라서 기존의 유무선 광대역 회선과 같은 매체를 이용하여 영상 정보를 전송하거나 기존의 저장 매체를 이용해 영상 정보를 저장하는 경우, 정보의 전송 비용과 저장 비용이 증가하게 된다.

[0003]

고해상도, 고품질 영상의 정보를 효과적으로 전송하거나 저장하고, 재생하기 위해 고효율의 영상 압축 기술을 이용할 수 있다.

[0004]

영상 압축의 효율을 높이기 위해, 현재 블록의 정보를 그대로 전송하지 않고, 현재 블록의 주변 블록들의 정보를 이용하여 예측하는 방법을 사용할 수 있다.

[0005]

예측의 방법으로서, 인터 예측과 인트라 예측을 이용할 수 있다.

[0006]

인터 예측(inter prediction) 방법에서는 다른 픽처의 정보를 참조하여 현재 픽처(picture)의 화소값을 예측하며, 화면 내 예측 방법(intra prediction)에서는 동일한 픽처 내에서 화소 간 연관 관계를 이용하여 화소값을

예측한다.

[0007] 인터 예측을 수행하는 경우에는, 다른 핵심에서 예측에 이용되는 부분을 지정하기 위해, 인터 예측 모드인 주변 블록으로부터 참조 핵심을 지시하는 정보와 움직임 벡터를 나타내는 정보를 활용할 수 있다.

## 발명의 내용

### 해결하려는 과제

[0008] 본 발명은 압축 효율을 높이고 정보 전송량을 감소시킴과 동시에 동영상 인코딩/디코딩 과정의 복잡도를 줄일 수 있는 예측 방법과 이를 이용하는 장치를 제공하는 것을 목적으로 한다.

[0009] 본 발명은 인터 예측에 있어서, 주변 정보를 효과적으로 이용하여 현재 블록에 대한 움직임 정보 예측자 후보들을 구성할 수 있도록 주변 블록을 스캔하는 방법과 이를 이용하는 장치를 제공하는 것을 목적으로 한다.

[0010] 본 발명은 주변 블록으로부터 획득하는 움직임 정보를 현재 블록에 대한 움직임 정보 예측자 후보로서 이용하기 위해 움직임 정보를 스케일링하는 방법과 이를 이용하는 장치를 제공하는 것을 목적으로 한다.

### 과제의 해결 수단

[0011] (1) 본 발명의 일 실시형태는 움직임 벡터 예측자 선택 방법으로서, 현재 블록에 대한 움직임 벡터 예측자 후보들을 선택하는 단계 및 상기 움직임 벡터 예측자 후보들 중에서 현재 블록에 대한 움직임 벡터 예측자를 선택하는 단계를 포함하며, 상기 현재 블록에 대한 움직임 벡터 예측자 후보들은, 상기 현재 블록의 좌측 주변 블록들 중에서 이용 가능한 블록으로 가장 먼저 검색되는 제1 후보 블록의 움직임 벡터 및 상기 현재 블록의 상측 주변 블록들 중에서 이용 가능한 블록으로 가장 먼저 검색되는 제2 후보 블록의 움직임 벡터를 포함할 수 있다.

[0012] (2) (1)에서, 상기 이용 가능한 블록은, 제1 조건으로서 상기 현재 블록과 동일한 참조 핵심 및 동일한 참조 핵심 리스트를 가질 것, 제2 조건으로서 상기 현재 블록과 동일한 참조 핵심 및 상이한 참조 핵심 리스트를 가질 것, 제3 조건으로서 상기 현재 블록과 상이한 참조 핵심 및 동일한 참조 핵심 리스트를 가질 것, 제4 조건으로서 상기 현재 블록과 상이한 참조 핵심 및 상이한 참조 핵심 리스트를 가질 것 중 어느 하나를 만족하는 블록으로서, 인터 예측 모드가 적용된 블록일 수 있다.

[0013] (3) (2)에서, 상기 현재 블록에 대한 움직임 벡터 예측자 선택 단계에서는,

[0014] 상기 제1 조건 내지 제4 조건 중 낮은 번호의 조건이 만족되는지 여부에 더 높은 우선 순위를 두고 검색할 수 있다.

[0015] (4) (2)에서, 상기 제1 후보 블록 또는 상기 제2 후보 블록 중 상기 제3 조건 또는 상기 제4 조건을 만족하는 블록의 움직임 벡터는 스케일링될 수 있다.

[0016] (5) (4)에서, 상기 스케일링은 상기 현재 블록이 속하는 현재 핵심과 상기 현재 블록의 참조 핵심 사이의 거리; 및 상기 현재 핵심과 상기 제1 후보 블록 또는 상기 제2 후보 블록 중 상기 제3 조건 또는 상기 제4 조건을 만족하는 블록의 참조 핵심 사이의 거리에 기반하여 수행될 수 있다.

[0017] (6) (4)에서, 상기 스케일링의 회수를 소정의 회수로 제한할 수 있다.

[0018] (7) (1)에서, 상기 좌측 주변 블록들은, 상기 현재 블록의 좌하측 코너에 위치하는 제1 주변 블록 및 상기 현재 블록 좌측의 하측에 위치하는 제2 주변 블록일 수 있다.

[0019] (8) (7)에서, 상기 현재 블록에 대한 움직임 벡터 예측자 후보들을 선택하는 단계는, 상기 제1 주변 블록 및 상기 제2 주변 블록에 대하여, 제1 조건으로서 상기 현재 블록과 동일한 참조 핵심 및 동일한 참조 핵심 리스트를 가질 것, 제2 조건으로서 상기 현재 블록과 동일한 참조 핵심 및 상이한 참조 핵심 리스트를 가질 것, 제3 조건으로서 상기 현재 블록과 상이한 참조 핵심 및 동일한 참조 핵심 리스트를 가질 것, 제4 조건으로서 상기 현재 블록과 상이한 참조 핵심 및 상이한 참조 핵심 리스트를 가질 것 중 어느 하나를 만족하는지를 제1 조건부터 제4 조건의 순서로 판단하는 조건 판단 단계 및 상기 제1 조건 내지 제4 조건 중 어느 하나를 만족하는 것으로 먼저 검색되는 블록을 상기 제1 후보 블록으로 결정하는 후보 블록 결정 단계를 포함할 수 있다.

[0020] (9) (8)에서, 상기 조건 판단 단계는, 상기 제1 주변 블록에 대하여 상기 제1 조건 및 제2 조건의 만족 여부를 판단하는 단계, 상기 제2 주변 블록에 대하여 상기 제1 조건 및 제2 조건의 만족 여부를 판단하는 단계, 상기 제1 주변 블록에 대하여 상기 제3 조건 및 제4 조건의 만족 여부를 판단하는 단계 및 상기 제2 주변 블록에 대

하여 상기 제3 조건 및 제4 조건의 만족 여부를 판단하는 단계를 포함할 수 있다.

[0021] (10) (1)에서, 상기 상측 주변 블록들은, 상기 현재 블록의 우상측 코너에 위치하는 제3 주변 블록, 상기 현재 블록 상측의 우측에 위치하는 제4 주변 블록 및 상기 현재 블록의 좌상측 코너에 위치하는 제5 주변 블록일 수 있다.

[0022] (11) (10)에서, 상기 현재 블록에 대한 움직임 벡터 예측자 후보들을 선택하는 단계는, 상기 제3 주변 블록, 제4 주변 블록 및 제5 주변 블록에 대하여, 제1 조건으로서 상기 현재 블록과 동일한 참조 픽처 및 동일한 참조 픽처 리스트를 가질 것, 제2 조건으로서 상기 현재 블록과 동일한 참조 픽처 및 상이한 참조 픽처 리스트를 가질 것, 제3 조건으로서 상기 현재 블록과 상이한 참조 픽처 및 동일한 참조 픽처 리스트를 가질 것, 제4 조건으로서 상기 현재 블록과 상이한 참조 픽처 및 상이한 참조 픽처 리스트를 가질 것 중 어느 하나를 만족하는지를 제1 조건부터 제4 조건의 순서로 판단하는 조건 판단 단계 및 상기 제1 조건 내지 제4 조건 중 어느 하나를 만족하는 것으로 먼저 검색되는 블록을 상기 제2 후보 블록으로 결정하는 후보 블록 결정 단계를 포함할 수 있다.

[0023] (12) (11)에서, 상기 조건 판단 단계는, 상기 제3 주변 블록에 대하여 상기 제1 조건 및 제2 조건의 만족 여부를 판단하는 단계, 상기 제4 주변 블록에 대하여 상기 제1 조건 및 제2 조건의 만족 여부를 판단하는 단계, 상기 제5 주변 블록에 대하여 상기 제1 조건 및 제2 조건의 만족 여부를 판단하는 단계, 상기 제3 주변 블록에 대하여 상기 제3 조건 및 제4 조건의 만족 여부를 판단하는 단계 및 상기 제5 주변 블록에 대하여 상기 제3 조건 및 제4 조건의 만족 여부를 판단하는 단계를 포함할 수 있다.

[0024] (13) 본 발명의 다른 실시 형태는 상기 현재 블록에 대한 움직임 벡터 예측자를 선택하는 방법을 이용하는 디코딩 장치로서 현재 블록에 대한 예측을 수행하는 예측부 및 상기 현재 블록의 주변 블록에 대한 정보를 저장하는 메모리를 포함하며, 상기 예측부는, 현재 블록에 대한 움직임 벡터 예측자 후보들을 선택하고, 상기 움직임 벡터 예측자 후보들 중에서 현재 블록에 대한 움직임 벡터 예측자를 선택하며, 상기 현재 블록에 대한 움직임 벡터 예측자 후보들은, 상기 현재 블록의 좌측 주변 블록들 중에서 이용 가능한 블록으로 가장 먼저 검색되는 제1 후보 블록의 움직임 벡터 및 상기 현재 블록의 상측 주변 블록들 중에서 이용 가능한 블록으로 가장 먼저 검색되는 제2 후보 블록의 움직임 벡터를 포함할 수 있다.

[0025] (14) (13)에서, 상기 이용 가능한 블록은, 제1 조건으로서 상기 현재 블록과 동일한 참조 픽처 및 동일한 참조 픽처 리스트를 가질 것, 제2 조건으로서 상기 현재 블록과 동일한 참조 픽처 및 상이한 참조 픽처 리스트를 가질 것, 제3 조건으로서 상기 현재 블록과 상이한 참조 픽처 및 동일한 참조 픽처 리스트를 가질 것, 제4 조건으로서 상기 현재 블록과 상이한 참조 픽처 및 상이한 참조 픽처 리스트를 가질 것 중 어느 하나를 만족하는 블록으로서, 인터 예측 모드가 적용된 블록이며, 상기 예측부는, 상기 제1 조건 내지 제4 조건 중 낮은 번호의 조건이 만족되는지 여부에 더 높은 우선 순위를 두고 검색할 수 있다.

[0026] (15) (14)에서, 상기 예측부는, 상기 제1 후보 블록 또는 상기 제2 후보 블록 중 상기 제3 조건 또는 상기 제4 조건을 만족하는 블록의 움직임 벡터를 스케일링할 수 있다.

### 발명의 효과

[0027] 본 발명에서 제시하는 예측 방법에 의하면, 압축 효율을 높이고 정보 전송량을 감소시킴과 동시에 동영상 인코딩/디코딩 과정의 복잡도를 줄일 수 있다.

[0028] 본 발명에서 제시하는 예측 방법에 의하면, 주변 정보를 효과적으로 이용하여 현재 블록에 대한 움직임 정보 예측자 후보들을 구성할 수 있다.

[0029] 본 발명에서 제시하는 예측 방법에 의하면, 주변 블록으로부터 획득하는 움직임 정보를 스케일링하여 현재 블록의 움직임 벡터 예측자 후보로서 이용할 수 있다.

### 도면의 간단한 설명

[0030] 도 1은 본 발명의 일 실시예에 따른 영상 인코딩 장치(인코더)를 개략적으로 도시한 블록도이다.

도 2는 본 발명의 일 실시예에 따른 영상 디코딩 장치(디코더)를 개략적으로 나타낸 블록도이다.

도 3은 인터 예측 모드에서 AMVP(Advanced Motion Vector Predictor)가 적용되는 경우, 움직임 벡터를 도출하는 방법에 관한 일 실시예를 개략적으로 설명하는 순서도이다.

도 4는 MVP 후보 리스트 생성 방법에 관한 일 실시예를 개략적으로 설명하는 개념도이다.

도 5는 본 발명에 따른 스케일링의 방법을 개략적으로 설명하는 개념도이다.

도 6은 본 발명에 따른 스케일링의 방법을 개략적으로 설명하는 순서도이다.

도 7은 본 발명에 따라서, 현재 픽처(현재 프레임) 내 현재 블록의 주변 블록이 가지는 움직임 벡터를 스케일링 하는 방법을 개략적으로 설명하는 도면이다.

도 8은 본 발명의 일 실시예에 따라서 움직임 벡터 경쟁을 수행하는 방법을 개략적으로 설명하는 도면이다.

도 9는 본 발명의 일 실시예에 따라서 AMVP를 적용하는 다른 방법을 개략적으로 설명하는 도면이다.

도 10은 본 발명에 따라서 현재 블록에 대한 움직임 벡터 예측자를 선택하는 방법을 개략적으로 설명하는 순서 도이다.

### 발명을 실시하기 위한 구체적인 내용

[0031] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니다. 본 명세서에서 사용하는 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명의 기술적 사상을 한정하려는 의도로 사용되는 것은 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0032] 한편, 본 발명에서 설명되는 도면상의 각 구성들은 영상 인코딩/디코딩 장치에서 서로 다른 특징적인 기능들에 관한 설명의 편의를 위해 독립적으로 도시된 것으로서, 각 구성들이 서로 별개의 하드웨어나 별개의 소프트웨어로 구현된다는 것을 의미하지는 않는다. 예컨대, 각 구성 중 두 개 이상의 구성이 합쳐져 하나의 구성으로 이를 수도 있고, 하나의 구성이 복수의 구성으로 나뉘어질 수도 있다. 각 구성이 통합 및/또는 분리된 실시예도 본 발명의 본질에서 벗어나지 않는 한 본 발명의 권리범위에 포함된다.

[0033] 이하, 첨부한 도면들을 참조하여, 본 발명에 관하여 상세하게 설명하고자 한다.

[0034] 도 1은 본 발명의 일 실시예에 따른 영상 인코딩 장치(인코더)를 개략적으로 도시한 블록도이다. 도 1을 참조하면, 영상 인코딩 장치(100)는 픽처 분할부(105), 예측부(110), 변환부(115), 양자화부(120), 재정렬부(125), 엔트로피 인코딩부(130), 역양자화부(135), 역변환부(140), 필터부(145) 및 메모리(150)를 구비한다.

[0035] 픽처 분할부(105)는 입력된 픽처를 적어도 하나의 처리 단위로 분할할 수 있다. 이때, 처리 단위는 예측 유닛(Prediction Unit, 이하 'PU' 라 함)일 수도 있고, 변환 유닛(Transform Unit, 이하 'TU' 라 함)일 수도 있으며, 코딩 유닛(Coding Unit, 이하 'CU' 라 함)일 수도 있다. 다만, 본 명세서에서는 설명의 편의를 위해, 예측 유닛을 예측 블록, 변환 유닛을 변환 블록, 인코딩 유닛을 인코딩 블록으로 표현할 수 있다.

[0036] 예측부(110)는 인터 예측(inter prediction, 화면 간 예측)을 수행하는 인터 예측부와 인트라 예측(intra prediction, 화면 내 예측)을 수행하는 인트라 예측부를 포함할 수 있다. 코딩 효율을 높이기 위해, 영상 신호를 그대로 인코딩하는 것이 아니라, 이미 인코딩된 영역을 이용하여 영상을 예측하고, 원래의 영상과 예측 영상 사이의 래지듀얼 값을 예측 영상에 더하여 영상을 복원할 수 있도록 인코딩 한다.

[0037] 예측에 이용되는 이미 인코딩된 영역을 포함하는 픽처로서, I 픽처(I 슬라이스), P 픽처(P 슬라이스), B 픽처(B 슬라이스) 등이 있다. I 슬라이스는 인트라 예측에 의해서만 디코딩되는 슬라이스이다. P 슬라이스는 각 블록의 샘플 값을 예측하기 위해 적어도 하나의 움직임 벡터 및 참조 픽처 인덱스를 이용한 인터 예측 또는 인트라 예측을 이용해서 디코딩될 수 있는 슬라이스이다. B 슬라이스는 각 블록의 샘플 값을 예측하기 위해 적어도 두 개의 움직임 벡터들과 참조 픽처 인덱스들을 이용한 인터 예측 또는 인트라 예측을 이용해서 디코딩될 수 있는 슬라이스이다.

[0038] 예측부(110)는, 픽처 분할부(105)에서 픽처의 처리 단위에 대하여 예측을 수행하여 예측된 샘플로 구성되는 예측 블록을 생성한다. 예측부(110)에서 픽처의 처리 단위는 CU일 수도 있고, TU일 수도 있고, PU일 수도 있다. 또한, 해당 처리 단위에 대하여 실시되는 예측이 인터 예측인지 인트라 예측인지를 결정하고, 각 예측 방법의

구체적인 내용(예컨대, 예측 모드 등)를 정할 수 있다. 이때, 예측이 수행되는 처리 단위와 예측 방법 및 구체적인 내용이 정해지는 처리 단위는 서로 다를 수 있다. 예컨대, 예측의 방법과 예측 모드 등은 PU 단위로 결정되고, 예측의 수행은 TU 단위로 수행될 수도 있다.

[0039] 인터 예측을 통해서는 현재 픽처의 이전 픽처 및/또는 이후 픽처 중 적어도 하나의 픽처의 정보를 기초로 예측을 수행하여 예측 블록을 생성할 수 있다. 또한, 인트라 예측을 통해서는 현재 픽처 내의 화소 정보를 기초로 예측을 수행하여 예측 블록을 생성할 수 있다.

[0040] 인터 예측에서는 현재 블록에 대하여, 참조 픽처를 선택하고 현재 블록과 동일한 크기의 참조 블록을 선택해서, 현재 블록에 대한 예측 블록을 생성할 수 있다. 예컨대, 인터 예측에서는 현재 블록과의 레지듀얼(residual) 신호가 최소화되며 움직임 벡터 크기 역시 최소가 되도록 예측 블록을 생성할 수 있다. 인터 예측의 방법으로서, 스kip(skip) 모드, 머지(merge) 모드, AMVP(Advanced Motion Vector Prediction) 등을 이용할 수 있다. 예측 블록은 1/2 화소 샘플 단위와 1/4 화소 샘플 단위와 같이 정수 이하 샘플 단위로 생성될 수도 있다. 이때, 움직임 벡터 역시 정수 화소 이하의 단위로 표현될 수 있다. 예컨대 휙도 화소에 대해서는 1/4 화소 단위로, 색차 화소에 대해서는 1/8 화소 단위로 표현될 수 있다.

[0041] 인터 예측을 통해 선택된 참조 픽처의 인덱스, 움직임 벡터 예측자, 레지듀얼 신호 등의 정보는 엔트로피 인코딩되어 디코딩 장치에 전달된다.

[0042] 인트라 예측을 수행하는 경우에도 예측이 수행되는 처리 단위와 예측 방법 및 구체적인 내용이 정해지는 처리 단위는 서로 다를 수도 있다. 예컨대, PU 단위로 예측 모드가 정해져서 PU 단위로 예측이 수행될 수도 있고, PU 단위로 예측 모드가 정해지고 TU 단위로 화면 내 예측이 수행될 수도 있다.

[0043] 인트라 예측에서 예측 모드는 33개의 방향성 예측 모드와 적어도 2개 이상의 비방향성 모드를 가질 수 있다. 비방향성 모드는 DC 예측 모드 및 플레너 모드(Planar 모드) 등을 포함할 수 있다.

[0044] 인트라 예측에서는 참조 샘플에 필터를 적용한 후 예측 블록을 생성할 수 있다. 이때, 참조 샘플에 필터를 적용할 것인지는 현재 블록의 인트라 예측 모드 및/또는 사이즈에 따라 결정될 수도 있다.

[0045] PU는 더 이상 분할되지 않는 CU로부터 다양한 사이즈/형태로 결정될 수 있다. 예컨대 인터 예측의 경우에 PU는  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ , 또는  $N \times N$  등의 크기를 가질 수 있다. 인트라 예측의 경우에 PU는  $2N \times 2N$  또는  $N \times N$  ( $N$ 은 정수) 등의 크기를 가질 수 있다. 이때,  $N \times N$  크기의 PU는 특정한 경우에만 적용하도록 설정할 수도 있다. 예컨대 최소 크기 코딩 유닛에 대해서만  $N \times N$ 의 PU를 이용하도록 정하거나 인트라 예측에 대해서만 이용하도록 정할 수 있다. 또한, 상술한 크기의 PU 외에,  $N \times mN$ ,  $mN \times N$ ,  $2N \times mN$  또는  $mN \times 2N$  ( $m < 1$ ) 등의 크기를 가지는 PU를 더 정의하여 사용할 수도 있다.

[0046] 생성된 예측 블록과 원본 블록 사이의 레지듀얼 값(레지듀얼 블록 또는 레지듀얼 신호)은 변환부(115)로 입력된다. 또한, 예측을 위해 사용한 예측 모드 정보, 움직임 벡터 정보 등은 레지듀얼 값과 함께 엔트로피 인코딩부(130)에서 인코딩되어 디코딩 장치에 전달된다.

[0047] 변환부(115)는 변환 단위로 레지듀얼 블록에 대한 변환을 수행하고 변환 계수를 생성한다. 변환부(115)에서의 변환 단위는 TU일 수 있으며, 쿼드 트리(quad tree) 구조를 가질 수 있다. 이때, 변환 단위의 크기는 소정의 최대 및 최소 크기의 범위 내에서 정해질 수 있다. 변환부(115)는 레지듀얼 블록을 DCT(Discrete Cosine Transform) 또는 DST(Discrete Sine Transform)를 이용하여 변환할 수 있다.

[0048] 양자화부(120)는 변환부(115)에서 변환된 레지듀얼 값들을 양자화하여 양자화 계수를 생성할 수 있다. 양자화부(120)에서 산출된 값은 역양자화부(135)와 재정렬부(125)에 제공된다.

[0049] 재정렬부(125)는 양자화부(120)로부터 제공된 양자화 계수를 재정렬한다. 양자화 계수를 재정렬함으로써 엔트로피 인코딩부(130)에서의 인코딩의 효율을 높일 수 있다. 재정렬부(125)는 계수 스캐닝(Coefficient Scanning) 방법을 통해 2차원 블록 형태의 양자화 계수들을 1차원의 형태로 재정렬할 수 있다. 재정렬부(125)에서는 양자화부에서 전송된 계수들의 확률적인 통계를 기반으로 계수 스캔ning의 순서를 변경함으로써 엔트로피 인코딩부(130)에서의 엔트로피 인코딩 효율을 높일 수도 있다.

[0050] 엔트로피 인코딩부(130)는 재정렬부(125)에 의해 재정렬된 양자화 계수들에 대한 엔트로피 인코딩을 수행할 수 있다. 엔트로피 인코딩에는 예를 들어, 지수 골롬(Exponential Golomb), CAVLC(Context-Adaptive Variable Length Coding), CABAC(Context-Adaptive Binary Arithmetic Coding)과 같은 인코딩 방법을 사용할 수 있다. 엔트로피 인코딩부(130)는 재정렬부(125) 및 예측부(110)로부터 전달받은 CU의 양자화 계수 정보 및 블록 타입

정보, 예측 모드 정보, 분할 단위 정보, PU 정보 및 전송 단위 정보, 움직임 벡터 정보, 참조 픽처 정보, 블록의 보간 정보, 필터링 정보 등 다양한 정보를 인코딩할 수 있다.

[0051] 또한, 엔트로피 인코딩부(130)는 필요한 경우에, 전송하는 파라미터 셋 또는 선택스에 일정한 변경을 가할 수도 있다.

[0052] 역양자화부(135)는 양자화부(120)에서 양자화된 값들을 역양자화하고, 역변환부(140)는 역양자화부(135)에서 역양자화된 값들을 역변환한다. 역양자화부(135) 및 역변환부(140)에서 생성된 레지듀얼 값은 예측부(110)에서 예측된 예측 블록과 합쳐져 복원 블록(Reconstructed Block)이 생성될 수 있다.

[0053] 필터부(145)는 디블록킹 필터, ALF(Adaptive Loop Filter), SAO(Sample Adaptive Offset)를 복원된 픽처에 적용할 수 있다.

[0054] 디블록킹 필터는 복원된 픽처에서 블록 간의 경계에 생긴 블록 왜곡을 제거할 수 있다. ALF(Adaptive Loop Filter)는 디블록킹 필터를 통해 블록이 필터링된 후 복원된 영상과 원래의 영상을 비교한 값을 기초로 필터링을 수행할 수 있다. ALF는 고효율을 적용하는 경우에만 수행될 수도 있다. SAO는 디블록킹 필터가 적용된 레지듀얼 블록에 대하여, 화소 단위로 원본 영상과의 오프셋 차이를 복원하며, 밴드 오프셋(Band Offset), 에지 오프셋(Edge Offset) 등의 형태로 적용된다.

[0055] 한편, 인터 예측에 사용되는 복원 블록에 대해서 필터부(145)는 필터링을 적용하지 않을 수 있다.

[0056] 메모리(150)는 필터부(145)를 통해 산출된 복원 블록 또는 픽처를 저장할 수 있다. 메모리(150)에 저장된 복원 블록 또는 픽처는 인터 예측을 수행하는 예측부(110)에 제공될 수 있다.

[0057] 도 2는 본 발명의 일 실시예에 따른 영상 디코딩 장치(디코더)를 개략적으로 나타낸 블록도이다. 도 2를 참조하면, 영상 디코딩 장치(200)는 엔트로피 디코딩부(210), 재정렬부(215), 역양자화부(220), 역변환부(225), 예측부(230), 필터부(235) 메모리(240)를 포함할 수 있다.

[0058] 인코딩 장치로부터 영상 비트 스트림이 입력된 경우, 입력된 비트 스트림은 인코딩 장치에서 영상 정보가 처리된 절차의 역과정에 따라서 디코딩될 수 있다.

[0059] 예컨대, 영상 인코딩 장치에서 엔트로피 인코딩을 수행하기 위해 CAVLC 등의 가변 길이 인코딩(Variable Length Coding: VLC, 이하 ‘VLC’ 라 함)가 사용된 경우에, 엔트로피 디코딩부(210)도 인코딩 장치에서 사용한 VLC 테이블과 동일한 VLC 테이블로 구현하여 엔트로피 디코딩을 수행할 수 있다. 또한, 영상 인코딩 장치에서 엔트로피 인코딩을 수행하기 위해 CABAC을 이용한 경우에, 엔트로피 디코딩부(210)는 이에 대응하여 CABAC을 이용한 엔트로피 디코딩을 수행할 수 있다.

[0060] 엔트로피 디코딩부(210)에서 디코딩된 정보 중 예측 블록을 생성하기 위한 정보는 예측부(230)로 제공되고 엔트로피 디코딩부에서 엔트로피 디코딩이 수행된 레지듀얼 값은 재정렬부(215)로 입력될 수 있다.

[0061] 재정렬부(215)는 엔트로피 디코딩부(210)에서 엔트로피 디코딩된 비트 스트림을 영상 인코딩기에서 재정렬한 방법을 기초로 재정렬할 수 있다. 재정렬부(215)는 1차원 벡터 형태로 표현된 계수들을 다시 2차원의 블록 형태의 계수로 복원하여 재정렬할 수 있다. 재정렬부(215)는 인코딩 장치에서 수행된 계수 스캐닝에 관련된 정보를 제공받고 인코딩 장치에서 수행된 스캐닝 순서에 기초하여 역으로 스캐닝하는 방법을 통해 재정렬을 수행할 수 있다.

[0062] 역양자화부(220)는 인코딩 장치에서 제공된 양자화 파라미터와 재정렬된 블록의 계수값을 기초로 역양자화를 수행할 수 있다.

[0063] 역변환부(225)는 영상 인코딩 장치에서 수행된 양자화 결과에 대해, 인코딩 장치의 변환부가 수행한 DCT 또는 DST에 대해 역DCT 또는 역DST를 수행할 수 있다. 역변환은 인코딩 장치에서 결정된 전송 단위 또는 영상의 분할 단위를 기초로 수행될 수 있다. 인코딩 장치의 변환부에서는 예측 방법, 현재 블록의 크기 및 예측 방향 등 복수의 정보에 따라 DCT 또는 DST를 선택적으로 수행될 수 있고, 디코딩 장치의 역변환부(225)는 인코딩 장치의 변환부에서 수행된 변환 정보를 기초로 역변환을 수행할 수 있다.

[0064] 예측부(230)는 엔트로피 디코딩부(210)에서 제공된 예측 블록 생성 관련 정보와 메모리(240)에서 제공된 이전에 디코딩된 블록 및/또는 픽처 정보를 기초로 예측 블록을 생성할 수 있다. 복원 블록은 예측부(230)에서 생성된 예측 블록과 역변환부(225)에서 제공된 레지듀얼 블록을 이용해 생성될 수 있다.

- [0065] 예측부(230)에서 수행하는 구체적인 예측의 방법은 인코딩 장치의 예측부에서 수행되는 예측의 방법과 동일하다.
- [0066] 현재 블록에 대한 예측 모드가 인트라 예측(intra prediction) 모드인 경우에, 현재 픽처 내의 화소 정보를 기초로 예측 블록을 생성하는 인트라 예측을 수행할 수 있다.
- [0067] 인트라 예측에서 예측 모드는 33개의 방향성 예측 모드와 적어도 2개 이상의 비방향성 모드를 가질 수 있다. 비방향성 모드는 DC 예측 모드 및 플레너 모드(Planar 모드) 등을 포함할 수 있다.
- [0068] 인트라 예측에서는 참조 샘플에 필터를 적용한 후 예측 블록을 생성할 수 있다. 이때, 참조 샘플에 필터를 적용할 것인지는 현재 블록의 인트라 예측 모드 및/또는 사이즈에 따라 결정될 수도 있다.
- [0069] 현재 블록에 대한 예측 모드가 인터 예측(inter prediction) 모드인 경우에, 현재 픽처의 이전 픽처 또는 이후 픽처 중 적어도 하나를 참조 픽처로 하고, 참조 픽처에 포함된 정보를 기초로 현재 블록에 대한 인터 예측을 수행할 수 있다. 구체적으로, 인터 예측에서는 현재 블록에 대하여, 참조 픽처를 선택하고 현재 블록과 동일한 크기의 참조 블록을 선택해서, 현재 블록에 대한 예측 블록을 생성할 수 있다. 예컨대, 인터 예측에서는 현재 블록과의 레지듀얼(residual) 신호가 최소화되며 움직임 벡터 크기 역시 최소가 되도록 예측 블록을 생성할 수 있다. 이때, 참조 픽처의 정보를 이용하기 위해, 현재 픽처의 주변 블록들의 정보를 이용할 수 있다. 예컨대, 스kip(skip) 모드, 머지(merge) 모드, AMVP(Advanced Motion Vector Prediction) 등을 통해, 주변 블록의 정보에 기반하여 현재 블록에 대한 예측 블록을 생성할 수 있다.
- [0070] 예측 블록은 1/2 화소 샘플 단위와 1/4 화소 샘플 단위와 같이 정수 이하 샘플 단위로 생성될 수도 있다. 이때, 움직임 벡터 역시 정수 화소 이하의 단위로 표현될 수 있다. 예컨대 휙도 화소에 대해서는 1/4 화소 단위로, 색차 화소에 대해서는 1/8 화소 단위로 표현될 수 있다.
- [0071] 현재 블록의 인터 예측에 필요한 움직임 정보, 예컨대 움직임 벡터, 참조 픽처 인덱스 등에 관한 정보는 인코딩 장치로부터 수신한 스kip 플래그, 머지 플래그 등을 확인하고 이에 대응하여 유도될 수 있다.
- [0072] 예측이 수행되는 처리 단위와 예측 방법 및 구체적인 내용이 정해지는 처리 단위는 서로 다를 수도 있다. 예컨대, PU 단위로 예측 모드가 정해져서 PU 단위로 예측이 수행될 수도 있고, PU 단위로 예측 모드가 정해지고 TU 단위로 화면 내 예측이 수행될 수도 있다.
- [0073] 예측부(230)로부터 출력된 예측 블록에 역변환부(225)로부터 출력된 레지듀얼 블록이 더해져서 원본 영상을 복원할 수 있다.
- [0074] 복원된 블록 및/또는 픽처는 필터부(235)로 제공될 수 있다. 필터부(235)는 복원된 블록 및/또는 픽처에 디블록킹 필터링, SAO(Sample Adaptive Offset) 및/또는 적응적 루프 필터링 등을 적용한다.
- [0075] 메모리(240)는 복원된 픽처 또는 블록을 저장하여 참조 픽처 또는 참조 블록으로 사용할 수 있도록 할 수 있고 또한 복원된 픽처를 출력부로 제공할 수 있다.
- [0076] 여기서는 설명의 편의를 위해 생략하였지만, 디코딩 장치에 입력되는 비트 스트림은 파싱 단계를 거쳐 엔트로피 디코딩부로 입력될 수도 있다. 또한, 엔트로피 디코딩부에서 파싱 과정을 수행하도록 할 수도 있다.
- [0077] 본 명세서에서, 코딩은 경우에 따라 인코딩 또는 디코딩으로 해석될 수 있고, 정보(information)는 값(values), 파라미터(parameter), 계수(coefficients), 성분(elements) 등을 모두 포함하는 것으로 이해될 수 있다.
- [0078] ‘화면’ 또는 ‘픽처(picture)’는 일반적으로 특정 시간대의 하나의 영상을 나타내는 단위를 의미하며, ‘슬라이스(slice)’, ‘프레임(frame)’ 등은 실제 비디오 신호의 코딩에 있어서 픽처의 일부를 구성하는 단위이며, 필요에 따라서는 픽처와 서로 혼용되어 사용 될 수도 있다.
- [0079] ‘픽셀(pixel)’, ‘화소’ 또는 ‘pel’은 하나의 영상을 구성하는 최소의 단위를 의미한다. 또한, 특정한 픽셀의 값을 나타내는 용어로서 ‘샘플(sample)’을 사용할 수 있다. 샘플은 휙도(Luma) 및 색차(Chroma) 성분으로 나누어질 수 있으나, 일반적으로는 이를 모두 포함하는 용어로써 사용될 수 있다. 상기에서 색차 성분은 정해진 색상들 간의 차이를 나타내는 것으로 일반적으로 Cb 및 Cr로 구성된다.
- [0080] ‘유닛(unit)’은 상술한 부호화 유닛(CU), 예측 유닛(PU), 변환 유닛(TU)과 같이 영상 처리의 기본 단위 또는 영상의 특정 위치를 지칭하며, 경우에 따라서는 ‘블록(block)’ 또는 ‘영역(area)’ 등의 용어와 서로 혼용하여 사용될 수 있다. 또한, 블록은 M개의 열과 N개의 행으로 구성된 샘플들 또는 변환 계수(transform coefficient)들

의 집합을 나타내는 용어로 사용될 수도 있다.

[0081] 한편, 인터 예측 모드의 경우에 인코딩 장치 또는 디코딩 장치는 현재 블록의 움직임 정보를 도출하고, 도출된 움직임 정보에 기반하여 현재 블록에 대한 인터 예측을 수행할 수 있다.

[0082] 현재 블록의 예측에 이용되는 영상을 참조 픽처(reference picture) 또는 참조 프레임(reference frame)이라고 한다. 참조 픽처 내의 영역은 참조 픽처를 지시하는 참조 픽처 인덱스(refIdx) 및 움직임 벡터(motion vector) 등을 이용하여 나타낼 수 있다.

[0083] 현재 픽처에 대하여, 예측을 위하여 사용되는 픽처들로 참조 픽처 리스트를 구성할 수 있으며, 참조 픽처 인덱스는 참조 픽처 리스트에서 특정 참조 픽처를 지시할 수 있다. P 픽처의 경우에는 하나의 참조 픽처 리스트, 예컨대 참조 리스트 0을 필요로 하며, B 픽처의 경우에는 두 개의 참조 픽처 리스트, 예컨대 참조 리스트 0 및 참조 리스트 1을 필요로 한다.

[0084] 구체적으로, I 픽처는 인트라 예측에 의해서 인코딩/디코딩되는 픽처이다. P 픽처는 각 블록의 샘플 값을 예측하기 위해 적어도 하나의 움직임 벡터 및 참조 픽처 인덱스를 이용한 인터 예측 또는 인트라 예측을 이용해서 인코딩/디코딩될 수 있는 픽처이다. B 픽처는 각 블록의 샘플 값을 예측하기 위해 적어도 두 개의 움직임 벡터들과 참조 픽처 인덱스들을 이용한 인터 예측 또는 인트라 예측을 이용해서 인코딩/디코딩될 수 있는 픽처이다.

[0085] P 픽처에서는 한 개의 참조 픽처 리스트를 필요로 하며, 이를 참조 픽처 리스트 0(reference picture list 0: L0)이라 지칭한다.

[0086] B 픽처는 하나 이상, 예를 들어 2 개의 참조 픽처를 이용하여 순방향, 역방향 또는 양 방향 인터 예측에 의해 부호화될 수 있는 픽처다. B 픽처는 두 개의 참조 픽처 리스트를 필요로 하며, 두 개의 참조 픽처 리스트는 각각 참조 픽처 리스트 0(reference picture list 0: L0), 참조 픽처 리스트 1(reference picture list 1: L1)이라 지칭한다.

[0087] L0으로부터 선택된 참조 픽처를 사용하는 인터 예측이라 하며, L0 예측은 주로 순방향 예측에 사용된다. L1으로부터 선택된 참조 픽처를 사용하는 인터 예측을 L1 예측이라 하며, L1 예측은 주로 역방향 예측에 사용된다. 또한 L0과 L1으로부터 각각 선택된 두 개의 참조 픽처를 사용하는 인터 예측을 쌍 예측(bi prediction)이라고도 한다.

[0088] 상기 I 픽처, P 픽처, B 픽처의 특징은 픽처 단위가 아닌 슬라이스 단위로도 정의될 수 있다. 예컨대, 슬라이스 단위에서 I 픽처의 특징을 갖는 I 슬라이스, P 픽처의 특징을 갖는 P 슬라이스, B 픽처의 특징을 갖는 B 슬라이스가 정의될 수 있다.

[0089] 예컨대, 현재 블록에 대한 슬라이스 타입이 B이고 L0으로부터 colPic이 선택되는 경우 또는 현재 블록에 대한 슬라이스 타입이 P인 경우에 colPic은 L0에서 선택될 수 있다.

[0090] 한편, 지연(low delay)을 방지하기 위하여, GPB(Generalized P and B) 슬라이스를 이용할 수도 있다. GPB에 있어서는 참조 리스트 0과 참조 리스트 1이 동일하게 된다.

[0091] 인터 예측에 있어서, 스킵 모드의 경우에는 주변 블록의 정보를 그대로 현재 블록에 이용할 수 있다. 따라서, 스kip 모드의 경우에는, 현재 블록의 움직임 정보로서 어떤 블록의 움직임 정보를 이용할 것인지를 지시하는 정보 외에 레지듀얼 등과 같은 신택스 정보를 전송하지 않는다.

[0092] 또한, 머지 모드의 경우에는, 주변 블록의 움직임 정보를 그대로 이용하여 현재 블록에 대한 예측 블록을 생성할 수 있다. 인코딩 장치는 머지 모드를 적용하는지에 관한 정보와 어떤 블록의 움직임 정보를 이용할지에 관한 정보, 레지듀얼 정보 등을 전송할 수 있다. 디코딩 장치는 예측 블록과 인코딩 장치로부터 전송되는 레지듀얼을 더하여 현재 블록을 복원할 수 있다.

[0093] 머지 모드에서 어떤 블록의 정보를 이용할 것인지를 지시하는 방법 등을 스kip 모드와 일반적인 인터 예측 모드의 경우에 적용할 수도 있다. 예컨대, 머지 모드에서 현재 블록의 움직임 정보로서 이용될 정보를 가지는 후보 블록을 다른 인터 예측 모드와 공동으로 이용할 수도 있고, 어떤 주변 블록의 움직임 정보를 이용할 것인지를 지시하는 방법을 동일하게 할 수도 있다.

[0094] 이때, 스kip 모드와 일반적인 인터 예측 모드에서는 주변 블록의 참조 픽처 인덱스나 예측 방향(참조 픽처 리스트)이 현재 블록의 참조 픽처 인덱스나 예측 방향(참조 픽처 리스트)에 부합하는 경우에만 이용 가능한 움직임 정보를 가지는 것으로 판단하여 현재 블록에 대한 움직임 벡터 예측자 후보로 해당 블록의 움직임 벡터를 이용

하도록 할 수도 있다. 혹은, 주변 블록의 움직임 벡터를 후술하는 바와 같이 스케일링하여 현재 블록에 대한 움직임 벡터 예측자 후보로 이용하도록 할 수도 있다.

[0095] AMVP의 경우에, 인코딩 장치는 주변 블록들의 움직임 정보를 이용하여 현재 블록의 움직임 벡터를 예측하고, 현재 블록의 움직임 벡터와 예측된 움직임 벡터 사이의 차이를 참조 픽처를 지시하는 참조 픽처 인덱스와 함께 전송할 수 있다. 디코딩 장치는 주변 블록의 움직임 정보들을 이용하여 현재 블록의 움직임 벡터를 예측하고, 인코딩 장치로부터 수신한 레지듀얼을 이용하여 현재 블록에 대한 움직임 벡터를 유도할 수 있다. 디코딩 장치는 유도한 움직임 벡터와 인코딩 장치로부터 수신한 참조 픽처 인덱스 정보를 기반으로 현재 블록에 대한 예측 블록을 생성할 수 있다.

[0096] 도 3은 인터 예측 모드에서 AMVP(Advanced Motion Vector Predictor)가 적용되는 경우, 움직임 벡터를 도출하는 방법에 관한 일 실시예를 개략적으로 설명하는 순서도이다.

[0097] 도 3을 참조하면, 인코딩 장치 또는 디코딩 장치는 현재 블록에 대한 움직임 벡터 예측자(Motion Vector Predictor: MVP) 후보 리스트를 생성할 수 있다(S310). 움직임 벡터 예측자는 현재 블록의 움직임 벡터에 대한 예측값을 나타낼 수 있다.

[0098] 인코딩 장치 또는 디코딩 장치는 현재 블록에 인접하고 이용 가능한(available) 주변 블록(설명의 편의를 위해 '주변 블록'이라 함) 및/또는 현재 블록과 동일 위치에 있는(co-located) 각 참조 픽처의 블록들 중에서 이용 가능한(available) 블록(설명의 편의를 위해 'Col 블록' (co-located block)이라 함)의 움직임 벡터를 이용하여 MVP 후보 리스트를 생성할 수 있다.

[0099] 인코딩 장치 또는 디코딩 장치는 MVP 후보 리스트에 포함된 MVP 후보들 중 현재 블록에 대한 MVP를 선택할 수 있다(S320).

[0100] 인코딩 장치는 MVP 후보 리스트에 포함된 MVP 후보들에 대해 움직임 벡터 경쟁(Motion Vector Competition: MVC)을 적용하여, 현재 블록에 대한 최적의 MVP를 선택할 수 있다. 인코딩 장치는 비트 스트림을 통해 선택된 MVP 인덱스, MVD(Motion Vector Difference, 혹은 Difference of Motion Vector: DMV 라고도 함), 그리고 참조 픽처 인덱스를 디코딩 장치로 전송할 수 있다. MVP 인덱스는, 현재 블록의 MVP를 지시하는 인덱스로서 MVP 후보 리스트에 포함된 MVP 후보들 중에서 어느 하나를 지시할 수 있다. 또한, 참조 픽처 인덱스는, 현재 블록의 참조 픽처를 지시하는 인덱스로서 참조 픽처 리스트에 포함된 참조 픽처 후보들 중에서 어느 하나를 지시할 수 있다. MVD는 현재 블록의 움직임 벡터와 선택된 MVP의 차이로서, MVD의 값이 적을수록 전송되는 정보량을 줄일 수 있다.

[0101] 디코딩 장치는 인코딩 장치로부터 MVP 인덱스와 참조 픽처 인덱스를 수신할 수 있다. 수신한 MVP 인덱스를 이용하여, 디코딩 장치는 MVP 후보 리스트에 포함된 MVP 후보들 중 현재 블록에 대한 MVP를 선택할 수 있다. 수신한 참조 픽처 인덱스를 이용하여, 디코딩 장치는 참조 픽처 리스트에 포함된 참조 픽처 후보들 중에서 현재 블록에 대한 참조 픽처를 선택할 수 있다.

[0102] 디코딩 장치는 선택된 MVP 및 참조 픽처를 수신한 MVD와 함께 이용하여 현재 블록의 움직임 벡터를 도출할 수 있다(S330). 예컨대, 디코딩 장치는 MVP에 수신한 MVD를 더하여 현재 블록의 움직임 벡터를 복원할 수 있다.

[0103] 도 4는 MVP 후보 리스트 생성 방법에 관한 일 실시예를 개략적으로 설명하는 개념도이다.

[0104] 설명의 편의를 위해, 현재 블록(400) 좌하측 코너의 주변 블록인 좌하측 주변 블록  $A_0$ (410) 및 현재 블록(400)의 좌측 주변 블록들  $A_1$ (420)을 포함하는 후보 그룹을 좌측 후보 그룹이라 한다. 이때, 인코딩 장치 또는 디코딩 장치는 좌측 블록들(420) 중에서 특정 위치의 블록만을 이용할 수도 있다. 설정하여 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 좌측 주변 블록들(420) 중 최하단의 블록(420-1)을 좌측 블록  $A_1$ 으로 설정할 수 있다.

[0105] 또한 현재 블록(400) 우상측 코너의 주변 블록인 우상측 주변 블록  $B_0$ (430), 현재 블록(400)의 상측 주변 블록들  $B_1$ (440) 및 현재 블록(400) 좌상측 코너의 주변 블록  $B_2$ (450)을 포함하는 후보 그룹을 상측 후보 그룹이라 한다. 이때, 인코딩 장치 또는 디코딩 장치는 상측 블록들(440) 중 특정 위치의 블록만을 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 상측 주변 블록들 중 최우측의 블록(440-1)을 상측 블록  $B_1$ 으로 설정할 수 있다.

- [0106] 인코딩 장치 또는 디코딩 장치는 좌측 후보 그룹( $A_0, A_1$ )에서 하나의 MVP 후보를 선택할 수 있다. 좌측 후보 그룹에서 선택된 MVP 후보를  $MV_A$ 라 한다. 예컨대, 인코딩 장치 또는 디코딩 장치는 좌측 후보 그룹에 포함된 블록들을  $A_0 \rightarrow A_1$ 의 순서로 스캔하면서, 현재 블록과의 참조 픽처 인덱스 동일 여부, 참조 픽처 리스트 동일 여부 등을 판단하여, 이용 가능한 첫 번째 블록의 움직임 벡터를 현재 블록의 MVP 후보( $MV_A$ )로 선택할 수 있다.
- [0107] 인코딩 장치 또는 디코딩 장치는 또한, 상측 후보 그룹( $B_0, B_1, B_2$ )에서 하나의 MVP 후보를 선택할 수 있다. 상측 후보 그룹에서 선택된 MVP 후보를  $MV_B$ 라 한다. 예를 들어, 인코딩 장치 또는 디코딩 장치는 상측 후보 그룹에 포함된 블록들을  $B_0 \rightarrow B_1 \rightarrow B_2$ 의 순서로 스캔하면서, 현재 블록과의 참조 픽처 인덱스 동일 여부, 참조 픽처 리스트 동일 여부 등을 판단하여, 이용 가능한 첫 번째 블록의 움직임 벡터를 현재 블록의 MVP 후보( $MV_B$ )로 선택할 수 있다.
- [0108] 또한, 인코딩 장치 또는 디코딩 장치는 참조 픽처 내의 Col 블록 COL(460)의 움직임 벡터를 현재 블록의 MVP 후보로 선택할 수도 있다. 여기서, Col 블록(660)의 움직임 벡터는  $mvCol$ 로 표시할 수 있다. 이때, 인코딩 장치 또는 디코딩 장치는  $MV_A$ 와  $MV_B$ 를 검색하는 과정에서 필요한 수의 공간적 후보를 확보하지 못한 경우에 Col 블록의 움직임 벡터를 후보로 설정하도록 할 수도 있다.
- [0109] 상술한 실시예에서, 좌측 후보 그룹에서 선택된 MVP 후보 및 상측 후보 그룹에서 선택된 MVP 후보는 공간적 MVP(Spatial Motion Vector Predictor: SMVP)라고 할 수 있다. 또한, 참조 픽처 내의 Col 블록을 이용하여 선택된 MVP 후보는 시간적 MVP(Temporal Motion Vector Predictor: TMVP)라고 할 수 있다. 따라서, MVP 후보 리스트는 SMVP 및/또는 TMVP를 포함할 수 있다.
- [0110] 인코딩 장치 또는 디코딩 장치는 상술한 방법에 의해 선택된 MVP 후보들 중 중복되는 후보가 있다면, 중복되는 후보들 중에서 가장 높은 순위의 후보를 제외한 나머지 후보를 제거할 수 있다.
- [0111] 여기서는, 좌측 후보 그룹( $A_0, A_1$ ), 상측 후보 그룹( $B_0, B_1, B_2$ )로 나누어 설명하였으나, 후보 그룹은 이에 한정되지 않고 설정할 수도 있다. 예컨대, 좌측 후보 그룹에  $A_1$ 만을 두고, 상측 후보 그룹에  $B_1$ 만을 두며,  $A_0, B_0, B_2$ 를 포함하는 코너 후보 그룹을 구성할 수도 있다. 좌측 후보 그룹, 상측 후보 그룹, 코너 후보 그룹으로 나누는 경우에도, 그룹별로 현재 블록과의 참조 픽처 인덱스 동일 여부, 참조 픽처 리스트 동일 여부 등을 판단하여 현재 블록의 MVP 후보를 소정 개수 선택할 수도 있다.
- [0112] 이때, 더 정확하게 현재 블록의 움직임 벡터를 예측하기 위해서, 주변 블록으로부터 선택되는 움직임 벡터(이웃 움직임 벡터)를 스케일링(scaling)할 수 있다.
- [0113] 예를 들어, 현재 블록의 움직임 벡터  $MV_C$ 의 참조 프레임(참조 픽처)에 대응하는 참조 프레임(참조 픽처)를 가지는 주변 블록이 없는 경우에, 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자(Predictor of Motion Vector: PMV 혹은 Motion Vector Predictor: MVP)를 생성할 수 있다.
- [0114] 도 5는 본 발명에 따른 스케일링의 방법을 설명하기 위한 도면으로서, 현재 블록과 주변 블록의 관계를 개략적으로 도시하고 있다. 도 5에서는 설명의 편의를 위해 현재 블록(510)의 블록들에 대하여 좌측 주변 블록과 상측 주변 블록 그리고 코너 주변 블록 중 각각 하나씩을 고려하여 움직임 벡터를 스케일링 하는 방법의 일 예를 설명한다. 도 5에서는 코너 블록들 중 현재 블록 상측의 우측 코너 블록을 주변 코너 블록들을 대표하는 블록으로서 설명한다. 도 5에 관하여 이하 설명하는 내용은 다른 주변 코너 블록들에 대해서도 동일하게 적용될 수 있다.
- [0115] 도 5를 참조하면, 현재 블록(510)의 상측(upper) 블록(520), 상우측(upper-right) 블록(530), 좌측(left) 블록(540)에 대하여,  $MV_C$ 는 현재 블록(510)의 움직임 벡터이며,  $MV_U$ 는 상측 블록(520)의 움직임 벡터이고,  $MV_{UR}$ 은 상우측 블록(530)의 움직임 벡터이며,  $MV_L$ 은 좌측 블록(540)의 움직임 벡터이다. 또한, 현재 블록(510)의 상측 블록(520)의 참조 픽처 인덱스는 1(Ref. #1)이며, 현재 블록(510)의 우상측 블록(530)의 참조 픽처 인덱스도 1(Ref. #1)이고, 현재 블록(510)의 좌측 블록(540)의 참조 픽처 인덱스는 0(Ref. #0)이라고 하자. 참조 픽처 인덱스는 각 블록의 움직임 벡터가 가리키는 참조 프레임(참조 픽처) 인덱스를 의미한다.
- [0116] 도 5에서, 메모리(버퍼)에 저장된, 참조 픽처 리스트에서 0번째 참조 프레임(참조 픽처)이 움직임 추정(motion

estimation)에 이용되면, 움직임 벡터  $MV_L$ 이 0번째 참조 프레임의 움직임 벡터 예측자를 생성하는데 이용될 수 있다. 또한, 참조 픽처 리스트에서 1번째 참조 프레임이 움직임 추정에 이용되면, 세 움직임 벡터( $MV_L$ ,  $MV_U$ ,  $MV_{UL}$ )가 모두 1번째 참조 프레임의 움직임 벡터 예측자를 생성하는데 이용될 수 있다. 참조 픽처 리스트에서 2번째 내지 4번째 참조 프레임에 대한 움직임 벡터 예측자들은 1번째 참조 프레임의 움직임 벡터 예측자와 동일한 값을 가질 수 있다.

[0117] 생성된 움직임 벡터 예측자를 기반으로, 현재 블록에 대하여 추정된 움직임 벡터  $MV_C$ 와 움직임 벡터 예측자의 차이값이 산출될 수 있으며, MVD는 디코딩 장치로 전송된다.

[0118] 도 5의 예에서, 주변 블록의 참조 프레임(참조 픽처)과 동일한 참조 프레임인, 참조 픽처 리스트의 0번째 또는 1번째 참조 프레임이 움직임 추정에 이용된다면, 적정한 움직임 벡터 예측자를 생성할 수 있다. 하지만, 주변 블록의 참조 프레임(참조 픽처)과 다른 참조 프레임이 움직임 추정에 이용된다면, 생성되는 움직임 벡터 예측자는 현재 블록의 움직임 벡터  $MV_C$ 와 큰 차이를 보일 수 있다. 왜냐하면, 현재 블록의 움직임 벡터  $MV_C$ 에 대한 참조 프레임에 대응하는 참조 프레임을 가지는 주변 블록이 없기 때문에, 주변 블록의 정보를 이용하여 현재 블록의 움직임 벡터를 효과적으로 예측하기 어렵기 때문이다.

[0119] 이 경우에, 현재 블록의 주변 블록들이 가지는 움직임 벡터들에 대한 상향 스케일링 또는 하향 스케일링을 통하여, 현재 블록의 움직임 벡터  $MV_C$ 에 더 근접한 움직임 벡터 예측자를 생성할 수 있고, 부호화되어 전송되는 MVD의 비트량을 줄일 수 있다.

[0120] 예를 들면, 현재 블록(510)의 주변 블록인 상측 블록(520), 상우측 블록(530), 좌측 블록(540)의 참조 픽처가 모두 현재 블록의 참조 픽처와 다르다고 가정할 때, 주변 블록들(520, 530, 540)의 움직임 벡터들은 수식 1의 예와 같이, 선형 인터플레이션(linear interpolation)에 의해 상향 스케일링되거나(up-scaled), 하향 스케일링될(down-scaled) 수 있다.

[0121] <수식 1>

$$MV_L' = \frac{f_{ME} - f_C}{f_L - f_C} \times MV_L$$

$$MV_U' = \frac{f_{ME} - f_C}{f_U - f_C} \times MV_U$$

$$MV_{UR}' = \frac{f_{ME} - f_C}{f_{UR} - f_C} \times MV_{UR}$$

[0122]

[0123] 수식 1에서,  $f_L$ 은 현재 블록의 좌측 블록의 참조 프레임(또는 참조 픽처 인덱스)이고,  $f_U$ 는 현재 블록의 상측 블록의 참조 프레임 넘버이며,  $f_{UR}$ 은 코너 블록들 중 현재 블록의 우상측 블록의 참조 프레임 넘버이고,  $f_{ME}$ 는 움직임 추정된 참조 프레임 넘버이며,  $f_C$ 는 현재 프레임 넘버이다.

[0124] 수식 1과 같은 선형 인터플레이션에 의해서, 현재 블록의 주변 블록들의 움직임 벡터들이 상향 스케일링 또는 하향 스케일링될 수 있다. 스케일링된 주변 블록들의 움직임 벡터들을 이용하여 움직임 벡터 예측자를 생성할 수 있다. 이때, 부호화 장치 또는 복호화 장치는 다양한 방법으로 움직임 벡터 예측자를 생성할 수 있다. 예컨대, 주변 블록들의 움직임 벡터들 중 어느 하나를 선택하여 움직임 벡터 예측자로 이용할 수도 있고, 주변 블록들의 움직임 벡터들을 평균한 값을 움직임 벡터 예측자로 이용할 수도 있으며, 주변 블록들의 움직임 벡터들을 미디언 필터링한 값을 움직임 벡터 예측자로 이용할 수도 있다.

[0125] 도 6은 본 발명에 따른 스케일링의 방법을 개략적으로 설명하는 순서도이다.

[0126] 도 6을 참조하면, 인코딩 장치 또는 디코딩 장치는 n 번째 참조 프레임을 이용하여 움직임 추정을 수행한다 (S610). n번째 참조 프레임은 참조 픽처 리스트에서 n 번째 참조 픽처(참조 프레임)으로서, 현재 블록의 참조

프레임(참조 꾹처)일 수 있다.

[0127] 인코딩 장치 또는 디코딩 장치는 주변 블록의 참조 프레임 넘버(참조 꾹처의 경우에는 참조 꾹처 인덱스)가 n과 상이한지를 판단한다(S620). 예컨대, 주변 블록으로서 현재 블록의 좌측 블록, 현재 블록의 상측 블록 그리고 코너 블록들 중에서 현재 블록의 상우측 코너 블록을 고려하는 경우에, 인코딩 장치 또는 디코딩 장치는 좌측 블록의 참조 프레임 넘버  $f_L$ , 상측 블록의 참조 프레임 넘버  $f_U$ , 상우측 블록의 참조 프레임 넘버  $f_{UR}$ 의 값들이 n과 상이한지를 판단한다.

[0128] 인코딩 장치 또는 디코딩 장치는 주변 블록의 참조 프레임 넘버가 n과 상이할 때, 주변 블록들의 움직임 벡터를 스케일링할 수 있다(S630). 예컨대, 상술한 바와 같이 현재 블록의 좌측, 상측, 상우측 블록을 주변 블록으로서 고려할 때, 인코딩 장치 또는 디코딩 장치는  $f_L$ ,  $f_U$ ,  $f_{UR}$ 의 값이 n과 상이한 경우에, 대응하는 블록의 움직임 벡터 즉, 현재 블록의 좌측 블록의 움직임 벡터  $MV_L$ , 현재 블록의 상측 블록의 움직임 벡터  $MV_U$ , 현재 블록의 상우측 블록의 움직임 벡터  $MV_{UR}$ 을 스케일링할 수 있다.

[0129] 인코딩 장치 또는 디코딩 장치는 현재 블록에 대한 움직임 벡터 예측자를 생성한다(S640). 인코딩 장치 또는 디코딩 장치는 주변 블록들 중 어느 하나가 현재 블록의 참조 프레임과 동일한 참조 프레임 넘버를 가지는 경우에는 해당 블록의 움직임 벡터를 이용하여 현재 블록에 대한 움직임 벡터 예측자를 생성할 수 있다. 예컨대,  $f_L$ ,  $f_U$ ,  $f_{UR}$ 의 값 중 적어도 하나가 n의 값을 갖는 경우에는, n의 참조 프레임 넘버를 가지는 주변 블록의 움직임 벡터를 이용하여 현재 블록에 대한 움직임 벡터 예측자를 생성할 수 있다. 또한, 인코딩 장치 또는 디코딩 장치는  $f_L$ ,  $f_U$ ,  $f_{UR}$ 의 값들이 모두 n과 상이한 값을 갖는 경우에는, 주변 블록의 움직임 벡터를 상술한 바와 같이 스케일링해서 현재 블록에 대한 움직임 벡터 예측자를 생성할 수 있다. 스케일링 방법으로는 상술한 수식 1을 이용할 수 있다.

[0130] 이때, 인코딩 장치 또는 디코딩 장치는 주변 블록의 스케일링된 움직임 벡터들 혹은 스케일링되지 않은 움직임 벡터들에 대하여 미디언 값을 구해서 현재 블록에 대한 움직임 벡터 예측자를 생성할 수도 있다. 혹은 인코딩 장치 또는 디코딩 장치는 주변 블록의 스케일링된 움직임 벡터들 혹은 스케일링되지 않은 움직임 벡터들에 대하여 평균값을 구해서 현재 블록에 대한 움직임 벡터 예측자를 생성할 수도 있다. 혹은 인코딩 장치 또는 디코딩 장치는 주변 블록의 스케일링된 움직임 벡터들 혹은 스케일링되지 않은 움직임 벡터들 중 어느 하나를 현재 블록에 대한 움직임 벡터 예측자로 이용할 수도 있다.

[0131] 여기서는 설명의 편의를 위해, 현재 블록의 주변 블록으로서, 현재 블록의 좌측에 위치하는 주변 블록들을 대표하는 좌측 블록(520), 현재 블록의 상측에 위치하는 주변 블록들을 대표하는 상측 블록(530), 현재 블록의 주변 코너에 위치하는 블록들을 대표하는 우상측 블록(540)을 예로 들었으나, 주변 블록을 모두 고려하거나, 주변 블록들 중 다른 위치의 블록을 고려하여 상술한 방법을 적용할 수도 있다.

[0132] 예컨대, 도 4와 같이, 코너 블록 3개(410, 430, 450), 좌측 블록들(420) 중 어느 하나 그리고 상측 블록들(440) 중 어느 하나를 이용하여 움직임 벡터 예측자를 생성할 수도 있다. 이 경우에, 좌측 블록들(420) 중에서 특정 위치의 블록(420-1)을 이용하거나 상측 블록들(440) 중에서 특정 위치의 블록(440-1)을 이용할 수도 있다.

[0133] 또한, 현재 블록의 참조 프레임(참조 꾹처)과 동일한 참조 프레임을 가지는 블록이 있는지를 좌측 후보 그룹(A0, A1)에 대해서 판단하고, 상측 후보 그룹(B0, B1, B2)에 대하여 판단해서, 하나씩의 후보들을 선택할 수도 있다. 좌측 후보 그룹의 블록들 중에 현재 블록의 참조 프레임(참조 꾹처)과 동일한 참조 프레임(꼭처)을 가지는 블록이 없는 경우에는 좌측 후보 블록들 또는 좌측 후보 블록들 중에서 선택된 블록의 움직임 벡터를 스케일링할 수 있다. 또한, 상측 후보 그룹의 블록들 중에 현재 블록의 참조 프레임(참조 꾹처)과 동일한 참조 프레임(꼭처)을 가지는 블록이 없는 경우에는 상측 후보 블록들 또는 상측 후보 블록들 중 선택된 블록의 움직임 벡터를 스케일링 할 수 있다.

[0134] 도 7은 본 발명에 따라서, 현재 꾹처(현재 프레임) 내 현재 블록의 주변 블록이 가지는 움직임 벡터를 스케일링하는 방법을 개략적으로 설명하는 도면이다.

[0135] 도 7을 참조하면, 현재 꾹처(700) 내의 현재 블록(740)의 주변 블록(730)의 참조 꾹처는 현재 블록(740)의 참조 꾹처와 동일할 수도 있고, 동일하지 않을 수도 있다.

[0136] 현재 블록(740)의 참조 꾹처와 주변 블록(730)의 참조 꾹처가 참조 꾹처(710)로 동일한 경우, 주변 블록(730)의 움직임 벡터(750)는 스케일링 없이 현재 블록의 움직임 벡터 예측자를 생성하는데 이용될 수 있다. 이때, 주변

[0137] 한편, 현재 블록(740)의 참조 픽처가 참조 픽처(710)이고 주변 블록(730)의 참조 픽처가 참조 픽처(720)로서 서로 상이한 경우라면, 주변 블록(730)의 움직임 벡터(760)을 스케일링해서 현재 블록의 움직임 벡터 예측자를 생성하는데 이용될 수 있다. 이때, 주변 블록(730)은 현재 블록(740)과 동일한 참조 픽처 리스트를 가질 수도 있고, 다른 참조 픽처 리스트를 가질 수도 있다. 예컨대, 현재 블록(740)의 참조 픽처가 참조 리스트 L0의 N번째 참조 픽처이고, 주변 블록(730)이 현재 블록(740)과 상이한 참조 픽처를 가지되 참조 픽처 리스트는 동일하다면, 주변 블록(730)의 움직임 벡터(760)를 스케일링한 움직임 벡터를 mvLON\_scaled로 표시될 수 있다.

[0138] 주변 블록의 움직임 벡터는 현재 블록의 참조 픽처와 현재 픽처 사이의 거리, tb 그리고 주변 블록의 참조 픽처와 현재 픽처 사이의 거리, td의 크기를 고려하여 수행될 수 있다. 주변 블록의 움직임 벡터를 스케일링하는 방법의 일 예로서, 수식 2의 방법을 이용할 수 있다.

[0139] <수식 2>

[0140]  $mvLON\_scaled = (DistScaleFactor * mvLON + 128) \gg 8$

[0141] 여기서, DistScaleFactor는 상술한 바와 같이, 두 거리 tb와 td의 크기를 고려하여 결정되는 스케일링 팩터이다. 이때, 반올림의 경우에는 수식 2와 같이 상수 128을 이용하고, 반내림의 경우에는 상수 128 대신 상수 127을 이용할 수도 있다. 예컨대, 쿼터 펠 블록(Quarter Pel Unit, 1/4 화소 블록)에서 스케일링을 수행한 결과가 1.5라면, 상수 128을 적용하는 경우에는 2로, 상수 127을 적용하는 경우에는 1로 라운딩되는 결과를 얻을 수 있다. 다만, 반내림의 경우, 0을 지향하는 반내림을 적용할 수도 있다. 예컨대, -1.5의 값에 대하여 반내림을 적용하는 경우에, -2가 아니라 -1의 값이 되도록 반내림을 적용할 수도 있다.

[0142] DistScaleFactor는 다양한 방법으로 결정될 수 있는데, 수식 3 및 수식 4는 DistScaleFactor를 결정하는 서로 다른 방법의 예들이다.

[0143] <수식 3>

[0144]  $DistScaleFactor = Clip3(-1024, 1023, (tb*tx + 32) \gg 6)$

[0145] <수식 4>

[0146]  $DistScaleFactor = Clip3(-4096, 4095, (tb*tx + 32) \gg 6)$

[0147] 이때, 수식 3과 4의 tx는 H.264/MPEG-4 AVC의 시간적 움직임 벡터 후보 유도에서와 같이  $tx = (16384 + (Abs(td)/2))/td$ 로 설정할 수 있다(H.264 표준 8.4.1.2.3 참조).

[0148] 한편, 움직임 벡터 예측자를 생성하는데 이용하는 주변 블록들은 움직임 벡터 경쟁(Motion Vector Competition)을 통해서 선택될 수 있다. 움직임 벡터 경쟁은 여러 개의 움직임 벡터 후보들 중에서 가장 적합한 움직임 벡터를 선택하여 예측값으로 사용하는 것을 말한다. 이 경우에는 현재 블록의 공간적(Spatial) 이웃 블록뿐만 아니라, 시간적(temporal) 이웃 블록인 col 블록까지 고려될 수 있다.

[0149] 도 8은 본 발명의 일 실시예에 따라서 움직임 벡터 경쟁을 수행하는 방법을 개략적으로 설명하는 도면이다. 도 8에서는, 현재 블록(800)의 주변 블록들 중에서 공간적 주변 블록들을 도시하고 있다.

[0150] 도 8을 참조하면, 도 8의 공간적 주변 블록들은, 현재 블록(800)의 상측에 있는 na (na는 1 이상의 정수) 개의 상측 그룹 블록들( $a_0, a_1, \dots, a_{na}$ )과 현재 블록(800)의 좌측에 있는 nb (nb는 1 이상의 정수) 개의 좌측 그룹 블록들( $b_0, b_1, \dots, b_{nb}$ ) 그리고 현재 블록(800)의 우상측, 좌하측, 좌상측 코너에 있는 코너 그룹 블록들( $c, d, e$ )를 포함한다.

[0151] 도 8의 예에서, AMVP(Advanced Motion Vector Prediction)에서 움직임 벡터 후보들은 수식 5의 예와 같이 구할 수 있다.

[0152] <수식 5>

[0153] 움직임 벡터 후보(motion vector candidates) = {median( $a', b', c'$ ),  $a', b', c'$ , 시간적 예측자}

(temporal predictor)}

[0154] 수식 5에서 세 개의 공간적 움직임 벡터 ( $a'$  ,  $b'$  ,  $c'$  )은 각 그룹(현재 블록의 상측에 있는 블록들, 현재 블록의 좌측에 있는 블록들, 현재 블록의 코너에 있는 블록들)에서 현재 블록(800)과 동일한 참조 리스트, 동일한 참조 픽처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터를 의미한다. 인코딩 장치 또는 디코딩 장치는 수식 5의 움직임 벡터 후보들 전부 또는 일부를 이용하여 도 3에서 설명한 움직임 벡터 예측자 후보 리스트를 생성할 수 있다.

[0155] 인코딩 장치 또는 디코딩 장치는 수식 5의 움직임 벡터 후보들 중에서 소정의 후보를 현재 블록의 움직임 벡터 예측자로 선정할 수 있다. 이때, 인코딩 장치 또는 디코딩 장치는 수식 5의 움직임 벡터들 중 일부 또는 전부를 고려하여 현재 블록의 MVP를 선택할 수 있다. 예컨대, 인코딩 장치는 수식 5의 움직임 벡터 후보들 중 현재 블록의 움직임 벡터와의 레지듀얼이 최소가 되는 후보를 현재 블록에 대한 MVP로 선택하고, 이에 관한 정보를 디코딩 장치에 전송할 수도 있다. 또한, 인코딩 장치는 수식 5의 움직임 벡터 중 일부의 움직임 벡터, 예컨대  $a'$  ,  $b'$  ,  $c'$  만을 고려하여 이용 가능한 움직임 벡터를 선택하고,  $a'$  ,  $b'$  ,  $c'$  가 고려의 대상이 될 수 없는 경우(이용 가능하지 않은 경우)에는 추가로 시간적 예측자를 고려할 수도 있다.

[0156] 수식 5에서  $\text{median}(a' , b' , c')$ 는  $a'$  ,  $b'$  ,  $c'$  의 미디안(median) 벡터로서, 미디안 벡터를 산출하는 방법의 일 예로서 아래의 1) 내지 4)를 이용할 수 있다.

[0157] 1)  $a'$  ,  $b'$  ,  $c'$  가 모두 이용 가능(available)한 경우에는 그대로  $a'$  ,  $b'$  ,  $c'$  의 값을 계산하여 미디안 벡터를 산출한다.

[0158] 2)  $a'$  ,  $b'$  ,  $c'$  중 2개만 이용 가능한 경우에는, 이용할 수 없는(not available) 움직임 벡터의 값을 0으로 설정해서 미디언 벡터를 산출한다.

[0159] 3)  $a'$  ,  $b'$  ,  $c'$  중 1개만 이용 가능한 경우에는, 이용 가능한 움직임 벡터를 미디언 벡터로 이용한다.

[0160] 4) 1) 내지 3) 이외의 경우에는 미디언 벡터의 값을 0으로 설정한다.

[0161] 이때,  $a'$  ,  $b'$  ,  $c'$  각각의 움직임 벡터 중 이용 가능하지 않은 움직임 벡터는 움직임 벡터 후보에 포함되지 않는다. 또한, 시간적 움직임 벡터 예측자는 현재 픽처(프레임)과는 다른 픽처(프레임)에서 현재 블록에 대한 하나의 대응 위치(co-located) 움직임 벡터를 의미한다.

[0162] 한편, 현재 블록(800)의 상측에 있는  $na$  ( $na$ 는 1 이상의 정수) 개의 상측 그룹 블록들( $a_0$ ,  $a_1$ , ...,  $a_{na}$ ) 중에서 현재 블록(800)과 동일한 참조 픽처 리스트, 동일한 참조 픽처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터를 구하기 위한 수도 코드(pseudo code)의 일 예는 표 1과 같다.

## 표 1

```

a' = not available

For (i=0 ; i<na ; i++) {
    if ( i번째 block이 inter mode이고 현재 블록과 같은 list의 같은
        ref_idx를 참조하는가?) {
        a' = ai
        break;
    }
}

```

[0163]

[0164] 표 1에서는 검색의 순서를  $a_0$ 에서부터  $a_{na}$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $a_{na}$ 에서부터  $a_0$ 쪽으로 검색하여 현재 블록(800)과 동일한 참조 픽처 리스트, 동일한 참조 픽처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터를 구해낼 수도 있다.

[0165] 한편, 표 1의 예와 달리, 참조 픽처 리스트와 참조 픽처 인덱스에 상관없이, 상측 그룹 블록들 중에서 첫 번째

로 검색되는 인터 모드 블록의 움직임 벡터를 스케일링해서  $a'$ 의 값으로 할당할 수도 있다. 이 경우에 움직임 벡터  $a'$ 을 구하기 위한 수도 코드의 일 예는 표 2와 같다.

## 표 2

```

a' = not available

For (i=0 ; i<na ; i++) {
    if ( i번째 block이 inter mode) {
        a' = scale ( ai )
        break;
    }
}

```

[0166]

표 2에서도 검색의 순서를  $a_0$ 에서부터  $a_{na}$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $a_{na}$ 에서부터  $a_0$ 쪽으로 검색하여 현재 블록(800)과 동일한 참조 픽처 리스트, 동일한 참조 픽처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터를 스케일링해서 구해낼 수도 있다.

[0168]

현재 블록(800)의 좌측에 있는  $nb$  ( $nb$ 는 1 이상의 정수) 개의 좌측 그룹 블록들( $b_0, b_1, \dots, b_{nb}$ ) 중에서 현재 블록(800)과 동일한 참조 픽처 리스트, 동일한 참조 픽처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터  $b'$  도 표 1과 유사하게 얻을 수 있다.  $b'$ 를 구하기 위한 수도 코드(pseudo code)의 일 예는 표 3과 같다.

## 표 3

```

b' = not available

For (i=0 ; i<nb ; i++) {
    if ( i번째 block이 inter mode이고 현재 블록과 같은 list의 같은
        ref_idx를 참조하는가?) {
        b' = bi
        break;
    }
}

```

[0169]

표 3에서는 검색의 순서를  $b_0$ 에서부터  $b_{nb}$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $b_{nb}$ 에서부터  $b_0$ 쪽으로 검색하여 현재 블록(800)과 동일한 참조 픽처 리스트, 동일한 참조 픽처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터를 구해낼 수도 있다.

[0171]

한편, 표 3의 예와 달리, 참조 픽처 리스트와 참조 픽처 인덱스에 상관없이, 좌측 그룹 블록들 중에서 첫 번째로 검색되는 인터 모드 블록의 움직임 벡터를 스케일링해서  $b'$ 의 값으로 할당할 수도 있다. 이 경우에 움직임 벡터  $b'$ 을 구하기 위한 수도 코드의 일 예는 표 4와 같다.

## 표 4

```

b' = not available
For (i=0 ; i<nb ; i++) {
    if ( i번째 block이 inter mode) {
        b' = scale ( bi )
        break;
    }
}
}

```

[0172]

표 4에서도 검색의 순서를  $b_0$ 에서부터  $b_{nb}$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $b_{nb}$ 에서부터  $b_0$ 쪽으로 검색하여 현재 블록(800)과 동일한 참조 퍽처 리스트, 동일한 참조 퍽처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터를 스케일링해서 구해낼 수도 있다.

[0173]

현재 블록(800)의 코너에 있는 3 개의 코너 그룹 블록들(c, d, e) 중에서 현재 블록(800)과 동일한 참조 퍽처 리스트, 동일한 참조 퍽처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터  $c'$  도 표 1과 유사하게 얻을 수 있다.  $c'$  를 구하기 위한 수도 코드(pseudo code)의 일 예는 표 5와 같다. 표 5에서는 설명의 편의를 위해,  $c_0=c$ ,  $c_1=d$ ,  $c_2=e$ 로 기재한다.

## 표 5

```

c' = not available
For (i=0 ; i<3 ; i++) {
    if ( i번째 block이 inter mode이고 현재 블록과 같은 list의 같은
        ref_idx를 참조하는가?) {
        c' = ci
        break;
    }
}

```

[0175]

표 5에서는 검색의 순서를  $c_0$ 에서부터  $c_2$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $c_2$ 에서부터  $c_0$ 쪽으로 검색하거나,  $c_1 \rightarrow c_2 \rightarrow c_0$  또는  $c_1 \rightarrow c_0 \rightarrow c_2$  순으로 검색하여 현재 블록(800)과 동일한 참조 퍽처 리스트, 동일한 참조 퍽처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터를 구해낼 수도 있다.

[0176]

한편, 표 5의 예와 달리, 참조 퍽처 리스트와 참조 퍽처 인덱스에 상관없이, 코너 그룹 블록들 중에서 첫 번째로 검색되는 인터 모드 블록의 움직임 벡터를 스케일링해서  $c'$  의 값으로 할당할 수도 있다. 이 경우에 움직임 벡터  $c'$  을 구하기 위한 수도 코드의 일 예는 표 6와 같다.

## 표 6

```

c' = not available

For (i=0 ; i<3 ; i++) {
    if ( i번째 block이 inter mode) {
        c' = scale ( ci )
        break;
    }
}

```

[0178]

표 6에서도 검색의 순서를  $c_0$ 에서부터  $c_2$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $c_2$ 에서부터  $c_0$ 쪽으로 검색하거나,  $c_1 \rightarrow c_2 \rightarrow c_0$  또는  $c_1 \rightarrow c_0 \rightarrow c_2$  순으로 검색하여 현재 블록(800)과 동일한 참조 피처 리스트, 동일한 참조 피처 인덱스를 가지는 첫 번째 이용 가능한 블록의 움직임 벡터를 스케일링해서 구해낼 수도 있다.

[0179]

한편, 움직임 벡터가 이용 가능하지 않은 경우에만, 참조 피처 리스트, 참조 피처 인덱스에 상관없이 해당 주변 블록 그룹 내에서 첫 번째 검색되는 인터 모드 블록의 움직임 벡터를 스케일링 해서 움직임 벡터 후보로 할당할 수도 있다.

[0180]

표 1 내지 6에서는 현재 블록과 참조 피처 및 참조 피처 리스트가 같은 블록을 이용 가능한 블록으로 설정한다. 다만, 표 2, 4, 6에서는, 현재 블록과 참조 피처 및 참조 피처 리스트가 상이한 경우에도, 인터 예측 모드의 블록인 경우에 스케일링에 의해 해당 블록의 움직임 벡터를 이용할 수 있다.

[0181]

이하 표 7 내지 표 12에서도, 현재 블록과 참조 피처 및 참조 피처 리스트가 상이한 경우에, 인터 예측 모드의 블록인 경우에 스케일링에 의해 해당 블록의 움직임 벡터를 이용하는 것을 설명하고 있다.

[0182]

표 7은 상측 그룹 블록들의 움직임 벡터가 이용 가능하지 않은 경우에, 상측 그룹 블록들 중에서 첫 번째로 검색되는 인터 모드 블록의 움직임 벡터를 스케일링하여  $a'$  을 구해내기 위한 수도 코드의 일 예이다.

## 표 7

```

a' = not available

For (i=0 ; i<na ; i++) {
    if ( i번째 block이 inter mode이고 현재 블록과 같은 list의 같은
        ref_idx를 참조하는가? ) {
        a' = ai
        break;
    }
}

If (a' is not available) {
    For (i=0 ; i<na ; i++) {
        if ( i번째 block이 inter mode ) {
            a' = scale ( ai )
            break;
        }
    }
}

```

[0184]

[0185] 표 7에서는 검색의 순서를  $a_0$ 에서부터  $a_{na}$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $a_{na}$ 에서부터  $a_0$ 쪽으로 검색하여 첫 번째 이용 가능한 블록의 움직임 벡터를 스케일링해서 구해낼 수도 있다.

[0186] 표 8은 좌측 그룹 블록들의 움직임 벡터가 이용 가능하지 않은 경우에, 좌측 그룹 블록들 중에서 첫 번째로 검색되는 인터 모드 블록의 움직임 벡터를 스케일링하여  $b'$ 을 구해내기 위한 수도 코드의 일 예이다.

## 표 8

```

b' = not available
For (i=0 ; i<nb ; i++) {
    if ( i번째 block이 inter mode이고 현재 블록과 같은 list의 같은
        ref_idx를 참조하는가? ) {
        b' = bi
        break;
    }
}
If (a' is not available) {
    For (i=0 ; i<nb ; i++) {
        if ( i번째 block이 inter mode) {
            b' = scale ( bi )
            break;
        }
    }
}

```

[0187]

[0188] 표 8에서도 검색의 순서를  $b_0$ 에서부터  $b_{nb}$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $b_{nb}$ 에서부터  $b_0$ 쪽으로 검색하여 첫 번째 이용 가능한 블록의 움직임 벡터를 스케일링해서 구해낼 수도 있다.

[0189]

표 9는 좌측 그룹 블록들의 움직임 벡터가 이용 가능하지 않은 경우에, 좌측 그룹 블록들 중에서 첫 번째로 검색되는 인터 모드 블록의 움직임 벡터를 스케일링하여  $c'$  을 구해내기 위한 수도 코드의 일 예이다.

## 표 9

```

c' = not available

For (i=0 ; i<3 ; i++) {
    if ( i번째 block이 inter mode이고 현재 블록과 같은 list의 같은
        ref_idx를 참조하는가?) {
        c' = ci
        break;
    }
}

If (a' is not available) {
    For (i=0 ; i<3 ; i++) {
        if ( i번째 block이 inter mode) {
            c' = scale ( ci )
            break;
        }
    }
}

```

[0190]

[0191] 표 9에서도 검색의 순서를  $c_0$ 에서부터  $c_2$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $c_2$ 에서부터  $c_0$ 쪽으로 검색하거나,  $c_1 \rightarrow c_2 \rightarrow c_0$  또는  $c_1 \rightarrow c_0 \rightarrow c_2$  순으로 검색하여 첫 번째 이용 가능한 블록의 움직임 벡터를 스케일링해서 구해낼 수도 있다.

[0192] 한편, 상술한 방법들과 유사하지만, 현재 블록의 주변 블록들의 움직임 벡터가 이용 가능하지 않은 경우에, 아래와 같은 우선 순위를 두고 스케일링의 대상이 되는 움직임 벡터를 결정할 수도 있다.

[0193] <우선 순위>

[0194] 1) 현재 블록과 동일한 참조 꽈치를 참조하는 블록 (리스트 0(L0)과 리스트 1(L1)이 동일한 참조 꽈치들을 가지는 경우에 발생할 수 있다.)

[0195] 2) 1)의 검토 후에도, 주변 블록들의 움직임 벡터들 중에서 이용 가능한 움직임 벡터가 없는 경우에는, 같은 참조 꽈치 리스트를 가지는 블록

[0196] 3) 2)의 검토 후에도, 주변 블록들의 움직임 벡터들 중에서 이용 가능한 움직임 벡터가 없는 경우에는, 나머지 인터 모드 블록

[0197] 따라서, 인코딩 장치 또는 디코딩 장치는 1) 내지 3)의 우선 순위를 두고 현재 블록의 주변 블록 중 어느 한 블록을 결정하고, 해당 블록의 움직임 벡터를 스케일링 할 수 있다.

[0198] 표 10은 상술한 우선 순위에 기초해서, 상측 그룹 블록들 중에서 어느 한 블록을 선택하고 해당 블록의 움직임 벡터를 스케일링하여  $a'$  을 구해내기 위한 수도 코드의 일 예이다.

## 표 10

```

a' = not available
For (i=0 ; i<na ; i++) {
    if ( i번째 block이 inter mode이고 현재 블록과 같은 list의 같은 ref_idx를
    참조하는가?) {
        a' = ai
        break;
    }
}
If (a' is not available) {
    For (i=0 ; i<na ; i++) {
        if ( i번째 block이 inter mode이고, 현재 블록과 같은 ref_pic을
        참조하는가?) {
            a' = ai // 같은 ref_pic을 참조하므로 별도의 scaling이
            불필요하다.
            break;
        }
    }
}
If (a' is not available) {
    For (i=0 ; i<na ; i++) {
        if ( i번째 block이 inter mode이고, 현재 블록과 같은 list를
        참조하는가?) {
            a' = scale ( ai )
            break;
        }
    }
}
If (a' is not available) {
    For (i=0 ; i<na ; i++) {
        if ( i번째 block이 inter mode인가?) {
            a' = scale ( ai )
            break;
        }
    }
}
If (a' is not available) {
    For (i=0 ; i<na ; i++) {
        if ( i번째 block이 inter mode) {
            a' = scale ( ai )
            break;
        }
    }
}

```

[0199]

표 10에서는 검색의 순서를  $a_0$ 에서부터  $a_{na}$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $a_{na}$ 에서부터  $a_0$ 쪽으로 검색하여 첫 번째 이용 가능한 블록의 움직임 벡터를 스케일링해서 구해낼 수도 있다.

[0201]

표 11은 상술한 우선 순위에 기초해서, 좌측 그룹 블록들 중에서 어느 한 블록을 선택하고 해당 블록의 움직임 벡터를 스케일링하여  $b'$ 을 구해내기 위한 수도 코드의 일 예이다.

## 표 11

```

b' = not available
For (i=0 ; i<nb ; i++) {
    if ( i번째 block이 inter mode이고 현재 블록과 같은 list의 같은 ref_idx를
    참조하는가?) {
        ba' = bi
        break;
    }
}
If (b' is not available) {
    For (i=0 ; i<nb ; i++) {
        if ( i번째 block이 inter mode이고, 현재 블록과 같은 ref_pic을
        참조하는가?) {
            b' = bi // 같은 ref_pic를 참조하므로 별도의 scaling이
            불필요하다.
            break;
        }
    }
}
If (b' is not available) {
    For (i=0 ; i<nb ; i++) {
        if ( i번째 block이 inter mode이고, 현재 블록과 같은 List를
        참조하는가?) {
            b' = scale ( bi )
            break;
        }
    }
}
If (b' is not available) {
    For (i=0 ; i<nb ; i++) {
        if ( i번째 block이 inter mode인가?) {
            b' = scale ( bi )
            break;
        }
    }
}
If (b' is not available) {
    For (i=0 ; i<nb ; i++) {
        if ( i번째 block이 inter mode) {
            b' = scale ( bi )
            break;
        }
    }
}

```

[0202]

표 11에서도 검색의 순서를  $b_0$ 에서부터  $b_{nb}$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $b_{nb}$ 에서부터  $b_0$ 쪽으로 검색하여 첫 번째 이용 가능한 블록의 움직임 벡터를 스케일링해서 구해낼 수도 있다.

[0203]

표 12는 상술한 우선 순위에 기초해서, 코측 그룹 블록들 중에서 어느 한 블록을 선택하고 해당 블록의 움직임 벡터를 스케일링하여  $c'$  을 구해내기 위한 수도 코드의 일 예이다.

## 표 12

```

c' = not available
For (i=0 ; i<3 ; i++) {
    if ( i번째 block이 inter mode이고 현재 블록과 같은 list의 같은 ref_idx를
    참조하는가?) {
        c' = ci
        break;
    }
}
If (c' is not available) {
    For (i=0 ; i<3 ; i++) {
        if ( i번째 block이 inter mode이고, 현재 블록과 같은 ref_pic을
        참조하는가?) {
            c' = ci // 같은 ref_pic을 참조하므로 별도의 scaling이
            불필요하다.
            break;
        }
    }
}
If (c' is not available) {
    For (i=0 ; i<3 ; i++) {
        if ( i번째 block이 inter mode이고, 현재 블록과 같은 List를
        참조하는가?) {
            c' = scale ( ci )
            break;
        }
    }
}
If (c' is not available) {
    For (i=0 ; i<3 ; i++) {
        if ( i번째 block이 inter mode인가?) {
            c' = scale ( ci )
            break;
        }
    }
}
If (c' is not available) {
    For (i=0 ; i<3 ; i++) {
        if ( i번째 block이 inter mode) {
            c' = scale ( ci )
            break;
        }
    }
}

```

[0205]

[0206] 표 12에서도 검색의 순서를  $c_0$ 에서부터  $c_2$ 쪽으로 검색하는 경우를 예로 설명하였으나, 이에 한정하지 않고,  $c_2$ 에서부터  $c_0$ 쪽으로 검색하거나,  $c_1 \rightarrow c_2 \rightarrow c_0$  또는  $c_1 \rightarrow c_0 \rightarrow c_2$  순으로 검색하여 첫 번째 이용 가능한 블록의 움직임 벡터를 스케일링해서 구해낼 수도 있다.

[0207] 표 10 내지 12에서는 상술한 바와 같은 우선 순위에 따라서, 스케일링 대상을 선택하고 이용 가능한 움직임 벡터 후보를 산출하는 것을 설명하였으나, 우선 순위는 상술한 바에 한정되지 않고, 필요에 따라서 다양하게 변경 적용할 수 있다.

[0208] 한편, 상술한 우선 순위를 조건으로 정리하면, 아래와 같은 검토 조건으로 표현될 수 있다.

[0209] <검토 조건>

[0210] 스캔(scan)하는 대상 블록의 움직임 정보(참조 픽처, 참조 픽처 인덱스)가,

[0211] (1) 현재 블록과 동일한 참조 픽처이고 동일한 참조 픽처 리스트

[0212] (2) 현재 블록과 동일한 참조 픽처이고 상이한 참조 픽처 리스트

[0213] (3) 현재 블록과 상이한 참조 픽처이고 동일한 참조 픽처 리스트

[0214] (4) 현재 블록과 상이한 참조 픽처이고 상이한 참조 픽처 리스트

[0215] 상기 검토 조건과 관련하여, 인코딩 장치 또는 디코딩 장치는 움직임 벡터 예측(motion vector prediction)을 위한, 하나의 공간적 움직임 벡터 후보를 결정하기 위해 소정 개수, 예컨대  $n$  개의 블록을 정해진 순서에 따라서 스캔한다. 이때, 인코딩 장치 또는 디코딩 장치는 현재 블록 주변의  $n$  개의 블록들을 차례로 스캔하면서 상기 검토 조건에 해당하는 블록인지를 판단할 수도 있고, 현재 블록의 주변 블록을 그룹 별로, 예컨대 상측 그룹, 좌측 그룹, 코너 그룹별로 스캔하면서, 스캔하는 그룹에 대하여 상기 검토 조건에 해당하는 블록이 있는지를 판단할 수도 있다. 또한, 상측 그룹, 좌측 그룹, 코너 그룹으로 나누는 대신, 좌상측 코너 블록과 우상측 코너 블록을 포함하는 상측 그룹과 좌하측 코너 블록을 포함하는 좌측 블록으로 현재 블록의 주변 블록을 나누고, 두 그룹에 대하여 그룹별로 스캔하면서 스캔하는 그룹에 상기 검토 조건을 만족하는 블록이 있는지를 판단할 수도 있다.

[0216] 상기 검토 조건에 대하여, 인코딩 장치 또는 디코딩 장치는 낮은 번호의 검토 조건에 더 높은 우선 순위를 둘 수 있다. 왜냐하면, 더 낮은 번호의 검토 조건을 만족하는 주변 블록일수록 현재 블록과 더 유사한 움직임 정보를 가지는 것으로 볼 수 있기 때문이다.

[0217] 이하에서는 우선 현재 블록 주변의  $n$  개의 블록들을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인되는 첫 번째 블록의 움직임 벡터를 그대로 또는 스케일링하여 현재 블록의 움직임 벡터를 예측하기 위한 움직임 벡터 예측자의 후보로서 사용하는 방법을 표를 참조하여 설명한다.

[0218] 여기서, 이용 가능한 블록이라 함은 현재 블록의 인터 예측에 이용할 수 있는 블록을 의미한다.

[0219] 스케일링은 상술한 바와 같이, 현재 블록의 참조 픽처와 현재 픽처 사이의 시간적 거리 및 주변 블록의 참조 픽처와 현재 픽처 사이의 시간적 거리를 고려하여 수행될 수 있다.

[0220] 아래의 표들에 기재된 1 내지 12의 숫자는 스캔(확인)의 순서를 나타낸다. 여기서는 설명의 편의를 위해, 현재 블록의 주변 블록으로서 블록 0, 블록 1, 블록 2을 고려한다. 블록 0 내지 블록 2는 공간적 움직임 벡터 후보를 구하기 위한 후보 블록들의 위치를 나타낸다. 블록 0 내지 블록 2 각각의 위치는 현재 블록의 주변의 좌측, 상측, 코너 중 어느 하나에 위치한다. 설명의 편의를 위해, 블록 0 내지 블록 2는 스캔 순서에 따라서 순차적으로 위치한다고 가정한다.

[0221] 표 13은 현재 블록 주변의 블록 0 내지 블록 2를 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 일 예를 나타낸 것이다.

표 13

	블록 0	블록 1	블록 2
검토 조건 1	1	2	3
검토 조건 2	4	5	6
검토 조건 3	7	8	9
검토 조건 4	10	11	12

[0222]

[0223] 표 13의 예에서는, 블록 0에서 검토 조건 1을 만족하는 움직임 정보가 있는지를 검색해서, 검토 조건 1을 만족하는 움직임 정보가 블록 0에 있는 경우에는 블록 0의 움직임 벡터(mv)를 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 활용하고, 검토 조건 1을 만족하는 움직임 정보가 블록 0에 없는 경우에는 스캔 순서에서 2번째인 블록 1을 검색하여 검토 조건 1을 만족하는 움직임 정보가 블록 1에 있는지를 확인하는 방식으로 현재 블록의 공간적 움직임 벡터 예측자 후보를 결정한다.

[0224] 표 14는 현재 블록 주변의 블록 0 내지 블록 2를 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 다른 예를 나타낸 것이다.

표 14

	블록 0	블록 1	블록 2
검토 조건 1	1	5	9
검토 조건 2	2	6	10
검토 조건 3	3	7	11
검토 조건 4	4	8	12

[0225]

표 14의 예에서는, 블록 0에서 검토 조건 1을 만족하는 움직임 정보가 있는지를 검색해서, 검토 조건 1을 만족하는 움직임 정보가 블록 0에 있는 경우에는 블록 0의 움직임 벡터(mv)를 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 활용하고, 검토 조건 1을 만족하는 움직임 정보가 블록 0에 없는 경우에는 블록 0에 검토 조건 2를 만족하는 움직임 정보가 있는지를 스캔(확인)한다. 블록 0에 검토 조건 1 내지 4를 만족하는 움직임 정보가 없는 경우에는, 블록 1에 대하여 검토 조건 1부터 검토 조건 4까지 순서대로 검토하면서 해당 조건을 만족하는 움직임 정보가 있는지를 스캔하는 방식으로 현재 블록에 대한 공간적 움직임 벡터 예측자 후보를 결정한다.

[0227]

표 15는 현재 블록 주변의 블록 0 내지 블록 2를 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 또 다른 예를 나타낸 것이다.

표 15

	블록 0	블록 1	블록 2
검토 조건 1	1	2	3
검토 조건 2	4	7	10
검토 조건 3	5	8	11
검토 조건 4	6	9	12

[0228]

표 15의 예에서는, 블록 0에서 검토 조건 1을 만족하는 움직임 정보가 있는지를 검색해서, 검토 조건 1을 만족하는 움직임 정보가 블록 0에 있는 경우에는 블록 0의 움직임 벡터(mv)를 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 활용한다. 검토 조건 1을 만족하는 움직임 정보가 블록 0에 없는 경우에는 블록 1과 블록 2에 검토 조건 1을 만족하는 움직임 정보가 있는지를 순서대로 스캔한다. 블록 0 내지 2에 검토 조건 1을 만족하는 움직임 정보가 없는 경우에는, 블록 0에 대하여 검토 조건 2부터 검토 조건 4까지 순서대로 검토하면서 해당 조건을 만족하는 움직임 정보가 있는지를 스캔하고, 블록 0에 검토 조건 2 내지 검토 조건 4를 만족하는 움직임 정보가 없는 경우에는 블록 1에 대하여 순서대로 검토 조건 2부터 검토 조건 4까지 순서대로 검토하는 방식으로 현재 블록에 대한 공간적 움직임 벡터 예측자 후보를 결정한다.

[0230]

표 16은 현재 블록 주변의 블록 0 내지 블록 2를 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 또 다른 예를 나타낸 것이다.

## 표 16

	블록 0	블록 1	블록 2
검토 조건 1	1	5	6
검토 조건 2	2	7	8
검토 조건 3	3	9	10
검토 조건 4	4	11	12

[0231]

표 16의 예에서는, 블록 0에서 검토 조건 1부터 시작해서 검토 조건 4까지, 해당 조건을 만족하는 움직임 벡터가 있는지를 순차적으로 확인한다. 예컨대, 검토 조건 1을 만족하는 움직임 정보가 있다면, 해당 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 이용한다. 검토 조건 1을 만족하는 움직임 정보가 없다면 검토 조건 2를 만족하는 움직임 정보가 있는지를 확인한다. 블록 0에 검토 조건 1 내지 검토 조건 4를 만족하는 움직임 정보가 없는 경우에는 블록 1과 블록 2에 검토 조건 1을 만족하는 움직임 정보가 있는지를 순서대로 스캔한다. 블록 0 내지 2에 검토 조건 1을 만족하는 움직임 정보가 없는 경우에는, 블록 1과 블록 2에 대하여 검토 조건 2를 만족하는 움직임 정보가 있는지를 순서대로 확인하는 방식으로 현재 블록에 대한 공간적 움직임 벡터 예측자 후보를 결정한다.

[0232]

상술한 표 13의 예는 예컨대, 블록의 위치보다는 검토 조건에 우선 순위를 두는 스캔 순서라고 할 수 있다. 이에 반하여, 표 14는 검토 조건보다는 블록의 위치에 우선 순위를 두는 스캔 순서의 예라고 할 수 있다.

[0233]

표 15 및 표 16은 검토 조건과 블록의 위치의 우선 순위를 적절히 혼합한 스캔 순서의 예라고 하겠다. 표 15는 상술한 바와 같이, 검토 조건 1을 최우선으로 하되, 검토 조건 1을 만족하는 블록이 없는 경우에, 블록의 위치를 고려하는 스캔 순서의 예라고 할 수 있다. 또한, 표 16은 블록 0에서 각 검토 조건의 이용 가능성(availability)를 검토한 후, 이용 가능한 움직임 정보가 없는 경우에, 조건을 차례대로 고려하는 스캔 순서의 예라고 할 수 있다.

[0234]

상술한 방법들에 의해 스케일링된 움직임 벡터(예컨대,  $a'$ ,  $b'$ ,  $c'$ )는  $a'$ ,  $b'$ ,  $c'$ 의 미디언 벡터 값에 영향을 줄 수 있다. 또한, 후보 움직임 벡터를 결정할 때 고려할 움직임 벡터의 개수 등에 영향을 줄 수도 있다. 예컨대, 스케일링에 의해 새롭게 이용 가능한 움직임 벡터로 분류되는 주변 블록의 움직임 벡터가 생길 수 있다.

[0235]

상술한 방법들은 AMVP 기술을 직접적으로 적용하는 경우가 아니어도 이용할 수 있다. 예컨대, 앞서 언급한  $a'$ ,  $b'$ ,  $c'$ 를 구하고,  $a'$ ,  $b'$ ,  $c'$ 로부터 미디언(median)을 구하는 과정 등을 통해서 현재 블록의 움직임 벡터 예측을 수행할 수도 있다. 이때,

[0236]

$a'$ ,  $b'$ ,  $c'$ 를 구하는 과정은 AMVP에서  $a'$ ,  $b'$ ,  $c'$ 를 구하는 과정과 동일하게 상술한 방법을 이용할 수도 있다.

[0237]

한편, AMVP 방법을 도 9와 같이 변형하여 적용할 수도 있다.

[0238]

도 9는 본 발명의 일 실시예에 따라서 AMVP를 적용하는 다른 방법을 개략적으로 설명하는 도면이다. 도 9를 참조하면, 움직임 벡터 후보들로서 도 8에 도시된 현재 블록의 주변 블록들 중 일부를 이용할 수 있다. 예컨대, 도 9의 예에서는, 현재 블록(900)의 좌측에 있는  $m$ 개의 좌측 블록들( $A_0, A_1, \dots, A_m$ )들 중에서 현재 블록 좌측의 최하측에 위치하는 블록( $A_m$ ), 현재 블록(900)의 좌하측 코너 블록( $A_{m+1}$ ), 현재 블록(900)의 좌상측 코너 블록( $B_{-1}$ ), 현재 블록(900)의 상측에 있는  $n$ 개의 상측 블록들 중에서 현재 블록 상측의 최우측에 위치하는 블록( $B_n$ ), 현재 블록(900)의 상우측 코너 블록( $B_{n+1}$ )을 AMVP의 후보 블록으로 이용할 수 있다.

[0239]

이 경우, 각 코너 블록들, 좌하측 블록(420-1) 및 우상측 블록(440-1)을 이용하는 도 4의 경우와 동일하게, 현재 블록의 주변 블록들  $A_m, A_{m+1}, B_{-1}, B_n, B_{n+1}$ 을 이용한다. 도 9의 예에서도, 좌측 블록들( $A_m, A_{m+1}$ )을  $A_{m+1} \rightarrow A_m$ 의

순서(910)로 스캔(또는 확인)하고, 상측 블록들( $B_{-1}$ ,  $B_n$ ,  $B_{n+1}$ )을  $B_{n+1} \rightarrow B_n \rightarrow B_{-1}$ 의 순서(920)로 스캔(또는 확인)한다.

[0241] 이때, 현재 블록에 대한 움직임 벡터 후보로서 수식 6과 같은 움직임 벡터들(움직임 벡터 예측자: MVP)을 고려할 수 있다.

[0242] <수식 6>

[0243] 움직임 벡터 후보들 = { $a'$  ,  $b'$  , 시간적 예측자(temporal predictor)}

[0244] 수식 6은, 도 9의 예에 있어서, 현재 블록에 대한 움직임 벡터 예측자로서 이용될 수 있는 움직임 벡터의 후보들을 나타낸 것이다. 인코딩 장치 또는 디코딩 장치는 수식 6의 움직임 벡터 후보들 전부 또는 일부를 이용하여도 3에서 설명한 움직임 벡터 예측자 후보 리스트를 생성할 수 있다.

[0245] 수식 6을 참조하면, 도 9의 예에서는 현재 블록의 움직임 벡터를 예측하기 위해, 현재 블록의 공간적 주변 블록들 중 두 블록의 움직임 벡터를 고려한다. 인코딩 장치 또는 디코딩 장치는 수식 6에 나타난 현재 블록의 두 공간적 주변 블록의 움직임 벡터들과 현재 블록의 시간적 주변 블록의 움직임 벡터(시간적 예측자)를 모두 고려할 수 있다. 또한, 인코딩 장치 또는 디코딩 장치는 수식 6에 나타난 두 공간적 주변 블록의 움직임 벡터들을 우선 고려하고, 두 공간적 주변 블록의 움직임 벡터들 중 이용할 수 없는(unavailable) 움직임 벡터가 있는 경우에 시간적 예측자를 고려할 수도 있다.

[0246] 수식 6에서  $a'$  은 현재 블록의 좌측 블록들( $A_m$ ,  $A_{m+1}$ )을  $A_{m+1} \rightarrow A_m$ 의 순서(910)로 스캔(검토)할 때, 이용 가능한 것으로 제일 먼저 검색되는 블록의 움직임 벡터로서 현재 블록에 대한 움직임 벡터 예측자 후보이다.  $a'$  을 좌측 움직임 벡터 예측자(left mv predictor)라고 할 수도 있다.

[0247] 또한, 수식 6에서  $b'$  은 현재 블록의 상측 블록들( $B_{-1}$ ,  $B_n$ ,  $B_{n+1}$ )을  $B_{n+1} \rightarrow B_n \rightarrow B_{-1}$ 의 순서(920)로 스캔(검토)할 때, 이용 가능한 것으로 제일 먼저 검색되는 블록의 움직임 벡터로서, 현재 블록에 대한 움직임 벡터 예측자 후보이다.  $b'$  을 상측 움직임 벡터 예측자(top mv predictor)라고 할 수도 있다.

[0248] 이용 가능한 것으로 선택된 주변 블록의 참조 픽처 인덱스가 현재 블록의 참조 픽처 인덱스와 동일한 경우, 즉 선택된 주변 블록의 참조 픽처가 현재 블록의 참조 픽처와 동일한 경우에, 해당 블록의 움직임 벡터는 스케일링 없이 현재 블록에 대한 움직임 벡터 예측자로서 이용될 수 있다. 이 경우는 상기 검토 조건 1 또는 검토 조건 2의 경우에 해당한다고 할 수 있다.

[0249] 이용 가능한 것으로 선택된 주변 블록의 참조 픽처 인덱스가 현재 블록의 참조 픽처 인덱스와 상이한 경우, 즉 선택된 주변 블록의 참조 픽처가 현재 블록의 참조 픽처와 상이한 경우에, 해당 블록의 움직임 벡터는 스케일링을 통해 현재 블록에 대한 움직임 벡터 예측자로서 이용될 수도 있다. 이 경우는 상기 검토 조건 3 또는 검토 조건 4의 경우에 해당한다고 할 수 있다.

[0250] 도 9의 경우에도 상기 4 단계의 검토 조건 및 주변 블록의 위치를 고려하여 다양한 순서로 현재 블록의 주변 블록에 대한 움직임 벡터의 스캔이 가능하다. 이때, 인코딩 장치 또는 디코딩 장치는 좌측 블록들( $A_m$ ,  $A_{m+1}$ )로부터 구해지는 움직임 벡터 예측자 후보, 예컨대 좌측 움직임 벡터 예측자와 상측 블록들( $B_{-1}$ ,  $B_n$ ,  $B_{n+1}$ )로부터 구해지는 움직임 벡터 예측자 후보, 예컨대 상측 움직임 벡터 예측자를 따로 결정할 수 있다.

[0251] 도 8의 경우와 마찬가지로, 도 9의 경우에도 아래의 표에서 정해진 순서에 따라서 스캔을 하면서 검토 조건을 만족하는지를 확인하며, 검토 조건이 만족되는 경우에는 스캔을 중단하고, 검토 조건을 만족하는 블록의 움직임 벡터를 그대로 혹은 스케일링하여 현재 블록의 움직임 벡터를 예측하기 위한 움직임 벡터 예측자(MVP) 후보로서 이용한다.

[0252] 인코딩 장치 또는 디코딩 장치는 선택된 움직임 벡터 예측자 후보를 기반으로 움직임 벡터 예측자(MVP) 리스트를 구성할 수 있다. 인코딩 장치는 MVP 리스트에서 현재 블록의 예측에 이용될 움직임 벡터 예측자를 움직임 벡터 예측자 인덱스 등을 통해 지시할 수 있다. 디코딩 장치는 인코딩 장치로부터 수신한 움직임 벡터 예측자 인덱스가 움직임 벡터 예측자 리스트에서 지시하는 움직임 벡터 예측자를 이용하여 현재 블록의 움직임 벡터를 유도할 수 있다.

[0253] 이하, 좌측 움직임 벡터 예측자를 선택하는 방법과 상측 움직임 벡터 예측자를 선택하는 방법을 차례대로 설명한다. 이하의 표에서도 1 내지 8의 숫자는 스캔의 순서를 나타낸다.

[0254] 표 17은 도 9에 있어서 현재 블록(900) 좌측의 주변 블록  $A_m$ ,  $A_{m+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 혹은 스케일링하여 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 일 예를 나타낸 것이다.

표 17

	블록 $A_m$	블록 $A_{m+1}$
검토 조건 1	1	5
검토 조건 2	2	6
검토 조건 3	3	7
검토 조건 4	4	8

[0255]

[0256] 표 17의 예에서는, 블록  $A_m$ 이 상기 검토 조건들 중 하나를 만족하는지를 확인한다. 블록  $A_m$ 이 검토 조건 1을 만족하지 않는다면, 블록  $A_m$ 이 검토 조건 2를 만족하는지를 확인한다. 이와 같은 방식으로 스캔하여, 블록  $A_m$ 이 검토 조건 1 내지 검토 조건 4를 만족하지 않는 경우에는 스캔 순서에 따라서 블록  $A_{m+1}$ 이 검토 조건 1 내지 검토 조건 4 중 하나를 만족하는지를 순서대로 확인한다.

[0257]

스캔 과정에서 해당 블록이 검토 조건을 만족하면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 꾹처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0258]

표 18은 도 9에 있어서 현재 블록(900) 좌측의 주변 블록  $A_m$ ,  $A_{m+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 혹은 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 다른 예를 나타낸 것이다..

표 18

	블록 $A_m$	블록 $A_{m+1}$
검토 조건 1	1	2
검토 조건 2	3	6
검토 조건 3	4	7
검토 조건 4	5	8

[0259]

[0260] 표 18의 예에서는, 블록  $A_m$ 이 검토 조건 1을 만족하는지를 확인한다. 블록  $A_m$ 이 검토 조건 1을 만족하지 않는 경우에는, 블록  $A_{m+1}$ 이 검토 조건 1을 만족하는지를 확인한다. 블록  $A_{m+1}$ 도 검토 조건 1을 만족하지 않는 경우에는, 블록  $A_m$ 이 검토 조건 2 내지 4 중 하나를 만족하는지를 차례로 확인한다. 블록  $A_m$ 이 검토 조건 2 내지 검토 조건 4를 만족하지 않으면, 스캔 순서에 따라서 블록  $A_{m+1}$ 이 검토 조건 2 내지 검토 조건 4 중 하나를 만족하는지를 차례로 확인한다.

[0261]

스캔 과정에서 해당 블록이 검토 조건을 만족하면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록

의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 픽처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0262] 표 19는 도 9에 있어서 현재 블록(900) 좌측의 주변 블록  $A_m$ ,  $A_{m+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 또는 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 또 다른 예를 나타낸 것이다.

표 19

	블록 $A_m$	블록 $A_{m+1}$
검토 조건 1	5	1
검토 조건 2	6	2
검토 조건 3	7	3
검토 조건 4	8	4

[0263]

[0264] 표 19의 예에서는, 표 17의 스캔 순서와 다르게, 블록  $A_{m+1}$ 이 검토 조건 1 내지 검토 조건 4를 만족하는지를 순서대로 확인하고, 블록  $A_m$ 이 검토 조건 1 내지 검토 조건 4를 만족하지 않는 경우에, 블록  $A_m$ 이 검토 조건 1 내지 검토 조건 4를 만족하는지를 순서대로 확인한다.

[0265] 스캔 과정에서 검토 조건을 만족하는 블록을 찾으면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 픽처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0266] 표 20은 도 9에 있어서 현재 블록(900) 좌측의 주변 블록  $A_m$ ,  $A_{m+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 혹은 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 또 다른 예를 나타낸 것이다.

표 20

	블록 $A_m$	블록 $A_{m+1}$
검토 조건 1	2	1
검토 조건 2	6	3
검토 조건 3	7	4
검토 조건 4	8	5

[0267]

[0268] 표 20의 예에서는, 표 18의 예에서 적용한 스캔 순서를, 표 18의 예와 달리 블록  $A_{m+1}$ 에 우선 순위를 두고 적용한다. 예컨대, 표 20의 예에서는 블록  $A_{m+1}$ 이 조건 1을 만족하는지를 확인한다. 블록  $A_{m+1}$ 이 검토 조건 1을 만족하지 않는 경우에는, 스캔 순서에 따라서 블록  $A_m$ 이 검토 조건 1을 만족하는지를 확인한다. 블록  $A_m$ 도 검토 조건

1을 만족하지 않는 경우에는 스캔 순서에 따라서, 블록  $A_{m+1}$ 이 검토 조건 2 내지 4 중 어느 하나를 만족하는지를 순서대로 확인한다. 블록  $A_{m+1}$ 이 검토 조건 2 내지 검토 조건 4를 만족하지 않으면, 스캔 순서에 따라서 블록  $A_m$ 이 검토 조건 2 내지 검토 조건 4 중 어느 하나를 만족하는지를 확인한다.

[0269] 스캔 과정에서 검토 조건을 만족하는 블록을 찾으면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 꾹처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0270] 표 21은 도 9에 있어서 현재 블록(900) 좌측의 주변 블록  $A_m$ ,  $A_{m+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 혹은 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 또 다른 예를 나타낸 것이다.

표 21

	블록 $A_m$	블록 $A_{m+1}$
검토 조건 1	2	1
검토 조건 2	4	3
검토 조건 3	7	5
검토 조건 4	8	6

[0271]

표 21의 예에서는, 블록  $A_{m+1}$ 이 검토 조건 1을 만족하는지를 먼저 확인한다. 블록  $A_{m+1}$ 이 검토 조건 1을 만족하지 않는 경우에는, 스캔 순서에 따라서 블록  $A_m$ 이 검토 조건 1을 만족하는지를 확인한다. 블록  $A_m$ 도 검토 조건 1을 만족하지 않는 경우에는 스캔 순서에 따라서, 블록  $A_{m+1}$ 이 검토 조건 2를 만족하는지를 확인한다. 블록  $A_{m+1}$ 이 검토 조건 2를 만족하지 않으면, 스캔 순서에 따라서 블록  $A_m$ 이 검토 조건 2를 만족하는지를 확인한다. 블록  $A_m$ 이 검토 조건 2를 만족하지 않으면, 블록  $A_{m+1}$ 이 검토 조건 3 내지 검토 조건 4 중 하나를 만족하는지를, 블록  $A_m$ 이 검토 조건 3 내지 검토 조건 4 중 하나를 만족하는지를 순서대로 확인한다.

[0273] 스캔 과정에서 검토 조건을 만족하는 블록을 찾으면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 꾹처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0274] 표 22는 도 9에 있어서 현재 블록(900) 좌측의 주변 블록  $A_m$ ,  $A_{m+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 혹은 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 또 다른 예를 나타낸 것이다.

표 22

	블록 $A_m$	블록 $A_{m+1}$
검토 조건 1	3	1
검토 조건 2	4	2
검토 조건 3	7	5
검토 조건 4	8	6

[0275]

표 22에서는 각 블록별로 두 개씩의 검토 조건을 순서대로 확인한다. 구체적으로, 인코딩 장치 또는 디코딩 장치는 블록  $A_{m+1}$ 이 검토 조건 1을 만족하는지를 먼저 확인한다. 블록  $A_{m+1}$ 이 검토 조건 1을 만족하지 않으면, 인코딩 장치 또는 디코딩 장치는 블록  $A_{m+1}$ 이 검토 조건 2를 만족하는지를 확인한다. 블록  $A_{m+1}$ 이 검토 조건 2를 만족하지 않으면, 인코딩 장치 또는 디코딩 장치는 블록  $A_m$ 이 검토 조건 1을 만족하는지를 확인한다. 블록  $A_m$ 이 검토 조건 1을 만족하지 않으면, 인코딩 장치 또는 디코딩 장치는 블록  $A_m$ 이 검토 조건 2를 만족하는지를 확인한다. 블록  $A_m$ 이 검토 조건 2를 만족하지 않으면, 인코딩 장치 또는 디코딩 장치는 블록  $A_{m+1}$ 이 검토 조건 3을 만족하는지를 확인한다. 블록  $A_{m+1}$ 이 검토 조건 3을 만족하지 않으면, 인코딩 장치 또는 디코딩 장치는 블록  $A_{m+1}$ 이 검토 조건 4를 만족하는지를 확인한다. 블록  $A_{m+1}$ 이 검토 조건 4를 만족하지 않으면, 인코딩 장치 또는 디코딩 장치는 블록  $A_m$ 이 검토 조건 3을 만족하는지를 확인한다. 블록  $A_m$ 이 검토 조건 3을 만족하지 않으면, 인코딩 장치 또는 디코딩 장치는 블록  $A_m$ 이 검토 조건 4를 만족하는지를 확인한다.

[0277]

스캔 과정에서 검토 조건을 만족하는 블록을 찾으면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 픽처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0278]

상기 표 17 내지 표 22의 예와 같이, 현재 블록의 좌측 주변 블록들  $A_m$ 과  $A_{m+1}$  중에서 현재 블록에 대한 움직임 벡터 예측자 후보, 즉 좌측 움직임 벡터 후보(left mv candidate)을 구하기 위해 다양한 스캔 순서를 적용할 수 있다.

[0279]

이때, 스캔을 수행하는 경우의 수를 줄여서 복잡도의 이득을 얻기 위해서 일부 스캔을 생략 또는 제거할 수도 있다. 예컨대, 표 17의 예에서, 순서 1 내지 5까지의 스캔만 수행하고, 순서 6 내지 8에 해당하는 스캔은 생략할 수도 있다. 유사하게, 표 18 내지 표 22의 예에서도, 후순위(예컨대 5 내지 8, 6 내지 8, 7 내지 8 또는 8 등)의 스캔은 프로세스의 복잡도를 줄이기 위해 생략할 수도 있다.

[0280]

또한, 검토 조건에 해당하는지를 확인하는 절차를 검토 조건별로 묶어서 수행함으로써 복잡도의 이득을 얻을 수도 있다. 예컨대, 표 22의 경우에는 각 블록별로 두 개씩의 검토 조건을 차례로 확인한다.

[0281]

또한, 표 21의 예와 표 22의 예는, 검토 조건 1과 검토 조건 2에 우선 순위를 두는 방법이다. 상술한 바와 같이, 검토 조건 1과 검토 조건 2의 경우에 해당하는 블록은 현재 블록과 동일한 참조 픽처를 가지므로, 스케일링 할 필요없이 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 이용할 수 있다. 따라서, 스케일링의 빈도를 줄이고 복잡도를 감소시킬 수 있다.

[0282]

이제, 현재 블록의 상측 주변 블록들  $B_{-1}$ ,  $B_n$ ,  $B_{n+1}$ 으로부터 현재 블록에 대한 움직임 벡터 예측자 후보, 즉 상측 움직임 벡터 예측자를 선택하는 방법을 설명한다. 상측 주변 블록들로부터 현재 블록에 대한 움직임 벡터 예측자 후보를 선택하는 방법은 스캔 대상 블록이 3개라는 점을 제외하면, 좌측 주변 블록들로부터 현재 블록에 대한 움직임 벡터 예측자 후보를 선택하는 방법과 기본적으로 동일하다. 이하의 표에서 1 내지 12의 숫자는 스캔

의 순서를 나타낸다.

[0283] 표 23은 도 9에 있어서 현재 블록(900) 상측의 주변 블록  $B_{-1}$ ,  $B_n$ ,  $B_{n+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 또는 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 일 예를 나타낸 것이다.

표 23

	블록 $B_{n+1}$	블록 $B_n$	블록 $B_{-1}$
검토 조건 1	1	5	9
검토 조건 2	2	6	10
검토 조건 3	3	7	11
검토 조건 4	4	8	12

[0284]

표 23의 예에서는, 블록  $B_{n+1}$ 가 상기 검토 조건들 중 하나를 만족하는지를 확인한다. 블록  $B_{n+1}$ 이 검토 조건 1을 만족하지 않는다면 다음으로 블록  $B_{n+1}$ 이 검토 조건 2를 만족하는지를 확인한다. 이와 같은 방식으로 스캔하여, 블록  $B_{n+1}$ 이 검토 조건 1 내지 검토 조건 4를 만족하지 않는 경우에는 블록  $B_n$ 이 검토 조건 1 내지 검토 조건 4 중 하나를 만족하는지를 순서대로 확인한다. 또한, 블록  $B_n$ 이 검토 조건 1 내지 검토 조건 4를 만족하지 않는 경우에는, 블록  $B_{-1}$ 이 검토 조건 1 내지 검토 조건 4 중 하나를 만족하는지를 순서대로 확인한다.

[0286] 스캔 과정에서 해당 블록이 검토 조건을 만족하면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 픽처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0287] 표 24는 도 9에 있어서 현재 블록(900) 좌측의 상측 블록  $B_{-1}$ ,  $B_n$ ,  $B_{n+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 혹은 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 다른 예를 나타낸 것이다.

표 24

	블록 $B_{n+1}$	블록 $B_n$	블록 $B_{-1}$
검토 조건 1	1	2	3
검토 조건 2	4	5	6
검토 조건 3	7	8	9
검토 조건 4	10	11	12

[0288]

표 24의 예에서는, 블록  $B_{n+1}$ 이 검토 조건 1을 만족하는지를 확인한다. 블록  $B_{n+1}$ 이 검토 조건 1을 만족하지 않는 경우에는, 블록  $B_n$ 이 검토 조건 1을 만족하는지를 확인한다. 블록  $B_n$ 이 검토 조건 1을 만족하지 않는 경우에는 블록  $B_{-1}$ 이 검토 조건 1을 만족하는지를 확인한다. 블록  $B_{-1}$ 이 검토 조건 1을 만족하지 않으면, 다시 블록  $B_{n+1}$ 부

터  $B_{-1}$ 의 순서로, 검토 조건 2를 만족하는지를 확인한다. 블록  $B_{n+1}$ ,  $B_n$ ,  $B_{-1}$ 이 검토 조건 2를 만족하지 않으면, 다시  $B_{n+1}$ 부터  $B_{-1}$ 의 순서로, 검토 조건 3를 만족하는지를 확인한다. 이와 같은 방법으로, 블록  $B_{n+1}$ ,  $B_n$ ,  $B_{-1}$  중에 검토 조건 1 내지 검토 조건 4 중 하나를 만족하는 블록이 있는지를 스캔 순서에 따라서 순차적으로 확인한다.

[0290] 스캔 과정에서 검토 조건을 만족하는 주변 블록을 찾으면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 픽처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0291] 표 25는 도 9에 있어서 현재 블록(900) 상측의 주변 블록  $B_{-1}$ ,  $B_n$ ,  $B_{n+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인한 첫 번째 블록의 움직임 벡터를 그대로 혹은 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 또 다른 예를 나타낸 것이다.

표 25

	블록 $B_{n+1}$	블록 $B_n$	블록 $B_{-1}$
검토 조건 1	1	2	3
검토 조건 2	4	7	10
검토 조건 3	5	8	11
검토 조건 4	6	9	12

[0292]

[0293] 표 25의 예에서는, 검토 조건 1을 만족하는지를 블록  $B_{n+1}$ ,  $B_n$ ,  $B_{-1}$ 의 순서로 확인한다. 검토 조건 1을 만족하는 상측 주변 블록이 없는 경우에는, 블록  $B_{n+1}$ 에 대하여 검토 조건 2 내지 검토 조건 4 중 하나를 만족하는지 순서대로 확인한다. 블록  $B_{n+1}$ 이 검토 조건 2 내지 검토 조건 4를 만족하지 않는 경우에는, 블록  $B_n$ 이 검토 조건 2 내지 검토 조건 4 중 하나를 만족하는지 순서대로 확인한다. 블록  $B_n$ 이 검토 조건 2 내지 검토 조건 4를 만족하지 않는 경우에는, 블록  $B_{-1}$ 이 검토 조건 2 내지 검토 조건 4 중 하나를 만족하는지 순서대로 확인한다.

[0294] 스캔 과정에서 검토 조건을 만족하는 블록을 찾으면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 픽처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0295] 표 26은 도 9에 있어서 현재 블록(900) 상측의 주변 블록  $B_{-1}$ ,  $B_n$ ,  $B_{n+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 또는 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 또 다른 예를 나타낸 것이다.

표 26

	블록 $B_{n+1}$	블록 $B_n$	블록 $B_{-1}$
검토 조건 1	1	2	3
검토 조건 2	4	5	6
검토 조건 3	7	9	11
검토 조건 4	8	10	12

[0296]

표 26의 예에서는, 표 25의 예와 달리, 검토 조건 1 및 2에 대해서는, 각 검토 조건을 블록  $B_{n+1}$ ,  $B_n$ ,  $B_{-1}$ 의 해당 여부를 차례로 판단하며, 조건 3 및 4에 대해서는 각 블록별로 해당 여부를 차례로 판단한다. 예컨대, 우선 검토 조건 1을 만족하는지를 블록  $B_{n+1}$ ,  $B_n$ ,  $B_{-1}$ 의 순서로 확인하고, 검토 조건 1을 만족하는 상측 주변 블록이 없는 경우에는, 검토 조건 2를 만족하는지를 블록  $B_{n+1}$ ,  $B_n$ ,  $B_{-1}$ 의 순서로 확인한다. 검토 조건 1과 검토 조건 2 중 하나를 만족하는 상측 블록이 없는 경우에는, 블록  $B_{n+1}$ 이 검토 조건 3 또는 검토 조건 4 중 하나를 만족하는지 순서대로 확인하고, 블록  $B_{n+1}$ 이 검토 조건 3과 검토 조건 4를 만족하지 않는 경우에는, 블록  $B_n$ 이 검토 조건 3 또는 검토 조건 4 중 하나를 만족하는지 순서대로 확인한다. 블록  $B_n$ 이 검토 조건 3과 검토 조건 4를 만족하지 않는 경우에는, 블록  $B_{-1}$ 이 검토 조건 3 또는 검토 조건 4 중 하나를 만족하는지 순서대로 확인한다.

[0298]

스캔 과정에서 검토 조건을 만족하는 블록을 찾으면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 픽처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0299]

표 27은 도 9에 있어서 현재 블록(900) 상측의 주변 블록  $B_{-1}$ ,  $B_n$ ,  $B_{n+1}$ 을 상기 검토 조건의 우선 순위에 따라서 차례로 스캔하면서 이용 가능한 것으로 확인된 첫 번째 블록의 움직임 벡터를 그대로 혹은 스케일링하여 현재 블록에 대한 공간적 움직임 벡터 예측자 후보로 사용하는 방법의 또 다른 예를 나타낸 것이다.

표 27

	블록 $B_{n+1}$	블록 $B_n$	블록 $B_{-1}$
검토 조건 1	1	3	5
검토 조건 2	2	4	6
검토 조건 3	7	9	11
검토 조건 4	8	10	12

[0300]

표 27의 예에서는, 표 22의 예와 유사하게, 각 블록당 2개씩의 검토 조건을 확인한다. 구체적으로 인코딩 장치 또는 디코딩 장치는 블록  $B_{n+1}$ 이 검토 조건 1 또는 검토 조건 2 중 하나를 만족하는지 순서대로 확인한다. 블록  $B_{n+1}$ 이 검토 조건 1과 검토 조건 2를 만족하지 않는 경우에는, 블록  $B_n$ 이 검토 조건 1 또는 검토 조건 2 중 하나를 만족하는지 순서대로 확인한다. 블록  $B_n$ 이 검토 조건 1과 검토 조건 2를 만족하지 않는 경우에는, 블록  $B_{-1}$ 이 검토 조건 1 또는 검토 조건 2 중 하나를 만족하는지 순서대로 확인한다. 상측 블록들이 검토 조건 1과 검토 조

건 2를 만족하지 않으면, 블록  $B_{n+1}$ 이 검토 조건 3 또는 검토 조건 4 중 하나를 만족하는지 순서대로 확인한다. 블록  $B_{n+1}$ 이 검토 조건 3과 검토 조건 4를 만족하지 않으면, 블록  $B_n$ 이 검토 조건 3 또는 검토 조건 4 중 하나를 만족하는지 순서대로 확인한다. 블록  $B_n$ 이 검토 조건 3과 검토 조건 4를 만족하지 않으면, 블록  $B_{-1}$ 이 검토 조건 3 또는 내지 검토 조건 4 중 하나를 만족하는지 순서대로 확인한다.

[0302] 스캔 과정에서 검토 조건을 만족하는 블록을 찾으면, 인코딩 장치 또는 디코딩 장치는 스캔을 중지하고 해당 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택한다. 인코딩 장치 또는 디코딩 장치는 필요에 따라서 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수도 있다. 예컨대, 인코딩 장치 또는 디코딩 장치는 주변 블록과 현재 블록의 참조 픽처가 상이한 경우에는, 해당 주변 블록의 움직임 벡터를 스케일링하여 움직임 벡터 예측자 후보로서 이용할 수 있다. 이때 스케일링 방법으로서, 상술한 수식 1 또는 수식 2의 방법을 이용할 수도 있다.

[0303] 표 23 내지 표 27에서 보듯이, 현재 블록의 좌측 블록들로부터 현재 블록에 대한 움직임 벡터 예측자 후보를 구하는 것과 유사하게, 현재 블록의 상측 블록들로부터 현재 블록에 대한 움직임 벡터 예측자 후보를 구하기 위해서도 다양한 스캔 순서를 적용할 수 있다.

[0304] 상측 블록들로부터 현재 블록에 대한 움직임 벡터 예측자 후보, 즉 상측 움직임 벡터 후보(top mv candidate)를 구하는 방법에 있어서도, 복잡도를 줄이기 위해, 스캔을 수행하는 경우의 수를 줄일 수도 있다. 예컨대, 검토 조건 1을 만족하는지를 확인하는 스캔이거나 각 검토 조건별로 첫 번째 스캔인 경우에만 스캔을 수행하도록 할 수 있다. 이 경우, 표 23의 예에서는 순서 6, 7, 8, 10, 11, 12의 스캔을 생략할 수 있다.

[0305] 또한, 표 26 및 표 27의 예는 표 21 및 표 22의 예와 유사하게, 검토 조건 1과 검토 조건 2에 우선 순위를 두는 방법이다. 검토 조건 1과 검토 조건 2에 해당하는 블록은 현재 블록(예컨대 현재 예측 유닛)과 동일한 참조 픽처를 참조하므로, 움직임 벡터를 스케일링하지 않고 그대로 현재 블록에 대한 움직임 벡터 예측자 후보로 이용할 수 있다. 따라서, 움직임 벡터를 스케일링하는데 따르는 복잡도를 줄일 수 있다.

[0306] 한편, 상술한 스캔 방법 외에 다양한 검토 조건의 조합 및 검토 조건의 순서, 블록 위치의 순서를 적용하는 것이 가능하다. 또한, 상술한 스캔 방법을 조합하여 적용하는 것 역시 가능하다.

[0307] 또한, 본 명세서에서는 움직임 벡터 예측자라는 표현을 이용하였으나, 이는 설명의 편의를 위한 것으로서, 움직임 벡터 예측자는 예측 움직임 벡터(Predicted Motion Vector: PMV) 등으로 불릴 수도 있다.

[0308] 도 4, 도 5, 도 7, 도 8, 도 9에 있어서, 현재 블록 및 현재 블록의 주변 블록들은 예측 유닛일 수 있다. 또한, 도 4, 도 5, 도 7, 도 8, 도 9에 있어서, 현재 블록 및 현재 블록의 주변 블록들은 예측 유닛일 수도 있다. 뿐만 아니라, 도 4, 도 5, 도 7, 도 8, 도 9에 있어서, 현재 블록 및 현재 블록의 주변 블록들은 변환 유닛일 수도 있다. 따라서, 도 4, 도 5, 도 7, 도 8, 도 9에서 설명한 내용은, 현재 블록 및 현재 블록의 주변 블록이 부호화 유닛이거나 변환 유닛인 경우에도 동일하게 적용될 수 있다.

[0309] 한편, 본 명세서에서는 움직임 벡터 예측자 후보를 선택하고, 선택된 움직임 벡터 예측자 후보를 이용하여 현재 블록에 대한 예측을 수행하는 과정이 인코딩 장치 또는 디코딩 장치에서 이루어지는 것으로 설명하였으나, 이는 설명의 편의를 위한 것으로서, 본 발명은 이에 한정되지 않는다. 움직임 벡터 예측자 후보를 선택하고, 선택된 움직임 벡터 예측자 후보를 이용하여 현재 블록에 대한 예측을 수행하는 과정은 인코딩 장치 또는 디코딩 장치 내 소정의 모듈 혹은 유닛에서 수행될 수도 있다. 예컨대, 움직임 벡터 예측자 후보를 선택하고, 선택된 움직임 벡터 예측자 후보를 이용하여 현재 블록에 대한 예측을 수행하는 과정을 인코딩 장치 또는 디코딩 장치 내부의 예측부에서 수행할 수 있다.

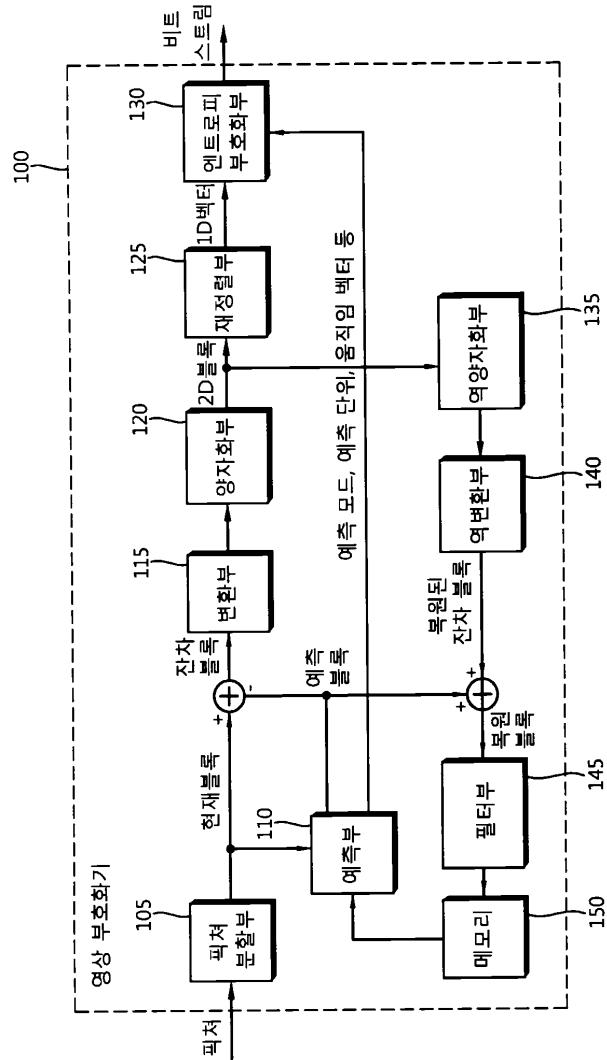
[0310] 도 10은 본 발명에 따라서 현재 블록에 대한 움직임 벡터 예측자를 선택하는 방법을 개략적으로 설명하는 순서 도이다. 도 10의 각 단계는 인코딩 장치 또는 디코딩 장치가 수행할 수 있다. 또한, 인코딩 장치 또는 디코딩 장치의 예측부가 도 10의 각 단계를 수행할 수도 있다. 여기서는 설명의 편의를 위해 인코딩 장치 또는 디코딩 장치 내의 예측부가 도 10의 각 단계를 수행하는 것으로 설명한다.

[0311] 도 10을 참조하면, 예측부는 현재 블록에 대한 움직임 벡터 예측자 후보들을 선택한다(S1010). 예측부는 현재 블록의 주변 블록들을 검색해서 이용 가능한 것으로 검색되는 첫 번째 블록의 움직임 벡터를 현재 블록에 대한 움직임 벡터 예측자 후보로 선택할 수 있다. 이때, 예측부는 현재 블록의 주변 블록들을 소정의 그룹으로 나누어 검색을 수행하고, 각 그룹별로 현재 블록에 대한 움직임 벡터 예측자 후보를 선택할 수도 있다.

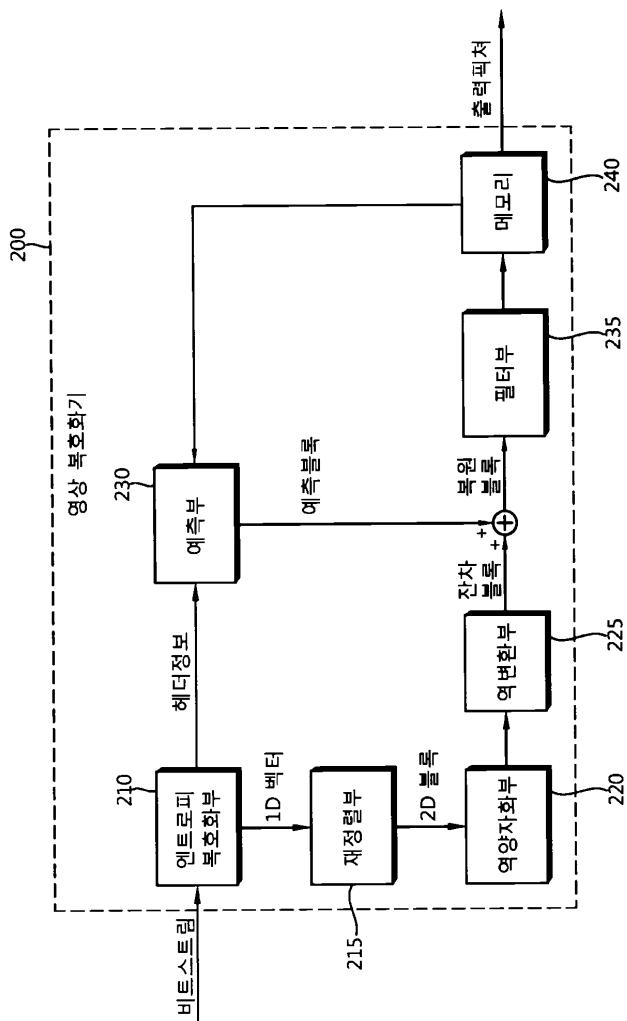
- [0312] 이때, 이용 가능한 블록은 인터 예측 모드로 인코딩/디코딩되는 블록으로서 상술한 검토 조건을 만족하는 블록일 수 있다. 현재 블록의 주변 블록들을 어떻게 그룹으로 나누는지는 상술한 바와 같다.
- [0313] 또한, 예측부는 이용 가능한 것으로 검색되는 첫 번째 블록의 움직임 벡터를 스케일링하여 현재 블록에 대한 움직임 벡터 예측자 후보로 선택할 수도 있다. 예컨대, 예측부는 이용 가능한 것으로 검색되는 첫 번째 블록이 상술한 검토 조건 3 또는 검토 조건 4에 해당하는 경우에는 해당 블록의 움직임 벡터를 스케일링할 수 있다. 스케일링의 구체적인 방법은 상술한 바와 같다.
- [0314] 예측부는 공간적인 주변 블록들로부터 구할 수 있는 움직임 벡터 예측자 후보의 개수가 소정의 개수에 미치지 못하는 경우에는 시간적인 주변 블록들로부터 움직임 벡터 예측자 후보를 구할 수도 있다.
- [0315] 예측부는 움직임 벡터 예측자 후보들 중에서 현재 블록에 대한 움직임 벡터 예측자를 선택할 수 있다(S1020). 예측부는 선택한 움직임 벡터 예측자 후보들로 움직임 벡터 예측자 리스트를 구성할 수 있다. 인코딩 장치의 예측부는 전송 정보량을 최소로 하는 움직임 벡터 예측자를 움직임 벡터 예측자 리스트에서 선택할 수 있다. 디코딩 장치의 예측부는 인코딩 장치로부터 전송된 움직임 벡터 예측자 정보(예컨대, 움직임 벡터 예측자 인덱스)가 지시하는 움직임 벡터 예측자를 움직임 벡터 예측자 리스트에서 선택하여, 현재 블록에 대한 예측을 수행할 수 있다.
- [0316] 상술한 예시적인 시스템에서, 방법들은 일련의 단계 또는 순서도를 기초로 설명되고 있지만, 본 발명은 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 상술한 실시예들은 다양한 양태의 예시들을 포함한다.
- [0317] 따라서, 본 발명은 상술한 실시예들을 동시에 적용하거나 조합하여 적용하는 실시 형태를 포함한다.

## 도면

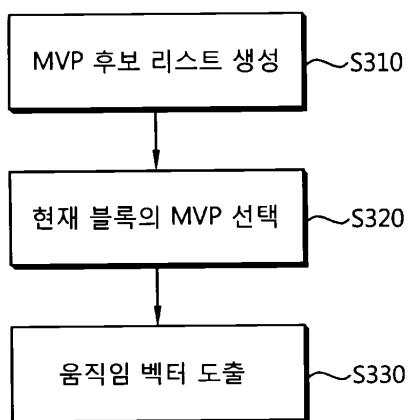
## 도면1



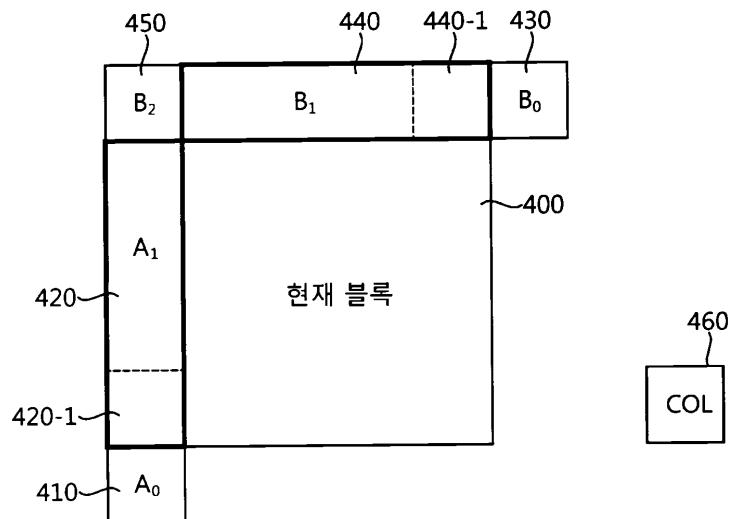
## 도면2



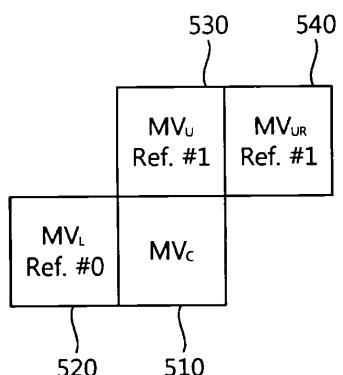
## 도면3



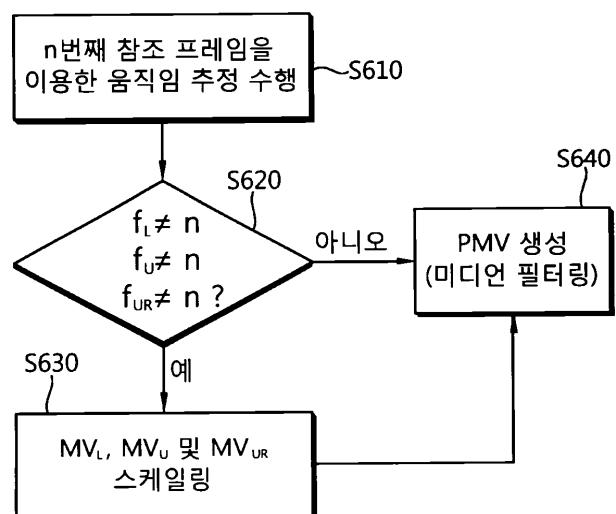
## 도면4



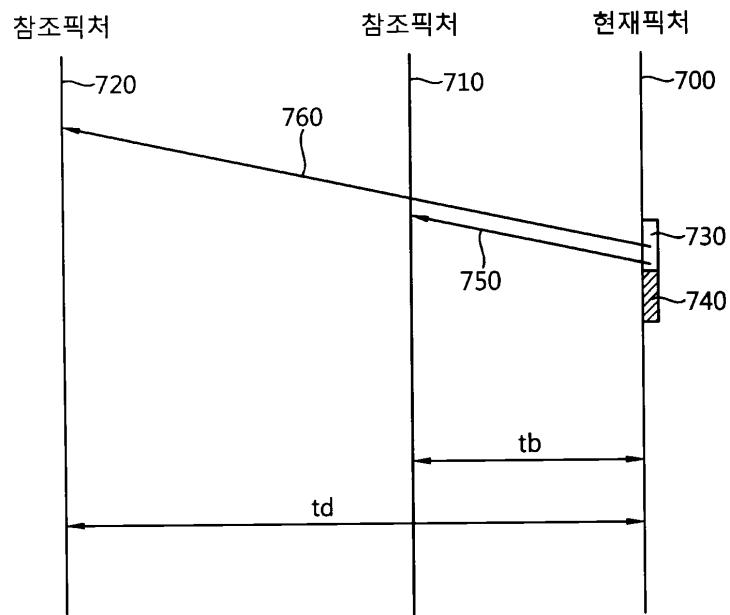
## 도면5



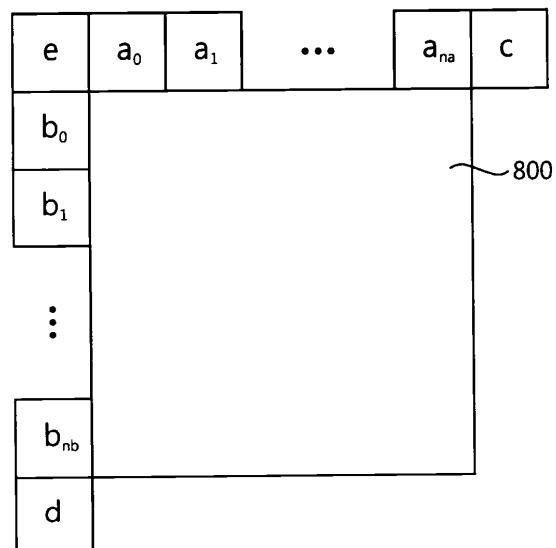
## 도면6



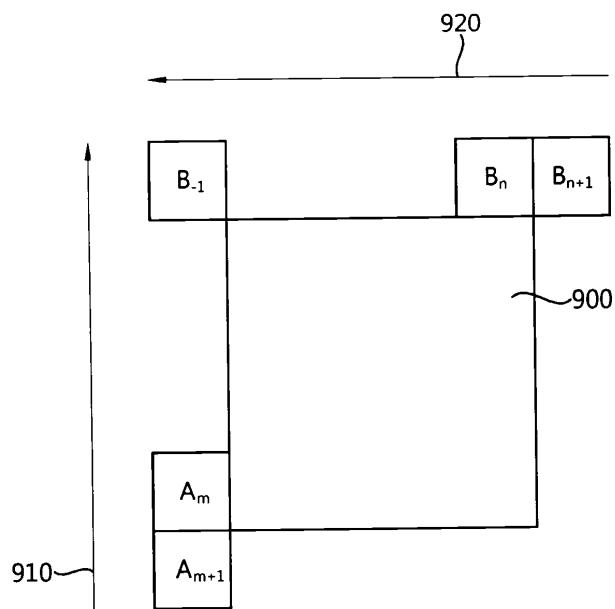
## 도면7



## 도면8



도면9



도면10

