논문-98-3-1-08

고속 움직임 추정 알고리즘에 적합한 VLSI 구조 연구

이 재 헌*, 나 종 범*

A VLSI Architecture for Fast Motion Estimation Algorithm

Jaehun Lee* and Jong Beom Ra*

요 약

동영상 부호화에서 블록 정합 움직임 추정 기법은 움직임 추정 기법으로 가장 많이 쓰이고 있는 방법이다. 이 논문에서는 블록 정합 움직임 추정 기법의 하나로 최근에 제안된 공간적 상관 관계와 계층적 탐색방법을 이용한 고속 움직임 추정 알고리즘의 구현에 적합한 VLSI 구조를 제안한다. 제안된 구조는 systolic array에 바탕을 둔 탐색 기본 단위와 두 개의 shift register array 등으로 이루어지며 수평/수직 -32~+31 화소 크기의 탐색을 수행한다. 이 때 탐색 기본 단위는 반복하여 사용하게 함으로써 게이트 수를 최소화 하였다. 탐색 기본 단위의 구조로는 전역 탐색을 수행할 수 있는 기존의 여러 가지 systolic array 들이 사용 가능하며, 그 선택에 따라 칩의 크기와 속도 사이의 절충이 가능하다. 본 논문에서는 PE(processing element)의 개수를 줄여 전체적인 칩 사이즈를 줄이는데 중점을 두고 탐색 기본 단위의 구조를 결정하였다. 제안된 구조를 이용하면 352×288 크기의 영상, 탐색 영역 수평/수직 -32~+31 화소에 대해서 클럭 주파수가 35MHz일 때 최대 30Hz까지 실시간 처리를 할 수 있는 움직임 추정 칩을 20,000 게이트 이하로 구현할 수 있다. 더 높은 전송률의 입력 영상(720×480, 30Hz)에 적용할 경우에는 단순히 PE 개수를 늘린 구조를 탐색 기본 단위로 선택함으로써 실시간 구현이 가능하다.

Abstract

The block matching algorithm is the most popular motion estimation method in image sequence coding. In this paper, we propose a VLSI architecture for implementing a recently proposed fast bolck matching algorith, which uses spatial correlation of motion vectors and hierarchical searching scheme. The proposed architecture consists of a basic searching unit based on a systolic array and two shift register arrays. And it covers a search range of -32~+31. By using the basic searching unit repeatedly, it reduces the number of gatyes for implementation. For basic searching unit implementation, a proper systolic array can be selected among various conventional ones by trading-off between speed and hardware cost. In this paper, a structure is selected as the basic searching unit so that the hardware cost can be minimized. The proposed overall architecture is fast enough for low bit-rate applications (frame size of 352×288, 30frames/sec) and can be implemented by less than 20,000 gates. Moreover, by simply modifying the basic searching unit, the architecture can be used for the higher bit-rate application of the frame size of 720×480 and 30 frames/sec.

I. 서 론

동영상 부호화에서는 높은 데이터 압축율을 얻기 위해서 인접한 프레임간의 시간적 중복성을 제거하는 움직임 보상 부호화가 널리 쓰이고 있다. 움직임 보상 부호화 기법은 움직임 추정 기법을 통하여 입력 영상에 가장 유사한 영상을 시간적으로 이전 프레임으로부터 예측하고, 예

* 한국과학기술원 전기 및 전자공학과
Dept. of Electrical Engineering KAIST

측된 영상과 입력 영상의 차를 변환 부호화하는 방법이다. 움직임 추정 기법은 동영상 부호화에서 전체적인 부호화 성능과 시간에 영향을 미치는 중요한 부분이다.

움직임 추정 기법으로는 블록 정합 기법(Block Matching Algorithm: BMA)이 주로 이용된다. 블록 정합 기법은 블록 내의 모든 화소가 같은 움직임 벡터(Motion vector: MV)를 갖는다는 가정하에 이전 프레임으로부터 블록 단위로 움직임을 추정하는 방법이다. 블록 정합 기법의 대표적인 방법으로는 전역 탐색 기법(Full Search BMA)이 있다. 전역 탐색 기법은 탐색 영역 내부의 모든 탐색점들에 대해서 탐색을 수행하기 때문에 성능은 최적이지만 계산량이 많아서 하드웨어로 실시간 처리가 가능하도록 구현하고자 할 때 칩 사이즈가 커지는 단점이 있다. 이러한 단점을 극복하고자 그동안 여러 고속 알고리즘들과 그 알고리즘에 적합한 VLSI 구조 또한 제안되어 왔다.[1-11] 하지만 제안된 고속 알고리즘들은 계산량을 줄이는 데에만 초점을 두어 데이터 입출력 대역폭이 크거나 많은 부가적인 메모리의 사용 등으로 하드웨어로 구현하기에 부적합 하다거나[12], 하드웨어로 구현할 때 칩 사이즈가 줄어드는 장점은 있으나 전역 탐색에 비해 성능이 떨어진다는 단점을 가지고 있다[13]. 하드웨어로 구현하기 용이한 대표적인 고속 알고리즘인 TSS(Three Step Search)의 경우 PSNR(Peak Signal-to-Noise ratio)이 전역 탐색 기법에 비해 약 1dB 정도 떨어진다[13].

이런 가운데 최근 블록 정합 알고리즘의 하나인 HSBMA3S(Hierarchical search BMA by using three candidates and spatial correlation)라는 새로운 고속 알고리즘이 제안되었다[14]. HSBMA3S는 기본적으로 계층적 구조를 지니고 있으며, 다수의 MV 후보와 MV들의 공간적 상관 관계를 사용한다.

따라서 이 알고리즘은 기존의 여러 고속 알고리즘들이 가지고 있던 국소적인 최소치(local minimum)문제를 해결하여 전역 탐색 기법에 비해 계산량이 훨씬 적으면서 전역 탐색 기법의 성능을 거의 그대로 유지하는 특징을 가진다. MPEG-2 비디오 코덱에 적용했을 때 전역 탐색 기법에 비해 최악의 경우에도 0.2dB 정도만 떨어지는 성능을 보인다[14]. 또한 알고리즘 자체가 하드웨어로 구현하기에 용이하다.

따라서 HSBMA3S를 하드웨어로 구현했을 때, 전역 탐색 기법과 비슷한 성능을 발휘하면서 적은 계산량으로 인해 칩 사이즈가 현저하게 줄어드는 장점이 있다.

본 논문에서는 HSBMA3S에 적합한 VLSI 구조를 제안한다. 제안한 구조는 실시간으로 구현 가능한 범위 내에서 속도 보다는 칩 사이즈가 줄어드는 쪽에 중점을 둔 구조이다.

그리고 구현하고자 하는 움직임 추정 칩의 적용 범위는 저 전송률(352×288, 30Hz)이고 탐색 영역의 크기는 수평/수직 -32~+31 화소로써 이동단말기에 적용하는 것을 궁극적인 목표로 두고 있다.

본 논문의 구성은 다음과 같다. 2장에서는 구현하고자 하는 HSBMA3S 알고리즘과 제안한 VLSI 구조에 대해 설명하고, 3장에서는 제안된 구조를 사용할 때의 외부 메모리 대역폭과 칩의 동작 주파수, 게이트 수 등을 알아보고 전역 탐색 기법과 비교한다. 마직막으로 4장에서 결론을 맺는다.

II. 본 문

1. HSBMA3S 알고리즘

이 논문에서는 비디오 폰 등에 적용 가능한 움직임 추정 기법을 하드웨어로 구현하기 위해서 고속 움직임 추정 알고리즘의 하나인 HSBMA3S를 채택하였다. 비디오 폰 등의 적용에서는 영상의 크기가 작고 영상의 움직임이 대체적으로 적으므로 352×288 크기의 30Hz 영상을 대상으로 크기가 16×16인 매크로 블록(MB)을 처리 단위로 수평/수직 -32~+31 화소인 탐색 영역에 대해서 HSBMA3S 알고리즘을 적용한다. 이 알고리즘은 그림 1에서 보는 바와 같이, 상위 레벨, 중간 레벨, 하위 레벨의 3단계로 나누어져 있다.

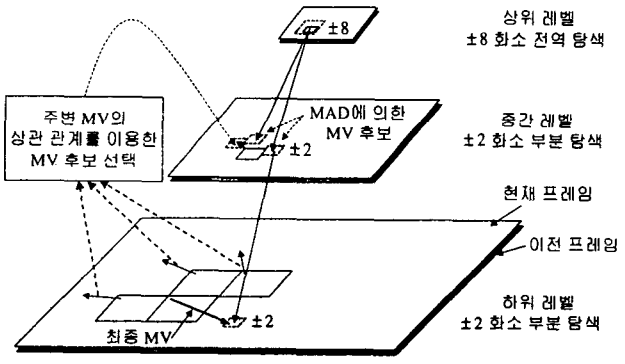


그림 1. HSBMA3S 개념도
Fig. 1. Conceptual diagram of HSBMA3S

1) 상위레벨

상위 레벨에서는 영상으로부터 필터링되고 subsampling된 가로, 세로 각각 1/4 크기인 영상을 이용하여 탐색을 수행한다. 그림 2와 같이 가로, 세로 각각 1/4씩 줄어든 4×4 크기의 블록 단위로 탐색 영역의 크기도 가로, 세로 각각 1/4씩 줄인 수평/수직 ±8 화소 영역에 대해서 전역 탐색을 수행한다. 정합 기준으로는 MAD(Mean Absolute Difference)를 사용하며, 중간 레벨의 탐색을 위한 MV 후보로 MAD가 최소인 두 점을 선택한다.

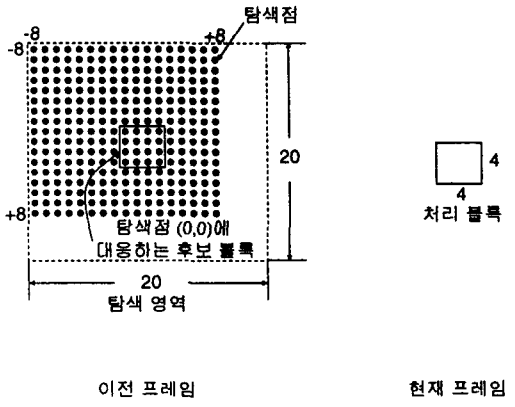


그림 2. 상위 레벨에서의 전역 탐색
Fig. 2. Full search in the upper level.

2) 중간 레벨

중간 레벨에서는 가로, 세로 각각 입력 영상의 1/2 크기의 영상을 이용하여 탐색을 수행한다. 중간 레벨에서는 총 3개의 MV 후보에 대해서, 그림 3에서와 같이 각각의 MV 후보를 초기점으로 8×8 크기의 블록 단위로 수평/수직 ±2 화소 크기의 부분 탐색을 수행한다.

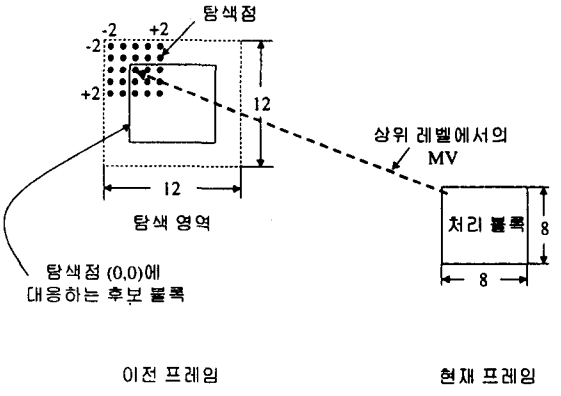


그림 3. 중간 레벨에서의 부분 탐색
Fig. 3. Local search in the middle level.

여기서 사용되는 3개의 MV 후보 중 두 개는 상위 레벨에서 MAD 를 정합 기준으로 하여 선택된 것들이고, 나머지 하나는 MV들의 공간적 상관 관계를 이용하여 선택된 것이다. 3개의 후보에 대해서 탐색을 수행하여 최소 MAD를 가지는 점을 각각 하나씩 얻게 되면, 그 중에 MAD가 최소인 한 점을 하위 레벨에서의 탐색 초기점으로 선택한다.

가. 공간적 상관 관계를 이용한 MV 후보의 선택

HSBMA3S에서는 시간적으로 우선 처리된 인접 MB들의 MV들의 유사성을 이용하여 MV 후보를 하나 결정하여 중간 레벨의 탐색에서 초기점으로 사용한다.

인접 MB들의 MV들을 이용하여 MV 후보를 선택하기 위해서 [14]에서는 주변 MV들의 형태에 따라 적절한 수의 군으로 분류한 다음 군에 따라 적절히 선택된 주변 MV들의 평균값을 MV후보로 결정하는 방법을 사용하였다. 본 논문에서는 하드웨어 구현이 용이하도록 다음과 같은 방법을 사용한다. 즉 그림 4와 같이 시간적으로 우선 처리되어 이미 최종 MV를 알고 있는 3개의 MB들의 MV들의 x성분과 y성분을 각각 따로 중간값을 취해 하나의 후보를 결정한다.

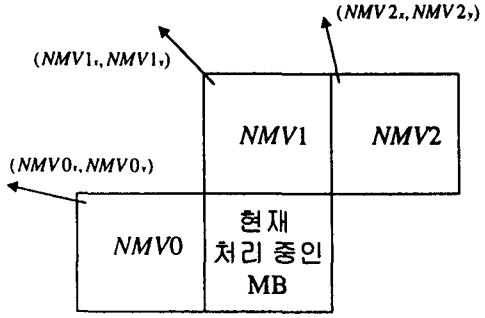


그림 4. 공간적 상관 관계를 이용한 MV 후보 선정에 이용되는 주변 MB들
Fig. 4. Neighboring MB's for estimating MV candidate using spatial correlation of motion field.

수식으로 표현하면 다음과 같다.

$$MV_x = \text{Median}\{NMV0_x, NMV1_x, NMV2_x\}$$
$$MV_y = \text{Median}\{NMV0_y, NMV1_y, NMV2_y\}$$

이렇게 얻어진 MV 후보를 1/2로 크기를 줄여 중간 레벨에서 탐색의 초기점으로 사용한다.

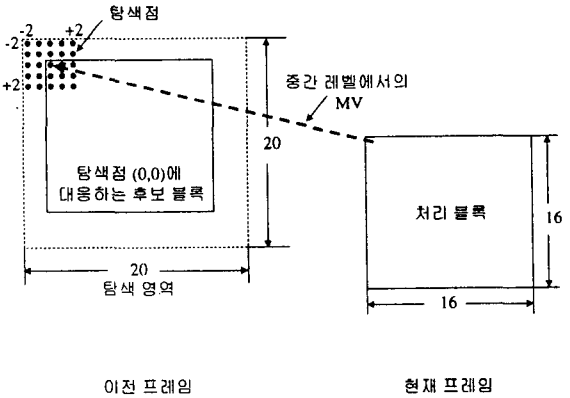


그림 5. 하위 레벨에서의 부분 탐색
Fig. 5. Local search in the lower level.

3) 하위 레벨

하위 레벨에서는 입력 영상을 그대로 이용하여 중간 레벨에서 얻어진 점을 중심으로 그림 5와 같이 16×16 크기의 MB 단위로 수평/수직 ±2 화소의 부분 탐색을 수행하여 최종 MV를 하나 결정하게 된다.

이상에서 살펴본 HSBMA3S의 전체 구조는 그림 1과 같다.

2. 제안한 구조

다수의 PE(processing element)로 이루어진 systolic arrary 구조는 움직임 추정 알고리즘을 구현하는데 적합하다. 이 논문에서 제안한 구조도 systolic array에 바탕을 두었다. Systolic array를 사용할 경우 PE의 개수는 일반적으로 MB의 크기나 탐색 영역의 크기와 비례하게 된다.

HSBMA3S는 3개의 레벨이 있고 각각의 레벨마다 처리하는 블록의 크기와 탐색 영역의 크기가 다르다. 가장 쉽게 생각할 수 있는 구조로는 각각의 레벨에 대해 탐색을 수행하는 서로 다른 크기의 systolic array를 따로 두는 것이다[2] 혹은 탐색 영역의 크기나 처리 블록의 크기가 같은 경우에는 두 레벨을 하나의 systolic array에서 순차적으로 처리할 수도 있다. [2]에서는 3개의 레벨 중에서 탐색 수행 시간이 적은 상위 두 레벨을 하나의 systolic array에서 처리하게 만들어서 systolic array의 개수를 줄였다. 본 논문에서는 하나의 탐색 기본 단위를 두어서, 모든 레벨에서 순차적으로 사용할 수 있도록 하여 PE의 개수를 최소화 하는데 중점을 두었다. 즉, 처리 블록의 크기는 각 레벨에서 처리하는 블록 크기 중 가장 작은 4×4 크기를 선택하고 탐색 영역의 크기는 각 레벨의 탐색 영역 중 가장 작은 수평/수직 ±2 화소를 선택하여 이를 처리할 수 있는 하나의 systolic array를 탐색 기본 단위로 두었다. 모든 레벨에서 이 탐색 기본 단위를 순차적으로 반복 사용함으로써 탐색을 수행할 수 있게 된다.

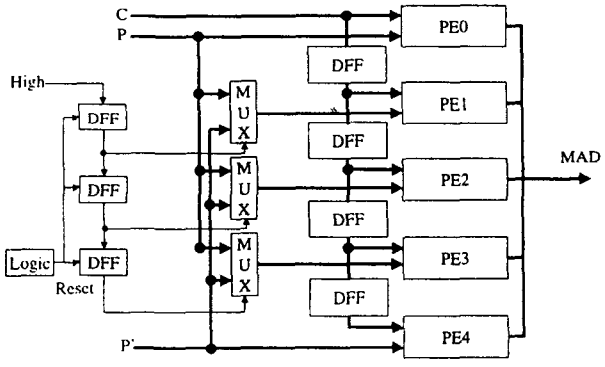


그림 6. 탐색 기본 단위
Fig. 6. Basic searching unit.

1) 탐색 기본 단위

탐색 기본 단위는 4×4 크기의 블록에 대해서 수평/수직 ±2 화소(25개의 탐색점)의 탐색을 수행할 수 있는 systolic array이다. 탐색 기본 단위의 구조로는 기존의 여러 전역 탐색을 수행할 수 있는 구조들을 그대로 적용할 수 있다. 본 논문에서는 [1]에서 제안된 구조를 약간 변경하여 탐색 기본 단위의 구조를 결정하였다. 탐색 기본 단위는 그림 6에서 보는 바와 같이 5개의 PE와 flip-flop, mux, 그리고 mux를 제어하는 간단한 로직 등으로 이루어져 있다.

2) 탐색 기본 단위의 적용

가. 상위 레벨에의 적용

상위 레벨은 4×4 크기의 블록에 대해서 수평/수직 ±8 화소의 탐색을 수행한다. 탐색 기본 단위는 한번에 수평/수직 각각 5개씩의 탐색점들에 대한 탐색을 수행하므로 탐색 기본 단위의 적용을 위해서 탐색 영역을 수평/수직 +7 화소(수평/수직 15개씩의 탐색점)로 약간 수정하였다. 탐색 기본 단위는 4×4 크기의 블록에 대해서 수평/수직 ±2 화소, 즉 가로, 세로 각각 5개씩의 탐색점들에 대해서 탐색을 수행하는 구조이다. 따라서 수평/수직 ±7 화소, 즉 가로, 세로 15개씩의 탐색점들에 대해서 적용하려면 우선 그림 7에서와 같이 수평/수직 ±7 화소 225개의 탐색점을 포함하는 영역을 가로, 세로 각각 3등분하여 총 9개의 영역으로 나눈다. 각각의 나누어진 영역은 가로, 세로 5개씩 총 25개의 탐색점들을 갖게 되어 탐색 기본 단위로 한번에 처리할 수 있는 크기가 된다.

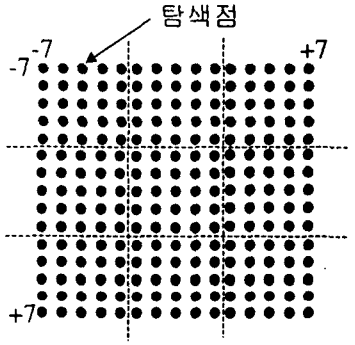


그림 7. 탐색 기본 단위의 상위 레벨에의 적용
Fig. 7. Application of the basic searching unit to the upper level.

이렇게 나누어진 영역에 대해서 차례대로 탐색 기본 단위를 이용해서 9번 처리하면 탐색 영역내의 모든 탐색점들에 대한 탐색을 수행할 수 있다. 탐색 기본 단위로 25개의 탐색점들을 처리하는데 총 84 사이클이 소요되고 상위 레벨을 처리하는 데는 모두 724사이클이 소요된다.

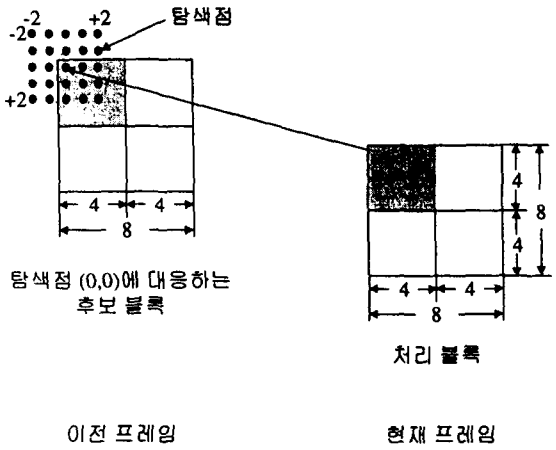


그림 8. 탐색 기본 단위의 중간 레벨에의 적용
Fig. 8. Application of the basic searching unit to the middle level.

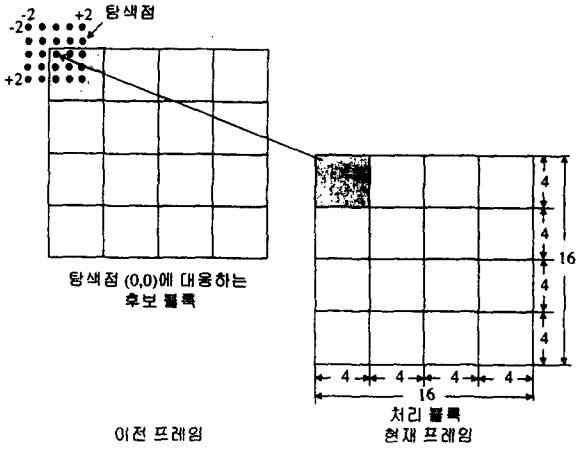


그림 9. 탐색 기본 단위의 하위 레벨에의 적용
Fig. 9. Application of the basic searching unit to the lower level.

나. 중간 레벨에의 적용

중간 레벨에서 처리하는 블록의 크기는 8×8이므로 여기서는 처리하는 기준 블록을 나누어 처리한다. 즉 그림 8과 같이 8×8 크기의 블록을 4×4 크기의 블록으로 4등분하여 탐색 기본 단위를 이용하여 4번 처리함으로써 탐색을 마칠 수 있다. 이때 탐색 기본 단위에서 얻어지는 MAD는 8×8 크기의 블록에 대한 완전한 MAD가 아니라 일부분인 4×4 크기의 블록에 대한 부분 MAD가 된다. 그림 8에서 색칠된 블록은 탐색점 (0,0)에 대한 MAD 계산 시 처리되는 탐색 영역의 블록을 나타낸다. 처리 블록의 나누어진 네 개의 각 블록을 처리할 때 탐색점 (0,0)에서 구한 네 개의 부분 MAD들을 모두 합치면 탐색점 (0,0)에 대한 8×8 블록의 완전한 MAD가 계산된다. 즉 한 탐색점에 대한 MAD는 탐색 기본 단위를 4번 사용하여 얻을 수 있다.

탐색 기본 단위를 사용하여 4번의 탐색을 수행하면 하나의 블록에 대한 처리가 끝나게 되고, 중간 레벨에서는 MV 후보가 모두 3개이므로 탐색 기본 단위를 총 12번 사용함으로써 탐색을 마치게 된다. 중간 레벨에서 MV 후보 3개에 대하여 각각 수평/수직 ±2 화소의 탐색을 수행하는 데에 총 964 사이클이 소요된다.

다. 하위 레벨에의 적용

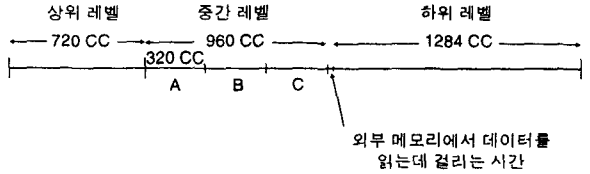
하위 레벨에서는 처리 블록의 크기는 16×16이므로 그림 9에서와 같이 4×4 크기의 블록으로 모두 16개로 나누어 순차적으로 처리한다. 나누어진 블록에 대해서 탐색 기본 단위를 순차적으로 16번 사용하게 되면 하위 레벨에 대한 탐색이 끝나게 된다. 중간 레벨에서와 마찬가지로 탐색 기본 단위로부터 나오는 부분 MAD들을 더하여 완전한 MAD를 구하는 과정이 필요하다. 하위 레벨에서 소요되는 시간은 총 1,284 사이클이다.

라. 중간 레벨에서의 탐색 순서

그림 10은 탐색 기본 단위를 이용하여 한 MB을 처리할 동안의 클럭 사이클을 나타낸 것이다.

MV 후보가 3개인 중간 레벨에서는 탐색 순서를 다음과 같이 정의하였다.

- 첫번째 주변 MV들의 상관 관계를 이용한 MV 후보에 대한 탐색
- 두번째 최소 MAD에 의한 MV 후보에 대한 탐색
- 세번째 두번째 최소 MAD에 의한 MV 후보에 대한 탐색



A. 주변 MV를 이용한 후보 처리
B. 최소 MAD를 가지는 후보 처리
C. 두 번째 최소 MAD를 가지는 후보 처리

그림 10. 한 MB 처리시 타이밍 도
Fig. 10. Timing diagram for processing a MB.

주변 MV들의 상관 관계를 이용한 MV 후보는 이전 MB에 대한 탐색이 끝나면 바로 구할 수 있다. 따라서 중간 레벨에서 상관 관계를 이용한 후보를 맨 처음 처리하게 되면 상위 레벨의 탐색이 완전히 끝나기 전에 미리 필요한 데이터를 읽어 들인 다음에 상위 레벨의 탐색이 끝남과 동시에 중간 레벨의 탐색을 시작할 수 있다. 즉 계층적 구조에서 피할 수 없는 레벨간의 의존성(inter level dependency)에 의한 시간 지연을 막을 수 있다. 중간 레벨과 하위 레벨사이에서의 레벨간의 의존성에 의한 시간

지연은 피할 수 없다. 이 경우 20 클럭 사이클 정도의 시간 지연이 생긴다. 따라서 한 MB을 처리하는데 총 2989 클러 사이클이 소요된다.

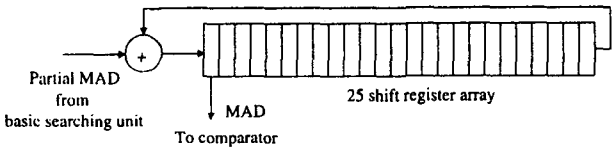


그림 11. 부분 MAD로부터 완전한 MAD를 계산하기 위한 shift register array
Fig. 11. shift register array for calculating a perfect MAD from a partial MAD.

3) 탐색 기본 단위 이외의 필요한 구조들

가. 완전한 MAD를 계산하기 위한 shift register array

앞에서 설명한 것과 같이 처리 블록을 여러 개의 블록으로 나누어 탐색 기본 단위로 처리하려면 부분 MAD를 저장할 수 있는 저장 공간이 필요하다. 25개의 탐색점들에 대해서 부분 MAD를 저장하고 새롭게 계산된 부분 MAD와 합하여 완전한 MAD를 계산하기 위해서 그림 11과 같이 25개의 shift register로 구성된 array를 사용한다. Shift register array는 블록을 나누어서 처리하는 중간 레벨과 하위 레벨에서만 사용하게 된다.

나. 주변 MV들의 상관 관계를 이용하여 MV 후보를 선택하기 위한 shift register array

그림 4에 표시된 3개의 주변 MB들의 MV들의 상관 관계를 이용하여 MV 후보를 선택하려면, 이미 탐색을 수행하여 계산된 이전 최종 MV들을 저장하고 있어야 한다. 3개의 주변 MV들 중 시간적으로 가장 우선한 블록에 해당되는 NMV1은 현재 처리중인 MB으로부터 22번째 이전 MB의 MV이고, NMV2와 NMV0는 각각 21번째, 1번째 이전 MB의 MV이다. 따라서 이 세개의 MV를 이용하기 위해서는 22개의 shift register가 필요하다. 최종 MV가 계산되면 차례대로 shift register array에 저장된다. 그 다음 그림 12와 같이 1번째, 21번째, 22번째 레지스터로부터 세 개의 MV를 가져와서 x성분, y성분 각각 독립적으로 중간값을 취해 MV 후보 하나를 얻게 된다.

이상에서 살펴 본 움직임 추정 칩의 개념적인 전체 블록 도는 그림 13과 같다.

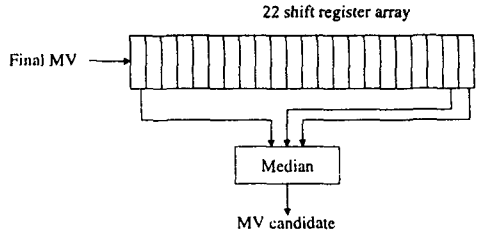


그림 12. 주변 MV들을 이용하여 MV 후보를 선정하기 위한 shift register array
Fig. 12. Shift register array for estimating MV candidate using neighboring MV's

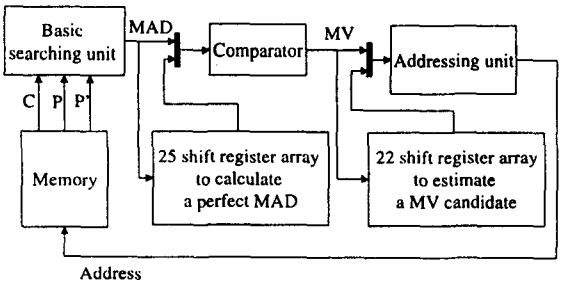


그림 13. 움직임 추정 칩의 개념적인 전체 블록도
Fig. 13. Functional block diagram of the ME chip

III. 구현 결과 및 고찰

본 논문에서 제안한 구조를 VHDL을 통하여 기술하고 시뮬레이션한 뒤 합성한 결과 20,000 게이트 이하로 구현 가능하였다. 실시간으로 처리하기 위해 필요한 최소의 동작 주파수와 외부 메모리의 대역폭을 구해 보면 표 1과 같다.

표 1. 최소 동작 주파수(영상 크기:352×288)와 외부 메모리 대역폭
Table 1. Minimum clock frequency (Picture size: 352×288) and external memory bandwidth

프레임 율		10 Hz	20 Hz	30 Hz
최소 동작 주파수(Hz)		11.88 M	23.76 M	35.64 M
외부 memory BW (Bytes/sec)	현재 frame data	1.33 M	2.66 M	3.99 M
	탐색 영역 data	9.38 M	18.75 M	28.13 M

표 2. 전역 탐색과 HSBMA3S의 비교 (영상 크기: 352×288) 프레임 율:30Hz)
Table 2. Comparison between FSBMA and HSBMA3S (Image size: 352×288, frame rate: 30Hz)

	FSBMA	HSBMA3S
초당 처리해야 할 MB 수 (N_{block})	11,880	11,880
초당 계산량 (MOPS)	12,850	176
필요한 PE 개수 (N_{PE})	512	5
계산 부분에 사용된 총 gate 수	210,929	2,552

표 2는 전역 탐색 기법과 HSBMA3S의 구현 시 복잡도를 비교한 것이다. 여기서 초당 처리해야 할 MB의 개수(N_{block})와 초당 계산량(OP), 그리고 필요한 PE의 개수(N_{PE})는 다음과 같이 계산된다.

$$N_{block} = \frac{352 \times 288 \times 30}{16^2} \quad (1)$$

$$OP_{FSBMA} = (2 \times 32)^2 \cdot 16^2 \cdot N_{block} \quad (2)$$

$$OP_{HSBMA3S} = (2 \times 7 + 1)^2 \cdot (\frac{16}{2^2})^2 + 3 \cdot (2 \times 2 + 1)^2 \cdot (\frac{16}{2})^2 + (2 \times 2 + 1)^2 \cdot 16 \quad (3)$$

$$N_{PE}^{FSBMA} = \frac{OP_{FSBMA}}{f_c} \quad (4)$$

$$N_{PE}^{HSBMA3S} = \frac{OP_{HSBMA3S}}{f_c} \quad (5)$$

여기서 f_c는 동작 주파수이다. 전역 탐색 기법의 경우 PE의 개수는 2-D array로 구현할 경우 계산된 결과보다 큰 가장 가까운 256의 배수를 선택한다. 동작 주파수가 50MHz 이하인 경우에는 N_{PE}^{FSBMA}는 512가 되고, HSBMA 3S의 경우에는 5개의 PE로 처리 가능하다.

표 2에서 보는 바와 같이 HSBMA3S로 구현했을 경우에 계산 부분(computational core)에 사용된 게이트 수는 전역 탐색 기법의 1/83 정도가 된다. 전역 탐색의 경우는 계산 부분이 전체의 대부분을 차지하지만, HSBMA3S의 경우는 제어 부분과 부가적으로 사용된 shift register array들이 계산 부분에 비해 상대적으로 더 많은 부분을 차지한다. HSBMA3S는 20,000 게이트 이하로 구현 가능하고 전역 탐색 기법으로 구현한 경우보다 게이트 수를 1/10 이하로 줄일 수 있다.

적용 범위는 다르지만, [2]에서 제안된 HSBMA3S와 비슷한 계층적 탐색 기법을 구현하기 위한 구조와 살펴보면 본 논문에서 제안된 구조는 각 레벨에서 공통으로 사용할 수 있는 단일 systolic array를 둠으로써 systolic array 사용에 있어서 더 효율적이다. 그리고 적은 게이트 수로 구현 가능하면서 다른 고속 알고리즘들에 비해서 알고리즘 자체의 성능이 더 뛰어나다는 장점을 가진다.

Ⅳ. 결 론

HSBMA3S는 전역 탐색 기법의 성능을 그대로 유지하면서 계산량은 현저하게 적은 알고리즘으로써 하드웨어 구현 시 적은 칩 사이즈로 전역 탐색 기법과 비슷한 성능을 내는 움직임 추정칩을 구현할 수 있다. 본 논문에서는 탐색 기본 단위를 두고 이를 반복적으로 사용하는 구조로써 HSBMA3S에 적합하며 systolic array의 크기를 줄일 수 있는 VLSI 구조를 제안하였다. 탐색 기본 단위로는 기존의 여러 전역 탐색을 수행하는 구조를 그대로 적용할 수 있으며, 구조의 선택에 따라 속도와 게이트 수 사이의 절충이 가능하다. 본 논문에서는 저 전송률의 영상에 적용하기 위해 실시간 처리 가능한 범위 내에서, 속도 보다는 크기를 줄이는데 중점을 두어 5개의 PE로 이루어진 구조를 탐색 기본 단위로 선택하였다. 선택한 구조는 간단하고 PE의 효율이 거의 100%에 가깝고 데이터 I/O핀 수가 적은 장점이 있다. 제안한 구조를 VHDL을 사용하여 기술하고 합성한 결과 총 20,000 게이트 이내로 움직임 추정 칩의 구현이 가능하였다. 제안된 구조는 352×288 크기의 영상, 탐색 영역 수평/수직 -32~+31 화소에 대해서 30Hz까지 실시간 처리가 가능하다.

참 고 문 헌

[1] K. M. Yang, M. T. Sun and L. Wu, "A family of VLSI gesigns for the motion compensation block-matching algorithm," *IEEE Trans. on Circuits and Systems*, vol. 36, no. 10, pp. 1317-1325, Oct. 1989.

[2] L. D. Vos, "VLSI-architectures for the hierarchical block-matching algorithm for HDTV applications," *in SPIE*, vol. 1360 (Visual Communications and Image Processing), pp. 398-409, 1990.

[3] L.D. Vos and M. Stegherr, "Parameterizable VLSI architectures for the full-search block-matching algorithm," *IEEE Trans. on Circuits and Systems*, vol. 36, no. 10, pp. 1309-1316, Oct. 1989.

[4] T. K omarek and P. Pirsch, "array architectures for block matching algorithms," *IEEE Trans. on Circuits and Systems*, vol. 36, no. 10, pp. 1301-1308, Oct. 1989.

[5] H. Yeo and Y. H. Hu, "A novel modular systolic array architecture for full-search block matching motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 5, no. 5, pp. 407-416, Oct. 1995.

[6] Y. S. Jehng, L. G. Chen, and T. D. Chiueh, "An efficient and simple VLSI tree architecture for motion estimation algorithms," *IEEE Trans. on Signal Processing*, vol. 41, no. 2, pp. 889-900, Feb. 1993.

[7] G. Gupta and C. Chakrabarti, "Architectures for hierarchical and other block matching algorithms," *IEEE Trans. on Circuits and Systems for Video*

Technology, vol. 5, no. 6, pp. 477-489, Dec. 1995.
[8] C. Y. Lee, "A novel VLSI architecture for block matching algorithms," *in Proc. of ICASSP' 95*, vol. 3, no. 5, pp. 639-651, Sep. 1994.
[9] C. H. Hsieh and T. P. Lin, "VLSI architecture for block-matching motion estimation algorithm," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 2, no. 2, pp. 169-175, Jun. 1992.
[10] B. M. Wand, J. C. Yen, and S. Ching, "Zero waiting-cycle hierarchical block matching algorithm and its array architectures," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no, 1, pp. 18-27, Feb. 1994.
[11] H. M. Jong, L. G. Chen, and T. D. Chiueh, "Parallel architectures for 3-step hierarchical search block-matching algorithm," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 407-415, Aug. 1994.
[12] J. Chalidabhongse and C. C. J. Kuo, "Fast motion vector estimation using multiresolution-spatio-temporal correlations," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 477-4888, Jun. 1997.
[13] S. C. Cheng and H. M. Hang "A comparison of block-matching algorithms mapped to systolic-array implementation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 741-757, Oct. 1997.
[14] K. W. Lim and J. B. Ra, "Improved hierarchical search block matching alogorithm by using multiple motion vector candidates," *IEEE Electronics letters*, pp. 1771-1772, Oct. 1997.

저 자 소 개

이 재 헌

1992. 3. ~ 1996. 2. 한국과학기술원 전기 및 전자공학과(공학사)
1996. 3. ~ 1998. 2. 한국과학기술원 전기 및 전자공학과(공학석사)
1998. 3. ~ 현재 한국과학기술원 전기 및 전자공학과 박사 과정 재학중
주관심분야:저 전송률에서의 영상 압축, 움직임 추정 기법

나 종 범

1975. 2. 서울대학교 전자공학과 졸업
1977. 2. 한국과학기술원 전기 및 전자공학과 석사학위 취득
1983. 2. 한국과학기술원 전기 및 전자공학과 박사학위 취득
1983. 7. ~ 1987. 6. 미국 Columbia 대학교 연구 조교수
1987. 7. ~ 현재 한국과학기술원 전기 및 전자공학과 교수
주관심분야:디지탈 영상 처리, 비디오 신호 처리, 3차원 시각화