

MP@ML Half-pel을 지원하는 고성능 완전 탐색 움직임 추정기 VLSI 설계

최 홍 규, *남 승 현, 이 문 기
연세대학교 전기전자공학과, *(주)Eupuls, 연세대학교 전기전자공학과
전화 : 02-2123-4731 / 핸드폰 : 018-228-9886

Design of High Performance Full search Motion Estimation VLSI with Half-pel

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

Abstract

The block matching algorithm motion estimation is a soft-core for hardwired motion estimation block in MPEG-2, H.261 encoder.

This motion estimation has been tested and verified to be valid for implementation of FPGA. Efficiency performance of the synthesized motion estimation was up to 89%, and the average PSNR between the original image and the motion-compensated image is 38dB.

I. 서론

위성통신 시스템과, 광통신 시스템 등의 활성화로 고속 디지털 회선 사용이 확대되고, 동화상 통신의 대중화, HDTV 방송의 상용화가 이루어짐으로 인해, 영상 신호의 압축 및 처리 기술과, 이를 효율적으로 실현할 수 있는 전용 고집적 회로 소자의 개발이 요구된다. 동영상 정보의 디지털화는 엄청난 양의 정보를 전송 또는 저장하게 되고, 이를 위해 MPEG-2, H.261 등의 대역 압축을 위한 영상부호화 기법이 사용된다. 이들 응용에서 영상 정보 압축을 위해서 전 화면에서 현재 화면을 예측하는 예측부호화(predictive coding)가 사용된다.

이런 예측부호화의 근거를 제시하는 움직임 추정(Motion estimation)을 통한 움직임 벡터(Motion vector)는 여러 가지 움직임 추정 알고리즘을 통해 구할 수 있다. 이중 블록정합 알고리즘(Block Matching Algorithm, BMA)은 픽셀 단위로 움직임을 추정하는 것보다, 계산량이 적고, 성능에 대한 효율이 뛰어나 MPEG-2, H.261 등의 응용에 쓰이고 있다. 실제적으로 동영상 압축 부호화에서 부호화를 위한 전체 연산의 60%를 차지하는 핵심 모듈이 움직임 벡터 추출기이다. 움직임 벡터를 추출을 위한 여러 알고리즘들이 있지만, 이중 가장 성능이 뛰어난 것은 완전탐색 블록정합 알고리즘(Full search BMA)이다. 이 알고리즘은 계산량이 매우 많아, 삼단계 탐색, 계층적 탐색 등 다른 여러 고속 알고리즘들이 연구 개발되었다. 하지만 공정 기술 등의 발달로 성능이 뛰어난 완전탐색 알고리즘을 이용한 VLSI 설계가 용이하게 되었다. 본 논문은 많은 블록정합 알고리즘 중 성능이 가장 뛰어난 완전탐색 블록정합 움직임 추정 알고리즘을 이용하여, 가능한 고성능 움직임 추정기를 구현하였다. 또한 주변 화소의 인터폴레이션을 사용한 반 화소(Half-pel)의 개념을 도입하여 반 화소 움직임 추정을 구현하였다. 그리고 많은 계산을 실시간으로 수행하기 위해, 시스토크 어레이(systolic array)구조를 사용하여, 성능을 향상시켰다[1][2]. 이를 HDL 시뮬레이션을 통한 동작 수준 검증과, FPGA 구현을 통해, 기존 움직임 추정기와 비교 평가하고자 한다.

II. 시스토틱 어레이를 이용한 Full Search Motion Estimation

2.1 Half pixel Full Search Motion Estimation

블록 정합 움직임 추정 은 이전 화면에서 현재 화면을 예측하는 방법이다. 이는 이전화면에서 탐색 영역을 선택하고 이를 현재화면의 기준블록과 비교하여 가장 유사한 위치를 찾아 움직임 벡터를 추출함으로써 이루어진다[3].

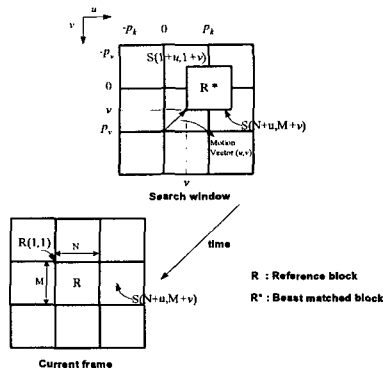


그림 1. 블록 정합 알고리즘

이 과정은 기준블록과 이전 화면 블록간의 화소값 차이를 계산하여 그 값이 가장 작은 블록을 택하는 방식으로 진행된다. 이 때 행해지는 연산은 보통 회로 구현에 있어 우수한 MAD(Mean Absolute Difference)가 사용된다.

$$MAD(u, v) = \sum_{i=1}^N \sum_{j=1}^M |R(i, j) - S(i+u, j+v)| \quad (1)$$

반 화소(Half-pixel)은 실제로 존재하는 화소는 아니고

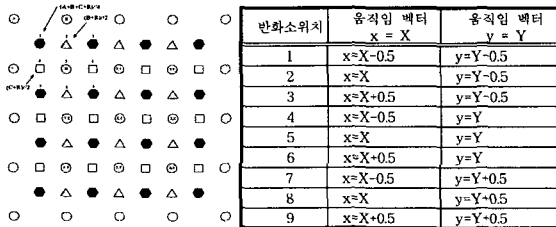


그림 2. 최소 왜곡블럭 주변의 Half pixel 형성과 Half 블록 위치에 따른 Half vector 값

$$(int \ x=X, int \ y=Y)$$

주변 화소들을 통해 보간(interpolation)으로써 얻은 화소값이다. 반 화소 예측은 환 화소 예측에 의해 보상된 화소의 질을 더욱 향상 시키기 위해 움직임 벡터의 정밀도를 ±0.5 높이는 것이다[4].

2.2 시스토틱 어레이 기법을 이용한 Full Search Motion Estimation 구현

완전탐색 블록정합 알고리즘을 이용하여 동영상 압축 부호화에 응용하고자 할 때, 응용분야에 대한 연산수는 표 1과 같다.

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

표 1. 응용에 따른 블록 정합 알고리즘의 연산수

이처럼 수백 GIPS(Giga operations per second)에 달하는 연산을 수행하기 위해서, 고속의 프로세서가 필요하다. 이는 프로세서의 병렬처리로 해결할 수 있다. 병렬로 분산 연산을 수행하므로, 프로세서의 개수만큼의 처리속도의 향상을 얻을 수 있다.

이 문제는 시스토틱 어레이(Systolic array) 기법을 사용하여, 완전탐색 블록 정합 알고리즘을 구현하여 해결할 수 있다. 시스토틱 어레이화 방법은 구현하고자 하는 알고리즘을 기본동작으로 분산시킨 후 각각의 연산 결과가 고유의 변수에 할당되는 단일할당(Single assignment principle) 원리를 적용한다 만약 그 변수가 n개의 인덱스(Index)를 갖는다면 이 알고리즘은 n차원 인덱스 공간으로 정의된다.

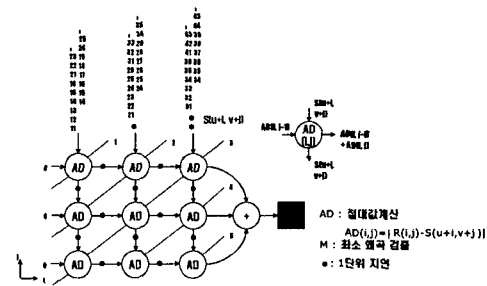


그림 5. 시스토틱 어레이를 이용한 MAD계산 신호처리 흐름도

그림 3 은 3x3 시스토틱 어레이의 예로써 3차원 종속 그래프로부터 (i, j)평면으로 투시하여 얻은 시스토틱 어레이 구조와 신호처리 흐름도를 나타낸 것이다. 이 신호처리 흐름도는 각 처리기(AD)에 각각 하나의 기준블럭 데이터 값이 필요하며 탐색영역의 데이터가 매 사이클마다 탐색영역의 상단부와 하단부로부터 두개의 데이터가 병렬 입력되며, 각 처리기가 필요한 데이터를 선택하여 병렬 및 파이프라인 동작으로 블록처리를 수행하는 구조를 나타낸다.

2.3 제안한 Full Search Motion Estimation의 영상 입력 방법

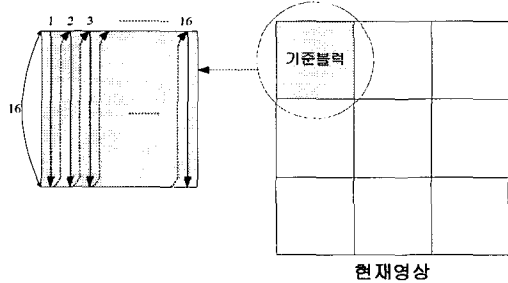


그림 6. 기준 블록 데이터 입력

시스토크 어레이로 구성된 처리기(Processing Element: PE)에 기준블록의 데이터를 입력하고, 이에 대응되는 탐색 영역의 데이터를 입력할 때, 단지 순차적으로 입력 받는다면 무효한 사이클이 발생된다.

왜냐하면 순차적으로 데이터를 입력받을 때, 이전 화면(Previous Frame)의 후보 블록의 다음행의 왜곡 연산에 필요한 해당 데이터가 준비되지 않음으로 불필요한 연산을 하게 되고, 성능저하를 가져온다. 즉 한 기준블록의 블록 왜곡 연산 과정에서 $(2p_v - 1)(N - 1)$ 사이클 동안 무효한 동작을 수행하게 된다. 예로 수직 탐색범위 p_v 가 8값을 가질 경우 무효한 사이클의 수는 225 사이클로 이상적인 사이클 수인 256 사이클에 가까운 연산 지연을 발생한다. 이는 탐색범위가 증가할수록 심각한 성능저하를 발생한다.

이를 이전화면의 입력을 상단 영역과 하단 영역으로 분리하여 입력하므로 해결할 수 있다. -8/7.5 모드에서 탐색영역 데이터 입력은 그림 5와 같다. 탐색 영역 데이터는 USW, LSW 두 입력을 통해서 입력되는데, USW는 첫 번째 행에서 17번째 행까지 LSW는 17번째 행에서 33번째 행까지 동시에 입력된다. 이 동작은 16번 반복하여 총 272사이클 동안 탐색영역 16x33(가로x세로) 부분이 입력된다. 이후 24사이클동안 쉬고, 다시 나머지 탐색영역의 16x33(가로x세로)부분이 입력된다.

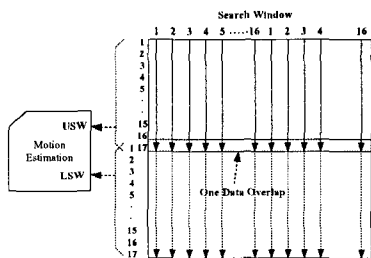


그림 5. -8/7.5 모드에서 USW와 LSW 탐색영역 데이터 입력

III. 시스토크 어레이를 이용한 Full Search Motion Estimation 설계 및 검증

Motion Estimation은 그림 6과 같이 한 화소 벡터 시스템(Int-Pixel Vector System)과 반 화소 벡터 시스템(Half-Pixel Vector System)으로 분류된다. 한 화소 벡터 시스템은 PE 어레이, 시프트 레지스터 (SRA: Shift Register Array), 병렬 덧셈기(Parallel Adder), 최소 블록 왜곡 추출기(MDD: Minimum Distortion Detector), 탐색 데이터 선택기(Data Selector), 로컬 메모리 (Local memory), 주소발생기(Address generator) 및 제어기로 구성되어 있다[5].

기준블록 데이터는 REF핀을 통하여 PE 어레이에 입력되고, 탐색영역 데이터는 USW, LSW핀을 통하여 PE어레이와 SRA에 입력된다. 이들 데이터는 PE 어레이, 병렬 덧셈기에서 블록 처리되어 블록 왜곡(MAD: Minimum Absolute Difference)을 계산하고, 블록 왜곡 중 최소의 블록왜곡값을 검출하기 위해 최소 블록 왜곡 추출기에 저장되어 있는 현재까지의 최소 블록왜곡값과 비교된다.

시프트 레지스터 (SRA: Shift Register Array)는 연속적으로 PE어레이에 탐색 영역 데이터를 공급하여 dead cycle의 발생을 막기 위함이다. 블록처리 과정에서 블록왜곡 연산을 지연없이 처리하도록 하기위해 USW와 LSW 탐색 데이터 가운데 유효 데이터를 선택하여야 하며 이는 탐색 데이터 선택기의 제어신호에 의해 결정된다. 로컬 메모리는 연속된 기준블록 처리 시 중복되는 탐색영역 데이터를 저장한다. 버퍼는 로컬 메모리 출력을 16 cycle 지연시키기 위함이다. 반 화소 벡터 시스템은 최소 블록 왜곡 한 화소 추출기 (Best Matched Int-Pixel Extractor), 반화소 발생기 (Half Pixel Generator), 반화소 블록 왜곡 연산기 (Half Pixel Distortion Calculator), 탐색데이터 선택기 (Data Selector), 기준 블록 세이버 (Reference Block Saver), 최소블록 왜곡 추출기(MDD: Minimum Distortion Detector)로 구성되어 있다.

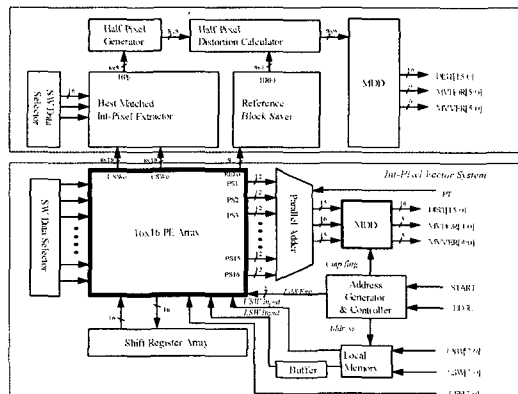
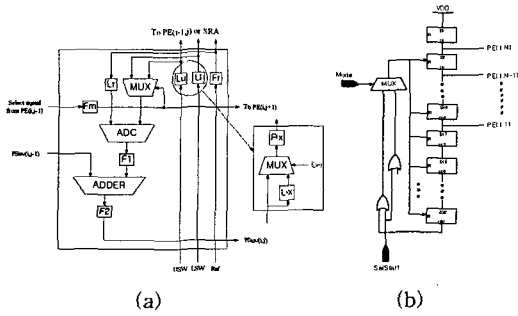
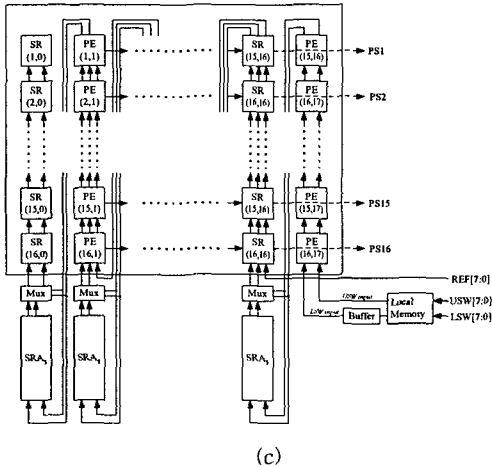


그림 6. 움직임 추정기의 전체 구조



(a) (b)



(c)

그림 7. (a)PE(Processing Element), (b)탐색 데이터 선택기, (c) 16x16 PE& 시프트 레지스터 어레이

최소 블록 왜곡 한 화소 추출기는 반화소를 발생하는데 필요한 최소블록 왜곡 데이터와 주변의 18x18 int pixel을 저장한다. 탐색 데이터 선택기는 최소 블록 왜곡 한 화소 추출기의 탐색 데이터의 흐름을 제어한다.

반화소 발생기는 최소 블록 왜곡 한 화소 추출기의 탐색 데이터의 흐름을 제어한다. 반화소 발생기는 최소 블록 왜곡 한 화소 추출기의 출력인 int-pixel을 입력으로 하여 반화소를 발생한다. 기준블럭 새이버는 반화소 블록왜곡을 계산하는데 필요한 기준블럭데이터를 저장한다. 반화소 블록 왜곡 연산기는 기준블럭데이터와 반화소 데이터를 입력으로 8개의 반화소 블록왜곡을 계산한다. 최소 블록 왜곡 추출기는 8개의 반화소 블록 왜곡을 계산한다. 최소 블록 왜곡 추출기는 8개의 반화소 블록 왜곡과 한 화소 블록 왜곡을 비교하여, 최소의 블록왜곡과 반화소 움직임 벡터를 발생한다.

이에 대한 HDL 시뮬레이션 결과 파형과, FPGA 라우팅 사진은 각각 그림 8, 그림 9와 같다. 결과적으로 약 100,000 Gate의 크기와, 10MHz의 동작주파수에서 동작하는 움직임 추정을 구현할 수 있었다.

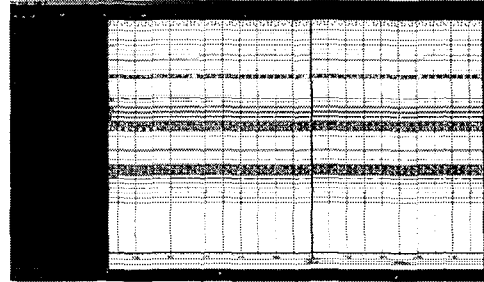


그림 8. HDL Simulation 결과 파형



그림9. FPGA 이용한 검증

IV. 결론

움직임 추정은 동영상 압축을 하기 위한 필수적인 기능 블록이다. 하지만 연산량이 매우 방대하여, 이를 실시간으로 수행하기 위해서는 병렬적인 연산처리가 가능해야 한다. 본 논문에서는 이를 위해 시스톱릭 어레이를 이용하여 병렬 처리가 가능하도록 하였고, 실시간으로 동작함을 확인할 수 있었다. 움직임 추정을 FPGA로 구현하여 영상 신호를 인가하고, 움직임 벡터가 추출됨을 확인하였다. 또한 PSNR 값이 평균 38dB로 매우 높은 수치를 기록하였다. 본 논문은 현재 HDTV와 화상통신 응용에 사용되는 고성능 움직임 추정기 응용의 자료로 활용 될 수 있을 것이다.

참고문헌

- [1] S.H. Nam, T.Y. Lee, J.S. Baek and M. K. Lee "A VLSI Design for Full Search Block Matching Motion Estimation", Proc. IEEE International ASIC Conference, Rochester NY, pp254-257, Sep., 1994
- [2] S.H. Nam, J.S. Baek and M. K. Lee, G. S. Hwang and C. D. Oh, "A Fast Array Architecture for Block Matching Algorithm", Proc. IEEE International symposium on circuits and system, London, pp.211-214, May 1994
- [3] T.Koga et al, "Motion compensated interframe coding for video conferencing," Proc. Nat. Telecom Conf., New Orleans, pp.G5.3.1-5.3.5, Nov 1981
- [4] Uramoto et al., "A half-pel precision motion estimation processor for NTSC-resolution video," proc. IEEE Custom Integ. Circ. Conf., pp.11.2.1-11.2.4, 1993
- [5] 백종섭, 남승현, 이문기, "완전탐색에 의한 움직임 추정기 시스톱릭 어레이 설계", 대한전자공학회, 제 31권, B편, 제12호, pp27-34,1994