

움직임 정보를 이용한 H.264에서의 적응적 참조 화면 선택 기법

Adaptive Reference Picture Selection Method
using Motion Information in H.264

권 혁 민*, 최 윤 식**

(Hyok-Min Kwon and Yoon-Sik Choe)

* 연세대학교 전기전자공학과(전화:(02)2123-2774, 팩스:(02)313-0960, E-mail : KongMin@hitel.net)

** 연세대학교 전기전자공학과(전화:(02)2123-2774, 팩스:(02)313-0960, E-mail : yschoe@yonsei.ac.kr)

요약문 : 본 논문에서는 최신 동영상 압축 표준 규격 H.264에서 화면 간 예측 부호화시 참조 화면 수가 제한되어 있을 때 부호화 효율을 최대화할 수 있는 적응적 참조 화면 선택 기법을 제안하였다. 참조 화면 선택 시 시간적인 거리를 우선시 하는 현재 방식은 부호화 효율 측면에서 최적의 선택과는 거리가 있다. 따라서 참조 화면 선택에 있어서 실시간 온라인 상황을 가정하고 부가적인 계산을 최소화하는 것을 조건으로 했을 때 대표 화면과의 간접 비교를 통하여 현재 참조 화면들 간 상호 유사성이 가장 큰 화면을 제외하는 방법을 통해 문제를 해결하고자 하였다. 화면 간 예측 부호화 시 참조 화면 간 중복되는 요소를 최대한 줄이는 방향으로 참조 화면을 선택해 나감으로서 참조되는 샘플들을 다양화할 수 있다. 참조 화면 간 상호 유사성 판별은 대표 화면과 각 참조 화면 간 움직임 정보를 고려해 화면 간 부호화시 계산되는 8*8 블록의 R-D 비용함수 값을 사용하였다. 실험 결과 기존의 슬라이딩 윈도우 방식을 이용한 참조 화면 선택 방법에 비해 부가적인 계산이 거의 없이 부호화 효율의 개선이 있었으며 그 개선 정도는 기존의 방식에서 참조 화면을 1~2장 더 사용한 결과와 같았다. 이는 기존의 방법에 의해 참조 화면을 선택할 경우 나타나는 참조 화면 간 중복성을 줄였기 때문이다.

핵심단어 : H.264, 화면간 화면 간 예측 부호화, 다중 참조 화면

1. 서 론

최근 표준화 과정에 있는 동영상 압축 표준 H.264에서는 기존의 표준들에 비해 부호화 효율을 비약적으로 증가시켰고 상대적으로 그 부호화를 위한 도구들(Tool)은 그 종류를 단순화 하였다. 특히 4x4 정수 변환, 다양한 화면내/화면간(Intra/Inter) 예측 부호화 모드, Deblocking 필터, 향상된 엔트로피(Entropy) 부호화 등을 통해 그 부호화 효율을 높이고 있으며 어느 하나의 부호화 도구에 의한 큰 효율 개선이기 보다는 각각의 도구를 이용한 작은 개선으로 전체적인 큰 효율 향상을 나타내고 있다.[1]

이중에서 화면내/화면간 예측 부호화 부분은 부호화 효율의 개선에 있어서 가장 많은 효과를 보이지만 부호화 과정에서의 복잡도와 시간은 전체 부호화 과정에서 크게 늘어났다. 따라서 예측 부호화 모드의 효율적인 사용에 따른 부호화 효율 개선보다는 실제적인 응용을 목적으로 효율과 복잡도의 균형을 이루는 기법에 대한 연구들이 주로 많이 이루어지고 있다.

그러나 복잡도와 속도의 문제뿐만 아니라, H.264 표준에 채택되고 정의되어 있는 부호화를 위한 여러 도구들이 표준화 과정 중에 고려된 기법이나 효율성 외에도 어떠한 기법으로 활용이 가능하고 어느 정도

까지 그 효율을 낼 수 있는지 등의 활용과 효율에 관한 문제도 중요할 것이다. 이런 관점에서 본 논문에서는 H.264의 다중 참조 화면을 이용한 화면간 예측 부호화시 부가적인 복잡도와 시간의 증가 없이 최대의 부호화 효율을 위한 참조 화면 선택 기법을 제안하고 JM(Joint Model)에 적용하여 기존의 방식과 비교 실험 및 분석을 하고자 한다.

II. H.264의 다중 참조 화면

H.264에서는 부호화기가 현재 화면을 부호화한 후 부호화기 내부의 복호화기에서 재 복호한 화면(Reconstructed Picture)이 메모리 버퍼 DPB(Decoded Picture Buffer)에 저장되고 난 후 슬라이스 헤더의 정보에 따라 DPB에 있는 화면들에 대한 참조 화면으로서의 지정과 해제, 재순서화 등의 제어가 이루어진다. DPB의 참조 화면 저장을 위한 버퍼는 별도의 제어 신호 없이 참조 화면 수만 정하면 그 수만큼 최근의 복호된 화면을 FIFO 방식으로 선택해 저장하는 Short-Term Memory와 시간적인 거리나 지속 시간과 상관없이 임의로 참조 화면으로서 지정하고 해제할 수 있으며 따라서 상대적으로 참조 화면이 길게 저장되어 있을 수 있는 Long-Term Memory로 나누어진다.

Short-Term과 Long-Term을 합친 수가 전체 참조 화면 수가 된다. 현재 유효하게 활성화된 참조 화면의 수와 참조 화면 목록은 슬라이스 마다 갱신될 수 있다.

H.264의 슬라이스 헤더에서 참조 화면 제어를 위한 구문(Syntax) 중 참조 화면의 적응적인 선택을 위해서는 `dec_ref_pic_marking()` 구문(DRPM)이 가장 중요하다. DRPM구문의 정보에 따라 현재 복호화한 화면까지 포함하여 DPB의 참조 화면들은 참조 화면에서 제외, Short-term으로 지정, Long-Term으로 지정 등의 제어가 가능하다. 특히 DRPM의 `adaptive_ref_pic_marking_mode_flag`에 의해 적응적인 참조 화면 제어를 할 것인지 아닌지를 결정할 수 있다. 이 flag가 0일 경우 그 외의 부가적인 정보 없이 참조 화면 선택은 Short-Term 참조 화면에 대해 슬라이딩 윈도우(Sliding window) 방식으로 FIFO의 순서로 이루어진다는 것을 의미한다. 반면 1일 경우 적응적인 참조 화면 선택이 가능해지며 이 구문 이후로 참조 화면 제어를 위한 구문들이 존재하고 그 구문들을 통해 참조 화면의 제어를 하겠다는 의미이다. 이때 참조 화면 제어를 위한 구문 `memory_management_control_operation` (MMCO)에는 모두 7가지가 있으며 그에 따라 참조 화면에 대한 정보를 가지고 있는 참조 화면 목록이 갱신되고 이렇게 갱신된 목록으로 다음 화면의 화면간 예측 부호화시 참조 된다. 참조 화면 목록은 단방향 예측을 위한 Reference Picture List 0와 B 슬라이스일 경우의 양방향 예측을 위한 Reference Picture List 1 두개가 있을 수 있다.

현재 H.264의 JM에서는 참조 화면 수만큼 최근 복호된 화면을 Short-Term 참조 화면으로 저장하고 슬라이딩 윈도우 방식으로 DPB에서 참조 화면 선택이 이루어지는 방식을 기본적으로 취하고 있다. 실험적으로 보면 영상 신호를 화면간 예측 부호화 하는 경우 시간적인 거리가 다양한 참조 화면을 여러 장 두고 부호화 했을 때 그 시간적 거리가 짧은, 즉 가장 최근의 참조 화면일수록 참조되는 빈도가 높다는 것을 알 수 있다. 따라서 구현이 간단한 장점까지 가지고 있는 슬라이딩 윈도우 방식의 Short-Term 참조 화면 제어 기법을 기본으로 하고 있는 것이다. 참조 화면 수 M이 주어졌을때 하나의 화면이 부호화 및 재부호화가 끝나면 DPB의 M개의 참조 화면 중 가장 오래전의 것을 버리고(Mark as Unused for Reference) 방금 복호된 화면을 참조 화면 리스트의 인덱스 0으로 하여 저장한 후 참조 화면 리스트의 인덱스를 재정렬한다. 그 후 다음 화면에 대한 부호화를 시작하는 것이다.

최근의 복호된 화면들을 우선으로 가지고 있는 현재의 Short-Term 참조 화면 선택 방식은 일반적으로 타당해 보인다. 실험적으로 시간적 거리가 짧을 수록 그 화면이 참조되는 경우가 많고 그 알고리즘 또한 간단하기 때문이다.[2] 하지만 시간적 거리가 가장 작은 화면들이 다음 화면을 부호화 하는데 있어서 가장 효

율적인 참조 화면이 되는지는 생각해 볼 문제다. 화면 사이의 시간적인 거리는 화면사이의 유사성과는 별개의 문제이고 따라서 참조 화면을 선택 할 때에는 각 참조 화면의 참조 화면으로서의 유용성 또는 중요도를 우선시해서 고려해야 할 것이다. 또한 참조 화면의 개수는 움직임 추정시 복잡도와 연산량, 움직임 보상시 요구되는 메모리 대역폭, 메모리 크기에 따른 하드웨어적인 비용(cost) 등의 이유로 제한될 수 밖에 없기 때문에 참조 화면 수가 M으로 제한 되어있다는 상황에서 최적의 참조 화면 선택에 대한 문제를 시작해야 한다. 따라서 본 논문에서는 현재 JM의 슬라이딩 윈도우 방식의 참조 화면 선택 방법은 참조 화면으로서의 중요도를 고려하지 않기 때문에 화면간 예측 부호화 시 그 이상의 부호화 효율 개선의 가능성을 항상 가지고 있음을 그 한계점으로 인식하고 주어진 참조 화면 개수 하에서 가장 최적의 부호화 효율을 얻을 수 있는 참조 화면들을 선택하는 문제에 대한 해결점을 찾고자 하였다.

III. 제안한 기법

이 문제를 해결하기 전에 두 가지 조건을 두었다. 현재 부호화한 화면과 과거에 입력 받은 영상 화면들만을 고려하고 미래의 화면에 대해서는 미리 알고 있지 않은 실시간 온라인 상태를 조건으로 한다. 또 하나의 조건은 참조 화면 선택과정에서 부가적인 복잡도와 연산량을 최소화해야 한다는 것이다.

첫 번째 조건에 의하면 참조 화면 선택에 대한 결정은 현재 참조 화면으로 두고 있는 M장의 참조 화면들 $R(0) \sim R(M-1)$ 과 현재 부호화 후 재부호화한 화면 $F(n)$ 만을 고려해서 이루어져야 한다. 따라서 참조 화면으로서의 기본적인 선택 조건은 다음 화면 $F(n+1)$ 을 부호화 하는 데에 가장 좋은 부호화 효율을 나타낼 수 있는 화면이어야 한다는 것이다. 즉 현재 화면을 부호화하고 난 후 참조 화면의 선택을 결정 할 수 있어야 하며 매 장 마다 최적의 선택이 이루어져야 한다.

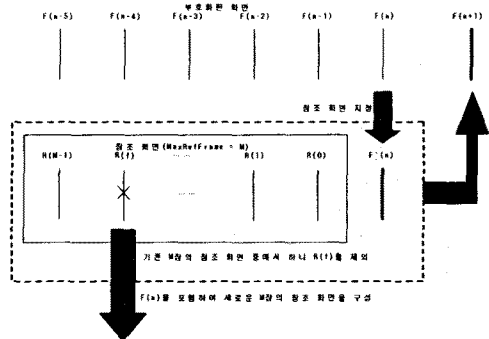


그림 1. 참조 화면 선택 과정
Fig. 1. Reference Picture Selection Process

이때 현재 화면 $F'(n)$ 을 부호화한 후 예측 부호화 시 참조했던 M 장의 참조 화면 $R(0) \sim R(M-1)$ 중에서 선택적으로 $R(f)$ 를 참조 화면에서 제외하고 현재 부호화하고 재 부호화한 화면 $F'(n)$ 을 대신 참조 화면으로 지정하여 저장하는 그림 1의 상황을 가정하자. 그러면 다음 화면 $F(n+1)$ 을 부호화 할 경우에는 참조 화면 $R(0) \sim R(f-1), R(f+1) \sim R(M-1), F'(n)$ 등 이렇게 M 장의 새롭게 구성된 참조 화면을 이용하여 예측 부호화 할 것이다. 이때 참조 화면 선택 과정에서 일어나는 손해는 $F(n+1)$ 이 예측 부호화될때 참조 화면 $R(f)$ 가 빠진다는 것이다. 따라서 참조 화면 중 하나 $R(f)$ 를 제외하는 문제는 그 것을 제외함으로써 일어나는 $F(n+1)$ 의 부호화 효율 손실을 최대한 억제하는 방향으로 선택해 나가야 할 것이다. 또한 $F(n+1)$ 이 불확실한 상황에서 부호화 효율을 높이기 위해서는 $F(n)$ 과 $R(0) \sim R(M-1)$ 과의 관계 보다는 참조 화면들끼리 참조 화면으로서의 가치가 얼마나 있는나 더 중요할 것이다. 참조 화면으로서의 가치는 그 참조 화면이 움직임 예측 부호화시 블록간 벡터 양자화[2]의 코드북으로서 얼마나 다양한 움직임 위치 관계의 샘플들을 가지고 있는나가 된다. 다시 말하면 각각의 참조 화면이 참조 화면으로서 높은 가치를 지니려면 다른 참조 화면들과 비교하여 상대적으로 다른 움직임 위치 관계를 가지고 있어야 하며 결과적으로 미지의 화면 $F(n+1)$ 의 예측 부호화를 위해 더 좋은 참조 화면 조합이 될 것이다. 결국 $F(n+1)$ 을 예측 부호화 하는데에 참조 화면으로서 그 가치가 떨어진다고 판단되어 버려지는 $R(f)$ 는 $F(n)$ 부호화시 가장 적은 효율 기여도를 보인 참조 화면이 아니라 다른 참조 화면과 움직임 관계나 텍스처(Texture)가 유사하여 그 참조 화면이 없어도 $F(n+1)$ 을 부호화시 다른 참조 화면에서 충분히 그 역할을 대신 할 수 있는, 즉 다양한 코드북으로서의 가치가 가장 적은 참조 화면이 되어야 한다. 이로써 $F(n+1)$ 의 예측 부호화를 위한 참조 화면 선택은 $R(0) \sim R(M-1)$ 중 서로 움직임 위치 관계와 텍스처(texture)가 비슷하여 다른 참조 화면들과 중복성이 많은 참조 화면을 구하는 문제로 바꿀 수 있다.

참조 화면 중에서 움직임의 위치관계가 다른 화면들과 중복성이 많고 유사성이 많은 화면을 찾는 문제는 두 번째 조건을 우선 고려하여야 한다. 그러면 참조 화면들끼리 어떤 계산이나 예측을 통해 중복성을 찾는 방법은 상당한 부가적인 계산 과정이 요구되기 때문에 배제한다. 즉 움직임의 위치 관계에 따른 유사성 또는 중복성을 찾기 위해 참조 화면들 서로를 비교하기 보다는 대표될 만한 하나의 화면과 비교하여 간접적으로 그 중복성을 구하는 것이 계산량에 있어서 훨씬 이득이 될 것이며 또한 블록 단위로 그 중복성을 구할 수 있어야 한다. 그 대표가 될 수 있는 화면으로는 $F(n+1)$ 과 시간적으로 가장 가까운 거리에 있고 각 참조 화면 $R(0) \sim R(M-1)$ 에 대해 움직임 추정 과정을

수행했던 $F(n)$ 이 가장 적합하다 할 수 있다. 따라서 현재 화면 $F(n)$ 을 참조 화면 $R(0) \sim R(M-1)$ 들과 움직임 예측 부호화할 때 그 과정에서 나오는 움직임에 관한 정보를 이용하여 간접적인 참조 화면 간 비교를 통해 움직임의 위치 관계가 가장 많이 중복되고 유사성이 많은 참조 화면 $R(f)$ 를 구하고자 한다.

H.264의 JM에서는 P 슬라이스의 매크로블록 하나를 부호화 하는데 각 분할 모드에 따라 움직임 추정을 통해 최적의 움직임 벡터와 참조 화면을 구하고 이를 이용하여 부호화시 소요되는 비트량(Rate)과 그 왜곡 정도(Distortion)를 고려하여 라그랑주 유효한 모드 결정 기법을 사용하여 그 비용 함수가 가장 적은 최적의 매크로블록 모드를 결정한다.[3] 이 과정에서 얻을 수 있는 정보는 매크로블록 타입과 각 모드에 따른 비용 함수값, 움직임 벡터, 참조 화면 번호, 차이 계수에 대한 것들이고 이 정보들을 이용하면 부호화하는 매크로블록이 각 참조 화면과 일만큼의 움직임 변위가 일어났고 차이(Distortion)가 어느 정도 인지 알 수 있을 것이다. 따라서 $F(n)$ 의 각 매크로블록을 부호화 할때 마다 M 장의 참조 화면 중 $F(n)$ 과 움직임 변위 정도와 차이가 가장 비슷한 두 참조 화면을 찾을 수 있을 것이다. 그리고 $F(n)$ 의 모든 매크로블록에 대해 수행하고 나면 가장 많이 유사함이 발견된 참조 화면을 찾을 수 있고 이 참조 화면이 다른 참조 화면들과 움직임 위치 관계의 유사성 또는 중복성이 가장 많은 화면이라 판단할 수 있다. 이때 한 가지 더 고려해야 할 것이 있는데 바로 실제 부호화되는 출력 비트량이다. 즉 참조 화면 인덱스나 움직임 벡터의 변위가 다르면 출력 비트 또한 달라지기 때문에 부호화 효율의 측면에서 유사성과 중복성도 고려하려면 출력 비트량 또한 고려해야 한다. 게다가 매크로블록은 다양한 크기의 분할 모드로 나누어 질 수 있어 비교 되는 참조 화면 대상 간 움직임 벡터의 개수조차 달라질 수 있다. 따라서 출력 비트량과 다양한 블록 크기를 동시에 고려한다면 예측 부호화 과정 중 식(1)을 선택할 수 있다.

$$J = SSD + \lambda Rate(MV, REF) \quad (1)$$

식 (1)은 하나의 매크로블록을 네 개의 8*8 블록으로 나누고 각 블록마다 모든 모드와 참조 화면에 대해 움직임 추정을 R-D비용함수에 따라 수행하는 식이다. 또한 움직임 벡터가 개별적으로 참조 화면 번호를 가질 수 있는 최소 단위가 8*8 블록이고 실제 부호화 소요 비트량이 고려되어 있는 식이기 때문에 이 식을 각 8*8 블록에 대한 유사도의 판별을 위한 식으로 활용할 것이다. 즉 $F(n)$ 의 각 8*8 블록에 대해서 각 참조 화면마다의 비용 함수 값이 예측 부호화 과정에서 구해지면 그중 비용 함수값이 서로 가장 비슷한 참조 화면들을 구하고 그 참조 화면들의 유사성 발견 횟수를 저장하는 변수를 증가시킨다. 그리고 모든 매크로블록에 대해 수행하고 나서 가장 많은 유사성 발견 횟수의 참조 화면을 참조 화면에서 제외시킨다. 이때 만일 매크로블록이 Intra4*4 또는 Intra16*16일 경우에는 $F(n)$ 이

참조 화면간 유사성을 판단하기 위한 기준 화면으로서 보기 힘들다고 간주 하고 그 매크로블록에 대해서 참조 화면 간 유사성에 대한 판단을 하지 않는다. 하지만 매크로블록 모드가 Skip일 경우에는 Inter 모드와 마찬가지로 참조 화면 간 유사성을 판단한다. 이렇게 찾아낸 참조 화면은 MMCO를 이용해서 매 화면마다 참조 화면에서 제외시킨다.

IV. 실험 결과

제한한 참조 화면 선택 기법을 H.264의 JM 7.3에 적용하여 실험하였다. 실험 조건은 7가지 모든 Inter 예측 모드를 사용하였고 움직임 추정 범위는 -16~16 화소로 하였다. 엔트로피 부호화는 CAVLC를 사용하였고 Hadamard 변환과 R-D 최적화를 사용하였다. B 슬라이스는 사용하지 않았고 IPPP구조로 고정 QP값으로 26을 주었다. Frame모드로 부호화하고 참조 화면의 수는 1~5장을 두고 Frame Skip=1로 하여 총 30장을 부호화하였다. 실험 결과는 참조 화면 수에 따른 QCIF크기의 foreman과 akiyo영상의 P화면 PSNR과 비트량을 나타낸 것이다.

표 1. foreman영상의 실험 결과
Table 1. Experimental Result for foreman

	Sliding Window		ARPS	
	PSNR	Bit	PSNR	Bit
M=1	37.55	208544	37.55	208544
M=2	37.69	205128	37.69	205128
M=3	37.72	203680	37.77	202104
M=4	37.76	202426	37.80	199312
M=5	37.78	200040	37.82	197800

표 2. akiyo영상의 실험 결과
Table 2. Experimental Result for akiyo

	Sliding Window		ARPS	
	PSNR	Bit	PSNR	Bit
M=1	39.81	41824	39.81	41824
M=2	39.81	40272	39.81	40272
M=3	39.84	40488	39.86	40000
M=4	39.86	39552	39.88	39408
M=5	39.88	39488	39.87	39358

실험 결과를 살펴보면 제안한 방식(ARPS)을 사용했을 때 슬라이딩 윈도우 방식(Sliding Window)보다 PSNR과 총 비트량에서 개선이 있음을 알 수 있다. 그 개선 정도는 절대치로 보면 PSNR은 0.02~0.05dB, 비트량은 0.5~1.5% 정도로 미비하나 그 효과는 주어진 참조 화면 수 M보다 1~2장 더 사용한 것과 거의 같음을 알 수 있다. 이는 비록 참조 화면 수가 제한되어 있더라도 참조 화면간 중복성을 제거함으로써 그 이상의 장수와 같은 효과를 얻었기 때문이다. M이 2 이하일 경우는 결과적으로 슬라이딩 윈도우와 같은 참조 화면들을 사용하므로 3 이상일 때 부터 그 개선 효과를

얻을 수 있는데 특히 M이 적을수록 더 많은 개선 효과를 보인다. 이는 참조 화면이 증가할 수록 그에 의한 효과가 감소하기 때문이며 그 효과는 M의 로그값과 거의 선형적인 관계가 있다고 알려져있다.[4] 따라서 M이 1->2로 증가할 때와 2->4로 증가할 때 그 효과가 비슷하다 할 수 있으며 제안한 ARPS방식이 더 이러한 특성과 부합한다 할 수 있다. 또한 입력 영상에 따라 그 개선 정도가 다른데 이는 foreman같은 상대적으로 동적인 영상이 다중 참조 화면에 의한 효과가 크기 때문이며 제안한 방식 또한 중복성 제거를 이용해 참조 화면 수를 더 늘이는 것과 같은 효과를 나타내는 것이므로 foreman이 akiyo보다 더 효율 개선 정도가 크다. 결과적으로 참조 화면 수 M을 3~4정도로 두고 제안한 기법을 사용하면 참조 화면 수 M을 5장 이상 둔 것과 같은 부호화 효율 개선 효과를 낼 수 있으며 또한 같은 부호화 효율을 내면서 참조 화면 수를 줄일 수 있기 때문에 결과적으로 움직임 예측 시간을 줄이는 효과도 나타낼 수 있다.

V. 결론

본 논문에서는 다중 참조 화면을 이용해서 화면간 예측 부호화시 부호화 효율을 높일 수 있는 참조 화면 선택에 대한 기법을 제안하였다. 부호화 과정에서 R-D비용함수로 계산되는 움직임 정보를 고려해 부가적인 계산 없이 참조 화면 간 중복성을 제거하는 방향으로 참조 화면을 적용적으로 선택함으로써 계산량 증가 없이 효율 개선을 얻을 수 있었으며 실험 결과를 통해 확인하였다. 기존의 일률적인 참조 화면 선택보다는 단기간이라도 참조 화면 간 특징을 고려해 적용적으로 참조 화면을 선택함으로써 제한된 참조 화면 수에서도 부호화 효율을 개선할 수 있음을 보인 것이다.

참고 문헌

- [1] JVT of ITU-T and ISO/IEC JTC 1, "Draft ITU-T Recommendation and FDIS of Joint Video Specification", JVT-G050r1, May, 2003.
- [2] T.Wiegand X.Zhang, and B. Girod, "Long-Term Memory Motion-Compensated Prediction", IEEE Transaction on Circuits and Systems for Video Technology, Sep, 1998
- [3] JVT of ITU-T and ISO/IEC JTC 1, "JM Reference Encoding Methods and Decoding Concealment Methods", JVT-I049d0, Sep, 2003.
- [4] B. Girod, "Efficiency Analysis of Multihypothesis Motion-Compensated Prediction for Video Coding", IEEE Transaction on Image Processing, Vol9, No2, Feb, 2000