



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0140592
(43) 공개일자 2012년12월31일

(51) 국제특허분류(Int. Cl.)

H04N 7/32 (2006.01)

(21) 출원번호 10-2012-0005948

(22) 출원일자 2012년01월18일

심사청구일자 없음

(30) 우선권주장

1020110060285 2011년06월21일 대한민국(KR)

(뒷면에 계속)

(71) 출원인

한국전자통신연구원

대전광역시 유성구 가정로 218 (가정동)

경희대학교 산학협력단

경기도 용인시 기흥구 덕영대로 1732, 국제캠퍼스 내 (서천동, 경희대학교)

(72) 발명자

김휘용

대전광역시 유성구 지족동 열매마을아파트 601-201

박광훈

경기 성남시 분당구 분당동 45번지 동아빌라 B동 302호

(뒷면에 계속)

(74) 대리인

양문옥

전체 청구항 수 : 총 34 항

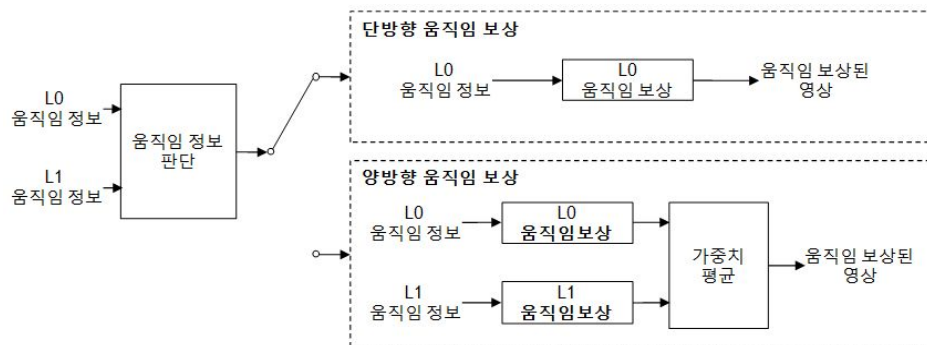
(54) 발명의 명칭 움직임 보상의 계산 복잡도 감소 및 부호화 효율을 증가시키는 방법 및 장치

(57) 요약

본 발명은 양방향 예측을 사용하여 부호화된 블록에 대하여 두 예측 방향이 같은 움직임 정보를 가질 경우, 움직임 보상 과정을 단축하여 계산 복잡도를 감소시키고 부호화 효율을 증가시키는 방법을 포함함.

본 발명은 비디오 부호화에서 계산 복잡도 감소시키고 부호화 효율을 증가시키는데 이용된다.

대표도 - 도6



(72) 발명자

김경용

경기도 수원시 영통구 영통동 1025-15번지 402호

임성창

대전광역시 유성구 신성동 254-8번지 세종빌라 201호

이진호

대전광역시 유성구 신성동 210-51번지 우노빌 302호

최진수

대전광역시 유성구 반석동 613번지 반석마을6단지 아파트 609동 1605호

김진웅

대전광역시 유성구 전민동 엑스포아파트 305-1603

(30) 우선권주장

1020110065714 2011년07월01일 대한민국(KR)

1020110066173 2011년07월04일 대한민국(KR)

이 발명을 지원한 국가연구개발사업

과제고유번호 11921-02001

부처명 방송통신위원회

연구사업명 방송통신기술개발사업(●ETRI연구개발지원)

연구과제명 무안경 다시점 3D 지원 UHDTV 방송 기술 개발

주관기관 한국전자통신연구원

연구기간 2011.03.01 ~ 2015.02.28

특허청구의 범위

청구항 1

두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행시키는 방법을 적용하는 부호화/복호화 방법 및 장치.

청구항 2

제 1항에 있어서, 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같은지를 판단하는 단계; 단방향 움직임 보상을 수행하는 단계; 양방향 움직임 보상을 수행하는 단계; 두 방향의 움직임 정보가 같을 때 단방향 움직임 보상 과정을 수행하도록 하는 단계; 두 방향의 움직임 정보가 다를 때 양방향 움직임 보상 과정을 수행하도록 하는 단계; 를 포함하는 부호화/복호화 방법 및 장치.

청구항 3

움직임 결합(Merge) 과정에서 현재 블록의 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법 및 장치

청구항 4

제 3항에 있어서, 현재 블록의 주변 혹은 대응되는(Collocated) 블록의 예측 방향을 현재 블록의 움직임 정보 후보에 대한 예측 방향으로 설정하는 단계; 현재 블록의 주변 혹은 대응되는(Collocated) 블록의 L0움직임 정보를 현재 블록의 L0 움직임 정보 후보로 등록하는 단계; 현재 블록의 주변 혹은 대응되는(Collocated) 블록에서 L0 움직임 정보와 L1움직임 정보가 같은지를 판단하는 단계; 현재 블록의 주변 혹은 대응되는(Collocated) 블록의 L1움직임 정보를 현재 블록의 L1 움직임 정보 후보로 등록하는 단계; 현재 블록의 움직임 정보 후보에 대한 예측 방향을 단방향으로 설정하는 단계;를 포함하는 부호화/복호화 방법 및 장치

청구항 5

제 4항에 있어서, 현재 블록의 주변 혹은 대응되는(Collocated) 블록에서 L0 움직임 정보와 L1 움직임 정보가 같을 경우, 현재 블록의 L1 움직임 정보 후보를 등록하지 않고 현재 블록의 움직임 정보 후보에 대한 예측 방향을 단방향으로 설정하는 방법 및 장치

청구항 6

제 4항에 있어서, 현재 블록의 주변 혹은 대응되는(Collocated) 블록에서 L0 움직임 정보와 L1움직임 정보가 같지 않을 경우, 현재 블록의 주변 혹은 대응되는(Collocated) 블록의 L1움직임 정보를 현재 블록의 L1 움직임 정보 후보로 등록하는 방법 및 장치

청구항 7

제 3항에 있어서, 움직임 결합(Merge) 과정에서 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에 움직임 정보를 가지는 블록이 존재하지 않는다면, P 픽처(단방향예측픽처)일 경우에는 기존의 방법을 유지하고 B 픽처(양방향예측픽처)일 경우 현재 블록을 단방향 예측으로 설정하고 L0움직임 정보에서 움직임 벡터를 (0,0)으로 설정하고 현재 블록에 대한 L1 움직임 정보에서 움직임 벡터는 설정하지 않거나 혹은 (0,0)으로 설정하는 방법 및 장치

청구항 8

현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트에 따라 움직임 정보를 설정하는 방법 및 장치

청구항 9

제 8항에 있어서, 대응되는 블록의 L0 움직임 정보가 존재하는지 판단하는 단계; 대응되는 블록의 L0 움직임 정보를 현재 블록에 대한 L0 움직임 정보로 설정하는 단계; 현재 블록에 대한 L0 움직임 정보에서 움직임 벡터를

(0,0)으로 설정하는 단계; 현재 픽처가 B 픽처(양방향 예측 픽처)인지를 판단하는 단계; 대응되는 블록의 L1 움직임 정보가 존재하는지 판단하는 단계; 대응되는 블록의 L1 움직임을 정보를 현재 블록에 대한 L1 움직임을 정보로 설정하는 단계; 현재 블록에 대한 L1 움직임을 정보에서 움직임을 벡터를 (0,0)으로 설정하는 단계;를 포함하는 부호화/복호화 방법 및 장치

청구항 10

제 9항에 있어서, 대응되는 블록의 L0 움직임을 정보가 존재한다면, 대응되는 블록의 L0 움직임을 정보를 현재 블록에 대한 L0 움직임을 정보로 설정하는 방법 및 장치

청구항 11

제 9항에 있어서, 대응되는 블록의 L0 움직임을 정보가 존재하지 않는다면, 현재 블록에 대한 L0 움직임을 정보를 설정하지 않거나 혹은 대응되는 블록의 L1 움직임을 정보를 현재 블록에 대한 L0 움직임을 정보로 설정하거나 혹은 현재 블록에 대한 L0 움직임을 정보에서 움직임을 벡터를 (0,0)으로 설정하는 방법 및 장치

청구항 12

제 9항에 있어서, 현재 픽처가 B 픽처(양방향 예측 픽처)일 경우, 현재 블록에 대한 L1 움직임을 정보를 설정하는 과정을 수행하고 그렇지 않을 경우 종료하는 방법 및 장치

청구항 13

제 9항에 있어서, 대응되는 블록의 L1 움직임을 정보가 존재한다면, 대응되는 블록의 L1 움직임을 정보를 현재 블록에 대한 L1 움직임을 정보로 설정하는 방법 및 장치

청구항 14

제 9항에 있어서, 대응되는 블록의 L1 움직임을 정보가 존재하지 않는다면, 현재 블록에 대한 L1 움직임을 정보를 설정하지 않거나 혹은 L1 움직임을 정보에서 움직임을 벡터를 (0,0)으로 설정하는 방법 및 장치

청구항 15

시간적으로 유추된 두 리스트(L0/L1)의 움직임을 정보(참조픽처번호, 움직임을 벡터)가 같을 경우, L0의 움직임을 정보만을 현재 블록에 대한 움직임을 정보로 사용하는 방법 및 장치

청구항 16

제 15항에 있어서, 시간적으로 유추된 움직임을 정보에서 L0 움직임을 정보와 L1움직임을 정보가 같은지를 판단하는 단계; 시간적으로 유추된 움직임을 정보의 예측 방향을 단방향으로 설정하는 단계; 시간적으로 유추된 움직임을 정보에서 L0움직임을 정보만을 현재 블록에 대한 움직임을 정보로 사용하거나 혹은 현재 블록에 대한 움직임을 정보 후보로 등록하는 단계; 를 포함하는 부호화/복호화 방법 및 장치

청구항 17

제 16항에 있어서, 시간적으로 유추된 움직임을 정보에서 L0 움직임을 정보와 L1움직임을 정보가 같지 않을 경우, 시간적으로 유추된 움직임을 정보를 그대로 현재 블록에 대한 움직임을 정보로 사용하거나 혹은 현재 블록에 대한 움직임을 정보 후보로 등록하는 방법 및 장치

청구항 18

제 16항에 있어서, 시간적으로 유추된 움직임을 정보에서 L0 움직임을 정보와 L1움직임을 정보가 같을 경우, 현재 블록에 대한 움직임을 정보(혹은 움직임을 정보 후보)를 단방향 예측으로 설정하고 시간적으로 유추된 움직임을 정보에서 L0움직임을 정보만을 현재 블록에 대한 움직임을 정보로 사용하거나 혹은 현재 블록에 대한 움직임을 정보 후보로 등록하는 방법 및 장치

청구항 19

L0의 참조픽처리스트와 L1의 참조픽처리스트가 같을 경우(예를 들어, 순방향 B 픽처일 경우), 시간적으로 유추된 움직임을 정보를 모두 단방향 예측으로 설정하고 L0움직임을 정보만을 현재 블록에 대한 움직임을 정보로 사용하

나 혹은 현재 블록에 대한 움직임 정보 후보로 등록하여 사용하는 방법 및 장치

청구항 20

움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 현재 블록에 움직임 정보를 설정하는 방법 및 장치

청구항 21

제 20항에 있어서, 현재 픽처가 P픽처일 경우 현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 '0'으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정하는 단계; 현재 픽처가 현재 픽처가 B 픽처이고, L0 참조픽처와 L1 참조픽처가 동일하면, 현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 '0'으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정하는 단계; 현재 픽처가 현재 픽처가 B 픽처이고, L0 참조픽처와 L1 참조픽처가 동일하지 않으면, 현재 블록을 양방향 예측으로 설정하고 L0와 L1 참조픽처번호는 '0'으로 설정하고 L0와 L1 움직임 벡터는 (0,0)으로 설정하는 단계; 를 포함하는 방법 및 장치

청구항 22

최종적으로 결정된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법

청구항 23

제 22항에 있어서, 움직임 정보를 판단하는 단계; 움직임 정보를 단방향 예측으로 설정하고 L0에 대한 움직임 정보만을 현재 블록에 사용하는 단계; 입력된 L0와 L1의 움직임 정보를 현재 블록에 사용하는 단계; 를 포함하는 방법 및 장치

청구항 24

제 22항에 있어서, 입력된 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같으면 움직임 정보를 단방향 예측으로 설정하고 L0에 대한 움직임 정보만을 현재 블록에 사용하고, 만약 입력된 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같지 않으면 입력된 L0와 L1의 움직임 정보를 현재 블록에 사용하는 방법 및 장치

청구항 25

움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 순방향 B픽처에서 현재 블록에 대한 움직임 정보를 설정하는 방법 및 장치

청구항 26

제 25항에 있어서, 현재 픽처가 P픽처일 경우 현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 '0'으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정하는 단계; L0 참조픽처와 L1 참조픽처가 동일하면, 현재 블록을 양방향 예측으로 설정하고 L0 참조픽처번호는 '0'으로 설정하고 L1 참조픽처번호는 '1'로 설정하고 L0와 L1 움직임 벡터는 (0,0)으로 설정하는 단계; L0 참조픽처와 L1 참조픽처가 동일하고, 참조픽처리스트 안의 참조픽처 개수가 1개이하이면, 현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 '0'으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정하는 단계; L0 참조픽처와 L1 참조픽처가 동일하지 않는다면, 현재 블록을 양방향 예측으로 설정하고 L0와 L1 참조픽처번호는 '0'으로 설정하고 L0와 L1 움직임 벡터는 (0,0)으로 설정하는 단계; 를 포함하는 방법 및 장치

청구항 27

현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B 픽처에서 찾으려는 현재 블록의 리스트(예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법 및 장치

청구항 28

제 27항에 있어서, 현재 블록의 리스트(예측방향) 정보에 따라 대응되는 픽처를 설정하는 단계; L0와 L1의 참조픽처의 동일성에 따라 대응되는 픽처를 설정하는 단계; 참조픽처리스트의 개수에 따라 대응되는 픽처를 설정하는 단계; L0 참조픽처리스트에서 1번째 참조픽처를 대응되는 픽처로 설정하는 단계; L1 참조픽처리스트에서 L0

참조픽처와 다른 참조픽처를 대응되는 픽처로 설정하는 단계; L1 참조픽처리스트에서 1번째 참조픽처를 대응되는 픽처로 설정하는 단계; L1 참조픽처리스트에서 2번째 참조픽처를 대응되는 픽처로 설정하는 단계; 를 포함하는 방법 및 장치

청구항 29

제 27항에 있어서, “예측 방향 판단부”에서는 현재 블록의 리스트(예측 방향) 정보가 L0리스트의 움직임 정보를 유추하는 것이라면 L0 참조픽처리스트의 1번째 참조픽처를 대응되는 픽처로 설정하는 방법

청구항 30

제 27항에 있어서, 현재 블록의 리스트(예측방향) 정보가 L1리스트의 움직임 정보를 유추하는 것이고, L0 참조픽처와 L1참조픽처가 동일하지 않다면 L1 참조픽처리스트의 1번째 참조픽처를 대응되는 픽처로 설정하는 방법

청구항 31

제 27항에 있어서, 현재 블록의 리스트(예측방향) 정보가 L1리스트의 움직임 정보를 유추하는 것이고, L0 참조픽처와 L1참조픽처가 동일하고 참조픽처리스트의 개수가 1개 이하이면, L1 참조픽처리스트의 1번째 참조픽처를 대응되는 픽처로 설정하는 방법

청구항 32

제 27항에 있어서, 현재 블록의 리스트(예측방향) 정보가 L1리스트의 움직임 정보를 유추하는 것이고, L0 참조픽처와 L1참조픽처가 동일하면 L1 참조픽처리스트에서 L0참조픽처와 다른 참조픽처를 대응되는 픽처로 설정하는 방법

청구항 33

두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법 및 장치

청구항 34

제 33항에 있어서, 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같은지를 판단하는 단계; 단방향 움직임 보상을 수행하는 단계; 양방향 움직임 보상을 수행하는 단계; L0의 참조픽처와 L1의 참조픽처가 같고 L0 움직임 벡터와 L1 움직임 벡터가 같을 경우, L1참조픽처리스트에서 입력된 L1참조픽처와 다른 참조픽처를 가져와서 L1 참조픽처로 설정하는 단계; 두 방향의 움직임 정보가 다를 때 양방향 움직임 보상 과정을 수행하도록 하는 단계; 를 포함하는 부호화/복호화 방법 및 장치

명세서

기술분야

[0001] 본 발명은 영상 처리에 관한 것이다.

배경기술

[0002] 인터 예측은 화면 간 예측이다.

발명의 내용

해결하려는 과제

[0003] 본 발명은 고효율 동영상 부호화(HEVC; High Efficiency Video Coding)에서 계산 복잡도를 감소시키고 부호화 효율을 증가시키는 방법 및 장치에 관한 것임.

과제의 해결 수단

[0004] 본 발명에서는 움직임 보상 시 계산 복잡도를 낮추고 부호화 효율을 증가시키는 방법을 제안한다.

발명의 효과

[0005] 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행시키는 방법을 적용하여 계산복잡도를 감소시키고 부호화효율을 증가시키는 효과가 있다.

도면의 간단한 설명

- [0006]
- 도 1. HEVC의 부호화 구조도의 일 예
 - 도 2. HEVC의 복호화 구조도의 일 예
 - 도 3. 양방향 예측의 경우, 움직임 보상 방법의 일 예
 - 도 4. 양방향 예측의 경우, 고정확도(HA; High Accuracy) 움직임 보상 방법의 일 예
 - 도 5. 부호화된 영상의 움직임 정보의 일 예 (BasketballDrill, 832x480, QP 32, POC 2)
 - 도 6. 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행함으로써 계산 복잡도를 감소시킬 수 있는 방법 1의 부호화/복호화 구조도의 일 예
 - 도 7. 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 가중치 예측에서 단방향 움직임 보상 과정을 수행함으로써 계산 복잡도를 감소시킬 수 있는 방법의 부호화/복호화 구조도의 일 예
 - 도 8. 움직임 결합(Merge) 과정에서 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법2에 대한 순서도의 일 예
 - 도 9. 현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트에 따라 움직임 정보를 설정하는 방법3에 대한 순서도의 일 예
 - 도 10. 시간적으로 유추된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법4에 대한 순서도의 일 예
 - 도 11. 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 현재 블록에 움직임 정보를 설정하는 방법5에 대한 순서도의 일 예
 - 도 12. 최종적으로 결정된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법6에 대한 순서도의 일 예
 - 도 13. 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 순방향 B픽처에서 현재 블록에 대한 움직임 정보를 설정하는 방법7에 대한 순서도의 일 예
 - 도 14. 현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B 픽처에서 찾으려는 현재 블록의 리스트(예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법8에 대한 순서도의 일 예
 - 도 15. 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법 9에 대한 순서도의 일 예
 - 도 7. 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법 9에 대한 순서도의 일 예
 - 도 16. 두 방향의 움직임 정보가 같을 경우, 움직임 보상 과정을 한번만 수행함으로써 계산 복잡도를 감소시킬 수 있는 방법의 부호화/복호화 장치의 일 예
 - 도 17. 움직임 결합(Merge) 과정에서 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법2의 부호화/복호화 장치의 일 예
 - 도 18. 현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트에 따라 움직임 정보를 설정하는 방법3의 부호화/복호화 장치의 일 예
 - 도 19. 시간적으로 유추된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움

직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법4의 부호화/복호화 장치의 일예

도 20. 현재 블록의 주변 블록 혹은 대응되는 블록에 움직임 정보가 없을 경우, (0,0) 움직임 정보를 사용하는 방법5의 부호화/복호화 장치의 일예

도 21. 최종적으로 결정된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법6의 부호화/복호화 장치의 일예

도 22. 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 순방향 B 픽처에서 현재 블록에 대한 움직임 정보를 설정하는 방법 7의 부호화/복호화 장치의 일예

도 23. 현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B 픽처에서 찾으려는 현재 블록의 리스트(예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법8의 부호화/복호화 장치의 일예

도 24. 두 방향의 움직임 정보가 같을 경우, 움직임 보상 과정을 한번만 수행함으로써 계산 복잡도를 감소시킬 수 있는 방법의 부호화/복호화 장치의 일예

발명을 실시하기 위한 구체적인 내용

[0007] 최근 HD(High Definition) 해상도(1280x1024 혹은 1920x1080)를 가지는 방송 서비스가 국내뿐만 아니라 세계적으로 확대되면서, 이제 많은 사용자들이 고해상도, 고화질의 영상에 많이 익숙해지고 있으며 그에 발맞춰서 많은 기관에서는 차세대 영상기기에 대한 개발에 박차를 가하고 있다. 또한 HDTV와 더불어 HDTV의 4배 이상의 해상도를 갖는 UHD(Ultra High Definition)에 대한 관심이 증대되면서 동영상 표준화 단체들은 보다 높은 해상도, 고화질의 영상에 대한 압축기술의 필요성을 인식하게 되었다. 또한 현재 HDTV, 휴대전화, 블루레이 플레이어에 사용되는 H.264/AVC보다 높은 압축 효율을 통해 동일한 화질을 유지하면서도 주파수 대역이나 저장 측면에서 많은 이득을 얻을 수 있는 새로운 표준이 절실한 실정이다. 현재 MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)는 공동으로 차세대 비디오 코덱인 HEVC(High Efficiency Video Coding)를 표준화하고 있으며, UHD 영상까지 포함한 영상을 H.264/AVC 대비 2배의 압축효율로 부호화를 하는 것을 목표로 하고 있다. 이는 HD, UHD 영상뿐만 아니라 3D 방송 및 이동통신망에서도 현재보다 낮은 주파수로 고화질의 영상을 제공할 수 있다.

[0008] HEVC는 현재 2010년 4월에 첫 JCT-VC(Joint Collaboration Team Video Coding) 회의가 열린 이후 각 기관의 기고를 통해 HM(HEVC Test Model)이라는 명칭의 코덱을 표준의 성능을 측정하기 위해 책정하였다. 현재 HEVC의 부호화 구조도의 일 예는 [도 1]과 같다

[0010] 도 1. HEVC의 부호화 구조도의 일 예

[0011] HEVC의 부호화 수행 과정은 다음과 같다. 영상을 입력 받아 인트라(Intra) 모드 또는 인터(Inter) 모드로 부호화를 수행하고 비트스트림을 출력한다. 인트라 모드일 경우, 스위치가 인트라로 전환이 되며, 인터 모드일 경우에는 스위치가 인터로 전환이 된다. 부호화 과정의 주요한 흐름은 먼저 입력된 블록 영상에 대한 예측 블록을 생성한 후, 입력된 블록과 예측 블록의 차분을 구해 그 차분을 부호화하는 것이다. 먼저 예측 블록의 생성은 인트라 모드와 인터 모드에 따라 수행이 된다. 먼저 인트라 모드일 경우에는 인트라 예측 과정에서 현재 블록의 이미 부호화된 주변 화소값을 이용하여 공간적 예측을 수행하여 예측 블록을 생성하며, 인터 모드일 경우에는 움직임 예측 과정에서 참조 영상 버퍼에 저장되어 있는 참조 영상에서 현재 입력된 블록과 가장 매치가 잘 되는 영역을 찾아 움직임 벡터(Motion Vector)를 구한 후, 구한 움직임 벡터를 이용하여 움직임 보상을 수행함으로써 예측 블록을 생성한다. 상기 설명한 것과 같이 현재 입력된 블록과 예측 블록의 차분을 구하여 잔여 블록(Residual Block)을 생성한 후, 이에 대한 부호화를 수행한다. 잔여 블록에 대한 부호화는 변환, 양자화, 엔트로피(Entropy) 부호화의 순서로 수행이 된다. 먼저 변환 과정에서는 입력된 잔여 블록을 입력 받아 변환(Transform)을 수행하여 변환계수(Transform Coefficient)를 출력한다. 그리고 양자화 과정에서는 입력된 변환계수를 양자화 파라미터에 따라 양자화를 수행한 양자화된 계수(Quantized Coefficient)를 출력한다. 그리고 엔트로피 부호화 과정에서는 입력된 양자화된 계수를 확률 분포에 따른 엔트로피 부호화를 수행하여 비트스트림으로 출력된다. HEVC는 프레임간(Inter-frame) 예측 부호화를 수행하기 때문에 현재 부호화된 영상을 이후에 입력

된 영상의 참조 영상으로 사용하기 위해 복호화하여 저장할 필요가 있다. 따라서 양자화된 계수를 역양자화과정과 역변환을 수행하여 예측 영상과 가산기를 통해 재구성된 블록을 생성한 다음 디블록킹 필터를 통해 부호화 과정에서 발생한 블록킹 현상(Blocking Artifact)을 제거한 후, 참조 영상 버퍼에 저장한다.

[0012] HEVC의 복호화 구조도의 일 예는 [도 2]와 같다.

[0014] 도 2. HEVC의 복호화 구조도의 일 예

[0015] HEVC의 부호화를 수행하면 비트스트림이 출력되는데, 이 비트스트림을 입력 받아 인트라(Intra) 모드 또는 인터(Inter) 모드로 복호화가 수행되어 재구성된 영상을 출력한다. 인트라 모드일 경우, 스위치가 인트라로 전환이 되며, 인터 모드일 경우에는 스위치가 인터로 전환이 된다. 복호화 과정의 주요한 흐름은 먼저 예측 블록을 생성한 후, 입력 받은 비트스트림을 복호화한 결과 블록과 예측 블록을 더하여 재구성된 블록을 생성하는 것이다. 먼저 예측 블록의 생성은 인트라 모드와 인터 모드에 따라 수행이 된다. 먼저 인트라 모드일 경우에는 인트라 예측 과정에서 현재 블록의 이미 부호화된 주변 화소값을 이용하여 공간적 예측을 수행하여 예측 블록을 생성하며, 인터 모드일 경우에는 움직임 벡터를 이용하여 참조 영상 버퍼에 저장되어 있는 참조 영상에서 영역을 찾아 움직임 보상을 수행함으로써 예측 블록을 생성한다. 엔트로피 복호화 과정에서는 입력된 비트스트림을 확률 분포에 따른 엔트로피 복호화를 수행하여 양자화된 계수(Quantized Coefficient)를 출력한다. 양자화된 계수를 역양자화과정과 역변환을 수행하여 예측 영상과 가산기를 통해 재구성된 블록을 생성한 다음 디블록킹 필터를 통해 블록킹 현상(Blocking Artifact)를 제거한 후, 참조 영상 버퍼에 저장한다.

[0016] HEVC에서는 화면 간 예측 시, 단방향 예측(P 픽처) 뿐만 아니라 양방향 예측(B 픽처)을 사용하여 부호화한다. 양방향 예측을 사용할 경우, 참조 픽처는 두 개를 가질 수 있으며 블록당 움직임 정보(참조픽처번호, 움직임벡터)도 두 개를 가질 수 있다. 본 발명에서 “픽처”는 문맥에 따라 “프레임”, “필드, 혹은 “슬라이스”로 대체되어 사용될 수 있으며, 이러한 구분은 본 분야의 통상의 지식을 가진 사람은 쉽게 할 수 있다. 여기서 참조 픽처들은 참조픽처 리스트에 의해 관리되고, 일반적으로 순방향(과거 영상) 참조픽처는 참조픽처리스트 0(L0; List 0)에 할당되고 역방향(미래 영상) 참조픽처는 참조픽처리스트 1(L1; List 1)에 할당된다. 참조픽처들이 참조픽처리스트에 할당되어지는 순서는 일반적으로 참조픽처번호에 따라 내림차순으로 할당되며, 이는 참조픽처리스트재배열(RPLR; Reference Picture List Reordering) 혹은 메모리관리컨트롤(MMCO; Memory management control operation) 명령에 의해 재배열되어 질 수 있다. 따라서 양방향 예측을 사용할 경우, [도 3]과 같이 이 두 개의 참조픽처 리스트에서 각각 하나씩 참조픽처를 할당받아 예측에 사용하게 된다. 이때 L0 움직임 보상과정과 L1 움직임 보상 과정은 서로 독립적으로 수행된다. L0 움직임 보상된 영상과 L1 움직임 보상된 영상은 가중치 평균되어 최종적으로 하나의 움직임 보상된 영상을 출력한다.

[0018] 도 3. 양방향 예측의 경우, 움직임 보상 방법의 일예

[0019] 그리고 HEVC에서는 양방향 움직임 보상이 발생하는 라운딩 에러(Rounding error)로 인해 라운딩 조절(Rounding control)을 수행하고 있다. 이를 위해 [도 4]와 같이 움직임 보상 수행 전에 영상의 비트 깊이(Bit depth)를 증가시키고, 마지막 가중치 평균이 수행될 때 다시 영상의 비트 깊이(Bit depth)를 감소시키는 고정확도(HA; High Accuracy) 화소 보간(IF-Interpolation Filter) 방법을 이용한 움직임 보상을 수행하며 또한 고정확도(HA; High Accuracy) 가중치 평균 방법을 이용하여 하나의 움직임 보상된 영상을 만든다.

- [0021] 도 4. 양방향 예측의 경우, 고정확도(HA; High Accuracy) 움직임 보상 방법의 일예
- [0022] 그 결과 단방향 예측시 사용되는 움직임 보상과 양방향 예측시 사용되는 움직임 보상 간의 방법의 차이가 발생된다.
- [0023] 또한 HEVC에서는 저지연 응용환경에서의 화면 간 예측 효율을 높이기 위하여 순방향 예측만 가능한 B픽처를 사용할 수 있다. 본 발명에서는 이러한 B픽처를 “순방향 B픽처”라 칭하기로 한다. 이러한 순방향 B픽처는 블록당 움직임 정보(참조픽처번호, 움직임 벡터)를 두 개 가질 수 있으며, 통상 참조 픽처 리스트 L0와 L1이 항상 같아지도록 설정하여 사용한다. [도 5]는 표준화가 진행 중인 HEVC의 참조모델인 HM3.0을 이용하여 부호한 결과 영상이며, 움직임 정보를 보여주고 있다. [도 5]의 영상이 순방향 B 픽처 사용하여 부호화되었기 때문에 각 블록에 움직임 정보가 두 개까지 존재한다는 것을 알 수 있다.
- [0025] 도 5. 부호화된 영상의 움직임 정보의 일 예
- [0026] (BasketballDrill, 832x480, QP 32, POC 2)
- [0027] 그런데 [도 5]에서 움직임 정보(참조픽처번호, 움직임 벡터)가 두 개인 블록들을 살펴보면, 순방향 B 픽처 픽처에서 L0의 움직임 정보(참조픽처번호, 움직임 벡터)와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같은 블록이 다수 존재한다는 것을 확인할 수 있다. 이는 복호화에서 화면 간 보상을 수행할 때 같은 과정을 두 번 반복하여 수행하게 됨으로써 매우 비효율적이라 할 수 있다. 따라서 두 방향의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 움직임 보상 과정을 한번만 수행함으로써 계산 복잡도를 감소시킬 수 있다.
- [0028] 순방향 B 픽처 픽처에서 L0의 움직임 정보(참조픽처번호, 움직임 벡터)와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같은 블록이 발생하는 현상은 시간적으로 움직임을 예측하는 방법 때문에 일어난다. 시간적으로 움직임을 예측하는 방법은 참조픽처에서 현재 블록에 대응되는(Collocated) 블록의 움직임을 이용하는 방법이다. 만일 현재 블록의 L0에 해당하는 움직임을 시간적으로 유추할 때, 참조픽처에서 현재 블록에 대응되는 블록의 L0 움직임을 사용한다. 그런데 만일 참조픽처에서 현재 블록에 대응되는 블록의 L0 움직임 정보가 없다면 대응되는 블록의 L1 움직임을 이용하여 현재 블록의 L0 움직임을 사용한다. 반대로 현재 블록의 L1에 해당하는 움직임을 시간적으로 유추할 때, 참조픽처에서 현재 블록에 대응되는 블록의 L1 움직임을 사용한다. 그런데 만일 참조픽처에서 현재 블록에 대응되는 블록의 L1 움직임 정보가 없다면 대응되는 블록의 L0 움직임을 이용하여 현재 블록의 L1 움직임을 사용한다. 그 결과 현재 블록의 L0움직임 정보와 L1 움직임 정보가 같아지는 현상이 발생하게 된다.
- [0029] 이렇게 L0와 L1의 움직임 정보가 같은 블록이 발생되면, 해당 블록이 다음에 부호화되는 블록에 영향을 미친다. 그러한 영향을 받는 예로 움직임 정보 결합 방법에 의해서 해당 블록이 전파되게 된다. 움직임 정보 결합(Merge) 방법은 주변 혹은 대응되는(Collocated) 블록의 L0/L1 움직임 정보(참조픽처번호, 움직임 벡터) 모두를 현재 블록의 움직임 정보로 그대로 사용하는 방법이다. 그 결과 L0와 L1의 움직임 정보가 같은 블록이 더욱 많이 발생하게 된다.
- [0030] 따라서 이러한 현상을 해결하여 계산 복잡도를 감소시키고 부호화 효율을 향상시키는 방법을 제안한다.
- [0031] 본 발명에서는 양방향 예측을 수행한 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 계산 복잡도를 감소시키고 부호화 효율을 향상시킬 수 있는 방법을 제안한다.
- [0032] [방법1] 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행함으로써 계산 복잡도를 감소시킬 수 있는 움직임 보상 방법

- [0033] 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행함으로써 계산 복잡도를 감소시킬 수 있는 움직임 보상 방법의 부호화/복호화 구조도의 일 예는 [도 6]와 같다.
- [0035] 도 6. 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행함으로써 계산 복잡도를 감소시킬 수 있는 방법 1의 부호화/복호화 구조도의 일 예
- [0036] [도 6]는 L0 움직임 정보와 L1 움직임 정보를 입력받아 현재 영상(블록)에 대한 움직임 보상된 영상(블록)을 출력하는 구조도의 일 예이다. 움직임 정보 판단 과정에서는 L0와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같은지를 판단하여 움직임 보상 과정을 선택적으로 수행하도록 한다. 만약 L0의 참조픽처와 L1의 참조픽처가 같고 L0 움직임 벡터와 L1 움직임 벡터가 같을 경우, 단방향 움직임 보상 과정을 수행한다. 그렇지 않으면, 양방향 움직임 보상 과정을 수행한다. L0 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 또한 L0 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 그리고 L1 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L1 움직임 정보를 입력받아 L1 움직임 보상된 영상(블록)을 출력한다. 가중치 평균 과정은 양방향 예측에서 수행되며, L0 움직임 보상 영상(블록)과 L1 움직임 보상 영상(블록)을 입력받아 가중치 평균을 수행하여 하나의 움직임 보상된 영상(블록)을 출력한다.
- [0037] 이러한 방법을 가중치 예측 방법에도 동일하게 적용할 수 있다.
- [0038] 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 가중치 예측에서 단방향 움직임 보상 과정을 수행함으로써 계산 복잡도를 감소시킬 수 있는 움직임 보상 방법의 부호화/복호화 구조도의 일 예는 [도 7]과 같다.
- [0040] 도 7. 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 가중치 예측에서 단방향 움직임 보상 과정을 수행함으로써 계산 복잡도를 감소시킬 수 있는 방법의 부호화/복호화 구조도의 일 예
- [0041] [도 7]는 L0 움직임 정보와 L1 움직임 정보를 입력받아 현재 영상(블록)에 대한 움직임 보상된 영상(블록)을 출력하는 구조도의 일 예이다.
- [0042] 우선 가중치 예측을 사용할 경우에는 가중치 움직임 보상이 수행된다. 움직임 정보 판단 과정에서는 L0와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같은지를 판단하여 움직임 보상 과정을 선택적으로 수행하도록 한다. 만약 L0의 참조픽처와 L1의 참조픽처가 같고 L0 움직임 벡터와 L1 움직임 벡터가 같을 경우, 단방향 가중치 움직임 보상 과정을 수행한다. 그렇지 않으면, 양방향 가중치 움직임 보상 과정을 수행한다. 단방향 가중치 움직임 보상이 수행된 경우, L0 가중치 움직임 보상 과정이 수행된다. L0 가중치 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 가중치 움직임 보상된 영상(블록)에 L0 오프셋(offset)을 더한 움직임 보상된 영상(블록)을 출력한다. 양방향 가중치 움직임 보상이 수행된 경우, L0 가중치 움직임 보상 과정과 L1 가중치 움직임 보상 과정이 모두 수행된다. L0 가중치 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 가중치 움직임 보상된 영상(블록)을 출력한다. L1 가중치 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L1 움직임 정보를 입력받아 L1 가중치 움직임 보상된 영상(블록)을 출력한다. 가중치 평균 과정은 L0 가중치 움직임 보상 영상(블록)과 L0 오프셋(offset)과 그리고 L1 가중치 움직임 보상 영상(블록)과 L1 오프셋(offset)을 입력받아 가중치 평균을 수행하여 하나의 움직임 보상된 영상(블록)을 출력한다.
- [0043] 위에서 가중치 예측을 사용하는 경우, L0와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 때, 여러가지 방법으로 가중치 예측이 수행될 수 있다.

- [0044] 1.L0와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 때, 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 가중치 움직임 보상된 영상(블록)에 L0 오프셋(offset)을 더한 움직임 보상된 영상(블록)을 출력한다.
- [0045] 2.L0와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 때, 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)에 L0 가중치와 L1 가중치를 곱한 영상에 L0 오프셋(offset)과 L1 오프셋(offset)값을 더하여 평균한 움직임 보상된 영상(블록)을 출력할 수 있다.
- [0046] 3.L0와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 때, 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)에 L0 가중치를 곱한 영상에 L0 오프셋(offset)과 L1 오프셋(offset)을 평균한 값을 더하여 움직임 보상된 영상(블록)을 출력할 수 있다.
- [0047] 4.L0와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 때, 양방향 가중치 움직임 보상이 수행될 수 있다.
- [0048] 이외에도 다양하게 가중치 예측이 수행될 수 있다.
- [0049] 가중치 예측을 사용하지 않는 경우에는 일반적인 움직임 보상 과정이 수행된다. 움직임 정보 판단 과정에서는 L0와 L1의 움직임 정보(참조픽처번호, 움직임 벡터)가 같은지를 판단하여 움직임 보상 과정을 선택적으로 수행하도록 한다. 만약 L0의 참조픽처와 L1의 참조픽처가 같고 L0 움직임 벡터와 L1 움직임 벡터가 같을 경우, 단방향 움직임 보상 과정을 수행한다. 그렇지 않으면, 양방향 움직임 보상 과정을 수행한다. L0 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 또한 L0 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 그리고 L1 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L1 움직임 정보를 입력받아 L1 움직임 보상된 영상(블록)을 출력한다. 가중치 평균 과정은 양방향 예측에서 수행되며, L0 움직임 보상 영상(블록)과 L1 움직임 보상 영상(블록)을 입력받아 가중치 평균을 수행하여 하나의 움직임 보상된 영상(블록)을 출력한다.
- [0050] 아래의 내용은 상술한 움직임 보상 방법에 따른 가중치 예측 과정을 상세히 설명한 것이다.
- [0051] 사용되는 용어 및 변수:
- [0052] -weighted_bipred_idc; 가중치 예측을 수행할지에 대한 정보를 나타내며, 이 값이 '0' 이면 가중치 예측이 수행되지 않고 일반적인 예측 과정이 수행된다. 이 값이 '1' 이면 명시적 가중치 예측이 수행되며, 이 값이 '2' 이면 묵시적 가중치 예측이 수행된다.
- [0053] -ClipH(x): 밝기신호(Luma)의 경우 H는 Y로, 색차신호(Chroma)의 경우 H는 C로 대체된다..
- [0054] -Clip1Y(x) = Clip3(0, (1 << BitDepthY) - 1, x)
- [0055] -Clip1C(x) = Clip3(0, (1 << BitDepthC) - 1, x)
- [0056] -Clip1C(x) = Clip3(0, (1 << BitDepthC) - 1, x)
- [0057] -
- $$Clip3(x, y, z) = \begin{cases} x & ; z < x \\ y & ; z > y \\ z & ; \text{otherwise} \end{cases}$$
- [0058] 변수 shift1은 14 - bitDepth 이며, 변수 shift2 는 15 - bitDepth이다.

- [0059] 변수 offset1 은 $1 \ll (\text{shift1} - 1)$ 이며, 변수 offset2는 $1 \ll (\text{shift2} - 1)$ 이다.
- [0060] 1.가중치 샘플 예측이 없는 예측 과정의 예 (방법1)
- [0061] 이 단계는 weighted_bipred_idc 값이 '0' 일 때 수행된다.
- [0062] 이 단계에서 입력은 아래와 같다.
- [0063] -현재 PU 블록의 좌상단 위치; (xB, yB)
- [0064] -현재 PU 블록의 너비와 높이를 가르키는 변수; (nPSW, nPSH)
- [0065] -샘플에 대한 움직임 벡터; (mvL0와 mvL1)
- [0066] -참조픽처인덱스; refIdxL0와 refIdxL1
- [0067] -예측 리스트 이용가능 여부(L0혹은 L1예측을 수행하는지)를 알려주는 변수; predFlagL0와 predFlagL1
- [0068] -샘플의 비트 값; bitDepth
- [0069] 이 단계에서 출력은 아래와 같다.
- [0070] -(nPSW x nPSH) 크기의 예측 샘플; predSamples[x, y]
- [0071] mvL0, mvL1, refIdxL0, refIdxL1, PredFlagL0, PredFlagL1을 아래 과정에 의해 유도한다.
- [0072] -만일 PredFlagL0가 1이고 PredFlagL1가 0이라면 아래를 수행한다.
- [0073] $\text{predSamples}[x, y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, (\text{predSamplesL0}[x, y] + \text{offset1}) \gg \text{shift1})$
- [0074] -그렇지 않고, 만일 PredFlagL0가 0이고 PredFlagL1가 1이라면 아래를 수행한다.
- [0075] $\text{predSamples}[x, y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, (\text{predSamplesL1}[x, y] + \text{offset1}) \gg \text{shift1})$
- [0076] -그렇지 않고, 만일 PredFlagL0가 1이고 PredFlagL1가 1이고, 움직임 벡터 mvL0와 mvL1가 같고, RefPicOrderCnt(currPic, refIdxL0, L0)와 RefPicOrderCnt(currPic, refIdxL1, L1)이 동일하다면(즉, 참조픽처의 픽처번호가 동일하다면)면 아래를 수행한다.
- [0077] $\text{predSamples}[x, y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1, (\text{predSamplesL0}[x, y] + \text{offset1}) \gg \text{shift1})$
- [0078] -그렇지 않다면 아래를 수행한다.
- [0079] $\text{predSamples}[x, y] = \text{Clip3}(0, (1 \ll \text{bitDepth}) - 1,$
- [0080] $(\text{predSamplesL0}[x, y] + \text{predSamplesL1}[x, y] + \text{offset2}) \gg \text{shift2})$
- [0081] 2.가중치 샘플 예측을 수행하는 예측 과정의 예 (방법1)
- [0082] 이 단계는 weighted_bipred_idc 값이 '0' 일 때 수행된다.
- [0083] 이 단계에서 입력은 아래와 같다.
- [0084] -현재 PU 블록의 좌상단 위치; (xB, yB)
- [0085] -현재 PU 블록의 너비와 높이를 가르키는 변수; (nPSW, nPSH)
- [0086] -샘플에 대한 움직임 벡터; (mvL0와 mvL1)
- [0087] -참조픽처인덱스; refIdxL0와 refIdxL1

- [0088] -예측 리스트 이용가능 여부(L0혹은 L1예측을 수행하는지)를 알려주는 변수; predFlagL0와 predFlagL1
- [0089] -샘플의 비트 값; bitDepth
- [0090] -가중치 예측 변수들; logWDC, logWDC, w0C, w1C, o0C, o1C, 여기서 C는 Luma 일경우 L로 Cb일경우 Cb, Cr 일경우 Cr로 변경된다. logWDC는 weighting factor의 분모이며, w0C는 L0의 가중치 값이고 w1C는 L1의 가중치 값이며, o0C는 L0의 오프셋 값이고 o1C는 L1의 오프셋 값이다.
- [0091] 이 단계에서 출력은 아래와 같다.
- [0092] -(nPSW x nPSH) 크기의 예측 샘플; predSamples[x, y]
- [0093] mvL0, mvL1, refIdxL0, refIdxL1, PredFlagL0, PredFlagL1을 아래 과정에 의해 유도한다.
- [0094] -만일 PredFlagL0가 1이고 PredFlagL1가 0이라면 아래를 수행한다.
- [0095] logWDC 값이 1 이상이면
- [0096]
$$\text{predSamples}[x, y] = \text{ClipH}((\text{predSamplesL0}[x, y] * w0C + 2^{\log\text{WDC} - 1}) \gg \log\text{WDC}) + o0C)$$
- [0097] logWDC 값이 1 미만이면
- [0098]
$$\text{predSamples}[x, y] = \text{ClipH}(\text{predSamplesL0}[x, y] * w0C + o0C)$$
- [0099] -그렇지 않고, 만일 PredFlagL0가 0이고 PredFlagL1가 1이라면 아래를 수행한다.
- [0100] logWDC 값이 1 이상이면
- [0101]
$$\text{predSamples}[x, y] = \text{ClipH}((\text{predSamplesL1}[x, y] * w1C + 2^{\log\text{WDC} - 1}) \gg \log\text{WDC}) + o1C)$$
- [0102] logWDC 값이 1 미만이면
- [0103]
$$\text{predSamples}[x, y] = \text{ClipH}(\text{predSamplesL1}[x, y] * w1C + o1C)$$
- [0104] -그렇지 않고, 만일 PredFlagL0가 1이고 PredFlagL1가 1이고, 움직임 벡터 mvL0와 mvL1가 같고, RefPicOrderCnt(currPic, refIdxL0, L0)와 RefPicOrderCnt(currPic, refIdxL1, L1)이 동일하다면(즉, 참조픽처의 픽처번호가 동일하다면)면 아래를 수행한다.
- [0105]
$$\text{predSamples}[x, y] = \text{ClipH}((\text{predSamplesL0}[x, y] * (w0C + w1C) + 2^{\log\text{WDC}}) \gg (\log\text{WDC} + 1)) + ((o0C + o1C + 1) \gg 1))$$
- [0106] -그렇지 않다면 아래를 수행한다.
- [0107]
$$\text{predSamples}[x, y] = \text{ClipH}((\text{predSamplesL0}[x, y] * w0C + \text{predSamplesL1}[x, y] * w1C + 2^{\log\text{WDC}}) \gg (\log\text{WDC} + 1)) + ((o0C + o1C + 1) \gg 1))$$
- [0108] 2.가중치 예측을 수행하는 예측 과정의 예 (방법2)
- [0109] 이 단계는 weighted_bipred_idc 값이 '0' 일 때 수행된다.
- [0110] 이 단계에서 입력은 아래와 같다.
- [0111] -현재 PU 블록의 좌상단 위치; (xB, yB)
- [0112] -현재 PU 블록의 너비와 높이를 가르키는 변수; (nPSW, nPSH)
- [0113] -샘플에 대한 움직임 벡터; (mvL0와 mvL1)
- [0114] -참조픽처인덱스; refIdxL0와 refIdxL1
- [0115] -예측 리스트 이용가능 여부(L0혹은 L1예측을 수행하는지)를 알려주는 변수; predFlagL0와 predFlagL1

- [0116] -샘플의 비트 값; bitDepth
- [0117] -가중치 예측 변수들; logWDC, logWDC, w0C, w1C, o0C, o1C, 여기서 C는 Luma 일경우 L로 Cb일경우 Cb, Cr 일경우 Cr로 변경된다. logWDC는 weighting factor의 분모이며, w0C는 L0의 가중치 값이고 w1C는 L1의 가중치 값이며, o0C는 L0의 오프셋 값이고 o1C는 L1의 오프셋 값이다.
- [0118] 이 단계에서 출력은 아래와 같다.
- [0119] -(nPSW x nPSH) 크기의 예측 샘플; predSamples[x, y]
- [0120] mvL0, mvL1, refIdxL0, refIdxL1, PredFlagL0, PredFlagL1을 아래 과정에 의해 유도한다.
- [0121] -만일 PredFlagL0가 1이고 PredFlagL1가 0이라면 아래를 수행한다.
- [0122] logWDC 값이 1 이상이면
- [0123]
$$\text{predSamples}[x, y] = \text{ClipH}((\text{predSamplesL0}[x, y] * w0C + 2^{\log\text{WDC} - 1}) \gg \log\text{WDC}) + o0C$$
- [0124] logWDC 값이 1 미만이면
- [0125]
$$\text{predSamples}[x, y] = \text{ClipH}(\text{predSamplesL0}[x, y] * w0C + o0C)$$
- [0126] -그렇지 않고, 만일 PredFlagL0가 0이고 PredFlagL1가 1이라면 아래를 수행한다.
- [0127] logWDC 값이 1 이상이면
- [0128]
$$\text{predSamples}[x, y] = \text{ClipH}((\text{predSamplesL1}[x, y] * w1C + 2^{\log\text{WDC} - 1}) \gg \log\text{WDC}) + o1C$$
- [0129] logWDC 값이 1 미만이면
- [0130]
$$\text{predSamples}[x, y] = \text{ClipH}(\text{predSamplesL1}[x, y] * w1C + o1C)$$
- [0131] -그렇지 않고, 만일 PredFlagL0가 1이고 PredFlagL1가 1이고, 움직임 벡터 mvL0과 mvL1가 같고, RefPicOrderCnt(currPic, refIdxL0, L0)와 RefPicOrderCnt(currPic, refIdxL1, L1)이 동일하다면(즉, 참조픽처가 동일하다면) 아래를 수행한다.
- [0132] logWDC 값이 1 이상이면
- [0133]
$$\text{predSamples}[x, y] = \text{ClipH}((\text{predSamplesL0}[x, y] * w0C + 2^{\log\text{WDC} - 1}) \gg \log\text{WDC}) + o0C$$
- [0134] logWDC 값이 1 미만이면
- [0135]
$$\text{predSamples}[x, y] = \text{ClipH}(\text{predSamplesL0}[x, y] * w0C + o0C)$$
- [0136] -그렇지 않다면 아래를 수행한다.
- [0137]
$$\text{predSamples}[x, y] = \text{ClipH}((\text{predSamplesL0}[x, y] * w0C + \text{predSamplesL1}[x, y] * w1C + 2^{\log\text{WDC}}) \gg (\log\text{WDC} + 1)) + ((o0C + o1C + 1) \gg 1)$$
- [0138] [방법2] 움직임 결합(Merge) 과정에서 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법
- [0139] L0와 L1의 움직임 정보가 같은 블록이 발생되면, 해당 블록이 다음에 부호화되는 블록에 영향을 미친다. 그러한 영향을 받는 예로 움직임 정보 결합 방법에 의해서 해당 블록이 전파되게 된다.
- [0140] 움직임 결합(Merge) 방법은 현재 블록의 주변 혹은 대응되는 블록의 움직임 정보가 존재하면 해당 블록의 움직임 정보를 현재 블록에 대한 움직임 정보 후보로 등록한다. 주변 혹은 대응되는 블록의 움직임 정보들이 모두 등록되면, 그것들 중에서 가장 부호화 효율 높은 것이 선택되어 현재 블록에 대한 움직임 정보로 사용된다. 그

리고 선택된 움직임 정보 후보에 대한 번호는 비트스트림에 포함된다.

- [0141] 움직임 정보 결합(Merge) 방법은 주변 혹은 대응되는(Collocated) 블록의 L0/L1 움직임 정보(참조픽처번호, 움직임 벡터) 모두를 현재 블록의 움직임 정보로 그대로 사용하는 방법이다. 그 결과 L0와 L1의 움직임 정보가 같은 블록이 더욱 많이 발생하게 된다. L0와 L1의 움직임 정보가 같은 블록이 다음 블록으로 전파되는 것을 막기 위하여 움직임결합(Merge) 과정에서 움직임 정보를 유추하는 부분을 수정함으로써 해결할 수 있다.
- [0142] 즉, 움직임 결합(Merge) 과정에서 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법을 이용하는 것이다. 해당 방법에 대한 순서도의 일 예는 [도 8]과 같다.
- [0144] 도 8. 움직임 결합(Merge) 과정에서 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법2에 대한 순서도의 일 예
- [0145] [도 8]은 움직임 결합(Merge) 과정에서 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법에 대한 순서도의 일 예이다.
- [0146] [도 8]은 움직임 결합(Merge) 과정에서 현재 블록의 주변 혹은 대응되는 블록의 움직임 정보를 현재 블록에 대한 움직임 정보 후보로 등록되는 과정을 보여준다. [도 8]에서 주변 움직임 정보는 현재 블록에 인접한 주변 블록의 움직임 정보를 말하며, 대응되는 블록이란, 참조픽처에서 현재 블록과 동일한 위치에 있는 블록이거나 혹은 그 블록(참조픽처에서 현재 블록과 동일한 위치에 있는 블록) 주변의 블록들을 의미한다. 움직임 정보에는 예측방향정보(단방향인지 혹은 양방향인지)와 L0와 L1에 대한 참조픽처번호와 움직임 벡터 등을 포함한다.
- [0147] 먼저, 현재 블록에 대한 움직임 정보 후보의 예측 방향(단방향인지 혹은 양방향인지)을 주변 혹은 대응되는 블록의 예측 방향으로 설정한다. 다음으로 현재 블록의 주변 혹은 대응되는 블록의 L0 움직임 정보를 현재 블록에 대한 L0 움직임 정보 후보로 등록한다. 현재 픽처가 B 픽처(양방향 예측 픽처)일 경우, L1에 대한 움직임 정보 후보 등록이 진행된다. 여기서 현재 블록의 주변 혹은 대응되는 블록의 L0 움직임 정보와 L1 움직임 정보가 같지 않다면, 그 블록의 L1 움직임 정보를 현재 블록에 대한 L1 움직임 정보 후보로 등록한다. 그렇지 않다면(현재 블록의 주변 혹은 대응되는 블록의 L0 움직임 정보와 L1 움직임 정보가 같다면), 그 블록의 L1 움직임 정보를 현재 블록에 대한 L1 움직임 정보 후보로 등록 하지 않고 현재 블록에 대한 움직임 정보 후보의 예측 방향을 단방향 예측으로 설정한다.
- [0149] [방법3] 현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트(현재 블록에 대한 예측 방향) 정보에 따라 움직임 정보를 설정하는 방법
- [0150] 시간적으로 움직임 벡터를 예측하는 방법은 참조픽처에서 현재 블록에 대응되는(Collocated) 블록의 움직임 정보를 이용하는 방법이다. 만일 현재 블록의 L0에 해당하는 움직임 정보를 시간적으로 유추할 때, 참조픽처에서 현재 블록에 대응되는 블록의 L0 움직임 정보를 사용한다. 그런데 만일 참조픽처에서 현재 블록에 대응되는 블록의 L0 움직임 정보가 없다면 대응되는 블록의 L1 움직임 정보를 이용하여 현재 블록의 L0 움직임 정보로 사용한다. 반대로 현재 블록의 L1에 해당하는 움직임 정보를 시간적으로 유추할 때, 참조픽처에서 현재 블록에 대응되는 블록의 L1 움직임 정보를 사용한다. 그런데 만일 참조픽처에서 현재 블록에 대응되는 블록의 L1 움직임 정보가 없다면 대응되는 블록의 L0 움직임 정보를 이용하여 현재 블록의 L1 움직임 정보로 사용한다. 그 결과 현재 블록의 L0움직임 정보와 L1 움직임 정보가 같아지는 현상이 발생하게 된다.
- [0151] 이러한 문제를 해결하기 위하여 현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트(현재 블록에 대한 예

측 방향) 정보에 따라 움직임 정보를 설정하는 방법을 제안하며, 해당 방법3에 대한 순서도의 일예는 [도 9]과 같다.

- [0153] 도 9. 현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트에 따라 움직임 정보를 설정하는 방법3에 대한 순서도의 일 예
- [0154] [도 9]은 현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트(현재 블록에 대한 예측 방향) 정보에 따라 움직임 정보를 설정하는 방법에 대한 순서도의 일 예이다. 여기서 대응되는 블록이란, 참조픽처에서 현재 블록과 동일한 위치에 있는 블록이거나 혹은 그 블록(참조픽처에서 현재 블록과 동일한 위치에 있는 블록) 주변의 블록들을 의미한다.
- [0155] 먼저, 대응되는 블록의 L0 움직임 정보가 존재한다면, 그 블록의 L0 움직임 정보를 현재 블록에 대한 L0 움직임 정보로 설정한다. 그렇지 않다면(대응되는 블록의 L0 움직임 정보가 존재하지 않는다면), 현재 블록에 대한 L0 움직임 정보를 설정하지 않을 수 있다. 혹은 대응되는 블록의 L1 움직임 정보를 현재 블록에 대한 L0 움직임 정보로 사용할 수 있다. 혹은 현재 블록에 대한 L0 움직임 정보에 움직임 벡터 (0,0)을 설정할 수 있다.
- [0156] 다음으로 현재 픽처가 B 픽처(양방향 예측 픽처)일 경우, 현재 블록에 대한 L1 움직임 정보 설정이 진행된다. 여기서 대응되는 블록의 L1 움직임 정보가 존재한다면, 그 블록의 L1 움직임 정보를 현재 블록에 대한 L1 움직임 정보로 설정한다. 그렇지 않다면(대응되는 블록의 L1 움직임 정보가 존재하지 않는다면), 현재 블록에 대한 L1 움직임 정보를 설정하지 않을 수 있다. 혹은 L1 움직임 정보에 움직임 벡터 (0,0)을 설정할 수 있다.
- [0158] [방법4] 시간적으로 유추된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법
- [0159] 시간적으로 유추된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법4에 대한 순서도의 일 예는 [도 10]와 같다.
- [0161] 도 10. 시간적으로 유추된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법4에 대한 순서도의 일 예
- [0162] [도 10]에서 시간적으로 유추된 움직임 정보는 참조픽처에서 현재 블록과 동일한 위치 혹은 그 주변의 위치에 있는 블록의 움직임 정보를 말한다. 움직임 정보에는 예측방향정보(단방향인지 혹은 양방향인지)와 L0와 L1에 대한 참조픽처번호와 움직임 벡터 등을 포함한다.
- [0163] 시간적으로 유추된 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같다면, 현재 블록에 대한 움직임 정보를 단방향 예측으로 설정하고 시간적으로 유추된 움직임 정보에서 L0움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하거나 혹은 현재 블록에 대한 움직임 정보 후보로 등록한다. 그렇지 않다면(시간적으로 유추된 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같지 않다면), 시간적으로 유추된 움직임 정보를 그대로 현재 블록에 대한 움직임 정보로 사용하거나 혹은 현재 블록에 대한 움직임 정보 후보로 등록한다.
- [0164] 시간적으로 유추된 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같은 경우가 발생하는 것을 해결하기 위한 또 다른 방법은 시간적으로 유추된 움직임 정보를 모두 단방향 예측으로 설정하고 L0움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하거나 혹은 현재 블록에 대한 움직임 정보 후보로 등록하여 사용할 수 있다. 단, L0의 참조픽처리스트와 L1의 참조픽처리스트가 같을 경우(예를 들어, 순방향 B 픽처일 경우)에만 이러한 방

법을 사용할 수 있다.

- [0165] 현재 블록에 대한 움직임 정보 후보로 등록하여 사용할 수 있다. 단, L0의 참조픽처리스트와 L1의 참조픽처리스트가 같을 경우(예를 들어, 순방향 B 픽처일 경우)에만 이러한 방법을 사용할 수 있다.
- [0167] [방법5] 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 현재 블록에 움직임 정보를 설정하는 방법
- [0168] 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 현재 블록에 움직임 정보를 설정하는 방법5의 일 예는 [도 11]과 같다.
- [0170] 도 11. 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 현재 블록에 움직임 정보를 설정하는 방법5에 대한 순서도의 일 예
- [0171] [도 11]는 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 현재 블록에 움직임 정보를 설정하는 방법5에 대한 순서도이며, 자세한 알고리즘은 다음과 같다.
- [0172] Step 1)만일 현재 픽처가 P 픽처이면 Step 2로 분기한다. 만약 그렇지 않으면(B픽처라면) Step 3으로 분기한다.
- [0173] Step 2)현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 '0' 으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정한다. L0 참조픽처번호를 '0' 으로 설정한다는 것은 L0참조픽처리스트에서 맨처음에 위치한 픽처를 참조픽처로 사용하겠다는 의미이다.
- [0174] Step 3)만일 L0 참조픽처와 L1 참조픽처가 참조픽처들이 동일하면, Step 4로 분기한다. 만약 그렇지 않으면 Step 5로 분기한다. 여기서 L0 참조픽처와 L1 참조픽처의 동일성을 판단하기 위하여 L0 참조픽처리스트에서 1번째 참조픽처의 픽처번호와 L1 참조픽처리스트에서 1번째 참조픽처의 픽처번호가 같은지를 판단하여 동일성을 판단할 수 있다. 또한 L0 참조픽처번호와 L1 참조픽처번호가 같은지를 판단하여 동일성을 판단할 수도 있다. 참조픽처의 동일성을 판단하는 방법은 상황에 따라 다양하게 변경될 수 있다.
- [0175] Step 4)현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 '0' 으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정한다.
- [0176] Step 5)현재 블록을 양방향 예측으로 설정하고 L0와 L1 참조픽처번호는 '0' 으로 설정하고 L0와 L1 움직임 벡터는 (0,0)으로 설정한다.
- [0177] 아래는 실제 WD3: Working Draft 3 of High-Efficiency Video Coding[JCTVC-E603]에서 해당 되는 부분을 변경한 것이다.
- [0178] 현재 블록의 주변 블록 혹은 대응되는 블록에 움직임 정보가 없을 경우, mvLX, refIdxLX, predFlagLX는 픽처의 형태에 따라 달라진다. 여기서 'X' 는 '0' 혹은 '1' 로 대체될 수 있다. 예를 들어 X가 '0' 일 경우에는 mvL0, refIdxL0, predFlagL0를 나타내며 이는 L0 움직임 정보와 관련된 변수들을 의미한다. mvLX는 움직임 벡터(x성분, y성분)를 말하며 mvLX[0]은 x성분의 움직임 벡터를 의미하며, mvLX[1]은 y성분의 움직임 벡터를 의미한다. refIdxLX는 참조픽처가 저장된 LX 참조픽처리스트에서의 인덱스를 나타내며, '0' 일 경우에는 LX참조픽처리스트에서 맨 앞에 있는 것을 나타내고 '-1' 일 경우에는 참조픽처가 없다는 것을 나타낸다. predFlagLX는 움직임 보상을 수행하여 예측된 블록을 생성하는지에 대한 여부를 나타내며, '1' 일 경우에는 움직임 보상을 수

행한다.

[0179] 만약, 픽처 형태가 P 픽처(순방향 예측 픽처)라면, 아래 조건을 수행한다.

[0180] $mvLX[0] = 0$

[0181] $mvLX[1] = 0$

[0182] $refIdxL0 = 0$

[0183] $refIdxL1 = -1$

[0184] $predFlagL0 = 1$

[0185] $predFlagL1 = 0$

[0186] 그렇지 않지 않다면(픽처 형태가 B 픽처라면), 아래 조건을 수행한다.

[0187] 만약, L0 참조픽처리스트의 1번째 참조픽처와 L1 참조픽처의 1번째 참조픽처가 동일하다면, 아래 조건을 수행한다.

[0188] $mvLX[0] = 0$

[0189] $mvLX[1] = 0$

[0190] $refIdxL0 = 0$

[0191] $refIdxL1 = -1$

[0192] $predFlagL0 = 1$

[0193] $predFlagL1 = 0$

[0194] 그렇지 않으면, 아래 조건을 수행한다.

[0195] $mvLX[0] = 0$

[0196] $mvLX[1] = 0$

[0197] $refIdxL0 = 0$

[0198] $refIdxL1 = 0$

[0199] $predFlagL0 = 1$

[0200] $predFlagL1 = 1$

[0201] [방법6] 최종적으로 결정된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법

[0202] 최종적으로 결정된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법6에 대한 순서도의 일 예는 [도 12]과 같다.

[0204] 도 12. 최종적으로 결정된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법6에 대한 순서도의 일 예

[0205] [도 12]은 최종적으로 결정된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법6에 대한 순서도의 일 예이다.

- [0206] [도 12]에서 움직임 정보에는 예측방향정보(단방향인지 혹은 양방향인지)와 L0와 L1에 대한 참조픽처번호와 움직임 벡터 등을 포함한다. 먼저, 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같지 않다면, L0와 L1의 움직임 정보를 현재 블록에 사용한다. 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같다면, 움직임 정보를 단방향 예측으로 설정하고 L0에 대한 움직임 정보만을 현재 블록에 사용한다.
- [0208] [방법7] 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 순방향 B픽처에서 현재 블록에 대한 움직임 정보를 설정하는 방법
- [0209] 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 순방향 B픽처에서 현재 블록에 대한 움직임 정보를 설정하는 방법7의 일 예는 [도 13]와 같다.
- [0211] 도 13. 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 순방향 B픽처에서 현재 블록에 대한 움직임 정보를 설정하는 방법7에 대한 순서도의 일 예
- [0212] [도 13]는 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 순방향 B픽처에서 현재 블록에 대한 움직임 정보를 설정하는 방법7에 대한 순서도이며, 자세한 알고리즘은 다음과 같다.
- [0213] Step 1)만일 현재 픽처가 P 픽처이면 Step 2로 분기한다. 만약 그렇지 않으면(픽처가 B 픽처라면) Step 3으로 분기한다.
- [0214] Step 2)현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 '0' 으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정한다. L0 참조픽처번호를 '0' 으로 설정한다는 것은 L0참조픽처리스트에서 맨처음에 위치한 픽처를 참조픽처로 사용하겠다는 의미이다.
- [0215] Step 3)만일 L0 참조픽처와 L1 참조픽처가 동일하면, Step 4로 분기한다. 만약 그렇지 않으면 Step 5로 분기한다. 여기서 L0 참조픽처와 L1 참조픽처의 동일성을 판단하기 위하여 L0 참조픽처리스트에서 1번째 참조픽처의 픽처번호와 L1 참조픽처리스트에서 1번째 참조픽처의 픽처번호가 같은지를 판단하여 동일성을 판단할 수 있다. 또한 L0 참조픽처번호와 L1 참조픽처번호가 같은지를 판단하여 동일성을 판단할 수도 있다. 참조픽처의 동일성을 판단하는 방법은 상황에 따라 다양하게 변경될 수 있다. 여기서 만약 참조픽처리스트 안의 참조픽처가 1개 이하라면 Step 6으로 분기할 수 있다.
- [0216] Step 4)현재 블록을 양방향 예측으로 설정하고 L0 참조픽처번호는 '0' 으로 설정하고 L1 참조픽처번호는 '1' 로 설정하고 L0와 L1 움직임 벡터는 (0,0)으로 설정한다.
- [0217] Step 5)현재 블록을 양방향 예측으로 설정하고 L0와 L1 참조픽처번호는 '0' 으로 설정하고 L0와 L1 움직임 벡터는 (0,0)으로 설정한다.
- [0218] Step 6)현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 '0' 으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정한다.
- [0219] 아래는 실제 WD3: Working Draft 3 of High-Efficiency Video Coding[JCTVC-E603]에서 해당 되는 부분을 변경한 것이다.
- [0220] 현재 블록의 주변 블록 혹은 대응되는 블록에 움직임 정보가 없을 경우, mvLX, refIdxLX, predFlagLX는 픽처의 형태에 따라 달라진다. 여기서 'X' 는 '0' 혹은 '1' 로 대체될 수 있다. 예를 들어 X가 '0' 일 경우에는

mvL0, refIdxL0, predFlagL0를 나타내며 이는 L0 움직임 정보와 관련된 변수들을 의미한다. mvLX는 움직임 벡터 (x성분, y성분)를 말하며 mvLX[0]은 x성분의 움직임 벡터를 의미하며, mvLX[1]은 y성분의 움직임 벡터를 의미한다. refIdxLX는 참조픽처가 저장된 LX 참조픽처리스트에서의 인덱스를 나타내며, '0' 일 경우에는 LX참조픽처 리스트에서 맨 앞에 있는 것을 나타내고 '-1' 일 경우에는 참조픽처가 없다는 것을 나타낸다. predFlagLX는 움직임 보상을 수행하여 예측된 블록을 생성하는지에 대한 여부를 나타내며, '1' 일 경우에는 움직임 보상을 수행한다.

[0221] 만약, 픽처 형태가 P 픽처(순방향 예측 픽처)라면, 아래 조건을 수행한다.

[0222] mvLX[0] = 0

[0223] mvLX[1] = 0

[0224] refIdxL0 = 0

[0225] refIdxL1 = -1

[0226] predFlagL0 = 1

[0227] prefFlagL1 = 0

[0228] 그렇지 않다면(픽처 형태가 B 픽처), 아래 조건을 수행한다.

[0229] 만약, L0 참조픽처리스트의 1번째 참조픽처와 L1 참조픽처의 1번째 참조픽처가 동일하다면, 아래 조건을 수행한다.

[0230] mvLX[0] = 0

[0231] mvLX[1] = 0

[0232] refIdxL0 = 0

[0233] refIdxL1 = 1

[0234] predFlagL0 = 1

[0235] prefFlagL1 = 1

[0236] 그렇지 않으면, 아래 조건을 수행한다.

[0237] mvLX[0] = 0

[0238] mvLX[1] = 0

[0239] refIdxL0 = 0

[0240] refIdxL1 = 0

[0241] predFlagL0 = 1

[0242] prefFlagL1 = 1

[0243] [방법8] 현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B 픽처에서 찾으려는 현재 블록의 리스트(예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법

[0244] 현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B 픽처에서 찾으려는 현재 블록의 리스트(예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법8에 대한 순서도의 일예는 [도 14]과 같다.

[0246] 도 14. 현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B 픽처에서 찾으려는 현재 블록의 리스트(예

측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법8에 대한 순서도의 일 예

- [0247] [도 14]은 현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B 픽처에서 찾으려는 현재 블록의 리스트 (예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법에 대한 순서도의 일 예이다. 여기서 대응되는 픽처란, 이미 부호화 및 복호화 되어 참조픽처리스트안에 참조픽처로서 존재하는 픽처를 말하며, 참조픽처리스트안의 모든 픽처들이 대응되는 픽처로 사용될 수 있다. 일반적으로 현재 부호화하려는 픽처와 픽처번호(POC) 순서상 가장 차이가 적은 픽처를 대응되는 픽처로 설정한다.
- [0248] 자세한 알고리즘은 다음과 같다.
- [0249] Step 1)만일 현재 블록에서 L1움직임 정보를 시간적으로 유추하려고 한다면, Step 2로 분기한다. 만약 그렇지 않으면 Step 5로 분기한다.
- [0250] Step 2)만일 L0 참조픽처와 L1참조픽처가 동일하다면, Step 3으로 분기한다. 만약 그렇지 않다면 Step 4로 분기한다. 여기서 참조픽처리스트의 참조픽처 개수가 1개이하일 경우에는 Step 4로 분기할 수도 있다.
- [0251] Step 3)L1 참조픽처리스트에서 L0의 참조픽처와 다른 참조픽처를 대응되는 픽처로 설정한다.
- [0252] Step 4)L1 참조픽처리스트의 1번째 참조픽처를 대응되는 픽처로 설정한다.
- [0253] Step 5)L0 참조픽처리스트의 1번째 참조픽처를 대응되는 픽처로 설정한다.

- [0254] 상기 여러 방법들은 아래 표 1과 같이 각 블록의 부호화 모드에 따라 적용할 수 있는 방법들의 조합은 여러가지가 될 수 있다.

- [0255] 표 1. 각 블록의 부호화 모드에 따라 적용할 수 있는 방법들의 조합

	조합 1	조합 2	조합 3	조합 4	조합 5	조합 6
움직임 결합 블록 (Merge)	방법 2	방법4	방법 4 방법 5	방법 3	방법 8 방법 5	방법 8 방법 7
움직임 결합이 아닌 블록(Inter)	방법 1	방법1	방법 6	방법 6	방법 1	방법 6

- [0256]
- [0257] 예를 들어, 조합 1의 경우, 움직임 결합 블록에서는 방법2의 “주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법”을 적용하고 움직임 결합이 아닌 블록에서는 방법1의 “두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행하는 방법”을 조합하여 사용할 수 있다.
- [0258] 조합 2의 경우, 움직임 결합 블록에서는 방법4의 “시간적으로 유추된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법”을 적용하고 움직임 결합이 아닌 블록에서는 방법1의 “두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행하는 방법”을 조합하여 사용할 수 있다.
- [0259] 조합 3의 경우, 움직임 결합 블록에서는 방법4의 “시간적으로 유추된 두 리스트(L0/L1)의 움직임 정보(참조픽

처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법”과 방법5의 “현재 블록의 주변 블록 혹은 대응되는 블록에 움직임 정보가 없을 경우, (0,0) 움직임 정보를 사용하는 방법”을 적용하고 움직임 결합이 아닌 블록에서는 방법6의 “두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법”을 조합하여 사용할 수 있다.

[0260] 조합 4의 경우, 움직임 결합 블록에서는 방법3의 “현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트에 따라 움직임 정보를 설정하는 방법”을 적용하고 움직임 결합이 아닌 블록에서는 방법6의 “두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법”을 조합하여 사용할 수 있다.

[0261] 조합 5의 경우, 움직임 결합 블록에서는 방법8의 “현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B픽처에서 찾으려는 현재 블록의 리스트(예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법”과 방법 5의 “현재 블록의 주변 블록 혹은 대응되는 블록에 움직임 정보가 없을 경우, (0,0) 움직임 정보를 사용하는 방법”을 적용하고 움직임 결합이 아닌 블록에서는 방법1의 “두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행하는 방법”을 조합하여 사용할 수 있다.

[0262] 조합 6의 경우, 움직임 결합 블록에서는 방법8의 “현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B픽처에서 찾으려는 현재 블록의 리스트(예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법”과 방법 7의 “현재 블록의 주변 블록 혹은 대응되는 블록에 움직임 정보가 없을 경우, 순방향 B픽처에서 (0,0) 움직임 정보를 사용하는 방법”을 적용하고 움직임 결합이 아닌 블록에서는 방법6의 “두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법”을 조합하여 사용할 수 있다.

[0263] [방법9] 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법

[0264] 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법9에 대한 순서도의 일예는 [도 15]과 같다.

[0266] 도 15. 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법 9에 대한 순서도의 일 예

[0267] [도 15]는 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법9에 대한 부호화/복호화 구조도의 일 예이다.

[0268] L0와 L1의 움직임 벡터와 참조픽처, 그리고 L1참조픽처리스트를 입력받아 현재 영상(블록)에 대한 움직임 보상된 영상(블록)을 출력하는 구조도의 일 예이다. 움직임 정보 및 참조픽처 판단 과정에서는 L0와 L1의 움직임 정보(움직임 벡터, 참조픽처)가 같은지를 판단하여 움직임 보상 과정을 선택적으로 수행하도록 한다. 만약 L0의 참조픽처와 L1의 참조픽처가 같고 L0 움직임 벡터와 L1 움직임 벡터가 같을 경우, L1참조픽처리스트에서 입력된 L1참조픽처와 다른 참조픽처를 가져와서 L1 참조픽처로 설정하고 L1 움직임 벡터를 그대로 사용할 수 있으며, 혹은 L1 참조 픽처에서 새로운 움직임 벡터를 찾아서 사용할 수 있다. 그 후 양방향 움직임 보상과정을 수행한다. 여기서 참조픽처리스트의 참조픽처가 1개 이하일 경우에는 단방향 움직임 보상이 수행된다. 그렇지 않으면 (L0의 참조픽처와 L1의 참조픽처가 같지 않거나 L0 움직임 벡터와 L1 움직임 벡터가 같지 않을 경우), 양방향 움직임 보상 과정을 수행한다. L0 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 또한 L0 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 그리고 L1 움직임 보상 과정에서는 참조영상 버퍼에서 입력받은 참조픽처와 L1 움직임 정보를 입력받아 L1 움직임 보상된 영상(블

록)을 출력한다. 가중치 평균 과정은 양방향 예측에서 수행되며, L0 움직임 보상 영상(블록)과 L1 움직임 보상 영상(블록)을 입력받아 가중치 평균을 수행하여 하나의 움직임 보상된 영상(블록)을 출력한다.

[0269] 이러한 방법을 가중치 예측 방법에도 동일하게 적용할 수 있다.

[0270] 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 가중치 예측에서 단방향 움직임 보상 과정을 수행함으로써 계산 복잡도를 감소시킬 수 있는 움직임 보상 방법의 부호화/복호화 구조도의 일 예는 [도 16]과 같다.

[0272] 도 16. 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법에 대한 순서도의 일 예

[0273] [도 16]은 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법에 대한 부호화/복호화 구조도의 일 예이다.

[0274] 우선 가중치 예측을 사용하지 않는 경우에는 방법 9를 사용하며, 가중치 예측을 사용할 경우에는 가중치 움직임 보상이 수행된다. 가중치 예측 움직임 보상을 수행할 때 방법 9를 이용하여 참조픽처를 선택적으로 사용할 수 있다. 즉 만약 L0의 참조픽처와 L1의 참조픽처가 같고 L0 움직임 벡터와 L1 움직임 벡터가 같을 경우, L1참조픽처리스트에서 입력된 L1참조픽처와 다른 참조픽처를 가져와서 L1 참조픽처로 설정하여 양방향 움직임 보상과정을 수행할 수 있다. 이외의 과정은 방법1의 가중치 움직임 보상 과정을 따른다.

[0276] [발명의 장치]

[0277] [방법1의 부호화/복호화 장치]

[0278] 본 발명에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 단방향 움직임 보상 과정을 수행함으로써 계산 복잡도를 감소시키는 방법1의 부호화/복호화 장치도는 [도 17]과 같다.

[0280] 도 17. 두 방향의 움직임 정보가 같을 경우, 움직임 보상 과정을 한번만 수행함으로써 계산 복잡도를 감소시킬 수 있는 방법의 부호화/복호화 장치의 일 예

[0281] [도 17]은 L0 움직임 정보(참조픽처번호, 움직임 벡터)와 L1 움직임 정보(참조픽처번호, 움직임 벡터)를 입력받아 현재 영상(블록)에 대한 움직임 보상된 영상(블록)을 출력하는 장치의 일 예이다. “움직임 정보 판단부”에서는 L0 움직임 정보(참조픽처번호, 움직임 벡터)와 L1 움직임 정보(참조픽처번호, 움직임 벡터)를 입력받아서 두 움직임 정보가 같은지를 판단하여, 같으면 “단방향 움직임 보상부”를 수행하고, 그렇지 않으면 “양방향 움직임 보상부”를 수행한다. “단방향 움직임 보상부”에서는 L0 움직임 정보만을 입력받아 단방향 움직임 보상을 수행하며, “양방향 움직임 보상부”에서는 L0와 L1 움직임 정보를 입력받아 양방향 움직임 보상을 수행한다. “단방향 움직임 보상부”의 “L0 움직임 보상부”에서는 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 또한 “양방향 움직임 보상부”의 “L0 HA 움직임 보상부”에서는 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 그리고 “양방향 움직임 보상부”의 “L1 HA 움직임 보상부”에서는 참조픽처와 L1 움직임 정보를 입력받아 L1 움직임 보상된 영상(블록)을 출력한다. “HA 가중치 평균부”에서는 L0 움직임 보상 영상(블록)과 L1 움직임 보상 영상(블록)을 입력받아 가중치 평균을 수행하여 하나의 움직임 보상된 영상(블록)을 출력한다.

[0282] [방법2의 부호화/복호화 장치]

- [0283] 본 발명에서 움직임 결합(Merge) 과정에서 현재 블록의 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법2의 부호화/복호화 장치도는 [도 18]과 같다.
- [0285] 도 18. 움직임 결합(Merge) 과정에서 주변 혹은 대응되는(Collocated) 블록으로부터 움직임 정보를 유추할 때, 주변 혹은 대응되는(Collocated) 블록에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보 후보로 등록하는 방법2의 부호화/복호화 장치의 일 예
- [0286] [도 18]에서 ‘대상 블록’이란, 현재 블록의 주변 블록이거나 혹은 참조픽처에서 현재 블록과 동일한 위치에 있는 블록이거나 혹은 그 블록(참조픽처에서 현재 블록과 동일한 위치에 있는 블록) 주변의 블록들을 의미한다. 그리고 예측방향정보는 단방향예측인지 혹은 양방향예측인지를 나타내며, 움직임 정보는 L0와 L1에 대한 참조픽처번호와 움직임 벡터 등을 말한다.
- [0287] [도 18]의 “L0 움직임 정보 후보 등록부”에서는 대상 블록의 L0 움직임 정보를 현재 블록에 대한 L0 움직임 정보 후보로 등록한다. “움직임 정보 판단부”에서는 대상 블록의 L0 움직임 정보와 L1 움직임 정보가 같은지를 판단한다. 만일 대상 블록의 L0 움직임 정보와 L1 움직임 정보가 같지 않다면, 대상 블록의 L1 움직임을 현재 블록에 대한 L1 움직임 정보 후보로 등록한다. 그렇지 않다면(대상 블록의 L0 움직임 정보와 L1 움직임 정보가 같다면), 대상 블록의 L1 움직임을 현재 블록에 대한 L1 움직임 정보 후보로 등록하지 않고 “예측 방향 재 설정부”가 수행된다. “예측 방향 재 설정부”는 대상 블록의 L0 움직임 정보와 L1 움직임 정보가 같을 경우에 수행되는 것으로 현재 블록에 대한 움직임 정보 후보의 예측 방향을 단방향 예측으로 설정하고 L0 움직임 정보만을 갖도록 한다.
- [0288] [방법3의 부호화/복호화 장치]
- [0289] 본 발명에서 현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트에 따라 움직임 정보를 설정하는 방법3의 부호화/복호화 장치도는 [도 19]과 같다.
- [0291] 도 19. 현재 블록의 움직임 정보를 시간적으로 유추할 때, 리스트에 따라 움직임 정보를 설정하는 방법3의 부호화/복호화 장치의 일 예
- [0292] [도 19]에서 대응되는 블록이란, 참조픽처에서 현재 블록과 동일한 위치에 있는 블록이거나 혹은 그 블록(참조픽처에서 현재 블록과 동일한 위치에 있는 블록) 주변의 블록들을 의미한다.
- [0293] [도 19]에서 “예측 방향 설정부”에서는 현재 블록에 대한 움직임 정보 후보의 예측 방향을 대상 블록의 예측 방향으로 설정한다. “L0 움직임 정보 판단부”는 대응되는 블록의 L0 움직임 정보가 존재하는지 판단한다. 대응되는 블록의 L0 움직임 정보가 존재한다면, “L0 움직임 정보 설정부”가 수행된다. 그렇지 않다면(대응되는 블록의 L0 움직임 정보가 존재하지 않는다면), L0움직임 정보를 설정하지 않을 수 있다. 혹은 대응되는 블록의 L1 움직임을 현재 블록에 대한 L0 움직임 정보로 사용할 수 있다. 혹은 현재 블록에 대한 L0 움직임 벡터를 (0,0)으로 설정할 수 있다. “L0 움직임정보 설정부”는 대응되는 블록의 L0 움직임을 현재 블록에 대한 L0 움직임 정보로 설정한다. “픽처 판단부”에서는 현재 픽처가 B 픽처(양방향 예측 픽처)인지를 판단한다. 만일 현재 픽처가 B 픽처(양방향 예측 픽처)일 경우, 다음 과정이 수행되며, 그렇지 않으면(만일 현재 픽처가 B 픽처(양방향 예측 픽처)가 아니라면), 종료한다. 혹은 현재 블록에 대한 움직임 정보를 단방향 예측으로 설정하고 종료할 수 있다. “L1 움직임 정보 판단부”는 대응되는 블록의 L1 움직임 정보가 존재하는지 판단한다. 대응

되는 블록의 L1 움직임 정보가 존재한다면, “L1 움직임 정보 설정부”가 수행된다. 그렇지 않다면(대응되는 블록의 L1 움직임 정보가 존재하지 않는다면), 현재 블록에 대한 L1 움직임 정보를 설정하지 않을 수 있다. 혹은 현재 블록에 대한 L1 움직임 정보에서 움직임 벡터를 (0,0)으로 설정할 수 있다. “L1 움직임정보 설정부”는 대응되는 블록의 L1 움직임 정보를 현재 블록에 대한 L1 움직임 정보로 설정한다.

[0294] [방법4의 부호화/복호화 장치]

[0295] 본 발명에서 시간적으로 유추된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법4의 부호화/복호화 장치도는 [도 20]과 같다.

[0297] 도 20. 시간적으로 유추된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법4의 부호화/복호화 장치의 일예

[0298] [도 20]에서 ‘시간적으로 유추된 움직임 정보’는 참조픽처에서 현재 블록과 동일한 위치 혹은 그 주변의 위치에 있는 블록의 움직임 정보를 말한다. 움직임 정보에는 예측방향정보(단방향인지 혹은 양방향인지)와 L0와 L1에 대한 참조픽처번호와 움직임 벡터 등을 포함한다.

[0299] “움직임 정보 판단부”에서는 시간적으로 유추된 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같은지를 판단한다. 시간적으로 유추된 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같다면, “예측방향 재설정부”와 “L0움직임정보설정부”가 수행된다. 시간적으로 유추된 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같지 않다면, 시간적으로 유추된 움직임 정보를 그대로 현재 블록에 대한 움직임 정보로 사용하거나 혹은 현재 블록에 대한 움직임 정보 후보로 등록한다. “예측 방향 재설정부”에서는 시간적으로 유추된 움직임 정보의 예측 방향을 단방향으로 설정한다. “L0 움직임 정보 설정부”에서는 시간적으로 유추된 움직임 정보에서 L0움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하거나 혹은 현재 블록에 대한 움직임 정보 후보로 등록한다.

[0300] [방법5의 부호화/복호화 장치]

[0301] 본 발명에서 현재 블록의 주변 블록 혹은 대응되는 블록에 움직임 정보가 없을 경우, (0,0) 움직임 정보를 사용하는 방법5의 부호화/복호화 장치도는 [도 21]와 같다.

[0303] 도 21. 현재 블록의 주변 블록 혹은 대응되는 블록에 움직임 정보가 없을 경우, (0,0) 움직임 정보를 사용하는 방법5의 부호화/복호화 장치의 일예

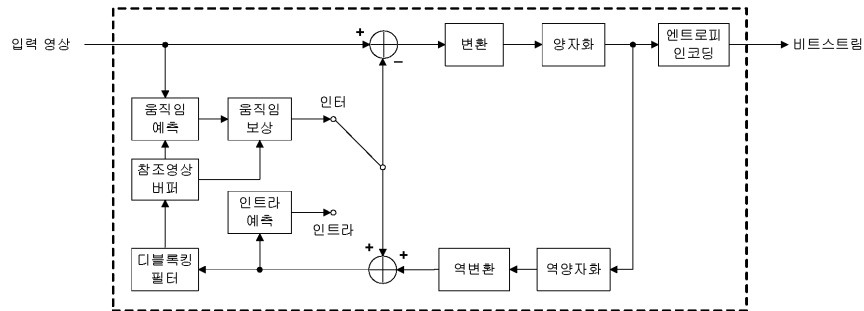
[0304] [도 21]에서 “판단부”에서는 픽처 형태와 L0 그리고 L1 참조픽처를 입력받아 상황에 따라 “단방향 움직임 정보 설정부” 혹은 “양방향 움직임 정보 설정부”를 수행시킨다. 만일 현재 픽처가 P 픽처이거나 혹은 현재 픽처가 B 픽처이고, L0 참조픽처와 L1 참조픽처가 동일하면, “단방향 움직임 정보 설정부”가 수행된다. 그렇지 않으면(현재 픽처가 현재 픽처가 B 픽처이고, L0 참조픽처와 L1 참조픽처가 동일하지 않으면), “양방향 움직임 정보 설정부”가 수행된다. “단방향 움직임 정보 설정부”에서는 현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 ‘0’으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정한다. “양방향 움직임 정보 설정부”에서는 현재 블록을 양방향 예측으로 설정하고 L0와 L1 참조픽처번호는 ‘0’으로 설정하고 L0와 L1 움직임 벡터는 (0,0)으로 설정한다.

- [0305] [방법6의 부호화/복호화 장치]
- [0306] 본 발명에서 최종적으로 결정된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법6의 부호화/복호화 장치도는 [도 22]과 같다.
- [0308] 도 22. 최종적으로 결정된 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, L0의 움직임 정보만을 현재 블록에 대한 움직임 정보로 사용하는 방법6의 부호화/복호화 장치의 일예
- [0309] [도 22]에서 움직임 정보에는 예측방향정보(단방향인지 혹은 양방향인지)와 L0와 L1에 대한 참조픽처번호와 움직임 벡터 등을 포함한다.
- [0310] 먼저, “움직임 정보 판단부”에서는 입력된 움직임 정보에서 L0 움직임 정보와 L1 움직임 정보가 같은지 판단한다. 만약 같다면 “예측방향재설정 및 L0 움직임정보 등록부”가 수행되고 만약 같지 않다면 “L0 L1 움직임 정보 등록부”가 수행된다. “예측방향재설정 및 L0 움직임정보 등록부”에서는 움직임 정보를 단방향 예측으로 설정하고 L0에 대한 움직임 정보만을 현재 블록에 사용한다. “L0 L1 움직임 정보 등록부”에서는 입력된 L0와 L1의 움직임 정보를 현재 블록에 사용한다.
- [0311] [방법7의 부호화/복호화 장치]
- [0312] 본 발명에서 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 순방향 B픽처에서 현재 블록에 대한 움직임 정보를 설정하는 방법 7의 부호화/복호화 장치도는 [도 23]와 같다.
- [0314] 도 23. 움직임 정보로 사용되는 모든 후보 블록들이 움직임 정보를 가지고 있지 않을 때, 순방향 B픽처에서 현재 블록에 대한 움직임 정보를 설정하는 방법 7의 부호화/복호화 장치의 일예
- [0315] [도 23]에서 “판단부”에서는 입력받은 픽처 형태와 L0 그리고 L1 참조픽처를 통해 “단방향 움직임 정보 설정부” 혹은 “양방향 움직임 정보 설정부” 혹은 “다른 참조픽처로 양방향 움직임 정보 설정부”를 수행시킨다. 만일 현재 픽처가 P 픽처라면, “단방향 움직임 정보 설정부”가 수행된다. 그렇지 않으면(현재 픽처가 B 픽처), 참조픽처의 동일성을 판단한다. 만약 L0 참조픽처와 L1 참조픽처가 동일하지 않으면, “양방향 움직임 정보 설정부”가 수행된다. 그렇지 않고 L0 참조픽처와 L1 참조픽처가 동일하면, “다른 참조픽처로 양방향 움직임 정보 설정부”가 수행된다. 여기서 참조픽처리스트 안의 참조픽처의 개수가 1개 이하이면, “단방향 움직임 정보설정부”가 수행된다. “단방향 움직임 정보 설정부”에서는 현재 블록을 단방향 예측으로 설정하고 L0 참조픽처번호는 ‘0’으로 설정하고 L0 움직임 벡터는 (0,0)으로 설정한다. “양방향 움직임 정보 설정부”에서는 현재 블록을 양방향 예측으로 설정하고 L0와 L1 참조픽처번호는 ‘0’으로 설정하고 L0와 L1 움직임 벡터는 (0,0)으로 설정한다. “다른 참조픽처로 양방향 움직임 정보 설정부”에서는 현재 블록을 양방향 예측으로 설정하고 L0 참조픽처번호는 ‘0’으로 설정하고 L1 참조픽처번호는 ‘1’로 설정하고 L0와 L1 움직임 벡터는 (0,0)으로 설정한다.
- [0316] [방법8의 부호화/복호화 장치]
- [0317] 본 발명에서 현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B 픽처에서 찾으려는 현재 블록의 리스트(예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법8의 부호화/복호화 장치도는 [도 24]과 같다.

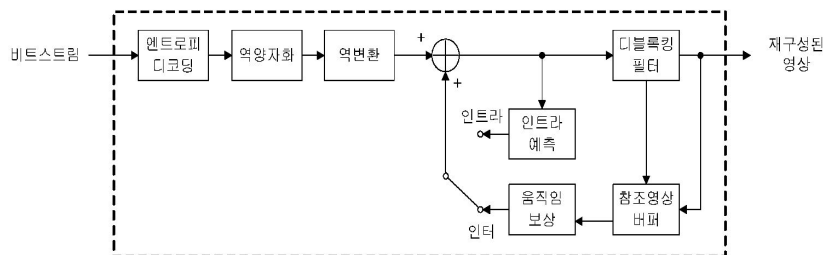
- [0319] 도 24. 현재 블록의 움직임 정보를 시간적으로 유추할 때, 순방향 B 픽처에서 찾으려는 현재 블록의 리스트(예측 방향) 정보에 따라 대응되는 픽처를 설정하는 방법8의 부호화/복호화 장치의 일예
- [0320] [도 24]은 현재 블록의 리스트(예측 방향) 정보와 L0/L1 참조픽처를 입력받아 현재 픽처에서 시간적으로 움직임을 유추할 때 사용할 대응되는 픽처를 설정한다. “예측 방향 판단부”에서는 현재 블록의 리스트(예측 방향) 정보가 L0리스트의 움직임 정보를 유추하는 것이라면 L0 참조픽처리스트의 1번째 참조픽처를 대응되는 픽처로 설정한다. 만약 현재 블록의 리스트(예측방향) 정보가 L1리스트의 움직임 정보를 유추하는 것이라면 “참조픽처 동일성 판단부”가 수행된다. “참조픽처 동일성 판단부”에서는 L0 참조픽처와 L1참조픽처가 동일한지 판단한다. 만일 L0 참조픽처와 L1참조픽처가 동일하지 않다면 L1 참조픽처리스트의 1번째 참조픽처를 대응되는 픽처로 설정한다. 만일 L0 참조픽처와 L1참조픽처가 동일하다면 L1 참조픽처리스트에서 L0참조픽처와 다른 참조픽처를 대응되는 픽처로 설정한다. 만일 참조픽처리스트의 개수가 1개 이하라면, L1 참조픽처리스트의 1번째 참조픽처를 대응되는 픽처로 설정한다.
- [0321] [방법9의 부호화/복호화 장치]
- [0322] 본 발명에서 두 리스트(L0/L1)의 움직임 정보(참조픽처번호, 움직임 벡터)가 같을 경우, 부호화 효율을 향상시키는 방법9의 부호화/복호화 장치도는 [도 25]과 같다.
- [0324] 도 25. 두 방향의 움직임 정보가 같을 경우, 움직임 보상 과정을 한번만 수행함으로써 계산 복잡도를 감소시킬 수 있는 방법의 부호화/복호화 장치의 일예
- [0325] [도 25]은 L0와 L1의 움직임 벡터와 참조픽처, 그리고 L1참조픽처리스트를 입력받아 현재 영상(블록)에 대한 움직임 보상된 영상(블록)을 출력하는 장치의 일 예이다. “움직임 정보 판단부”에서는 L0와 L1 움직임 벡터와 참조픽처, 그리고 L1참조픽처리스트를 입력받아서 상황에 따라 “단방향 움직임 보상부”를 수행하고, 그렇지 않으면 “양방향 움직임 보상부”를 수행한다. 만일 L0와 L1 움직임 벡터와 참조픽처가 같을 경우, L1참조픽처리스트에서 입력된 L1참조픽처와 다른 참조픽처를 가져와서 L1 참조픽처로 설정하고 양방향 움직임 보상부를 수행한다. 만일 L0와 L1 움직임 벡터가 같지 않거나 참조픽처가 같지 않을 경우에는 양방향 움직임 보상부가 수행된다. 만일 L0와 L1 움직임 벡터와 참조픽처가 같고 참조픽처리스트의 참조픽처가 1개 이하라면 단방향 움직임 보상부가 수행된다. “단방향 움직임 보상부”에서는 L0 움직임 정보만을 입력받아 단방향 움직임을 보상을 수행하며, “양방향 움직임 보상부”에서는 L0와 L1 움직임 정보를 입력받아 양방향 움직임을 보상을 수행한다. “단방향 움직임 보상부”의 “L0 움직임 보상부”에서는 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 또한 “양방향 움직임 보상부”의 “L0 HA 움직임 보상부”에서는 참조픽처와 L0 움직임 정보를 입력받아 L0 움직임 보상된 영상(블록)을 출력한다. 그리고 “양방향 움직임 보상부”의 “L1 HA 움직임 보상부”에서는 참조픽처와 L1 움직임 정보를 입력받아 L1 움직임 보상된 영상(블록)을 출력한다. “HA 가중치 평균부”에서는 L0 움직임 보상 영상(블록)과 L1 움직임 보상 영상(블록)을 입력받아 가중치 평균을 수행하여 하나의 움직임 보상된 영상(블록)을 출력한다.

도면

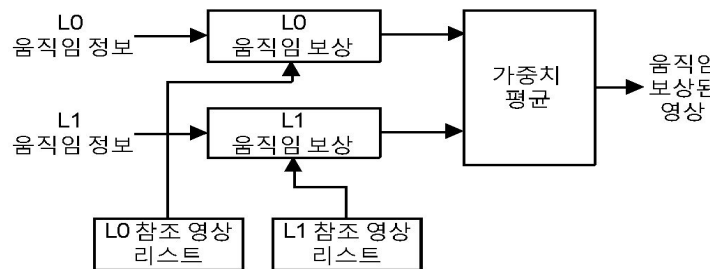
도면1



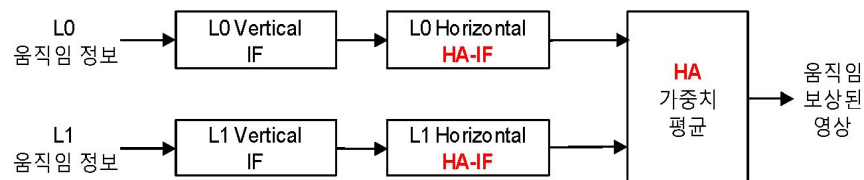
도면2



도면3



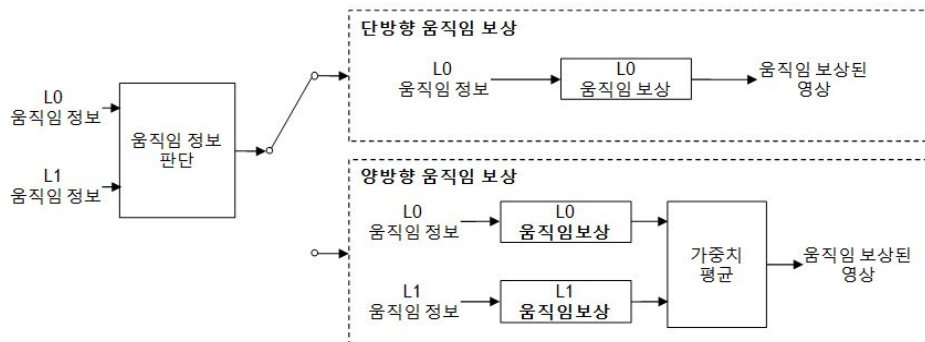
도면4



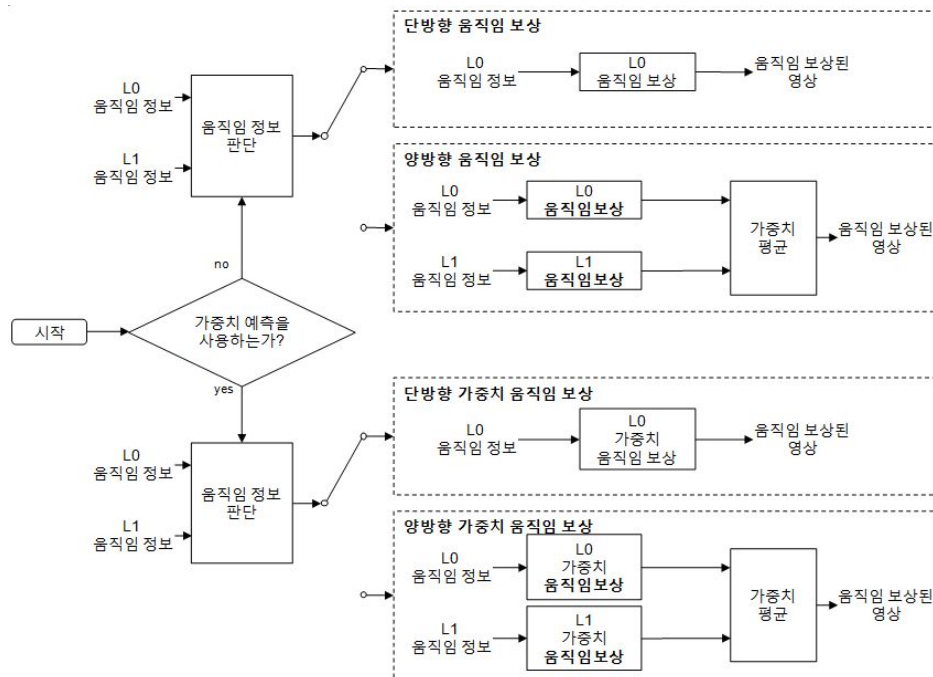
도면5



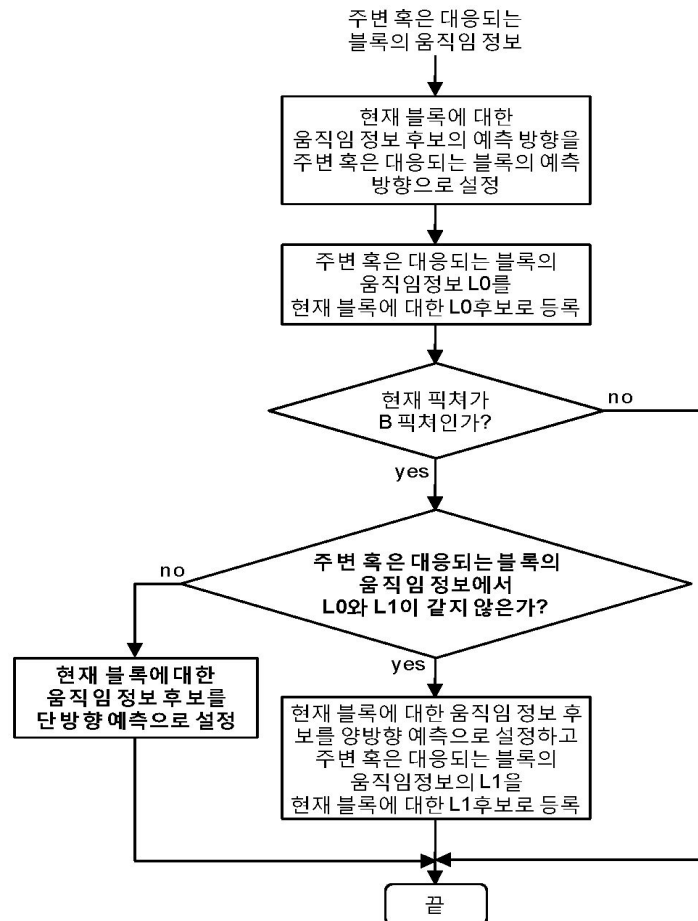
도면6



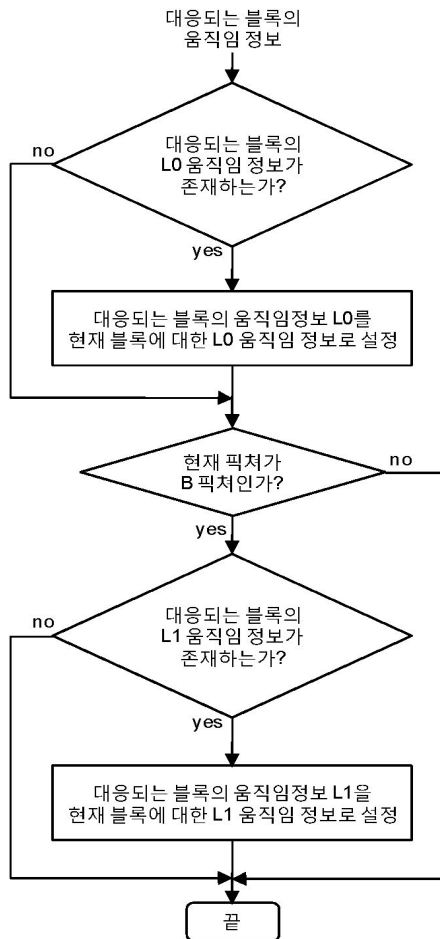
도면7



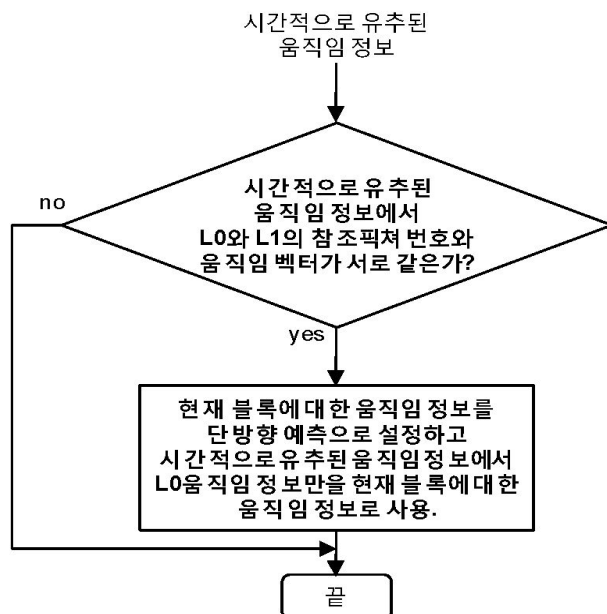
도면8



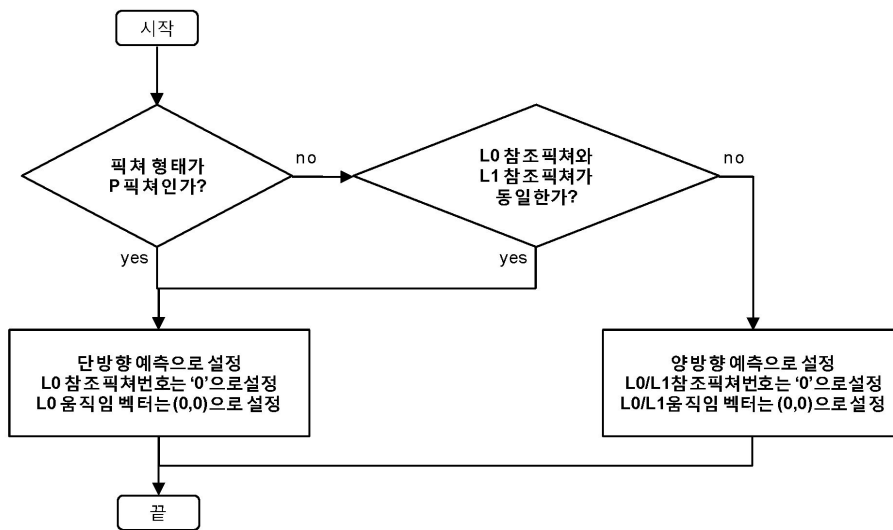
도면9



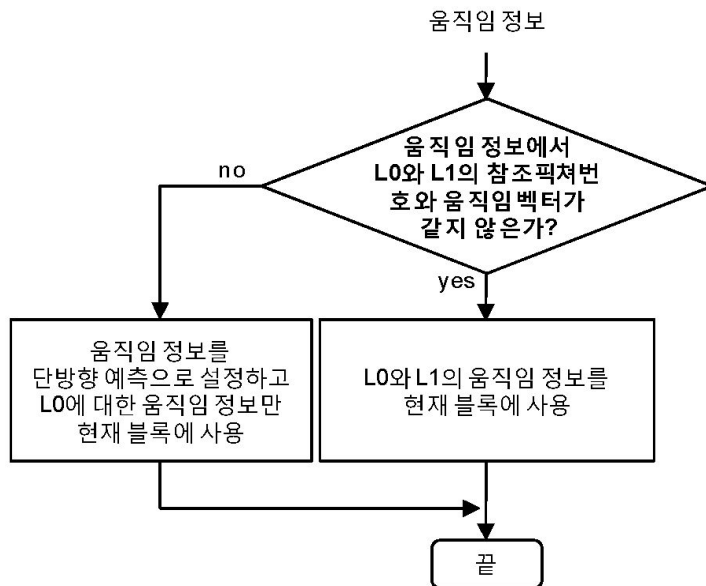
도면10



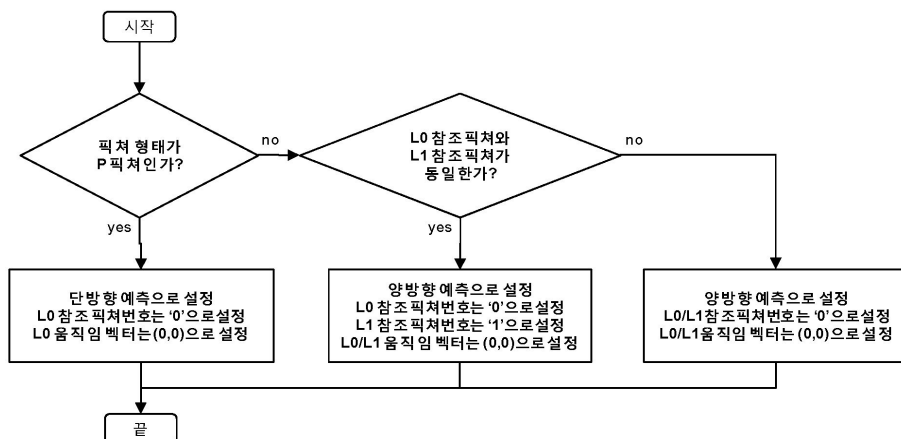
도면11



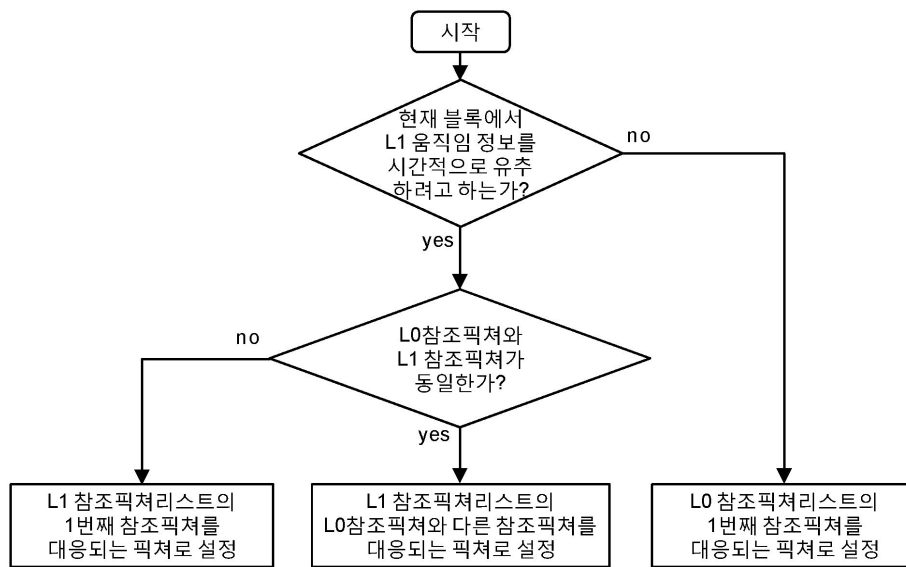
도면12



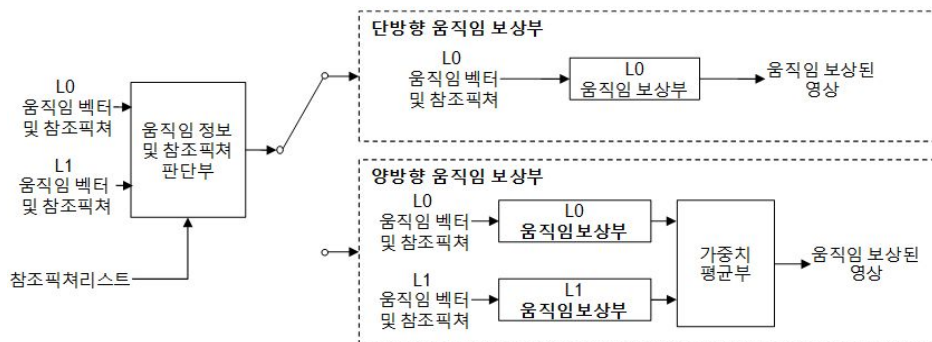
도면13



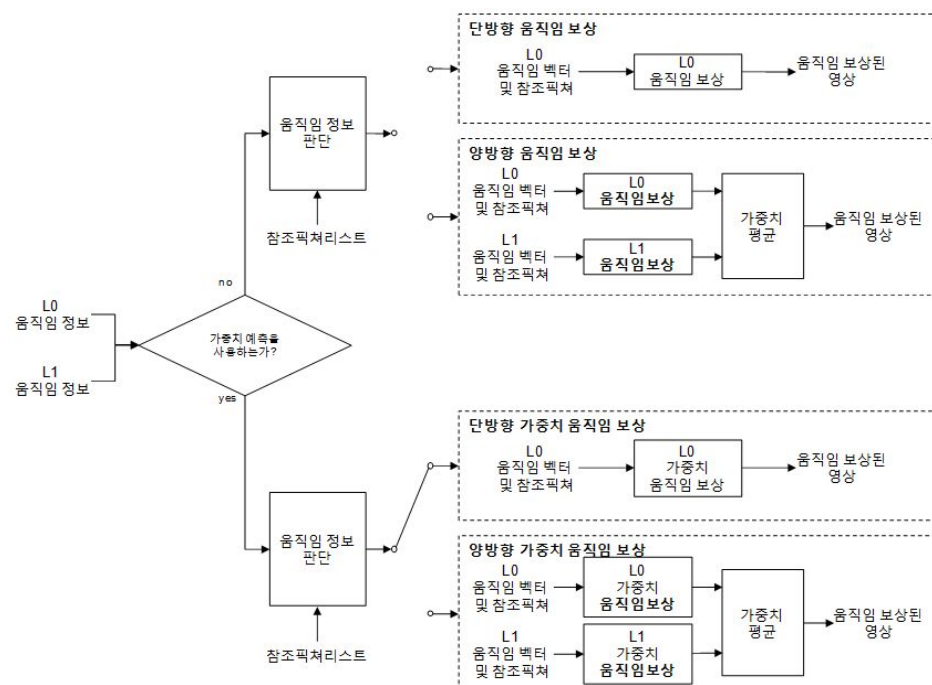
도면14



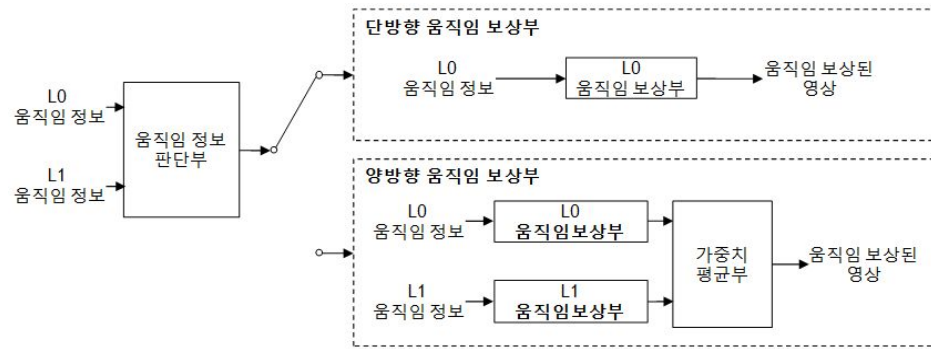
도면15



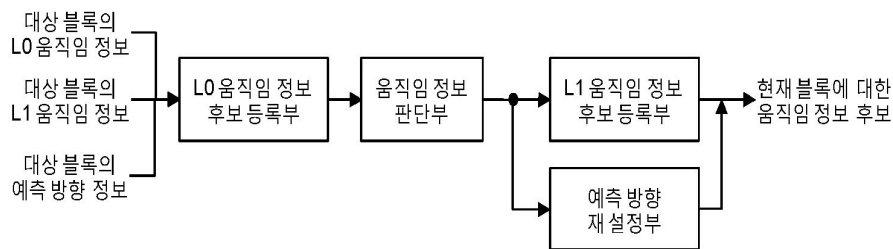
도면16



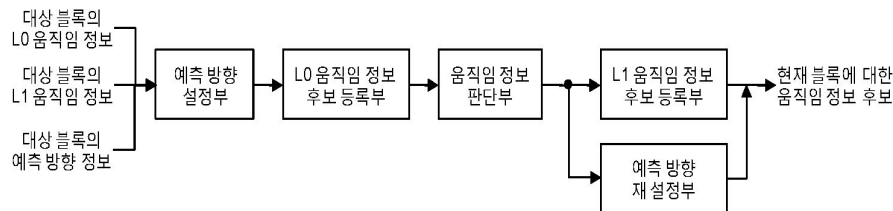
도면17



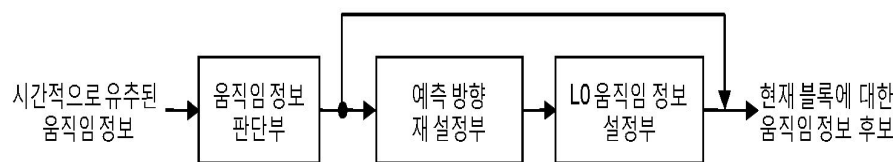
도면18



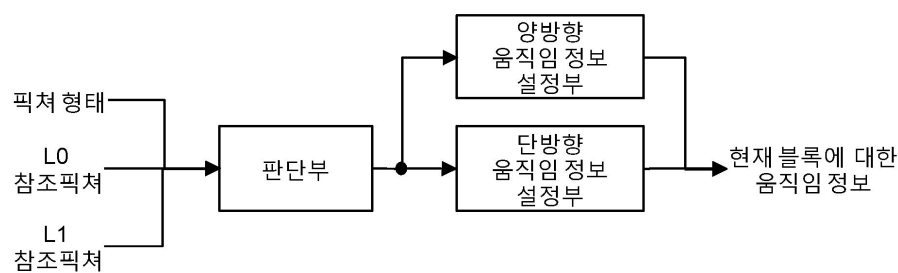
도면19



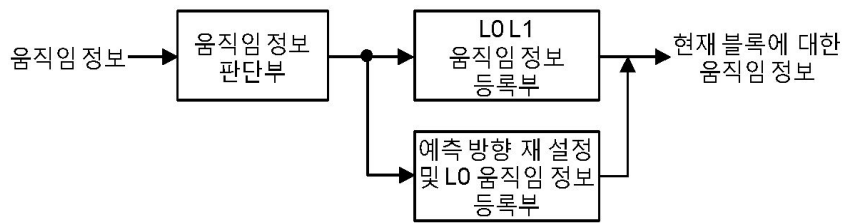
도면20



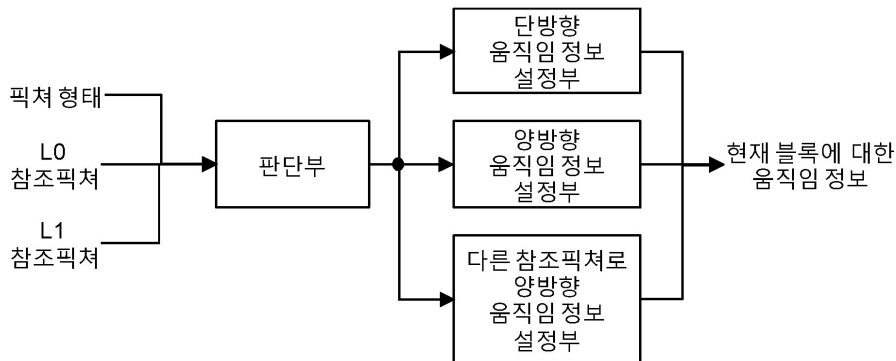
도면21



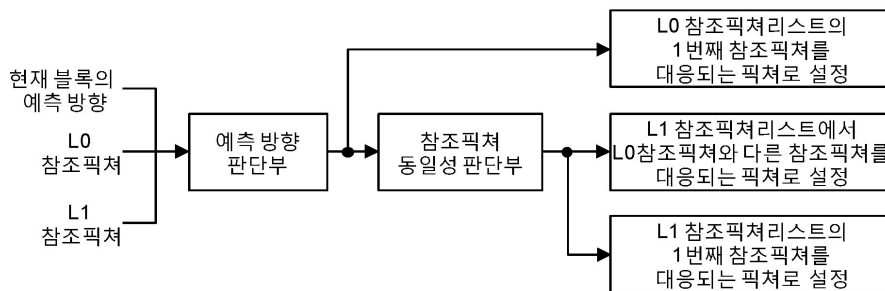
도면22



도면23



도면24



도면25

