

논문-05-10-4-06 .

초저복잡도 H.264 부호기의 움직임 추정 및 모드 결정 알고리즘

유영일^{a)}, 김용태^{a)}, 이승준^{b)}, 강동욱^{a)*}, 김기두^{a)}

Motion Estimation and Mode Decision Algorithm for Very Low-Complexity H.264/AVC Video Encoder

Youngil Yoo^{a)}, Yong Tae Kim^{a)}, Seung-Jun Lee^{b)}, Dong Wook Kang^{a)*}, and Ki-Doo Kim^{a)}

요 약

H.264는 우수한 부호화 성능으로 말미암아 DMB와 차세대 DVD 등 새로운 멀티미디어 서비스를 위한 비디오 코덱으로 채택되어 왔다. 그러나 이 표준의 표준 코덱인 Joint Model(JM)은 연산 자원이 한정된 임베디드 환경에서 사용되기에는 너무 복잡한 다수의 알고리즘을 포함하고 있다. 본 논문은 임베디드 환경에서 사용할 수 있도록 설계된 초저복잡도 H.264 부호화 알고리즘을 제시한다. 제안하는 알고리즘은 압축 성능의 급격한 악화를 가져오지 않는 범위 내에서 몇 가지 부호화 도구의 사용을 제한하고, 또 움직임 추정과 모드 결정 과정에 몇 가지의 조기 종료 조건과 우회 조건을 추가하는 방법으로 구현하였다. 제안하는 알고리즘을 사용하여 7.5fps QCIF 영상을 64kbps로 부호화하는 경우, 표준 JM에 비해서 평균적으로 0.4 dB 정도의 미약한 PSNR 열화를 나타내에도 불구하고, 단지 15%의 계산량만을 필요로 할 뿐만 아니라, 필요한 메모리의 양과 전력 사용률을 획기적으로 낮추는 것이 가능하다. 제안하는 알고리즘의 H.264 부호기를 Intel PXA255 Processor를 사용하는 PDA에 이식하는 실험을 통하여 PDA에서 H.264 기반의 MMS(Multimedia Messaging Service)의 실현 가능성을 검증하였다.

Abstract

The H.264 has been adopted as the video codec for various multimedia services such as DMB and next-generation DVD because of its superior coding performance. However, the reference codec of the standard, the joint model (JM) contains quite a few algorithms which are too complex to be used for the resource-constraint embedded environment. This paper introduces very low-complexity H.264 encoding algorithm which is applicable for the embedded environment. The proposed algorithm was realized by restricting some coding tools on the basis that it should not cause too severe degradation of RD-performance and adding a few early termination and bypass conditions during the motion estimation and mode decision process. In case of encoding of 7.5fps QCIF sequence with 64kbps with the proposed algorithm, the encoder yields worse PSNRs by 0.4 dB than the standard JM, but requires only 15% of computational complexity and lowers the required memory and power consumption drastically. By porting the proposed H.264 codec into the PDA with Intel PXA255 Processor, we verified the feasibility of the H.264 based MMS(Multimedia Messaging Service) on PDA.

Keyword: Keyword: H.264, 예측 부호기, 움직임 추정, 모드 결정, 임베디드 비디오 코덱

a) 국민대학교
Kookmin University
b) ㈜엠큐브웍스
MCubeWorks Inc.

* 본 연구는 2005년도 국민대학교 교내연구비를 지원받아 수행된 연구입니다.

I. 서 론

최신 비디오 압축 표준인 H.264 또는 MPEG-4 AVC (Advanced Video Coding)는 H.261, MPEG-1, MPEG-2, H.263, MPEG-4 와 같은 기존의 동영상 압축 표준과 마찬가지로 하이브리드 블록 기반 코덱이다^[1]. 기존 표준과 비교할 때, 예측 부호화를 위해서 사용하는 영상의 예측 방식이 보다 다양하고, 이 다양한 예측 방식을 매우 효과적으로 선택해내는 최적화된 모드 결정법이 존재하여 기존 표준에 비해 대폭 향상된 압축 성능을 나타낸다. 그러나 부호화 효율의 향상을 위해서 도입된 예측 모드 선택의 다양성과 향상된 적정성을 제공하기 위해서 H.264 부호기의 구조가 기존 표준 부호기에 비해서 급격히 복잡해졌으며, 이와 아울러 훨씬 많은 횡수의 연산을 필요로 하게 되었다. 따라서 계산 능력이 제한된 휴대형 단말기의 경우에는 예측 모드 결정과 움직임 추정 과정에서 병목현상이 발생하기 때문에 H.264를 최대의 부호화 성능을 나타내는 모드로 사용하는 것은 불가능하다^[2]. 따라서 제한된 자원을 갖는 임베디드 H.264 부호기를 구현하고자 할 때는 부호기의 기능과 그에 따른 복잡도를 최소화하여야만 한다. 이는 H.264 참조 소프트웨어인 Joint Model (JM)에 정의된 부호화 도구들 중 몇몇 도구를 비용 효율 관점에서 사용을 제한하고 모드 결정 과정에 조기 종료 조건 및 우회 조건을 두어 고속화하며, 모드 탐색 패턴 또한 간략한 방식으로 수정함으로써 비로소 가능하다.

지금까지 H.264 표준에 다양한 고속 움직임 추정 및 모드 결정 알고리즘을 적용함으로써, JM에 포함되어 있는 고속 탐색 옵션과 비교할 때, 압축 효율의 별다른 열화 없이 부호기의 복잡도를 20%~40%정도 줄일 수 있는 연구 결과들이 발표된 바 있다^{[3]-[9]}. 그러나 이 정도의 복잡도 저감은 실제 임베디드 시스템에 적용하기에는 여전히 너무 높은 수준이어서 보다 근본적이고 획기적인 복잡도 저감 기법이 여전히 요구되고 있다. 본 논문에서는 PDA에서 7.5fps의 QCIF 프레임 포맷 영상을 64kbps 이하로 부호화 하는 것을 목표로 H.264 baseline profile (BP) 부호기를 단순화시키는 방법론을 제시하고자 한다.

논문의 구성은 다음과 같다. 2장에서는 H.264 BP의

부호화 도구들을 임베디드 환경에 맞도록 재설정함으로써 본 논문에서 구현하고자 하는 초저복잡도 H.264 부호기의 구현 방식에 대하여 소개하고, 3장에서는 그러한 설정 하에서 매크로블록 부호화 모드의 탐색과 움직임 추정에 필요한 계산량을 더욱 낮추기 위한 기법들을 제안한다. 4장에서는 실험 결과를 보이고, 5장에서 결론을 맺는다.

2. 초저복잡도 H.264 부호기

H.264 BP 부호기는 7 가지의 인터 모드(inter mode), 13 가지의 인트라 모드(intra mode), 그리고 하나의 스킵 모드(skip mode) 등 서로 다른 예측 방식에 따른 총 21개의 모드 중에서 하나의 모드를 선택하여 각각의 매크로블록을 부호화한다. 또한 인터 모드에서의 예측의 정확도를 기존 동영상 압축 표준보다 더욱 높이기 위하여, 즉 예측 후 잔류 신호의 에너지를 더욱 줄일 수 있도록 하기 위해서 여러 장의 이전 프레임을 참조하여 예측할 수 있는 "다중 참조 프레임"과 움직임 벡터의 정밀도를 1/4 화소 단위까지 낮춘 "1/4화소 움직임 추정" 등이 새롭게 사용되고 있다^[1].

그러나 PDA에서와 같이 ARM기반의 RISC 프로세서에서 H.264를 동작시키기 위해서는 이처럼 복잡한 부호화 도구들을 모두 사용한다는 것은 불가능하다. 즉, 임베디드 환경에서 H.264 부호화를 행하는 경우에는 부호화 효율의 저하를 일부 감수하고서도 H.264의 몇 가지 부호화 도구의 사용을 제한하여야만 한다. 이와 같은 제한된 조건에서 설계된 부호화기를 초저복잡도 H.264 부호기라고 부른다. H.264 부호기에서 제공하는 부호화 도구의 사용 제한은 필연적으로 부호화 효율을 저하를 야기한다. 그런데 사용을 제한하는 부호화 도구가 어떤 것인가에 따라서 그 부호화 효율의 저하 정도가 많은 차이를 보인다. 따라서 적절하게 선택된 방식으로 부호화 도구의 사용을 제한함으로써 부호화 효율의 저하를 최대한 억제할 수 있게 되는 것이다. 결과적으로 본 논문에서 제시하는 초저복잡도 H.264 부호기에 적용된 부호화 도구 사용 방식은 표 1과 같다.

표 1. 설계된 PDA용 임베디드 부호기에 적용된 부호화 도구 사용 방식
Table 1. Encoding Tools used in the designed embedded encoder applicable to the PDA

Encoding Tool	Usage
RD-optimization	Off
Hadamard transform	Off
# of ref. frames	1
MV Search range	8
MV resolution	1/4
Possible coding MB modes	All (7 block shapes)

표 1과 같은 부호화 도구 사용 방식을 결정하게 된 근거는 아래와 같은 분석에 따른 것이다. 아래의 분석은 QCIF 형식의 동영상을 실제로 부호화한 결과를 분석한 내

용이며, 그림 1은 그 중에서 Foreman 영상에 대한 실험 결과를 보여주고 있다. 나머지 영상에서의 실험 결과도 그림 1과 유사한 경향을 보인다.

첫째, 부호율-왜곡 최적화(RD: rate-distortion optimization) 기반에서 모드를 결정하는 고복잡도 (high-complexity) 모드 결정은 예측잔류신호들의 절대값의 합, 즉 SAD(sum absolute differences)를 비교하여 모드를 결정하는 저복잡도 (low-complexity) 모드 결정에 비하여 약 5% 정도 높은 압축 효율을 보인다 (그림 1 참조). 그러나 고복잡도 모드 결정 기법에서는 각각의 모드마다 DCT 변환과 역변환 그리고 엔트로피 부호화를 수행해야 하기 때문에 임베디드 환경에서 과중한 연산 부담이 되어서 사용할 수

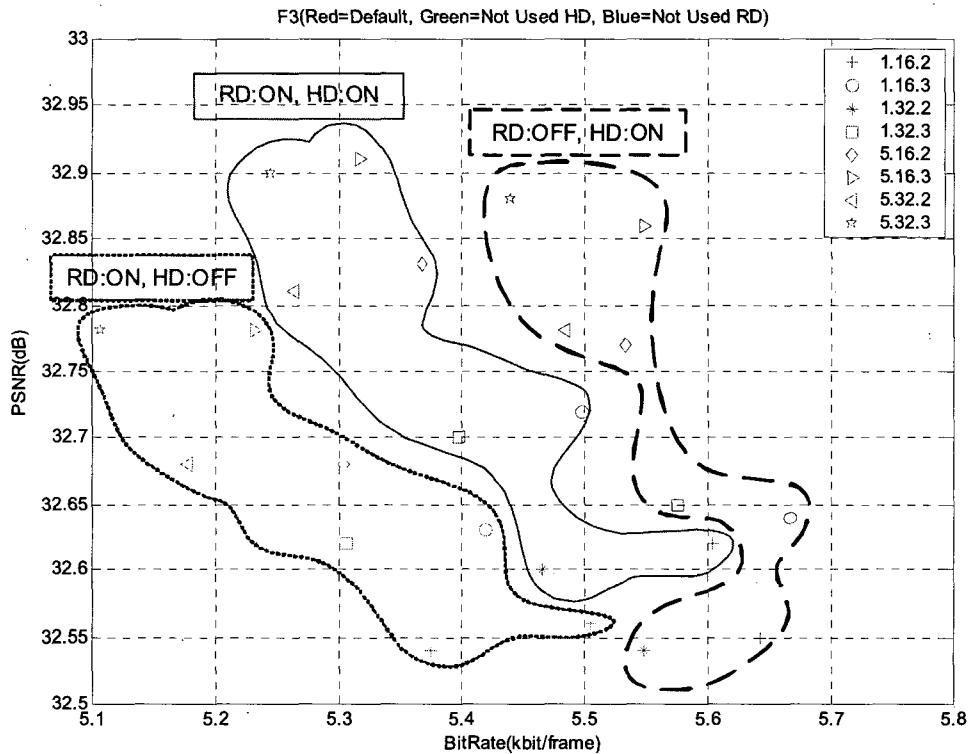


그림 1. RD 최적화, HD 변환에 따른 RD 성능(Foreman 영상). 실험 조건을 나타내는 기호에 대한 x.y.z의 의미는 각각 참조 프레임의 수, 움직임 벡터의 탐색 범위, 그리고 모드 탐색의 타입(2는 8x4, 4x8, 4x4 블록 세이프를 제외하고 모드 탐색을 하는 경우를, 3은 이를 포함한 모든 가능한 모드를 탐색하는 경우를 의미)를 나타낸다.

Fig. 1. RD performance accorsing to the RD optimization and Hadmard transform. Parameters x.y.z representing the simulation conditions mean the number of reference frames, search range for motion vectors, and type of mode search (type 2 searches all modes except 8x4, 4x8, and 4x4, while type 3 searches all modes including them)

없다. 그리고 DC 계수에 대한 추가적인 Hardmard 변환에 의해서는 그 보다 낮은 1-2%의 압축 효율 개선을 기대할 수 있으나, 계산 부담을 줄이기 위해서 제안하는 부호기는 Hardmard 변환을 사용하지 않도록 설계되었다.

둘째, 다중 참조 프레임을 사용하여 움직임을 추정/보상하는 경우 부호화에 따른 압축 효율이 한 장의 프레임을 참조하여 움직임 보상 부호화하는 경우에 비해서 대략 최대 3% 정도 향상된다 (그림 1 참조). 그러나 다중 참조 프레임에 의한 움직임 추정은 참조하는 프레임 수에 비례하여 계산량과 소모 메모리의 크기가 증가하기 때문에 초저 복잡도 부호기에서 다중 프레임 참조를 행하는 것은 불합리한 것으로 판단하여 제안하는 부호기는 움직임 보상을 위한 참조 프레임의 수를 1로 고정하였다.

셋째, 움직임 벡터의 탐색 범위를 변화시키면 최종 복원 화질의 차이보다는 압축 효율에서 더 분명한 차이를 보인다 (그림 1 참조). 그리고 그 차이는 부호화하고자 하는 대상 영상의 특성에 따라 크게 편차를 나타낸다. 반면에 부호기의 복잡도는 대략적으로 탐색 범위와 정비례하여 증가한다. QCIF 규격의 영상에 적용하는 경우 +/-8 화소의 움직임 벡터 탐색 범위가 압축 성능과 부호기 복잡도 사이의 적절한 타협으로 판단하여, 제안하는 부호기는 움직임 벡터의 탐색 범위를 +/-8 화소로 결정하였다.

넷째, 탐색하는 모드의 수와 인터 모드에서 추정하고자 하는 움직임 벡터의 정밀도는 특별한 제한을 가하지 않고 표준을 따르도록 하였다. H.264는 모드에 따라서 예측하는 블록의 크기를 가변적으로 변경시키면서 예측함으로써 압축 성능을 크게 개선하고 있다. 그림 1은 이를 잘 보여주고 있다. 예를 들어 1.16.2(참조 프레임의 수=1; 움직임 벡터 탐색 범위= +/-16; 8x8 이상 크기를 갖는 블록 세이프를 갖는 모드만 탐색)와 1.16.3(참조 프레임의 수=1; 움직임 벡터 탐색 범위= +/-16; 4x4, 4x8, 8x4를 포함하여 모든 모드 탐색) 조건에서의 RD 특성을 비교하면, 모드 탐색의 범위를 확장함으로써 약 1dB의 화질 차이를 보인다. 이와 같은 가변 블록 예측의 장점을 살리기 위해서는 모드의 수를 합부로 제한할 수 없으며, 특히 모드 선택 과정에서 모드간 상관성을 이용함으로써 부호기의 복잡도를 원하는 수준까지 낮추는 것이 가능하기 때문에 제안하는 부호기는 탐색

모드의 수를 제한하지 않는다.

다섯째, 추정하는 움직임 벡터 정밀도와 관련하여서는, 부호소 단위의 움직임 추정을 통하여 화소 단위의 움직임을 추정하는 것보다 결과적인 부호화 성능을 약 3dB 정도 향상시킬 수 있다. 그러나 부호소 단위의 움직임을 추정하기 위해서는 부호소 위치에서의 화소 값을 보간해야 하는데, 이 보간 과정에서 부호기의 계산적 복잡도와 요구 메모리의 크기가 대폭 증대된다. 본 논문에서는 이와 같은 딜레마를 해결하기 위해서 예측 오차 비용함수 평면에 대한 2차원 다항식 모델링을 통하여, 부호소 위치에서의 보간을 단 1회로 한정하는 기법을 사용하였다. 즉 탐색 범위의 부호소 위치마다 실제로 보간을 행하여 SAD를 계산하지 않고 그 부호소 위치를 둘러싸고 있는 9개의 화소 위치에서의 예측 오차 비용함수를 사용하여 2차원 다항식으로 근사화된 비용함수 평면을 구하고, 그 평면에서 비용함수 값의 극소점이 발생하는 위치를 계산함으로써 부호소 움직임 벡터를 추정한다.

3. 초저복잡도 부호기를 위한 모드 결정과 움직임 추정 기법의 제안

앞 절에서 논의된 바와 같이 부호화 모드의 수를 제한하거나 움직임 벡터의 정밀도를 낮추는 것은 부호화 효율을 과도히 떨어뜨릴 가능성이 있기 때문에, 본 논문에서 제안하는 임베디드 부호기에서는 표준에서 정한 모드를 모두 지원하고 움직임 벡터의 정밀도도 1/4화소로 하여 부호화하도록 설계되었다. 그러나 모드 탐색과 움직임 벡터 추정을 JM에서 제시하는 표준 방식대로 행하는 것은 임베디드 환경에서는 실현 불가능하다. 따라서 다음과 같은 세 가지의 모드 탐색과 움직임 벡터 추정을 위한 새로운 저복잡도 방식을 제안하고자 한다.

3.1 모드 탐색 과정에서의 우회 경로 사용

그림 2는 하나의 매크로블록의 부호화 모드를 결정하기 위한 모드 탐색 및 결정을 위한 신호처리의 흐름도이다. 이 과정에서 계산량을 줄이기 위해서 표준에 더하여 새롭게 제안, 흐름도에 추가된 부분은 2개의 우회 경로이다.

그림 2에 나타난 첫 번째 우회 경로는 16x16 블록 크기 (mode 1)에 대한 움직임을 추정한 다음, 이 시점에서 추정된 움직임 벡터를 사용하여 skip 모드 조건을 만족할 가능성이 있으면, skip 모드 조건을 만족하는지를 먼저 검토하여, 조건을 만족하는 경우 나머지 다른 모드에 대한 탐색을 중단하고 해당 매크로블록의 부호화 모드를 skip 모드로 판정한다. 두 번째 우회 경로는 16x16, 16x8, 8x16 블록 크기에 대한 비용함수를 서로 비교하여, 16x16 블록 크기가 최소의 비용함수 값을 가지는 경우에는 8x8 크기 이하의 블록을 갖는 모드에 대한 비용 계산을 생략하는 것이다. 이는 위의 조건이 만족될 경우, 하위 모드로 진행함

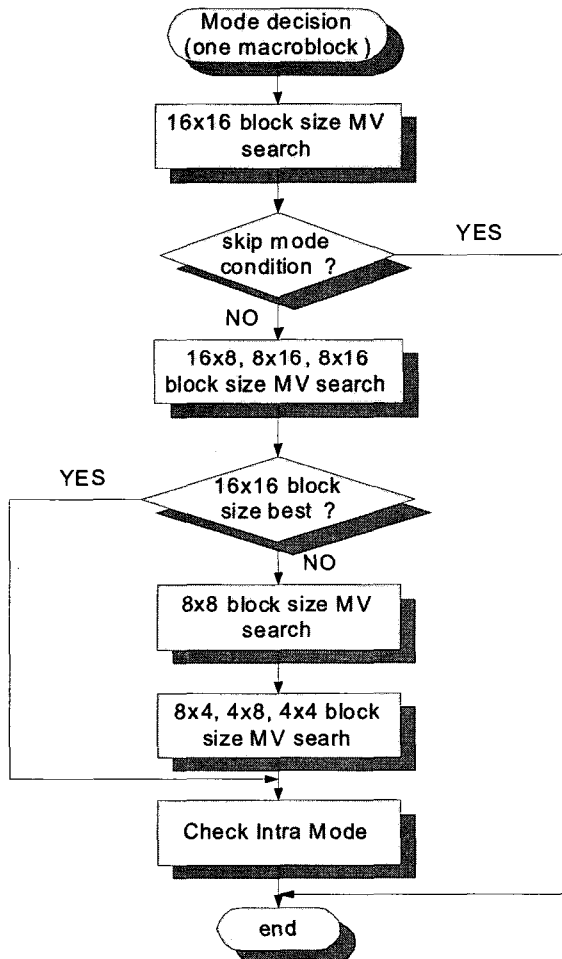


그림 2. 모드 탐색/결정을 위한 신호처리 흐름도
Fig. 2. Flowchart for mode search/decision

에 따라 SAD가 감소하여 잔류신호를 부호화하기 위한 비트는 줄어들겠지만 이와는 달리 전송해야 할 움직임 벡터의 수가 늘어남에 따라서 이를 표현하기 위해 필요한 비트수가 증가하기 때문에 전체적인 비용함수의 감소가 크지 않기 때문이다.

3.2 화소 단위 움직임 벡터 추정시 모드에 따른 탐색 패턴의 제한

제안하는 움직임 추정기는 화소 단위 움직임 벡터를 탐색할 때, 정방향 블록 (16x16 및 8x8)을 갖는 모드와 그 밖의 모드들에서의 탐색 패턴을 다르게 하였다. 그림 3과 4는 각각의 경우에 대한 제안하는 탐색 패턴을 보여주고 있다. 기본적으로 정방향 블록에 대한 움직임 벡터 추정에 비해서 그렇지 않은 블록에 대한 움직임 벡터 추정을 간략히 하는 구조를 채택하고 있는 것을 확인할 수 있을 것이다. 이는 제안하는 움직임 추정 과정에서 정방향 블록에 대한 움직임 벡터를 그 하위 모드에서 움직임 벡터 추정의 시작점으로 삼아서 움직임을 추정하는 관계로 정방향 블록의 움직임 추정 성능이 결과적으로 전체적인 움직임 추정 성능을 좌우하기 때문에 취한 불가피한 선택이다.

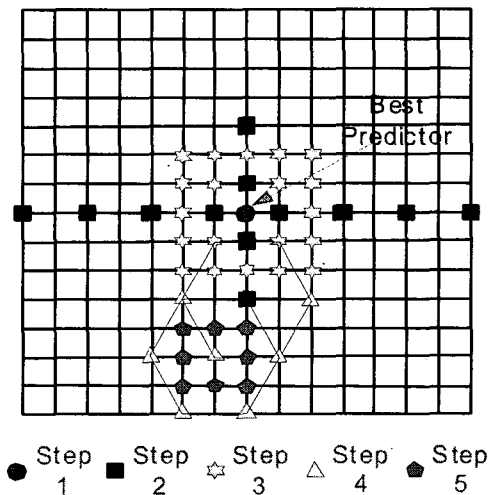


그림 3. 16x16, 8x8 블록에 대한 움직임 추정 탐색 패턴
Fig. 3. Search pattern of the motion estimation for the 16x16 and 8x8 blocks

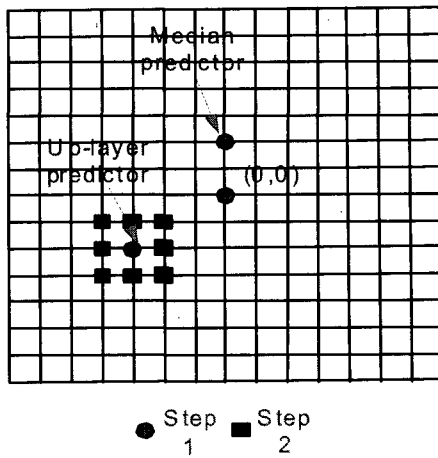


그림 4. 16x8, 8x16, 8x4, 4x8, 4x4 블록에 대한 움직임 추정 탐색 패턴
 Fig. 4. Search pattern of the motion estimation for the 16x8, 8x16, 8x4, 4x8 and 4x4 blocks

그림 3에 나타내었듯이 16x16 블록과 8x8 블록에 대한 화소 단위 움직임 추정의 탐색 패턴은 JM에 구현되어 있는 고속 움직임 추정 알고리즘의 탐색 패턴에서 Big-Hexagon 탐색 과정을 생략한 것을 제외하고는 동일하다^[3]. 즉, (0,0) 위치와 이웃 블록의 추정된 움직임 벡터들의 메디안을 구하는 메디안 예측 벡터 중 최소 비용값을 갖는 위치를 중심으로 크로스 탐색(Cross Search)을 수행한다. 이때, 영상 내의 움직임이 통계적으로 평행(좌우) 움직임이 수직(상하)움직임에 비하여 크기 때문에 평행 움직임에 대하여 넓은 영역의 탐색을 수행한다. 그 후, 크로스 탐색을 통하여 얻어진 최소 비용점을 중심으로 5x5 크기의 탐색점을 전역 탐색하여 크로스 탐색에서 얻어진 최소점을 보정한다. 다음으로 헥사 패턴으로 탐색 한후, 최소 비용값이 탐색 패턴에 중심에 있을 때, 그 점을 중심으로 사각형 패턴으로 탐색을 진행한다. 위의 탐색 과정 중 조기 종료 조건은 JM에 구현되어 있는 고속 움직임 추정 알고리즘에서 사용하는 조기 종료 조건을 사용하였다.

반면, 16x8, 8x16, 8x4, 4x8, 4x4 블록에 대해서는 그림 4에 보인 바와 같이 화소 단위 움직임 벡터의 추정 과정을 보다 간단하게 수정하였다. 즉 첫 번째 단계에서는 (0,0) 벡터, 이웃 블록에서 추정된 움직임 벡터들의 메디안을 통한 메디안 예측 벡터, 그리고 상위 레이어(16x8와 8x16 블록

의 상위 레이어는 16x16 블록을 의미하며, 8x4, 4x8, 4x4 블록의 상위 레이어는 8x8 블록을 의미한다)에서 추정된 움직임 벡터 등 세 개의 후보 중에서 최소 비용 함수값을 갖는 위치를 찾고, 두 번째 단계에서는 이 벡터를 중심으로 최인접의 8개 움직임 벡터 중에서 움직임 벡터를 최종적으로 추정하는 방식이다.

3.3 부화소 단위 움직임 벡터의 外插 (extrapolation)

부화소 위치의 움직임 벡터에 대해서 움직임 보상된 참조 블록과 처리 블록과의 정합도 평가를 하기 위해서는 참조 블록 내의 화소 값들의 보간하여 비교해야 하기 때문에, 화소 위치의 움직임 벡터에 비해서 매우 큰 부가적인 계산 부담을 준다. JM의 경우에는 참조 프레임 전체를 미리 보간하여 메모리에 저장한 다음, 부화소 단위 움직임 벡터 추정시 블록 정합도를 계산할 때에는 따로 보간을 하지 않고 메모리에 저장된 내용을 읽어와서 사용하는 구조를 가지고 있다. 그런데 이러한 방식은 부호기의 요구 메모리의 크기를 크게 증가시킨다. 단순 계산으로 1/4 화소 움직임 벡터 탐색을 사용할 경우, 단일 참조 프레임을 저장하는데 16배의 메모리가 필요하다. 본 논문에서 다루는 임베디드 환경의 경우에는 이와 같은 크기의 프레임 메모리를 사용한다는 것이 불가능하다. 따라서 참조 프레임은 입력 영상의 크기와 동일한 크기로 저장한 후, 부화소 단위의 움직임 벡터를 추정하기 위해서 보간이 필요한 경우에 필요한 위치값에 대해서만 직접(on-the-fly) 보간 연산을 행하는 방법을 사용할 수 밖에 없다. 이와 같은 경우에는 보간에 횟수에 비례하여 연산량이 증가하기 때문에, 알고리즘의 실시간성을 확보하기 위해서는 보간의 횟수는 최소화하여야 한다. 아래의 설명은 이와 같은 논리에 근거하여 보간의 횟수를 최소화시키면서도 부호화 단위의 움직임 벡터를 추정해내기 위한 제안하는 기법이며, 그 기본 원리는 다음과 같다. 각 탐색점에 대한 부호화 비용함수를 2차원 다항식의 형태로 모델링하고 화소 단위의 움직임 벡터에 대한 비용함수 값을 사용하여 모델 파라미터를 결정한 다음, 이 비용함수의 최소값(극소값)을 나타내는 위치를 계산해내어 이 위치와 가장 가까운 거리에 존재하는 1/4 화소 위치를 찾음으로써 부화소 단위 움직임 벡터를 추정하는 방식이다. 이 방법

을 사용할 경우, 위의 과정에서 추론된 움직임 벡터에 대해서 단 한번의 보간만을 행하여 정합도를 계산하고 그 결과를 모드 판정에 사용할 수 있기 때문에 부호기의 복잡도를 획기적으로 낮추는 것이 가능하다.

H.264 저복잡도 부호기에서는 식(1)과 같이 정의되는 비용함수를 움직임 벡터에 대한 비용함수로 사용하고 있다.

$$J_{x,y} = SAD_{x,y} + \lambda R_{x,y} \quad (1)$$

여기서 (x,y) 가 움직임 벡터의 해당 탐색점이다.

비용함수는 실제 움직임 벡터에서 최소값을 갖고, 그 부근에서는 실제 값으로부터의 거리에 비례하여 증가하는 것이 일반적이다. 이 성질을 이용하여 본 논문에서는 비용함수를 다음과 같이 2차원 다항식으로 모델링하였다.

$$J_{x,y} = J_{0,0} + c_1x + c_2y + c_3x^2 + c_4y^2 + c_5xy \quad (2)$$

모델에 대한 계수 값을 구하기 위하여 그림 5에 보이는 것과 같이 정수 화소 움직임에서 얻어진 최적 움직임 벡터와 주변 8화소의 실제로 계산된 비용 값을 이용하여 구해진다.

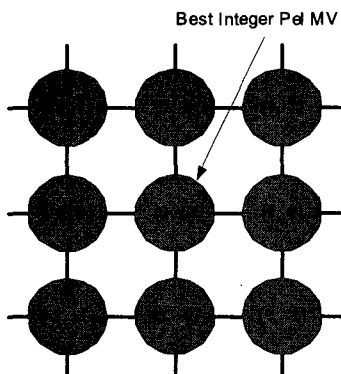


그림 5. 부화소 단위 위치에 대한 비용 값 모델의 계수를 구하기 위해 비용함수 값이 이용되는 정수 화소

Fig. 5. Integer-pels whose cost values are used for the calculation of the cost model coefficients at the sub-pel position

결정해야 할 계수는 5개이고 실제 이용 가능한 비용 값은 총 9개 이므로 식(3)과 같이 비용 함수(risk function)를 정의하여 모델의 각 계수 값을 결정하였다.

$$Risk = \sum_{(x,y) \in S} (D_{x,y})^2 \quad (3)$$

여기서, $D_{x,y}$ 는 식 (4)와 같다.

$$D_{x,y} = J_{0,0} + c_1x + c_2y + c_3x^2 + c_4y^2 + c_5xy - J_{x,y} \quad (4)$$

최적 모델 계수값은 식 (3)의 각 계수에 대한 편미분이 0이 될 때의 계수값을 최적 계수로 선택한다. 즉, 식(5) ~ 식(10)과 같다.

$$\frac{\partial Risk}{\partial c_j} = 0 \quad (5)$$

$$\frac{\partial Risk}{\partial c_1} = 2 \sum_{(x,y) \in S} xD_{x,y} = 0 \quad (6)$$

$$\frac{\partial Risk}{\partial c_2} = 2 \sum_{(x,y) \in S} yD_{x,y} = 0 \quad (7)$$

$$\frac{\partial Risk}{\partial c_3} = 2 \sum_{(x,y) \in S} x^2D_{x,y} = 0 \quad (8)$$

$$\frac{\partial Risk}{\partial c_4} = 2 \sum_{(x,y) \in S} y^2D_{x,y} = 0 \quad (9)$$

$$\frac{\partial Risk}{\partial c_5} = 2 \sum_{(x,y) \in S} xyD_{x,y} = 0 \quad (10)$$

식 (11)은 식(5) ~ 식 (10)에 대한 매트릭스 형태이다.

$$Y = A \cdot C^T \quad (11)$$

여기서, A는 식 (12), C는 식 (13), 그리고 Y는 식 (14)와 같다.

$$A = \begin{bmatrix} \sum_i x_i^2 & \sum_i x_i y_i & \sum_i x_i^3 & \sum_i x_i y_i^2 & \sum_i x_i^2 y_i \\ \sum_i x_i y_i & \sum_i y_i^2 & \sum_i x_i^2 y_i & \sum_i y_i^3 & \sum_i x_i y_i^2 \\ \sum_i x_i^3 & \sum_i x_i^2 y_i & \sum_i x_i^4 & \sum_i x_i^2 y_i^2 & \sum_i x_i^3 y_i \\ \sum_i x_i y_i^2 & \sum_i y_i^3 & \sum_i x_i^2 y_i^2 & \sum_i y_i^4 & \sum_i x_i y_i^3 \\ \sum_i x_i^2 y_i & \sum_i x_i y_i^2 & \sum_i x_i^3 y_i & \sum_i x_i y_i^3 & \sum_i x_i^2 y_i^2 \end{bmatrix} \quad (12)$$

$$C = [c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5] \quad (13)$$

$$Y = \begin{bmatrix} \sum_i x_i J_{x_i, y_i} - J_{0,0} \sum_i x_i \\ \sum_i y_i J_{x_i, y_i} - J_{0,0} \sum_i y_i \\ \sum_i x_i^2 J_{x_i, y_i} - J_{0,0} \sum_i x_i^2 \\ \sum_i y_i^2 J_{x_i, y_i} - J_{0,0} \sum_i y_i^2 \\ \sum_i x_i y_i J_{x_i, y_i} - J_{0,0} \sum_i x_i y_i \end{bmatrix} \quad (14)$$

여기서, 각 사용되는 비용 값의 위치는 그림 4와 같이 고정되어 있으므로 A는 식 (15)와 같은 고정된 값을 갖는다.

$$A = \begin{bmatrix} 6 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 6 & 4 & 0 \\ 0 & 0 & 4 & 6 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix} \quad (15)$$

식 (14)를 식 (16)와같이 전개한 하여 식 (11)을 풀면, c1, c2, c3는 각각 식 (17), (18), (19)가 된다.

$$Y = \begin{bmatrix} J_{1,1} - J_{-1,1} + J_{1,0} - J_{-1,0} + J_{1,-1} - J_{-1,-1} \\ J_{1,1} + J_{0,1} + J_{-1,1} - J_{1,-1} - J_{0,-1} - J_{-1,-1} \\ J_{1,1} + J_{-1,1} + J_{1,0} + J_{-1,0} + J_{1,-1} + J_{-1,-1} - 6J_{0,0} \\ J_{1,1} + J_{0,1} + J_{-1,1} + J_{1,-1} + J_{0,-1} + J_{-1,-1} - 6J_{0,0} \\ J_{1,1} - J_{-1,1} - J_{1,-1} + J_{-1,-1} \end{bmatrix} \quad (16)$$

$$c_1 = \frac{1}{6} (J_{1,1} - J_{-1,1} + J_{1,0} - J_{-1,0} + J_{1,-1} - J_{-1,-1}) \quad (17)$$

$$c_2 = \frac{1}{6} (J_{1,1} + J_{0,1} + J_{-1,1} - J_{1,-1} - J_{0,-1} - J_{-1,-1}) \quad (18)$$

$$c_5 = \frac{1}{4} (J_{1,1} - J_{-1,1} - J_{1,-1} + J_{-1,-1}) \quad (19)$$

또한, A의 sparse 매트릭스를 이용하여 c3와 c5의 값을 구한다.

$$\begin{bmatrix} 6 & 4 \\ 4 & 6 \end{bmatrix} \begin{bmatrix} c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} S_x \\ S_y \end{bmatrix} \quad (20)$$

$$\begin{bmatrix} c_3 \\ c_4 \end{bmatrix} = \frac{1}{10} \begin{bmatrix} 3S_x - 2S_y \\ 3S_y - 2S_x \end{bmatrix} \quad (21)$$

$$c_3 = \frac{1}{10} (J_{1,1} + J_{-1,1} + J_{1,-1} + J_{-1,-1} + 3J_{1,0} + 3J_{-1,0} - 2J_{0,1} - 2J_{0,-1} - 6J_{0,0}) \quad (22)$$

$$c_4 = \frac{1}{10} (J_{1,1} + J_{-1,1} + J_{1,-1} + J_{-1,-1} - 2J_{1,0} - 2J_{-1,0} + 3J_{0,1} + 3J_{0,-1} - 6J_{0,0}) \quad (23)$$

계수가 결정되면, 비용함수의 모델 식에 대입하여 비용 함수 모델을 완성하고 이 모델에서 최소값을 나타내는 위치를 찾음으로써 부화소 단위의 움직임 벡터를 추정하게 된다. 식 (24) ~식 (27)은 모델에 의한 최적 부화소 단위 움직임 벡터의 결정 방법이다.

$$\frac{\partial J_{x,y}}{\partial x} = c_1 + 2c_3x + c_5y = 0 \quad (24)$$

$$\frac{\partial J_{x,y}}{\partial y} = c_2 + 2c_4y + c_5x = 0 \quad (25)$$

$$\begin{bmatrix} 2c_3 & c_5 \\ c_5 & 2c_4 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} -c_1 \\ -c_2 \end{bmatrix} \quad (26)$$

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \frac{1}{4c_3c_4 - c_5^2} \begin{bmatrix} c_2c_5 - 2c_1c_4 \\ c_1c_5 - 2c_2c_3 \end{bmatrix} \quad (27)$$

최종적으로 움직임 벡터는 식 (28)과 식(29)과 같이 범위를 제한함으로써 추정된다.

$$m_x = \begin{cases} (\text{int})(4\hat{x} + 0.5) & |\hat{x}| < 1 \\ \text{sgn}(\hat{x}) & \text{elsewhere} \end{cases} \quad (28)$$

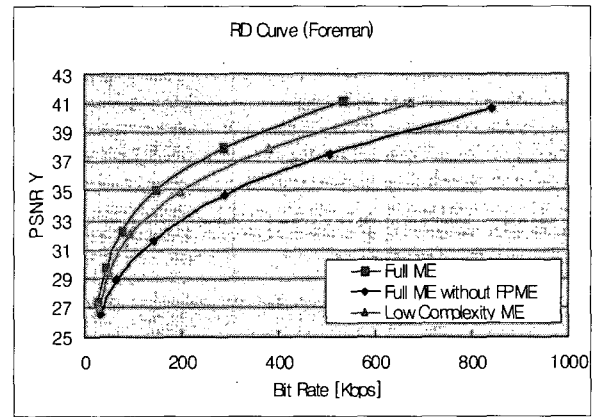
$$m_y = \begin{cases} (\text{int})(4\hat{y} + 0.5) & |\hat{y}| < 1 \\ \text{sgn}(\hat{y}) & \text{elsewhere} \end{cases} \quad (29)$$

4. 실험결과

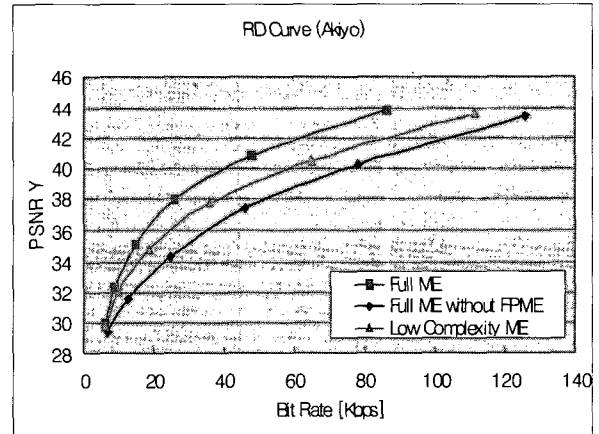
먼저 앞의 3.2절과 3.3절에서 각각 설명한 화소 단위 움직임 벡터 추정시 탐색 패턴의 제한과 부화소 단위 움직임 벡터 外插 (extrapolation)의 성능에 대해서 분석하였다. 두 가지의 QCIF 동영상, Foreman과 Akiyo 및 한 가지의 CIF 동영상 Mobile&calendar를 사용하여 각각 300 프레임에 대해서 6개의 QP 값을 사용한 constant-quality 부호화를 행하여 부호화한 결과로 얻게 되는 평균 비트율과 평균 PSNR을 분석하였다. 비교 대상으로는 JM을 사용한 표준 알고리즘 (화소 단위 움직임 벡터를 초기 탐색점으로 하여 다이아몬드 세이프를 탐색하여 최적의 부화소 움직임 벡터

를 탐색)과 JM을 사용하되 움직임 벡터의 추정을 화소 단위까지만 탐색하는 화소 단위 움직임 추정 알고리즘을 선택하여 RD-성능을 비교하였다.

그림 6은 움직임 추정 기법에 따른 RD-성능을 비교하여 보이고 있다. 제안하는 외삽에 의한 부화소 움직임 벡터의 추정 기법을 사용함으로써 JM 표준 알고리즘과 화소 단위 움직임 벡터 추정 알고리즘과의 중간적인 성능을 얻을 수 있음을 보여주고 있다. 비교적 복잡한 움직임 있는 Foreman 영상과 단순한 움직임의 Akiyo 영상 모두 유사한



(a)



(b)

그림 6. 제안하는 부화소 움직임 벡터 외삽 기법의 RD-성능: (a) Foreman, (b) Akiyo.

Fig. 6. RD-performance of the proposed sub-pel motion estimation using motion vector extrapolation. (a) Foreman, (b) Akiyo

경향을 보인다.

제안하는 부화소 움직임 벡터의 外插을 사용하는 경우 화소 단위 움직임 추정 알고리즘에 비해서 늘어나는 연산은 외삽에 의해 추정된 움직임 벡터를 보상한 영상을 얻기 위한 1회의 영상 보간이다. 반면에 16x8, 8x16, 8x4, 4x8, 4x4 블록에 대한 화소 단위 움직임 벡터 추정시 탐색 패턴을 줄임으로써 화소 단위 움직임을 추정하는 표준 알고리즘과 비교할 때에도 외삽에 의한 부화소 움직임 벡터 추정을 포함하는 제안 알고리즘의 전체적인 부호화 소요 시간이 50% 이하의 수준으로 줄어든다. 그림 7은 이 결과를 잘 보여주고 있다.

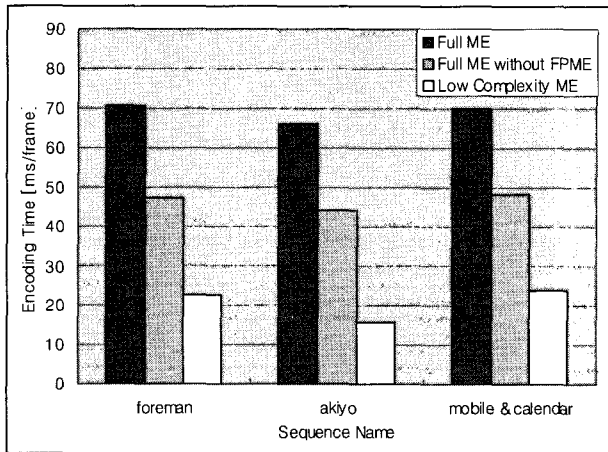


그림 7. 부호화 시간 비교
Fig. 7. Comparison of encoding time

Forman, Irene, Student, Mother & Daughter 등 4 종류의 7.5fps QCIF 영상을 64kbps로 부호화하는 경우에, 인텔 PXA255 400MHz PDA 상에서 임베디드 H.264 부호기의 구현 가능성에 대해서 실험하였다. 실험에 적용한 부호화 도구의 사용 여부는 앞에서 보인 표 1과 같다. 이 경우, 실험에 사용한 인텔 PXA255 400MHz 머신의 경우, 움직임 추정 및 모드 결정 알고리즘의 요구 계산량은 화소 단위 움직임 추정에는 매크로 블록당 30회 이내의 SAD 연산, 부화소 움직임 추정에는 15회 이하의 SAD 연산을 포함해야 하는 것으로 분석되었다.

표 2는 표준 JM에 포함된 고속 움직임 추정 알고리

즘을 사용하는 경우 RD 성능과 SAD 연산 횟수를 분석한 결과이다. SAD 연산 횟수가 구현하고자 하는 임베디드 환경에 비해서 약 4배 이상 많이 발생함을 보여주고 있다.

표 2. 표준 JM 고속 움직임 추정 알고리즘의 RD 성능 및 복잡도
Table 2. RD performance and complexity of the reference fast motion estimation of the JM

시퀀스	PSNR (dB)	비트율 (kbps)	SAD 연산 횟수	
			화소 움직임	부화소 움직임
Foreman	34.47	63.83	186.63	45.44
Irene	37.61	64.01	102.38	41.50
Students	38.40	64.00	91.06	41.13
M & D	38.16	64.01	112.88	42.56

표 3은 제안하고 있는 임베디드용 초저복잡도 알고리즘에 RD 성능과 SAD 연산 횟수를 분석한 결과이다. 제안하는 알고리즘의 경우에는 SAD 연산 횟수가 구현하고자 하는 임베디드 환경에서도 사용할 수 있는 수준에 도달하였음을 보여주고 있다.

표 3. 제안하는 임베디드용 초저복잡도 부호화 알고리즘의 RD 성능 및 복잡도
Table 3. RD performance and complexity of the proposed very low-complexity algorithm

시퀀스	PSNR (dB)	비트율 (kbps)	SAD 연산 횟수	
			화소 움직임	부화소 움직임
Foreman	33.84	63.89	29.88	13.56
Irene	37.32	63.92	17.88	10.31
Students	37.88	63.92	15.63	10.25
M & D	37.71	64.01	18.44	11.31

표 4는 표준 JM 고속 움직임 추정 알고리즘과 제안하고 있는 임베디드용 초저복잡도 알고리즘의 RD 성능과 복잡도를 비교하여 보여주고 있다. 제안하는 임베디드용 초저복잡도 알고리즘이 표준 JM 방식에 비하여 평균적으로 0.4dB 정도의 화질 열화를 보이지만, 알고리즘의 요구 복잡도는 15% 이하로 줄어들었음을 알 수 있다.

표 4. 표준 JM 고속 알고리즘과 제안하는 임베디드용 초저복잡도 알고리즘의 성능비교

Table 4. Performance comparison between the reference fast algorithm of the JM and the proposed very low-complexity algorithm

시퀀스	PSNR 손실 (dB)	SAD 연산의 감소율(%)	
		확소 움직임	부확소 움직임
Foreman	0.63	84.94%	69.82%
Irene	0.29	83.32%	74.85%
Students	0.52	84.09%	74.92%
M & D	0.45	84.60%	73.19%
Average	0.47	84.24%	73.19%

5. 결론

본 논문에서는 H.264 레퍼런스 코덱인 Joint Model (JM)에서 사용하고 있는 몇 가지 부호화 도구의 사용을 제한하고, 몇 가지 부호화 알고리즘에 대해서는 조기 종료 조건과 우회 조건을 도입하여 고속화함으로써 임베디드 환경에서 응용 가능한 수준의 초저복잡도 부호기를 구현할 수 있음을 보였다. 제안된 초저복잡도 부호기는 JM의 15% 수준의 연산 복잡도로 동작하지만, JM에 비하여 단지 0.4 dB 이내의 PSNR 열화를 나타낼 뿐이다. 이 부호기를 인텔 PXA255 400MHz CPU를 사용한 PDA에 이식하여 multimedia messaging service (MMS) 서비스를 제공하기 위한 임베디드 비디오 부호기로서 안정적으로 동작함을 확인하였다.

참고 문헌

- [1] Iain E.G. Richardson, H.264 and MPEG-4 Video Compression, John Wiley & Sons Ltd, 2003.
- [2] JVT Reference Software version 8.2, <http://bs.hhi.de/~suehring/tml/download>.
- [3] Zhibo Chen, Peng Zhou, Yun He, "Fast integer pel and fractional pel motion estimation for JVT," JVT-F017r1.doc, Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, 6th meeting, Awaji, JP, 5-13 Dec. 2002.
- [4] Zhibo Chen, Peng Zhou, Yun He, "Fast motion estimation for JVT," JVT-G016.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 7th meeting, Pattaya II, Thailand, 7-14 Mar. 2003.
- [5] Zhibo Chen, JianFeng Xu, Yun He, "Simplifications on fast motion estimation," JVT-H026.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 8th meeting, Geneva, Switzerland, 23-27 May 2003.
- [6] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. LI, G. N. Feng, D. J. Wu, S. Wu, "Fast mode decision for intra prediction," JVT-G013.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 7thmeeting, Pattaya II, Thailand, 7-14 Mar. 2003.
- [7] K. P. Lim, S. Wu, D. J. Wu, S. Rahardja, X. Lin, F. Pan, Z. G. LI, "Fast INTER mode selection," JVT-I020.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 9th meeting, San Diego, US, 2-5 Sep. 2003.
- [8] Byeungwoo Jeon, Jeyun Lee, "Fast mode decision for H.264," JVT-J033.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 10th meeting, Waikoloa, USA, 8-12 Dec. 2003.
- [9] Deepak S. Turaga, Tsuhan Chen, "Classification based mode decisions for video over networks," IEEE Transactions on Multimedia, vol.3, no.1, Mar. 2001.

저 자 소 개



유 영 일

- 2003년 : 국민대학교 전자공학과 졸업
- 2005년 : 국민대학교 전자공학과 공학석사
- 2005년~현재 : 국민대학교 전자공학과 박사과정
- 주관심분야 : 비디오 코덱, DMB



김 용 태

- 2004년 : 국민대학교 전자공학과 졸업
- 2004년~현재 : 국민대학교 전자공학과 석사과정
- 주관심분야 : 비디오 코덱, 디지털 통신 및 임베디드 시스템



이 승 준

- 1997년 : 서울대학교 전자공학과 공학박사
- 1998년 : 스탠포드대학 객원연구원
- 1997년~2000년 : SK텔레콤 선임연구원
- 2001년~현재 : 엠큐브웍스(주) 연구소장
- 주관심분야 : 멀티미디어 신호처리, 통신방송 융합 멀티미디어 서비스



강 동 욱

- 1986년 2월 : 서울대학교 전자공학과 졸업
- 1988년 2월 : 서울대학교 전자공학과 공학석사
- 1995년 2월 : 서울대학교 전자공학과 공학박사
- 2000년~2001년 : Lucent Technology MTS
- 1995년~현재 : 국민대학교 전자정보통신공학부 부교수
- 주관심분야 : 비디오 코딩, 디지털 TV, 멀티미디어 신호처리



김 기 두

- 1980년 2월 : 서강대학교 전자공학과 졸업
- 1988년 8월 : The Pennsylvania State University, MS(Electrical Eng.)
- 1990년 12월 : The Pennsylvania State University, Ph.D.(Electrical Eng.)
- 1980년 3월~1985년 12월 : 국방과학연구소 연구원
- 1998년 2월~1999년 2월 : USCD, USA 연구교수
- 1991년 2월~현재 : 국민대학교 전자정보통신공학부 교수
- 주관심분야 : 디지털 신호처리, 디지털 통신