논문 2010-47SP-2-8

움직임 벡터와 참조 프레임간의 거리를 이용한 고속 다중 참조 프레임 움직임 추정

(Fast Multi-Reference Frame Motion Estimation Algorithm Using a Relation of Motion Vector with Distance of Each Reference Frame)

변 주 원*, 최 진 하*, 김 재 석**

(Juwon Byun, Jinha Choi, and Jaeseok Kim)

요 약

본 논문에서는 새로운 고속 다중 참조 프레임 움직임 추정 알고리즘을 제안한다. 제안된 알고리즘은 참조 프레임간의 거리와 움직임 벡터 간의 선형 관계를 이용하여 검색영역을 줄인다. 새로운 알고리즘은 0번 참조 프레임과 1번 참조 프레임에서는 전영역 움직임 추정을 사용한 후, 2번 참조 프레임, 3번 참조 프레임, 4번 참조 프레임에서는 0번 참조 프레임과 1번 참조프레임에서의 움직임 추정의 결과를 사용하여 검색 영역을 최소화 한다. 제안된 알고리즘은 문턱값을 사용하지 않기 때문에 하드웨어 구현과 프로세싱 스케줄을 지정하는 것이 용이하고, 많은 연산량을 줄일 수 있다. 이러한 몇 가지 특징으로 인하여 제안된 알고리즘은 다중 참조 프레임 움직임 추정기의 하드웨어 구현에 용이하다. 시뮬레이션 결과는 제안된 알고리즘의 PSNR 감소와 bitrate의 증가가 기존의 고속 다중 참조 프레임 움직임 추정 알고리즘에 비해서 좋은 성능을 나타내고 있고, 연산량은 기존의 5장의 참조 프레임을 사용하는 전영역 검색 방법에 비해서 52.5%의 감소를 나타낸다.

Abstract

This paper proposed a new fast multi-reference frame motion estimation algorithm. The proposed algorithm reduces search areas of motion estimation using a linear relation of motion vector with distance of each reference frame. New algorithm executes full search area motion estimation in reference frame 0 and reference frame 1. After that, search areas in reference frame 2, reference frame 3 and reference frame 4 are minimized by distance of each reference frame and results of motion estimation in reference frame 0 and reference frame 1. The proposed algorithm does not use a threshold value which is obstacle of hardware implementation and processing time schedule. Also, it reduced computation quantity of multi-reference motion estimation. Hardware implementation of multi-reference frame motion estimation is possible by these features. Simulation results show that PSNR drop and bitrate increase of proposed algorithm are lower than those of previous fast multi-reference frame motion estimation algorithm. The number of computation of new algorithm is reduced 52.5% and quality of result is negligible when compared with full search area motion estimation which has 5 reference frames.

Keywords : H.264/AVC, Motion estimation

Ⅰ. 서 론

H.264/AVC는 ITU-T와 ISO/IEC가 함께 표준화 작업을 진행시켜 얻어낸 새로운 비디오 압축 표준 방식으로 기존의 동영상 압축 방식과는 달리 높은 압축 효율과 다양한 전송환경에서도 에러에 강한 특성을 지닌 최신의 영상 압축 표준 방식이다.[1~2] H.264/AVC 영상 압축 표준 방식이 기존의 H.263+ 또는 MPEG-4 Simple Profile 표준에 비해서 약 50% 이상의 비트율을 감소할

* 학생회원, ** 정회원, 연세대학교 전기전자공학과
(Dept. of Electrical & Electronic Engineering, Yonsei University)
※ 이 논문은 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2009-8-1535)
접수일자: 2009년10월30일, 수정완료일: 2010년2월10일

수 있다. 이를 위해서 7가지 가변 블록에 대한 움직임 예측, 다수의 다중 프레임을 이용한 움직임 예측, 1/4화소 단위의 움직임 추정, 화면내 예측, 다양하고 효율적인 엔트로피 부호화, 정수기반 DCT 변환, 디블럭킹 필터 등의 기술을 적용하였다. 이러한 다양한 기술의 채택으로 인해서 기존 표준에 비해 압축 효율은 2배 이상 향상되었지만, 처리해야 할 연산량이 많아졌고 복잡도는 최대 16배 이상 증가했다. 특히 움직임 영상 부호화기에서 가장 많은 연산을 차지하는 움직임 예측에서 기존의 1/2단위 정확도의 움직임 추정을 1/4 화소 단위의 움직임 추정으로 확장하였고, 가변 블록의 종류를 좀 더 계층적이고 다양한 7가지로 확장하여 사용하였다. 또한, 기존에 한 장의 참조 영상만을 사용하던 것을 여러 장의 참조 영상을 사용하는 다중 참조영상 움직임 추정이 적용 되어 움직임 예측의 연산량이 기존의 표준인 H.263+ 나 MPEG-4 Simple Profile에 비해서 크게 증가하였고 많은 고속 알고리즘이 제안되었다.[3~5] 특히 다중 프레임 움직임 예측의 경우에는 프레임의 수에 비례해서 연산량이 증가하게 된다. 따라서 직전 프레임만을 사용하던 이전의 표준에 비해서 연산량이 비약적으로 증가하게 되었다. 이에 따라서 PVR(Personal Video Recoder)과 같이 실시간 HD급 영상 압축에 어려움이 발생하였다. 이러한 문제로 인해서 고속 다중 참조영상 움직임 예측 알고리즘의 필요성이 증대되었다. 본 논문에서는 움직임 벡터와 각 참조 영상간의 거리를 이용하여 검색영역을 축소시켜서 연산량이 기존의 다중 참조영상 움직임 예측보다 적고 성능이 우수한 고속 다중 참조영상 움직임 예측 방법을 제안한다.

본 논문은 다음과 같이 구성되어 있다. Ⅱ장에서는 기존의 고속 다중 참조 프레임 움직임 추정 알고리즘과 문제점을 분석한 후, 일반적인 다중 참조 프레임 움직임 추정의 특성을 분석하였고, 이를 바탕으로 새로운 고속 다중 참조 프레임 움직임 추정 알고리즘을 제안하였다. Ⅲ장에서는 기존의 고속 알고리즘과의 비교를 통한 성능을 평가하였고, Ⅳ장에서 결론을 맺었다.

Ⅱ. 본 론

1. 기존의 고속 다중 참조 프레임 움직임 추정 알고리즘

최신의 동영상 압축 표준인 H.264/AVC에서는 움직임 추정에 다중 참조 프레임 움직임 추정이 추가되면서 더욱더 많은 연산이 필요하게 되었다. 따라서 이를 극복하기 위한 여러 고속 알고리즘이 제안되었다.

가. Flexible Multi-frame Motion Estimation Algorithm with Adaptive Search Strategies (FMASS)

다중 참조 프레임 움직임 추정에서 참조 프레임 3과 참조 프레임 4를 결과 값으로 갖게 될 경우는 전체의 3% 이하이다. 따라서 참조 프레임 0과 참조 프레임 1, 참조 프레임 2 에서의 움직임 추정을 끝낸 후에 각 참조 프레임의 움직임 벡터와 SAD (Sum of Absolute Difference)값이 수식 (1)과 수식 (2) 중 하나를 만족할 경우에는 참조 프레임 3과 참조 프레임 4에 대한 움직임 추정을 수행하지 않는다.[6] FMASS의 경우에도 문턱값을 사용하므로 하드웨어 설계를 할 경우에 가변적인 타이밍을 갖게 된다.

$$|2\times MV_{ref0} - MV_{ref1}| < T_{MV}$$
$$\text{and} \quad |3\times MV_{ref0} - MV_{ref2}| < T_{MV} \tag{1}$$

$$SAD_{ref0} < SAD_{ref1}$$
$$\text{and} \quad SAD_{ref0} < SAD_{ref2} \tag{2}$$

나. Adaptive and Fast Multi-frame Selection Algorithm(AFMFSA)

AFMFSA 알고리즘은 다중 참조 프레임 움직임 추정에서 움직임 추정을 수행하는 프레임을 선택적으로 수행하는 고속 알고리즘이다. AFMFSA 알고리즘은 주변 블록과의 연관성을 이용한 방법과 참조 프레임 0과 참조 프레임 1의 움직임 벡터를 이용한 방법, 각 참조 프레임의 SAD 값을 이용하는 방법 등 3가지로 구성되어 있다. 주변블록과의 연관성을 이용한 방법은 우선 좌측, 상측, 좌상의 이미 부호화된 주변 블록의 참조 프레임이 모두 같은 프레임을 참조할 경우에 현재 블록의 움직임 추정은 주변 블록의 참조 프레임만을 수행한다. 그렇지 않을 경우에 우선 참조 프레임 0과 참조 프레임 1에서 움직임 추정을 수행한 후에 구해진 움직임 벡터를 통해 수식 (3)의 V_{ave}를 구한 후, 수식 (4)를 만족할 경우에는 참조 프레임 2, 참조 프레임 3, 참조 프레임 4의 움직임 추정은 생략한다. 수식 (4)에서 T_0는 큰 움직임에 대한 문턱값이고, T_1은 작은 움직임에 대한 문

턱값이다.

$$V_{ave} = \frac{|MV_{ref0}| + |MV_{ref1}|}{2} \quad (3)$$

$$V_{ave} > T_0 \text{ or } V_{ave} < T_1 \quad (4)$$

수식 (4)를 만족하지 않을 경우에는 참조 프레임 2를 수행하여 SAD값을 구한 후, 참조 프레임 0, 참조 프레임 1, 참조 프레임 2의 SAD값에 따라 그림 1과 같은 4가지의 경우로 나누게 된다. (a)의 경우에는 참조 프레임 2까지 움직임 추정을 수행하고 이후는 생략한다. (b)와 (c)의 경우에는 참조 프레임 3에 대해 움직임 추정을 수행하고 참조 프레임 2의 SAD값이 참조 프레임 3의 SAD값보다 클 경우에 참조 프레임 4에 대한 움직임 추정을 수행하고 만족하지 않을 경우에는 생략한다. (d)의 경우에는 참조 프레임 3과 참조 프레임 4에 대한 움직임 추정을 모두 수행한다. AFMFSA은 매 참조 프레임 마다 가변적으로 움직임 추정 여부를 결정한다. 이는 하드웨어 설계 시 타이밍을 가변적으로 만들며, 참조 프레임마다 데이터 의존성이 존재하기 때문에 파이프라인 구조나 병렬 구조의 구현이 어렵게 된다.[7]

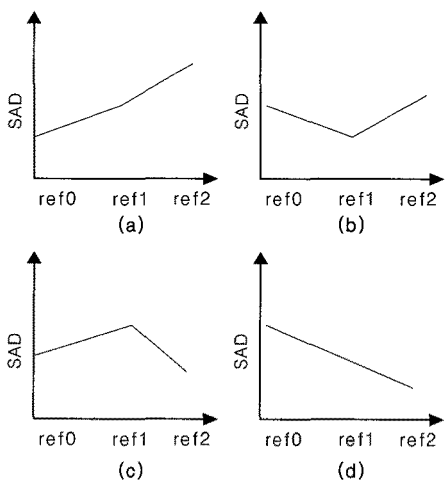

그림 1. AFMFSA에서의 SAD 값에 따른 종류
Fig. 1. Type of SAD value in AFMFSA.

다. 기존 알고리즘의 문제점

FMASS는 최소 3장, 최대 5장의 참조 프레임, AFMFSA는 최소 1장 최대 5장의 참조 프레임을 갖게 된다. 이러한 이유로 기존의 알고리즘은 최악의 경우에는 고속 알고리즘의 효과가 나타나지 않는다. 실시간 동영상 부호화기 설계를 위해서는 어느 경우에든 작동을 보장해야 한다. 따라서 최악의 경우인 참조 프레임이 5장인 경우를 기준으로 하드웨어 설계를 하게 된다. 때문에 기존의 고속 다중 참조 프레임 움직임 예측 알고리즘은 하드웨어 구현에 있어서 복잡도나 클럭 속도를 줄이는 역할을 하지 못한다. 또한 기존의 알고리즘은 각 참조 프레임의 예측 결과를 본 후에 다음 프레임의 움직임 예측 수행여부를 결정한다. 특히 AFMFSA 알고리즘의 경우에는 매 프레임 움직임 예측을 수행한 후에 결과를 검색하게 된다. 이는 매 프레임 결과의 데이터 의존성을 발생시키고 병렬 구조와 파이프라인 구조를 구현하는데 큰 어려움을 주게 된다. 또한 타이밍이 가변적으로 되어 하드웨어 구현에 어려움을 주게 된다. 이러한 문제로 인해서 기존의 고속 다중 참조 프레임 움직임 예측 알고리즘과 달리 고정적인 탐색영역과 타이밍을 갖고 데이터 의존도를 낮추어 하드웨어 구현에 유리한 고속 다중 참조 프레임 움직임 예측 알고리즘이 요구되었다.

2. 다중 참조 영상 움직임 예측의 특성

H.264/AVC 표준은 기존의 표준과는 다르게 다중 참조 프레임 움직임 추정과 더욱 세분화 된 가변 블록 크기 움직임 추정 등의 새로운 기술이 추가되었고, 이에 따라서 연산량이 증가하였다. 연산량을 줄이기 위해서 움직임 추정의 특성을 분석하였다.

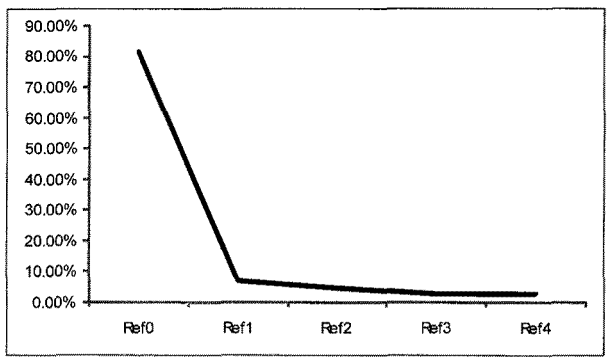

그림 2. 각 참조 프레임의 선택 확률
Fig. 2. Probability of selecting reference frame.

가. 참조 프레임별 선택 확률

일반적으로 현재 압축하는 프레임과 시간적으로 가장 가까운 참조 프레임이 0번 참조 프레임이 되고, 현재 프레임과 시간상으로 멀리 존재할수록 참조 프레임의 번호는 증가하게 된다. H.264/AVC 부호화기에서는 일반적으로 5장의 참조 프레임을 사용하기 때문에 현재

프레임과 시간상으로 멀게 존재하는 참조 프레임은 4번 참조 프레임이 된다. QCIF(176×144) 사이즈의 akiyo, foreman, carphone, container, mobile 영상의 경우 각 참조 프레임이 선택될 확률은 그림 2와 같이 나타난다. 참조 프레임 0과 참조 프레임 1을 선택할 확률이 88% 이상으로 대부분을 차지함을 볼 수 있다. 또한 참조 프레임 3과 참조 프레임 4를 선택할 확률이 각각 3% 미만임을 확인할 수 있다. 따라서 대부분의 경우 참조 프레임 0과 참조 프레임 1만을 움직임 추정해도 크게 성능저하가 없을 것을 생각할 수 있다.

나. 움직임 벡터와 시간축의 관계

부호화 하려는 매크로 블록이 일정한 속도의 움직임을 갖고 있다고 가정하면, 그림 3과 같이 각 참조 프레임의 움직임 추정 결과는 현재 프레임과 각 참조 프레임의 시간상의 거리로부터 추정할 수 있다.

시간상의 거리는 각 참조 프레임의 POC(Picture Order Count)를 통해서 계산할 수 있다. 또한, 앞서 살펴본 것과 같이 대부분의 경우 가장 인접한 참조 프레임인 0번 참조 프레임이 선택될 확률이 높다. 표 1은 QCIF사이즈의 akiyo, foreman, carphone container, mobile 영상을 이용해, 0번 참조 프레임의 움직임 벡터를 현재 프레임과 각 참조 프레임 사이의 시간상의 거리를 이용해 비례하여 참조 프레임 1, 참조 프레임 2, 참조 프레임 3, 참조 프레임 4에 해당하는 확장된 움직임 벡터를 구했을 때, 확장된 움직임 벡터의 주변에 각 참조 프레임의 움직임 추정 결과 움직임 벡터가 각각 2×2, 4×4, 6×6, 8×8 영역 안에 존재할 확률을 나타낸 것이다. 표 1의 결과를 살펴보면 akiyo나 container와 같이 움직임이 적거나 일정한 속도의 움직임이 많은 영상의 샘플 영상에서는 참조 프레임 0의 움직임 벡터로부터 확장된 움직임 벡터가 각 참조 프레임의 움직임 벡터와 2×2 블록 안에 존재할 확률이 90% 이상의 높은 확률을 보인다. 그에 반해서 일정한 속도의 움직임이 적은 foreman, carphone과 같은 영상에서는 매우 작은 40% 이하의 확률을 보였다. 하지만, 이 블록의 크기를 키울수록 확률이 증가하였고 최종적으로 8×8 블록 안에 존재할 경우는 약 90% 정도의 높은 확률을 보였다.

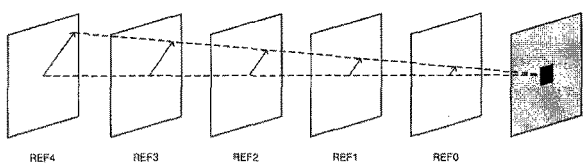

그림 3. 각 참조 프레임과 움직임 벡터의 관계
Fig. 3. Relation between motion vector and each reference frame.

표 1. 참조 프레임 0의 움직임 벡터에 의해 추정된 움직임 벡터의 정확도
Table 1. Accuracy of estimated MV by MV in reference frame 0.

	2×2	4×4	6×6	8×8
akiyo	92.4%	97.6%	98.9%	99.3%
foreman	30.7%	59.1%	73.9%	82.8%
carphone	38.8%	61.2%	73.4%	81.2%
container	94.0%	97.6%	98.6%	99.0%
mobile	74.3%	84.2%	87.1%	88.9%
Average	66.0%	79.9%	86.4%	90.2%

약 95% 이내의 오차율을 갖기 위해서 1번 참조 프레임의 움직임 벡터도 포함시켜, 각 참조 프레임의 움직임 벡터가 0번 참조 프레임과 1번 참조 프레임의 움직임 벡터 주변의 일정한 블록 안에 있을 확률은 다음 표 2와 같다. 참조 프레임 0과 참조 프레임 1의 움직임 벡터를 확장한 벡터와 참조 프레임 2, 참조 프레임 3, 참조 프레임 4 의 움직임 추정 결과 움직임 벡터가 8×8

표 2. 확장된 참조 프레임 0과 참조 프레임 1의 움직임 벡터의 정확도
Table 2. Accuracy of estimated MV by MV in reference frame 0 and reference frame 1.

	2×2	4×4	6×6	8×8
akiyo	98.0 %	99.4 %	99.7 %	99.8 %
foreman	59.3 %	83.8 %	90.9 %	94.0 %
carphone	62.8 %	81.7 %	88.1 %	91.7 %
container	98.3 %	99.3 %	99.5 %	99.7 %
mobile	85.3 %	89.3 %	90.8 %	91.8 %
Average	80.7 %	90.7 %	93.8 %	95.4 %

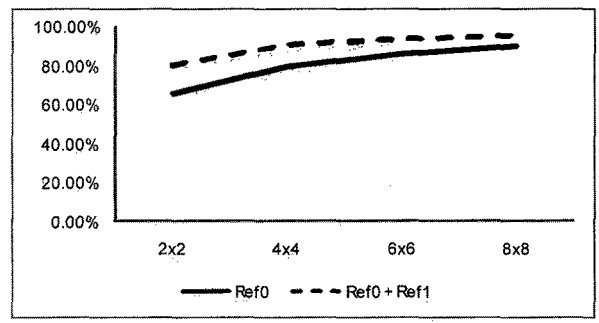

그림 4. 확장된 움직임 벡터의 적중 확률
Fig. 4. Graph of probability of estimated MV accuracy.

내에 존재할 확률이 평균적으로 95.4%가 됨을 확인할 수 있다. 그림 4는 참조 프레임 0의 움직임 벡터만을 확장하여 실험한 결과와 참조 프레임 0과 참조 프레임 1의 움직임 벡터를 모두 사용하여 구했을 경우의 움직임 벡터의 정확도를 나타냈다. 전체적으로 참조 프레임 0과 참조 프레임 1을 모두 사용하여 확장 하였을 때의 경우가 참조 프레임 0 만을 사용하여 확장 하였을 때보다 적중률이 높았음을 확인할 수 있다.

3. 제안하는 고속 다중 참조 프레임 알고리즘

제안된 알고리즘은 각 참조 프레임에 따라서 움직임 추정을 수행하는 영역을 다르게 하는 것을 기본으로 한다. 앞서 살펴 본 것과 같이 약 90%의 경우에 0번 참조 프레임과 1번 참조 프레임을 사용하게 된다. 또한 약 95%의 경우에 0번 참조 프레임의 움직임 벡터와 1번 참조 프레임의 움직임 벡터를 시간축의 비례관계로부터 확장시킨 확장된 움직임 벡터 주변 8×8 크기 안에 참조 프레임 1, 참조 프레임 2, 참조 프레임 3의 움직임 벡터가 존재한다. 이러한 실험 결과에 따라 제안하는 다중 참조 프레임 움직임 추정 알고리즘의 순서는 그림 5와 같다.

우선 참조 프레임 0과 참조 프레임 1에 대해서 전영역 검색인 32×32의 크기로 움직임 추정을 수행하여 움직임 벡터 MV_1과 MV_2를 구한다. 구한 움직임 벡터 MV_1과 MV_2와 각 참조 프레임의 POC와 현재 프레임의 POC를 이용하여 수식 (5)를 이용하여 참조 프레임 2, 참조 프레임 3, 참조 프레임 4의 검색 영역의 중심점 $CP_{n,i}$를 구한다.

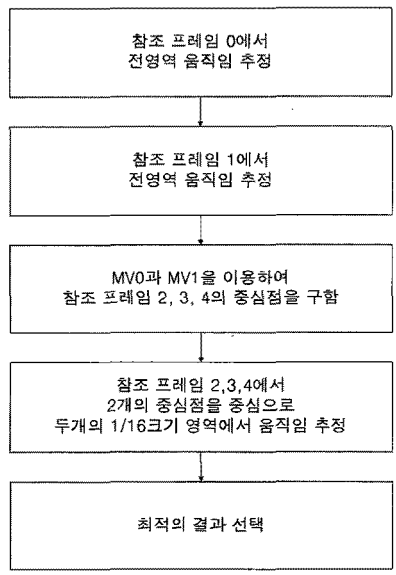

그림 5. 제안한 알고리즘의 순서
Fig. 5. Flow chart of proposed algorithm.

$$CP_{n,i}(x,y) = MV_i \times \frac{POC_n - POC_{curr}}{POC_i - POC_{curr}} \quad (5)$$

where i=0 or 1, n = 2 or 3 or 4

이후에 앞에서 구한 검색영역의 중심점을 바탕으로 움직임 추정 영역을 1/16으로 줄여서 움직임 추정을 수행하게 된다. 참조 프레임 2, 참조 프레임 3, 참조 프레임 4 각각 참조 프레임 0의 움직임 벡터에 대한 중심점과 참조 프레임 1의 움직임 벡터에 대한 중심점 2개의 검색영역의 중심점이 구해진다. 구한 검색영역의 중심점을 중심으로 각각 전영역 검색의 1/16로 검색영역을 줄여서 움직임 추정을 수행하게 된다. 이후에 모든 결과를 비교해서 최적의 결과를 선택하게 된다. 그림 6은

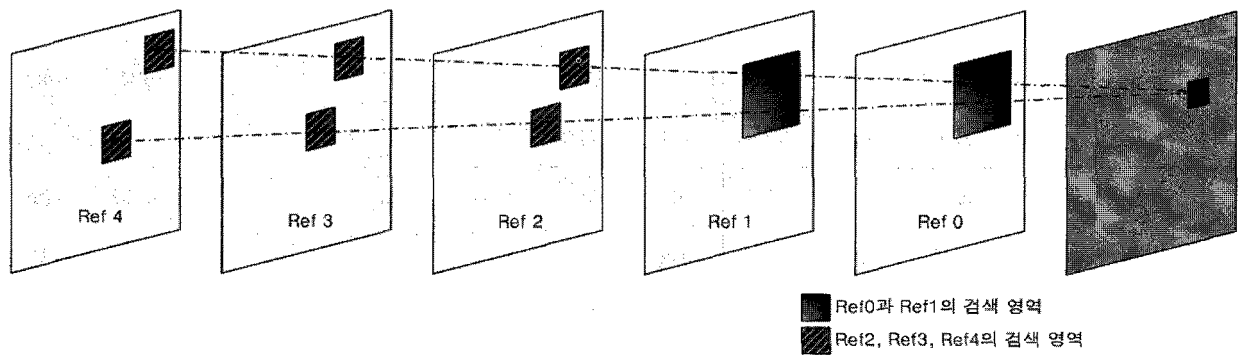

그림 6. 제안한 알고리즘의 검색 영역
Fig. 6. Search range of proposed algorithm.

제안된 알고리즘의 각 참조 프레임에서의 검색 영역을 나타낸 그림이다. 제안된 알고리즘은 문턱값을 사용하지 않음으로써 프로세싱 스케쥴을 정하는 데 유리하며, 최악의 경우에도 변화없는 연산량으로 인하여 실제적으로 연산기의 수를 줄일 수 있는 장점이 있다. 이러한 특징은 제안된 고속 다중 참조 프레임 움직임 추정 알고리즘이 기존의 고속 다중 참조 프레임 움직임 추정 알고리즘에 비하여 하드웨어 구현에 보다 적합한 알고리즘임을 보여준다.

Ⅲ. 실 험

제안된 알고리즘은 표준의 5장의 다중 참조 프레임을 사용하는 움직임 추정을 2.375장의 참조 프레임을 사용하도록 변화시킨 알고리즘이다. 움직임 추정에 대한 연산량은 참조 프레임의 수에 비례한다. 따라서 연산량은 참조 프레임의 수로 비교하기로 하였다. 제안된 알고리즘은 참조 프레임을 5장 사용하는 표준에 비해서 52.5%의 연산량이 감소하게 된다. 제안된 알고리즘의

표 3. 제안된 알고리즘과 기존의 알고리즘의 비교
Table 3. Comparison of proposed algorithm with previous fast algorithms.

		FMASS				AFMFSA				Proposed			
		평균 참조 프레임의 수		△PSNR	△Bit	평균 참조 프레임의 수		△PSNR	△Bit	평균 참조 프레임의 수		△PSNR	△Bit
QCIF	akiyo	3.011	-39.7%	-0.046	0.29%	1.287	-74.2%	-0.087	4.73%	2.375	-52.5%	0.000	0.00%
	carphone	3.232	-35.3%	-0.074	2.02%	2.676	-46.4%	-0.100	4.15%			-0.009	0.36%
	coastguard	3.092	-38.1%	-0.010	0.61%	2.258	-54.8%	-0.019	1.33%			0.011	-0.36%
	container	3.017	-39.6%	-0.006	3.81%	1.820	-63.6%	-0.054	13.58%			0.004	0.38%
	foreman	3.214	-35.7%	-0.014	1.14%	2.994	-40.1%	-0.018	1.77%			-0.002	-0.40%
	mobile	3.113	-37.7%	-0.073	5.09%	2.532	-49.3%	-0.120	8.03%			0.007	0.20%
	mother-daughter	3.040	-39.2%	-0.060	0.68%	1.360	-72.7%	-0.104	1.58%			-0.007	-0.77%
CIF	coastguard	3.189	-36.2%	-0.014	0.42%	2.136	-57.2%	-0.028	1.18%			-0.004	0.00%
	container	3.037	-39.2%	-0.038	4.96%	1.870	-62.6%	-0.045	6.91%			0.000	0.03%
	foreman	3.283	-34.3%	-0.054	0.98%	2.306	-53.8%	-0.047	1.28%			0.000	0.03%
	mobile	3.229	-35.4%	-0.072	4.51%	2.624	-47.5%	-0.118	7.36%			-0.006	-0.03%
	mother-daughter	3.059	-38.8%	-0.022	1.16%	1.219	-75.6%	-0.101	1.90%			-0.006	0.37%
	news	3.069	-38.6%	-0.028	0.16%	1.208	-75.8%	-0.054	1.75%			0.011	0.04%
	silent	3.109	-37.8%	-0.022	0.43%	1.258	-74.8%	-0.045	2.54%			-0.014	0.33%
4CIF	city	3.336	-33.2%	-0.017	0.74%	1.265	-74.7%	-0.088	5.52%			-0.004	0.75%
	crew	3.593	-28.1%	-0.008	0.28%	1.844	-63.1%	-0.034	1.76%			-0.002	0.59%
	harbour	3.460	-30.7%	-0.009	0.37%	1.767	-64.6%	-0.057	2.62%			-0.003	0.24%
	ice	3.254	-34.9%	-0.015	0.24%	1.284	-74.3%	-0.054	2.17%			-0.007	0.60%
Total Average		3.185	-36.2%	-0.032	1.55%	1.873	-62.5%	-0.065	3.90%	2.375	-52.5%	-0.002	0.13%

성능을 분석하기위하여 표준 Reference code인 JM 9.6을 이용하여 제안된 알고리즘의 시뮬레이터를 제작하였다. 또한 비교대상인 5장의 참조 프레임을 갖는 표준 코드도 JM 9.6을 사용하였다. 화질의 객관적 수치인 PSNR 비교를 위한 시뮬레이션에 사용한 영상은 QCIF 사이즈(176×144)의 akiyo, carphone, coastguard, container, foreman, mobile, mother-daughter 영상과 CIF 사이즈(352×288)의 coastguard, container, foreman, mobile, mother-daughter, news, silent, 4CIF 사이즈(704×576)의 city, crew, harbour, ice 영상을 이용하였다. 인코딩한 영상 프레임 수는 모두 100 Frame을 인코딩 하였으며, 인코딩에 사용한 프로파일은 Baseline Profile을 사용하였다. 또한 엔트로피 코딩은 CAVLC를 이용하였다. 기본 Search Range는 32×32로 설정하여 움직임 추정을 수행하였다. 기본 Search Range가 32×32이기 때문에 1/16으로 줄어든 Search Range는 8×8이 된다. 표 3은 QP가 28일 경우에 제안된 알고리즘과 FMASS, AFMFSA를 5장의 참조 프레임을 사용하는 표준 방식과 비교한 결과이다. 표 3에 있는 △PSNR, △Bit는 각각 PSNR과 Bitrate의 표준 참조 프레임 5장의 다중 참조 프레임 움직임 추정 알고리즘과의 차이다. FMASS 알고리즘의 경우 최대 PSNR Drop이 0.074dB이고 최대 Bit 증가량은 5.09% 이다. 또한 연산량이 최대 39.7% 감소하지만 최악의 경우 28.1% 감소한다. AFMFSA 알고리즘의 경우에는 최대 PSNR Drop이 0.12dB 이고 최대 Bit 증가량은 13.58%이다. 특히 QCIF에 container 샘플 영상에 취약한 성능을 보임을 알 수 있다. 연산량에 있어서는 최대 75.8%의 감소가 있었고, 최악의 경우 40.1%의 감소가 있었다. AFMFSA영상의 경우는 영상에 따라 성능차이가심하며 참조 프레임의 수도 크게 변함을 알 수 있다. 이에 반해서 제안된 알고리즘은 거의 모든 샘플 영상에 일정한 1%이하의 bit 증가율을 보여주었으며, PSNR의 Drop도 최대 0.014dB로 다른 두 알고리즘에 비해서 좋은 성능을 보여주었다. 또한, 언제나 같은 연산량을 지니고 있음을 볼 수 있다.

PSNR과 Bitrate 증가율을 하나의 수치로 보이기 위하여 BDPSNR(Bjonteggard Delta PSNR)[8]을 이용하였다. 표 4는 QP 26, 28 30, 32, 34의 결과를 BDPSNR을 이용하여 나타낸 결과이다. 표 4의 결과에 따르면 QP 값이 변하여도 언제나 제안한 알고리즘의 성능이 뛰어남을 확인할 수 있다. 특히 제안된 알고리즘은 FMASS

표 4. BDPSNR 비교
Table 4. Comparison of BDPSNR.

QP	Standard	FMASS	AFMFSA	Proposed
26	37.738	37.657	37.516	37.725
28	36.423	36.336	36.189	36.407
30	35.043	34.955	34.798	35.021
32	33.718	33.629	33.474	33.694
34	32.479	32.414	32.251	32.462

보다 평균적으로 약 0.8장의 참조 프레임을 덜 사용하였지만, 성능은 FMASS보다 0.05dB 이상 높게 나타났고, AFMFSA에 비해서는 평균적으로 약 0.5장의 참조 프레임을 더 사용하였지만, 0.21dB 이상 높은 성능을 보여주었다.

Ⅳ. 결 론

본 논문에서는 고속 다중 참조 프레임 영역 선택 움직임 추정에 대해 연구하였다. 다중 참조 프레임 영역 선택 알고리즘을 제안하고 알고리즘의 성능을 비교 실험하였다. 제안된 알고리즘은 기존의 FMASS, AFMFSA와 같은 다중 참조 프레임 움직임 추정의 고속 알고리즘이다. 기존의 고속 다중 프레임 움직임 추정 알고리즘이 문턱값을 이용하여 타이밍이 가변적인 문제가 있지만, 제안된 알고리즘은 타이밍이 고정되어 있다. 최악의 경우에는 전영역 검색을 하는 기존의 고속 알고리즘은 하드웨어 구현에 있어서 하드웨어 크기와 클럭 스피드에 대해서 효과를 줄 수 없지만, 제안된 고속 알고리즘은 하드웨어 구현 시 크기를 줄이거나 클럭 속도를 낮출 수 있다. 제안된 알고리즘은 영상안의 물체가 선형적인 움직임을 보이는 대부분의 일반적인 영상에서는 기존의 FMASS, AFMFSA 알고리즘에 비해서 성능저하가 작고 평균적인 연산량이 작다는 장점이 있다. 마지막으로 알고리즘 내부적으로 2번 참조 프레임, 3번 참조 프레임, 4번 참조 프레임에서는 데이터 의존성이 없기 때문에 이들에 대한 연산은 파이프라인 구조나 병렬 구조가 가능하다. 제안된 알고리즘은 BDPSNR 기준으로 FMASS보다 평균적으로 약 0.8장의 참조 프레임을 덜 사용하였지만, 성능은 FMASS보다 0.05dB 이상 높게 나타났고, AFMFSA에 비해서는 평균적으로 약 0.5장의 참조 프레임을 더 사용하였지만, 0.21dB 이상 높은 성능을 보여주었다.

참 고 문 헌

[1] Iain E.G. Richardson 저, "H.264 and MPEG - 4 Video Compression, Video Coding for Next - generation Multimedia", WILEY 출판사

[2] Jorn Ostermann, Jan Bormans, Peter List, Detlev Marpe, Matthlas Narroshke, Fernando Pereira, Thomas Stockhammer and Thomas Wedi, "Video Coding with H.264/AVC : Tools, Performance and Complexity", IEEE Circuits and Systems Magazine, First Quarter 2004.

[3] 최진하, 이원재, 김재석, "H.264/AVC의 효율적인 전 영역 움직임 추정을 위한 새로운 움직임 벡터 예측 방법 제안", 대한전자공학회 논문지, 제 44권 SP편 제 3호, pp. 70-79, 2007년 5월.

[4] 최웅일, 전병우, "H.264 표준의 가변 움직임 블록을 위한 고속 움직임 탐색 기법", 대한전자공학회 논문지, 제 41권 SP편 6호, 209-220 쪽, 2004년 11월.

[5] 이융기, 이영렬, "가변 크기 블록에서 정수단위 화소 움직임 벡터의 빠른 검색", 대한전자공학회 논문지, 제 40권 SP편 5호, 388-396 쪽, 2003년 9월.

[6] Xiang Li Li, E.Q. Yen-Kuang Chen, "Fast multi-frame motion estimation algorithm with adaptive search strategies in H.264", ICASSP'04, 17-21, May. 2004.

[7] Liquan Shen, Zhi Liu, Zhaoyang Zhang, Xuli Shi, "An adaptive and fast H.264 multi frame selection algorithm based on information from previous searches", IEEE SIGNAL PROCESSING LETTERS, Vol. 14, No. 11, Nov. 2007.

[8] Bjontegaard, G.: Recommended Simulation Condition for H.26L, ITU-T Q6/SG16, Doc. #VCEG-L38, 9-12 Jan, 2001.

저 자 소 개

변 주 원(학생회원)
2007년 연세대학교 전기전자 공학부 학사 졸업.
2009년 연세대학교 전기전자 공학과 석사 졸업.
2009년~현재 연세대학교 전기 전자공학과 박사 과정
<주관심분야 : H.264/AVC, ISP, SoC 설계>

김 재 석(정회원)
1977년 연세대학교 전자공학과 학사 졸업.
1979년 한국과학원 전기 및 전자 공학과 석사 졸업.
1988년 Rensselaer Polytechnic Institute 전자공학과 박사 졸업.
1993~1995년 한국 전자통신 연구원 책임 연구원.
1995~현재 연세대학교 전기전자공학과 교수.
<주관심분야: 통신 및 영상 시스템, VLSI 신호처리, 임베디드 S/W 및 SoC 구현>

최 진 하(학생회원)
2005년 연세대학교 전기전자 공학부 학사 졸업.
2007년 연세대학교 전기전자 공학과 석사 졸업.
2007년~현재 연세대학교 전기 전자공학과 박사 과정
<주관심분야 : H.264/AVC, ISP, SoC 설계>