

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl. ⁷ H04N 7/28	(45) 공고일자 2000년03월 15일	(11) 등록번호 10-0249223
(21) 출원번호 10-1997-0047149	(24) 등록일자 1999년12월23일	(65) 공개번호 특1999-0025488
(22) 출원일자 1997년09월 12일	(43) 공개일자 1999년04월06일	

(73) 특허권자	엘지전자주식회사 구자홍 서울특별시 영등포구 여의도동 20번지
(72) 발명자	김현문 서울특별시 강남구 수서동 708 삼익아파트 405-309 이영수 서울시강남구역삼1동661-2,303호 김성득 대전광역시 서구 월평동 1122 나중범 대전시유성구전민동464-1엑스포아파트404-506
(74) 대리인	심창섭, 김용인, 김용인, 심창섭, 강용복

심사관 : 정성중

(54) 엠팩(MPEG)-4의 움직임벡터코딩방법

요약

본 발명은 효율적인 움직임벡터 부호화(Coding)을 위해 3개의 후보 움직임벡터중 가장 적은 비트율을 갖는 것을 찾아 그 움직임벡터의 차분치와 모드비트를 디코더쪽으로 보냄으로써 전체적으로 비트율을 줄이기 위한 MPEG-4의 움직임벡터 부호화방법에 관한 것으로서, 3개의 움직임벡터 후보중에서 움직임벡터 코딩에 가장 적은 비트율을 가지는 하나의 후보를 선택하는 스텝과, 움직임벡터 예측에러와 예측모드 정보를 디코더쪽으로 전송하는 스텝과, 움직임벡터 차분치를 코딩하기 위한 최소비트율을 추측하는 스텝과, 최소율 예측을 이용하여 X 및 Y성분에 대한 움직임벡터를 코딩하는 스텝을 포함하여 이루어지는 것을 특징으로 한다.

대표도

도6

명세서

도면의 간단한 설명

제1도는 종래기술에 따른 8×8 모드(mode)로 선택된 매크로블록(macrobloc)의 4개의 블록 움직임벡터가 가리킬 수 있는 바운드(bound)를 나타낸 도면

제2도는 종래기술에 따른 8×8 모드 매크로블록에서의 움직임벡터 예측후보를 도시한 도면

제3도는 종래 움직임벡터의 중간값 예측에서 제한범위를 넘는 경우의 예를 나타낸 구성도

제4도는 본 발명의 움직임벡터 부호화방법에 따른 움직임벡터 부호화에서 MVD코드비트와 모드비트의 비트 스트림 신텍스(bitstream systanx)

제5a도 내지 제5b도는 본 발명의 움직임벡터 부호화에 따른 모드비트 감소의 일예를 엔코더 및 디코더측면에서 도시한 도면

제6도는 본 발명의 MPEG-4의 움직임벡터 부호화방법을 설명하기 위한 플로우차트

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 동영상의 부호화방법에 관한 것으로서 특히, 현재 표준화가 진행되고 있는 MPEG(Moving Picture Experts Group)-4 비디오 VM(Verification Model)에서 움직임벡터(Motion Vector)의 코딩효율성

을 향상시키는데 적당한 MPEG-4의 움직임벡터 코딩방법에 관한 것이다.

일반적으로 시간에 따라 변화하는 비디오 시퀀스(Video Sequence)를 효율적으로 압축하기 위해서는 영상 데이터가 갖고 있는 2차원 공간상의 중복성뿐만 아니라 시간축의 중복성제거가 절대적으로 필요하다.

MPEG에서는 2차원 공간상의 중복성을 제거하기 위하여 DCT(Discrete Cosine Transform)를 이용하여 시간축의 중복성을 제거하기 위해 움직임 보상방법을 이용한다.

DCT는 2차원 축변환을 통해서 데이터의 상관성을 제거하는 방법으로 픽처(Picture)를 블록(block)단위로 나눈 후 각각의 블록을 DCT식에 따라 축변환시킨다.

이렇게 축변환된 데이터들은 한쪽방향(저역쪽)으로 몰리는 경향이 있는데 이렇게 몰려진 데이터들만을 양자화해서 전송하게 된다.

시간축으로 연속된 픽처들은 주로 화면의 중앙부분에서 사람이나 물체의 움직임이 있기 때문에 움직임 보상방법에서는 이러한 성질을 이용하여 시간축의 중복성을 제거한다.

즉, 화면의 변하지 않는 부분이나 움직였다 하더라도 비슷한 부분을 바로 전 픽처에서 가져와서 채움으로써 전송해야 할 데이터량을 큰 폭으로 줄일 수 있다.

이렇게 픽처 사이에서 가장 비슷한 블록을 찾는 일을 움직임 예측(motion estimation)이라고 하며, 얼마만큼 움직였는지를 나타내는 변위를 움직임벡터라 한다.

다시말해서, 움직임벡터란 영상의 움직임 보상을 위해서 현재의 픽처(Picture)나 필드(Field)의 좌표로부터 기준프레임(reference frame)이나 기준필드의 좌표의 오프셋을 나타내주는 2차원 벡터를 말한다.

현재 표준화가 진행중인 MPEG-4에서 완전한 픽셀의 움직임 예측(integer pixel motion estimation)시에 수행되는 8×8 블록별 탐색은 16×16 움직임벡터를 중심으로 ± 2 픽셀(pixel)의 탐색 윈도우(search window)내에서 수행한다.

따라서, 특정 매크로블록(macroblock)이 8×8 모드로 선택되었을 때, 매크로블록내의 4블록에 대한 각 움직임벡터는 일정한 바운드내에 모두 들어오게 된다.

제1도는 종래기술에 따른 8×8 모드(mode)로 선택된 매크로블록(macroblock)의 4개의 블록 움직임벡터가 가리킬 수 있는 바운드(bound)를 나타내었다.

이러한 4개의 움직임벡터들의 바운드는 움직임벡터의 코딩효율을 올리는데 크게 기여한다.

움직임벡터는 매크로블록당 최대 4개까지 나올 수 있는데 이를 그냥 전송하게 되면 비트량이 많기 때문에 바로 전 매크로블록의 움직임벡터와의 차이만을 가변길이 부호화(VLC:Variable Length Coding)하여 전송한다.

이때 움직임 벡터의 차이를 MVD(Motion Vector Difference)로 표시한다.

제2a도~제2d도에서와 같이 각 블록 (가)에 해당하는 움직임 벡터(MV)의 X 및 Y성분들을 인접한 3개의 후보움직임 벡터(MV1, MV2, MV3)의 중간값을 예측값으로 하여 그들의 차성분(MVD_x, MVD_y)이 부호화된다.

이를 식으로 표현하면,

$$P_x = \text{Median}(MV1_x, MV2_x, MV3_x)$$

$$P_y = \text{Median}(MV1_y, MV2_y, MV3_y)$$

$$MVD_x = MV_x - P_x, MVD_y = MV_y - P_y \text{ 이다.}$$

제1도에서 움직임 벡터 추정의 제한범위를 나타내었으나, 이는 다음과 같은 이유로 완벽하게 MPEG-4의 VM(Verification Model)을 지원하지 못하였다.

즉, 제2a도~제2d도에서와 같이 8×8 모드 매크로 블록에서의 움직임 벡터 예측부호를 보면 중간 예측값으로 하여 사용되는 움직임 벡터값이 8×8 모드의 움직임 제한범위, 즉 도 1에 도시된 최대허용범위 ± 2.5 를 초과하는 경우는 가장 허용범위를 많이 초과하는 블록 1(도 2a)를 제외하면 블록 2(도 2b)에 해당하는 MVD₂뿐이다. 이는 3개의 움직임 벡터 예측후보(MV1, MV2, MV3) 중 단 하나만이 8×8 모드에 선택된 매크로블록 내에 속하여 있기 때문에 중간값으로서 구하여진 예측값으로부터 얻어진 MVD₂의 절대치는 제한범위 5.0(상술한 최대허용범위 ± 2.5 에 상당)을 초과할 가능성이 있다.

이에 반하여, 나머지 블록 3이나 또는 블록 4는 적어도 2개 이상이 8×8 모드에 선택된 매크로 블록 내에 속하여 있기 때문에 예측값으로 얻어진 MVD₃나 MVD₄의 절대치는 제한범위 5.0을 초과하는 가능성이 거의 없다.

움직임벡터 제한 범위인 최대허용범위 ± 2.5 를 넘게되는 경우 즉, 중간값으로 인한 예측치가 제한 범위 밖에 있는 경우가 발생하는데, 블록2의 MV를 MV_{CURRENT BLOCK} (해당블록의 움직임벡터)로, 블록2의 MV1을 MV_{Inside MV Prediction} (바운드내의 움직임벡터 예측값)로, 블록2의 중간값 예측치 Median(MV1, MV2, MV3)를 MV_{Median Prediction}로 정의하였을 경우 도 3에서와 같이 MV_{Median Prediction}가 제한 범위를 넘어 바로 인접한 점에 놓인 경우를 보인다.

여기서 제3도는 움직임벡터의 중간값 예측에서 제한범위를 넘는 경우의 예를 나타낸 구성도이다.

이와같이 종래 움직임벡터 코딩방법은 하나의 움직임벡터를 예측하기 위해서는 인접한 3개의 후보 움직임 벡터의 중간값을 이용한다.

발명이 이루고자 하는 기술적 과제

그러나 상기와 같은 종래 움직임벡터 코딩방법은 모든 픽셀에 대해 일률적으로 3개의 후보 움직임벡터의 중간값으로 예측하게 되면 최적으로 되지 않기 때문에 보다 복잡한 움직임 시퀀스(motion sequence)에서는 부호화 효율이 현저하게 감소되는 문제점이 있었다.

본 발명은 상기한 문제점을 해결하기 위해 안출한 것으로서, 3개의 후보 움직임벡터중 가장 차이가 적은 값을 찾아 그 차이값과 모드비트(mode bit)를 전송함으로써 전체적인 비트율을 감소시켜 부호화 효율을 향상시키는데 적당한 MPEG-4의 움직임벡터 부호화방법을 제공하는데 그 목적이 있다.

발명의 구성 및 작용

상기의 목적을 달성하기 위한 본 발명의 MPEG-4의 움직임벡터 부호화방법 3개의 움직임벡터 후보중에서 움직임벡터 코딩에 가장 적은 비트율을 가지는 하나의 후보를 선택하는 스텝과, 움직임벡터 예측에러와 예측모드 정보를 디코더쪽으로 전송하는 스텝과, 움직임벡터 차분치를 코딩하기 위한 최소비트율을 추측하는 스텝과, 최소율 예측을 이용하여 X 및 Y성분에 대한 움직임벡터를 코딩하는 스텝을 포함하여 이루어지는 것을 특징으로 한다.

이하, 첨부된 도면을 참조하여 본 발명의 MPEG-4의 움직임벡터 부호화방법을 설명하기로 한다.

먼저, 본 발명의 움직임벡터 부호화방법은 3개의 후보 움직임벡터 중에서 움직임벡터의 코딩에 가장 적은 비트율을 가지는 하나의 후보 움직임벡터를 선택한다.

그리고 움직임벡터 예측에러(실제 움직임벡터와 예측한 움직임벡터와의 차이값)와 예측모드정보를 보낸다.

이때, 예측후보의 수는 대부분 3개이고 움직임벡터 차이값을 코딩하는 비트율은 쉽게 추측할 수 있다.

제4도는 본 발명의 움직임벡터 부호화방법에 따른 움직임벡터 부호화에서 MVD코드비트와 모드비트의 비트 스트림 신텍스(bitstream syntax)를 도시하였다.

움직임벡터 부호화(CODING)은 X와 Y에 대해서 각각 실시한다.

그리고 모드비트는 MVD를 코딩할 때, 3개의 후보 움직임벡터중 어느 후보를 사용하였는지를 가리킨다.

여기서, 모드비트에 앞서는 MVD최소비트율은 MVD최소비트 정보를 이용함으로써 감소시킬 수 있다.

이와같이 최소율 예측을 사용한 움직임벡터 X성분은 아래와 같이 코딩하고 Y(취도)성분도 동일한 방법으로 코딩한다.

```
MR_MVD_coding(MV_x_code,*vlc_mag,*residual,*bitstream)
{
    /*Find the minimum rate predictor among three neighboring
    candidate,*/
    R1_x = RATE(MV_x - MV1_x),
    R2_x = RATE(MV_x - MV2_x),
    R3_x = RATE(MV_x - MV3_x),
    R_x_min = MIN(R1_x, R2_x, R3_x).
    P_x_min_rate =MV1_x corresponding to R_x_min.
    (At the same rate, the median has the highest priority for P_x_min_
    rate)
    MVD_x_min_rate = MV_x-P_x_min_rate
    /*Coding MVD_x_min_rate
    MVD_encoding(MVD_x_min_rate,f_code,vlc_code_mag,residual,bitstream);
}
```

한편, 제5a도 내지 제5b도는 본 발명의 움직임벡터 부호화에 따른 모드비트 감소의 일예를 엔코더 및 디코더측면에서 도시하였다.

먼저, 제5a도는 엔코더 측면에서 도시한 것으로서, MV1=0, MV2=0, MV3=3 이고 현재블록의 움직임벡터 MV=5일 경우, MV5에서 멀리떨어진 MV1과 MV2보다 더 가까운 MV3를 하나의 후보 움직임벡터로 선택한다.

이때 최소율 움직임벡터 차이값(MVD_min_rate)은 2가 된다.

상기한 바와같이 엔코더는 최소율 예측에 의해 MVD_min_rate를 코딩한다. 그 다음 엔코더는 디코더가 실제 움직임벡터를 검출할 수 있는지를 최소의 정보로 조사한다.

이어 제5b도는 디코더 측면에서 도시한 것으로서, 본 발명의 실시예에서는 단지 두 개의 후보가 있음을

알 수 있다.

그러나 후보 1은 엔코더로부터 받은 MVD_{min_rate} 로 디코딩해보면 실제 MVD_{min_rate} 와는 차이가 있으므로 실제 움직임벡터가 아님을 알 수 있다.

따라서, 이 경우에 엔코더는 3개의 후보 움직임벡터 중에서 실제 움직임벡터를 찾기 위한 모드비트를 디코더에 보낼 필요가 없다.

여기서 본 발명의 움직임벡터 부호화방법에 따른 알고리즘은 아래와 같다.

```

max = MAX(MV1x, MV2x, MV3x)
min = MIN(MV1x, MV2x, MV3x)

inside_mv_cnt=The number of neighboring candidates
             whose corresponding blocks are inside of the VOP
distinct_mv_cnt=The number of distinct neighboring candidates.
                (e.g. distinct_mv_cnt=2, if MV1x=0, MV2x=0, and MV3x)
if( |max-min| >= THR && inside_mv_cnt >= 2 && distinct_mv_cnt==3)
{
  /* Minimum rate prediction is applied */
  /* Minimum rate prediction and the corresponding MVDx min rate           coding */
  MR_MVD_coding(MVx, f_code, &vlc_code_mag, &residual, bs);
  /* Determine MODE */
  /* Step1 : Evaluate three MV candidates. */
  Candidate1x = MVD_decoding(f_code, vlc_code_mag, residual, MV1x)
  Candidate2x = MVD_decoding(f_code, vlc_code_mag, residual, MV2x)
  Candidate3x = MVD_decoding(f_code, vlc_code_mag, residual, MV3x)
  /* Step2 : Check if "vlc_code_magts" and "resulting from
             Candidate ix(i=1,2,and 3)encoding, have the same values as
             "vlc_code_mag "and"residual", respectively. */
  Candidate_num = 0
  MR_MVD_coding(Candidate1x, f_code, &vlc_code_mag, ts, &residual_ts, ts);
  if(vlc_code_mag, ts==vlc_code_mag, && ,residualts==residual)
  candidatenum++;
  MR_MVD_coding(Candidate2x, f_code, &vlc_code_mag, ts, &residual_ts, ts);
  if(vlc_code_mag, ts==vlc_code_mag, && ,residualts==residual)
  candidatenum++;
  MR_MVD_coding(Candidate3x, f_code, &vlc_code_mag, ts, &residual_ts, ts);
  if(vlc_code_mag, ts==vlc_code_mag, && ,residualts==residual)
  candidatenum++;
  /* Step3 : Determine the MODEx and code it. */
  if (candidate_num == 1)
    No bit allocated for MODEx;
  else if (candedate_num == 2)
    1 bit for MODEx;
  else if (candidate_num == 3)
    1~2 bit for MODEx;
}

```

```

else
{
  /* Median prediction is applied. */
  No bit allocated for MODEx;
}

```

이와같은 알고리즘을 이용한 본 발명의 MPEG-4의 움직임벡터 부호화방법을 플로우차트를 참조하여 설명하기로 한다.

제6도는 본 발명에 따른 MPEG-4의 움직임벡터 부호화방법을 설명하기 위한 플로우차트이다.

제6도에 도시한 바와같이 3개의 움직임벡터 후보중에서 움직임벡터 코딩에 가장 적은 비트율을 가지는 하나의 후보를 선택한다.

이어, 움직임벡터 예측에러와 예측모드 정보를 디코더쪽으로 전송한다. 여기서, 움직임벡터 예측에러란 실제의 움직임벡터와 예측한 움직임벡터와의 차이값을 말한다.

이어, 움직임벡터 차분치를 코딩하기 위한 최소비트율을 추측한다.

최소율 예측을 이용하여 X 및 Y성분에 대한 움직임벡터를 코딩하면 본 발명의 MPEG-4의 움직임벡터 부호화가 이루어진다.

발명의 효과

이와 같은 본 발명의 MPEG-4의 움직임벡터 부호화방법은 움직임벡터의 부호화시 효율성을 향상시키기 위해 주위의 후보 움직임벡터 3개중에서 움직임벡터와 가장 차이가 적은 것을 찾아 그 차분치와 모드비트를 보냄으로써 전체적으로 비트율을 감소시켜 동영상의 압축율을 높이는 효과가 있다.

(57) 청구의 범위

청구항 1

부호화하고자 하는 매크로 블록 내에 움직임 벡터를 예측하기 위한 3개의 후보 움직임 벡터 중에서 움직임 벡터 부호화에 제일 적은 비트율을 가지는 하나의 후보 움직임 벡터를 선택하는 스텝과,

실제의 움직임 벡터와 예측한 움직임 벡터와의 차분치인 움직임 벡터 예측 에러와 예측모드정보를 디코더에 전송하는 스텝과,

상기 차분치를 부호화하기 위한 최소비트율을 추측하는 스텝과,

최소 비트율 예측을 이용하여 X성분에 대한 움직임 벡터의 부호화와 Y성분에 대한 움직임 벡터의 부호화를 수행하는 스텝을 구비함을 특징으로 하는 MPEG-4의 움직임벡터 부호화방법.

청구항 2

제1항에 있어서,

상기 X성분에 대한 움직임벡터의 부호화는

```
MR_MVD_coding(MVx_code,*vlc_mag,*residual,*bitstream)
```

```
{
  /*Find the minimum rate predictor among three neighboring
candidate,*/
```

```
R1x = RATE(MVx - MV1x),
```

```
R2x = RATE(MVx - MV2x),
```

```
R3x = RATE(MVx - MV3x),
```

```
Rx_min = MIN(R1x, R2x, R3x).
```

```
Px_min_rate =MVix corresponding to Rx_min.
```

```
(At the same rate, the median has the highest priority for Px_min_ rate)
```

```
MVDx_min_rate = MVx-Px_min_rate
```

```
/*Coding MVDx_min_rate
```

```
MVD_encoding(MVDx_min_rate,f_code,vlc_code_mag,residual,bitstream);
```

```
}
```

과 같은 알고리즘으로 통해 이루어지는 것을 특징으로 하는 MPEG-4의 움직임벡터 부호화방법.

청구항 3

제1항에 있어서,

상기 Y성분에 대한 움직임벡터의 부호화는

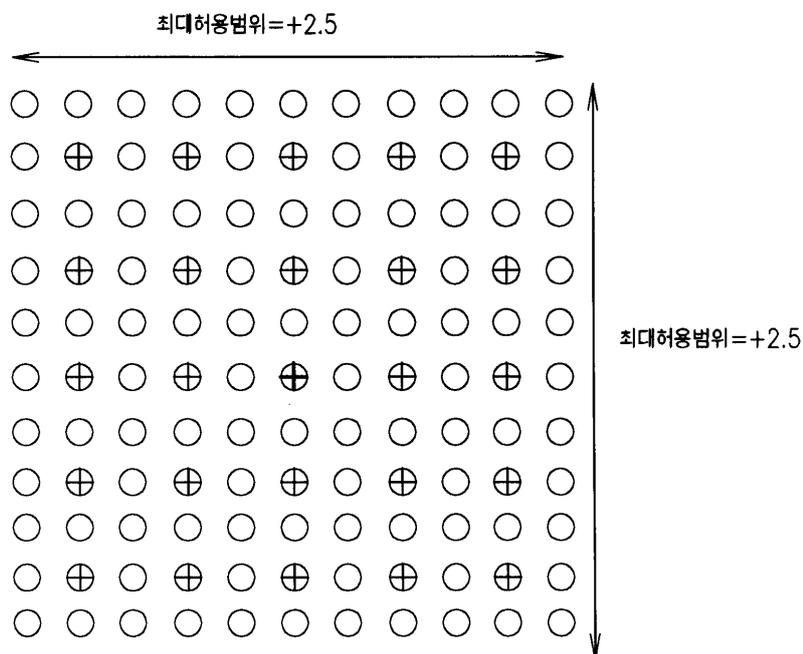
MR_MVD_coding(MV_Y_code, *vlc_mag, *residual, *bitstream)

```
{
  /*Find the minimum rate predictor among three neighboring candidate,*/
  R1Y = RATE(MVY - MV1Y),
  R2Y = RATE(MVY - MV2Y),
  R3Y = RATE(MVY - MV3Y),
  RY_min = MIN(R1Y, R2Y, R3Y).
  PY_min_rate = MViY corresponding to RY_min.
  (At the same rate, the median has the highest priority for PY_min_rate)
  MVDY_min_rate = MVY - PY_min_rate
  /*Coding MVDY_min_rate
  MVD_encoding(MVDY_min_rate, f_code, vlc_code_mag, residual, bitstream);
}
```

과 같은 알고리즘으로 통해 이루어지는 것을 특징으로 하는 MPEG-4의 움직임벡터 부호화방법.

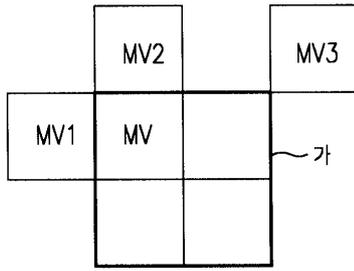
도면

도면1

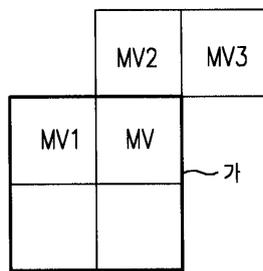


- + 정수단위 픽셀 위치
- 반화소 위치
- + 8×8벡터 탐색영역의 센터 포인트

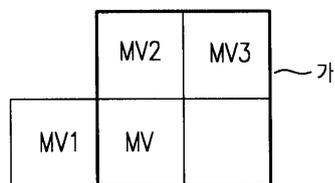
도면2a

블록 1 (MVD₁)

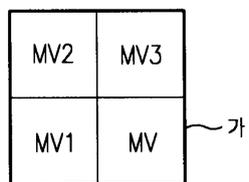
도면2b

블록 1 (MVD₂)

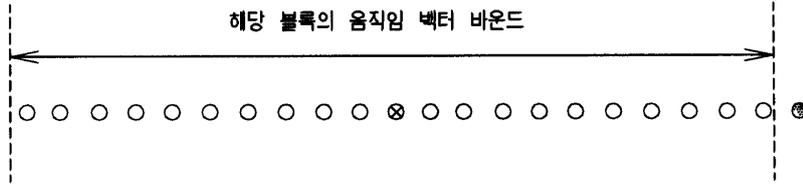
도면2c

블록 1 (MVD₃)

도면2d

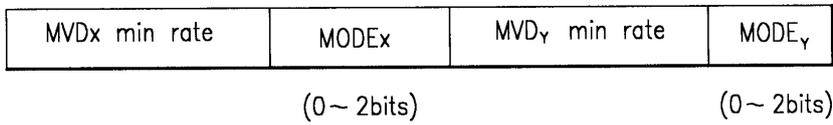
블록 1 (MVD₄)

도면3

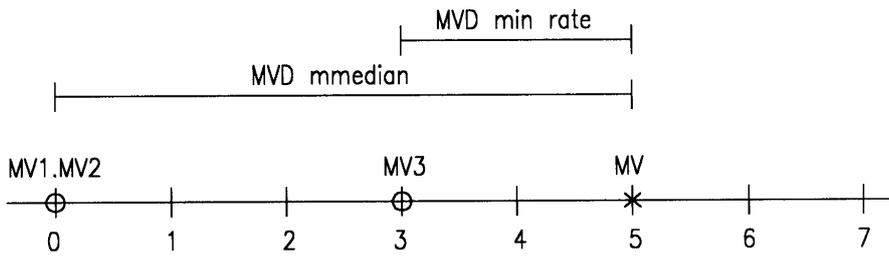


- X 바운드내의 움직임 벡터 예측값
- 해당 블록의 움직임 벡터 예측 후보
- ⊗ 움직임 벡터 중간 예측값

도면4

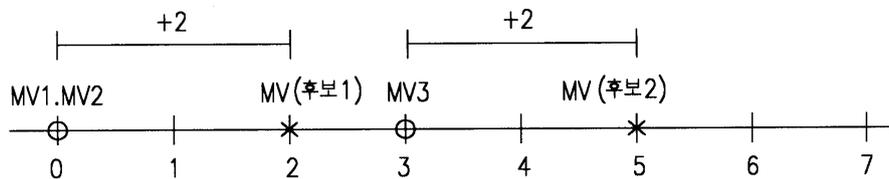


도면5a



MV1=0, MV2=0, MV3=3, MV=5
 최소율 예측(Minimum rate predictor)=MV3
 최소율 움직임 벡터 차이 (MVD min rate)=2

도면5b



후보1의 최소율 움직임 벡터 차이 (MVD min rate of candidate 1)=-1
 그러므로 MV=후보2
 모드비트(Mode bit)불필요

도면6

