



저작자표시-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권으로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

H.264/AVC에서 고속 윌왜곡 최적화 및
선택적 인트라 예측에 대한 연구



부경대학교 대학원

이미지시스템공학과

신 세 일

공학석사 학위논문

H.264/AVC에서 고속 윌왜곡 최적화 및
선택적 인트라 예측에 대한 연구

지도교수 오 정 수

이 논문을 공학석사 학위논문으로 제출함

2008년 2월

부경대학교 대학원

이미지시스템공학과

신 세 일

申世一の 工學碩士 學位論文 認准함

2008年 2月



主 審 工學博士 金 志 泓

委 員 工學博士 鄭 容 錫

委 員 工學博士 金 鐘 守

목 차

목차	i
그림목차	iii
표목차	iv
Abstract	v
 I. 서론	 1
II. H.264/AVC의 새로운 움직임 추정 및 보상 기술	5
1. 가변 블록 움직임 추정 및 보상	5
2. 1/4화소 정밀도의 움직임 추정 및 보상	7
3. 다중 참조 픽처 움직임 추정 및 보상	10
4. 율-왜곡 최적화	12
5. 인트라 예측 부호화	14
5.1. 휘도 성분 4x4 인트라 예측 부호화	14
5.2. 휘도 성분 16x16 인트라 예측 부호화	15
5.3. 색차 성분 8x8 인트라 예측 부호화	16
III. 제안하는 알고리즘	18
1. 적응적 율-왜곡 최적화 알고리즘	18
1.1. RD_cost와 MV_cost관계	21

1.2. 제안하는 알고리즘	22
2. 선택적 인트라 예측 부호화 알고리즘	28
2.1. 영상 복잡도	28
2.2. P-picture에서의 인트라 예측 부호화 성능 분석	29
2.3 P-picture에서 인트라 예측 모드의 선택 조건 분석	31
2.4 제안하는 알고리즘	33
IV. 모의 실험 및 결과	38
V. 결 론	45
VI. 참고문헌	47

그 립 목 차

그림 1. H.265/AVC의 가변 블록들의 크기	5
그림 2. 1/4 화소 정밀도의 화소값 생성 방법	9
그림 3. 단일 참조 픽처와 다중 참조 픽처 방식	11
그림 4. 율-왜곡 곡선과 lagrangian	13
그림 5. 4x4 크기의 인트라 예측 모드	14
그림 6. 16x16 크기의 인트라 예측 모드	16
그림 7. JM에서의 움직임 추정 및 보상 루프	24
그림 8. 제안된 고속 율-왜곡 최적화 알고리즘	27
그림 9. QP에 따른 weight	35

표 목 차

표 1. Low Complexity Mode (RDO-off)	19
표 2. 모드 1~3, 8에 대한 MV_cost와 RD_cost, Best_mode	23
표 3. QP 별 인트라 예측 모드 선택 경향	31
표 4. 최적 모드의 종류에 따른 매크로 블록의 영상 복잡도	32
표 5. 인트라 모드가 최적으로 선택 될 경우 최적의 인터 모드 ...	33
표 6. 실험 영상 크기 및 환경	38
표 7. 실험 환경	39
표 8. P-picture에서 인트라 예측 부호화 없이 울-왜곡 최적화에 따른 결과	40
표 9. 제안하는 적응적 울-왜곡 최적화, 선택적 인트라 예측 부호화 알고리즘의 결과	43
표 10. 제안하는 알고리즘의 공동 적용 결과	44

Fast Rate Distortion Optimization Algorithm for H.264/AVC

Se Il Shin

*Department of Image System Science & Engineering,
The Graduate School, Pukyong National University*

Abstract

H.264/AVC is video image international standard that has superior video compression performance to conventional video image standard. H.264/AVC uses new technologies to improve encoding efficiency. Various inter prediction block modes and intra prediction block modes and rate distortion optimization algorithm among new technologies decide the best block mode from them. It greatly improves a bit rate but increases an amount of computation of H.264/AVC. This paper proposes fast rate distortion optimization algorithm and optional intra prediction algorithm in H.264/AVC. Fast rate distortion optimization algorithm omits to compute the optimization cost by predicting it from the cost calculated for motion estimation. And optional intra prediction algorithm in H.264/AVC omits intra prediction to using image complexity and best inter mode and predicted motion vectors. The simulation results show that mix up two proposed algorithm, on average, keeps about 95% of a bit rate gained by the rate distortion optimization while reduces about 50% of computation added by it.

I. 서론

1980년대 중반 네트워크와 음성 및 영상 저장미디어 정보의 디지털화가 진행되면서 멀티미디어의 시대가 시작 되었으며 최근에는 광대역 시대로의 급속한 발전과 더불어 컴퓨터, 통신 기구 등의 하드웨어적 기술 향상으로 통신과 멀티미디어의 융합이 주목 받고 본격적인 영상 통신 시대로 도래하고 있다. 이러한 시대적 요청에 의해 정보 전달 매체로서 동영상이 부각되고 있으나 문서 및 정지 영상과 같은 이전의 정보 전달 매체와 비교하였을 때, 그 방대한 데이터 용량은 전송이나 저장 시 큰 문제점으로 지적되었다. 이를 해결하기 위한 방법으로 동영상 압축에 대한 관심과 필요도가 커지고 지난 20년간 활발히 연구가 진행되어 눈부신 발전을 하여 근래에는 동영상 압축은 필수적인 요소가 되었고, 더 나아가 스케일러블 코딩 등의 다양한 응용 기법들로의 발전으로 전송 및 저장 매체를 보다 효율적으로 사용할 수 있도록 해주는 역할도 하고 있다.

다양한 플랫폼 상에서 압축된 동영상 데이터를 효율적으로 사용하기 위해서는 호환성이 필요하였고, 이를 위해 동영상 압축의 표준안들이 등장하였다. 동영상 압축의 표준안은 ISO/IEC (international organization for standardization & international electrotechnical commission)의 MPEG (moving picture experts group)과 ITU-T (international telecommunication union - telecommunication standardization sector)의 VCEG (video coding experts group)를 중심으로 만들어 졌다. MPEG은 ISO/IEC의 멀티미디어 부호화 작업 그룹이며, 주로 CD-ROM과 DVD 같은 저장 멀티미디어의 규격을 표준화 하였다. MPEG에서 발표한 동영상 압축의 표준안들은 MPEG-x로 명칭이 붙여지며 MPEG-1, MPEG-2,

MPEG-4, MPEG-7, MPEG-21, MPEG-AVC 등이 있다.^[1-4] VCEG는 ITU-T 조직의 멀티 미디어 부호화 작업 그룹이며 화상전화와 화상회의 같은 비디오 통신과 관련된 표준화를 진행하였다. VCEG에서 발표한 동영상 압축 표준안들은 H.26x로 명칭이 붙여지며 H.261, H.262, H.263, H.264 등이 있다.^{[2][4-6]} 또한 ISO/IEC의 MPEG과 ITU-T의 VCEG가 같이 JVT (joint video team)라고하는 통합 그룹을 조직하여 각 그룹의 특징을 잘 반영하는 기존의 MPEG-x, H.26x 압축 표준안 보다 뛰어난 동영상 압축 성능을 가지는 새로운 표준안을 개발하기도 하였다. MPEG-2와 H.262에서 최초로 공동 작업을 하였으며, 최근 2003년 5월에 MPEG-4와 H.264에서 공동 발표한 것으로 MPEG-4 Part 10 Advanced Video Coding과 H.264라는 명칭이 각각 있으나 통칭 H.264/AVC라고 불리는 표준안을 발표하였다. H.264/AVC는 기존의 표준안인 MPEG-4와 H.263에 비해 동일 비트율에서 2배 이상의 압축 효율을 가지고 있는 획기적인 표준안이다. H.264/AVC는 부호화 효율의 개선을 위해 가변 블록 크기의 움직임 추정 및 보상 (variable block size motion estimation/compensation), 1/4 화소 정밀도의 움직임 추정 및 보상 (quarter pixel accurate motion estimation/compensation), 다중 참조 픽처 움직임 추정 및 보상 (multiple reference picture motion estimation/compensation), DCT 기반의 4x4 정수 변환 (4x4 integer transform based DCT), CABAC (context-adaptive binary arithmetic coding), 율-왜곡 최적화 (rate-distortion optimization), 인트라 예측 부호화 등의 새로운 기법들이 사용되었다.^[7-10]

H.264/AVC에서 새롭게 시도된 많은 기법들은 기존의 표준안들에 비해 부호화 성능을 개선하여 압축 효율을 상당히 높여주지만, 부호기의 복잡도가 극도로 높아져 실시간 구현이 어려운 단점이 있다. 이런 단점을 극복하고 실시간 구현을 가능하게 하기 위해 부호기의 복잡도를 감소시키는 고속

알고리즘에 대한 관심이 커지며 H.264/AVC의 발표 이후 지금까지 많은 고속 알고리즘들이 연구 및 발표 되어지고 있다.^[11-13]

본 논문에서는 H.264/AVC 부호기의 복잡도를 증가시키는 주요 원인 중 윌-왜곡 최적화와 인트라 예측 부호화에 관련된 계산량을 감소시키면서도 화질 저하는 미미한 효율적인 고속 알고리즘을 제안한다. 윌-왜곡 최적화는 QP 특성에 맞추어 가변 블록과 skip 모드, 인트라 모드에 대하여 모두 RD_cost를 계산하여 최소의 RD_cost를 가지는 모드를 해당 블록의 최적 모드로 결정하는 기법이다. 하지만, RD_cost를 계산하기 위하여는 왜곡 정보와 해당 모드로 압축하고 코딩했을 때 발생하는 비트량 즉, 윌 정보를 구해야 하므로 선택되지 않는 모드라 할지라도 모든 모드에 대해 코딩 과정까지 진행해야하며 이는 인코더의 계산량을 증가시키는 요인이 된다. 그리고 다중 참조 픽처를 적용할 때 최적의 참조 픽처를 선택하기 위해서 사용되는 MV_cost는 왜곡 정보와 예측 모션 벡터와 실제 모션 벡터간의 차이 정보 등을 이용하여 계산되고 이는 최적의 블록 모드를 선택하기 위해 사용되는 RD_cost와 유사한 경향을 가질 가능성이 크다. 그러므로 본 논문에서는 MV_cost를 사용하여 미리 RD_cost값을 예측하고 이를 근거로 윌-왜곡 최적화를 생략하는 알고리즘을 제안한다. 그리고 H.264/AVC에서는 P-picture에서도 인트라 예측 모드를 적용하는데, 이 때 최적으로 선택되는 인트라 예측 모드는 평균적으로 3%정도이다. 하지만 이 3%를 위하여 모든 매크로 블록마다 인트라 예측 부호화를 시행하고 윌-왜곡 최적화를 거쳐 최종 모드 결정을 하고 있다. 인트라 예측 부호화는 P-picture에서 QP의 특성을 더욱 높여주는 역할을 하는데, 낮은 QP에서는 비트량 감소의 역할을 하며, 높은 QP에서는 화질적 보완을 해준다. 하지만, 인트라 예측을 수행하는데 사용되는 계산량이 많아 인코더의 복잡도를 증가시키므로 인트라 예측 모드가 최적으로 선택되는 조건을 영상 복잡도를 이용하여 판

별하고 이전에 선택된 최적의 가변블록과 예측 모션 벡터에 의해서 인트라 예측의 모드도 선택적으로 수행하여 복잡도를 줄여주는 알고리즘을 제안한다.

모의 실험 결과 QP에 따라 차이는 생기지만, 전자의 알고리즘은 미미한 화질 저하는 생기지만 코딩 시간을 평균 5%감소 시켰으며, 후자의 알고리즘 역시 미미한 화질 저하가 생기지만 평균 35%의 코딩 시간을 감소시켰다. 그리고 두 알고리즘을 동시에 사용하였을 때는 평균 40%이상의 코딩 시간을 감소시켰다.

본 논문은 서론에 이어 2장에서는 H.264/AVC에서 압축 효율을 높이기 위해 도입된 움직임 추정 및 보상 기술에 대해 기술하고, 3장에서는 새롭게 제안하는 적응적 율-왜곡 최적화 알고리즘과, 제한적 인트라 예측 부호화 알고리즘에 대한 내용을 기술하며 4장에서는 모의 실험을 통해 기존의 방법과 제안하는 알고리즘의 성능을 비교 분석하고 5장에서 결론을 맺도록 구성된다.

II. H.264/AVC의 새로운 움직임 추정 및 보상 기술

본 장에서는 부호화 효율을 개선하기 위해 이전 표준안의 움직임 추정 및 보상 기술과는 다른 H.264/AVC에서 새롭게 적용시킨 움직임 추정 및 보상 기술들에 대해 기술한다.

1. 가변 블록 움직임 추정 및 보상

기존의 표준안인 MPEG-2에서는 16x16 크기로 고정된 블록을 사용하였고, MPEG-4에서는 16x16 크기와 8x8 크기의 두 종류의 블록을 사용하였다. 이처럼 기존의 표준안에서는 블록 크기는 거의 고정되어 있었으나 이와 달리 H.264/AVC에서는 그림 1과 같이 16x16 크기의 블록으로부터 4x4 크기의 블록까지 7가지 크기의 다양한 블록을 제공하고 있다.

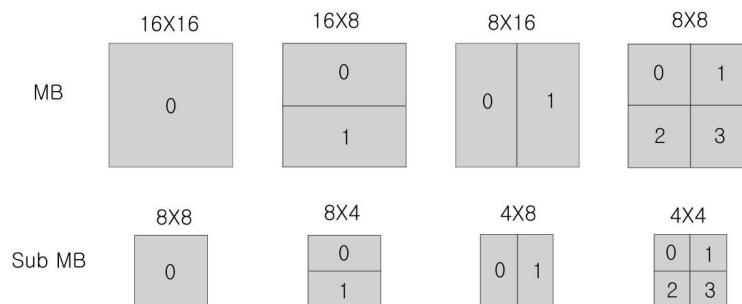


그림 1. H.265/AVC의 가변 블록들의 크기

위의 그림에서 MB는 매크로 블록을 의미하며 MB는 16x16, 16x8, 8x16, 8x8 크기로 나뉘고 그 중 가장 작은 8x8 크기의 블록은 다시 8x4, 4x8, 4x4 크기의 sub MB로 나뉘게 된다. 부호화 순서는 H.264/AVC의 참조 소프트웨어인 JM에서는 16x16, 16x8, 8x16 크기의 블록에 대해 움직임 추정 및 보상을 하고, 8x8, 8x4, 4x8, 4x4에 대해서는 8x8 크기 이하로 결정되었을 경우 즉, 8x8, 8x4, 4x8, 4x4 크기의 블록들만으로 이루어진 결과를 도출한 뒤, 앞에서 미리 움직임 추정 및 보상을 한 3가지 크기의 블록에 대한 결과와 비교하여 최종적인 결과를 선택하게 된다. 이에 대해서는 본문에 자세히 언급되어있다. 또한 움직임 추정 및 보상에 필요한 참조 픽처 정보를 나타내는 참조 픽처 인덱스는 16x16, 16x8, 8x16 크기의 블록일 때는 각 블록마다 부호화하고, 8x8 크기 이하의 블록들일 경우에는 참조 픽처 인덱스를 8x8 블록 단위로 부호화해서 움직임 보상에 따라 정보량 증가를 방지한다.

색차 성분에 대한 움직임 추정 및 보상은 휘도 성분의 1/2 서브 샘플링된 크기의 해상도를 가지고 있을 뿐 휘도 성분과 동일한 방법으로 분할되고 같은 모드를 적용시켜 부호화하게 된다. 즉, 휘도 성분의 최적 모드가 8x16 모드로 선택되었다면 색차 성분의 최적 모드 역시 같은 형태인 4x8 크기로 적용하여 사용하게 된다.

2. 1/4화소 정밀도의 움직임 추정 및 보상

움직임 추정 및 보상에 사용되는 화소 정밀도는 높으면 높을수록 세밀한 움직임 벡터를 구하여 정밀한 예측을 할 수 있으나 그에 따른 추가적인 계산량의 추가로 복잡도는 증가하게 된다. 기존의 MPEG-2, MPEG-4, H.263에서는 움직임 추정 및 보상 시 1/2화소 정밀도를 사용하였다. 그러나 MPEG-4의 advanced simple profile, advanced coding efficiency profile에서 1/4화소 정밀도의 움직임 추정 및 보상을 하도록 채택되었으며 이는 H.264/AVC에까지 이어지게 된다. H.264/AVC에서는 정수 화소가 아닌 화소를 생성하기 위하여 6-tap FIR(finite impulse response) 필터와 2-tap 평균 필터에 의해 고주파 성분까지 재현할 수 있는 보다 향상된 1/4 화소 정확도의 화소들을 생성하여 움직임 추정 및 보상에 사용한다.

움직임 추정 및 보상 시 정수 화소 이하의 화소는 화소들 간의 보간(interpolation)을 통해 생성한다. H.264/AVC에서 정수 화소 이하의 화소를 생성하는 방법은 다음과 같으며 그림 2를 참고로 하여 설명한다. 먼저 정수 화소 값들에 대해 6-tap FIR 필터를 사용하여 1/2화소 정밀도의 화소를 생성한다. 6-tap FIR 필터는 매 화소 위치마다 6개 입력 화소값에 각각의 계수를 곱한 후 총합을 출력하는 필터로써 계수는 $1/32, -5/32, 20/32, 20/32, -5/32, 1/32$ 이다. 이 때 두 개의 정수 화소 사이에 수평방향으로 위치한 1/2 화소는 수평 방향으로 필터를 사용하고, 수직 방향으로 위치한 1/2 화소는 수직 방향으로 필터링 한다. 예를 들어, 그림 4에서 1/2화소 cc는 E, F, G, H, I, J로부터 계산되며, 또 다른 1/2 화소인 ii는 A, C, G, K, M, O를 필터링하여 보간 된다. cc와 ii에 대한 필터링 계산은 식 (1)과 같이 한다.

$$\begin{aligned} cc &= \text{round}((E - 5F + 20G + 20H - 5I + j)/32) \\ ii &= \text{round}((A - 5C + 20G + 20K - 5M + O)/32) \end{aligned} \quad (1)$$

위의 식에서 round는 반올림을 의미하고, 네 개의 정수 화소값 사이에 위치한 1/2 화소값들 즉, jj와 같은 경우는 6-tap FIR 필터를 수평 또는 수직 방향에 적용하여 계산되며, 어떤 방향으로 적용하든지 결과는 같다.

1/2화소 정밀도의 화소값들을 생성한 후, 1/4 화소 정밀도의 화소값들을 생성된다. 수평 또는 수직 방향으로 인접한 두 개의 1/2 화소값과 정수 화소값 사이에 위치한 1/4 화소 정밀도의 화소는 두 화소값을 선형 보간 즉, 평균값을 이용하여 생성된다. 예를 들어 그림 2에서 두 화소 사이에 수평 방향으로 위치한 1/4 화소 a는 G와 cc를 선형 보간 하여 생성되며, 수직 방향으로 위치한 1/4 화소 c는 cc와 jj를 선형 보간 하여 생성되고 각각의 계산식은 식 (2)과 같다.

$$\begin{aligned} a &= \text{round}((G + cc)/2) \\ c &= \text{round}((cc + jj)/2) \end{aligned} \quad (2)$$

나머지 1/4 화소 정밀도의 화소는 대각선 방향의 두 개의 1/2 화소 정밀도의 화소를 선형 보간 하여 생성된다. 예를 들어 그림 2에서 화소 G, cc, ii, jj는 대각선 방향으로 선형 보간 하여 계산되며, 화소 b는 식 (3)과 같이 계산된다.

$$b = \text{round}((ii + cc)/2) \quad (3)$$

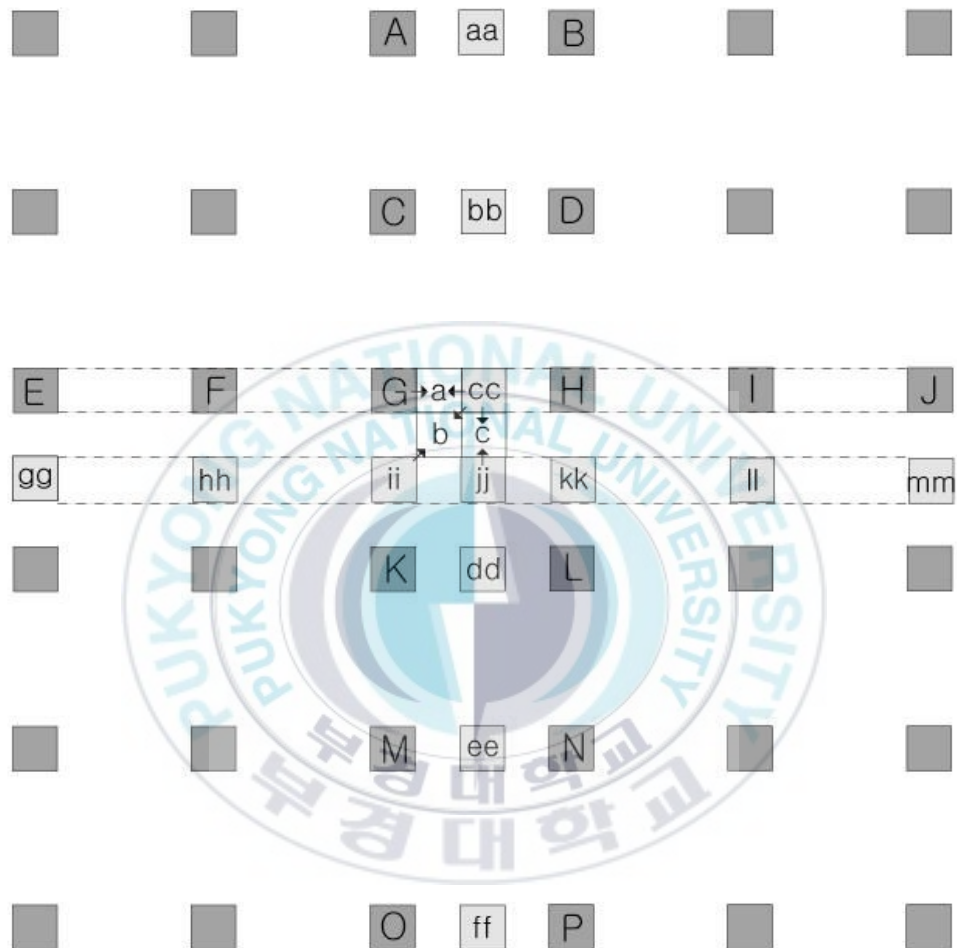


그림 2. 1/4 화소 정밀도의 화소값 생성 방법

3. 다중 참조 픽처 움직임 추정 및 보상

H.264/AVC 이전의 표준안인 MPEG-4, MPEG-2, H.263 등의 표준안은 현재 P 픽처의 값을 예측하기 위해 이전에 위치한 하나의 I-picture 또는 P-picture만을 사용하여 움직임 보상을 한다. H.263++에서 기존 표준안보다 강화된 참조 픽처 선택 방법이 정식으로 채택되었으며, H.264/AVC에서는 H.263++의 구문을 이어받아 최소 8x8 화소 단위의 다중 참조 픽처 선택이 가능해졌다. H.263/AVC는 다수의 참조 픽처를 사용하여 기존의 표준안들보다 더욱 정밀하게 최적의 블록을 선택할 수 있게 되어 더 높은 효율을 가지는 움직임 추정 및 보상을 구현할 수 있다. 아래 그림 3에 기존의 단일 참조 픽처의 움직임 추정 및 보상과 H.264/AVC의 다중 참조 픽처의 움직임 추정 및 보상에 대해 나타내었다.

이러한 다중 참조 픽처를 사용하게 되면 움직임이 있는 물체들이 다른 물체에 의해 가려졌다가 나타나는 현상이나 회전에 의해 물체의 모양이 변하는 현상, 반복적인 영상에 대한 움직임 추정 및 보상을 더욱 효율적으로 할 수 있게 된다. 실제로 표준 영상들 중 아주 복잡하고 회전등의 움직임이 존재하는 Mobile, Tempete와 같은 영상들은 다중 참조 픽처를 사용하여 움직임 추정 및 보상을 한 결과 PSNR (peak signal to noise rate)이 최고 1 dB 증가한 것으로 알려졌다.

앞서 언급했듯이 H.264/AVC에서는 참조할 픽처를 선택하기 위해 움직임 보상 블록마다 참조 픽처 번호 (ref_idx)를 할당하지만 비트량의 증가를 막기 위해 블록 크기가 8x8 이하라면 8x8 크기 블록 단위로 참조 픽처를 할당한다. H.264/AVC에서 최대 참조할 수 있는 참조 픽처 수는 프로파일과 레벨에 따라 정의되어 있다.

하지만 다중 참조 픽처를 사용한 움직임 추정 및 보상은 H.264/AVC 부호기의 복잡도를 증가시키는 주요 요인 중 하나로 작용하며, 움직임 추정 및 보상 시간은 참조 픽처의 수에 비례하여 증가하게 된다.

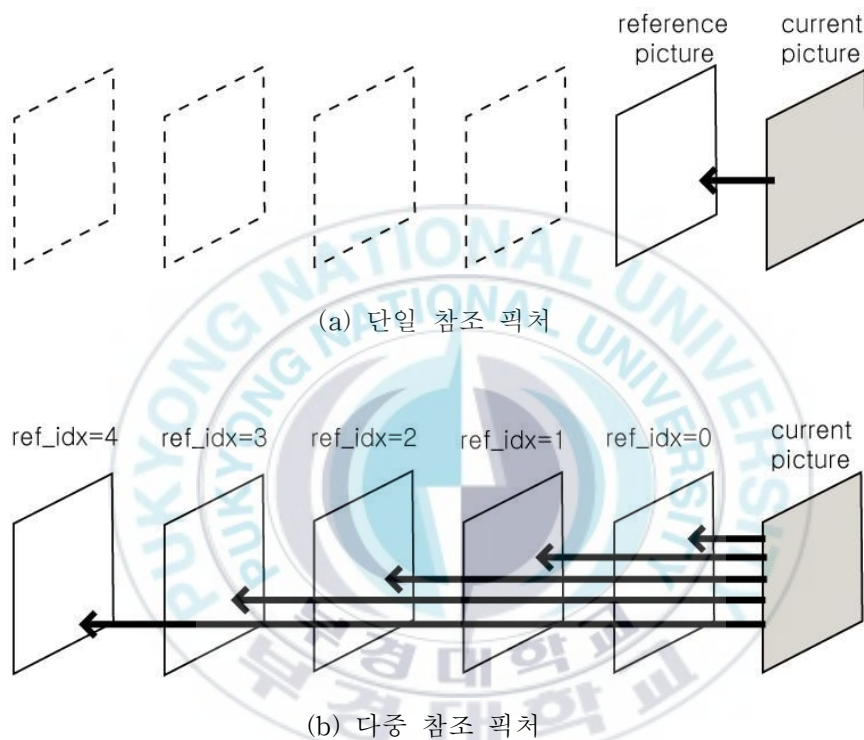


그림 3. 단일 참조 픽처와 다중 참조 픽처 방식

4. 율-왜곡 최적화

율-왜곡 최적화는 H.264/AVC에서 사용되고 있는 다양한 블록 모드에서 최적 블록 모드를 선택할 수 있도록 해주는 기법이다. H.264/AVC는 최적 블록 모드를 선택하기 위해 두 단계 처리를 수행하고 있다. 첫 번째 단계는 다중 참조 픽처를 적용하여 움직임 추정 및 보상을 할 때 최적의 인터 모드를 선택하는데 사용되며, 두 번째 단계에서는 매크로 블록의 최적 모드를 인터 모드들과 스킵 모드, 인트라 예측 모드 중에서 선택할 수 있도록 해준다. 율 왜곡 최적화는 최적의 인터 모드를 선택할 때는 MV_cost 라는 비용함수를 사용하고 최적의 블록 모드를 선택할 때는 RD_cost 라는 비용 함수를 사용한다. 각각의 비용함수는 아래 식 4에 의해 계산되어 진다.

$$\begin{aligned} MVcost &= SAD + \lambda_{Motion} \times (mvbits[MV - PMV]) \\ RDcost &= SSD + \lambda_{Mode} [R(Header) + R(residual data)] \end{aligned} \quad (4)$$

위의 식에서 SAD (sum of absolute difference)는 현재 블록과 후보 블록의 밝기값 차이들의 총합의 절대값이며, SSD (sum of square difference)는 현재 블록과 복원된 영상의 밝기값 차이의 제곱의 합을 의미하고 둘 값들 모두 정합 오차로서 왜곡 정보를 의미한다. $R(x)$ 는 “x”를 부호화할 때 발생하는 비트량을 의미한다. $R(Header)$ 에는 매크로 블록의 최적 모드와 움직임 벡터, CBP (coded block pattern)등의 정보를 의미하고 $R(residual data)$ 는 그 외의 잉여 정보로써 직교변환과 양자화 처리 후의 변환계수부분의 비트 등을 의미한다. 단 CABAC에 의한 발생비트를 예상하는 것은 매우 많은 연산량을 필요로 하기 때문에, 엔트로피 부호화 방식으로

CABAC을 사용하는 경우에도 모드를 판정할 때는 VLC(가변길이 부호)를 사용한다. 즉, 모든 후보모드에 대해서, 부호화/복호처리가 요구되므로 다중 경로 부호화가 필요하다. 그리고 λ_{Motion} 과 λ_{MODE} 는 *lagrangian* 승수로써 각각 식 (5), (6)과 같이 계산되어 진다.

$$\begin{aligned}\lambda_{Motion.P} &= \sqrt{0.85 \times 2^{(QP-12)/3}} \\ \lambda_{Motion.B} &= \sqrt{4 \times 0.85 \times 2^{(QP-12)/3}}\end{aligned}\quad (5)$$

$$\begin{aligned}\lambda_{Mode.I.P} &= 0.85 \times 2^{(QP-12)/3} \\ \lambda_{Mode.B} &= 4 \times 0.85 \times 2^{(QP-12)/3}\end{aligned}\quad (6)$$

QP (quantization parameter)는 양자화 파라미터를 의미하며 이는 양자화 스텝 크기를 결정하는 값이다. QP가 커질수록 *lagrangian* 승수가 지수적으로 커지고 이는 비용 함수들의 율 정보에 가중치로 적용되어 화질보다 발생 비트량이 작은 것을 최적의 결과로 선택되게 한다. *lagrangian* 승수는 그림 4와 같이 율-왜곡선에서의 접선의 기울기를 나타낸다.

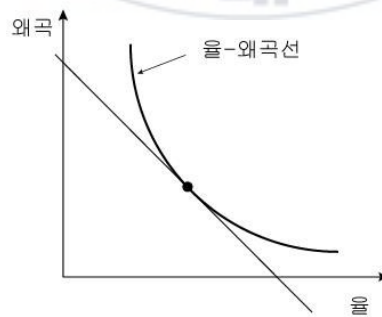


그림 4. 율-왜곡 곡선

5. 인트라 예측 부호화

5.1. 휘도 성분 4x4 인트라 예측 부호화

4x4 크기의 블록 단위의 인트라 예측 부호화는 그림 5와 같이 방향성을 지닌 9가지 예측 모드를 가지고 있으며 4x4 크기의 블록마다 최적 모드 1개씩 선택하고, 선택된 예측방향(예측모드)을 4x4화소의 블록단위로 부호화한다. 예측모드 2가 의미하는 평균값은 좌측블록의 4화소와 상단블록의 4화소, 즉 총 8화소의 평균값을 구해 대상블록의 4x4화소 모두를 예측하는 방법이다.

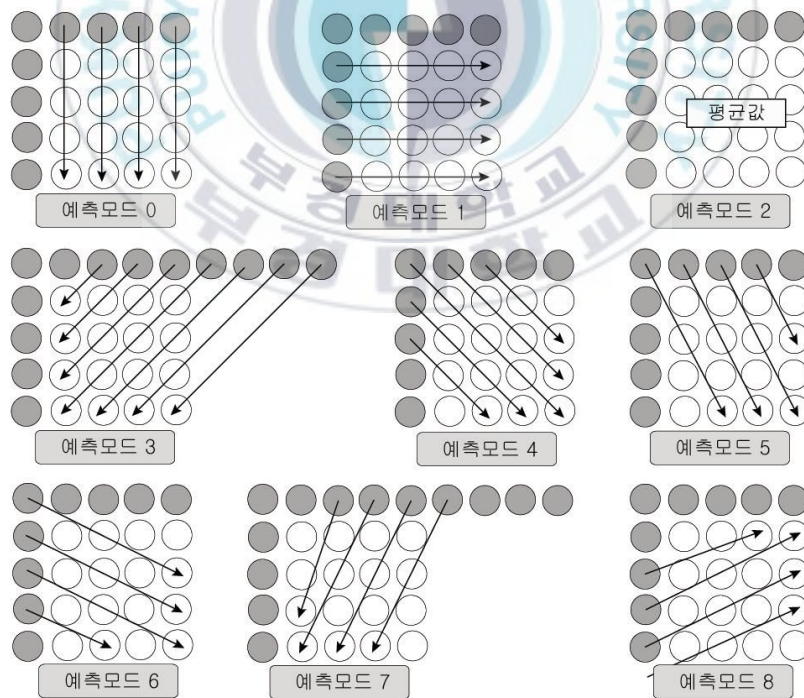


그림 5. 4x4 크기의 인트라 예측 모드

예측 모드의 번호는 가장 발생 확률이 높을수록 앞쪽으로 배치하여 결정된 것이며, mode 3~8은 방향성 가중 평균 (directional weighted average) 방식으로 예측을 수행하게 된다. 그리고 픽처의 좌측경계에 위치하는 블록과 픽처의 상단 경계에 위치하는 블록의 경우 좌측 블록과 상단 블록이 각각 픽처 바깥쪽에 위치하게 된다. 이 경우 픽처 경계를 벗어난 블록을 참조할 수 없기 때문에 예측 방향의 사용이 제한된다. 예외적으로 모드 2 평균치 예측의 경우, 픽처의 경계를 벗어나지 않는 화소만을 참조해서 평균치를 계산하는 예측을 수행한다. 즉, 픽처의 최상단에 위치하는 블록에서는 좌측 블록 4화소 평균치를 예측값으로 하고, 픽처의 좌측경계에 위치하는 블록에서는 상단블록 4화소의 평균치를 예측값으로 한다.

5.2. 휘도 성분 16x16 인트라 예측 부호화

16x16 크기의 인트라 예측 부호화는 상단 매크로 블록의 16화소와 좌측 매크로 블록의 16화소로부터 그림 6과 같이 수직예측(모드0), 수평예측(모드1), 평균치예측(모드2), 평균예측(모드3)의 4 가지 예측 방법을 제공한다. 16x16 크기의 인트라 예측 모드들도 발생 확률이 높을수록 낮은 모드 번호를 가지고 있다. 모드 3인 평면예측의 경우 방향성 가중 평균 방식에 의해 부호화한다. 모드 2에 해당하는 평균치예측의 경우 평균치는 상단블록의 16화소와 좌측블록의 16화소의 평균값을 의미하는데, 최상단 및 좌측 블록과 같은 예외적인 처리는 4x4 크기의 인트라 예측 부호화와 같은 방법으로 대처한다. 16x16 크기의 인트라 예측의 경우 4x4 크기의 인트라 예측 부호화에 비해 필요한 비트수가 적기 때문에 파란하늘과 같이 평탄한 영상부분에서는 16x16화소단위를 사용하더라도 좋은 예측을 할 수 있고

압축률을 크게 향상시킬 수 있다.

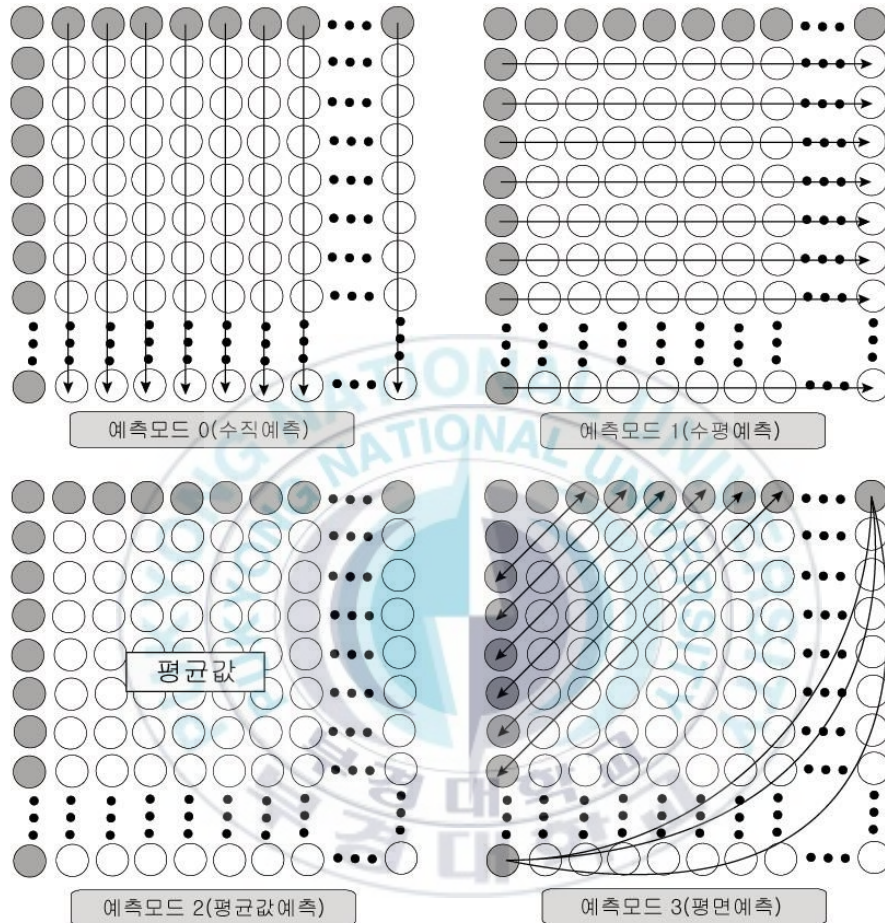


그림 6. 16x16 크기의 인트라 예측 모드

5.3. 색차 성분 8x8 인트라 예측 부호화

색차 성분에 대한 인트라 예측 부호화는 휘도 성분의 인트라 예측 부호화의 결과와는 독립적으로 이루어지며 8x8 크기에 대해서만 수행되고

16x16크기의 휘도 성분과 똑같은 방식의 4가지 모드를 제공하지만, 발생 확률이 달라져 모드 번호에 따른 예측 방법은 다르다. 8x8 크기의 색차 성분 인트라 예측 모드는 예측 모드 0이 평균값 예측, 예측 모드 1이 수평 예측, 모드 2가 수직 예측, 모드 3이 평면 예측 순으로 정해져있다.



III. 제안하는 알고리즘

1. 적응적 율-왜곡 최적화 알고리즘

율-왜곡 최적화 기법은 해당 매크로 블록의 최적 참조 픽처 또는 최적 모드를 QP에 따른 특성에 맞게 MV_cost와 RD_cost라는 비용 함수를 이용하여 선택하는 기법이다. 이 때 QP에 따른 특성은 낮은 QP일수록 적은 비트량의 모드보다는 왜곡이 적은 모드들이 선택되어 화질이 우수하며, 높은 QP일수록 왜곡이 적은 모드보다는 비트량이 적은 모드를 선택하게 된다. 하지만, 율-왜곡 최적화 기법 중 모드 선택을 위한 RD_cost를 계산하기 위해서는 모든 후보 모드 즉, 선택되지 않을 모드들에 대해서도 전송 직전의 단계까지 인코더 과정을 진행시킨 뒤 발생하는 비트량을 계산해야 하는 복잡한 과정을 매번 거쳐야만 한다. 물론 율-왜곡 최적화 기법을 사용할 때 발생하는 복잡도가 가변 블록 단위의 움직임 추정 및 보상이나 인트라 예측 모드를 수행하는 과정에서 발생하는 방대한 복잡도에 비해서는 많다고 할 수 없지만, 인코더 전체의 복잡도를 증가시키는 요인 중 하나로 작용한다. H.264/AVC에서는 율-왜곡 최적화 기법의 사용 여부를 선택할 수 있다. 율-왜곡 최적화 기법을 사용할 경우는 고화질 모드 방식라고 명하며 앞서 언급한 것과 같이 RD_cost를 이용하며, 율-왜곡 최적화 기법을 사용하지 않을 경우는 고속 모드 방식 (Low Complexity Mode)라고 명하며 아래의 식 7에 의해 계산되는 비용함수로 최적의 모드를 결정하게 된다.

$$Cost(Mode) = SAD + SAD_0 \quad (7)$$

SAD₀는 양자화 파라미터를 이용한 양자화 스케일의 변화값과 헤더정보의 비트를 사용하여 구하게 된다.

표 1. Low Complexity Mode (RDO-off)

size	sequence	QP	Δ Total time [sec]	Δ SNR Y [dB]	Δ bit rate [kbits/s]
QCIF	container	28	-47.90%	0.07	11.30%
		32	-45.68%	0.05	8.62%
		36	-43.89%	0.13	9.53%
		40	-42.74%	0.22	7.83%
	news	28	-48.87%	0.14	7.46%
		32	-46.71%	0.05	6.06%
		36	-44.87%	0	5.51%
		40	-43.37%	0.04	6.48%
	foreman	28	-47.54%	0.12	7.96%
		32	-45.04%	0.13	6.84%
		36	-43.02%	0.17	7.15%
		40	-41.65%	0.18	10.24%
	scilent voice	28	-48.37%	0.18	6.53%
		32	-45.58%	0.02	5.86%
		36	-43.43%	0.04	5.29%
		40	-41.88%	-0.01	5.16%
CIF	paris	28	-27.25%	0.16	5.94%
		32	-24.89%	0.08	5.65%
		36	-23.25%	0.06	7.20%
		40	-22.03%	0.13	10.14%
	mobile	28	-30.35%	0.04	4.79%
		32	-27.54%	0.1	7.16%
		36	-25.06%	0.19	12.09%
		40	-22.93%	0.36	19.89%
	tempete	28	-27.52%	0.03	7.01%
		32	-24.84%	0.06	8.58%

		36	-23.13%	0.16	11.21%
		40	-21.57%	0.25	15.04%

표 1에서 $\Delta Total\ time$, $\Delta SNR\ Y$, $\Delta bit\ rate$ 는 각각 전체 코딩 시간, 휘도 성분의 SNR, 비트율에 대하여 오프-왜곡 최적화를 사용하였을 경우를 기준으로 오프-왜곡 최적화를 사용하지 않았을 경우와의 차이를 나타내는 값들이며 아래의 식 8, 9, 10에 의해서 결정된다.

$$\Delta Totaltime = \left(\frac{Totaltime_{RDO-off}}{Totaltime_{RDO-on}} - 1 \right) \times 100 \quad [\%] \quad (8)$$

$$\Delta SNR\ Y = SNR\ Y_{RDO-off} - SNR\ Y_{RDO-on} \quad [dB] \quad (9)$$

$$\Delta bit\ rate = \left(\frac{bit\ rate_{RDO-off}}{bit\ rate_{RDO-on}} - 1 \right) \times 100 \quad [\%] \quad (10)$$

표 1에서 오프-왜곡 최적화를 적용하지 않았을 경우 전체적으로 모든 QP에 대해 코딩 시간은 QCIF의 경우 평균 45% 이상 감소하고 화질은 평균 0.07dB 증가하며 비트량도 평균 7% 이상 증가하였다. 하지만, 화질과 비트량이 QP에 따른 경향 즉, 높은 QP에서는 화질과 비트량의 감소가 비교적 커야하며, 낮은 QP에서는 화질과 비트량의 감소가 적어야 하지만 그러한 특성을 따르지 않는다는 것을 알 수 있다. 이러한 경향은 CIF에서도 마찬가지이다. 그리고 CIF의 결과에서 전체 코딩시간 감소량이 QCIF에 비해 절반 수준인 20~30% 정도만 감소되는 이유는 영상 크기가 커진 것에 대한 움직임 추정 및 보상과 오프-왜곡 특성의 계산량은 비례적으로 증가하게 되지만, 영상의 크기가 커짐에 따라 탐색 영역도 커지므로 움직임 추정 및 보상의 계산량이 더욱 더 증가하여 전체 코딩 시간에서 오프-왜곡 최적화로 인한 복잡도의 비율이 줄어들었기 때문이다. 본 장에서는 고속 오프-왜곡 최

적화를 제안하여 QP에 따른 압축 특성을 따르면서도 율-왜곡 최적화 기법으로 인해 발생한 인코더의 복잡도를 감소시키는 방법을 제안한다.

1.1. RD_cost와 MV_cost관계

H.264/AVC에서는 움직임 추정 및 보상 과정 중 이전 시점의 픽처 한장이 아닌 다수의 참조 픽처를 사용하며, 그 중에서 한가지의 결과를 선택하는 기법을 사용하는데, 이를 다중 참조 픽처 기법이라고 한다. 이 기법으로 인해 모든 가변 블록 모드마다 참조 픽처의 수만큼의 움직임 추정 결과를 얻게 되고, 이 결과들 중 최적의 결과를 결정하기 위해 율-왜곡 최적화가 사용되며 MV_cost라는 비용함수를 사용한다. 이 때 사용되는 MV_cost는 결국 가변 블록 모드 별로 최적의 참조 픽처를 결정하게 된다. MV_cost의 파라미터는 왜곡 정보와 함께 모션 벡터관련 정보, *lagrangian* 승수가 사용된다. 이 때 왜곡 정보는 동일한 가변 블록 모드 내에서 여러장의 참조 픽처를 이용한 결과 중 최적을 선택하기 때문에 정규화 되지 않은 왜곡 정보를 사용한다. 그리고 모션 벡터 관련 정보는 예측 모션 벡터와 실제 모션 벡터의 차이를 비용 함수 파라미터로 사용한다. 반면 RD_cost는 MV_cost를 통하여 결정된 각 가변 블록의 모드들과 가변 블록 이외의 후보 모드들 중 해당 매크로 블록의 최종적인 최적 모드를 선택하는 비용 함수이다. RD_cost의 파라미터로는 왜곡 정보와 켄 비트, *lagrangian* 승수가 사용된다. RD_cost는 모든 가변 블록들 간의 결과 중 최적의 모드를 결정하기 때문에 왜곡 정보에 대한 정규화가 이루어지고 있다. 켄 비트의 데이터에는 해당 블록의 모드와 참조 픽처 정보, 움직임 벡터관련 정보, 프레임 정보 등이 있으나, 움직임 추정에 의한 결과 즉, 모드의 종류와 움직임 벡터가 큰 의미를 가지며 모드에 따라서 움직임 벡터의

수 가 달라지는 등 발생하는 비트량에 큰 영향을 줄 가능성이 크다. 결국 MV_cost와 RD_cost에 사용되는 왜곡 정보는 정규화 유무에 따른 차이가 존재하며, 비트량에 관련된 정보는 두 비용함수 모두 움직임 벡터와 관련된 정보를 가지고 있으므로 비슷한 경향을 가질 가능성이 크다. 그러므로 MV_cost와 RD_cost는 서로 비슷한 경향을 가지는 파라미터를 사용하는 비용 함수들이라고 할 수 있으며, 상호간의 밀접한 상관관계를 가지고 있다고 할 수 있다.

1.2. 제안하는 알고리즘

앞서 언급했듯이 RD_cost와 MV_cost가 밀접한 상관관계를 가지고 있다면, MV_cost를 이용하여 사전에 미리 RD_cost를 예측할 수 있다면 이 예측된 값을 근거로 하여 율-왜곡 최적화 기법을 적응적으로 할 수 있으며 그 결과는 율-왜곡 최적화로 인해 증가했던 인코더의 복잡도를 줄일 수 있을 것이다. 아래 표 2에 QP 32에서 QCIF크기의 foreman영상에 대하여 조사한 MV_cost와 RD_cost, 그리고 해당 블록의 최적모드 즉, Best_mode를 조사하여 일부분을 표시하였다. 이 때 mode 1은 16x16 크기의 가변 블록의 결과이며 mode2 ,3은 각각 16x8, 8x16 크기의 가변 블록의 결과이고 mode 8은 8x8이하 크기의 가변 블록 모드들로 결정된 모드들의 집합체이다. H.264/AVC의 참조 소프트웨어인 JM에서는 크기가 큰 가변 블록인 16x16, 16x8, 8x16 크기의 모드에 대해 움직임 추정 및 보상을 하고 크기가 작은 가변 블록인 8x8, 8x4, 4x8, 4x4 크기의 모드에 대해서는 움직임 추정 및 보상과 동시에 율-왜곡 최적화 기법을 미리 사용하여 8x8 이하의 크기로 구성된 최적의 모드들을 미리 결정하고 이를 mode 8로 정하고는 최종 율-왜곡 최적화 기법을 적용시킬 때 크기가 큰 가변 모드와 인트라

예측 모드, 스킵 모드들과 함께 비교하는 루프로 동작하게 프로그래밍 되어있으므로 결과 비교는 mode 1, 2, 3, 8을 비교하였다.

표 2. 모드 1~3, 8에 대한 MV_cost와 RD_cost, Best_mode

mode 1		mode 2		mode 3		mode 8		best mode
MV cost	RD cost	MV cost	RD cost	MV cost	RD cost	MV cost	RD cost	
887	3162.546	603	3496.965	462	3746.965	411	4777.220	1
527	1057.418	317	1402.837	277	1402.837	237	2266.383	1
1734	9339.418	1209	9684.837	791	8944.255	600	9439.220	3
1679	7716.418	778	7192.255	1006	8061.837	543	8228.802	2
1484	5710.546	758	6376.674	595	6327.383	373	7494.639	1
2201	10057.128	988	9539.255	1158	9249.255	730	10744.220	3
2293	10716.092	1259	11451.766	1111	11740.220	693	11650.639	1
1325	8001.674	493	6956.929	700	10139.348	315	7865.057	2
1412	7882.965	812	7777.092	686	7029.383	325	7272.220	3
1689	7728.128	855	9006.802	1065	8883.965	603	9492.894	1
2468	11940.128	1125	12466.255	1200	11623.255	680	12983.220	3
1697	11338.128	845	11654.766	473	12469.383	285	12635.731	1
3597	22933.979	1832	23387.079	1593	22763.277	709	23522.724	3
1804	11385.128	536	12936.092	819	12651.348	281	12671.440	1
2107	12567.731	1086	12433.674	724	13333.929	522	12404.639	8
3604	22928.547	2127	25406.710	2315	25725.356	1313	23122.129	1
5094	34035.866	2353	34446.313	2142	33966.030	1099	32990.576	8
4448	31358.703	1968	28589.072	2014	30269.873	1067	29578.576	2

위의 표 2에서 살펴보면 각 가변 블록 모드의 MV_cost를 블록 크기에 따른 정규화를 하게 되면 그 값들의 경향이 RD_cost의 경향과 같아지는 것을 알 수 있다. 예를 들면 표에서 가장 윗줄의 데이터를 보면 RD_cost가 mode 1이 3162.546으로 가장 작아 최적의 모드로 선택된 것을 알 수 있다. 반면에 각 모드들의 MV_cost를 크기에 따른 정규화를 해보면 모드 별로 각각 887, 1206, 924, 1644라는 값이 나오고, 이때도 mode 1의 값이 가장 작다는 것을 알 수 있다. 앞서 언급했듯이 MV_cost와 RD_cost사이에는 비트량 관련 정보는 같은 경향일 확률이 크나, 왜곡 정보의 정규화 유무에 따른 차이점이 있었다. 하지만 이 차이는 표 2에서 보듯이 MV_cost의 정

규화 작업을 통해 최소화할 수 있으며 이 값은 RD_cost의 경향을 예측한 값이 된다. 그리고 그 결과에 따라 적응적으로 율-왜곡 최적화를 수행할 수 있다. 본 장에서는 정규화된 MV_cost를 이용한 적응적 율-왜곡 최적화 알고리즘을 제안한다.

제안하는 알고리즘은 참조 소프트웨어인 JM의 루프에 맞추어 적용된다. 아래 그림 7에 JM의 움직임 추정 및 보상 루프를 블록도로 나타내었다.

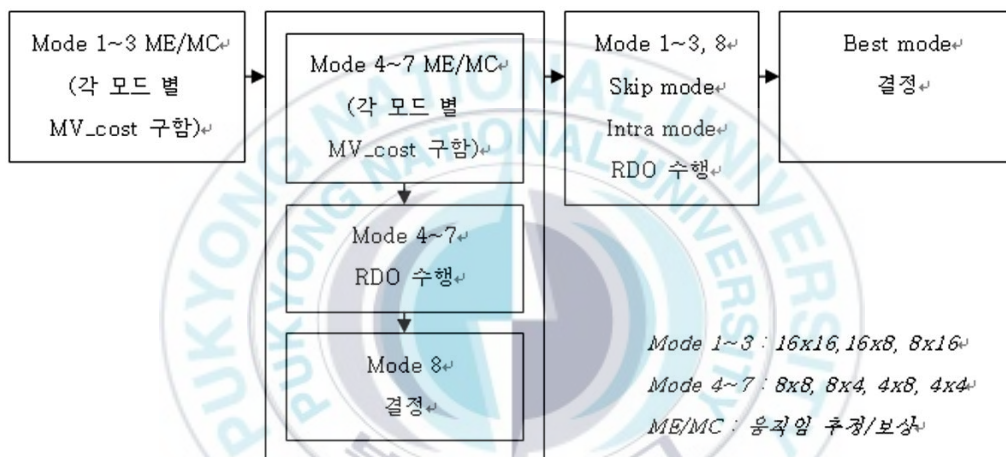


그림 7. JM에서의 움직임 추정 루프

즉, 최종적인 율-왜곡 최적화 이전에 작은 크기의 가변 블록들로만 율-왜곡 최적화를 수행하여 mode 8을 정하고 나머지 후보 모드들과 함께 율-왜곡 최적화를 다시 수행하게 된다. 그러므로 제안하는 알고리즘도 1차, 2차 정규화와 1차, 2차, 적응적 율-왜곡 최적화 기법으로 나누어 적용하게 된다. 제안하는 알고리즘의 우선 16x16, 16x8, 8x16 크기의 가변 블록에 대해서 움직임 추정 및 보상을 수행하면서 각각에 대한 MV_cost를 구하고 이를 각각 MV_cost₁₋₃으로 명한다. 그 후 8x8, 8x4, 4x8, 4x4 크기의 가변 블록에 대하여 움직임 추정 및 보상을 수행하면서 각각에 대한 MV_cost

인 MV_cost_{4-7} 을 구한다. 그 후 JM의 루프에 따라 MV_cost_{4-7} 들 간의 1차 정규화를 수행하고 이를 RMV_cost_{4-7} 로 명명하고 그 값을 근거로 1차 고속 유효-왜곡 최적화를 수행한다. MV_cost_{4-7} 의 정규화는 아래의 식에 의해 계산된다.

$$RMV_cost_{4-7} = MV_cost_{4-7} \times \frac{\text{number of pixels mode4}}{\text{number of pixels mode4} \sim 7} \quad (11)$$

8x8 크기의 가변 블록은 유효-왜곡 최적화를 수행함과 동시에 RMV_cost_4 를 미리 구하고 다른 가변 블록들은 유효-왜곡 최적화 이전에 RMV_cost_4 와 RMV_cost_{5-7} 을 비교하고 RMV_cost_4 보다 큰 경우에는 해당 가변 블록의 RD_cost 도 8x8 크기의 가변 블록의 RD_cost 보다 클 것으로 판단하여 유효-왜곡 최적화를 생략한다. 예를 들어 RMV_cost_{5-7} 모두가 RMV_cost_4 보다 크다면 mode 5-7에 해당하는 유효-왜곡 최적화는 생략하고 mode 8은 8x8 크기의 가변 블록 4개로 이루어진 후보 모드로 결정된다. 1차 정규화 및 유효-왜곡 최적화를 통해 mode 8을 결정한 뒤에는 2차 정규화와 2차 적응적 유효-왜곡 최적화를 수행한다. 2차 정규화는 MV_cost_{1-3} 과 mode 8에서의 최소 RMV_cost 에 대해 수행된다. 2차 정규화는 아래의 식에 의해 계산된다.

$$RMV_cost_{1-3,8} = MV_cost_{1-3,8} \times \frac{\text{number of pixels mode1}}{\text{number of pixels mode1} \sim 3,8} \quad (12)$$

이 때 MV_cost_8 은 mode 8의 결정할 때 선택된 RMV_cost 중 최소 RMV_cost 를 사용하며 number of pixels mode 8은 8x8크기로 1차 정규화

된 값에 해당하는 픽셀 수이므로 8×8 에 해당하는 픽셀의 수를 사용한다. 2차 윌-왜곡 최적화는 16×16 크기의 가변 블록에 대해서는 윌-왜곡 최적화를 수행하고, RMV_cost_1 을 계산하고, 나머지 가변 블록 모드에 대해서는 1차 윌-왜곡 최적화와 같이 RMV_cost_1 과 $RMV_cost_{2,3,8}$ 을 비교하고 RMV_cost_1 보다 큰 경우에는 해당 가변 블록의 RD_cost 도 mode 1의 RD_cost 보다 클 것으로 판단하여 윌-왜곡 최적화를 생략하게 된다. 이 때 이 고속 윌-왜곡 최적화 알고리즘은 가변 블록을 결정하는 윌-왜곡 최적화에만 적용되고, 인트라 예측 부호화, 스킵 모드의 윌-왜곡 최적화는 정상적으로 수행한다. 그림 8은 본 장에서 제안된 고속 윌-왜곡 최적화 알고리즘을 전체 흐름도를 보여준다.



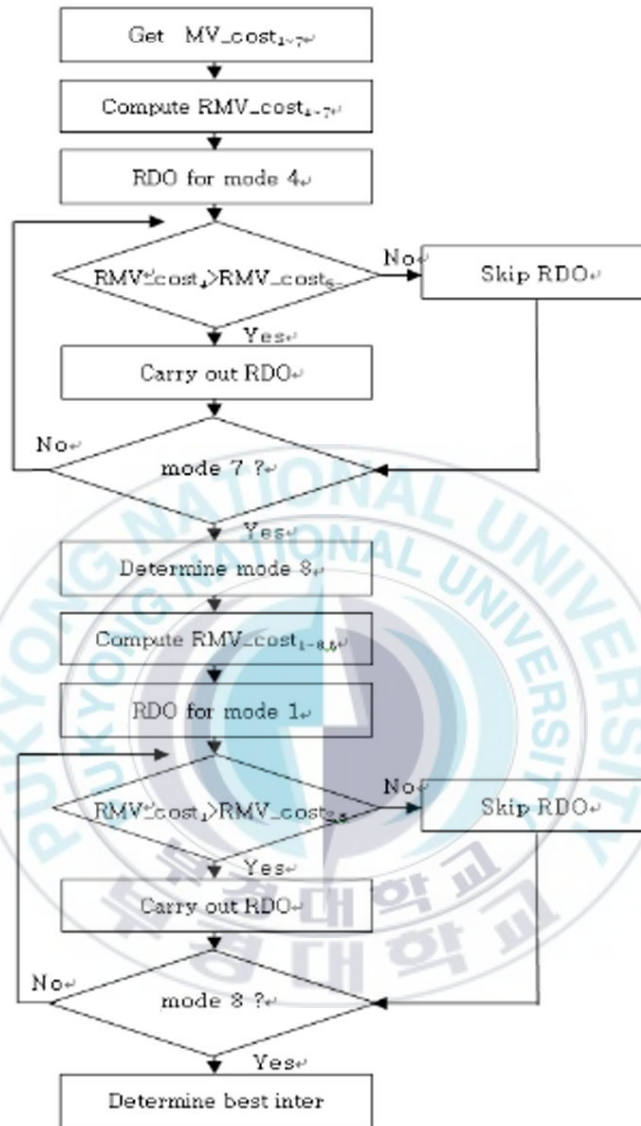


그림 8. 제안된 고속 윌-왜곡 최적화 알고리즘

2. 선택적 인트라 예측 부호화 알고리즘

인트라 예측 부호화는 참조 픽처 없이 해당 픽처를 예측하는 방법이다. 주로 가장 첫 번째 픽처인 I-picture 또는 누적 에러 방지 및 화면 전환 등의 이유로 사용자가 지정한 픽처를 예측 할 때 사용한다. 하지만 H.264/AVC에서는 참조 픽처를 사용하여 예측하는 P-picture에서도 인트라 예측과 같이 인트라 예측을 적용시켜 사용하고 있다. 앞서 언급했듯이 H.264/AVC에서는 주변에 코딩된 화소값을 이용하여 해당 매크로 블록의 화소값들을 바로 예측하는 형태의 인트라 예측 부호화를 사용한다. 그러므로 압축 효율 측면에서는 좋은 결과물을 얻을 수 있지만, 인코더의 복잡도는 더욱 증가하게 된다.^[14-15] 본 장에서는 P-picture에서 적용되는 인트라 예측 부호화를 영상 복잡도와 예측 움직임 벡터, 최적의 가변 블록 모드를 이용하여 선택적으로 적용하여 인트라 예측 부호화를 사용할 때의 이점은 살리며 인코더의 복잡도를 감소시키는 알고리즘을 제안한다.

2.1. 영상 복잡도

영상 복잡도 (Image Complexity)란 영상 내 밝기 변화의 정도를 나타내는 기준 또는 척도를 의미한다. 영상 복잡도가 큰 영상의 경우는 영상 내에 급격한 밝기 변화 구간이 많다는 것을 의미하며, 이는 경계선 (Edge)과 같은 중요한 정보를 많이 포함하고 있다는 의미가 된다. 또한 영상 복잡도가 클수록 정합 오차가 크게 발생하게 된다. 영상 복잡도가 클수록 영상 내에 밝기 차가 급격한, 즉 경계선들이 많이 존재하므로 주로 화소값의 차이를 이용하는 정합 오차의 값은 경계선이 없는 영상보다는 커질 가능성이 크며 이러한 관계는 Taylor series expansion을 통해서 이미 증명되었다.

영상 복잡도는 고속 움직임 추정 알고리즘이나 영상 분석 등에 많이 활용되고 있다.

영상 복잡도를 계산하는 방법에는 여러 가지가 있지만, 가장 기본적인 방법으로는 Laplacian, Roberts, Sobel과 같은 경계선 검출 마스크를 이용하여 구할 수 있다. 하지만, 이런 경계선 검출 마스크를 이용한 방법들은 부가적인 계산량을 요구하는 단점이 있다. 그리고 정합 오차를 이용하여 복잡도를 판단할 수 있는데 이는 앞서 언급한 듯이 Taylor series expansion에 의해 가능하다.

본 논문에서는 16x16 크기의 매크로 블록에 대한 영상 복잡도 정보를 계산하며, 계산 방법은 공간상의 밝기 차이를 이용하는 방법이다. 각 화소에 대하여 식 13과 같이 인접한 오른쪽과 아래쪽의 화소와의 차이를 더한 값으로 계산한다.

$$IC(x, y) = |f(x, y) - f(x+1, y)| + |f(x, y) - f(x, y+1)| \quad (13)$$

위의 식에서 $f(x, y)$ 는 현재 화소의 위치를 의미하며 매크로 블록의 IC는 매크로 블록 내의 모든 화소의 영상 복잡도 값을 더하여 계산한다.

2.2. P-picture에서의 인트라 예측 부호화 성능 분석

원래 인트라 예측 부호화는 참조 픽처가 존재하지 않는 I-picture의 코딩을 위해 사용되는 기법이다. 이미 코딩된 주변의 화소들을 이용하여 매크로 블록의 화소들을 예측해내는 방식을 사용하므로 복잡한 영상보다는 단순한 영상 즉, 배경 같은 영상의 예측이 더 쉽고 정확하게 이루어 질 가능성이 크다. 그리고 전송되는 값은 인트라 예측 모드로 코딩되었다는 것

을 알려주는 플래그와 최적으로 선택된 인트라 모드의 종류만 알려주게 되므로 발생 비트량이 가변 블록들의 발생 비트량에 비해 적다. 하지만 H.264/AVC에서는 I-picture뿐만 아니라 P-picture에서도 인트라 예측 부호화를 적용시키고 있으며, 그 결과 QP에 따른 특성을 더욱 부각시킬 수 있다. 낮은 QP에서는 유효-왜곡 최적화를 통해 화질 위주의 모드 선택이 이루어지는데 최적으로 선택된 가변 블록 모드의 화질과 최적으로 선택된 인트라 예측 모드의 화질이 비슷할 경우에는 정보량이 적은 인트라 예측 부호화가 선택됨으로써 비트량을 감소시킬 수 있다. 반면 높은 QP에서는 유효-왜곡 최적화를 통해 비트량 위주의 모드 선택이 이루어지는데 동일 비트량에서 인트라 예측 모드의 화질이 반드시 좋다고는 할 수 없으나, 배경 같이 단순한 영상에 대해서는 화질이 더 좋을 가능성이 있으며, 화질이 더 좋을 경우 최종 최적 모드로 선택되어 전체 화질을 높여주게 된다. 하지만, 인트라 예측 부호화를 P-picture에 적용할 경우 인코더의 복잡도가 더욱 증가하는 큰 단점이 생기게 된다. 앞서 언급했듯이 인트라 예측 부호화는 휘도 성분의 4x4크기의 9가지 모드와 16x16크기의 4가지 모드, 색차 성분의 4가지 모드가 존재하고 각각에 대해 유효-왜곡 최적화를 통해 최적 인트라 최적의 인트라 예측 모드를 결정하게 된다. 인트라 예측 부호화는 매크로 블록의 모든 화소를 예측하므로 많은 시간이 소요되고, 최적 인트라 예측 모드를 결정하기 위해서는 휘도 성분에서 각 모드 별로 1번씩 유효-왜곡 최적화를 수행해야 하며 그에 따라 색차 성분의 수만큼 유효-왜곡 최적화를 수행해야 한다. 즉, $\{[(\text{휘도 } 16 \times 16 \text{ 4회}) + (\text{휘도 } 4 \times 4 \text{ 9회} \times 16)] \times \text{색차 } 8 \times 8 \text{ 4회}\}$ 라는 계산에 의해 무려 592회의 유효-왜곡 최적화가 이루어지게 된다. 하지만 실제 코딩 시 인트라 예측 모드가 최적으로 선택되는 것은 아래 표 3에서와 같이 평균적으로 3%정도이다. 즉, 선택되어지는 비율에 비해 할당된 계산량이 극도로 많다는 것을 알 수 있다.

표 3. QP 별 인트라 예측 모드 선택 경향

sequence	QP							
	25		30		35		40	
container	166	1.69%	167	1.70%	157	1.60%	117	1.19%
news	124	1.27%	151	1.54%	177	1.81%	193	1.97%
foreman	664	6.77%	658	6.71%	684	6.98%	586	5.98%
silent	395	4.03%	409	4.17%	444	4.53%	512	5.22%

표 3은 QCIF 크기의 영상 container, news, foreman, silent voice에 대해 IPPP형태의 100 frame에서 QP별로 첫 번째 I-picture를 제외한 P-picture에서 인트라 예측 모드가 최적으로 선택되는 횟수를 나타내었으며, 옆의 백분율 값은 인트라 예측 부호화가 최적으로 선택되는 비율을 의미한다.

2.3 P-picture에서 인트라 예측 모드의 선택 조건 분석

인트라 예측 부호화는 주변 블록의 경계 부분의 화소값을 이용하여 매크로 블록의 화소값들을 예측해내는 방식이므로 매크로 블록 내에 물체 즉 경계선이 많고 움직임 있는 복잡한 영상보다는 움직임이 없이 정지해 있거나 단조로운 배경과 같은 단순한 영상에 대해서 효율적인 예측을 할 가능성이 크다. 그러므로 아래 표 4에 인트라 예측 모드가 최적으로 선택되었을 때와 그렇지 않을 때의 매크로 블록의 영상 복잡도를 계산하여 비교하여 나타내었다. 표 4에는 QP 32에 QCIF 크기의 영상 3가지를 대상으로 IPPP 형태의 100 frame을 코딩하였을 때 P-picture에 해당하는 매크로 블록의 영상 복잡도를 구하여 비교하였다. 그 결과 인트라 예측 모드가 최적으로 선택 될 경우는 그렇지 않은 경우보다 영상 복잡도가 상당히 작다는

것을 알 수 있다. carphone과 stefan의 경우 인트라 예측 모드가 최적일 경우 매크로 블록의 영상 복잡도 평균은 그렇지 않은 경우의 반 정도로 나타나며, mobile의 경우는 3% 정도로 나타나 매크로 블록의 영상 복잡도가 작을 경우 인트라 예측 모드를 선택할 가능성이 높은 것을 알 수 있다.

표 4. 최적 모드의 종류에 따른 매크로 블록의 영상 복잡도

	carphone		mobile		stefan	
	inter mode	intra mode	inter mode	intra mode	inter mode	intra mode
avg complexity	4127.2	2436.31	9665.61	337.5	8941.85	4073.18

그리고 H.264/AVC에서 여러 가지 크기의 가변 블록을 이용하여 움직임 추정 및 보상을 하고, 인트라 예측 부호화 시 4x4와 16x16으로 블록을 나누는 이유도 영상에 따라, 또는 같은 영상이라도 픽처에 따라 달라지는 영상 복잡도에 적응적으로 대처하기 위함이다. JM에서는 최종적인 율-왜곡 최적화를 적용시켜 최적의 모드를 결정할 때 스킵 모드와 각 가변 블록들을 비교하고 그 결과를 나머지 인트라 모드와 비교하여 결정하게 구현되어 있다. 그러므로 인트라 모드를 비교하기 전에 이미 최적의 인터 모드는 결정되는 것이며 이때의 인터 모드의 블록 크기는 매크로 블록의 영상 복잡도를 의미한다고도 할 수 있다. 즉, 최적의 인터 모드가 mode 1-3일 경우 해당 매크로 블록은 단순한 영상이라고 할 수 있으며, mode 8인 경우는 복잡한 영상이라고 할 수 있다. 그러므로 최적의 인터 모드의 블록 크기와 인트라 예측 모드의 블록 크기는 상관관계가 있을 가능성이 크다. 아래 표 5에 QCIF 크기의 영상 carphone과 stefan에서 QP가 32일 때 10 frame에서 인트라가 최적으로 선택될 때 최적의 인터 모드를 조사하여 나타내었다. carphone의 경우 4x4 크기의 인트라 모드가 7, 16x16 크기의 인트라 모드가 9번 선택 되었으며, stefan의 경우는 16x16 크기의 인트라 모드만

5번이 선택되었다. 이 때 각각에 대해 최적으로 선택된 인터 모드를 보면 16x16 크기의 인트라 모드에 대해서는 모두 크기가 큰 인터 모드나 skip모드가 최적인 상태이었으며, 4x4 크기는 7번 중 5번이 작은 크기의 인터 모드가 최적인 상태였다. 그러므로 최적의 인터 모드의 블록 크기는 인트라 예측 모드의 블록 크기와 상관 관계가 있음을 알 수 있다.

표 5. 인트라 모드가 최적으로 선택 될 경우 최적의 인터 모드

	carphone		stefan
	I4MB	I16MB	I16MB
mode 0		2	3
mode 1	1	3	2
mode 2	1	4	
mode 3			
mode 8	5		

2.4 제안하는 알고리즘

P-picture에서의 인트라 예측 부호화는 평균 3%의 선택 비율을 차지하고 있지만, QP의 특성을 더욱 살리고 압축 효율을 높이기 위해 적용되고 있다. 하지만, 선택되는 비율에 비해 할당되는 계산량은 극도로 많아 인코더의 복잡도를 증가시키는 큰 요인으로 작용하므로 단계적으로 선택적 인트라 예측 부호화를 적용시켜 계산량을 감소시키는 알고리즘을 제안한다.

인트라 예측 부호화는 단순한 영상일 때 효율적으로 예측 할 수 있으며, 이 때 최적의 모드로 선택될 가능성도 높았다. 그러므로 해당 첫 번째 단계로 매크로 블록이 단순한 영상일 경우에만 인트라 예측 부호화를 적용시킨다. 그러기 위해 인트라 예측 부호화를 적용하기 이전에 매크로 블록의 영상 복잡도를 구하고 이 값을 임계값과 비교하여 단순한 영상으로 판단되

면 인트라 예측 부호화를 적용하고 복잡한 영상으로 판단되면 인트라 예측 부호화를 생략한다. 이 때 인트라 예측 부호화의 생략 여부를 판단하는 임계값은 TH_{intra} 라고 하고 이전 픽처의 매크로 블록들의 영상 복잡도의 평균값과 율-왜곡 관점을 고려한 가중치인 *lagrangian* 승수를 응용한 *weight*를 파라미터로 한 값을 이용하여 아래의 식 14와 같이 계산된다.

$$TH_{intra} = avg_complexity_{t-1} + (avg_complexity_{t-1} \times weight) \quad (14)$$

위의 식에서 $avg_complexity_{t-1}$ 은 이전 픽처의 평균 영상 복잡도를 의미하고, *weight*는 *lagrangian* 승수의 계산법을 응용한 가중치로서 식 15와 같이 계산된다.

$$weight = \frac{2^{[(QP-32)/6]} - 1}{2} \quad (15)$$

이 때 $2^{[(QP-32)/6]}$ 은 *lagrangian* 승수의 계산법을 응용한 것으로 율-왜곡 곡선의 접점의 기울기인 *lagrangian* 승수의 경향을 따르기 위한 수식이며, QP에서 32를 빼는 것은 현재 가장 많이 사용되는 QP인 32를 기준으로 가중치를 결정하기 위함이고 나머지 계산 부분은 가중치의 크기를 조절하기 위해 사용된다. 식 15에 의해 계산된 *weight*는 아래 그림 9와 같이 율-왜곡 커브의 기울기를 따르는 값들을 가지게 된다. 이렇게 계산된 TH_{intra} 는 I-picture와 첫 번째 P-picture를 제외한 두 번째 P-picture부터 적용되며, 인트라 예측 부호화가 적용되기 전에 해당 매크로 블록의 영상 복잡도와 비교하여 해당 매크로 블록의 영상 복잡도가 TH_{intra} 보다 작을 경우에만 인트라 예측 부호화를 수행하고 반대의 경우에는 인트라 예측 부호화 과정을

생략한다.

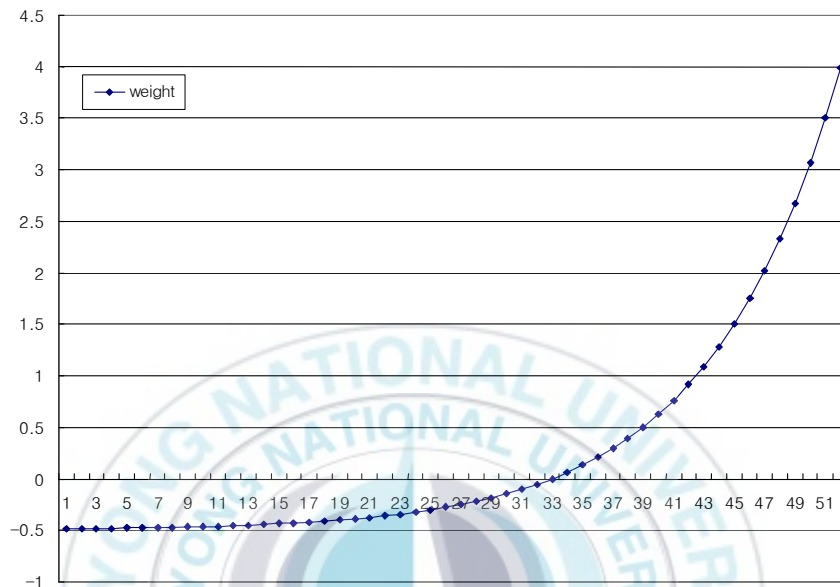


그림 9. QP에 따른 weight

해당 매크로 블록이 TH_{intra} 보다 작아서 인트라 예측 부호화를 수행하는 것으로 결정 된 뒤 인트라 예측 부호화를 수행 하는 매크로 블록에만 한하여 두 번째 선택적 인트라 예측 부호화를 적용한다. 두 번째 선택적 인트라 예측 부호화는 최적의 인터 모드를 이용한다. 앞서 언급했듯이 최적의 인터 모드와 인트라 예측 모드의 블록 크기는 상당히 밀접한 관계를 가지고 있었으므로 최적의 인터 모드의 블록 크기에 맞는 인트라 예측 모드를 수행한다. 즉, 최적 인터 모드가 mode 0-3 (skip, 16x16, 16x8, 8,16)일 경우는 16x16 크기의 인트라 예측 부호화만 적용하며, 최적의 인터 모드가 mode 8 (8x8 이하의 가변 블록으로 구성)일 경우에는 4x4 크기의 인트라 예측 부호화를 적용한다.

이렇게 두 번에 걸친 선택적 인트라를 적용 후에도 인트라 예측 부호화를 수행해야 할 경우에는 세 번째 선택적 인트라 예측 부호화를 적용한다. 세 번째 선택적 인트라 예측 부호화는 매크로 블록내의 움직임 여부를 고려하여 인트라 예측 부호화의 생략 여부를 결정한다. 이를 위해서 예측 움직임 벡터를 사용한다. 이미 최적의 인터 모드의 블록 크기에 따라 인트라 예측 모드의 블록 크기가 정해진 상태이므로 그때의 해당하는 예측 움직임 벡터가 모두 (0, 0)일 경우에만 인트라 예측 부호화를 생략하고, 움직임이 조금이라도 있을 경우에는 인트라 예측 부호화를 적용한다. 인트라 예측 모드의 블록 크기에 따른 예측 움직임 벡터의 적용 방법은 16x16 크기의 인트라 예측 부호화를 적용할 경우 16x16, 16x8, 8x16, 8x8 크기에 해당하는 가변 블록의 예측 움직임 벡터를 조사하고, 4x4 크기의 인트라 예측 부호화를 적용할 경우 8x8, 8x4, 4x8, 4x4 크기에 해당하는 가변 블록의 예측 움직임 벡터를 조사하여 인트라 예측 부호화의 생략 여부를 결정한다. 제안한 선택적 인트라 예측 부호화 알고리즘의 전체적인 과정은 다음과 같다.

SETP 1 : 이전 픽처의 평균 영상 복잡도를 이용한 TH_{intra} 계산

SETP 2 : 해당 매크로 블록의 영상 복잡도를 구하여 TH_{intra} 와 비교

- 영상 복잡도 $> TH_{intra}$: 인트라 예측 부호화 생략
- 영상 복잡도 $\leq TH_{intra}$: STEP 3

SETP 3 : 최적의 인터 모드 조사

- 최적의 인터 모드 = mode 0-3 : STEP 4
- 최적의 인터 모드 = mode 8 : STEP 5

STEP 4 : mode 1-4의 예측 움직임 벡터 조사

- 모든 예측 움직임 벡터=(0, 0) : 인트라 예측 부호화 생략

- 모든 예측 움직임 벡터 $\neq (0, 0)$: 16x16 인트라 예측 부호화 적용

STEP 5 : mode 4-7의 예측 움직임 벡터 조사

- 모든 예측 움직임 벡터 $= (0, 0)$: 인트라 예측 부호화 생략
- 모든 예측 움직임 벡터 $\neq (0, 0)$: 4x4 인트라 예측 부호화 적용



IV. 모의 실험 및 결과

본 장에서는 H.264/AVC에서 공통 실험 조건을 위해 제공하는 참조 소프트웨어인 JM을 이용하여 제안된 알고리즘에 대한 모의 실험을 수행하고 그 결과를 표준인 H.264/AVC의 결과와 비교 및 검토한다. 모의 실험은 제안된 알고리즘을 독립적으로 적용한 경우와 공동으로 적용한 경우로 나누어 수행되고 H.264/AVC에서 권고하는 공통 실험 조건에 따랐으며, 그림 10의 영상들을 표 6에의 조건으로 사용하고, 표 7에 나타난 환경에서 수행된다.

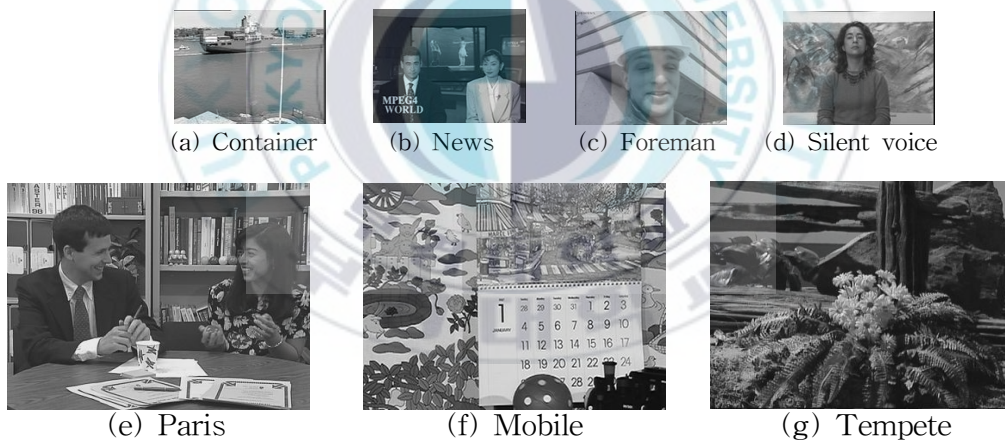


그림 9. 실험 영상

표 6. 실험 영상 크기 및 환경

size	sequences	# frame	frequency[Hz]
QCIF	Container	300	30
	News	300	30
	Foreman	300	30
	Silent Voice	300	30

CIF	Paris	300	30
	Mobile	300	30
	Tempete	300	30

표 7. 실험 환경

CPU	Intel Pentium 4 / 2.4 GHz
Memory	256 MHz
O/S	Windows XP Home Edition
encoder	JM 9.5
QP	28, 32, 36, 40
Search range	± 16 (QCIF) / ± 32 (CIF)
vetc	H.264/AVC Baseline Profile Rate-Distortion Optimization IPPP CAVLC RS, FMO, ASO Disable

표 8은 제안한 적응적 율-왜곡 최적화 알고리즘의 결과에 대한 고찰을 위하여 P-picture에서 인트라 예측 부호화를 하지 않게 설정한 상태에서 율-왜곡 최적화를 적용하였을 때와 하지 않았을 때의 결과를 나타내었다. 인트라 예측 모드에 관련된 율-왜곡 최적화의 영향을 제외한 인터 모드에 대한 율-왜곡 최적화 영향을 보기 위해 P-picture에서 인트라 예측 모드를 적용하지 않았으며, 이때 율-왜곡 최적화가 부호화의 전체 복잡도에서 얼마만큼의 비율을 차지하는지 알아보기 위해 율-왜곡 최적화를 적용시켰을 때와 시키지 않았을 때의 결과를 조사하였다. 표 8에서 보면 율-왜곡 최적화를 사용에 따라 Δ Total time이 7-8% 정도 차이가 나고 있다. 율-왜곡 최적화를 사용하지 않아도 다른 비용함수를 사용하여 최적 모드를 결정하는 것을 감안하면 인터 모드에 대해서 율-왜곡 최적화 기법이 차지하는 비율은 약 5% 정도라고 할 수 있겠다. 이는 제안하는 알고리즘을 적용하였을 때 기대할 수 있는 시간 감소량이 5%전후의 값이 될 것이라는 의

미이다.

표 8. P-picture에서 인트라 예측 부호화 없이 율-왜곡 최적화에 따른 결과

sequence	QP	skip intra in inter / RDO - on			skip intra in inter / RDO - off		
		Δ Total time [sec]	Δ SNR Y [dB]	Δ bit rate [kbits/s]	Δ Total time [sec]	Δ SNR Y [dB]	Δ bit rate [kbits/s]
container	28	-43.34%	-0.03	0.95%	-50.56%	0.02	11.60%
	32	-40.99%	-0.06	4.38%	-48.20%	-0.04	8.48%
	36	-38.98%	-0.07	0.33%	-46.64%	-0.05	8.31%
	40	-37.44%	-0.15	0.39%	-45.36%	-0.07	5.26%
news	28	-44.04%	-0.03	1.43%	-51.60%	0.14	8.70%
	32	-41.92%	-0.02	2.61%	-49.50%	0.01	8.49%
	36	-39.55%	-0.10	3.31%	-47.67%	-0.04	7.74%
	40	-38.37%	-0.05	3.27%	-46.30%	-0.05	9.70%
foreman	28	-42.46%	-0.05	1.50%	-50.28%	0.10	9.47%
	32	-40.00%	-0.05	1.78%	-47.80%	0.10	8.66%
	36	-37.96%	-0.08	1.61%	-45.85%	0.11	9.06%
	40	-36.45%	-0.16	1.68%	-44.37%	0.06	11.07%
silent voice	28	-43.63%	0.00	1.30%	-51.13%	0.00	8.08%
	32	-40.87%	-0.01	1.89%	-48.52%	0.00	7.39%
	36	-38.54%	-0.03	1.00%	-46.29%	-0.01	7.04%
	40	-36.86%	-0.02	1.13%	-44.73%	-0.11	1.42%

제안하는 적응적 율-왜곡 최적화 알고리즘과 선택적 인트라 예측 부호화 알고리즘을 독립적으로 실험한 결과를 표 9에 나타내었다. proposed 1은 적응적 율-왜곡 최적화 알고리즘을 의미하며, proposed 2는 선택적 인트라 예측 부호화 알고리즘을 의미한다. 이때 proposed 1은 표 8과 비교하여 분석하기 위해 P-picture에서 인트라 예측 부호화를 적용하지 않은 상태에서 실험을 수행하였다. proposed 1의 결과를 보면 표 8의 P-picture에서 인트라 예측 모드를 하지 않고 율-왜곡 최적화를 한 결과에 비해 Total time은 약 3-4% 정도 감소하였으며, 화질은 0.01-0.05 dB정도의 감소가 있고, 비

트율은 거의 동일하거나 미미한 저하를 보이고 있다. 표 8을 통해 예상했던 감소율 5%에 부합하는 감소율을 보이면서 화질과 비트율은 유지하는 결과를 보여 제안한 알고리즘이 인터 모드에 관련된 율-왜곡 최적화 기법의 계산량을 감소시켰음을 알 수 있다. proposed 2의 결과를 보면 Total time을 30-42% 감소시키면서 화질은 표준안의 화질을 유지하거나 최대 열화된 것이 0.06dB일 정도로 미미한 열화를 보였으며, 비트량도 최대 1.38% 증가하였을 뿐 표준안의 비트율을 거의 유지하는 수준의 결과를 보였다. 게다가 인트라 예측 부호화를 선택적으로 사용함으로써 QP에 따른 압축 특성을 따르고 있는 것을 알 수 있다. 즉, 높은 QP일수록 화질의 감소량과 비트량의 감소량이 거의 선형적으로 이루어져 압축 특성을 따르고 있다.

표 10에는 제안한 두 알고리즘을 동시에 적용하였을 경우의 결과를 나타내었다. 그리고 QCIF뿐 아니라 CIF영상 3가지에 대해서도 적용시켜 실험하였다. 우선 QCIF에 대한 결과를 보면 Total time은 34-43%를 감소시키면서 화질은 최대 0.08 dB의 열화를 보였으며, 비트량은 news가 2.31% 증가한 것을 제외하고는 2%미만의 증가량을 보이고 있다. 이러한 결과는 표 8의 P-picture에서 인트라 예측 부호화를 하지 않은 결과와 비교하여도 1-2%정도 차이나는 거의 동일한 시간 감소를 보이면서도 화질이나 비트량 측면에서 제안한 알고리즘이 더욱 우수한 결과를 보이는 것을 알 수 있다. CIF의 경우는 QCIF에 비해 영상의 크기가 4배가 늘어나므로 율-왜곡 최적화, 인트라 예측 부호화, 가변 블록을 사용한 움직임 추정 및 보상 기법 등의 계산량이 4배가 되게 된다. 하지만, 영상이 커짐에 따라 탐색 영역도 4배로 커져 가변 블록을 사용한 움직임 추정 및 보상 기법의 계산량은 모드 별로 다시 4배가 된다. 그러므로 전체 코딩 시간에서 움직임 추정 및 보상 기법이 차지하는 비율이 대부분이 되어 제안하는 알고리즘으로 감소

시킬 수 있는 시간이 한정되어 버리는 결과를 나타내었다. CIF에서는 17-22%의 시간적 감소를 보이고, 화질과 비트량은 표준안의 결과를 유지하는 수준의 결과를 보이고 있다.



표 9. 제안하는 적응적 울-왜곡 최적화, 선택적 인트라 예측 부호화 알고리즘의 결과

size	sequence	condition	QP	Total time [sec]	SNR Y [dB]	bit rate [kbits/s]	ΔTotal time [sec]	ΔSNR Y [dB]	Δbit rate [kbits/s]
QCIF	container	proposed 1	28	157.084	35.76	40.50	-46.46%	-0.04	1.25%
			32	156.119	32.94	21.54	-44.69%	-0.05	0.42%
			36	155.990	30.08	12.27	-42.98%	-0.13	-0.08%
			40	155.732	27.47	7.82	-41.62%	-0.16	0.39%
	news	proposed 1	28	155.513	36.52	76.24	-47.42%	-0.03	2.89%
			32	154.610	33.46	45.72	-45.71%	-0.03	3.02%
			36	154.314	30.52	27.77	-43.90%	-0.10	3.31%
			40	153.653	27.80	17.14	-42.57%	-0.11	3.88%
	foreman	proposed 1	28	161.414	35.67	134.17	-45.90%	-0.07	1.95%
			32	161.101	32.78	77.84	-43.58%	-0.08	1.63%
			36	160.846	30.13	48.05	-41.79%	-0.12	3.18%
			40	160.735	27.60	29.78	-40.45%	-0.17	2.37%
	silent voice	proposed 1	28	157.292	35.68	85.81	-47.07%	-0.03	1.63%
			32	157.083	32.73	51.07	-44.64%	-0.04	2.78%
			36	157.340	30.15	29.81	-42.48%	-0.06	2.33%
			40	157.705	27.69	17.33	-40.87%	-0.12	2.85%
QCIF	container	proposed 2	28	170.039	35.8	40.13	-42.05%	0	0.32%
			32	169.501	32.98	21.21	-39.95%	-0.01	-1.12%
			36	169.169	30.21	12.21	-38.16%	0	-0.57%
			40	169.918	27.57	7.8	-36.30%	-0.06	0.13%
	news	proposed 2	28	173.737	36.56	74.78	-41.26%	0.01	0.92%
			32	172.152	33.51	44.99	-39.55%	0.02	1.37%
			36	170.565	30.6	27.25	-38.00%	-0.02	1.38%
			40	171.646	27.92	16.68	-35.84%	0.01	1.09%
	foreman	proposed 2	28	189.821	35.72	132.59	-36.38%	-0.02	0.75%
			32	185.832	32.83	77.08	-34.92%	-0.03	0.64%
			36	183.623	30.21	46.86	-33.55%	-0.04	0.62%
			40	187.367	27.72	28.98	-30.58%	-0.05	-0.38%
	silent voice	proposed 2	28	181.47	35.71	84.71	-38.93%	0	0.33%
			32	177.578	32.75	50.17	-37.41%	-0.02	0.97%
			36	175.686	30.22	29.1	-35.78%	0.01	-0.10%
			40	177.04	27.8	16.81	-33.62%	-0.01	-0.24%

표 10. 제안하는 알고리즘의 공동 적용 결과

size	sequence	condition	QP	Total time [sec]	SNR Y [dB]	bit rate [kbits/s]	Δ Total time [sec]	Δ SNR Y [dB]	Δ bit rate [kbits/s]
QCIF	container	proposed 3	28	168.816	35.79	40.33	-42.47%	-0.01	0.82%
			32	159.498	32.98	21.35	-43.49%	-0.01	-0.47%
			36	158.475	30.17	12.24	-42.07%	-0.04	-0.33%
			40	170.323	27.63	7.87	-36.15%	0	1.03%
	news	proposed 3	28	171.279	36.54	75.15	-42.09%	-0.01	1.42%
			32	161.611	33.48	45.31	-43.26%	-0.01	2.10%
			36	159.53	30.56	27.5	-42.01%	-0.06	2.31%
			40	171.099	27.86	16.86	-36.05%	-0.05	2.18%
	foreman	proposed 3	28	180.886	35.71	133.54	-39.38%	-0.03	1.47%
			32	175.886	32.81	78.1	-38.40%	-0.05	1.97%
			36	171.434	30.17	47.62	-37.96%	-0.08	2.25%
			40	177.741	27.71	29.54	-34.14%	-0.06	1.55%
	silent voice	proposed 3	28	172.794	35.72	85.01	-41.85%	0.01	0.69%
			32	167.669	32.74	50.31	-40.90%	-0.03	1.25%
			36	161.307	30.19	29.34	-41.03%	-0.02	0.72%
			40	173.817	27.76	16.98	-34.83%	-0.05	0.77%
CIF	paris	proposed 3	28	1916.908	35.38	484.37	-21.36%	-0.01	0.21%
			32	1887.152	32.23	271.07	-20.61%	-0.01	0.37%
			36	1869.994	29.32	145.14	-19.67%	-0.03	0.26%
			40	1856.266	26.63	79.12	-19.43%	0	0.55%
	mobile	proposed 3	28	2005.503	33.69	1809.27	-22.20%	-0.02	0.03%
			32	2009.434	30.11	894.18	-19.09%	-0.02	0.08%
			36	1951.406	26.88	398.19	-19.06%	-0.03	0.15%
			40	1948.295	24.02	192.4	-17.47%	-0.02	0.40%
	tempete	proposed 3	28	1652.25	34.62	1305.68	-20.22%	-0.01	0.24%
			32	1622.5	31.32	628.58	-19.04%	-0.01	0.60%
			36	1606.812	28.4	292.32	-18.18%	-0.02	0.91%
			40	1584.513	25.84	148.77	-18.27%	-0.03	1.18%

V. 결 론

본 논문은 새로운 표준안인 H.264/AVC에서 부호화 효율을 개선하기 위해 새롭게 적용시킨 움직임 추정 및 보상 기술들을 기술하고, H.264/AVC의 단점으로 지적되는 부호기에서의 높은 복잡도의 주요 원인들 중 하나인 윌-왜곡 최적화 기법과 인트라 예측 부호화에 대하여 성능을 분석하고, 화질과 비트량은 유지하면서 부호기의 복잡도를 줄일 수 있는 새로운 고속 알고리즘을 제안하였다. 제안된 알고리즘은 매크로 블록의 최적 모드를 결정하기 위해 사용되는 비용 함수 RD_cost의 계산에 따른 복잡도를 줄이기 위해 최적의 참조 픽처를 선택할 때 사용되는 MV_cost를 이용하여 RD_cost를 사전에 예측하여 적응적으로 윌-왜곡 최적화를 생략하며 수행한다. 또한 매크로 블록의 영상 복잡도에 따라 인트라 예측 부호화의 생략 여부를 결정하고, 추가적으로 최적의 인터 모드의 블록 크기에 따라 인트라 예측 모드의 블록 크기를 결정하고 예측 움직임 벡터를 이용하여 다시 한번 인트라 예측 부호화를 생략 여부를 결정하는 알고리즘을 제안하였다. 모의 실험의 결과 전자의 알고리즘은 휘도 성분의 SNR은 평균 0.08 dB 감소 하였으며 비트량은 평균 2.11% 증가하지만, 전체 코딩 시간은 인터 모드에서 윌-왜곡 최적화가 차지하는 5% 전후의 감소량을 보였다. 후자의 경우는 평균 0.01dB의 화질 열화와 평균 0.38%의 비트량 증가를 보여 화질과 비트량은 거의 변화가 없는 상태에서 전체 코딩 시간은 평균 37%의 감소를 보였다. 두 알고리즘을 같이 사용할 경우는 알고리즘간의 상호 작용으로 인해서 QCIF에 대해서는 평균 0.03dB의 화질 열화와 1.23%의 비트량 증가에 평균 39.76%의 코딩 시간의 단축을 시키고 QP에 따른 압축

특성을 따르는 결과를 보였다. 즉, 극단적으로 율-왜곡 최적화나 P-picture에서 인트라 예측 부호화를 하지 않은 수준의 시간적 감소에 표준안의 화질과 비트량을 유지하는 결과를 보여주었다. CIF의 경우 영상의 크기가 커짐에 따라 탐색 영역도 크게 설정을 하여 실험을 하여 전체 코딩 시간에서 율-왜곡 최적화나 인트라 예측 부호화의 비율이 낮아져 QCIF의 경우만큼 만족스러운 결과는 얻지 못했다. 평균 0.01dB의 화질 열화와 평균 0.42%의 비트량 증가에 평균 19.55%의 시간 감소를 보이는 결과를 얻었다.

일반적으로 지금까지의 H.264/AVC의 고속 알고리즘들은 가변 블록이나 인트라 예측 부호화 시 모드의 수를 줄인다거나 참조 픽처, 탐색 영역의 조절 등을 이용한 방법들이 대부분이었다. 본 논문에서 제안한 고속 알고리즘을 이러한 여타 고속 알고리즘들과 같이 사용하면 더욱 좋은 성능을 보일 것으로 예상된다.

VI. 참고문헌

1. MPEG-1 : "Coding of moving pictures and associated audio for digital storage media at up to About 1.5 Mbit/s," ISO/IEC 1117-2 : video, Nov.,1991.
2. MPEG-2 : "Generic coding of moving pictures and associated audio information," ISO/IEC 13818-2 Video, draft international standard, Nov., 1994.
3. MPEG-4-Information Technology-Coding of Audio-Visual Objects-Part 2 : Visual, ISO/IEC 14496-2, 2000.
- 4.ISO/IEC 14496-10 and ITU-T Rec. H.264, advanced video coding, 2003.
5. H.261 : "Recommendation H.261, video codec for audiovisual services at p x 64 kbits/s", Geneva, 1990
6. "Video coding for low bitrate Communication," International Telecom-munications Union, ITU-T Recommendation H.263, 1998.
7. 카도노 신야, 키쿠치 유시히로, 스즈키 테루히코, "H.264/AVC 비디오 압축 표준," 홍릉과학출판사, 2005
8. T.Wiegand, G.Sullivan, G.Bjontegaard, A.Luthra, "Overview of the H.264/AVC video coding standard," IEEE TransactionsCircuits and Systems for Video Technology, vol. 13, pp. 560-576, July 2003.
9. R.Schafer, T.Wiegand, H.Schwarz, "The emerging H.264/AVC standard," Heinrich Hertz Institute, Berlin, Germany, Audio/Video Coding, Jan. 2003.
10. Lain E.G. Richardson, "H.264 and MPEG-4 video compression,"

Wiley, Inc., 2003.

11. C.Kim, H.-H.Shin and C.-C. J.Kuo, "Feature-based intra-prediction mode decision for H.264," in Proc. IEEE International Conference on Image Processing, 2004
12. X.Li, E.Q.Li, Y-K.Chen, "Fast multi-frame motion estimation algorithm with adaptive search strategies in H.264", in Proceedings of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, pp. 369-372, 2004
13. T.Toivonen and J.Heikkila, "Fast full search block motion estimation for H.264/AVC With multilevel successive elimination algorithm." University of Oulu, Finland, Oct. 2004.
14. 이웅호,이정호,조익환,정동석, "H.264의 고속 인트라 모드 결정을 위한 효과적인 인트라 모드,"한국통신학회논문지, vol.30, no.12C, 1144-1152, Dec.2005.
15. 이웅호,이정호,조익환,정동석, "다중 참조 영상의 적응적 선택 및 선택적 인트라 모드를 이용한 H.264/AVC 의 고속 모드 결정 방법", vol.31, no.3C, 271-278, March 2006.