

H.264의 인터/인트라 예측을 위한 적응적 고속 알고리즘

준회원 진 순 중*, 정회원 김 원 기**, 정 제 창**

Adaptive Fast Algorithm for Inter/Intra Prediction in H.264

Soonjong Jin* Associate Member, Wonki Kim**, Jechang Jeong** Regular Members

요 약

가장 최근에 표준화된 H.264 동영상 압축 부호화 방식은 이전의 비디오 부호화 방식에 비해 율-왜곡 관점에서 높은 성능을 보이지만 더불어 부호기의 복잡도가 상당히 증가하는 단점이 있다. 본 논문에서는 H.264 비디오 부호화에서 인터/인트라 예측을 위한 적응적이고도 효율적인 복잡도 감소 알고리즘을 제시한다. 제안하는 인터 예측 알고리즘은 효율적이고 간단한 모드 결정 방법을 제시하며 그에 따른 움직임 추정 기법을 적응적으로 적용하여 복잡도를 감소시킨다. 또한 인트라 예측에서 기존에 계산된 최소 율-왜곡 비용을 인트라 16×16과 인트라 4×4 모드 결정에 이용하여 복잡도를 줄이는 알고리즘을 제안한다. 제안하는 알고리즘은 전체 부호화 시간을 최대 60.9% 까지 감소시키는데 비해 PSNR이나 비트율에는 거의 손실을 발생시키지 않는 장점이 있다.

Key Words : H.264, mode decision, prediction, fast algorithm, complexity reduction

ABSTRACT

The latest standard H.264 coding methods show better efficiency compared to the former coding standards from the rate distortion point of view, but also lead to high complexity. In this paper, we propose an adaptive and fast inter/intra prediction algorithm for H.264 video coding. In the inter prediction algorithm, an effective and simple mode selection method is proposed, which can adaptively adjust to motion estimation to reduce complexity. And proposed intra prediction algorithm reduces its complexity by using calculated Minimum Rate-Distortion costs in the prediction of Intra 16×16 and 4×4 modes. From experiment using JM8.4, our algorithm can reduce about 60.9% of computation time as compared to JM, with only negligible PSNR and bitrate degradation.

I. 서론

H.264/AVC 표준은 ITU-T의 VCEG(Video Coding Experts Group)과 ISO/IEC의 MPEG (Moving Picture Expert Group)의 공동 작업으로 제정되었다 [1]. H.264/AVC 비디오 부호화 방식은 이전의 비디오 부호화 방식들(H.261, MPEG-1, MPEG-2,

MPEG-4 Part 2, H.263)에는 없는 새로운 부호화 기법들이 추가되었다[2-3].

본 논문은 매크로블록 모드 선택 과정에 있어서 인터/인트라 예측에 관한 적응적인 고속 알고리즘을 제안한다. 먼저 인터 예측에서는 평탄한 영역과 복잡한 영역에 따라 적응적으로 움직임 추정 방법을 사용함으로써 복잡도를 감소시킨다. H.264에 포함되

* 한양대학교 전자통신컴퓨터공학과 영상통신 및 신호처리 연구실 (Soonjong.Jin@gmail.com)

** 한양대학교 전자통신컴퓨터공학부 (jjeong@ece.hanyang.ac.kr)

논문번호 : KICS2005-09-372, 접수일자 : 2005년 9월 14일

어 있는 고속 전역 탐색(Fast Full Search) 방법과 전역 탐색(Full Search) 방법은 각각 나름대로의 장 단점을 가지는데 현재의 매크로블록의 주변 블록 모드를 이용하여 현재 매크로블록의 모드를 예측할 수 있으며 예측된 모드에 따라서 적응적인 움직임 추정이 가능하다. 이러한 특징을 이용하여 인터 예측에 대한 적응적인 방법을 제시한다.

또한 인트라 예측에서는 기존의 다른 모드에서 계산된 최소 율-왜곡 비용(Minimum Rate-Distortion Cost)을 이용하여 인트라4×4 모드와 인트라16×16 모드의 복잡도를 감소시키는 알고리즘을 제안한다. 대부분의 영상은 일반적으로 인트라 예측에 대한 비용이 인터 예측에 대한 비용보다 크다. 이러한 특징을 이용하여 인트라 예측에 대한 효율적인 방법을 제시한다. 제안하는 인트라 예측 방법은 기존의 모드를 예측하여 율-왜곡 비용 함수 계산을 줄이는 기존의 방법과는 달리 PSNR이나 비트율의 변화는 거의 없이 단지 계산 속도만을 향상시킨다.

본 논문의 2장에서는 참조 소프트웨어(JM 8.4)에 있는 매크로 블록 모드 선택 알고리즘에 대해서 기술하고 3장에서 참조 소프트웨어의 인터 / 인트라 예측 알고리즘을 기술한다. 4장에서는 제안하는 인터 / 인트라 예측에 대한 복잡도 감소 알고리즘에 대해서 각각 설명한다. 5장에서 여러 테스트 영상에 적용한 결과를 보여주고 6장에서 결론을 맺는다.

II. 참조 소프트웨어의 매크로블록 모드 선택 알고리즘

I 프레임 내의 매크로블록은 인트라16×16 또는 인트라4×4 모드 중에서 선택된다. 반면에 P 프레임 내의 매크로블록은 SKIP, 16×16, 16×8, 8×16, 8×8, 인트라16×16, 인트라 4×4 모드 중에서 선택된다. 율-왜곡 최적화 기법이 사용될 경우 매크로블록 모드의 선택은 모든 가능한 모드의 율-왜곡 비용 함수를 계산해서 최소 율-왜곡 비용을 가진 모드가 현재 매크로블록의 최종 모드로 선택된다. 이러한 다양한 매크로블록 모드와 율-왜곡 비용 함수의 사용은 H.264 부호기의 복잡도의 주된 원인이 된다.

2.1 매크로블록 모드의 정의

P 프레임 내의 매크로블록 모드들 중에서 인터 예측을 위한 모드인 SKIP, 16×16, 16×8, 8×16, 8×8 모드를 각각 SKIP, Mode1, Mode2, Mode3, Mode8로 정의하고, Mode8내의 선택 가능한 부-매

크로블록 모드인 8×8, 8×4, 4×8, 4×4 모드를 각각 Mode4, Mode5, Mode6, Mode7이라고 정의한다. 또한 인트라 예측을 위한 인트라16×16, 인트라4×4 모드는 각각 I16MB, I4MB로 정의한다.

2.1.1 인터 예측을 위한 매크로블록 모드

각각의 인터 예측 모드에 대한 블록 모양 및 블록의 상대적인 크기는 그림 1과 같다.

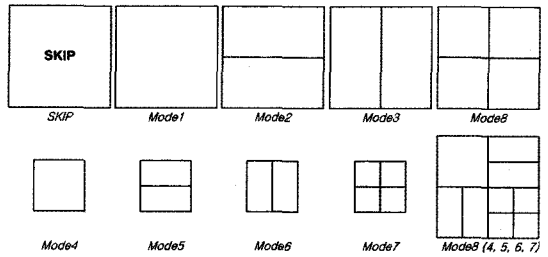


그림 1. 인터 예측의 매크로블록 모드 및 부-매크로블록 모드

그림 1에서 Mode8이 선택될 경우 8×8 크기의 부-매크로블록으로 나뉘며 각각의 부-매크로블록 모드는 Mode4, Mode5, Mode6, Mode7중에서 선택될 수 있다. 이때 하나의 매크로블록 내의 4개의 부-매크로블록 모드는 독립적으로 선택된다.

$$\text{Mode8}(\text{SubMB0}, \text{SubMB1}, \text{SubMB2}, \text{SubMB3}) \\ = \text{Mode8}(4|5|6|7, 4|5|6|7, 4|5|6|7, 4|5|6|7) \quad (1)$$

식 (1)은 4개의 부-매크로블록 모드를 갖는 Mode8의 표현식이며 (4|5|6|7)의 표현은 각각의 부-매크로블록 모드가 Mode4, Mode5, Mode6, Mode7 중에서 선택 가능함을 나타낸다. 그림 1의 가장 오른쪽 하단의 그림은 Mode8내의 각 부-매크로블록에 각각 Mode4, Mode5, Mode6, Mode7이 선택된 경우의 매크로블록 모양을 나타내며 이에 대한 수식적 표현은 식 (2)와 같이 정의한다.

$$\text{Mode8}(\text{SubMB0}, \text{SubMB1}, \text{SubMB2}, \text{SubMB3}) \\ = \text{Mode8}(4, 5, 6, 7) \quad (2)$$

SKIP의 경우 움직임 벡터가 예측된 움직임 벡터와 동일하고 화소의 예측 오차가 충분히 작은 경우 선택되며 부호기는 매크로블록의 모드만을 전송한다. Mode1의 경우 하나의 움직임 벡터만을 가지며 16×16 크기의 평탄한 영역에서 효율적인 모드이다. Mode2는 16×8 크기의 블록 단위로 움직임을 추정하며 두 개의 움직임 벡터를 가진다. Mode2는 가로 방향의 객체가 걸쳐 있는 경우에 효율적인 모드가

된다. *Mode3*은 8×16 크기의 블록 단위로 움직임 추정하며 역시 두 개의 움직임 벡터를 갖는다. *Mode3*은 객체가 세로 방향으로 걸쳐 있는 경우 효율적인 모드이다. *Mode8*의 움직임 벡터의 개수는 부-매크로블록의 모드에 따라 달라지며 4~16개의 움직임 벡터를 가질 수 있다. *Mode8*은 보다 세밀한 영상 영역에서 효율적인 모드로 선택된다.

2.1.2 인트라 예측을 위한 매크로블록 모드

인트라 예측을 위한 매크로블록 모드의 블록 모양과 상대적인 블록의 크기는 그림 2와 같다.

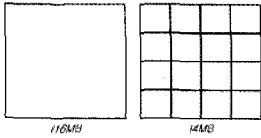


그림 2. 인트라 예측을 위한 매크로블록 모드 (I16MB, I4MB)

그림 2에서 I16MB로 예측될 경우 그림 3에서처럼 16×16 크기의 매크로블록을 4가지 공간 내 예측 방향으로 예측을 수행하며, 그림 2에서 I4MB로 예측될 경우 그림 4에서처럼 4×4 크기의 단위로 9가지의 공간 내 예측 방향으로 예측을 수행함으로써 예측 오차를 산출한다. 그림 3과 그림 4에 각각 I16MB 및 I4MB의 공간 내 예측 방향을 도시한다.

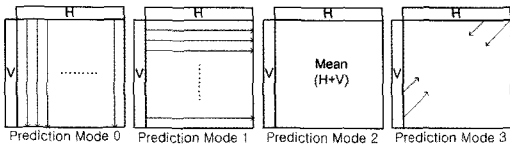


그림 3. I16MB에서의 4가지 공간 내 예측 방향

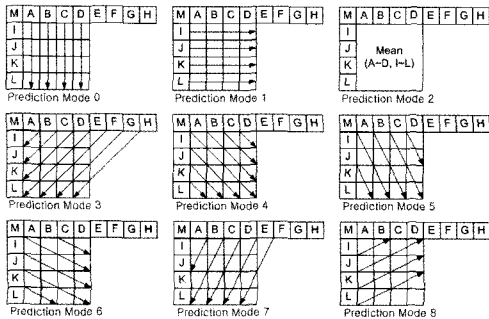


그림 4. I4MB에서의 9가지 공간적 예측 방향

2.2 참조 소프트웨어의 매크로블록 모드 선택 방법

참조 소프트웨어는 매크로블록의 모드를 선택하기 위해 식 (3)에 있는 윌-왜곡 비용함수를 사용한다.

$$RDCost = Distortion + \lambda \times Rate \quad (3)$$

윌-왜곡 최적화 기법이 사용되는 경우 윌-왜곡 비용 함수는 *Mode8* 내의 부-매크로블록 모드의 선택 및 전체 매크로블록 모드의 최종 선택 시 사용되며 높은 복잡도를 갖는다. 윌-왜곡 최적화 기법이 사용되는 경우, 참조 소프트웨어는 다음의 순서로 매크로블록 모드를 결정한다

Step 1. *Mode1*, *Mode2*, *Mode3*에 대한 참조 프레임 및 움직임 벡터 결정

$$\underset{ref, mv}{Arg\ Min} (MVcost(Mode\ i) + REFCost(Mode\ i)) \quad (4)$$

where, $i=1,2,3$

Step 2. *Mode8*내의 부-매크로블록 모드 선택

Step 2.1 첫 번째 부-매크로블록에 대한 참조 프레임 및 움직임 벡터 결정

$$\underset{ref, mv}{Arg\ Min} (MVcost(Mode\ i) + REFCost(Mode\ i)) \quad (5)$$

where, $i=4,5,6,7$

Step 2.2 첫 번째 부-매크로블록의 모드 선택

$$SubMB\ Mode = \min(RDCost(Mode4, Mode5, Mode6, Mode7)) \quad (6)$$

Step 2.3 잔여 부-매크로블록에 대해서 Step 2.1~Step 2.2 반복

Step 3. 최종 매크로블록 모드 결정

$$MB\ Mode = \min(RDCost(SKIP, Mode1 \sim Mode3, Mode8, I16MB, I4MB)) \quad (7)$$

식 (6)과 식 (7)에서 $RDCost(A, B, C)$ 의 표기는 다음과 같은 의미를 지닌다.

$$RDCost(A, B, C) = (RDCost(A), RDCost(B), RDCost(C)) \quad (8)$$

III. 참조 소프트웨어에서 인터 / 인트라 예측 방법

3.1 참조 소프트웨어의 인터 예측 방법

고속 전역 탐색 방법과 전역 탐색 방법은 각각 정수 픽셀에서 수행하는 움직임 추정 기법이며 본

논문에서 제안하는 알고리즘 역시 정수 픽셀 위치로만 제한한다[6]. 고속 전역 탐색 방법과 전역 탐색 방법을 각각 JM_FFS와 JM_FS라고 정의한다.

3.1.1 고속 전역 탐색 방법(Fast Full Search)

고속 전역 탐색 방법은 SAD(Sum of Absolute Difference) 복잡도를 줄이기 위해 고안된 알고리즘이다. 다양한 블록의 크기를 가지고 탐색 영역 내에서 움직임 추정을 할 경우 중복되는 SAD 계산이 많이 발생한다. 따라서 JM_FFS는 중복되는 계산을 없애기 위해서 처음에 탐색 영역 내에서 4x4단위로 SAD를 모두 계산하고 4x4단위로 계산된 SAD값들을 이용하여 다른 블록 크기(8x4, 4x8, 8x8, 16x8, 8x16, 16x16)에 대한 SAD를 도출해낸다.

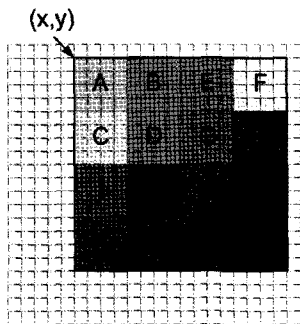


그림 5. 탐색 영역의 한 좌표에서 SAD 값 분포

그림 5에서 보이는 것처럼 4x4 단위로 미리 계산한 SAD 값들(A~P)을 가지고 다른 모드의 SAD를 계산한다. 탐색 영역 내의 (x, y) 좌표에 해당하는 4x4 단위의 SAD값 A와 (x+4, y) 좌표에 해당하는 4x4단위의 SAD값 B를 더해서 (x, y)좌표에서의 8x4 단위의 SAD값을 도출해 낸다. 또 (x, y)좌표에 해당하는 4x4 단위의 SAD값 A와 (x, y+4) 좌표에 해당하는 4x4단위의 SAD값 C를 더해서 (x, y)좌표에서의 4x8 단위의 SAD값을 도출해 낸다. 이렇게 처음 4x4 단위로 SAD를 계산하고 각각의 SAD를 합쳐 나가면서 모든 탐색 영역에서의 16x16

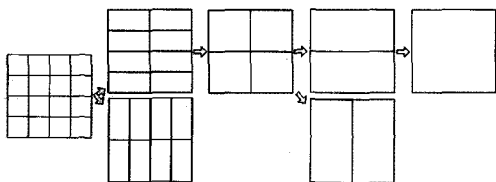


그림 6. 4x4 단위의 SAD로부터 다른 블록 크기에 대한 SAD 도출 과정

블록 크기까지의 SAD를 계산하므로 고속 전역 탐색이 가능한 것이다. 그림 6은 탐색 영역의 한 위치에서 4x4 단위의 SAD값들로부터 16x16 단위의 SAD 값까지 각각의 블록 크기에 대한 SAD 값들을 도출해 나가는 과정을 도식적으로 나타낸다.

3.1.2 전역 탐색 방법(Full Search)

참조 소프트웨어에서 제공되는 전역 탐색 방법은 불필요한 SAD 계산을 효율적으로 줄이기 위한 PDE(Partial Distortion Elimination) 알고리즘을 적용한 것이다[7]. 부분적으로 SAD(Partial_SAD)를 구해서 기존의 최소 SAD(minimum_SAD)와 비교하여 Partial_SAD가 minimum_SAD 보다 크면 SAD 연산을 중단함으로써 쓸모없는 연산을 생략하는 방법이다. JM_FS는 소위 나선 구조(spiral structure)로 움직임 추정을 실행할 때 성능이 극대화 된다. 그림 7은 탐색 순서에 대한 일반적인 움직임 추정 구조(Raster Structure)와 나선 구조를 나타낸다. 나선 구조는 대부분의 자연 영상의 움직임 벡터가 원점 근처에 위치하기 때문에 탐색할 영역의 중심부터 원을 그리듯이 움직임 벡터를 추정하므로 JM_FS와 동시에 적용할 때 빠른 속도로 움직임 추정을 할 수 있다. 즉 초기에 구한 SAD가 최소 SAD가 될 가능성이 높기 때문에 전체 탐색 영역에서 SAD 계산량을 많이 줄일 수 있다.

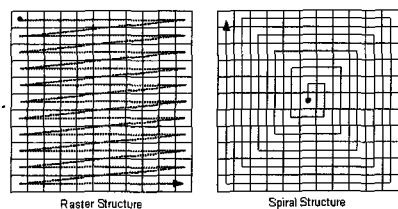


그림 7. 움직임 추정 구조

3.2 참조 소프트웨어의 인트라 예측 방법

I16MB는 평탄한 영역에서 인트라 예측을 할 때 적합하며 I4MB보다 낮은 복잡도를 갖는다. I4MB는 I16MB에 비해 더욱 세분화된 블록의 크기에 대하여 더욱 많은 방향성을 고려하므로 항상 I16MB 보다 작은 예측 오차를 가진다. 이는 I4MB의 세분화된 블록 크기와 방향성이 I16MB에서 고려하고 있는 예측 방향들을 포함하고 있기 때문이다. 하지만 I16MB가 하나의 매크로블록당 2비트(4개의 예측 모드)만을 부가 정보로 가지는 반면 I4MB의 경우에는 하나의 매크로블록당 16x3비트(9개의 예측 모

드)의 부가 정보를 갖는다. 따라서 14MB는 116MB와는 달리 복잡한 영역에서 적합하다.

3.2.1 14MB 선택 방법

매크로블록당 16개의 4x4 블록이 존재하며 각 블록당 9가지의 예측 모드가 존재 할 수 있다. 그림 8에서 하나의 매크로블록에서 14MB로 예측할 때 예측 순서를 볼 수 있다. 그림 8에 있는 순서대로 4x4 블록에 대해 앞에서 언급한 9가지 모드 각각의 울-왜곡 비용을 구해서 최소가 되는 모드가 해당 4x4 블록의 예측 모드로 결정된다. 14MB는 하나의 매크로블록에 대해서 최대 16x9번의 비용을 계산해야 하므로 116MB보다 상대적으로 복잡하다.

0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

14MB

그림 8. 14MB 모드 예측 순서

3.2.2 116MB 선택 방법

116MB는 4가지 방법으로 예측을 수행한다. 이러한 예측 모드에 대해서 SA(T)D가 최소가 되는 모드가 116MB의 최종 모드로 결정되며 그 모드에 대해서 울-왜곡 비용을 계산하므로 상대적으로 낮은 복잡도를 갖는다. 116MB의 모든 예측 모드가 가능할 때 그림 3에 언급한 예측 순서대로 진행된다.

IV. 제안하는 알고리즘

4.1 적응적 인트라 예측 알고리즘

JM_FFS는 모든 인트라 예측 모드를 검사 할 경우 JM_FS에 비해 상당히 효과적이다. JM_FS는 SKIP과 Model 같은 평탄한 영역이 많고 움직임이 아주 작은 영상의 경우 JM_FFS보다 효율적이다. 이러한 각각의 장단점을 이용하기 위해 이미 구해진 주변 매크로블록 모드들을 이용하여 현재 매크로블록 모드를 제한한 후, 제한된 모드에 따라서 JM_FFS와 JM_FS를 적응적으로 적용하면 인트라 예측에 있어서 상당한 속도 향상을 기대할 수 있다.

4.1.1 제안하는 매크로블록 모드 제한 방법

영상에 있어서 현재 매크로블록 모드는 주변 매크로블록 모드와 상관성이 아주 높다. 이런 특징을 이용하여 현재 매크로블록 모드를 예측할 수 있는데 그림 9에서 현재 매크로블록 모드 예측에 이용할 수 있는 주변 매크로블록 모드를 볼 수 있다.

크로블록 모드와 상관성이 아주 높다. 이런 특징을 이용하여 현재 매크로블록 모드를 예측할 수 있는데 그림 9에서 현재 매크로블록 모드 예측에 이용할 수 있는 주변 매크로블록 모드를 볼 수 있다.

pb	pc	pd	b	c	d
pa	px	pe	a	x	
ph	pg	pf			

Previous frame Macroblock

Current frame Macroblock

그림 9. 모드 예측을 위한 주변 매크로블록 위치

그림 9에서 x는 현재 부호화할 매크로블록의 위치를 나타내며 px는 이전 프레임의 같은 위치의 매크로블록의 위치를 나타낸다. 현재 매크로블록의 위치에 따라서 활용 가능한 주변 매크로블록의 위치가 바뀌게 되는데, 그림 10에서 매크로블록의 위치에 따라서 이용 가능한 주변 매크로블록을 9가지 경우로 분류하고 있다.

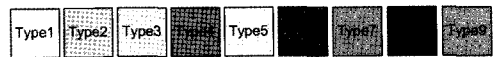
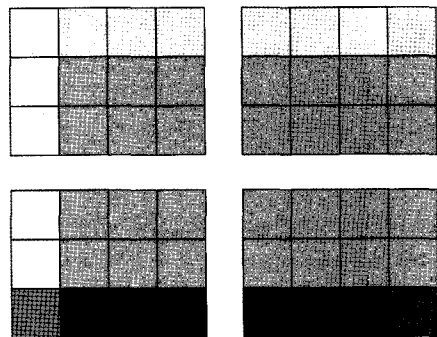


그림 10. 매크로블록의 위치에 따른 분류

식 (9)는 각각의 경우에 있어서 이용할 매크로블록의 위치를 나타낸다. 예를 들어 그림 10의 Type1의 경우 px, pe, pg에 해당하는 위치의 매크로블록의 모드를 이용함을 의미한다. 식 (9)는 대부분의 자연 영상에서 물체는 대각선 방향보다는 가로 세로 방향으로 많이 움직이며 공간적, 시간적으로 가까울수록 상관도가 높다는 가정에 근거를 두며 식 (9)를 이용하여 현재 위치의 매크로블록의 모드를 제한한다.

- Type1 = ModePred(px, pe, pg)
- Type2 = ModePred(a, px, pg)
- Type3 = ModePred(a, px, pe, pg)
- Type4 = ModePred(c, px, pe)
- Type5 = ModePred(c, px, pe, pg)
- Type6 = ModePred(a, c, px)
- Type7 = ModePred(a, c, px, pg)
- Type8 = ModePred(a, c, px, pe) (9)
- Type9 = ModePred(a, c, px, pe, pg)

식 (9)의 각 매크로블록들의 모드를 이용하여 현재 매크로블록의 모드를 제한하는 방법 (ModePred)은 다음과 같다.

- Case 1: 모든 매크로블록의 모드가 SKIP일 경우
⇒ 후보 모드는 SKIP과 Mode1로 제한
- Case 2: 모든 매크로블록의 모드가 Mode1일 경우
⇒ 후보 모드는 SKIP과 Mode1로 제한
- Case 3: 모든 매크로블록의 모드가 SKIP, Mode1, Mode2, Mode3중에 하나일 경우
⇒ 후보 모드는 SKIP, Mode1, Mode2, Mode3 으로 제한
- Case 4: 모든 매크로블록의 모드가 Mode8일 경우
⇒ 후보 모드에서 Mode1, Mode2, Mode3을 제거
- Case 5 : 그 밖의 경우
⇒ 모든 모드에 대해 예측 수행

Case 5의 경우에서 모든 모드에 대해 예측을 수행하므로 여러 누적을 방지한다. 표 1은 실험에 사용한 전체 영상에 대해서 평균적으로 제안한 모드 예측 방법의 정확도를 나타낸다. QP가 40일 경우 대부분의 영상에서 아주 정확함을 볼 수 있으며 다른 QP에 대해서도 전체적으로 상당히 높은 범위의 정확도를 갖는다는 것을 알 수 있다.

표 1. 제안한 모드 예측 방법의 범위 정확도(%)

	average	
	Hit	Miss
QP24	72.89	27.11
QP28	77.55	22.45
QP32	82.12	17.88
QP36	87.40	12.60
QP40	92.00	8.00
average	85.99	17.61

4.1.2 적응적인 움직임 추정 기법

제안된 모드 제한 기법에 따라서 JM_FFS와 JM_FS를 적응적으로 적용할 경우 복잡도를 상당히 감소시킬 수 있다. JM_FFS는 SKIP과 Mode1만을 검사하는 경우 JM_FS보다 성능이 더 떨어진다. 따라서 Case 1과 Case 2의 경우에는 JM_FS를 적용하고, Case 3~5의 경우에는 JM_FFS를 적용하는 것이 효율적이다.

4.2 고속 인트라 예측 알고리즘

대부분의 자연 영상은 공간 영역의 연관성(spatial correlation) 보다는 시간 영역의 연관성(temporal correlation)이 훨씬 크다. 실험에 사용한 전체 영상의 평균 비율로 IPPP구조를 사용했을 때 I프레임을 제외한 모든 P프레임에서 공간 영역의 연관성과 시간 영역의 연관성을 표 2에 나타내었다.

표 2. 인트라 모드와 인트라 모드의 비율(%)

	average	
	Inter	Intra
QP24	95.44	4.56
QP28	95.99	4.01
QP32	96.24	3.76
QP36	96.50	3.50
QP40	96.81	3.20
average	96.20	3.80

평균 96.2%에 해당하는 매크로블록이 인트라 모드로 선택되고 나머지 3.8%에 해당하는 매크로블록만 인트라 모드로 선택된다는 것은 대부분의 매크로블록에 있어서 인트라 예측 모드의 율-왜곡 비용은 인트라 예측 모드의 비용보다 작다는 것을 의미한다. 이런 특징을 이용하여 인트라 예측을 위해 제안하는 알고리즘은 이전에 고안된 PDE 기법을 적용하는 것이다. 즉 I16MB와 I4MB의 율-왜곡 비용을 구할 때 부분적으로 비용을 계산하고, 현재 다른 모드의 최소 율-왜곡 비용과 비교하여 남은 부분의 율-왜곡 비용을 계산할지 여부를 판단하는 것으로 특히 I4MB의 복잡도 감소를 극대화 할 수 있다.

4.2.1 모드별 율-왜곡 비용 정의

SKIP, Mode1, Mode2, Mode3, Mode8 각각의 율-왜곡 비용을 RDcost_SKIP, RDcost_16x16, RDcost_16x8, RDcost_8x16, RDcost_8x8 이라고 정의한다. 또한 I16MB, I4MB의 율-왜곡 비용을 각각 RDcost_I16MB, RDcost_I4MB로 정의한다. 마지막으로 참조 소프트웨어에서 매크로블록 모드를 결정

할 때 Step 3에서 정해지는 최소 율-왜곡 비용을 $RDcost_min$ 이라고 정의하면 식 (10)이 성립한다.

$$RDcost_Min = \text{Minimum}(RDcost_SKIP, RDcost_16 \times 16, RDcost_16 \times 8, RDcost_8 \times 16, RDcost_8 \times 8, RDcost_I16MB, RDcost_I4MB)$$

4.2.2 I4MB의 복잡도 감소 알고리즘

I4MB는 각각의 4x4 크기의 블록별로 9개의 율-왜곡 비용을 구해서 해당 블록의 최종 모드로 선택되므로 최대 16x9번의 율-왜곡 비용을 구해야 한다. 그러나 앞에서 대부분의 매크로블록에서 공간 영역의 상관성 보다는 시간 영역의 상관성이 더 크다는 특징을 이용하면 대체적으로 식 (11)을 만족한다.

$$RDcost_I4MB > RDcost(SKIP, mode1, mode2, mode3, mode8)$$

따라서 $RDcost_I4MB$ 를 구할 때 모든 4x4 크기의 블록에 대해 각각 9가지 예측 모드에 대한 율-왜곡 비용을 전부 구하기보다는 4x4 크기의 블록에 대해서 각각의 9가지 모드에 대한 최적의 4x4 블록의 율-왜곡 비용을 구한 다음, 현재 4x4 블록까지의 율-왜곡 비용 합과 기존의 다른 모드에서 구해진 $RDcost_min$ 과 비교하여 계산할 필요가 없는 나머지의 4x4 크기의 블록에 대한 율-왜곡 비용 연산을 줄이는 것이 효과적이다. 제안하는 알고리즘은 그림 8의 순서대로 율-왜곡 비용을 계산하여 부분 율-왜곡 비용의 합과 기존의 $RDcost_min$ 값을 비교한다.

4.2.3 I16MB의 복잡도 감소 알고리즘

I16MB는 매크로블록 크기와 같은 16x16 단위로 예측을 수행하므로 I4MB에서처럼 남은 4x4 블록에 대한 율-왜곡 비용 연산량 감소를 기대할 수 없다. 대신 4가지 예측 모드별로 SA(T)D를 구할 때 SA(T)D 부분합과 I16MB의 다른 모드에서 구해진 최소 SA(T)D와 비교하여 SA(T)D 복잡도를 감소시

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

그림 11. I16MB의 부분 SA(T)D 비교 단위 및 처리 순서

킬 수 있다. 그림 11은 I16MB에서 SA(T)D 계산을 수행할 때 부분 SA(T)D의 비교 단위와 처리 순서를 나타낸다. 즉 그림 11의 순서대로 부분 SA(T)D와 중간에 비교하므로 필요 없는 SA(T)D 연산을 중단할 수 있어 복잡도 감소 효과를 얻을 수 있다.

V. 실험 및 결과

제안된 알고리즘의 성능 평가를 위해서 인텔 펜티엄IV 2.4GHz 컴퓨터를 이용하였으며 JM 8.4 코덱을 사용하였다. IPPP 구조를 사용하였고 양자화 파라미터는 여러 상황을 고려하여 24, 28, 32, 36, 40을 모두 실험하였다. 율-왜곡 최적화를 사용하였고 움직임 탐색 영역은 16으로, 참조 프레임은 1장으로 실험을 수행하였다. 실험 영상은 모두 CIF 크기이며 영상에 대한 정보는 표 3과 같다.

표 3. 실험에서 사용한 테스트 영상에 관한 정보

이름	프레임수	이름	프레임수
akiyo	300	Foreman	300
Bridge	2000	Mobile	300
Bus	150	mother and daughter	300
Coastguard	300	Stefan	300
Container	300	Table	300
Flower	246	Tempete	260
Football(1)	90	Vectra	142
Football(2)	150	Waterfall	260

표 4. JM_FFS와 제안된 알고리즘의 성능 비교 표(QP=28)

QP 28	difference(Proposed-JM_FFS)				
	Sequences	PSNR [dB]	tot_t [%]	ME_t [%]	INTRA_t [%]
akiyo	-0.096	-50.178	-51.641	-46.088	0.313
bridge	-0.101	-40.060	-51.950	-29.891	0.965
bus	-0.095	-30.326	-17.878	-33.453	2.115
coastguard	-0.090	-28.225	-18.901	-26.640	0.652
container	-0.038	-37.841	-32.709	-35.915	1.441
flower	-0.081	-32.714	-17.830	-37.332	1.199
football(1)	-0.034	-15.411	-13.064	-12.787	0.467
football(2)	-0.038	-18.887	-10.383	-23.894	0.994
foreman	-0.064	-30.092	-22.338	-30.706	0.827
mobile	-0.126	-27.884	-14.308	-31.368	1.410
M&D	-0.060	-44.279	-43.886	-40.857	0.635
stefan	-0.070	-26.484	-15.590	-33.456	0.871
table	-0.064	-35.781	-30.512	-34.346	1.917
tempete	-0.098	-27.701	-14.522	-30.265	0.313
vectra	-0.071	-22.421	-15.126	-24.223	0.747
waterfall	-0.040	-32.174	-21.922	-32.861	-0.174
average	-0.073	-31.279	-24.535	-31.505	0.918

표 4와 표 5에 제안된 알고리즘과 JM_FFS, JM_FS 각각의 성능 차이를 표로 나타내었으며 각각의 QP에 따른 전체 영상에 대한 평균적인 결과를 표 6과 표 7에 각각 나타내었다.

표 5. JM_FS와 제안된 알고리즘의 성능 비교 표(QP=28)

QP 28	difference(Proposed-JM_FS)				
	Sequences	PSNR [dB]	tot_t [%]	ME_t [%]	INTRA_t [%]
akiyo	-0.097	-49.731	-50.535	-46.444	0.343
bridge	-0.103	-45.196	-67.018	-30.037	0.980
bus	-0.093	-43.212	-62.497	-34.035	2.271
coastguard	-0.081	-42.137	-63.364	-27.558	0.574
container	-0.040	-45.837	-60.029	-35.550	1.203
flower	-0.078	-42.023	-60.736	-37.037	1.185
football(1)	-0.031	-35.800	-63.868	-15.021	1.518
football(2)	-0.037	-35.781	-63.793	-24.048	1.061
foreman	-0.066	-40.073	-56.530	-29.619	0.898
mobile	-0.127	-39.366	-60.405	-31.435	1.462
M&D	-0.058	-46.378	-51.743	-41.503	0.568
stefan	-0.082	-37.479	-55.810	-34.695	11.668
table	-0.062	-46.793	-63.894	-36.169	2.097
tempete	-0.102	-39.305	-59.204	-30.255	0.318
vectra	-0.072	-35.666	-58.706	-25.304	1.081
waterfall	-0.041	-42.642	-60.291	-33.566	0.051
average	-0.073	-41.714	-59.901	-32.017	1.705

표 6. JM_FFS와 제안된 알고리즘의 성능 비교 표(평균값)

Sequences	PSNR [dB]	tot_t [%]	ME_t [%]	INTRA_t [%]	bitrate [%]
QP 40	-0.047	-40.414	-35.155	-40.128	0.769
QP 36	-0.063	-36.887	-31.021	-36.018	0.847
QP 32	-0.071	-33.981	-27.335	-34.036	0.863
QP 28	-0.073	-31.279	-24.535	-31.505	0.918
QP 24	-0.079	-27.849	-19.877	-28.530	0.710

표 7. JM_FS와 제안된 알고리즘의 성능 비교 표(평균값)

Sequences	PSNR [dB]	tot_t [%]	ME_t [%]	INTRA_t [%]	bitrate [%]
QP 40	-0.051	-46.891	-55.079	-40.210	1.175
QP 36	-0.061	-45.281	-57.359	-36.366	1.613
QP 32	-0.069	-43.649	-58.864	-34.099	1.732
QP 28	-0.073	-41.714	-59.901	-32.017	1.705
QP 24	-0.079	-39.102	-60.174	-29.176	1.310

QP가 40일 경우 JM_FFS에 비해서 전체 부호화 시간이 평균 40% 이상 감소하며 인트라/인트라 예측 시간이 각각 35%와 40% 이상씩 감소한다. 그에 반해 비트율은 0.769% 증가하고, PSNR은 0.047dB 감소로 아주 미미하다고 할 수 있다. 표를 자세히 살펴보면 akiyo 영상의 전체 부호화 시간 감소는 50% 이상으로 매우 큰데, 대부분의 매크로블록이 SKIP

또는 Mode1이므로 인트라 예측이 좀 더 효율적으로 수행되었음을 알 수 있다. 복잡한 움직임을 보이는 football(2)의 영상에 경우에는 SKIP과 Mode1이 상대적으로 적기 때문에 전체 부호화 시간이 akiyo 영상만큼 감소하지는 않았지만 PSNR이 0.069dB 정도의 미미한 손실에 비해서 18% 이상의 부호화 시간이 줄었음을 확인할 수 있다. 즉 제안된 알고리즘이 움직임이 적은 영상뿐만 아니고 움직임이 크고 복잡한 영상에도 잘 적용됨을 확인할 수 있다.

VI. 결론

H.264/AVC 비디오 부호화 방식은 여러 부호화 도구 중에서도 특히 다양한 블록 크기를 갖는 인트라 모드 선택 과정과 새로이 추가된 인트라 예측 과정으로 인해 매우 높은 복잡도를 갖는다. 따라서 본 논문은 매크로블록 모드 선택에 있어서 인트라 예측과 인트라 예측의 복잡도 감소에 주안점을 두었으며 이를 위해 크게 두 가지의 알고리즘을 제안하였다. 인트라 예측의 복잡도 감소를 위해서 현재 매크로블록의 모드 예측 방법을 제시하였고 예측된 모드에 대해서 적응적 움직임 추정 기법을 적용하였다. 인트라 예측의 복잡도 감소를 위해서는 I4MB와 I16MB의 복잡도 감소 알고리즘을 각각 제안하였으며 특히 I4MB의 선택에 있어서 큰 복잡도 감소를 이루었다. 두 가지 제안한 알고리즘을 함께 사용할 경우 최대 60.9% 정도의 전체 부호화 시간을 단축시키는 반면 PSNR이나 비트율의 변화는 매우 작은 실험 결과를 보였다. 제안된 인트라/인트라 알고리즘은 H.264의 다른 부분의 복잡도 감소 기법과 함께 사용할 경우 실시간 부호화가 가능할 것이다.

참고 문헌

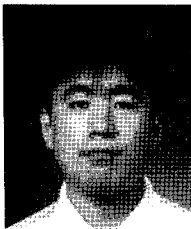
- [1] JVT G050r1, "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. h.264/ISO/IEC 14496-10 AVC)," May 2003.
- [2] Thomas Wiegand, Gary J. Sullivan, "Overview of the H.264/AVC Video Coding Standard," IEEE Trans Circuits and Systems for Video Technology, vol. 13, pp. 560-576, July 2003.
- [3] Thomas Wiegand, Heiko Schwarz, Anthony Joch, Faouzi Kossentini, "Rate-Constrained

Coder Control and Comparison of Video Coding Standard," IEEE Trans. Circuits and Systems for Video Technology, vol. 13, pp. 688-703, July 2003.

- [4] Peng Yin, Hye-Yeon Cheong Tourapis, Alexis Michael Tourapis, Jill Boyce, "Fast Mode Decision and Motion Estimation for JVT/H.264," Image Processing, 2003 International Conference on, vol. 3 pp. 853-856, Sept. 2003.
- [5] Donghyng 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, January 2005.
- [6] Chen-Fu Lin, Jin-Jang Leou, "An Adaptive Fast Full Search Motion Estimation Algorithm for H.264," IEEE International Symposium on Circuits and Systems, pp. 1493 - 1496, May 2005.
- [7] J. N. Kim and T. S. Choi, "Adaptive matching scan algorithm based on gradient magnitude for fast full search in motion estimation," IEEE Trans. On Consumer Electronics, vol. 45, no. 3, pp 762-772, Aug. 1999.

진 순 중 (Soonjong Jin)

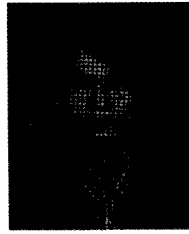
준회원



2004년 2월 한양대학교 전자컴퓨터공학부 졸업
 2004년 3월~현재 한양대학교 전자통신컴퓨터공학과 석사과정
 <관심분야> 영상처리 및 영상압축

김 원 기 (Wonki Kim)

정회원



2000년 8월 한양대학교 전기전자컴퓨터공학부 졸업
 2003년 2월 한양대학교 전자통신전파공학과 석사
 2003년 3월~현재 한양대학교 전자통신전파공학과 박사과정
 <관심분야> 영상통신, 영상처리,

디지털 방송

정 제 창 (Jechang Jeong)

정회원



1980년 2월 서울대학교 전자공학과 졸업
 1982년 2월 KAIST 전기전자공학과 석사
 1990년 미국 미시간대학 전기공학과 공학박사
 1980년~1986년 KBS 기술연구

소 연구원 (디지털 TV 및 뉴미디어 연구)
 1990년~1991년 미국 미시간대학 전기공학과 연구교수(영상 및 신호처리 연구)
 1991년~1995년 삼성전자 멀티미디어 연구소(MPEG, HDTV, 멀티미디어 연구)
 1995년~현재 한양대학교 전자전기컴퓨터공학부 교수(영상통신 및 신호처리 연구실)
 1998년 11월 27일 과학기술자상 수상
 1998년 12월 31일 정보통신부장관상 표창
 <관심분야> 영상처리 및 영상압축