

ADSP-21020을 이용한 Motion Estimation의 구현에 관한 연구

김 상기, 김 재영*, 변 재용*, 정 진현*
*광운 대학교 제어 계측 공학과

A study of Implementation of Motion Estimation
with ADSP-21020

Sang Ki Kim, Jae Young Kim*, Chae Ung Byun*, Chin Hyun Chung*
*Dept. of Control & Instrumentation Engineering, Kwangwoon Univ.

ABSTRACT

In this paper, a motion estimation module is made with ADSP-21020 based on MPEG-2 which is an international standard for moving picture compression. And, the block matching algorithm used as motion estimation method is easy for an hardware implementation. The ADSP-21020 of Analog Device is used for a main control processor. We used three block matching method (exhaustive search method, 2D-logarithmic search method, three step search method) for software simulation and implemented the three step search method to hardware. For the test of the estimation module, we used ping pong image sequences and mobile and calendar image sequences.

1. 서론

멀티 미디어의 발전에 따라 정보의 디지털화가 이루어지고 있다. 비디오 분야도 점차 디지털 방식으로 변화를 꾀하고 있다. 그러나, 비디오 데이터는 그 량이 너무 방대하기 때문에 처리나 전송에 있어서 많은 어려움이 있다. 따라서 이러한 비디오 데이터의 압축은 없어서는 안될 부분이 되었다. 동화상 압축에 사용되는 압축의 기법으로는 프레임내 압축(intraframe compression)과 프레임간 압축(interframe compression)으로 나누어진다. 프레임간의 압축은 parametric model과 nonparametric model로 나누어지고, nonparametric model은 다시 optical flow method, block matching method, pel recursive method 등이 있다. BMA는 영상을 단위 블록으로 나누어 블록 내의 pixel은 모두 동일한 움직임을 가지고 있다는 가정 하에 단순화시켰기 때문에 PRA(pel-recursive algorithm)에 비교해서 수식적으로 간단하고 계산이 빠르다. 동화상의 경우 초당 30 프레임으로 화상이 바뀌고 실시간 압축의 경우 매우 빠른 계산을 요구하기 때문에 이 BMA 방법은 현재

MPEG 영상 압축의 표준이 되어 있다. BMA중에서 제안되어 왔던 알고리즘은 여러 가지가 있지만 그 중 exhaustive search method, 2D-logarithmic search method, three step search method가 대표적인 방법이다.

2. BMA (Block Matching Algorithm)

BMA란 픽셀을 블록 단위로 나누고 한 블록에 대해서 다음 프레임을 검색하여 가장 큰 유사성을 가진 블록을 찾아 그 블록의 기준 점의 이동을 모션벡터(motion vector)로 정의하여 움직임을 추정하는 방법을 말한다. 이때 영역에서의 블록 유사성을 검사하기 위한 방법으로 다음의 식 (1)이나 (2)가 이용된다.

Mean Square Error (MSE)

$$M_1(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [U(m, n) - U_R(m+i, n+j)]^2 \quad -p \leq i, j \leq p \quad (1)$$

Mean Absolute Error (MAE)

$$M_2(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |U(m, n) - U_R(m+i, n+j)| \quad -p \leq i, j \leq p \quad (2)$$

여기서 $U(m, n)$ 은 현재 블록의 픽셀값이고 $U_R(m, n)$ 은 이전 프레임의 비교 대상 블록의 픽셀값이다. 위의 두 가지 식 중에 어느 것을 써도 무방하나 계산상의 이점을 고려하여 MAE를 이용하였다. 블록의 유사성을 검색하기

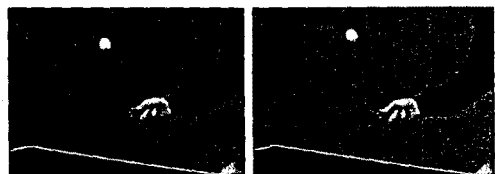


그림 1. 이전 프레임

그림 2. 현재 프레임

위한 방법으로 여러 가지가 있으나 본 논문에서는 exhaustive search method, 2D-logarithmic search method, three step search method에 대하여 실험하였다. 다음 그림1과 그림2는 본 논문에서 사용할 low-level의 연속된 2개의 프레임이다.

3. Exhaustive search method

완전 검색 방법은 현재 프레임에 $(M \times N)$ 크기의 단위 블록이 정해져 있을 때 이전 프레임에 대하여 가장 유사한 블록을 찾아가는 방식이다. 때문에 흔히 full search 방법으로 말하기도 한다. 여기에서 급격한 이동이 발생하지 않는 한 전체화면을 검색하는 것은 불필요한 계산을 가중시키는 것이므로 일반적인 적용에는 검색 영역을 상하 좌우로 도트 픽셀 p 만큼 정한다. 따라서 $(M \times N)$ 블록의 끝을 기준으로 p 만큼 정해진 영역, 즉 $(M+2p+1) \times (N+2p+1)$ 크기의 검색영역이 정해진다. 실제적으로 $(M+2p+1) \times (N+2p+1)$ 의 영역을 $(M \times N)$ 의 블록으로 비교해 가는 것 자체도 많은 계산량을 요구한다. 이에 따라 프레임내에서의 반복되는 무늬 등의 급격한 변화가 없다는 가정 하에 동일 블록을 찾아가는 것이 2D-logarithmic 검색 방법이며 여기서 보다 효율적으로 변형된 TSSP 등이 있다.

4. 2D logarithmic search method

2D logarithmic 검색 방법은 현재 위치를 기준으로 원점, 상하 ± 2 , 좌우 ± 2 의 변위의 5점을 비교하여 최소의 MAE를 가지는 점으로 원점을 이동한 후, 다시 ± 2 변위로 그 점과 주변 점을 검색, 비교하여 최소의 MAE를 가지는 점으로 이동하여 간다. 만일 중앙점이 최소의 오차값을 가지게 되면 그 점을 중심으로 변위 ± 1 의 주변 8개의 점을 검색, 비교하여 최종적으로 오차 값이 작은 점을 결과로 낸다. 여기서 계산 과정이 전 영역에 대해 일어날 수 있으므로 계산 횟수를 제한하게 되는데, 영역을 제한하거나 블록의 이동 횟수를 제한할 수 있다.

5. 3 step search method

TBSP(Three-step search procedure)는 중심점을 기준으로 8방향으로 블록의 MAE를 계산하고 비교하여 최적의 블록을 찾은 후 다시 그 지점을 중점으로 다시 세부적으로 8방향 비교를 하여 나간다. 첫 번째 블록의 변위는 4로 하였고 두 번째 블록에서는 변위를 2로 하고 마지막은 1로 하였다. 전체의 변위가 7로 한정되어 있어 빠르게 움직이는 물체는 추정하기 어렵지만 앞의 두 가지 방법에 비해 25번의 MSE 계산만이 요구되므로 아주 빠른 검색 시간을 가진다.

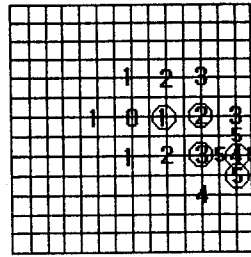


그림 3. 2D-logarithmic 검색 과정

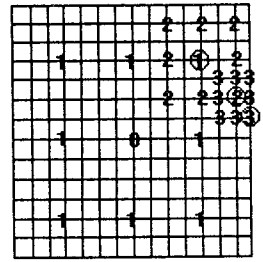


그림 4. TSSP의 검색 과정

5. 시스템의 구성

ADSP-21020은 33MHz의 클럭으로 동작하는 harvard architecture로 구성되어 있다. 평풍 이미지가 360×240 픽셀의 크기의 흑백 이미지이므로 최소 2개의 이미지를 저장하려면 86400×2 byte의 메모리가 필요하다. 따라서 메인 보드는 데이터 메모리는 $512K \times 32$ bit이고 프로그램 메모리는 $96K \times 48$ bit로 큰 메모리를 확보하였다. 그림 5는 전체 시스템의 대략적인 구성도이다.

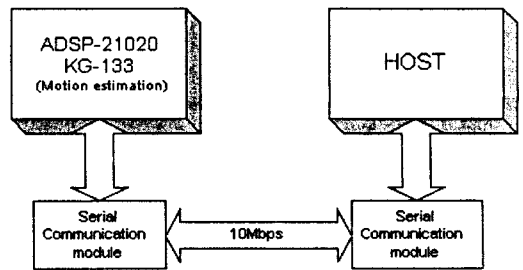


그림 5. 시스템 구성도

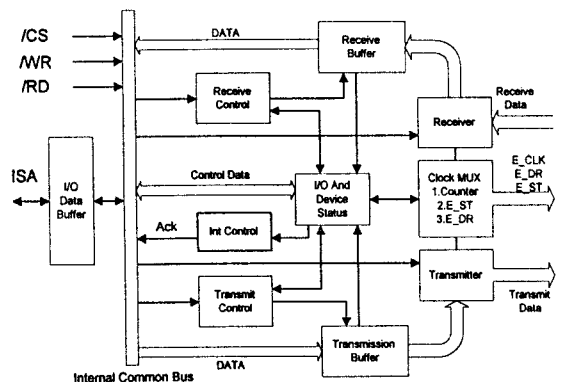


그림 6. 직렬 통신 모듈의 블록 선도

Motion estimation은 프레임 간의 압축이므로 DSP보드에서는 처리하기 위한 두 개의 프레임이 필요하다. 본 논문에서는 이 프레임들을 호스트(PC)로부터 전송 받아 사용하였다. 전송 시에는 고속의 전송을 위하여 ALTERA

의 EPLD를 이용하여 직렬 통신용 모듈을 제작하여 프레임 임을 받아 움직임 추정을 하였다. 그리고 motion vector를 다시 호스트로 전송하여 이미지를 복원한다. 그림6은 직렬 통신 모듈의 블록 선도로 나타낸 그림이다.

위의 직렬 통신 모듈은 8비트 병렬 데이터를 풀립플롭을 이용하여 직렬 데이터로 변환하여 전송하고, 직렬로 전송된 데이터 마찬가지로 병렬로 변환하여 전달해 주는 모듈이다. 이 모듈은 시간의 지연을 막기위해 MPU는 사용하지 않고 하드웨어로만 구성이 되었으며 ADSP-21020과 호스트의 데이터 전송에 사용되므로 2개가 필요하다.

6. 결 과

그림7 부터 그림12는 BMA의 exhaustive search method, 2D-logarithmic search method, three step search method 3가지 방법에 의한 motion vector와 복원된 이미지이다. 그림을 비교하면 exhaustive search method가 가장 잘 motion vector를 찾았지만 시간이 가장 오래 걸렸다. 물론 three step search method가 검색 시간이 가장 빨랐다. PSNR값은 exhaustive search method가 30.95dB, 2D-logarithmic search method가 28.54dB, three step search method가 29.37dB로 나왔다.

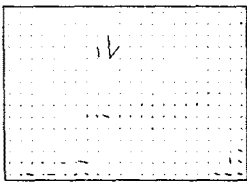


그림 7. Exhaustive search method



그림 8. Exhaustive search method

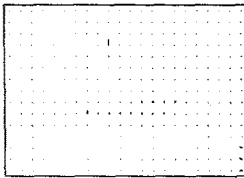


그림 9. 2D-logarithmic search method

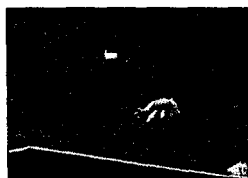


그림 10. 2D-logarithmic search method

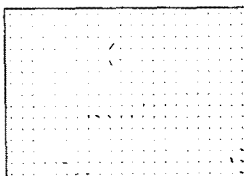


그림 11. Three step search method



그림 12. Three step search method

7. 추후 과제

본 논문의 움직임 추정 과정은 MPEG-II의 규정에 의해 구현되어 이 모듈을 더 발전시켜 MPEG 부호기의 제작이 가능하다. 물론 다른 여러 부분들이 보완되어야 할 것이다. 우선은 카메라로부터 직접 이미지를 얻어야 할 것이고, 이 이미지를 저장하기 위한 저장매체가 필요하다. 비록 ADSP-21020이 33MHz의 빠른 처리 속도를 가지지만, 이 보다 빠르게 MAE를 계산하는 과정을 ASIC과 같은 하드웨어로 구성하면 실시간 처리도 가능할 것이다.

참고 문헌

1. A. Murat Tekalp, "Digital video processing", Prentice Hall.
2. Anil K. Jain, "Fundamentals of digital image processing", Prentice Hall, 1989.
3. Rafael C. Gonzalez and Richard E. Wood, "Digital Image Processing," Addison Wesley, 1992.
4. Frederic Dufaux and Fabrice Moscheni, "Motion Estimation Techniques for Digital TV: A Review and a New Contribution", proc. IEEE, VOL. 83, NO. 6, JUNE 1995.
5. 정 제창 번역, "그림으로 보는 최신 MPEG", 교보문고, 1995.
6. 이 범로, "이미지 코딩을 위한 수정 퍼지 신경 회로 망에 관한 연구", 1994.
7. 이 병철, "동화상 압축을 위한 퍼지 움직임 추정에 관한 연구", 1995.
8. 김 재영, 이 병철, 이 기서, 정 진현, "ADSP-21020을 이용한 MPEG-II 오디오 인코더의 구현", 대한전기학회 학회논문집, 1995.
9. K.R.Rao and P.Yip, "Discrete Cosine Transform", Academic Press, 1990.
10. Phillip E. Mattison, "Practical Digital Video with Programming Examples in C", Jhon Wiley & Sons, Inc, 1994.
11. Herbert Schildt, "Teach Yourself C++", 2nd ed, Osborne McGrawHill, 1995.
12. "ADSP-21020/21010 User's Manual", Analog Device, 1993.