H.264/AVC 용 움직임 추정 알고리즘 및 하드웨어 구조

이 재 헌 (李 在 憲), 이 남 숙 (李 男 淑)
三星電子 株式會社 디지털미디어 硏究所
전화: (031) 200-3757 / 팩스: (031) 200-3147

MOTION ESTIMATION ALGORITHM AND HARDWARE ARCHITECTURE FOR H.264/AVC

Jae Hun Lee and Nam Suk Lee
Digital Media R&D center, Samsung Electronics Co., Ltd., Republic of Korea
E-mail: jh717.lee@samsung.com

Abstract

This paper presents a variable block size motion estimation (ME) algorithm and hardware architectures dedicated to H.264/AVC. Proposed ME architecture can achieve real-time processing for 720×480@30Hz with search range of [-64, +63] in the horizontal and [-32, +31] in the vertical direction at integer-pel accuracy and upto 7 reference frames at the operating frequency of 54MHz.

I. 서론

ITU-T VCEG와 ISO/IEC MPEG에 의해 최근에 개발된 H.264/AVC에는 H.263이나 MPEG-4 등의 기존 비디오 코덱에 비해 RD (rate-distortion) 관점에서 코딩 효율을 높이기 위한 많은 새로운 기법들이 적용되었다 [1]. 그 중에서 다양한 크기의 블록과 다수의 참조 프레임을 이용한 움직임 추정 부분은 인코더에서 가장 시간이 많이 소요되는 복잡한 부분이다. 따라서, 실시간 어플리케이션을 위해서는 고속의 움직임 추정 알고리즘과 그에 맞는 하드웨어 구조의 개발이 필수적이다.

그 동안 H.263이나 MPEG-4를 위한 움직임 추정 알고리즘과 하드웨어가 많이 개발되어 왔다 [2]. 하지만 이들은 H.264/AVC에서 지원하는 다양한 블록 크기 (16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4)에 대한 움직임 추정을 지원하지 않는다.

본 논문에서는 [2]에서 제안되었던 다해상도 계층적 움직임 추정 알고리즘 (Multi-Resolution Multiple Candidate Search: MRMCS)을 H.264/AVC의 다양한 블록 크기에 대응하도록 확장한 알고리즘과 하드웨어 구조를 제안한다. 이후 본 논문의 구성은 다음과 같다. 먼저 제 2절에서는 제안하는 움직임 추정 알고리즘인 AMRMCS에 대해서 설명한다. 제 3절과 4절에서는 AMRMCS의 하드웨어 구조와 구격에 대해서 기술한다. 마지막으로 5절과 6절에서 결론과 추후 연구에 대해서 논한다.

II. AMRMCS 알고리즘

그림 1은 제안하는 움직임 추정 알고리즘인 Advanced MRMCS (AMRMCS)를 나타낸다. AMRMCS는 상위, 중간, 하위 레벨로 구성된다. 상위 레벨에서는 원래의 영상을 수평, 수직 방향으로 4:1로 downsampling한 영상을 이용하여 4×4 서브-블록 단위로 움직임 추정을 수행한다. 탐색 영역의 크기가 수평, 수직으로 $[-p, +p]$라고 가정하면 상위 레벨에서의 탐색 영역은 $[-p/4, +p/4]$가 된다. 상위 레벨에서 탐색을 수행하여 정합 기준이 최소가 되는 2 점을 구한다. 그리고 인접한 매크로블록들 (왼쪽, 위쪽, 위 오른쪽 매크로블록)의 움직임 벡터 (MV)들의 중간값을 이용하여 한 점을 구한다. 이렇게 구해진 3개의 점을 중간 레벨의 탐색 중심점으로 이용한다. 중간 레벨에서는 원래의 영상을 수평, 수직으로 2:1로 downsampling한 영상에서 위의 3개의 탐색 중심점을 중심으로 8×8 블록 단위로 수평, 수직으로 [-2, +2]의 부분 탐색을 수행한다. 탐색 수행 후 정합 기준이 최소가 되는 하나의 점을 선택하여 하위 레벨의 탐색 중심점으로 이용한다.

하위 레벨에서는 원래의 영상에서 16×16 매크로블록 단위로 수평, 수직 [-2, +2]의 부분 탐색을 수행한다. H.264/AVC의 다양한 블록 크기 (16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4)를 지원하기 위해서 4×4 서브-블록 단위로 정합 기준을 구하고 이를 더하여 4×4 서브-블록보다 더 큰 블록에 대한 정합 기준을 계산하는 방식을 사용한다.

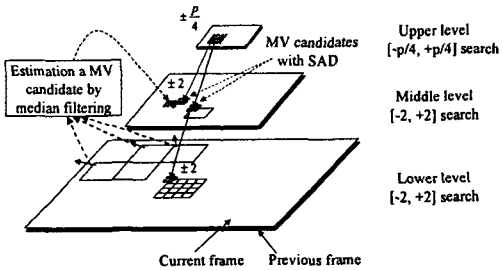

그림 1. AMRMCS

정합 기준으로는 JVT 소프트웨어인 JM [3]에서 사용하는 방식과 동일하게 SAD (Sum of absolute differences)와 MV를 부호화하는데 필요한 비트량을 사용한다. 하드웨어 구현 시 병렬 처리가 가능하도록 하기 위해서 정합 기준에 사용되는 MV를 부호화하는 데 필요한 비트량을 계산할 때, 정확한 MVP (Motion vector predictor)를 사용하는 대신 매크로블록 단위의 MVP를 사용한다. 또한, 인접한 매크로블록 간의 겹치는 탐색 영역 데이터를 재사용하여 외부 메모리 access를 줄이기 위해서 탐색 중심점은 MVP 대신에 (0, 0)을 사용한다. 탐색 영역의 크기가 큰 경우, MVP 대신 (0, 0)을 사용함으로 인한 PSNR의 저하는 거의 없다. 탐색 영역의 크기가 수평 [-64, +63], 수직 [-32, +31]인 경우 탐색 영역 데이터를 재사용함으로 인해 외부 메모리 access를 88.89% 줄일 수 있다.

그림 2는 Football과 Susie 테스트 영상에 대한 전역 탐색 (Full search)과 AMRMCS의 RD 곡선을 나타낸다. 비디오 코덱은 JM6.1e를 이용하고 I-P-P-P, 하나의 참조 프레임, CAVLC, RD 최적화 off, 탐색 영역은 각각 [-64, +63]와 [-16, +15]로 설정하였다. 그래프에서 보듯이 전역 탐색과 AMRMCS의 성능 차이는 크지 않음을 알 수 있다.

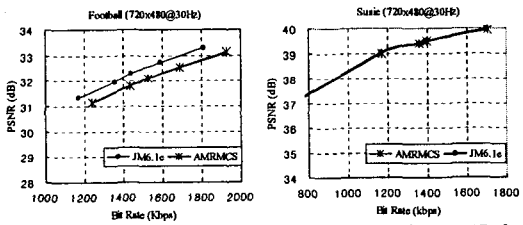

그림 2. 전역 탐색(JM6.1e)과 AMRMCS의 RD 곡선

III. 하드웨어 구조

하드웨어 설계 목표는 다음과 같다. 입력 영상: 720×480@30Hz, 탐색 영역: 수평 [-64, +63], 수직 [-32, +31].

제안하는 하드웨어 구조는 그림 3과 같이 기본 탐색 유닛 (Basic searching unit: BSU), 현재 매크로블록 픽셀을 저장하기 위한 2개의 16×16 레지스터 어레이, 탐색 영역 픽셀을 위한 2개의 12×12 레지스터 어레이, 전체 탐색 영역 픽셀들을 저장하기 위한 내부 SRAM, 4×4 정합 기준을 더하여 더 큰 블록들에 대한 정합 기준들을 구하고 정합 기준들을 비교하는 부분 등으로 이루어져 있다. 현재 매크로블록과 탐색 영역을 저장하는 레지스터 어레이는 더블 버퍼링을 위해서 2개씩 있으며, 내부 SRAM은 탐색 영역 데이터의 재사용, 더블 버퍼링 및 탐색 영역 데이터의 동시 access를 가능하게 하기 위해 10개의 SRAM으로 나누어진다.

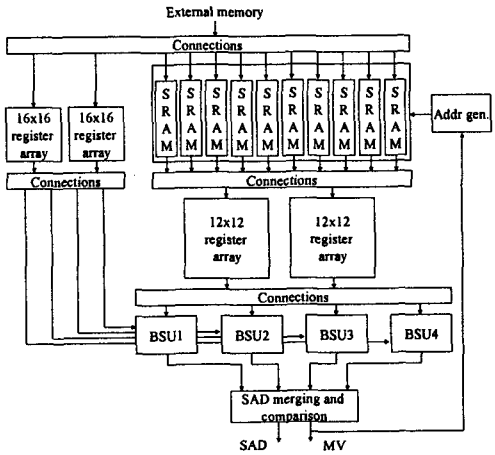

그림 3. 하드웨어 구조 I

III-1. 기본 탐색 유닛 (BSU)

우선, 효율적인 하드웨어 구조를 위해서 BSU의 개념을 사용한다. AMRMCS는 3개의 탐색 레벨을 가지며 각각의 레벨에서의 탐색 영역 크기 및 탐색 블록 크기가 서로 다르다. 3개의 탐색 레벨에서 가장 작은 탐색 블록 크기와 탐색 영역 크기는 각각 4×4와 [-2, +2]가 된다. 따라서, 4×4 서브-블록 단위로 [-2, +2] 탐색을 수행하는 BSU를 3개의 모든 탐색 레벨에서 공통으로 반복 사용할 수 있다.

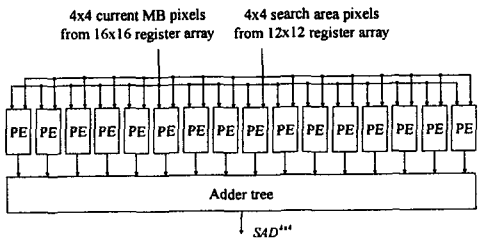

그림 4. 기본 탐색 유닛 (BSU)

BSU는 그림 4와 같이 16개의 PE (Processing

element)로 구성된 1차원 PE 어레이와 덧셈기 트리(Adder tree)로 이루어져 있다. PE는 현재 매크로블록 데이터와 탐색 영역 데이터의 차이의 절대값을 구하고 덧셈기 트리는 16개의 절대값들의 합, 즉 4×4 서브-블록 단위의 정합 기준을 구한다.

4×4 서브-블록 단위로 [-2, +2] 탐색을 4번 수행하는 것을 단위 탐색 프로세스 (Unit matching process)라고 정의한다. AMRMCS의 3개의 탐색 레벨은 단위 탐색 프로세스를 순차적으로 수행함으로써 처리할 수 있다.

III-2. 하드웨어 구조 I

하드웨어 구조 1은 4개의 BSU를 가진다. 4개의 BSU는 4개의 4×4 서브-블록에 대한 [-2, +2]의 탐색을 각각 처리한다. 그림 5는 탐색 영역 데이터를 위한 12×12 레지스터 어레이와 4개의 BSU의 연결을 나타낸다. 영역 R1에 있는 16개의 레지스터는 BSU1의 16개의 PE와 각각 연결된다. 영역 R2, R3 및 R4에 있는 16개의 레지스터들도 BSU2, BSU3 및 BS4의 16개의 PE들과 각각 연결된다.

목표 탐색 영역을 커버하기 위해서 상위 레벨에서의 탐색 영역은 수평 [-18, +17], 수직 [-9, +8]로 정하였다. 이 탐색 영역은 4×4 서브-블록에 대한 [-4, +4] 탐색을 수행하는 8개의 서브-영역으로 나누어 진다. 각각의 서브-영역에 대한 탐색을 위해서는 그림 6-(a)와 같은 12×12 탐색 영역 데이터가 필요하다. 4×4 서브-블록에 대한 [-4, +4] 탐색은 다시 4×4 서브-블록에 대한 4개의 [-2, +2] 탐색, R1, R2, R3, 및 R4로 나누어 진다. R1, R2, R3, 및 R4 영역에 대한 탐색은 각각 BSU1, BSU2, BSU3, 및 BSU4에서 병렬로 처리된다. 따라서, 상위 레벨은 단위 탐색 프로세스를 8번 순차적으로 수행하여 완료된다.

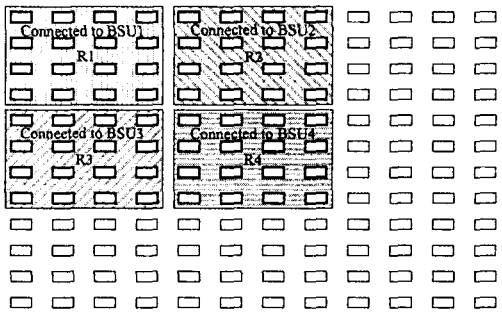

그림 5. 12×12 레지스터 어레이와 4개의 BSU의 연결

중간 레벨에서는 8×8 블록을 4개의 4×4 서브-블록으로 나누어 4개의 BSU에서 각각 4×4 서브-블록에 대한 [-2, +2] 탐색을 수행한다. 그 다음, 4개의 4×4 서브-블록의 정합 기준들을 합하여 8×8 블록의 정합 기준을 구한다. 8×8 블록에 대한 [-2, +2] 탐색을 위해서는 상위 레벨과 마찬가지로, 그림 6-(b)와 같이 12×12 탐색 영역 데이터가 필요하다. 중간 레벨에서는 3개의 후보를 가지므로, 3번의 단위 탐색 프로세스가 필요하다.

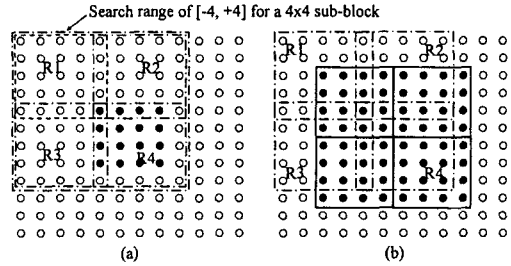

그림 6. 12×12 탐색 영역 (a) 상위, (b) 중간, 하위 레벨

하위 레벨에서는 16×16 매크로블록이 4개의 8×8 블록으로 나누어 지고, 8×8 블록은 또다시 4개의 4×4 서브-블록으로 나누어 처리된다. 즉, 중간 레벨과 마찬가지로 각각의 8×8 블록에 대한 [-2, +2] 탐색을 수행한다. 하위 레벨은 4번의 단위 탐색 프로세스가 필요하다.

그림 7은 [-2, +2] 탐색에 해당하는 25개 탐색점들의 탐색 순서와 탐색을 위한 데이터 흐름을 도시한 것이다. 매 클럭마다 4×4 현재 매크로블록 데이터와 4×4 탐색 영역 데이터가 각각 4개의 BSU로 공급된다. 클럭 0에서 4개의 BSU는 각각의 서브-블록에 대한 (-2, -2) 위치에서의 정합 기준을 계산한다. 클럭 1~4에서는 그림 7-(b)에서 보는 것처럼 12×12 탐색 영역 데이터는 좌측으로 한 픽셀씩 이동하면서 (-1, -2), (0, -2), (1, -2), 및 (2, -2) 위치에서의 정합 기준을 구한다. 클럭 5에서는 그림 7-(c)와 같이 탐색 영역 데이터를 위쪽으로 한 픽셀씩 이동시켜 (2, -1) 위치의 정합 기준을 계산한다. 클럭 6~9에서는 그림 7-(d)와 같이 탐색 영역 데이터를 우측으로 한 픽셀씩 이동시키면서 (1, -1), (0, -1), (-1, -1), 및 (-2, -1) 위치에서의 정합 기준을 순차적으로 계산한다. 이러한 방법으로 클럭 24에서 (2, 2) 위치에서의 정합 기준을 계산하게 된다.

위에서 설명한 것처럼 AMRMCS는 15 (=8+3+4)번의 단위 탐색 프로세스로 처리되며, 단위 탐색 프로세스는 4개의 BSU를 사용할 경우 25 클럭이 소요되므로, 한 매크로블록에 대한 움직임 추정을 수행하는데 375 (=15×25) 클럭이 소요된다.

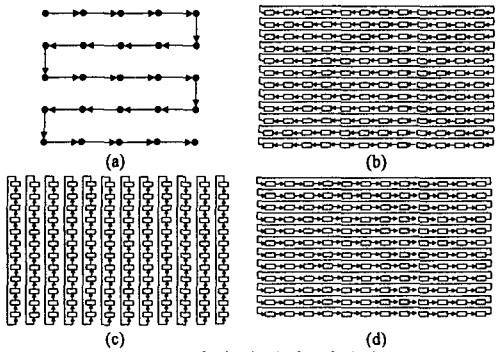

그림 7. 탐색 순서와 데이터 흐름

III-3. 하드웨어 구조 II

하드웨어 구조 II는 하드웨어 구조 I과 동일한 구조이고 20개의 BSU를 가지고 있다. 4×4 서브-블록에 대한 [-2, +2]의 탐색이 5개의 BSU에 의해 처리된다. 그림 8은 탐색 순서와 데이터의 흐름을 나타낸다. 그림 8-(b)처럼 12×12 탐색 영역 데이터가 매 클럭마다 위쪽으로 한 픽셀씩 이동하면서 5개의 BSU는 각각 (-2, n), (-1, n), (0, n), (1, n), 및 (2, n) 위치의 정합 기준을 계산한다. 따라서, 단위 탐색 프로세스는 5 클럭 소요된다. 하지만, 내부 SRAM으로부터 12×12 레지스터 어레이에 탐색 영역 데이터를 담는데 12 클럭이 소요되므로, 한 매크로블록에 대한 움직임 추정을 수행하는데 180 (=15×12) 클럭이 필요하다.

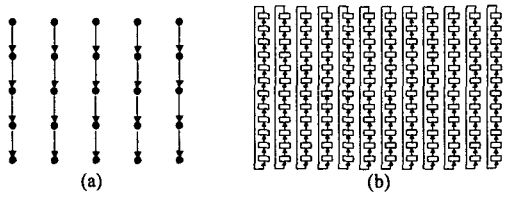

그림 8 (a) 탐색 순서, (b) 데이터 흐름

IV. 비교

표 1은 본 논문에서 제안한 움직임 추정 하드웨어 I, II의 규격을 나타낸다. 하드웨어 구조 I 및 II를 이용하여 720×480@30Hz 영상에 대한 하나의 참조 프레임을 이용한 정화소 단위 움직임 추정을 실시간으로 처리하기 위해서는 각각 16MHz, 7.7MHz의 동작 주파수가 필요하다. 동작 주파수 54MHz에서는 각각 3장, 7장의 참조 프레임을 이용한 정화소 단위 움직임 추정을 실시간으로 처리할 수 있다.

표 1. 하드웨어 구조 I, II의 규격

	구조 I	구조 II
PE 개수	64	320
입력 영상	720×480@30Hz	
탐색 영역	H: [-64, +63] V: [-32, +31]	
블록 크기	16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4	
하나의 참조 프레임 시 최소 동작 주파수	16 MHz	7.7MHz
동작 주파수 54MHz 시 최대 참조 프레임 수	3	7

V. 결론

본 논문에서는 H.264/AVC를 위한 움직임 추정 알고리즘과 하드웨어 구조를 제안하였다. 기본 탐색 유닛의 사용으로 4×4 서브-블록 단위로 정합 기준을 계산하고 이를 더 큰 블록의 정합 기준을 구하는데 이용함으로써 효율적인 하드웨어 구현이 가능하도록 하였다. 제안된 움직임 추정기는 동작 주파수가 54MHz일 때 720×480@30Hz 영상에 대해서 최대 7장의 참조 프레임에 대한 정화소 단위 움직임 추정을 실시간으로 처리할 수 있다.

VI. 향후 연구

본 논문에서 제안한 정화소 단위 움직임 추정을 위한 하드웨어 구조를 verilog를 이용하여 기술하고 FPGA를 이용하여 검증할 계획이다. 또한 H.264/AVC를 위한 부화소 단위 움직임 추정에 적합한 효율적인 하드웨어 구조에 대한 연구를 진행 중이다. 정화소 단위 움직임 추정 하드웨어와 부화소 단위 움직임 추정 하드웨어 구조가 효율적으로 결합될 수 있을 것이라고 예상한다.

참고문헌

[1] T. Wiegand, G. J. Sullivan, G. BjØntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560-576, July 2003.

[2] J. H. Lee, K. W. Lim, B. C. Song, and J. B. Ra, "A fast multi-resolution block matching algorithm and its LSI architecture for low bit-rate video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1289-1301, Dec. 2001.

[3] *Joint Video Team (JVT) software JM6.1e*