

논문 2008-45SP-6-1

다중참조 및 가변블록 움직임 추정을 위한 고속 참조영상 선택 방법

(Fast Frame Selection Method for Multi-Reference and Variable Block Motion Estimation)

김 성 대*, 선우 명 훈*

(Sung Dae Kim and Myung Hoon Sunwoo)

요 약

이 논문은 다중참조 및 가변블록 움직임 추정의 연산량을 효율적으로 줄이기 위해 세 가지 참조영상 선택 방법들을 소개한다. 제안된 RSP (Reference Selection Pass) 방법은 참조영상 선택의 추가적인 연산을 최소화 할 수 있고 MFS (Modified Frame Selection) 방법은 참조영상 선택 과정 중 영상의 움직임을 고려하여 참조영상 선택 시 연산 횟수를 기존 방식에 비해 17% 감소시킨다. 또한 TPRFS (Two Pass Reference Frame Selection) 방법은 H.264/AVC에서 요구하는 가변블록 움직임 추정을 지원하기 위한 추가적인 연산을 블록 크기에 따라 선택되는 참조영상의 특성을 이용하여 최소화 한다. 실험 결과 제안한 방식은 기존의 방식에 비해 화질의 열화 없이 50% 이상의 움직임 추정의 연산량을 감소시킬 수 있다. 또한 제안한 참조영상 선택 방법은 움직임 추정의 주된 연산인 블록정합 단계와 별개로 수행이 되기 때문에 기존의 어떠한 단일참조 고속 움직임 탐색 방법과도 같이 사용되어 효율적으로 다중참조 및 가변블록 움직임 추정 연산을 지원 할 수 있다.

Abstract

This paper introduces three efficient frame selection schemes to reduce the computation complexity for the multi-reference and variable block size Motion Estimation (ME). The proposed RSP (Reference Selection Pass) scheme can minimize the overhead of frame selection. The MFS (Modified Frame Selection) scheme can reduce the number of search points about 18% compared with existing schemes considering the motion of image during the reference frame selection process. In addition, the TPRFS (Two Pass Reference Frame Selection) scheme can minimize the frame selection operation for the variable block size ME in H.264/AVC using the character of selected reference frame according to the block size. The simulation results show the proposed schemes can save up to 50% of the ME computation without degradation of image quality. Because the proposed schemes can be separated from the block matching process, they can be used with any existing single reference fast search algorithms.

Keywords : H.264/AVC, Motion Estimation, Multi-Reference, Variable Block Size, Low Power Design

I. 서 론

최근의 다양한 멀티미디어 및 통신 분야에서 요구하

는 높은 압축률을 지원하기 위해 새로운 영상 압축의 표준인 H.264/AVC^[1]가 소개되었다. H.264/AVC는 기존의 다른 영상 압축 표준인 MPEG-2, MPEG-4, H.263 등과 마찬가지로 화면 간 부호화와 이산여형 변환 기반의 복합적인 코딩 방식을 기반으로 한다. 하지만 코딩의 효율을 보다 높이기 위해 특히 움직임 추정 연산부분에서 다중참조 영상을 사용하고, 가변블록 움직임 추정을 지원하며, 1/4 화소 보간을 사용한다^[2]. 하지만 이런 새로운 방식들은 메모리 접근과 연산량을 증가시키기 때문에 실시간 구현에 가장 큰 걸림돌이 되고 있다.

* 아주대학교 정보통신대학 전자공학부 SOC 연구실
(School of Electrical and Computer Engineering,
Ajou University)

※ 본 연구는 교육인적 자원부 2단계 BK(Brain Korea) 21 과제와 지식경제부가 지원하는 국가 반도체 연구개발사업인 “시스템집적반도체기반기술개발사업(시스템 IC 2010)” 그리고 반도체설계교육센터(IDECC)의 지원을 받아 수행되었습니다.

접수일자: 2008년6월13일, 수정완료일: 2008년10월10일

이런 다양한 기술 중 다중참조 움직임 추정은 가장 많은 메모리 접근 및 연산량을 요구하기 때문에 이를 해결하기 위해 많은 연구들이 진행되었다. 이런 연구들은 크게 3가지 분류로 나누어 볼 수 있다. 첫 번째 분류는 3차원 움직임 추정으로 명명하였다^[3~4]. 이런 기술은 본래 영상의 X좌표와 Y좌표를 이용하여 2차원으로 수행되던 움직임 추정 연산을 X좌표, Y좌표, 시간축 등의 3차원으로 확장 시킨다. 그렇지만 이런 연구는 넓은 범위의 참조 영상에 대한 메모리 접근을 요구하기 때문에 하드웨어 기반의 구현 보다는 소프트웨어 기반의 구현에 적합한 방식이다. 두 번째 분류는 움직임 벡터 조합에 따른 방식이다^[5~6]. 이런 방식은 다중 참조 움직임 추정을 위해 각각의 참조 영상에 대해 모두 움직임 추정 연산을 수행하는 것이 아니라 최초의 참조 영상에 대해 움직임 추정 연산을 수행 한 이후에는 그 다음 참조 영상에 대해서는 이전에 수행한 움직임 추정 연산의 움직임 벡터의 벡터 합을 이용하여 새로운 움직임 벡터를 생성한다. 연산량을 대폭 줄일 수 있는 방식이지만 이전 움직임 추정 연산의 움직임 벡터를 기반으로 연산을 하기 때문에 움직임 벡터의 정확성이 필요하여 상대적으로 정확성이 떨어지는 고속 탐색 움직임 추정 연산과 같이 병행하여 사용하기에는 화질 열화가 크다는 단점이 있다. 마지막으로 세 번째 분류는 참조 영상 선택 기술이다^[7~9]. 이 기술은 기존의 다양한 고속 탐색 알고리즘과 병행하여 수행이 가능하고 다중 참조움직임 추정의 연산량을 효율적으로 감소시킬 수 있지만 참조 영상을 선택하기 위해 부가적인 연산이 필요하다는 단점이 있다.

본 고에서는 다중참조 움직임 추정과 가변블록 움직임 추정을 지원하는 효율적인 세 가지 참조 영상 선택 방식을 제안한다. 첫 번째 방식은 RSP (Reference Selection Pass) 방식으로 이웃한 블록의 움직임 벡터를 사용하여 참조 영상 선택을 위해 필요한 연산을 줄이는 방식이다. 두 번째 방식은 MFS (Modified Frame Selection) 방식으로 참조 영상 선택을 위해 필요한 탐색 지점 수를 영상의 움직임을 고려하여 화질의 열화 없이 줄이는 방식이다. 세 번째 방식은 TPRFS (Two Pass Reference Frame Selection) 방식으로 가변 블록 움직임을 추정하기 위해 참조 영상을 16x16 블록과 8x8 블록 형식으로 나누어 처리하는 방식이다. 제안한 방식들은 화질의 열화 없이 움직임 추정 연산의 연산량을 기존에 비해 50% 이상 감소시킬 수 있다.

본 고의 구성은 다음과 같다. II장에서는 기존의 다

중 참조 움직임 추정 방법을 살펴보고 III장에서는 새로운 참조 영상 선택 방식을 제안한다. 그리고 IV장에서는 기존의 방식과 제안한 방식의 성능을 비교하고 마지막으로 V장에서 결론을 맺는다.

II. 기존의 다중참조 움직임 추정 방법

이 장에서는 다중참조 움직임 추정을 지원하기 위한 기존의 방법에 대해 소개한다. CBFS (Center-Biased Frame Selection) 방법은 미리 정의한 탐색 위치를 사용하여 참조 영상을 선택하는 방식이다^[7]. 그림 1은 미리 정의된 2가지 탐색 위치를 보여준다. LCP (Large Cross Pattern)와 LSP (Large Square Pattern)는 논문에 소개된 다른 탐색 위치에 비해 선택된 참조영상의 정확도 및 코딩 시 화질에서 월등한 두 가지 패턴이다.

이런 참조영상 선택 방식은 움직임 추정 연산의 블록 매칭 과정에서 분리되어 있기 때문에 기존의 단일참조 고속 움직임 추정 방식을 사용하여 다중참조 움직임 추정을 지원 할 수 있게 한다. 그렇지만 CBFS 방식은 참조영상을 선택하기 위해 부가적인 연산을 필요로 한다. 그림 1에 소개 된 2가지 탐색 패턴은 각각의 참조영상에서 9지점을 비교해야만 참조영상을 선택 할 수 있다. 따라서 지원하는 참조영상의 수에 따라 요구되는 부가적인 연산량은 다음 식 (1)을 이용해 계산 할 수 있다.

$$\text{Computation} = 9 \times \text{Num of Reference Frames} \quad (1)$$

예를 들어, 만약 5장의 참조영상을 지원한다면 CBFS 방식은 45 지점의 부가적인 연산이 필요하게 된다. Huang은 화면 내 부호화와 이전 참조 영상의 움직임 추정 연산 결과를 사용하여 모든 참조영상을 다 탐색하지 않고 중간에 연산을 중단 시키는 방법^[8]을 제안하였다. Saponara는 참조영상의 수를 동적으로 제어하는 방법^[9]을 제안하였다. 이런 방법들은 기존의 단일참조 움직임 추정 방법과 같이 사용될 수 있어 CBFS와 유사한

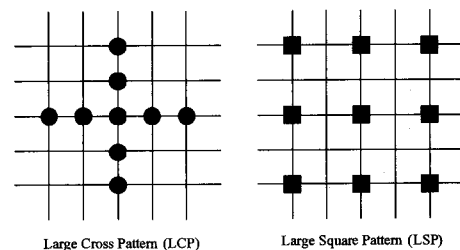


그림 1. 미리 정의된 탐색 위치
Fig. 1. Pre-defined patterns.

장점을 가지지만 탐색 과정이 언제나 시간축의 순서대로 시간 상 가까운 지점의 참조영상에서 먼 참조영상을 순서로 진행이 되기 때문에 참조영상의 선택을 최적화하지 못한다는 단점을 가지고 있다. AFMFSA (Adaptive and Fast Multi-Frame Selection Algorithm) 은 영상의 각 블록에 대해 물체의 경계면에 포함되어 있는지를 판별^[10]하여 기존의 움직임 추정 연산 방식에 비해 56~74.2%의 연산 량을 감소시키는 성능을 보여 준다 하지만 AFMFSA는 물체의 경계면을 판별하기 위해 MAD (Mean Absolute Deviation)을 사용하기 때문에 각각의 블록에 대한 필요 연산량이 많아진다는 단점이 있다.

III. 제안하는 참조영상 선택 방법

단일참조 고속 탐색 알고리즘은 오랜 기간 동안 영상 압축 부분에서 널리 연구^[11~12]되어 왔다. 이렇게 연구된 다양한 탐색 알고리즘을 다중참조 움직임 추정에 사용하기 위해 우리는 기존의 간단하고 하드웨어 구현에 적합한 CBFS 방식을 개선하였다. 앞서서도 언급하였듯이 CBFS 방식은 참조영상 선택 및 가변블록 크기를 지원하기 위해 추가적인 연산을 필요로 한다. 그러므로 우리는 3가지 참조영상 선택 방법을 제안하여 상기 문제점을 해결하였다.

1. 제안한 RSP(Reference Selection Pass) 방식

참조영상 선택은 추가적인 연산이 필요하기 때문에 이를 최소화하기 위해 AFMFSA는 현재 블록이 물체의 경계면에 포함되어 있는지 여부를 MAD 연산을 통해 판별 한 후 경계면에 있지 않으면서 주변의 이웃한 블록들이 같은 참조영상을 선택했을 때 따로 참조영상 선택에 필요한 연산을 수행하지 않고 기존에 선택된 참조영상을 사용하는 방식을 제안하였다. 하지만 MAD 연산은 현재 블록의 평균값을 필요로 하기 때문에 추가적인 연산이 필요하다. 따라서 우리는 MAD를 통한 물체의 경계면 판별을 주변의 움직임 벡터 값을 사용한 비교로 대체하여 추가적인 연산을 간략화 시켰다. 움직임 벡터는 영상의 움직임을 나타내기 때문에 그 값의 크기를 비교하여 영상의 움직임 정도를 판단 할 수 있다. 그리고 영상의 움직임이 적다면 주변의 블록과 유사한 특성을 가지기 때문에 우리는 MAD를 사용한 판단기준을 움직임 벡터를 사용하도록 변경하였다.

그림 2는 현재 블록(E)과 이웃한 블록(A, B, C, D)의

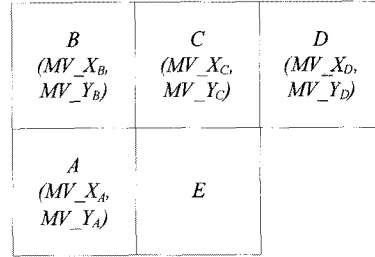


그림 2. 현재 블록과 이웃한 블록의 위치
Fig. 2. Position of current block and adjacent blocks.

위치 및 각각의 움직임 벡터 값을 표시한 그림이다. 그림의 MV_{X_A}, MV_{Y_A} 는 A 블록의 움직임 벡터 값을 나타내고 마찬가지로 MV_{X_B}, MV_{Y_B} 는 B 블록의 움직임 벡터 값을 나타낸다.

제안한 RSP 방식은 다음과 같이 동작한다.

1. 이웃한 블록 A, B, C, D의 사용가능 여부를 판별한다. 만약 모든 블록이 사용가능하면 다음 단계로 진행한다.

2. 이웃한 블록에서 선택된 참조영상을 비교한다. 만약 모든 블록이 같은 참조영상을 선택하였다면 다음 단계로 진행한다.

3. 모든 이웃한 블록의 움직임 벡터의 절대 합을 비교한다. 움직임 벡터의 절대 합은 식 (2)를 통해 얻는다. 예를 들어, 블록 A의 움직임 벡터의 절대 합은 그림 2의 MV_{X_A} 와 MV_{Y_A} 를 통해 계산된다. 이렇게 계산된 각 블록의 움직임 벡터의 절대 합이 미리 설정한 경계 값 (본 고에서는 실험 시 경계 값으로 4를 사용) 보다 모두 작을 경우 우리는 현재 블록에 대한 참조영상 선택 과정을 수행하지 않고 이웃한 블록의 참조영상을 현재 블록의 참조영상으로 사용하였다.

$$Absolute\ motion\ vector = |MV_X| + |MV_Y| \tag{2}$$

2. 제안한 MFS(Modified Frame Selection) 방식

CBFS 방식의 LCP, LSP 패턴은 각각의 참조영상에 대해 V장의 참조영상을 지원 할 시 45 지점을 비교하여야만 한다. LCP 또는 LSP 이외의 다른 패턴들은 영상 화질의 MAE (Mean Absolute Error) 비교와 참조영상의 정확도 면에서 떨어지므로 보다 적은 지점을 탐색하는 다른 패턴들은 고려하지 않았다. 이런 탐색 지점의 횟수를 화질의 열화 없이 줄이기 위해 우리는 MFS 방식을 제안하였다. CBFS 방식은 고정된 위치에서 탐

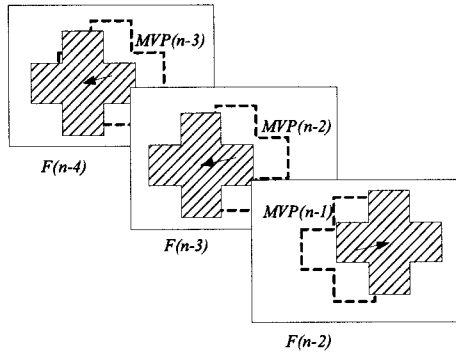


그림 3. 제안한 MFS 방식
Fig. 3. Proposed MFS scheme.

색을 하기 때문에 영상의 움직임 등을 고려하지 않는다. 반면에 제안한 MFS 방식은 각각의 참조영상에서 나타나는 영상의 움직임을 참조영상 선택 시에 적용하여 각 참조영상의 탐색시작 지점을 이전 참조영상의 움직임 벡터에 따라 조절한다. 그림 3은 영상의 움직임을 고려한 MFS 방식을 보여준다. 그림 3의 $F(n-2)$, $F(n-3)$, 그리고 $F(n-4)$ 는 참조 영상을 나타낸다. $MVP(n-1)$, $MVP(n-2)$, 그리고 $MVP(n-3)$ 은 참조영상 선택 과정 중에 생성된 중간 움직임 벡터 값을 나타낸다. 예를 들어, $MVP(n-2)$ 는 참조영상 $F(n-2)$ 의 중간 움직임 벡터 값이다. 제안한 MFS 방식에서는 최근의 참조영상의 중간 움직임 벡터 값이 현재 참조영상에 대한 탐색의 시작 지점이 된다. 다시 말하면, $F(n-2)$ 의 탐색시작 지점은 이전 $F(n-1)$ 의 중간 움직임 벡터 값인 $MVP(n-1)$ 에 따라 결정되고, 마찬가지로 $F(n-3)$ 의 탐색시작 지점은 $MVP(n-2)$ 에 따라 선택된다.

표 1은 본래 CBFS에서 사용한 고정된 위치의 패턴을 사용하였을 때와 시작위치가 변경된 패턴을 사용하였을 때의 참조영상 선택의 정확도를 보여준다.

표에서 나타나듯이, 제안한 MFS 방식은 기존의 방식

표 1. 기존방식과 제안한 방식 사이의 참조영상 정확도

Table 1. Hit ratio comparisons among the proposed scheme and original schemes.

영상 종류	LCP [7]	LSP [7]	MFS
Claire	96.86%	91.78%	97.75%
Miss A	70.72%	50.41%	72.54%
Sales	98.59%	95.95%	99.15%
Garden	82.75%	52.34%	83.12%
Football	94.20%	86.76%	94.34%
Tennis	73.43%	65.33%	73.60%
Average	86.09%	73.76%	87.10%

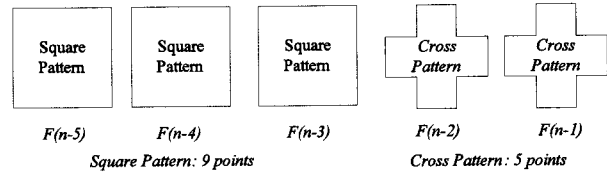


그림 4. 참조영상에 따른 변경된 탐색 패턴
Fig. 4. Modified search patterns for reference frame.

에 비해 보다 높은 정확도를 보인다. 따라서 우리는 화질의 열화 없이 탐색 횟수를 감소시킬 수 있었다.

그림 4는 각각의 참조영상에 대해 적용한 변경된 탐색 패턴을 보여준다. 여러 영상에 대한 실험 결과에 따라, 현재 블록에 대한 시간상의 거리에 비례하여 중앙 위치에서 최적의 지점이 위치 할 확률이 감소하는 성질을 이용하여 보다 정확한 탐색을 위해 현재 블록에서 멀리 위치한 참조영상의 경우 보다 많은 지점을 탐색하는 사각 패턴(Square Pattern)을 사용하였고 가까이 위치한 참조영상의 경우는 상대적으로 적은 지점을 탐색하는 격자 패턴 (Cross Pattern)을 사용하였다. 따라서 제안한 MFS 방법은 기존의 방식에 비해 5장의 참조영상을 지원 할 경우 탐색 횟수를 17% 감소 시켰다. 이런 참조영상 선택 연산은 앞에서 제안한 RSP 방식에 의해 바로 참조영상이 결정되는 경우가 아니라면 모든 블록에 대해서 연산이 수행되어야 하기 때문에 실제 영상의 부호화에 있어서 성능 증가는 처리하는 영상의 크기에 비례하여 증가한다.

3. 가변블록 움직임 추정을 위한 TPRFS

(Two Pass Reference Frame Selection) 방식

그림 5는 JM 참조 소프트웨어^[13]에서 매크로 블록의 모드를 결정하기 위한 연산의 과정을 보여준다. 매크로 블록은 7가지 화면 간 부호화 모드($P16x16$, $P16x8$, $P8x16$, $P8x8$, $SUB8x4$, $SUB4x8$, $SUB4x4$)와 2가지 화면 내 부호화 모드($I4MB$, $I16MB$)로 구성된다. JM 참조 소프트웨어의 경우 가변블록 움직임 추정을 지원하기 위해 4가지 부호화 모드($P16x16$, $P16x8$, $P8x16$, $P8x8$)에 대해 반복적으로 예측과정을 수행하고 나머지 3가지 부호화 모드($SUB8x4$, $SUB4x8$, $SUB4x4$)에 대해 $P8x8$ 모드와 같은 참조영상에 대해서 연산을 수행한다.

제안한 MFS 방식으로 가변블록 움직임 추정을 지원하기 위해서는 우리는 각각의 블록 크기에 따라 MFS 방식을 개별적으로 수행해야만 했다. 이런 방식은 많은 연산 량을 필요로 함으로 우리는 블록 크기에 따른

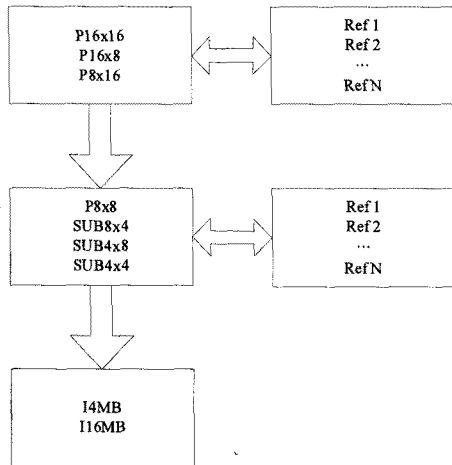


그림 5. JM 참조 소프트웨어의 매크로 블록 모드 결정 연산 과정

Fig. 5. Operation flow of the macro block mode decision in JM reference software.

MFS 방식의 특성을 사용하여 이런 추가적인 연산을 줄였다.

제안한 MFS 방식으로 가변블록 움직임 추정을 지원하기 위해서는 우리는 각각의 블록 크기에 따라 MFS 방식을 개별적으로 수행해야만 했다. 이런 방식은 많은 연산량을 필요로 함으로 우리는 블록 크기에 따른 MFS 방식의 특성을 사용하여 이런 추가적인 연산을 줄였다. 표 2는 블록 크기에 따라 MFS로 선택된 참조영상이 다른 블록 크기의 최적 참조영상과 일치 할 확률을 보여준다. 표 2의 첫 번째 열은 MFS 방식에 사용된 블록 크기를 나타낸다. 그리고 두 번째 열은 선택된 참조영상을 비교할 최적 참조영상의 블록 크기를 나타낸다.

8x8 보다 작은 블록 (*SUB8x4*, *SUB4x8*, *SUB4x4*)은 본래 *P8x8* 블록과 같은 참조영상을 사용하므로 표2의 분석에서는 제외하였다. 예를 들어, 표 2의 첫 번째 행

표 2. 블록 크기에 따른 MFS로 선택된 참조영상과 최적 참조영상의 일치 확률

Table 2. Probability of matching the selected reference frame with respect to the block size.

MFS 블록 크기	비교대상 블록 크기	Claire	Fore-man	Con-tainer	Car-phone	Sign_Irene
8x8	16x8	42%	35%	49%	42%	34%
	8x16	37%	43%	36%	39%	31%
	16x16	41%	39%	36%	31%	42%
16x16	8x8	51%	58%	46%	42%	47%
	16x8	78%	82%	85%	79%	76%
	8x16	81%	83%	88%	84%	79%

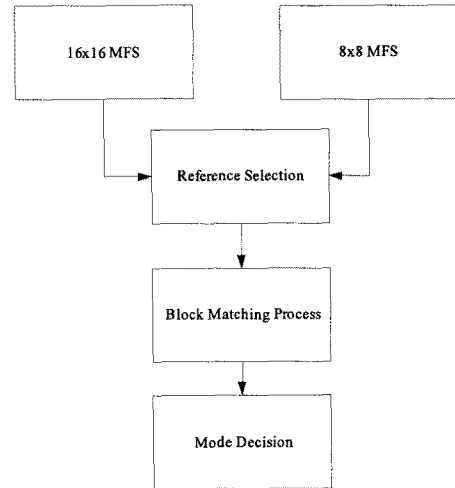


그림 6. 제안한 TPRFS 방법의 동작과정

Fig. 6. Operation flow of the proposed TPRFS scheme.

은 8x8 블록으로 제안한 MFS를 수행하였을 때 선택된 참조영상이 16x8, 8x16, 16x16 블록 크기의 최적 참조영상과 일치 할 확률을 보여준다. 표에서 나타나듯이 16x16 블록을 사용한 MFS 방식은 16x8과 8x16 블록에 대해 높은 상관성을 갖는 반면 8x8 블록에는 낮은 상관성을 보인다. 반대로 8x8 블록을 사용한 MFS 방식은 다른 16x8, 8x16, 16x16 블록과 낮은 상관성을 보인다. 따라서 우리는 블록 크기에 따른 참조영상의 상관도를 이용하여 TPRFS 방식을 제안하였다.

그림 6은 제안한 TPRFS 방식의 수행 순서를 나타낸다. TPRFS는 첫 번째 단계로 16x16 블록과 8x8 블록에 대해 각각 MFS를 수행한다. 그리고 다음으로 16x16 블록의 MFS로 선택된 참조영상을 8x16, 16x8 블록에 대해 같이 적용하고 8x8 블록의 MFS로 선택된 참조영상을 4x4, 4x8, 8x4 블록에 대해 적용한다. 마지막으로 최종적인 매크로 블록의 모드 결정을 수행한다.

IV. 성능 비교

우리는 제안한 세 가지의 참조영상 선택 방법을 JM12.2 참조 소프트웨어와 비교하였다. 실험에는 Akiyo, News, Claire, Foreman, Coastguard, Silent 등 6 종류의 영상을 사용하였다.

실험에 사용된 조건은 다음과 같다. 실험은 RD (Rate Distortion) 최적화를 수행하지 않는 저연산량 모드에서 수행하였다. 실험에 사용된 영상의 해상도는 Akiyo, News, Claire는 QCIF급 영상을 Foreman, Coastguard, Silent는 CIF급 영상을 사용하였다. 사용

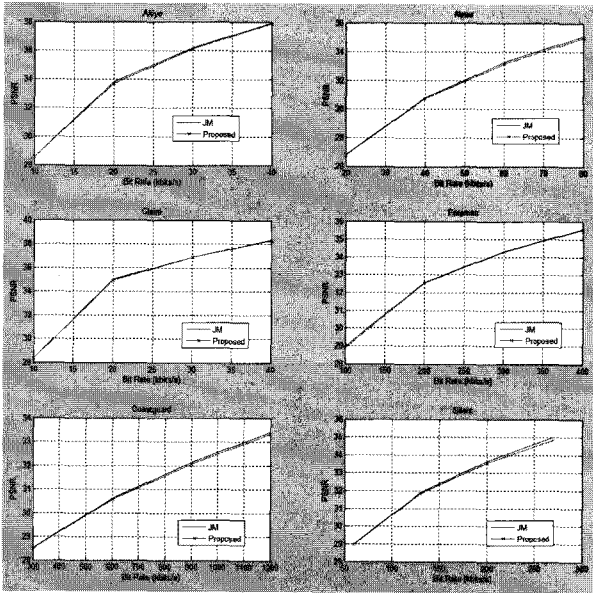


그림 7. 제안한 방법과 JM 12.2 참조 소프트웨어 사이의 RD-curve 비교

Fig. 7. RD-curve comparisons between JM 12.2 reference software and proposed schemes.

된 양자화 파라미터(Quantization Parameter: QP)의 값은 저 비트율 환경에서의 실험을 위해 28, 32, 36, 40을 사용하였다. 그리고 탐색 범위는 16으로 설정하였다. 그림 7은 위와 같은 조건으로 실험하여 얻은 결과를 바탕으로 그린 6종류 영상의 RD-curve이다. 시뮬레이션 결과, 같은 비트 레이트에서 제안한 방법의 평균 PSNR 감소는 단지 0.063 dB 정도였다. PSNR 성능비교를 영상의 움직임에 따라 나누어 보면 움직임이 많은 영상에서는 0.03에서 0.05 dB 정도로 그 차이가 크지 않은 반면 움직임이 적은 영상에서는 0.06에서 0.09 dB로 상대적으로 PSNR의 감소치가 보다 큰 값을 보인다.

이런 결과가 나타나는 이유는 제안한 방법 중 RSP 방법을 통해 참조 프레임 선택 연산 자체를 건너뛰는 경우가 움직임이 적은 영상에서 보다 많이 일어나기 때문으로 움직임 추정에 보다 적은 연산을 수행하면서 얻어지는 결과이다. 기존의 AFMFSA의 경우 전역탐색과 비교하여 PSNR 감소는 0.04에서 0.02 dB 정도로 제안한 방법과 유사한 결과를 보인다. 보다 정확한 성능비교를 위해 하나의 매크로 블록을 탐색하기 위해 필요한 참조영상의 수를 비교하였다. AFMFSA는 D1급 영상에 대해서도 실험을 진행한 반면 제안한 방법은 QCIF와 CIF급 영상에 대해 실험이 진행되어 공정한 비교를 위해 서로 같은 크기의 영상 4가지에 대한 비교 결과만을 제시하였다. 표 3은 전역탐색 방법과, AFMFSA, 그리고 제안한 방식의 하나의 매크로 블록을 처리하기 위해

표 3. 평균 탐색되는 참조영상 수

Table 3. Average searched reference frames.

영상종류	전역탐색	AFMFSA ^[10]	제안한 방법
Akiyo	5	1.29	1.79
News	5	1.63	1.80
Foreman	5	2.2	2.08
Coastguard	5	2.04	2.02

표 4. Early-termination을 위해 필요한 연산량

Table 4. Required computations for early-termination conditions.

동작	AFMFSA ^[10]	RSP
덧셈/뺄셈	46	10
쉬프트/곱셈	2	6
데이터 읽기	16	4

탐색해야 할 참조영상의 수를 나타낸다.

제안한 방법의 탐색 해야 할 참조영상의 수는 1.79에서 2.08로 이는 일반적인 전역탐색 움직임 추정 연산의 연산 량을 58.4~64.2% 감소시킬 수 있다는 것을 의미한다. Akiyo나 News 같은 몇몇 영상에서는 제안한 방법보다 AFMFSA 방식이 탐색해야 할 참조영상의 수에서 보다 좋은 결과를 보인다. 하지만 AFMFSA는 각각의 블록에서 MAD연산을 수행해야 한다. MAD연산은 블록의 모든 픽셀 데이터를 읽어 들여 이를 평균 한 후에 다시 각각의 픽셀과의 차이를 구하여 합해야 한다. 이 때 필요한 연산량을 4x4 블록을 기준으로 계산하여 표 4에 나타내었다.

반면 제안하는 RSP방법에서는 주변 블록의 MV값의 절대 합에 대한 비교 연산만이 필요하므로 움직임 추정 연산에 필요한 연산량의 측면에서 제안한 방식은 AFMFSA 보다 효율적이다.

V. 결론

본 고에서는 다중참조와 가변블록 움직임 추정을 효율적으로 지원하기 위한 세 가지 참조영상 선택 방법을 제안하였다. 제안된 RSP 방법은 기존 방식에 비해 참조영상의 선택 조건을 단순화 시켰으며, MFS 방식은 참조영상 선택 과정에서 영상의 움직임을 고려하여 화질의 열화 없이 참조영상 당 탐색횟수를 17% 감소 시켰다. 그리고 제안한 TPRFS 방법은 블록의 크기에 따른 MFS로 선택된 참조영상의 특성을 사용하여 효율적으로 가변블록 움직임 추정 연산을 지원한다. 제안한

참조영상 선택방법은 움직임 추정의 블록 정합 단계와 별개로 수행되기 때문에 기존의 어떤 단일참조 움직임 추정 알고리즘과도 같이 사용되어 다중참조 및 가변블록 움직임 추정 연산을 지원 할 수 있다.

참 고 문 헌

- [1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 (E) AVC). July 2004.
- [2] J. Ostermann, T. Wedi, et al., "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, pp.7-28, 2004.
- [3] Chi-Wang Ting, Wing-Hong Lam, and Lai-Man Po, "Fast block-matching motion estimation by recent-biased search for multiple reference frames," in *Proc. IEEE Int. Conf., On Image Processing*, Oct. 2004, pp. 1445-1448
- [4] Chen-Kuo Chiang and Shang-Hong Lai, "Fast multi-reference frame motion estimation via downhill simplex search," in *Proc. IEEE Int. conf., On Multimedia and Expo*, July 2006, pp.121-124
- [5] M.-J. Chen, Y.-Y. Chiang, H.-J. Li, and M.-C. Chi, "Efficient multi-frame motion estimation algorithms for MPEG-4/AVC/JVT/H.264," in *Proc. IEEE Int. Symp. on Circuits and Systems*, May 2004, pp.737-740
- [6] M. -J. Chen, G.-L. Li, Yi-Yen Chiang, and Ching-Ting Hsu, "Fast multiframe motion estimation algorithms by motion vector composition for the MPEG-4/AVC/H.264 standard," *IEEE Trans. Multimedia*, vol. 8, issue 3, pp.478-487, June 2006.
- [7] Chi-Wang Ting, Lai-Man Po, and Chun-Ho Cheung, "Center-biased frame selection algorithms for fast multi-frame motion estimation in H.264," in *Proc. IEEE Int. Conf. Neural Networks & Signal Processing*, Dec. 2003.
- [8] Huang, Y.-W., Hsieh, B.-Y., Chien, S, -Y., Ma, S.-Y., and Chen, L.-G, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC", *IEEE Trans, Circuits Syst., Video Technol.*, vol. 16, issue 4, pp.507-522, April 2006.
- [9] Saponara, S., Casula, M., Rovati, F., Alfonso, D., and Fanucci, L., "Dynamic control of motion estimation search parameters for low complex H.264 video coding," *IEEE Trans. Consum. Electron.*, vol. 52, issue 1, pp.232-239, Feb. 2006.
- [10] Liquan Shen, Zhi Liu, Zhaoyang Zhang, and Guozhong Wang, "An adaptive and fast multiframe selection algorithm for H.264 video coding," *IEEE Signal Processing Letters*, vol. 14, no. 11, pp.836-839, Nov. 2007.
- [11] Iain E. G. Richardson, *Video Codec Design: Developing Image and Video Compression Systems*, Wiley, 2002.
- [12] Choong Jin Hyun, and Myung H. Sunwoo, "Efficient memory reuse and sub-pixel interpolation algorithms for ME/MC of H.264/AVC," *IEEE Int. Workshop on signal Proc. Syst.*, Oct. 2006, pp.377-382
- [13] JM Reference Software version 12.2 (current version: 14.2)
<http://iphome.hhi.de/suehring/tml/download/>

저 자 소 개



김 성 대(정회원)

2003년 아주대학교 전자공학
학사 졸업.

2006년 아주대학교 전자공학
박사 수료.

<주관심분야 : 멀티미디어 코덱,
멀티미디어 신호처리, 멀티미디어
용 SOC 설계, 저전력 반도체 설
계>



선우명훈(정회원)

1980년 서강대학교 전자공학 학사
졸업.

1982년 한국과학기술원 전자공학
석사 졸업.

1982년~1985년 한국전자통신
연구소(ETRI) 연구원

1985년~1990년 Univ. of Texas at Austin
전자공학 박사.

1990년~1992년 Motorola, DSP Chip Division
(미국)

1992년~1996년 아주대학교 전기전자공학부
조교수.

1996년~2001년 아주대학교 전자공학부 부교수

2001년~현재 아주대학교 전자공학부 교수

<주관심분야 : VLSI 및 Parallel Architecture, 통
신 멀티미디어용 DSP 칩 및 ASIC 설계, 저전력
SOC 설계>