1999 年 度 大 韓 電 子 工 學 會 夏 季 綜 合 學 術 大 會 論 文 集 第 22 卷 第 1 號 99/6

# 비트 플레인을 이용한 움직임 추정기 설계의 관한 연구

김병철, 조원경
경희대학교 전자공학과
phillip@csvlsi.kyunghee.ac.kr

# A Study on Motion Estimator Design Using Bit Plane

Byungchul Kim, Wonkyung Cho
School of Electronic Engineering Kyunghee University
E-mail: phillip@csvlsi.kyunghee.ac.kr

## Abstract

Among the compression methods of moving picture information, a motion estimation method is used to remove time-repeating. The Block Matching Algorithm in motion estimation methods is the commonest one. In recent days, it is required the more advanced high quality in many image processing fields, for example HDTV, etc. Therefore, we have to accomplish not by means of Partial Search Algorithm, but by means of Full Search Algorithm in Block Matching Algorithm.

In this paper, it is suggested a structure that reduce total calculation quantity and size, because the structure using Bit Plane select and use only 3bit of 8bit luminance signal.

## I. 서론

HDTV와 같은 영상처리분야에서는 많은 데이터를 포함하는 고화질의 동영상을 고속으로 전송하기 위하여 압축기법을 필수적으로 사용하고 있다. 동영상 정보의 압축기법 중에서 시간적 중복성을 제거하는데는 움직임 추정 기법을 사용한다. 움직임 추정 기법 중 가장 많이 사용되는 블록 정합 알고리즘은 현재 프레임을 여러 개의 기준블록으로 분할하여 이전 프레임의 탐색영역 안에서 가장 유사한 블록을 찾아 기준블록에 대한 상대 위치를 추출하는 기법이다. 동영상의 전송 방법은 블록정합 알고리즘에 의해 추출된 움직임 벡터와 두 블록간의 화소값 차이를 전송하는 기법이며, 이는 국제 표준인 MPEG1,2,4나 H.261, H.262, H.263등 대부분의 표준 동영상 압축 기법에 채택되었다.[1][2]

블록 정합 알고리즘 중에는 3단계 탐색, 2차원 대수 탐색, One-at-a-time 탐색, Conjugate Direction 탐색 같은 부분탐색과 완전탐색 알고리즘으로 구별되는데, 완전탐색 알고리즘은 기준블록을 탐색영역 내의 모든 블록과 비교하는 방법으로 성능이 우수하고 데이터 흐름과 제어회로가 비교적 간단하다는 장점이 있으나, 탐색영역이 커질 경우 막대한 연산량을 필요로 하는 단점이 있다.[5][6][7][8]

본 논문에서는 화면의 밝기를 판별한 다음, 비트 플레인을 이용하여 휘도 8비트 신호 중 3비트만 선택·사용함으로써 전체적인 연산량과 크기를 줄일 수 있는 구조를 제안하였다. 제안한 구조를 C언어로 모델링하여 모의실험으로 검증하였으며, VHDL을 이용하여 설계 및 합성하였다.

## II. 비교선택기의 구조

그림1에서 보듯이 블록 당 휘도 평균값이 증가함에 따라 DCT 직류 값이 증가하는데, 이것은 DCT 직류 값이 화면의 밝기 정도를 나타냄을 알 수 있다.

따라서, 본 논문에서는 비트 플레인을 이용하여 휘도신호 8비트 중 DCT 직류값에 따라 자리비트와 좌우비트를 추가 3비트를 결정하여 처리함으로 프레임

메모리와 ME의 크기를 줄일 수 있는 비교선택기를 제안한다

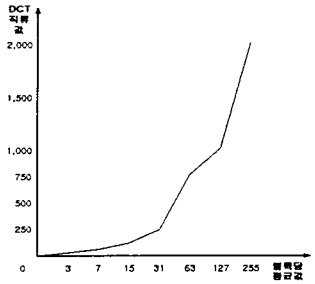



그림1. 휘도신호에 따른 DCT 직류값의 변화(8×8)

예를 들어, 8×8 블록에서 DCT 직류값에 따른 선택 비트를 표1과 같이 정할 수 있다.

표1. DCT 직류값에 따른 선택비트

| DCT 직류값 | 선택비트 |
|---|---|
| 0-24 | 1,2,3 |
| 25-56 | 2,3,4 |
| 57-120 | 3,4,5 |
| 121-248 | 4,5,6 |
| 249-760 | 5,6,7 |
| 761-2040 | 6,7,8 |

비교선택기의 구조는 그림2와 같은데, DCT 직류 값을 입력신호로 받아서 비교기에서 표1과 같이 DCT 직류 값이 어느 범위인지 판별한 다음 선택신호를 내보낸다. 그러면, 선택블록에서 선택신호에 따라 8비트 성분 중 3비트를 선택하게 된다.

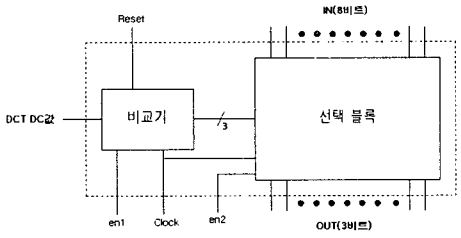



그림2. 비교선택기

비교선택기는 그림3와 같이 부호화 과정에서 I(Intra coded) 화면에 적응적으로 적용하고, P(Predicted coded) 화면에서도 같은 선택비트를 사용한다.

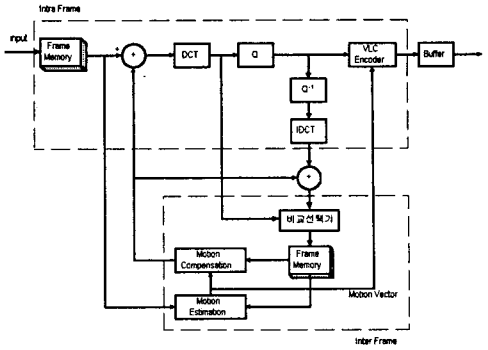



그림3. 비교선택기를 포함한 부호화 구조

## II. 움직임 추정기의 구조

본 논문에서는 기준블록 8×8, 탐색영역 23×23, 영상신호는 8bits(gray scale) 동영상을 처리하였으며, 본 논문에서 제안한 움직임 추정기(ME)의 처리 구조는 그림4과 같다.

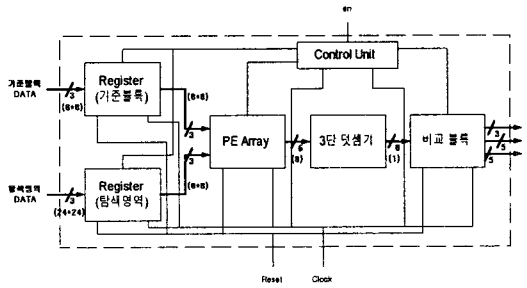



그림 4. ME 전체

본 논문에서 최적의 정합블록을 찾기 위한 정합 기준으로 사용한 것은 최소 MAD(mean of absolute difference) 방식인데, N×N 블록에서 식(1)과 같이 표현된다.

$$MAD(i,j)=\frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}|x_t(k,l)-x_{t-1}(k+i,l+j)| \quad (1)$$

여기서 $x_t(k,l)$은 현재 프레임의 $(k,l)$위치에 있는 화소의 밝기 값이고, $x_{t-1}(k+i,l+j)$는 이전 프레임의 $(k,l)$위치에서 $(i,j)$만큼 이동한 위치에 있는 화소의 밝기 값이다.

### 1.PE 배열

PE 배열은 수직 또는 수평으로의 8개 화소에 대한 절대값 차를 찾아 계속적으로 누적해 간다. PE 배열의 전체구조는 그림5와 같다.

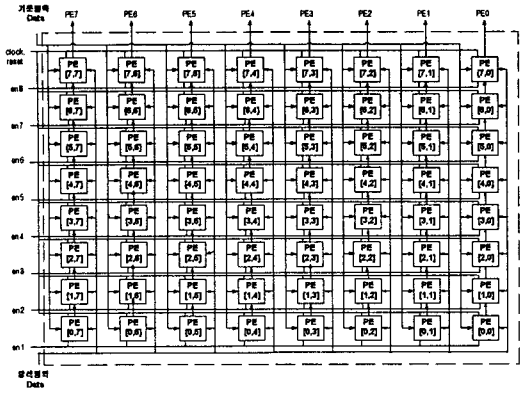


그림 5. PE배열

여기서 PE배열은 64개의 PE블록으로 구성되는데 각 PE블록(그림6)의 구조를 살펴보면, 기준블록과 탐색영역의 데이터의 차와 절대값 계산, 그리고 이 값과 이전 PE에서의 출력값과 더해 레지스터의 저장하는 누적기로 구성된다.[3]

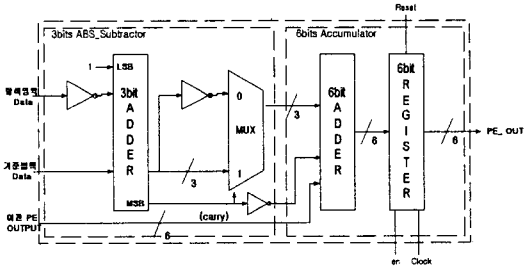



그림 6. PE블록

2. 3단 덧셈기

3단 덧셈기는 빠른 연산을 위해 PE배열에서 누적된 값을 다시 수평 또는 수직으로 8번 누적해 간다.(그림 7)

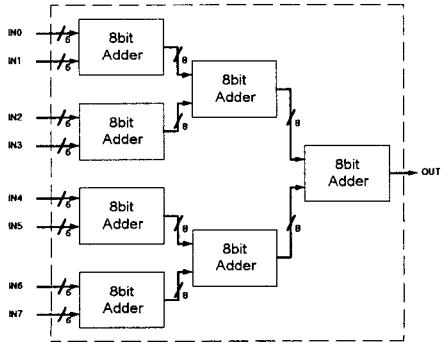



그림 7. 3단 덧셈기

3. 비교블록 연산기

비교블록 연산기는 3단 덧셈기에서 누적된 값을 탐색영역에 따라, 본 논문의 경우 256(16×16)번에 대해 각각 구해서 이 256개의 누적 값 중 가장 작은 값을 찾고 그 값을 찾게 된 위치의 벡터 값을 이 블록의 움직임 벡터로 선택한다(그림8).[4]

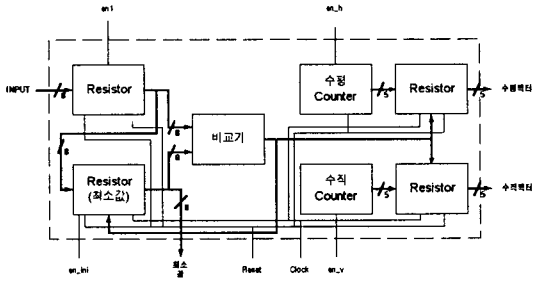
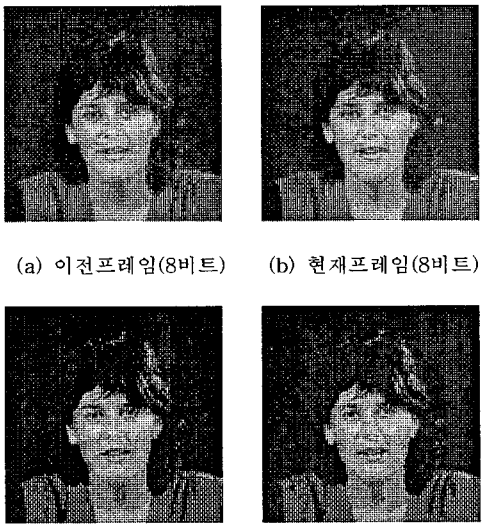



그림 8. 비교블록 연산기의 구조

## VI. 실험 및 고찰

본 논문에서 제안한 구조의 성능 평가를 위해 256×256 Miss America(26 Frame) 이미지(그림9)를 사용하였고, C언어로 모델링하였다.



(a) 이전프레임(8비트) (b) 현재프레임(8비트)

(c) 이전프레임(3비트) (d) 현재프레임(3비트)

그림9. Sample 이미지

"256×256 Miss America"의 I-picture에 대한 DCT 직류 값은 598,824인데, 8×8블록으로 환산하면 546이다. 표1에 따라 선택비트는 5,6,7비트가 된다.

26프레임에 대해서 8비트 성분을 모두 사용했을 때의 움직임 벡터와 5,6,7비트의 3비트 성분만을 선택하여 사용했을 때의 움직임 벡터가 같게 나옴을 알 수 있었다. 예를 들어, 그림8의 휘도신호 8비트를 모두 사용한 (a)와 (b), 5,6,7비트의 3비트 성분만 사용한 (c)와 (d)의 움직임이 가장 많은 눈과 입의 움직임 벡터를 각각 구해보면 표2과 같이 같게 나옴을 알 수 있었다.

표2. 움직임 벡터 비교

| 위치 | 움직임 벡터값 | |
|---|---|---|
| | 8비트 | 상위3비트 |
| (102,127) | (102,126) | (102,126) |
| (146,124) | (146,123) | (146,123) |
| (119,170) | (118,170) | (118,170) |
| (128,172) | (127,172) | (127,172) |
| (141,169) | (140,169) | (140,169) |

그리고, 이렇게 검증된 움직임 추정기를 VHDL을 이용하여 설계하였고, V-system을 사용하여 기능을 시뮬레이션 하였으며, 삼성 KG75 0.65um SOG 공정으로 합성하였다.

표3. 합성결과(Gate Equivalents)

| 구성부 | 8비트 | 3비트 |
|---|---|---|
| 비교선택기(64×64) | - | 106,507 |
| 프레임 메모리(256×256) | 550,912 | 206,848 |
| 기준블록 레지스터(8×8) | 538 | 202 |
| 탐색영역 레지스터(24×24) | 4,842 | 1,818 |
| PE 배열 | 2,816 | 1,216 |
| 3단 덧셈기 | 156 | 84 |
| 비교블록 | 59 | 42 |

표3는 합성한 결과를 나타내는데, 휘도성분 8비트를 모두 취했을 때보다 3비트만 선택해서 사용한 경우 약 40%의 크기가 줄었음을 알 수 있었으나, 128×128 비교선택기를 사용했을 경우에는 오히려 약 10% 크기가 늘어남을 알 수 있었다.

## V. 결론

본 논문에서는 완전탐색 블록정합 움직임 추정기를 설계하는데 있어서 DCT 직류 값을 이용하여 화면의 밝기를 판단하여 휘도신호 8비트 모두를 사용하지 않고, 비트 플레인을 이용하여 그 중에 3비트만 선택하는 비교선택기를 I-Picture에 적응적으로 적용하고, P-Picture에서도 같은 선택비트를 사용하는 구조를 제안하였다.

이 제안된 구조를 C로 모델링하여 모의실험으로 검증하였으며, VHDL을 이용하여 설계하고 삼성 KG75 0.65um SOG 공정으로 합성하여 프레임 메모리 및 ME 구조의 크기를 40% 줄일 수 있었음을 알 수 있었다.

앞으로 MPEG2 시스템의 이 ME구조를 실제로 적용해 봄으로써 시스템 검증을 하는 것이 필요하겠다.

## 참고문헌

[1] J.R.Jain *et al.*, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Comm.*, vol. COM-29, pp. 1799-1808, Dec. 1981.

[2] MPEG Test Model Editing Committee, "Generic coding of moving pictures and associated audio: MPEG-2 Test Model5(TM5)", ISO/IEC JTC1/SC29/WG11, MPEG94/No.702, Mar.1994

[3] Takao ONOYE, et al.,"A VLSI Architecture for MPEG2 MP@HL Real time motion Estimation", IEEE'96, pp 664-667

[4] 박상복,김이섭 "완전탐색 블럭정합 움직임 추정기에서의 변위벡터 추출을 위한 효율적인 논리회로의 구현", 전자공학회 추계종합 학술대회 논문집(II) Vol.17, No.2, pp.1403-1406,1994

[5] T.Koga and et al. "Motion compensated interframe coding for video-conferencing," Proc. Nat. Telecommun. Conf., New Orleans, LA, pp.G5.3.1-G5.3.5, 1981

[6] J.R.Jain and A.K.Jain, "Displacement measurement and its application in interframe image coding", IEEE Trans. on Commun., vol. COM-29, no.12, pp.1799-1808, DEC.1981.

[7] R.Srinvasan and K.R.Rao, "Predictive coding based on efficient motion estimation", IEEE Trans. on Commun., vol. COM-33, no.8, pp.888-896, Aug.1985.

[8] R.Srinvasan and K.R.Rao, "Predictive coding based on efficient motion estimation", ICC'1984, pp.541-526.