

저전송률 동영상 압축을 위한 새로운 계층적 움직임 추정기의 VLSI 구조

이 재 헌 (李 在 憲), 나 중 범 (羅 鍾 範)

한국과학기술원 전기 및 전자공학과

전화 : (042) 869-5434 / 팩스 : (042) 869-8360

A New VLSI Architecture of a Hierarchical Motion Estimator for Low Bit-rate Video Coding

Jaehun Lee, Jong Beom Ra

Dept. of E.E., Korea Advanced Institute of Science and Technology

E-mail : jhlee@issserver.kaist.ac.kr, jbra@ee.kaist.ac.kr

Abstract

We propose a new hierarchical motion estimator architecture that supports the advanced prediction mode of recent low bit-rate video coders such as H.263 and MPEG-4. In the proposed VLSI architecture, a basic searching unit (BSU) is commonly utilized for all hierarchical levels to make a systematic and small sized motion estimator. Since the memory bank of the proposed architecture provides scheduled data flow for calculating 8×8 block-based sum of absolute difference (SAD), both a macroblock-based motion vector (MV) and four block-based MVs are simultaneously obtained for each macroblock in the advanced prediction mode. The proposed motion estimator gives similar coding performance compared with full search block matching algorithm (FSBMA) while achieving small size and satisfying the advanced prediction mode.

I. 서론

움직임 추정은 동영상 압축 알고리즘에서 매우 중요한 부분이다. 일반적으로 움직임 추정에는 최적의 성능을 가지는 전역 탐색 블록 정합 기법 (FSBMA)이 사용된다. 하지만 FSBMA는 과도한 계산량으로 인해 실시간 동영상 코덱을 하드웨어로 구현하는데 있어서 칩 사이즈가 커지는 등의 단점을 가지고 있다. 이런 이유로 적은 계산량을 가지는 많은 고속 탐색 기법들이 제안되어 왔다. 이들 고속 탐색 기법들은 적은 계산량과 낮은 메모리 대역폭을 가지기 때문에 하드웨어로 구현 시 많은 이점을 가진다. 이러한 고속 탐색 기법들은 크게 두 가지 방식으로 구분할 수 있다. 첫째는 탐색점의 개수

를 줄여 계산량을 줄인 three-step search (TSS)에 바탕을 둔 탐색 기법들이고 둘째는 해상도를 줄인 여러 장의 영상을 이용한 계층적 탐색 기법들이다. 이들 중 TSS에 바탕을 둔 고속 탐색 기법들은 후자에 비해 더 적은 계산량과 더 간단한 구조를 가지기 때문에 하드웨어로 구현하기 용이하나 전역 탐색 기법에 비해 큰 성능 저하가 뒤따른다는 단점을 가지고 있다 [1]. 반면에 해상도가 다른 여러 장의 영상을 이용한 계층적 탐색 기법들은 TSS에 바탕을 둔 탐색 기법들보다는 더 좋은 성능을 나타내지만 하드웨어 구현보다는 소프트웨어 구현이 더 적합하였다. 일반적으로 움직임 추정기의 하드웨어 구현에는 구조의 간단함으로 인해 전역 탐색 기법이 선호되어 왔다 [3].

본 논문에서는 계층적 탐색 기법의 장점들을 살려 효율적인 움직임 추정기를 하드웨어로 구현하기 위해 계층적 탐색 기법에 적합한 새로운 VLSI 구조를 제안한다. 제안된 움직임 추정기는 매크로블록 별 SAD 값뿐만 아니라 8×8 블록 별 SAD 값들을 계산한다. 이 SAD 값들을 이용하여 H.263이나 MPEG-4의 advanced prediction mode를 지원하여 한 매크로블록 당 하나의 16×16 매크로블록 움직임 벡터뿐만 아니라 네 개의 8×8 블록 별 움직임 벡터들을 구하게 된다 [4-5]. 제안된 움직임 추정기는 매크로블록 별 SAD 값들을 계산하는 동시에 8×8 블록 별 SAD 값들을 구할 수 있는 입력 데이터 순서와 processing element (PE) 어레이 구조를 이용함으로써 매크로블록 별 움직임 벡터와 네 개의 블록 별 움직임 벡터들을 동시에 얻을 수 있다. 모의 실험 결과 제안된 움직임 추정기는 전역 탐색 기법과 비슷한 성능을 보였으며 하드웨어 구현 시 더 적은 개수의 계

이트로 구현 가능하였다.

II. 본론

1. 알고리즘

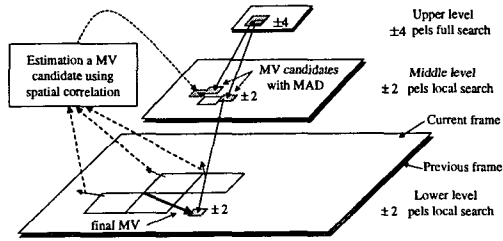


그림 1. 제안된 계층적 움직임 추정기의 알고리즘 개요

Figure 1. Algorithm overview of the proposed hierarchical motion estimator.

다수의 움직임 벡터 후보와 움직임 벡터들의 상관 관계를 이용한 계층적 탐색 기법을 채택하였다 [2]. 이 계층적 탐색 기법은 그림 1 과 같이 세 개의 계층, 상위, 중간, 하위 계층, 을 가진다. 탐색 영역이 [-16, +15] 인 경우를 예로 들어 설명하면, 상위 단계에서는 입력 영상을 4:1 로 subsampling 하여 저 해상도의 영상을 얻은 뒤 4x4 블록 단위로 탐색 영역 [-4, +4]에 대해서 전역 탐색을 수행한다. 상위 단계의 탐색 수행 결과 최소 SAD 값을 가지는 두 점을 중간 단계에서의 탐색 초기점으로 삼는다. 중간 단계에서는 입력 영상을 2:1 로 subsampling 한 영상에서 8x8 블록 단위로 상위 단계에서 얻은 초기 탐색점을 중심으로 [-2, +2] 탐색 영역에 대해 전역 탐색을 수행한다. 중간 단계에서는 초기 탐색 중심점으로 세 점을 사용한다; 두 점은 상위 단계에서 SAD 값을 이용하여 구한 점들이고 나머지 한 점은 주변 매크로블록들의 움직임 벡터들의 공간적 상관 관계를 이용하여 예측한 점이다. 이들 세 탐색 중심점에 대해 각각 전역 탐색을 수행한 뒤 최소 SAD 값을 가지는 한 점을 찾아내어 하위 단계의 탐색 초기점으로 사용한다. 하위 단계에서는 중간 단계에서 구해진 탐색 초기점을 중심으로 16x16 블록 단위로 [-2, +2] 탐색 영역에 대해 전역 탐색을 수행하여 최소 SAD 값을 가지는 점을 최종 움직임 벡터로 선택한다.

하위 단계의 탐색 수행 시 16x16 블록 단위의 탐색 뿐만 아니라 8x8 블록 단위의 탐색을 병행함으로써 MPEG-4 등의 advanced prediction mode 를 지원한다. 세 단계를 가지는 계층적 탐색 구조에서 마지막 미세 조정 단계인 하위 단계에서 8x8 블록에 대한 탐색을 수행하기 때문에 구해진 네 개의 8x8 블록 별 움직임 벡터들은 중간 단계까지 구해진 움직임 벡터를 중심으로 [-2, +2] 범위에 놓이게 된다. 결국 네 개의 블록 별 움직임 벡터들은 MPEG-4 VM 에서 권고한 대로 매크로블

록에 대한 움직임 벡터 주위에 분포하게 된다 [5].

2. 제안된 구조

2.1. 구조 개요

제안된 움직임 추정기는 그림 2 에서 보는 바와 같이 탐색 기본 단위 (Basic searching unit: BSU), shift register array, 비교기, on-chip 메모리, 주소 발생기 등으로 이루어져 있다. BSU 는 4x4 블록 단위로 [-2, +2] 탐색 영역에 대한 탐색을 수행하는 모듈로써 모든 계층에서 공통적으로 사용함으로써 움직임 추정기의 면적을 줄일 수 있다. Shift register array 는 BSU 에서 구해진 4x4 블록 별 SAD 값들을 이용하여 8x8 블록 별 SAD 및 매크로블록 별 SAD 값들을 구하기 위해 사용된다. On-chip 메모리는 탐색에 필요한 데이터를 적절한 순서로 BSU 에 공급한다.

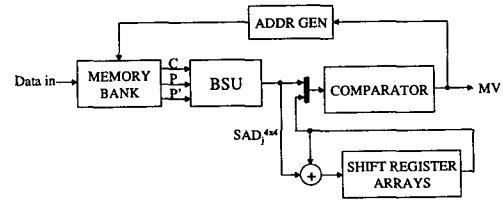


그림 2. 제안된 움직임 추정기의 전체 블록도

Figure 2. Block diagram of the proposed motion estimator

2.2. 탐색 기본 단위 (BSU)

일반적으로, 계층적 탐색 기법은 각 계층마다 처리하는 블록 크기와 탐색 영역의 크기가 서로 다르다. Processing element (PE)의 개수는 처리하는 블록의 크기 혹은 탐색 영역의 크기에 비례하므로 각 단계마다 서로 다른 개수의 PE 가 필요하게 되어 면적이나 PE 의 효율 면에서 효과적인 하드웨어 구조를 설정하기 어렵다.

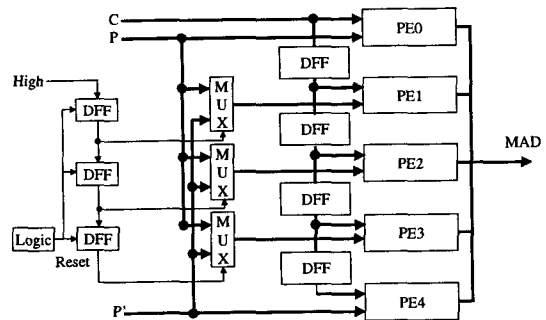


그림 3. 탐색 기본 단위

Figure 3. Basic searching unit

면적이나 PE 의 효율면에서 효과적인 하드웨어 구조

를 위해서 작은 크기의 BSU를 두고 이를 반복적으로 사용하여 탐색을 수행한다. BSU는 4x4 블록 단위로 [-2, +2] 탐색 영역에 대한 탐색을 수행하는 모듈이다; 4x4 블록은 세 단계에서 처리하는 블록의 크기 중 가장 작은 크기이고 탐색 영역 [-2, +2]는 세 단계의 탐색 영역 중 가장 작은 탐색 영역에 해당된다. BSU는 그림 3에서 보는 것처럼 5개의 PE, flip-flops, MUXs, 탐색 영역 데이터의 흐름 제어를 위한 간단한 로직 등으로 구성된 1-D systolic array 구조이다. BSU는 4x4 블록 단위의 SAD 값을 계산하므로 매크로블록 별 SAD 및 블록 별 SAD 값들을 구하기 위해서 shift register array와 덧셈기를 사용한다.

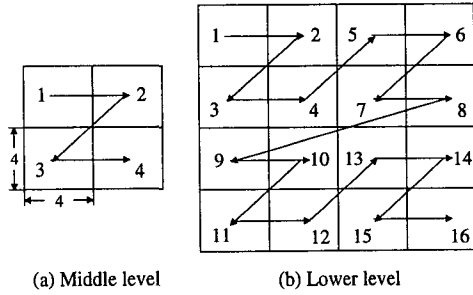


그림 4. 중간 단계 및 하위 단계에서의 4x4 블록 처리 순서
Figure 4. Sub-block ordering for the middle and lower levels.

표 1은 각 단계에서 BSU를 이용하여 탐색을 수행하는 방법을 나타낸다. 상위 단계에서는 탐색 영역 [-4, +4]를 네 영역으로 나누어 네 번에 걸쳐 탐색을 수행하게 된다. 각각의 나누어진 영역은 BSU가 처리할 수 있는 탐색 영역 [-2, +2]에 맞게 25개의 탐색점을 포함한다. 따라서 BSU를 사용하여 네 번 탐색을 수행함으로써 상위 단계에 대한 탐색을 마치게 된다. 중간 단계와 하위 단계에서는 상위 단계와는 달리 탐색 영역 대신 처리 블록을 나누어 처리한다. 그림 4에서 보는 것처럼 중간 단계에서는 8x8 블록을 네 개의 4x4 블록으로 나누어 각각의 4x4 블록에 대한 탐색을 BSU를 사용하여 순차적으로 수행한다. 마찬가지로 하위 단계에서는 16x16 블록을 16개의 4x4 블록으로 나누어 각각의 4x4 블록에 대한 탐색을 순차적으로 수행하게 된다.

2.3. SAD 계산 및 움직임 벡터 결정

상위 단계에서는 4x4 블록 단위의 SAD 값들을 사용하여 다음 단계의 탐색 초기점을 결정하므로 BSU에서 구해진 4x4 블록 별 SAD 값들을 그대로 사용한다. 중간 단계와 하위 단계에서는 BSU에서 구한 4x4 블록 별 SAD 값들을 이용하여 8x8 블록 단위의 SAD 및 매크로블록 단위의 SAD 값들을 각각 구한 뒤 최소 SAD 값을 갖는 탐색점을 선택하여야 한다. 매크로블록 단위

의 SAD 및 8x8 블록 단위의 SAD는 식 (1.a)와 식 (1.b)와 같이 표시된다.

$$SAD^{16 \times 16}(u, v) = \sum_{i=0}^{15} \sum_{j=0}^{15} |C(i, j) - P(i+u, j+v)| \quad (1.a)$$

$$SAD^{8 \times 8}(u, v) = \sum_{i=0}^7 \sum_{j=0}^7 |C(i, j) - P(i+u, j+v)| \quad (1.b)$$

여기서 $C(i, j)$ 는 현재 처리 중인 매크로블록의 데이터, $P(i+u, j+v)$ 는 이전 프레임의 탐색 영역 데이터, (u, v) 는 움직임 벡터 후보를 나타낸다.

중간 단계 및 하위 단계에서는 BSU는 그림 4에 표시된 순서대로 4x4 블록 단위의 탐색을 수행한다. 이 경우 SAD는 다음과 같이 표시된다.

$$SAD^{16 \times 16}(u, v) = \sum_{i=1}^4 SAD_i^{8 \times 8}(u, v) \quad (2)$$

$$SAD_i^{8 \times 8} = \sum_{j=(i-1) \times 4}^{i \times 4 - 3} SAD_j^{4 \times 4}(u, v), \quad 1 \leq i \leq 4 \quad (3)$$

여기서 $SAD_j^{4 \times 4}(u, v)$ 는 그림 4에서 j번째 처리되는 4x4 블록에 대한 SAD 값, $SAD_i^{8 \times 8}(u, v)$ 는 8x8 블록에 대한 SAD 값을 나타낸다. 중간 단계와 하위 단계에서는 (u, v) 는 [-2, +2]로 제한된다.

중간 단계에서는 BSU에서 구한 네 개의 $SAD_j^{4 \times 4}(u, v)$ 값들을 더해서 $SAD_i^{8 \times 8}(u, v)$ 값들을 구한 뒤 비교하여 최소값을 구하는 방식으로 탐색을 수행한다. 하위 단계에서는 advanced prediction mode를 지원하기 위해서 매크로블록 단위의 SAD 값 뿐만 아니라 8x8 블록 단위의 SAD 값들도 필요하다. 따라서 $SAD_1^{8 \times 8}(u, v)$, $SAD_2^{8 \times 8}(u, v)$, $SAD_3^{8 \times 8}(u, v)$, $SAD_4^{8 \times 8}(u, v)$ 및 $SAD^{16 \times 16}(u, v)$ 값들을 구해야 한다. 그림 4에 표시된 순서대로 탐색을 수행함으로써 이들 값들을 순차적으로 얻을 수 있게 된다. 하드웨어 구현 시에는 $SAD_j^{4 \times 4}(u, v)$ 값은 BSU를 통해 계산되고 $SAD_i^{8 \times 8}(u, v)$ 값들은 shift register array와 덧셈기를 이용하여 네 개의 $SAD_j^{4 \times 4}(u, v)$ 값들을 더함으로써 구해진다. 마찬가지로 방법으로 $SAD^{16 \times 16}(u, v)$ 값들은 16개의 $SAD_j^{4 \times 4}(u, v)$ 값들을 shift register array와 덧셈기를 이용하여 더하여 구한다. 이들 SAD 값들이 구해지면 매크로블록에 대한 움직임 벡터와 네 개의 8x8 블록에 대한 움직임 벡터들은 다음 식과 같이 표시되며 비교기를 이용하여 구한다.

$$MV^{16 \times 16} = \arg \min_{(u, v)} SAD^{16 \times 16}(u, v) \quad (4)$$

$$MV_i^{8 \times 8} = \arg \min_{(u, v)} SAD_i^{8 \times 8}(u, v) \quad (5)$$

3. 성능 비교 및 구현 결과

제안된 움직임 추정기를 저전송률 동영상 압축 표준인 H.263에 적용하여 다양한 테스트 영상에 대한 모의 실험 결과를 얻었다. 표 2는 advanced prediction mode에 대해 전역 탐색 기법을 이용한 경우와 제안된 움직임

추정기를 사용한 경우의 코딩 결과 비트율과 화질 (PSNR)을 비교한 것이다. 제안된 움직임 추정기는 전역 탐색과 거의 유사한 성능을 가짐을 알 수 있다. 또한 제안된 움직임 추정기를 합성 가능한 VHDL 코드로 기술한 뒤 테스트하고 합성한 결과 2만 5천여 개의 게이트가 소요되었다. 제안된 움직임 추정기를 Xilinx의 FPGA를 이용하여 구현하고 H.263 부호화기의 나머지 부분은 SA-110 ARM 프로세서를 이용한 소프트웨어 구현 방식으로 H.263 부호화기를 구현하여 제안된 움직임 추정기의 동작을 검증하였다.

III. 결론

저 전송률 동영상 압축 표준인 H.263이나 MPEG-4 등의 advanced prediction mode를 지원하고 면적면에서 효율적인 새로운 계층적 움직임 추정기의 VLSI 구조를 제안하였다. 제안된 움직임 추정기는 매크로블록 단위의 SAD 값들을 계산하는 동시에 추가적인 사이클 소요없이 8x8 블록 단위의 SAD 값들을 구할 수 있다. 이 SAD 값들을 이용하여 매크로블록 단위의 움직임 벡터 뿐만 아니라 8x8 블록 단위의 움직임 벡터를 구함으로써 H.263이나 MPEG-4 등의 advanced prediction

mode를 지원한다. 또한 BSU를 효율적으로 사용함으로써 전역 탐색 기법보다 훨씬 더 작은 게이트 수로 하드웨어 구현이 가능하면서도 전역 탐색 기법과 거의 유사한 성능을 보인다.

참고 문헌

- [1] S.-C. Cheng and H.-M. Hang, "A comparison of block-matching algorithms mapped to systolic-array implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 5, Oct. 1997.
- [2] K. W. Lim and J. B. Ra, "Improved hierarchical search block matching algorithm by using multiple motion vector candidates," *IEE Electronics letters*, pp. 1771-1772, Oct. 1997
- [3] L. D. Vos and M. Stegherr, "Parameterizable VLSI architectures for the full-search block-matching algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 36, no. 10, pp. 1309-1316, Oct. 1989.
- [4] Draft ITU-T Recommendation H.263, "Video coding for low bitrate communication," Mar. 1996.
- [5] Draft ISO/IEC MPEG-4 Video Verification Model Ver. 6.0, "Coding of moving pictures and associated audio information," Feb. 1996.

표 1. 제안된 움직임 추정기를 위한 탐색 기본 단위의 반복적인 사용 예

Table 1. Repetitive usage of the basic searching unit (BSU) for the proposed hierarchical motion estimator (we assume ± 16 search); the BSU performs ± 2 search for a 4x4 block.

계층	탐색 초기점 개수	처리 블록 크기	탐색 영역	BSU 사용 회수	비고
상위 단계	1	4x4	[-4, +4]	4	4 x Search range of the BSU
중간 단계	3	8x8	[-2, +2]	3 x 4	세 개의 탐색 초기점 4 x Block size of the BSU
하위 단계	1	16x16	[-2, +2]	16	16 x Block size of the BSU

표 2. 전역 탐색 기법과 제안된 움직임 추정기의 성능 비교 (QCIF 포맷, 300 프레임, advanced prediction mode on, [-16, +15] 정수 단위 탐색 후 동일한 반 화소 단위 탐색 수행)

Table 2. Comparison between FSBMA and the algorithm of the proposed motion estimator (QCIF format, 300 frames(IPPP...) advanced prediction mode on, [-16, +15] integer pel search followed by same half pel search)

Frame rate	15 Hz				30 Hz			
	FSBMA		Proposed		FSBMA		Proposed	
Sequences	PSNR (dB)	Bit-rate (Kbps)	PSNR (dB)	Bit-rate (Kbps)	PSNR (dB)	Bit-rate (Kbps)	PSNR (dB)	Bit-rate (Kbps)
Hall monitor	37.81	66.28	37.81	66.49	35.65	61.15	35.66	61.41
Container ship	35.34	52.23	35.35	52.20	34.56	67.68	34.55	67.93
Mother and daughter	37.91	57.00	37.92	57.45	36.01	50.03	36.02	50.18
Akiyo	39.87	45.74	39.88	45.96	38.61	53.88	38.62	53.83
Foreman	31.37	59.02	31.22	62.16	29.74	59.23	29.61	61.84
Coast guard	29.05	54.53	29.05	55.72	28.25	59.54	28.24	60.65
News	33.74	55.77	33.71	57.23	32.39	57.67	32.38	59.08
Silent voice	34.25	59.18	34.25	59.93	32.83	62.52	32.82	62.87