가변 탐색범위를 적용한 움직임 예측 모듈 구현

최덕영* · 손승일*

*한신대학교

Implementation of Motion Estimation Module with Variable Search Range

Dug-Young Choi* · Seung-Il Sonh**

*Hanshin University

E-mail : coolduck@daum.net

요 약

현재 상용화 되고 있는 DMB에서는 176*144의 작은 영상 사이즈를 표준으로 서비스하고 있다. 뿐만 아니라 서비스 되고 있는 콘텐츠들은 주로 움직임이 많은 영화나 스포츠 그리고 드라마 등이 주류를 이루고 있다. 따라서 시간적 압축 방식을 사용하는 움직임 예측 모듈이 더욱더 중요한 위치를 차지하게 됐으며 기존의 영상 표준안과 다르게 4*4와 같은 작은 블록 사이즈가 중요한 정보를 갖게 되었다.

본 논문은 DMB에서 서비스 하는 여러 가지 영화나 스포츠를 대상으로 실험한 결과 4*4와 같은 작은 사이즈의 블록이 움직임 예측시 많이 나타날 뿐 아니라 중요한 정보들로 이루어져 있다는 결과를 얻었으며 이를 토대로 좀 더 정확한 움직임 예측을 수행하기 위하여 가변 탐색범위를 제안하였다. 제안된 방법은 C언어를 통하여 검증하였으며 그 결과 고정의 탐색범위를 적용한 것보다 좋은 효율을 얻었다. 그리고 이를 다시 하드웨어 언어인 VHDL로 구현하였다.

키워드

H.264, 움직임 예측, 가변 탐색범위, VHDL

Ⅰ. 서 론

현재 압축표준안인 H.264에서는 움직임 예측 시 7개의 블록 모드를 이용하여 움직임 예측을 시행한다. 이때 탐색범위는 블록의 1배 이상부터 1/2배 크기만큼의 탐색범위를 갖는다. 하지만 탐색범위가 클수록 좀 더 정확한 움직임 벡터를 얻을 수 있으나 그만큼 많은 연산 량을 가지게 되는 단점이 있다. 반면 탐색범위가 작을수록 연산량은 적으나 정확한 움직임 벡터를 찾지 못한다. 이러한 장단점으로 인하여 영상과 상관없이 연산량 또는 정확한 움직임 벡터 추출 중 한쪽에 더 비중을 두고 어느 정도의 손실을 허용하거나 연산 량을 허용한다. 하지만 이러한 손실을 허용하더라도 보편적으로 사용하는 탐색범위가 블록의 1배인 4*4 블록에서는 효과적인 결과 값을 얻을 수 없다. 즉 4*4 블록일 경우 영상내의 움직임이 커서 가장 정확한 벡터 값이 탐색범위 외에 있게 되어 8*8 블록으로 움직임 예측 한 경우보다 좋지 않은 움직임 값을 갖게 된다. 이러한 경우는 영화나 스포츠 경기 같은 경우 더욱더 많이 나타나고 있다.

이에 본 논문에서는 현재 상용화 되고 있는 DMB 서비스의 콘텐츠가 주로 영화나 스포츠, 드라마라는 것에 중점을 두어 앞에서 말한 현상이 빈번히 나타난다는 것을 시뮬레이션을 통해 알아보았으며, 이러한 문제점을 보안하기 위하여 가변 탐색범위를 적용, 이전 영상과 현재 영상의 MAD를 이용하여 어떠한 탐색범위를 적용할 것인지 결정한다. 따라서 결정된 모드에 의하여 6*6 또는 4*4의 가변적인 탐색범위를 적용할 경우 일정한 탐색범위를 적용한 4*4블록 보다 더 효과적인 움직임 예측을 할 수 있다[1].

Ⅱ. 본 론

2.1 H.264

현재 DMB의 표준안으로는 H.264의 Baseline profile이 표준으로 채택되어 있으며, 영상 사이즈는 176*144의 영상을 사용하고 있다. H.264에서는 움직임 예측을 위하여 16*16, 16*8, 8*16, 8*8, 8*4,

4*8, 4*4와 같이 7개의 블록과 서브 블록을 가지고 움직임 예측을 수행하고 있으나 인코더에서는 각 회사마다 정해진 블록만을 선택하여 움직임 예측을 수행하고 있다. 이와 같이 7개의 모드중 일부만을 선택하는 것은 효율이 좋은 것만 선택하여 보냄으로써 여러 가지 이점을 얻을 수 있기 때문이다. 하지만 예측을 위한 블록을 아무리 여러 가지 제시한다고 하더라도 탐색범위에 따라서 좋은 효율을 얻을 수도 있고 그렇지 않을 수도 있다. 움직임 예측시 탐색범위는 블록의 1배 이상부터 1/2배의 크기만큼의 탐색범위를 정하는데 이때 탐색범위가 크면 연산 량이 많아지고 탐색범위가 작으면 오차가 커지게 된다. 따라서 이러한 장단점에도 불구하고 탐색범위를 고정하여 움직임 예측을 수행하는데 여기에 한가지의 문제점이 있다. 보여지는 영상 내에서도 움직임이 빠른 부분과 그렇지 못한 곳이 동시에 존재하기 때문이다. 영화를 보더라도 처음부터 끝까지 빠른 움직임만이 존재하지는 않는다는 것이다. 뿐만 아니라 드라마나 뉴스 같은 경우는 더더욱 이러한 현상이 많이 나타난다[2].

따라서 아무리 빠른 탐색 알고리즘이 연구되어지고 있다고 하더라도 영화나 스포츠 경기와 같은 탐색범위를 드라마나 뉴스에 적용시키는 것은 필요 없는 연산을 더 하게 되는 것이다[3].

2.2 가변 탐색범위를 적용한 움직임 예측

위에서 제시된 문제점을 보안하고자 본 논문에서는 176*144의 크기를 가지는 영화, 스포츠, 뉴스 이 3가지의 영상을 가지고 탐색범위 1.5배와 1배를 적용 시켰으며 블록 사이즈는 16*16, 8*8, 4*4의 크기를 적용시켜 실험해 보았다. 표1은 실험을 통해 얻은 움직임 예측시 오차 값이다.

표1. 탐색범위에 따른 오차값

영상의 종류	탐색범위	오차값
영화	1.5	299249
	1	339604
스포츠	1.5	120612
	1	135303
뉴스	1.5	98911
	1	108318

표1에서 나타난 것과 같이 영화나 스포츠와 같이 움직임이 많은 영상은 탐색범위에 따라 오차 값의 차이가 많이 나타났으며 뉴스와 같이 움직임이 적은 영상에서는 오차 값의 차이가 많지 않았다. 뿐만 아니라 움직임 예측에 적용된 블록 사이즈 빈도수를 본 결과 움직임이 많은 영상에서는 4*4의 작은 블록이 많이 사용되었고 또한 중요한 영상 정보를 가지고 있다는 것을 실험을 통해 확인 하였다[4].

표2. 탐색범위에 따른 4*4 블록의 사용빈도

영상의 종류	탐색범위	4*4 블록의 사용빈도
영화	1.5	548
	1	300
스포츠	1.5	336
	1	300
뉴스	1.5	264
	1	240

뿐만 아니라 16*16과 같이 큰 블록은 탐색범위가 블록의 1.5배나 1배일 때 오차의 차이가 크지 않다는 것을 확인하였다. 따라서 일정한 임계치 값을 설정하여 16*16, 8*8, 4*4 블록의 탐색영역을 각각 다르게 결정하면 불필요한 연산을 줄일 수 있으며 더 좋은 움직임 벡터 값을 얻을 수 있다.

이에 본 실험에서 나온 결과를 고찰한 결과 16*16 블록은 탐색범위를 1/2인 8로 실험하였을 경우 가장 좋은 효율을 얻을 수 있었고, 8*8은 탐색범위를 1배인 8로 적용 하였을 때 좋은 효율을 얻었다. 그러나 4*4 블록일 경우 탐색범위를 1.5배인 6으로 하였을 때 좋은 결과를 얻기는 하였지만 이로 인하여 엄청난 연산 량을 요구하게 되었다.

표3에서 보는 것과 같이 탐색범위를 4로 적용할 때 보다 2배 이상의 연산 횟수를 요구하고 있으며 이는 16*16, 8*8과 같은 크기의 블록으로 계산하였을 때 엄청난 연산 량으로 인하여 그 효율이 좋지 않음을 알 수 있다. 따라서 가변 탐색범위를 적용하기 위하여 연산 량 대 오차 값의 크기를 비교, 가장 적당하다고 여기는 탐색범위 4를 적용시킨 후 움직임 벡터와 오차 값을 얻고 여기서 얻어진 오차 값이 임계치보다 크게 되면 얻어진 벡터 값을 중심으로 탐색범위 2를 적용 시켜, 다시 움직임 예측을 시행하게 된다. 이는 움직임 큰 영상일 경우 그 움직임이 4*4의 탐색범위를 벗어났다고 가정하여 탐색범위를 벗어난 움직임 벡터를 더 정확하게 찾을 수 있다.

표3. 4*4 블록의 탐색범위에 따른 연산횟수와 오차 값

탐색범위	연산회수	오차값
8 적용	289	196222
6 적용	169	197823
4 적용	81	224637
4_2 적용	107	212222
3_3 적용	98	216868
4_1 적용	85	222308

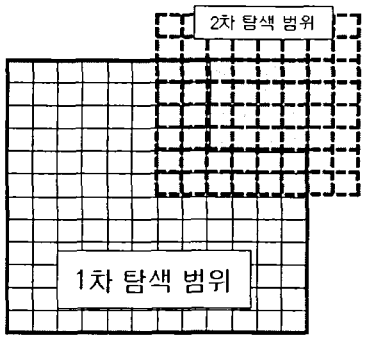

그림 1. 가변 탐색범위를 적용한 움직임 예측

III. 제안된 모듈의 설계

3.1 전체 블록도

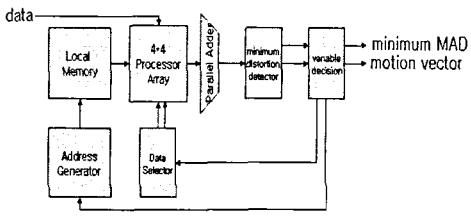

그림 2. 전체 블록도

그림 2는 제안한 모듈의 전체 블록 도를 보여주고 있다. 처음 초기화되고 현재 프레임의 블록 데이터가 들어오면 4*4 프로세스 어레이에서는 시스토릭 어레이 형식으로 4*4블록의 행의 MAD(Mean Absolute Difference)을 구하여 병렬 덧셈기(Parallel Adder)로 보내어 최종 4*4블록의 MAD를 구한다. 구해진 MAD는 최소 블록왜곡 검출기로 보내지어 최소 MAD값과 움직임 벡터 값을 얻어내고 얻어진 최소 MAD는 가변 탐색범위 결정 모듈에서 임계치 값에 의하여 가변 탐색범위를 적용할 것인지 결정하게 된다. 여기서 가변 탐색범위를 적용하게 되면 가변 탐색범위 결정 모듈은 주소 발생기(Address Generator)에 x, y에 대한 움직임 벡터 값을 보내주고 데이터 선택기(Data Selector)에 가변 탐색범위 결정 신호를 보내준다. x, y에 대하여 벡터 값을 받은 주소 발생기는 벡터 값을 기준으로 하여 탐색범위를 2로 하여 다시 최적의 움직임 벡터를 찾게 된다[5].

3.2 4*4 배열 처리 모듈 블록도

그림 3은 4*4 배열 처리 모듈의 블록도이다. 제안된 방법은 똑같은 모듈을 그래도 사용함으로써 칩으로 제작할 경우 기존의 방식보다 많은 면적을 차지하지 않는다는 것이다. 즉 1차 탐색 때는 MUX에서 데이터들을 그냥 넘겨주면 탐색범위가 4인 움직임 예측을 수행 할 수 있다. 하지만 가변 탐색범위가 선택되면 맨 아래 있는 레지스터를 통과하지 않고 바로 위 단계의 레지스터로 데이터를 전송함으로써 탐색범위 2인 2차 움직임 예측을 수행하게 된다.

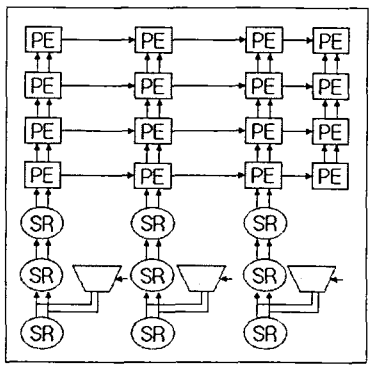

그림 3. 4*4배열 처리 모듈 블록도

IV. 결 과

4.1 PSNR

그림 4는 소프트웨어를 통하여 검증한 PSNR이다. 실험에 사용한 영상은 영화의 한 부분을 프레임 별로 나누어 움직임 예측을 수행하였고 여기서 얻은 최소 MAD를 이용하였다. 범위 4_2 적용은 본 논문에서 제안한 가변 탐색범위를 적용하여 얻은 값이다. 그림에서와 같이 탐색범위 6을 적용한 값에 가까이 나오는 것은 영상 전체적으로 움직임이 많기 때문이다[6].

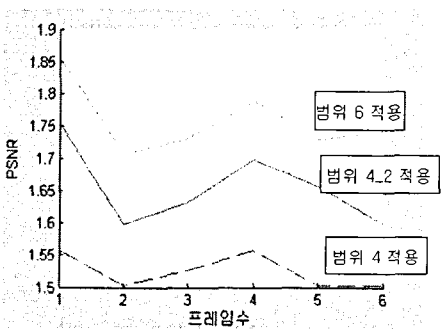

그림 4. 제안된 방법을 적용한 PSNR

4.2 출력 파형

그림 5와 그림 6은 설계된 모듈에서 나온 출력 파형이다. 현재 프레임의 데이터 값이 입력되면 처음 25클럭 동안 초기 세팅을 하고 첫 MAD의

값은 다시 4클럭 후에 나오게 된다. 이후 81번의 연산을 수행한 후 일정 임계치 값보다 크게 되면 다시 2차 탐색을 하는데 이때는 첫 19클럭 동안 초기화 한 후 다시 4클럭 후에 MAD값이 나온다.

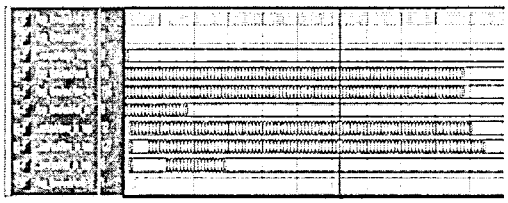

그림 5 데이터 입력 파형

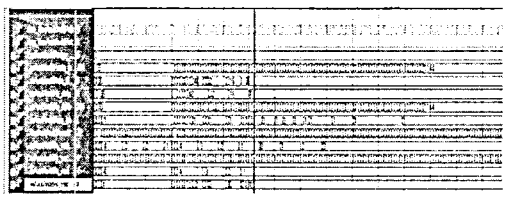

그림 6 움직임 벡터 출력 파형

V. 결 론

본 논문에서는 빠른 탐색 영역을 찾는 방법이 아닌 H.264에서 수행하는 4*4 블록의 움직임 예측의 특성에 대하여 실험 및 연구하였으며 이를 통하여 움직임 예측시 4*4 블록이 많이 나타나고 또한 4*4 블록에는 영상의 중요한 정보가 들어 있다는 것을 보여주었다. 따라서 4*4 블록을 어떻게 효과적으로 처리 할 수 있는가가 현재 176*144의 작은 영상 사이즈로 서비스 하는 DMB 서비스에서 효율을 높이는데 중요한 관점이 된다. 이에 무조건적인 탐색범위를 늘리는 것보다는 임계치 값을 사용하여 조건적으로 탐색범위를 늘려감으로써 불필요한 연산을 줄였으며 뿐만 아니라 더 좋은 효율을 얻을 수 있었다. 따라서 앞으로 빠른 움직임 예측 알고리즘을 적용 시키고 가변 탐색범위를 좀 더 유연하게 만들 수 있다면 현재보다 20%이상 효과적으로 압축 할 수 있을 것이다.

참고문헌

[1] Lain E.G. Richardson, "H.264 and MPEG-4", 홍릉과학출판사, 2004.9
[2] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, "JVT-G050r1", 2003.
[3] 권기상, "Edge 정보 기반 고속 전역 움직임 예측에서의 후보벡터 제거 기법 연구", 연세대학교 대학원, 2003.
[4] 최덕영외 2명, "DMB용 움직임 벡터 추출 프로그램 성능평가", 한국해양정보통신학회 vlo.9 no.1 pp641-645
[5] S.Y.Kung, "VLSI Array Processors", PRENTICE HALL, 1998.
[6] John Wiley · Sons, "Video CODEC Design", WILEY, 2002.