

H.264 인터모드 결정 방법 개선에 관한 연구

A Study on an Improved H.264 Inter mode decision method

공재웅* ,정재진* ,황의성* ,김태형* ,김두영*

Jae-woong Gong* ,Jae-jin Jung* ,Eui-sung Hwang* ,Tae-hyoung Kim* ,Doo-young Kim*

요약

본 논문에서는 H.264 인코딩 과정 중 모드 결정 및 움직임 예측 부분의 개선을 위한 새로운 방법을 제안한다. 제안한 방법은 H.264 모드 결정 과정 중 참조 프레임의 선택 과정 생략과 모드 결정 기준의 하나인 SAD 연산과정의 개선을 사용하여 인코딩 과정에서 수행되는 연산량을 감소시켜 인코딩 수행시간을 절감하는 방법이다. 제안하는 방법의 성능 평가를 위하여 H.264 표준에서 제공되는 QCIF 사이즈, YUV 4:2:0 포맷의 영상을 이용하여 실험 영상을 만들어 적용하였다. 본 논문에서 제안한 방법을 적용한 결과 알고리즘 1은 약간의 화질 열화를 보이며 수행시간은 10프레임씩 인코딩 하였을 때 평균 4.7% 감소하였으며, 알고리즘 2는 약 0.98dB 정도의 화질 열화를 보이지만 인코딩 수행시간은 평균 9.6% 감소하였다.

Abstract

In this paper, we propose a new method for improving the H.264 encoding process and motion estimation part. Our approach is a method to reduce the encoding running time through the omission of reference frame in the mode selection process of H.264 and an improvement of SAD computing process. To evaluate the proposed method, we used the H.264 standard image of QCIF size and YUV 4:2:0 format. Experimental results show that proposed SAD algorithm 1 can improve the speed of encoding running time by an average of 4.7% with a negligible degradation of PSNR. However, SAD algorithm 2 can improve the speed of encoding running time by an average of 9.6% with 0.98dB degradation of PSNR.

Keywords : H.264, Encoding, Mode decision, Motion estimation, Video quality, Inter mode, Intra mode

I. 서론

디지털 동영상 정보는 정보량이 방대하므로 코딩 방식에 대한 많은 연구가 진행되어 왔으며, 특히 통신채널의 급속한 보급에 따라 압축률이 더욱 향상된 코딩 방식이 필요하게 되었다. 이와 같은 요구에 따라 H.264 표준이 탄생하였다.

H.264는 MPEG-4 Visual 보다는 좁은 범위를 갖고 주로 효율적이고 강인한 부호화와 사각형 비디오 화면의 전송을 지원하기 위해 설계되었다. H.264 원래의 목표는 H.263+와 MPEG-4 Visual 과 같은 이전의 표준에 대해 유사한 기능을 제공하는 것이었지만 매우 높은 압축율과 신뢰성 있는 전송에 대해 향상된 지원을 갖는다. 목표로 하는 응용은 양방향 비디오 통신(화상회의, 화상전화), 고화질 비디오에 대한 방송을 위한 부호화, 그리고 패킷 네트워크 상에서 비디오 스트리밍을 포함한다.

H.264는 네트워크 상에서 안정적인 전송을 위한 지원은 내장되어 있으며, 가능한 한 넓은 범위의 프로세서 플랫폼 상에서 구현이 용이하도록 설계되어 있다.[1]

H.264 표준은 기존의 비디오 부호화 방식들과 마찬가지로, 이미 부호화된 이전 화면으로 부터의 움직임을 추정하여 예측신호를 구성하고, 예측오류 신호를 이산 코사인 변환(DCT)하여 부호화하는 움직임 보상(Motion Compensation)과 DCT 변환 방법을 혼합하여 사용한다. 하지만, H.264 표준에서는 압축 성능을 높이기 위해 기존의 방식에 비해 더 작은 블록 사이즈인 4X4 블록단위까지 움직임 보상과 DCT변환을 수행하며, 주관적 화질의 향상을 위해 DCT를 이용하는 화상부호화 방식의 결점이었던 블록 경계의 왜곡을 억제하는 필터(Deblocking Filter)를 사용한다. 또한, DCT 양자화한 후의 신호를 엔트로피 부호화할 때 주위의 정보를 토대로 부호표 테이블을 바꾸는 등으로 화상의 성질에 맞게 부호화하여 압축 성능을 높인다. 이러한 기술들을 기반으로 H.264 비디오 부호화 표준은 기존의 비디오 부호화 표준인 MPEG-4 ASP에 비하여 2배의 압축 성능을 가지고 있다.[2] 그러나 이와 같은 가변 블록크기의 움직임 추정은 많은 양의 계산을 필요로 하므로, 압축효율은 높일 수 있으나 연산량이 증가하고 부호화 시간이 매우 길어지는 단점이 있다.

* 동아대학교 전기전자컴퓨터공학부
 접수일자 : 2008. 7. 16 수정 완료 : 2008. 10. 06
 게재확정일자 : 2008. 10. 29
 ※ 이 논문은 2006학년도 동아대학교 학술연구비(공모과제) 지원에 의하여 연구되었음.

이러한 문제를 극복하기 위한 연구가 활발히 진행되고 있고, 대표적인 것으로 고속 탐색 방식으로 다이아몬드 탐색 기법과 그 응용 기법들이 사용되고 있다.[3] H.264 움직임 추정을 위한 전영역 탐색 방식이나 고속 탐색 방식 모두 기본적으로 16개의 4X4 SAD(Sum of Absolute Differences) 계산을 필요로 한다.

본 논문에서는 H.264 부호화 과정 중 수행시간 비율이 50% 이상을 차지하고 있는 매크로 블록 모드 결정 방법 중 최적모드인 SKIP 모드를 결정하기 위한 계산 과정에서 활용되는 참조영상이 실험에 의한 통계 결과 과거 첫 번째 참조 프레임이 90% 이상 선택되는 것을 바탕으로 참조영상 선택과정 생략과 움직임 벡터와 참조영상간의 절대값 차의 합인 SAD 연산과정을 개선하여 인코딩 수행 시간을 줄이는 것을 제안하였다.

II. H.264 모드 결정 방법

2.1 인터모드와 인트라모드

H.264는 매크로 블록 모드를 위해 5개의 인터모드와 3개의 인트라모드를 지원한다. 이중 SKIP, 16X16, 16X8, 8X16, P8X8은 인터모드에 해당되며 움직임 예측을 통한 매크로 블록 부호화에 이용된다. P8X8 모드는 각각의 8X8, 8X4, 4X8, 4X4 중의 하나로 나누어질 수 있다. 그림 1은 H.264에서 사용하는 인터모드의 종류를 나타내고 있다.

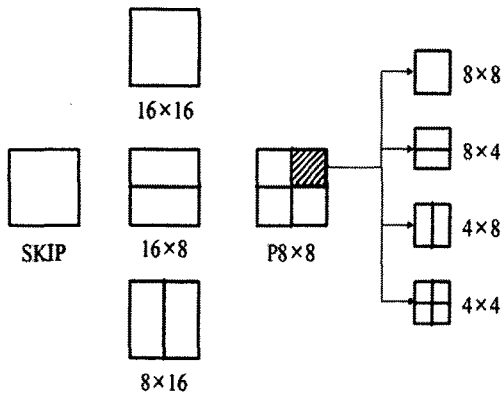


그림 1. H.264 인터 모드
Fig. 1. H.264 Inter mode

H.264는 인트라모드 예측을 위해 Intra4X4, Intra8X8, Intra16X16을 제공하여, 휘도 블록을 위한 Intra8X8 모드는 F-Rext 프로파일에서만 제공된다. 인트라모드에서는 이전에 부호화된 유효한 이웃 매크로 블록의 경계 화소를 이용해 현재 매크로 블록을 예측한다.

휘도 블록의 Intra4X4와 Intra8X8은 각각 4X4와 8X8 블록 단위로 총 9가지 예측 방향을 이용하여 예측을 수행하고, Intra16X16은 16X16 블록 단위로 총 4가지 예측 방향을 이용하여 현재 매크로 블록을 예측한다. 이렇게 예측된 인터모드와 인트라모드들 중에서 비트를 왜곡 비용이 가장

작은 모드가 최종 매크로 블록 모드로 결정된다. 인터모드에서 색차 신호는 휘도 신호의 움직임 벡터를 이용해 예측되지만, 인트라모드에서 색차 신호는 휘도 신호와 별도로 예측된다. 인트라모드에 대한 색차 신호 예측은 8X8 블록 단위로 이루어지며, 각 8X8 블록당 4가지 예측 방향을 제공한다. 그림 2에는 인트라 4X4 예측 방법을 보여 주고 있다.[4]

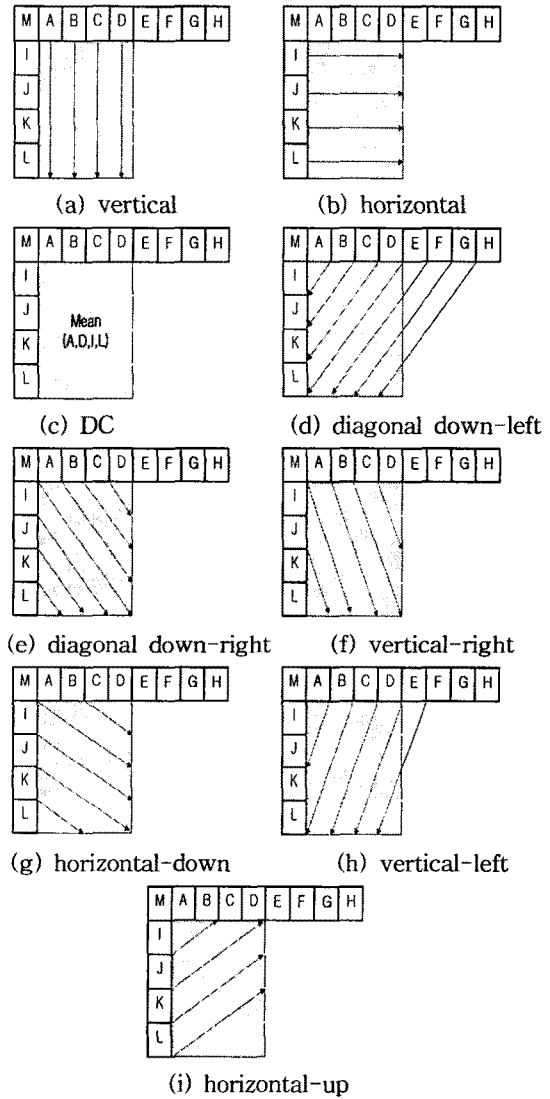


그림 2. 인트라 4X4 예측방법

- (a) 수직부, (b) 수평부, (c) DC,
- (d) 비스듬히 아래 왼쪽, (e) 비스듬히 아래 오른쪽,
- (f) 수직 오른쪽, (g) 수평 아랫쪽, (h) 수직 왼쪽,
- (i) 수평 윗쪽

Fig. 2. Intra 4X4 prediction

- (a) vertical, (b) horizontal, (c) DC,
- (d) diagonal down-left, (e) diagonal down-right,
- (f) vertical-right, (g) horizontal-down, (h) vertical-left,
- (i) horizontal-up

2.2 비트율-왜곡 최적화 기법

비트율-왜곡 최적화 기법에서는 각 매크로 블록 모드에 대한 비트율-왜곡 비용을 계산한다. 인터모드에서는 이 값을 계산하기 전에 움직임 벡터와 참조영상을 결정한다. 가변 블록에 대한 움직임 벡터와 참조영상은 식 (1)의 최소값으로 결정된다.

$$J_{motion}(MV, REF | \lambda_{motion}) = SAD(s, r(MV, REF)) + \lambda_{motion} \cdot R(MV, REF) \quad (1)$$

여기서 λ_{motion} 은 라그랑지안 계수로, 양자화 계수에 의존하는 값을 갖는다. $R(MV, REF)$ 는 움직임 벡터와 참조영상을 부호화하는데 필요한 비트 수로 표 참조 연산에 의해 얻어진다. $SAD(s, r(MV, REF))$ 는 원 영상과 움직임 예측에 의해 움직이 보상된 영상과의 차에 대한 절대값의 합이고, 블록에 대하여 식 (2)와 같이 정의된다.

$$SAD(s, r(MV, REF)) = \sum_{x \in H, y \in V} |s(x, y) - r(x - m_x, y - m_y)| \quad (2)$$

여기서 s 와 r 은 각각 원 블록과 복호된 블록의 화소 값을 나타낸다. H 와 V 는 각각 가변 블록의 넓이와 높이이며, m_x 와 m_y 는 움직임 벡터이다. P8X8 모드에서 각 8X8 블록에 대한 최적의 하위 매크로 블록 모드의 결정, 인트라 모드에서 최적의 예측 방향 결정과 매크로 블록에 대한 최적 모드 결정은 식 (3)의 최소값에 의해 결정된다.

$$J_{mode}(s, r, M | \lambda_{mode}) = SSD(s, r, M) + \lambda_{mode} \cdot R(s, r, M) \quad (3)$$

여기서 λ_{mode} 는 λ_{motion} 의 제곱값이다. M 은 매크로 블록 모드, 예측방향 모드 또는 하위 매크로 블록 모드이다. $R(s, r, M)$ 은 M 에 해당하는 모드로 부호화했을 때, 실제로 발생하는 비트 수며, H.264 참조 소프트웨어에서는 이 값을 계산하기 위해 모든 부호화 과정을 수행한다. $SSD(s, r, M)$ 는 원 영상과 복호된 영상과의 차이에 대한 제곱의 합이고 식 (4)와 같이 계산된다.

$$SSD(s, r, M) = \sum_{x \in H, y \in V} (s(x, y) - r(x - m_x, y - m_y))^2 \quad (4)$$

P8X8 모드의 하위 모드결정과 인트라모드에서 예측방향 결정의 경우에 식 (1)과 식 (3)은 휘도 성분만을 고려하여 계산하고, 최종 모드결정을 위한 경우는 휘도와 색차 성분 모두에 대해서 계산을 수행한다.[6]

2.3 H.264 부호기의 복잡도

그림 3는 H.264 부호기에서 주요 모듈 별 수행시간의 비율을 나타낸다.

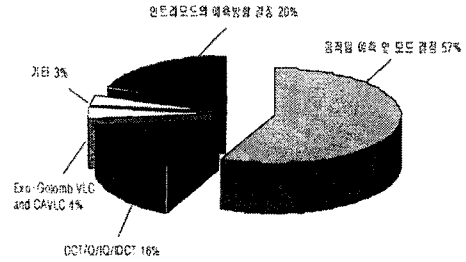


그림 3. H.264 부호기의 주요 모듈 별 수행시간 비율
Fig. 3. H.264 Encoder run-time ratio

그림 3에서 보듯이, 비용 생성을 위한 변환과 모드 결정 과정이 많은 부분을 차지하는 것을 알 수 있다. 모드 결정 과정에서 사용되는 식 (3)의 발생 비트 수를 계산하기 위해서는 각 모드에 대해 DCT/Hadamard 변환 및 양자화/역양자화, 역 DCT/Hadamard 변환과정, 그리고 엔트로피 부호화 과정을 거치게 된다. 엔트로피 부호화 과정에서 사용되는 Exp-Golomb 코드와 CAVLC, CABAC은 다소 복잡한 계산 과정을 필요로 하지만, 실제 부호기에서 차지하는 복잡도는 그렇게 크지 않다. 따라서, 부호기의 복잡도를 줄이기 위해서는 고속 모드결정 방법의 개발이 필수적이다. 특히, 고속 모드결정 방법의 핵심은 단순히 최적모드 결정 과정의 복잡도를 줄이는 것 외에 실제로 가장 큰 부호화 시간을 차지하는 움직임 예측 과정의 복잡도를 줄이는 것이다.[7]

2.4 H.264의 모드결정 과정

그림 4는 H.264 참조 소프트웨어 JM 10.2의 모드 결정 과정을 나타낸다. (A)는 인터모드를 위한 움직임 벡터와 그때의 참조영상이 저장된다.

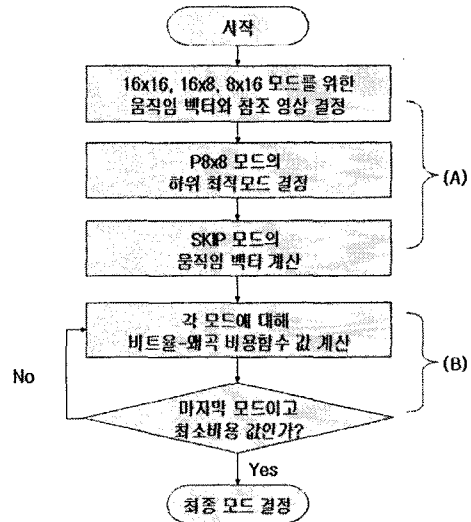


그림 4. 최종 모드 결정 방법의 흐름도
Fig. 4. Best-mode decision flowchart

P8X8 모드의 하위 최적모드를 결정하는 과정은 8X8과, 8X4, 4X8, 4X4 모드에 대해 움직임 벡터와 참조영상을 결정하는 과정과 이 값들을 이용하여 식 (3)를 이용하여 최적모드를 결정하게 된다. 이 과정에서는 인터모드들을 포함하게 되며 모든 모드들에 대해서 비용함수 값이 최소가 되는 모드가 최적모드가 된다.

H.264 참조 소프트웨어 JM 10.2에서는 식 (3)을 사용하여 최적모드를 결정하고 복잡도 모드(High Complexity Mode)외에 식 (1)의 계산 결과가 최소값을 가지는 경우 최적화 모드를 선택하는 부최적화 모드를 지원하는데, 이것은 참조 소프트웨어에서 비트율-왜곡 최적화 옵션을 저 복잡도 모드(Low Complexity Mode)로 설정했을 때 동작한다.

이 옵션은 설정파일에서 다음과 같은 부분을 수정하면 된다.

```

RDOptimization = 1 # rd-optimized mode decision
# 0 : RD-off (Low Complexity Mode)
# 1 : RD-on (High Complexity Mode)
# 2 : RD-on
# 3 : with losses
(Fast High Complexity Mode)
    
```

저 복잡도 모드는 고 복잡도 모드에 비해 적은 부호화 시간을 필요로 하지만, 실제 발생 되는 비트 수를 고려하지 않기 때문에 고 복잡도 모드로 부호화 했을 때 보다 발생 비트율 측면에서 좋지 않은 성능을 보인다.[8]

2.5 고속 모드 결정 방법

H.264 참조 소프트웨어 JM 9.4 버전부터 구현되어 있는 고속 모드 결정 방법은 조기 SKIP 모드결정을 통해 불필요한 계산을 생략한다. 다음과 같은 4가지의 조건을 모두 만족 할 때 SKIP 모드로 결정되며, 최적모드가 SKIP 모드로 결정되면 이하 모든 모드 결정 과정은 생략된다.

- 1) 움직임 보상 블록이 16X16
- 2) 참조영상은 바로 이전 영상
- 3) 16X16 모드의 움직임 벡터와 SKIP 모드의 움직임 벡터가 같음
- 4) 잔여 데이터의 비트 수가 0 (CBP=0)

실제 래퍼런스 소프트웨어 코드에서는 16X16 모드에 대한 움직임 예측을 수행하여 움직임 벡터와 참조영상을 결정한 후, 비트율-왜곡치를 계산한다. 이 계산과정을 통해 나온 결과가 2), 3), 4) 조건을 만족할 때, SKIP 모드로 결정된다. 최적모드가 SKIP 모드로 결정되면 이하 모든 모드 결정 과정은 생략된다.

SKIP 모드는 움직임 벡터가 주변의 이웃한 3개의 매크로 블록으로부터 예측된 움직임 벡터와 같은 16X16 모드의 한 경우로 생각할 수 있다. SKIP 모드의 비트율-왜곡치

는 움직임 예측 과정, 참조영상 결정, 잔여 데이터의 계산 과정 없이 움직임 벡터를 주변의 블록으로부터 예측된 움직임 벡터로 가정하고, 비트율-왜곡치를 계산하여 다른 모드와 비교한다.

H.264 표준에서 SKIP 모드는 잔여 데이터의 비트 수가 0 이어야 하지만, 실제 SKIP 모드의 비트율-왜곡치 계산 결과 비트 수가 0이 아니어도 다른 모드에 비해 그 값이 작다면 SKIP 모드로 결정될 수 있다. 따라서 16X16 모드의 비트율-왜곡치 계산 결과로부터 얻어진 네 번째 조건은 초기에 SKIP 모드를 결정하는데 충분한 조건을 제공한다. CBP(Coded Block Pattern)는 8X8 크기의 휘도 블록 4개와 색차 블록 2개로 이루어진 매크로 블록에서 어떤 블록이 0 이 아닌 DCT 변환값들을 가지고 있는지를 타나낸다. 따라서 CBP 값은 잔여 데이터의 DCT 계수들의 양자화된 값들이 모두 0일 때 0의 값을 갖는다.[9]

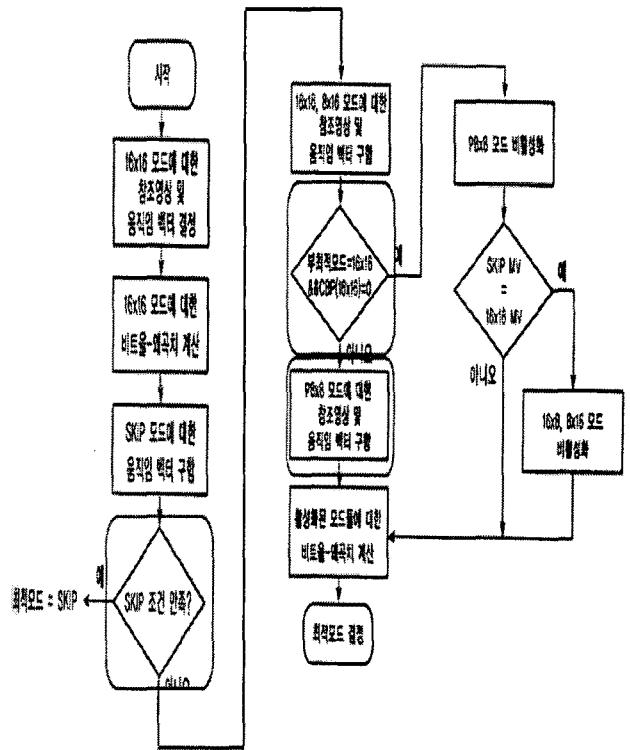


그림 5. 모드결정 방법의 흐름도
Fig. 5. The flowchart of mode decision method

그림 5는 가변 부호화 비트율을 위한 고속 인터모드 결정 방법의 흐름도를 보여주고 있다. 그림 5에서 보는 바와 같이, 처음에 16X16 모드에 대해서 참조영상 및 움직임 벡터를 결정한 후, 비트율-왜곡치를 계산하고 조기 SKIP 모드 결정 조건을 만족하는지 검사한다. 만약, 조기 SKIP 모드 조건이 만족되면 16X8과 8X16, P8X8 모드에 대한 참조영상 결정 및 움직임 벡터 계산을 생략하기 때문에 효율적으

로 부호화 시간을 단축할 수 있다.

조기 SKIP 모드 조건을 만족하지 않더라도 16X16, 16X8, 8X16 모드 중 부최적모드가 16X16이고 CBP가 0의 값을 가지면, 현재 부호화되는 매크로 블록은 영상에서 배경과 같은 복잡한 부분이 아니며 움직임이 일정하거나 거의 없는 부분이라고 생각할 수 있다. 그렇지 않다고 하더라도 양자화 계수값이 크다면 DCT 계수들의 값이 작아서 CBP의 값이 0이 나오는 경우가 증가하기 때문에 매크로 블록을 작은 단위의 블록으로 나눌 필요가 없다. 따라서 P8X8 모드에 대한 세부 블록 모드 결정 과정에 필요한 참조영상 결정 및 움직임 벡터 계산을 생략한다. 이 과정을 통해 양자화 계수값이 큰 경우에 SKIP이나 16X16 모드의 발생빈도가 증가하는 경우를 고속 모드 결정 과정에 반영할 수 있다.

P8X8 모드를 비활성화하고 SKIP 모드의 움직임 벡터가 16X16 모드의 움직임 벡터와 같다면, 16X8 과 8X16의 모드 또한 최적모드 결정을 위한 비트율-왜곡치 계산에서 배제한다. CBP의 값이 0이면서 이 조건을 만족하면 조기에 16X16 모드를 최적 인터모드로 결정하고 인트라모드와 비교하게 된다. 이러한 조건들을 모두 검사한 후, 마지막으로 인트라모드를 포함한 활성화된 인터모드들에 대해서 비트율-왜곡치를 계산하여 최적모드를 결정하게 된다.

III. 제안 알고리즘

3.1 참조 프레임 선택

H.264의 표준에서 모드 결정을 하기 위해서는 과거 프레임 2개와 미래의 프레임 2개에서 각각 SAD(Sum of Absolute Difference) 연산을 통하여 비용이 최소가 되는 프레임을 참조 프레임으로 결정을 하게 되고 SAD의 결과 값을 이용하여 인터모드 및 인트라모드를 결정하게 된다.

실험결과 선택되는 참조 프레임이 과거 첫 번째 프레임의 선택이 평균 90% 이상이 되는 것을 확인 할 수 있었는데 예를 들면 foreman 영상의 경우 과거 첫 번째 프레임이 87%, 과거 두 번째 프레임이 6%, 미래 첫 번째 프레임이 5%, 미래 두 번째 프레임이 2% 선택되는 것을 볼 수 있었다. 따라서 본 논문에서는 모드 결정에 사용되는 비용 감소를 위해 과거 첫 번째 프레임만 SAD 연산을 하도록 참조 소프트웨어의 코드를 수정하였다.

표 1은 QCIF 사이즈의 foreman, akiyo, carphone, container 영상에 각각 10 프레임씩 인코딩하여 참조 프레임 선택 결과를 백분율로 나타낸 것이다.

표 1. 최적 기준 프레임 분포(%)

Table 1. Distribution of optimal reference frame(%)

영상	과거 1	과거 2	미래 1	미래 2
foreman	87	6	5	2
akiyo	92	4	2	2
carphone	86	8	4	2
container	95	3	1	1
평균	90	5.25	3	1.75

본 논문에서 실험결과를 바탕으로 참조 소프트웨어의 코드 중 참조 프레임 결정과정 부분을 과거 첫 번째 프레임을 선택하도록 제한, 수정하였다.

3.2 SAD 연산 알고리즘1

2	1	3	4	2	3	3	2
9	7	8	10	4	3	5	5
5	4	3	2	3	6	5	4
1	1	1	1	3	3	3	3

(a) Reference (b) Current

그림 6. SAD 예

(a) 참조 블록, (b) 현재 블록

Fig. 6. SAD Example

(a) Reference, (b) Current

현재 블록과 참조 블록의 SAD 연산을 H.264 표준에서 사용하는 결과는 아래와 같다.

$$|2-2|+|1-3|+|3-3|+|4-2|+\dots+|1-3| = 40 \quad (5)$$

알고리즘1에서는 SAD의 연산을 참조 블록과 현재 블록 간의 전체 인덱스에 대한 연산을 수행하지 않고 현재 블록의 인덱스 증가를 x방향은 1씩, y방향은 2씩 증가하여 절반의 계산을 유도 하고, 나머지 계산을 하지 않는 부분은 이전 블록의 계산 결과 값을 그대로 사용하는 방법을 제안 하였다.

제안한 방법으로 계산된 SAD의 결과는 아래와 같다.

$$|2-2|+|1-3|+|3-3|+|4-2| \quad (6)$$

+ 이전 블록의 계산 값

$$+|5-3|+|4-6|+|3-5|+|2-4|$$

+ 이전 블록의 계산 값 = 39

3.3 SAD 연산 알고리즘2

2	1	3	4	2	3	3	2
9	7	8	10	4	3	5	5
5	4	3	2	3	6	5	4
1	1	1	1	3	3	3	3

(a) Reference

(b) Current

그림 7. Cross sample SAD 예

(a) 참조 블록, (b) 현재 블록

Fig. 7. Cross sample SAD Example

(a) Reference, (b) Current

알고리즘2에서는 SAD의 연산을 참조 블록과 현재 블록 간의 주요 성분 배치 영역인 Cross 지점의 데이터를 연산 대상으로 하였다. Cross 지점의 설정은 좌상부에서 우하향으로 우상부에서 좌하향으로 설정하였으며 나머지는 계산을 하지 않는 방법을 제안하였다.

Cross sample로 계산된 SAD의 결과는 아래와 같다.

$$|2-2|+|7-3|+|3-5|+|1-3|+\dots+|4-2| = 17 \quad (7)$$

하나의 알고리즘 내에서의 결과 값들 간의 수치비교는 의미가 있지만, 서로 다른 알고리즘간의 결과 값 비교는 의미가 없다.

IV. 실험 결과

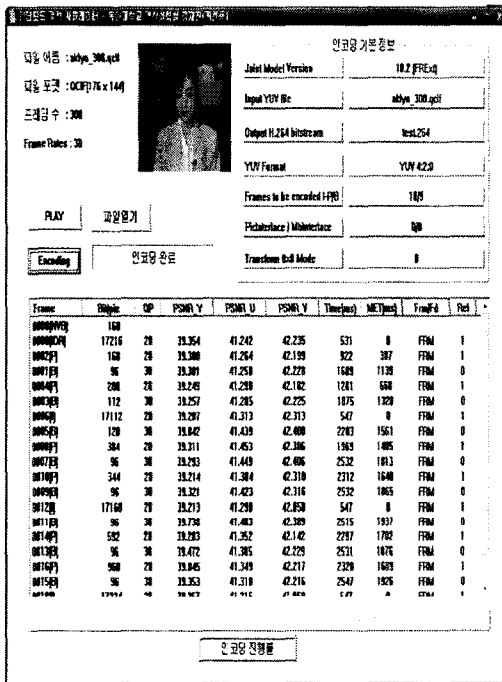


그림 8. GUI 환경의 인터 모드 결정 시뮬레이터
Fig. 8. Inter mode decision simulator for GUI

본 논문에서는 제안한 방법을 실험하기 위하여, Intel Pentium4 3.00 GHz PC의 Microsoft Windows XP 환경, H.264 참조 소프트웨어인 JM10.2를 사용하였고 프로파일은 MainProfile, 레벨은 1.3을 적용하여 실험하였다. 또한 실험의 편의성을 위하여 콘솔 테스트 환경인 JM10.2를 Microsoft WIN32 API를 적용하여 GUI 환경의 시뮬레이터를 제작하였다.

실험에 사용된 영상은 QCIF 사이즈와 YUV 4:2:0 포맷의 foreman, akiyo, container, carphone, hall monitor 의 영상을 각각 10 프레임씩 사용하였다.

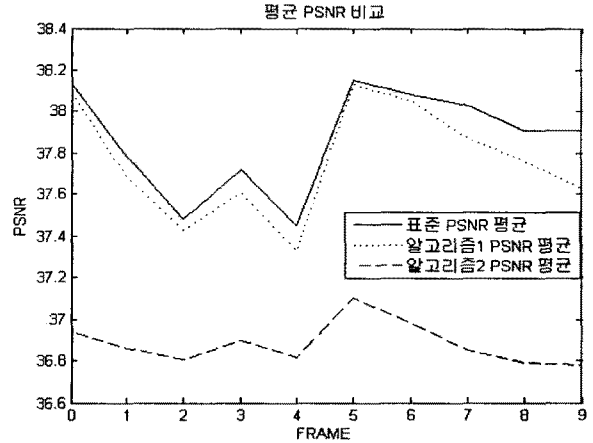


그림 9. 제안한 방식과 표준 방식의 평균 PSNR 비교
Fig. 9. The result of standard and proposed encoding comparison PSNR (average)

표 2. 제안한 방식과 표준 방식의 평균 PSNR 비교
Table 2. The result of standard and proposed encoding comparison PSNR (average)

프레임 NO. / 영상	1	2	3	4	5	6	7	8	9	10
표준	38.1	37.7	37.4	37.7	37.4	38.1	38.0	38.0	37.9	37.9
알고리즘1	38.0	37.6	37.4	37.6	37.3	38.1	38.0	37.8	37.7	37.6
알고리즘2	36.9	36.8	36.8	36.9	36.8	37.1	36.9	36.8	36.7	36.7

그림 9와 표 2에서 보듯이 표준 H.264는 평균 PSNR 값이 37.86dB, 알고리즘 1의 평균 PSNR 값은 37.76dB로 화질면에서 거의 차이가 없는 것을 알 수 있다. 알고리즘 2의 평균 PSNR 값 36.88dB로 표준에 비해 화질 열화가 발생하는 것을 볼 수 있다. 그리고 제안한 알고리즘의 특성상 비트량 변화는 거의 발생하지 않았다.

표 3. H.264 표준 인코딩 수행시간(ms)
Table 3. H.264 standard encoding time(ms)

프레임 NO. / 영상	1	2	3	4	5	6	7	8	9	10	평균
akiyo	547	1000	1735	1359	2078	578	2438	2094	2828	2469	1712.6
carphone	640	1094	1813	1437	2172	609	2547	2172	2891	2547	1792.2
container	641	1094	1796	1422	2140	640	2500	2157	2875	2531	1779.6
foreman	625	1078	1796	1438	2188	640	2562	2188	2922	2594	1803.1
hall monitor	625	1078	1750	1375	2156	609	2500	2141	2813	2500	1754.7

표 4. 알고리즘1 인코딩 수행시간(ms)
Table 4. Algorithm 1 encoding time(ms)

프레임 NO. 영상	프레임										평균
	1	2	3	4	5	6	7	8	9	10	
akiyo	531	922	1609	1281	1875	547	2203	1969	2532	2312	1578.1
carphone	625	1047	1703	1390	2015	610	2375	2109	2719	2469	1706.2
container	625	1031	1656	1359	2109	625	2453	2078	2672	2921	1752.9
foreman	610	1032	1672	1422	2031	609	2469	2156	2891	2516	1740.8
hall monitor	609	1000	1640	1344	1968	610	2281	2031	2609	2375	1646.7

표 3과 표 4에서 보듯이 본 논문에서 알고리즘 1은 H.264 표준 인코딩 결과에 비해 인코딩 수행시간이 평균 4.7% 감소함을 볼 수 있다.

표 5. 알고리즘2 인코딩 수행시간(ms)
Table 5. Algorithm 2 encoding time(ms)

프레임 NO. 영상	프레임										평균
	1	2	3	4	5	6	7	8	9	10	
akiyo	512	619	617	1021	1650	1353	2103	618	2445	2073	1301.1
carphone	620	1040	1699	1383	2007	603	2369	2102	2718	2464	1700.5
container	613	1025	1649	1354	2101	619	2443	2071	2661	2914	1745
foreman	603	1021	1663	1413	2023	603	2460	2146	2880	2509	1732.1
hall monitor	602	994	1635	1336	1957	602	2270	2024	2598	2365	1638.3

표 3과 표 5에서 보듯이 본 논문에서 알고리즘 2은 H.264 표준 인코딩 결과에 비해 인코딩 수행시간이 평균 9.6% 감소함을 볼 수 있다.

V. 결론

본 논문에서는 H.264 인코딩 수행과정 중 높은 비율을 차지하고 있는 인터모드 결정방법 중 실험결과를 통하여 과거 첫 번째 프레임의 참조 프레임 결정 비율이 90% 이상이 되는 것을 바탕으로 참조 프레임 선택과정의 생략과 모드 결정 조건 중 SAD 연산 과정을 개선하여 인코딩 수행시간을 줄이도록 제안하였다.

QCIF 사이즈, YUV 4:2:0 포맷의 영상 5개의 각각 10프레임씩 인코딩하여 실험한 결과 본 논문에서 알고리즘 1은 H.264 표준 인코딩 결과에 비해 인코딩 수행시간이 평균 4.7% 감소한 반면 화질열화는 표준 H.264는 평균 PSNR 값이 37.86dB, 알고리즘 1의 평균 PSNR 값은 37.76dB로 평균 약 0.11dB 정도의 화질 열화가 발생하였으나 육안으로 살펴본 결과 화질면에서 거의 차이가 없는 것을 알 수 있었다. 알고리즘 2은 H.264 표준 인코딩 결과에 비해 인코딩 수행시간이 평균 9.6% 감소한 반면 평균 PSNR 값은

36.88dB로 평균 약 0.98dB 정도의 화질 열화가 발생한 것을 볼 수 있었다. 그리고 제안한 알고리즘의 특성상 비트량 변화는 거의 발생하지 않았다.

참고 문헌

- [1] ITU-T Recommendation H.264(2003), "Advanced Video Coding for generic audiovisual services" | ISO/IEC 14496-10:2003, "Information technology, Coding of audio-visual objects-Part 10:Advanced vide coding", (2003).
- [2] Ian E. G. Richardson, "H.264 and MPEG-4 Video Compression", Wiley, p.163, (2003).
- [3] Donghyung Kim, Jechang Jeong, Xiao Song and Chengke Wu, " A Fast Macroblock Mode Selection Algorithm In The H.264/AVC Standard", IWAIT 2005, pp.157-162, Jan. 2005
- [4] 정제창, "H.264/AVC 비디오 압축 표준", 홍릉과학출판사, p.99 (2005)
- [5] Ian E. G. Richardson, "H.264/MPEG-4 Part 10 White Paper: Reconstruction Filter", May 30 (2003).
- [6] Ian E. G. Richardson, "H.264/MPEG-4 Part 10 White Paper: Variable-Length Coding", October 17 (2002).
- [7] G. Sullivan, T. Wiegand, "Rate-Distortion Optimization for Video Compression", IEEE Signal Processing Magazine, p.84, November (1998).
- [8] Feng Pan, Xiao Lin, Susanto Rahardja, Keng Pang Lim, Z. G. Li, Dajun Wu, and Si Wu, "Fast mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding", IEEE transactions on Circuits and Systems for Video Ttechnology, vol. 15, no. 7, July 2005.
- [9] C.-C. Cheng and T.-S. Chang, "Fast three step intra prediction algorithm for 4X4 blocks in H.264", Proc. IEEE ISCAS, vol. 2, pp 1509-1512, 2005
- [10] 이영주, 정제진, 공재웅, 김두영, "H.264 인터모드 결정 방법에 관한 연구", 한국 신호처리 시스템 학회 2008 하계 학술대회 논문집 vol. 9, no. 1, June 2008.
- [11] Dongming Zhang, Yanfei Shen, Shouxun Lin, Yongdong Zhang, "Fast inter frame encoding based on modes pre-decision in H.264", IEEE International Conf. Multimedia and Expo, p.4, July (2005).
- [12] Zhang Fan, Zhang Xudong, "Fast macroblock mode decision in H.264", IEEE International Conf. Signal Processing, Vol. 2, pp. 1179 - 1182, Aug (2004).



공 제 응(Jae-woong Gong)

2004년 2월 : 한국해양대학교
제어계측공학과 졸업
2006년 9월 : 동아대학교 전자공학과
공학석사
2007년 9월 ~ 현재 : 동아대학교
전자공학과 박사과정 중

<관심분야> 영상처리 및 컴퓨터 비전, 신경망



정 재 진(Jae-Jin Jung)

2005년 2월 : 동명정보대학교
컴퓨터공학과 졸업
2007년 2월 : 동아대학교 전자공학과
공학석사

<관심분야> 영상처리 및 컴퓨터 비전, 신경망



김 태 형(Tae-hyoung Kim)

1997년 2월 : 동아대학교 전자공학과
졸업
1999년 2월 : 동아대학교 전자공학과
공학석사
2001년 2월 : 동아대학교 전자공학과
박사수료

<관심분야> 영상처리 및 컴퓨터 비전, 웨이브렛 변환,
디지털워터마킹



김두영(Doo-Young Kim)

1972년 : 부산대학교 전자공학과 공학사
1975년 : 부산대학교 전자공학과
공학 석사
1983년 : 경북대학교 전자공학과
박사수료

1977년 ~ 현재: 동아대학교 전자공학과 교수
1998년 ~ 1999년: 대한전자공학회 부산경남지부장
2000년 ~ 2007년: 한국신호처리·시스템 학회장
2008년 ~ 현재: 한국신호처리·시스템 학회 명예회장
<관심분야> 디지털 신호처리, 신경망,
영상처리 및 컴퓨터 비전
