

H.264에서의 화면 간 예측을 위한 탐색 범위 선정 방법

박경석*, 이정원*, 조정현**, 김승호*

*경북대학교 컴퓨터공학과

**영남이공대학 모바일학과

e-mail: kspark, jwlee@mmlab.knu.ac.kr, petercho@ync.ac.kr, shkim@knu.ac.kr

A Selection Method of Adaptive Search Range for Inter Prediction in H.264

Kyung-Seok Park*, Jung-Won Lee*, Jeong-Hyun Cho*, Sung-Ho Kim*

*Dept. of Computer Engineering, kyungpook National University

**Dept. of Mobile Technology, Yeungnam College of Science & Technology

요 약

H.264/AVC는 동영상 데이터를 효율적으로 전송하기 위한 압축 기술로써 7가지 가변 블록에 대한 움직임 예측(Motion Estimation) 및 움직임 보상(Motion Compensation), DCT(Discrete Cosine Transform), 엔트로피 부호화(Entropy Coding), 디블록킹 필터 등 여러 가지 방법이 있다. 이러한 방법들로 인해 압축 효율은 높아졌으나 많은 계산량으로 인해 부호화 시간이 증가하게 되었다. 따라서 화질 손실을 최소한으로 줄이면서 소모되는 시간을 단축시키는 H.264 부호화 방법들이 개발되어야 한다. 그래서 본 논문에서는 H.264 부호화 과정 중 시간이 많이 소모되는 움직임 예측에서 움직임 벡터를 찾기 위한 방법을 제안한다.

1. 서론

1990년대에 들어오면서 기존의 아날로그 TV를 디지털 TV로 바꾸고자 하는 전 세계적 노력에 힘입어 영상 압축 기술이 비약적으로 발전되었다. 특히, 영상 전송에서 중요한 문제는 더욱 향상된 압축률, 전송 에러에 대한 유연성, 다양한 전송 환경에 대한 대처능력이다. 이러한 문제를 효과적으로 해결하기 위해 새로운 기술로써 H.264 또는 MPEG-4 Part 10 AVC (Advanced Video Coding)로 불리는 차세대 영상 압축 기술이 고안되었다.

H.264/AVC의 특징으로는 7가지 가변 블록에 대한 움직임 예측 및 RD (Rate-Distortion), 인트라 예측, 디블록킹 필터(Deblocking Filter), 다중 프레임 참조 등이 있으며 1/4 화소 정확도를 가지는 움직임 벡터(1/4 Pixel accuracy Motion Vector)의 사용으로 고품질 서비스가 가능하며 이전의 표준화 보다 강화된 엔트로피 부호화 방법이 제공된다[1, 2].

그러나 이러한 복잡한 연산의 증가로 인해 부호화 시간이 길어지고 고성능의 하드웨어를 필요로 하는 단점을 가지고 있다. 본 논문에서는 H.264 부호화 과정 중 시간이 많이 소모되는 부분인 움직임 추정에서 움직임 벡터를 찾기 위한 탐색 범위를 적응적으로 두어 움직임 추정의 소모시간을 줄이되 화질을 최소화하는 알고리즘을 제안한다.

본 논문의 구성으로 2장에서는 H.264의 주요기술을 설명하고 3장에서는 적응적으로 탐색 범위를 결정하는 알고리즘을 제안한다. 제안된 방법과 기존의 방법 간의 비교

실험을 4장에서 보이고 마지막장에서는 결론을 맺도록 하겠다.

2. H.264/AVC 주요 기술 소개

2.1 전역 탐색 기법

블록 움직임 추정 방법 중 탐색 영역 내에 모든 후보 블록과의 차이를 비교하여 가장 유사한 블록을 찾는 전역 블록 정합 방법은 예측 효율과 정확도 측면에서 가장 많이 사용되고 있는 기법이다. 전역 탐색 기법은 이전 프레임을 중심으로 현재 프레임을 추정하기 위해서 사용하는 방법이다. 이전 프레임 내에서 탐색 범위를 현재 매크로블록 크기의 2배로 정하고, 탐색 범위 내 전역을 탐색하여 최소 블록 정합 오차 값을 갖는 블록의 좌표를 찾아내어 움직임 벡터로 결정하는 방법이다. 그림 1은 전역 탐색 블록 정합을 나타낸 것이다.

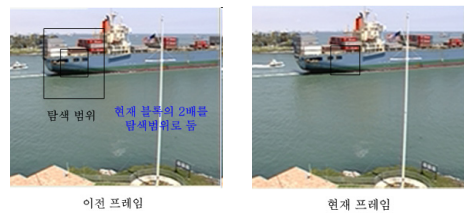


그림 1. 전역 탐색 블록 정합

그림 1과 같이 전역 탐색 블록 정합 기법은 탐색 영역 내의 모든 블록에 대해서 정합을 행하는 방식으로, 움직임 벡터를 정확하게 찾을 수 있으나 많은 계산량을 필요로 하게 된다. 이러한 이유로 부분 탐색 방법이나, 부분 정합 등을 이용하여 계산량을 감소시키면서 전역 탐색 블록 정합 방법과 유사한 화질을 갖는 고속 움직임 추정 방법이 많이 연구 되고 있다.

2.2 3단계 탐색 기법

결정된 탐색 영역 내에서 탐색 지점을 생략하는 고속 움직임 추정 방법으로 3단계 탐색 기법이 있다. 그림 2는 3단계 탐색 기법의 움직임 추정 패턴이다[3,4,5]. 그림에서 보듯이 넓은 탐색 간격의 초기 패턴으로부터 시작해서 탐색 간격을 1/2간격으로 좁히면서 3번의 단계를 거쳐서 움직임 벡터를 결정하게 된다.

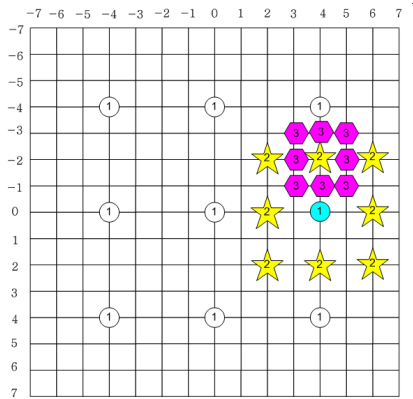


그림 2. 3단계 탐색 기법

3. 탐색 범위 선정 방법

움직임을 예측하는데 있어서 영상을 일정 크기의 블록들로 분할하고 움직임 벡터를 구하는 고속 블록 정합 기법들은 영상의 대부분의 블록들에서 움직임이 없거나 혹은 아주 작다는 특성에 바탕을 두고 있다. 그러므로 대부분 원점(0,0)을 중심으로 정해진 탐색 범위 이내에서 움직임 벡터를 찾는 과정에서 이루어진다. 예를 들어 3단계 탐색 기법의 경우 영상의 블록 정합 오차 값이 최고 정합 지점을 중심으로 거리가 멀어질수록 커진다는 가정 하에 움직임 추정을 수행한다. 그러나 이는 탐색 영역의 일부만 탐색을 수행하게 되므로 국부 최소점 (local minimum point)에 빠지게 되어 부정확한 움직임 벡터를 산출하게 된다. 이러한 단점을 해결하기 위해서 본 논문에서는 프레임 간에 배경은 거의 변하지 않고 물체는 빠르게 변하는 특징을 이용하여 각 블록마다 탐색 범위를 유동적으로 두어 움직임 예측의 소모시간을 줄이되 화질 열화를 최소화하는 알고리즘을 제안하고자 한다.

3.1 프레임간의 픽셀 값 차이 비교 방법

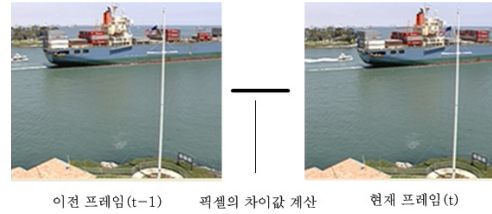


그림 3. 입력 영상 프레임간의 차이

그림 3과 같이 프레임간 사물은 빠르게 변하고 배경은 거의 변하지 않는다. 따라서 본 논문에서는 H.264에서 부호화과정을 거치기 전에 프레임 간에 픽셀 값 차이를 구해서 사물이 이동한 거리를 예측할 수 있다. 그리고 그림 4는 두 프레임의 픽셀간 차이 값을 나타낸다.

0	1	0	0	0	0	1	0	0	0	0	0	1	1	1	1
0	1	0	0	2	2	1	1	0	0	5	2	3	70	80	63
0	0	0	0	0	0	0	0	0	0	0	0	69	75	56	66
0	0	0	0	0	0	0	0	0	0	0	0	90	91	93	56
0	0	0	0	0	62	64	68	81	83	87	88	64	62	61	62
0	0	0	0	0	59	57	62	64	65	72	73	74	77	81	83
0	0	0	0	0	73	77	65	52	77	83	82	11	64	44	59
0	0	0	0	0	65	66	73	71	72	90	04	96	68	80	75
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	2	0	0	0	0	0	0	7	0
0	5	0	1	0	0	0	1	0	0	1	0	0	1	0	0
0	0	0	0	0	2	0	2	0	0	0	0	0	0	0	0
0	0	3	0	0	0	0	0	0	3	0	0	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

그림 4. 프레임 간의 차이를 나타내는 픽셀 값

그림 4와 같이 배경에서는 거의 픽셀의 차이가 발생하지 않지만 사물에서는 사물이 빠르게 이동하기 때문에 많은 차이가 나는 것을 확인할 수 있다. 그래서 본 논문에서는 사물인지 배경인지를 판단하기 위해서 기준치를 두었다. 픽셀 차이 값이 기준치 이상이면 사물이라고 판단하고, 기준치 이하일 경우에는 거의 움직임이 존재하지 않는다고 판단하였다. 기준치는 픽셀 차이 값에 따라서 움직임 추정할 때 걸리는 시간을 조사하였다. 그래서 기준치에 따라서 움직임 추정 시 걸리는 시간이 줄어드는 지점을 실험을 통해 기준치로 결정하였다. 실험 결과 기준치는 '33'으로 나오게 되었다. 즉, 프레임 간 픽셀 차이 값이 '33'이하인 경우에는 배경으로 간주하고, '33'이상의 값을 가지는 경우에는 사물로 판단하였다. 그래서 픽셀 차이 값과 기준치를 비교하여 픽셀 차이 값이 기준치보다 작은 경우에는 움직임 추정 과정 없이 현재 매크로블록의 정보를 그대로 전송 해 주게 된다. 이러한 프레임 간 픽셀 차이 값에 따라서 모드를 결정하게 되고 움직임 추정 시 픽셀 값에 따라서 탐색 범위를 유동적으로 두게 된다. 따라서 모드 결

정하는 과정이 간략화 되어서 계산량을 감소시킬 수 있다.

또한 요구되는 메모리를 줄이기 위해서 차이 픽셀 값이 기준치보다 큰 경우에는 1로 나타내고, 차이 픽셀 값이 기준치보다 작은 경우에 대해서는 0으로 나타내었다. 즉, 1인 경우는 사물로 볼 수가 있고 0인 경우는 배경으로 볼 수 있다.

3.2 블록 내에 픽셀 값의 비율에 따른 탐색 범위 선택 방법

H.264의 전역 탐색 기법은 움직임이 많거나 적거나 상관없이 동일한 탐색 범위를 적용시킨다. 하지만 본 논문에서는 사물의 움직임이 작은 부분부터 많은 부분까지 3가지로 분류해서 탐색 범위를 적용시킨다. 1의 값의 비율이 적으면 사물의 움직임이 많이 없는 경우이고, 1의 값의 비율이 높으면 사물의 움직임이 많은 경우이다. 그래서 블록 내에서 1의 값이 존재하는 픽셀 수에서 다르게 탐색 범위를 적용하여 PSNR(Peak Signal to Noise Ratio)을 비교해 보았다. 1의 값을 많이 가질수록 작은 탐색 범위로 움직임 벡터를 찾을 경우 오차가 날 확률이 높아진다. 따라서 1의 값을 가지는 픽셀의 수를 증가시켜 보면서 실험하였다. 실험 결과 82, 172 지점에서 같은 탐색 범위를 적용하였을 때 PSNR이 내려감을 알 수 있다. 따라서 픽셀의 비율을 1 ~ 33%, 34 ~ 66%, 67 ~ 100%로 나누어서 탐색 범위를 다르게 두었다.

또한, 탐색 범위를 증가시킬수록 화질의 열화는 적게 일어나지만 계산량은 증가하게 된다. 따라서 탐색 범위를 정해주기 위해서 탐색 범위에 따른 PSNR을 비교해 보았다. 탐색 범위가 현재 블록의 1.3배, 1.6배, 2배가 되는 지점에서 PSNR이 올라갔다. 따라서, 차이 픽셀 값이 33이상인 경우가 0 ~ 33%이면, 움직임이 거의 없는 영상이라고 판단하여 전역 탐색 범위보다 작은 현재 매크로블록에서 1.3배의 탐색 범위를 적용시킨다. 둘째, 차이 픽셀 값이 33상인 경우가 34 ~ 66%이면, 움직임이 조금 있는 경우로 판단하여 탐색 범위를 현재 매크로블록의 1.6배로 둔다. 셋째, 차이 픽셀 값이 33이상인 경우가 67 ~ 100%이면, 보통 물체의 경우로 움직임이 많은 경우로 판단하여 전역 탐색 범위로 둔다. 이렇게 하여 탐색 범위를 유동적으로 두어서 계산량을 줄이고자 하였다.

4. 실험 결과

빠르고 효율적인 매크로블록 모드의 결정 알고리즘을 통하여 부호기의 계산량을 감소시키기 위하여 제안한 본 알고리즘의 실제 성능을 실험하였다. 실험은 일반적으로 영상 압축 표준안을 실험할 때 많이 사용되는 QCIF크기의 실험 영상 container, news 등의 2개를 사용하였다. 그림 5는 시뮬레이션에 사용한 실험 영상 2개의 첫 화면이다.



그림 5. 2가지 QCIF 크기의 실험 영상

4.1 제안한 알고리즘의 실험 결과

본 논문에서 제안한 탐색 범위를 적용하기 위해서 픽셀의 기준치를 두어서 기준치를 초과하는 픽셀의 값을 조사하였다. 그래서 적합한 기준치를 조사하기 위해서 픽셀의 값에 따른 움직임 추정 시간을 측정하였다. 그림 6은 기준치에 따른 움직임 추정 시 소모되는 시간을 측정한 그래프이다.

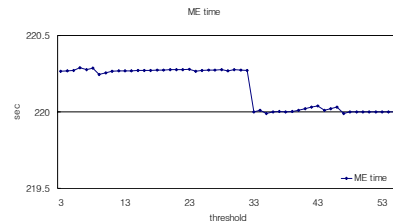


그림 6. 기준치에 따른 움직임 추정 시 시간 비교 그래프

그림 6에서와 같이 본 논문은 기준치의 값을 '33'으로 정하였다. 그래서 본 논문에서는 '33' 이상의 값에 대한 비율을 조사하여 탐색 범위를 유동적으로 두어서 부호화 시간을 줄이고자 하였다. 비율을 조사하기 위해서 매크로블록 안에 있는 픽셀의 개수에 따른 PSNR을 조사하였다. 매크로블록 안에 있는 픽셀의 개수는 256개이며, 개수에 따라서 탐색 범위를 정하였을 때 PSNR이 낮아지는 지점을 조사하였다. 그림 7은 픽셀의 개수에 따른 PSNR에 대한 그래프이다.

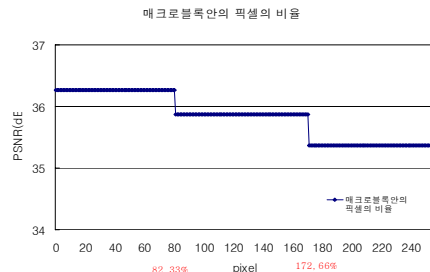
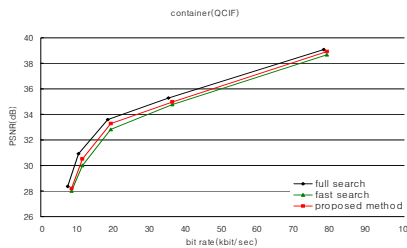


그림 7. 픽셀의 개수에 따른 PSNR

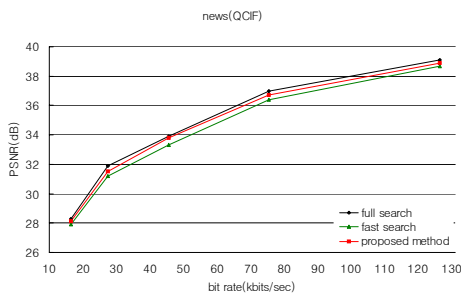
그림 7에서와 같이 기준치 이상의 픽셀 개수에 따라 PSNR이 약간씩 차이가 있음을 알 수 있다. 하나의 매크로블록 안에 기준치 이상의 픽셀의 개수가 82와 172에서

PSNR의 차이가 발생한다. 따라서 픽셀의 개수에 따른 확률을 계산하여 약 33%와 66%지점에서 각각 다른 탐색 범위를 적용시켜 준다.

지금까지 실험 결과를 통해 제안한 알고리즘을 적용하기 위해서 PSNR과 부호화 시간 그래프를 통해서 나타내었다. 그림 8은 비트량에 따른 PSNR 그래프이다. 그림 8에서와 같이 전역 탐색 기법과 본 논문에서 제안한 방법의 PSNR이 유사한 것을 볼 수 있다.



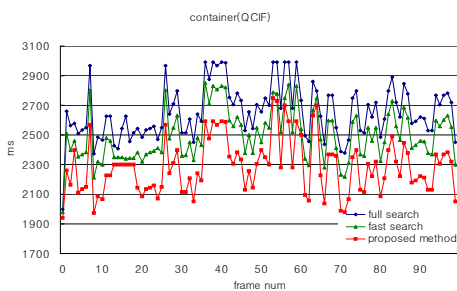
(a) Container(QCIF) 영상에서 bit rate에 따른 PSNR 비교



(b) News(QCIF) 영상에서 bit rate에 따른 PSNR 비교

그림 8. 기존의 방법과의 PSNR 비교 그래프

그림 9는 기존 방법과 제안하는 방법을 사용했을 때 각 프레임들을 코딩하는 시간(ms)을 비교해서 나타낸 그래프이다. 그림 9에서 두 종류 영상(container, news)의 경우 모두 제안하는 방법과 비교해서 각 프레임마다 0.4 ~ 0.5 초정도 빠르게 부호화됨을 볼 수 있다.



(a) Container(QCIF) 영상에서 시간 비교



(b) News(QCIF) 영상에서 시간 비교

그림 9. 기존의 방법과의 시간 비교 그래프

5. 결론

H.264에서는 미리 탐색 범위를 정의해 두고 7가지 가변 블록에 대한 매크로블록 모드에서 움직임 추정을 전역탐색 기법을 사용하여 시행한다. 그중에 가장 효율적인 하나의 모드와 움직임 벡터를 선택한다. 이와 달리 본 논문에서는 미리 부호화기에서 빠르게 움직이는 사물인지 거의 움직임이 없는 배경인지를 먼저 판단하여 움직임이 많은 사물인 경우에는 탐색 범위를 많이 두고 움직임이 적은 경우에는 탐색 범위를 적게 두는 방법을 제안하였다.

실험결과 제안하는 방법이 기존의 H.264 전역탐색 기법과 비교하여 화질과 비트 율은 거의 비슷하게 유지하면서 실험 영상들의 경우 모든 프레임에서 평균 0.4 ~ 0.5초 정도의 부호화 시간이 단축됨을 보였다. 제안된 방법을 이용하여 실시간 영상 코딩 응용 분야 및 소프트웨어 구현에 도움이 될 것이다.

참고 문헌

- [1] G. J. Sullivan and T. Wiegand, "Rate -Distortion Optimization for Video Compre -ssion", IEEE Signal Processing Magazine, vol.15, pp.74-90, Nov. 1998
- [2] P. List, A. Joch, J. Lainema, M. Karczewicz, "Adaptive Deblocking Filter," IEEE Transactions on Circuits and Systems for Video Technology, vol.13, no.7, pp. 614-619, July 2003
- [3] Jong-Ho Kim, Byung-Gyu Kim "Fast block mode decision algorithm in H.264/AVC video coding," J. Vis. Commun. Image R. 19 (2008) 175 - .183, 2007
- [4] S. Kappagantula and K. R. Rao, "Motion compensated predictive coding," in proc. Int. Tech. Smp., SPIE, San Diego, CA, pp.64-70, Aug.1983
- [5] M. Bruing, W. Niehsen, "Fast full-search blockmatching," IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no.7, pp. 241-247, Feb. 20