

MPEG-4를 위한 저전력 Motion Estimation 설계

최 홍 규, 이 문 기

연세대학교 전기전자공학과, 연세대학교 전기전자공학과
전화 : 02-2123-4731 / 핸드폰 : 018-228-9886

Design of Low Power Motion Estimation for MPEG-4

Hong Kyu Choi, Moon Key Lee
Dept. of EE, Yonsei University
E-mail : hkchoi@spark.yonsei.ac.kr

Abstract

The low power motion estimation for MPEG-4 is a soft-core for hardwired motion estimation block in MPEG-4. This motion estimation is modified by 10 difference mode. So, this motion estimation decrease a power consumption compare conventional step search. This modified 4SS Low power Motion Estimation has been tested and verified to be valid for implementation of FPGA. The average PSNR between the original image and the motion-compensated image is 28.25dB. And Power consumption is 26mW.

I. 서론

휴대폰과, 이동통신용 단말기의 범국민적 보급과, 여러 통신 서비스 지원의 발달, 동영상 콘텐츠의 확대는 채널 환경에 강하고, 압축률이 높은 동영상 압축 기술을 요구하고 있다. 이동통신과 더불어 방송통신에서도 적은 대역폭의 방송 서비스를 제공하여 이동 중에도 방송 서비스를 이용할 수 있도록 개발하고 있다. 뿐만 아니라, 이동 통신용 단말기를 통한 동화상 통신의 대중화도 이루어지고 있어 영상 신호의 압축 및 처리 기술과, 이를 효율적으로 실현할 수 있는 전용 고집적 회로 소자의 개발이 필요하다. 이동 통신용 단말기는 전력공급의 한계가 뚜렷하여 그 특성상 저전력 동영상

부호화기 회로의 구현이 요구된다.

현재 채널환경과 적은 데이터 양으로 각광을 받는 압축 표준이 MPEG-4이다. 움직임 추정기를 설계함에 있어, MPEG-4에서 요구하는 작은 하드웨어 크기, 저전력 구조 등을 중심으로 고려하여 설계하고자 한다.

저전력 동영상 부호화기 회로를 구현함에 있어 가장 많은 신호의 천이와, 연산수를 차지하는 핵심 모듈은 움직임 벡터 추출기이다. 이는 가장 많은 전력 소비를 하는 기능 블록이기도 하다. 이에 움직임 벡터를 추출하는 알고리즘은 여러가지 움직임 추정 알고리즘으로 제시되고 있다. 움직임 추정 알고리즘에 따라 연산의 양과 하드웨어 구조의 구성이 달라지며, 이를 통해 저전력 회로의 근거를 제시할 수 있다. 하지만 연산의 양이 줄어드는 알고리즘일 수록, 움직임 추정의 성능이 감소하므로, 알고리즘 선정과 함께 움직임 추정의 성능을 고려해야 한다. 따라서 성능과 연산 양의 관계를 적절히 고려하여 알맞은 알고리즘과 구조를 선택이 중요하다. 또한 선택의 문제를 넘어서서 적은 연산량의 저전력 회로를 구현하며 동시에 좋은 성능을 보이는 움직임 추정 알고리즘을 기존 알고리즘을 변형, 또는 제안을 통해 연구 개발해야 한다.

본 논문은 일반적으로 빠른 움직임 추정 알고리즘(Fast Motion Estimation Algorithm)에서 많이 쓰이는 삼단계 탐색(TSS)을 변형한 사단계 탐색(4SS)[1]을 이용하여, 이를 더 발전시킨 '변형된 사단계 탐색(Modified 4SS)'을 제안하고 이를 통해 성능검증을 하고자 한다.

또한 이를 HDL Simulation을 통한 동작 수준 검증과, FPGA 구현을 통해, 실제 하드웨어 구현과 전력 소비 추정(Power Estimation)을 통해 기존 움직임 추정기와 비교 평가하고자 한다.

II. 사단계 탐색을 이용한 변형된 사단계 탐색(Modified 4SS)

2.1 움직임 추정

움직임 추정은 화소 단위로 추정하는 방법과 화소를 모아 블록화시켜 블록 단위로 추정하는 방법이 있다. 화소 단위로 추정하는 방법은 그 성능은 우수할시 모르나 움직임 벡터를 계산하기 위해 너무 많은 연산이 필요하므로 블록 단위로 추정하는 방법을 주로 사용한

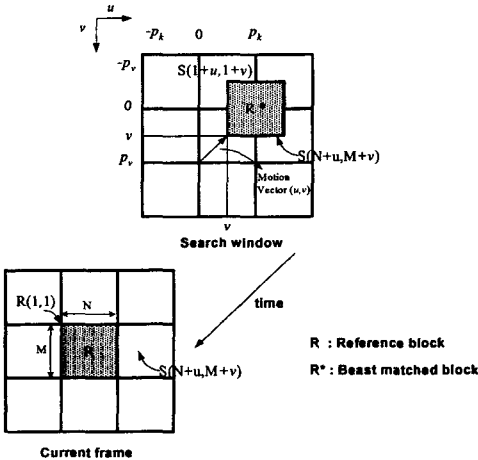


그림 1. 블록 정합 알고리즘

다. 블록 단위로 추정하는 데 있어서 이전화면에서 탐색 영역을 선택하고, 이를 현재화면의 기준블록과 비교하여 가장 유사한 위치를 찾아 움직임 벡터를 추출하는 블록 정합 움직임 추정이 사용된다.

이 과정은 기준블록과, 이전 화면 블록간의 화소값 차이를 계산하여, 그 값이 가장 작은 블록을 택하는 방식으로 진행된다. 이 차이값을 계산하는 방법이 매우 많이 소개되고 있으나, 저전력 회로 구현에 적합한 방법으로 SAD(Sum of Absolute Difference)가 주로 사용된다.[2]

$$SAD(x, y) = \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} |I_k(m, n) - I_{k-1}(m+dx, n+dy)|$$

$$(MV_x, MV_y) = \min_{(dx, dy) \in R} SAD(dx, dy) \quad (1)$$

만약 움직임 추정에서 가장 많은 연산 양을 갖는 완전 탐색(Full Search Motion Estimation) 알고리즘을 사용하고 16x16 단위 블록을 설정한 경우 적어도 2¹⁸번의 절대값 연산이 필요하다.

내용	화면크기 (Nv*Nh)	화면속도 (fr)	연산수 (Ot)
SDTV	720x576 화소	30장/초	9.6 GIPS
HDTV	1920x1152 화소	60장/초	102 GIPS

표 1. 응용에 따른 블록 정합 알고리즘의 연산 수

표1에서는 완전탐색의 경우 필요한 연산수를 보여준다. 따라서 고성능을 요구하는 움직임 추정이 아닌 경우에 완전 탐색 알고리즘을 사용하는 경우는 드물다. 대신 적은 연산 양을 갖는 빠른 움직임 추정 알고리즘(Fast Algorithm)을 택한다. 성능에서 어느 정도 손해를 보긴 하지만, 그에 비해 연산량이 현저히 줄기 때문에 빠른 알고리즘을 많이 택한다.

2.2 사단계 탐색(Four Step Search-4SS)

4단계 탐색은 빠른 움직임 추정에서 많이 쓰이는 방법이다. 4단계 탐색은 탐색 영역에서 서로 같은 거리만큼 떨어져 있는 9개의 점을 선택하여, 가장 절대값이 작은 점을 찾아 그 주위를 좀더 작은 거리 만큼 떨어진 9개의 점을 선택 같은 방법으로 4단계에 걸쳐 가장 차이값이 작은 점을 찾아 움직임 벡터를 구하는 방법이다.

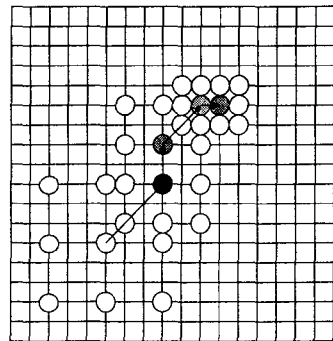


그림 2. 4단계 탐색

4단계 탐색의 경우 움직임 벡터 하나를 구하기 위해 36번의 SAD연산이 필요할 뿐이다. 이는 빠른 움직임 추정으로 대표되는 3단계 탐색에 비해 단 9번의 SAD 연산이 추가될 뿐 성능에서 득을 보기 때문에 빠른 움직임 추정에 많이 사용된다. 완전탐색과 비교해, 탐색 거리가 더 커질 수록, 4단계 탐색은 비교 연산량이 매우 감소하게 된다.

2.3 랜덤 방식을 이용한 변형된 사단계 탐색 알고리즘(Modified 4SS)

4단계 탐색 알고리즘에서 몇몇 아이디어를 사용하여 연산 양을 줄이는 방법이 시도되었다. 예를 들어 Fuzzy이론을 적용하여 이전 단계를 통해 다음 단계를 예측하고, 이에 따라 계산해야 하는 9개의 점을 다 계산하지 않고, 최소 3개 최대 9개를 계산하는 방법이 제시되기도 하였다. 하지만, 위의 방법이 실제로 Fuzzy이론을 적용한 사례라 보기에는 어려울 뿐더러, 실제로 대충 예측하는 것에 불과하기 때문에 사실 Fuzzy이론을 통해 득을 보았다기 보다는 단지 계산량을 줄이는 것뿐 그 이상의 의미는 갖지 못하였다[3]. 이에 본 논문에서는 4단계 탐색을 몇 가지 모드로 나누어, 이를 랜덤 방식을 통해 연산 양을 줄이는 방법을 시도해 본다.

우선 4단계 탐색을 10가지 모드로 나누어 보았다.

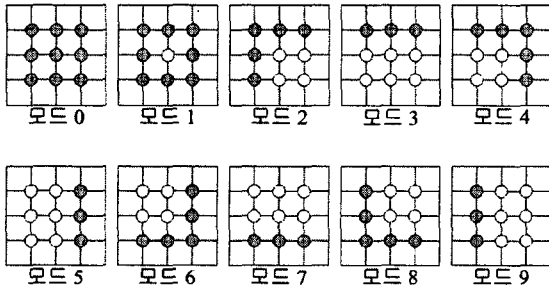


그림 3. 10개의 탐색 모드

그림 3에서 색이 칠해져 있는 부분은 연산을 하는 부분, 그리고 색이 칠해져 있지 않은 부분은 연산을 하지 않는 부분이다. 이처럼 10개의 모드에서 최대 SAD 연산을 9개를 하며, 최소 3개의 연산을 한다.

이처럼 경우를 나누어 4단계 탐색을 할 때, 총 36번의 연산을 하였던 데에서 이를 다 연산하지 않고, 적은 양의 연산을 통해 이를 4단계 탐색을 완료할 수 있다.

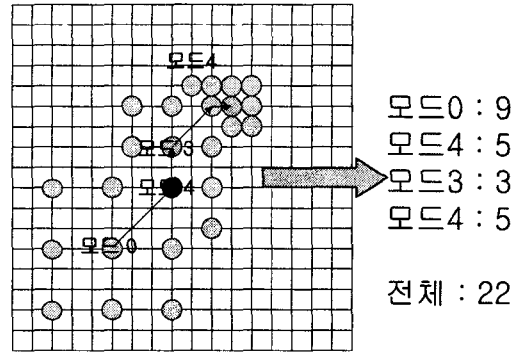


그림 4. 변형된 4단계 탐색의 연산 예

그림 4에서 보듯이 10가지 탐색 모드를 가지고 연산했을 때, 단 22번의 SAD 연산만을 통해서 움직임 벡터가 추출됨을 알 수 있다. 이는 3단계 탐색(TSS)의 27번보다도 작은 값이다.

2.4 저전력 움직임 추정기 설계

하드웨어를 구성함에 있어, 각각의 SAD연산을 수행하는 연산 요소(Processing Element-PE)를 제한한 10가지 모드를 만족시키기 위해 다음과 같이 분류하였다.

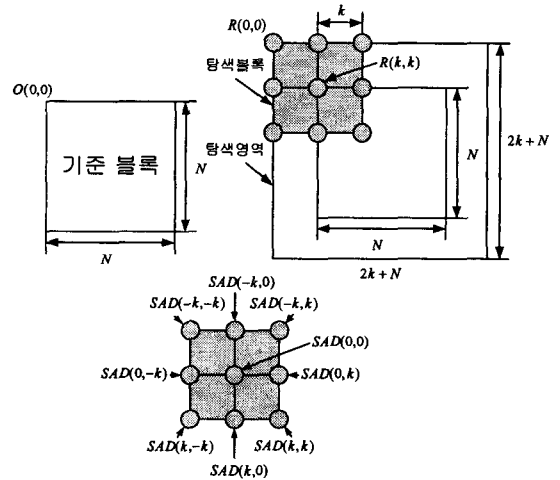


그림 5. 기준블록과 탐색영역과의 관계 및 연산 요소의 분류

그림 5에서 보듯이 기준블록이 $N \times N$ 의 크기 일 때, 탐색 영역의 한 번의 반복 연산은 $(2k + N) \times (2k + N)$ 이다.

$$SAD(-k, -k) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\alpha(i, j) - R(i, j)| \quad (2)$$

$$SAD(-k, 0) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\alpha(i, j) - R(i, j+k)| \quad (3)$$

$$SAD(-k, k) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\alpha(i, j) - R(i, j+2k)| \quad (4)$$

$$SAD(0, -k) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\alpha(i, j) - R(i+k, j)| \quad (5)$$

$$SAD(0, 0) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\alpha(i, j) - R(i+k, j+k)| \quad (6)$$

$$SAD(0, k) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\alpha(i, j) - R(i+k, j+2k)| \quad (7)$$

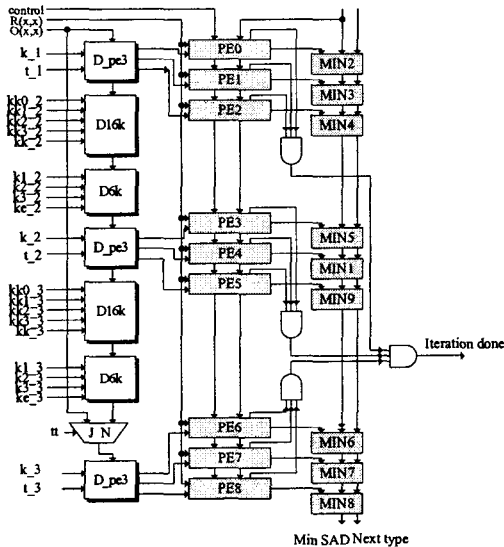
$$SAD(k, -k) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\alpha(i, j) - R(i+2k, j)| \quad (8)$$

$$SAD(k, 0) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\alpha(i, j) - R(i+2k, j+k)| \quad (9)$$

$$SAD(k, k) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\alpha(i, j) - R(i+2k, j+2k)| \quad (10)$$

(2)-(10)까지의 식은 각각의 SAD 연산 요소들의 식을 나타낸 것이다.

이를 다음과 같은 하드웨어로 구현하였다.



그리고 각각의 제어는 랜덤 발생기를 사용하여 랜덤하게 각각의 PE를 선택하도록 하였고, PE를 선택하는데 있어서, 가변 스위처 레지스터를 이용하여 제어하였다. 또한 각각의 PE블록은 사용하지 않을 때, 클럭 신호가 들어가지 않도록 설계하여 추가적인 전력소모를 막았다.

이를 HDL 시뮬레이션과 FPGA 검증을 통해 동작을 확인하였고, 이를 통해 산출된 성능은 다음과 같다.

	Akiyo	Coast Guard	Foreman	Average
완전탐색	30.621	27.127	27.959	28.569
제한한 범형된 4SS	30.284	26.828	27.651	28.254
3SS	29.936	26.631	27.714	28.094

표 2. 성능지표 PSNR(dB) 비교

전력 소비량은 Power analyzer를 이용하여 산출하였다. 산출된 전력 소비 추정치는 약 26mW이다.

V. 결론

동영상 압축에 있어 움직임 추정은 필수적인 기능 블록이다. 하지만 연산량이 방대하여 영상압축을 구현하는 이동 통신용 단말기나, 방송장비 등에 응용하기는 가장 큰 어려움을 주는 부분이다. 이를 위해 저전력 하드웨어가 필수적이다. 본 논문에서는 4단계 탐색 알고리즘을 선택하여, 연산 양이 적은 알고리즘을 택했고, 이를 변형하여 10가지 모드를 사용해 더욱 연산량을 감소 시킨 알고리즘을 제안하였다. 또한 이를 제어하는 블록을 랜덤 제어를 이용하므로 하드웨어의 복잡도도 감소시켰다. 그리고 연산요소(PE)가 사용되지 않을 때에는 클럭을 인가시키지 않아, 추가적인 전력 손실을 막았다. 성능 지표를 보았을 때, 완전탐색에 비해 많이 감소하지 않은 PSNR 수치를 얻을 수 있었고[4], 전력 소비량도 저전력 응용에 적합하였다. 본 논문은 이동 통신용 단말기와 이동 방송용 장비, 화상 회의를 위한 동화상 전송 응용의 자료로 활용 될 수 있을 것이다.

참고문헌

- [1] ISSPA 2001 "Low power motion estimation algorithm based on temporal correlation and its architecture" 2001. 9.
- [2] Peter Kuhn, "Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation" Kluwer Academic Publishers 1999
- [3] IEEE AP-ASIC Conference "A Novel Motion Estimation Algorithm and Its VLSI Architecture" 2002. 8.
- [4] S. H. Nam and M. L. Lee, "Flexible VLSI architecture of motion estimator for video image application," IEEE Trans. Circuits Syst. II, vol. 43, pp. 467-470, June 1996.