

경성 실시간 동작을 보장하는 움직임 추정 알고리즘

Motion Estimation Algorithm to Guarantee Hard Realtime Operation

양 현 철*, 이 성 수**

Hyeon-Cheol Yang*, Seongsoo Lee**

Abstract

This paper proposes a motion estimation algorithm with run-time adaptive workload control. It has optimized performance within limited hardware resources while guaranteeing hard realtime operation. It performs maximum searches within hard realtime constraints, since it determines search steps and workload adaptively. It reduces the hardware size to 1/4~1/400 of conventional algorithms, while its PSNR degradation is only 0.02~0.44 dB. It can be easily applied to most conventional fast algorithms, so it is useful to design realtime encoder chips.

요 약

본 논문에서는 움직임 추정기의 하드웨어 자원이 유한하게 주어졌을 때, 동작 중에 적응적으로 작업량을 제어함으로써 주어진 경성 실시간 동작 조건 내에서 최적의 성능을 얻는 움직임 추정 기법을 제안한다. 제안하는 움직임 추정 기법은 작업량과 탐색 단계를 적응적으로 결정함으로써 경성 실시간 동작을 보장하는 범위 내에서 최대한의 탐색을 수행할 수 있다. 이 기법은 하드웨어 크기를 기존 기법의 1/4~1/400까지 줄이면서도 PSNR 저하는 0.02~0.44 dB에 불과하며, 하드웨어의 사용 효율도 기존 기법의 3.7~21.5배에 달하였다. 이 기법은 기존의 고속 탐색 기법에 쉽게 적용이 가능하므로 실시간 처리가 가능한 인코더 칩을 설계하는데 유용하다.

Key words : motion estimation, hard realtime, low computation, n-step search, adaptive search order determination

1. 서론

동영상 저장 및 전송에 필요한 막대한 데이터량을 줄이기 위해 MPEG-4, H.264/AVC 등의 동영상 압축

* School of Electronic Engineering, Soongsil University, sslee@ssu.ac.kr, 010-9182-3835

★ Corresponding author

※ Acknowledgment

“This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2010-0025041)”
Manuscript received Mar. 4, 2013; revised Mar. 20, 2013; accepted Mar. 21, 2013

규약이 국제적으로 널리 쓰이고 있다. 이러한 동영상 압축 과정은 기본적으로 많은 연산량이 요구되기 때문에 대부분 하드웨어 칩으로 구현되는데[1][2], 특히 움직임 추정 (motion estimation: ME)[3]의 경우 전체 동영상 압축 과정에서 가장 높은 연산량을 가지기 때문에 하드웨어 칩 구현에서 가장 어려운 부분에 속한다. 이에 따라 움직임 추정의 연산량을 줄이기 위한 연구가 다양하게 연구되어 왔다.

조기 중단 기법 (early termination: ET)[4]은 SAD 계산 중간에 지금까지 계산된 중간값이 지금까지 찾아진 최소 SAD를 넘을 경우 더 이상의 SAD 계산을 중단하고 다음 탐색으로 넘어가는 기법으로서 성능 저하 없이 연산량을 줄일 수 있어서 많은 고속 움직임 추정 기법에 널리 사용되고 있다. 그러나 이 기법

의 경우 평균 연산량은 크게 줄지만 최대 연산량은 변화가 없기 때문에 실시간 동작을 수행해야 하는 영상 통화나 영상 감시 등에 적용하는 경우 칩 구현시의 크기를 줄이는 측면에서는 큰 효과를 기대하기 어렵다.

본 논문에서는 효과적인 조기 중단이 일어날 수 있도록 하는 적응적 탐색 순서 결정 기법 (adaptive search order determination: ASOD)[5]을 개량하여 실시간 동작을 보장하는 움직임 추정 기법인 (hard-realtime adaptive search order determination: HRASOD)을 제안하고자 한다. 제안하는 기법은 조기 중단 기법을 사용하여 연산량을 감소시키면서도 각종 실시간 어플리케이션에 적용이 가능하고, 실시간 동작이 보장되어야 하는 SoC 구현에도 적합하다.

II. 적응적 탐색 순서 결정 기법

1. 기존 고속 움직임 추정 기법

최초로 제안된 움직임 추정 기법은 전역 탐색 기법 (full search: FS)[3]이라 불리며, 탐색 영역 내의 모든 지점에 대해 탐색을 수행하므로 성능이 가장 우수한 반면에 연산량도 가장 많다. 이 연산량을 줄인 고속 움직임 추정 기법에서는 탐색 영역의 전체 지점에 대해서 연산하는 것이 아닌 특정 기준의 지점, 즉, 탐색 후보의 개수를 줄임으로서 연산해야 하는 지점을 줄여 기존 전역 탐색 방식의 높은 연산량을 감소시키는 방법이 일반적으로 사용되고 있다. 3단계 탐색 기법 (three-step search: TSS)[6] 및 이를 확장한 N단계 탐색 기법 (N-step search: nSS)[7][8], 계층적 탐색 기법 (hierarchy search)[9] 등 많은 기법이 제안되었으며, 최근에는 H.264/AVC 등의 동영상 압축 규약에 최적화되고 성능이 우수한 비대칭 다중 육각 탐색 (unsymmetrical multi-hexagonal-grid search: UMHEX)[10], 강화된 예측 영역 탐색 방식 (enhanced predictive zonal search: EPZS)[11] 등도 널리 사용되고 있다.

2. N단계 탐색 기법

N단계 탐색은 원점을 중심으로 일정 거리만큼의 8개의 지점에 대해서 유사도를 측정하고 가장 큰 유사도를 가지는 지점을 기준으로 연산을 반복하여서 최종 움직임 벡터를 구하는 방식을 말한다. 일반적으로 3단계 탐색 방법[6]이 주로 쓰였으며, 탐색 영역의 크

기에 따라 4단계[8] 및 그 이상으로 확장할 수 있다.

N단계 탐색 방법은 단계가 결정되면 탐색 영역의 크기에 관계없이 각 단계에 따라 탐색 지점의 개수가 결정되므로 어느 조건에서도 동일 연산량을 가져 하드웨어 구현이 용이하다는 장점을 가진다.

3. 조기 중단 기법

움직임 추정은 최소 SAD 지점을 찾는 연산을 의미하는 것이므로 이를 이용하여 기존에 구하여진 최소 SAD의 값보다 현재 연산중인 SAD의 중간 연산값이 더 커질 경우 더 이상 연산을 수행하지 않고 다음 탐색 위치에 대해서 연산을 수행하는 방법이 조기 중단 기법[4]이다. 조기 중단 기법을 사용할 경우 필요없는 잉여 연산을 수행하지 않으므로 연산량을 크게 줄일 수 있다.

조기 중단이 이루어지는 경우에는 원래 수행해야 할 연산의 일부를 중단하기 때문에 평균 연산량은 크게 감소하나, 조기 중단이 이루어지지 않는 경우도 존재하기 때문에 최대 연산량은 줄어들지 않는다. 따라서 실시간 수행이 필요한 경우에는 조기 중단 기법을 적용하더라도 하드웨어 설계 시에 최대 연산량을 기준으로 하드웨어 필요량을 산정할 수밖에 없다는 단점이 있다.

4. 적응적 탐색 순서 결정 기법

일반적인 3단계 탐색 기법의 경우, 그림 1과 같이 1단계는 9개, 2단계 이후는 8개의 탐색 위치를 고정적인 래스터 순서(raster order), 즉 좌상, 중상, 우상, 좌중, 우중, 좌하, 중하, 우하 순서로 탐색한다.

적응적 탐색 순서 결정 기법[5]은 기본적으로 탐색 영역 내에 일정한 영역을 나누고 조기 중단이 효과적으로 일어날 수 있는, 즉 SAD의 값이 가장 작게 나올 수 있는 영역부터 연산을 수행하는 방법이다. 크게 연산을 수행할 영역을 나누는 방법과 해당 영역들의 연산순서를 결정하는 방법의 두 가지 부분에 대한 알고리즘이다. 그러므로 다양한 움직임 추정 알고리즘에 적용이 가능하다.

일반적으로 현재 블록 움직임 벡터의 예측값은 확률적으로 SAD가 최소가 될 확률이 가장 높으며, 이 예측값에 가까운 탐색 위치일수록 SAD가 최소가 될 가능성이 높으므로 탐색해야 할 탐색 위치를 현재 블록 움직임 벡터의 예측값에 가까운 순서대로 탐색하는 방법을 사용하며 이 방법에 의해서 좀 더 빠른 연산이 가능하다.

적응적 탐색 순서 결정 기법을 간단한 고속 탐색

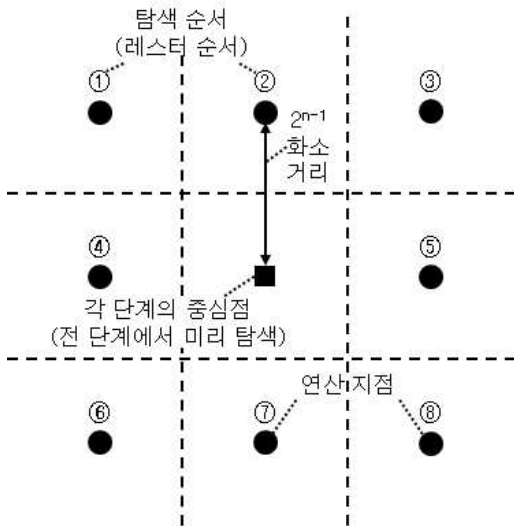


Fig. 1. Search order in conventional N-step search
 그림 1. 일반적인 N단계 탐색방법에서의 탐색 순서

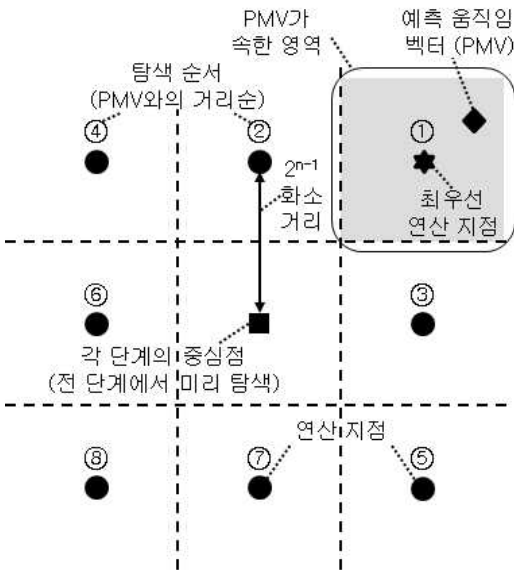


Fig. 2. Search order in N-step search using adaptive search order determination

그림 2. 적응적 탐색순서 결정 기법을 사용한 N단계 탐색방법에서의 탐색 순서

방법중 하나인 N단계 탐색 기법에 적용할 경우 그림 2와 같은 형태를 취한다.

이 경우에는 N단계 탐색 방법의 9개의 탐색 지점을 선택하기 위해서 9개의 영역으로 탐색영역을 구분

하였지만 탐색방법에 따라 영역을 나누는 방법은 다를 수 있다. 다른 방법의 경우도 마찬가지로 영역중 예측 움직임 벡터가 존재하는 지점을 최우선 연산지점으로 지정하고 최우선 연산지점을 기준으로 앞서 제시하였던 5개의 단계로 미리 정의된 연산순서를 소프트웨어의 경우에는 배열로 지정해서 저장하고 하드웨어에서는 룩업 테이블 또는 유한상태머신(finite state machine: FSM)의 형태로 정의할 수 있어서 최우선 연산 지점을 검출하는 연산을 제외하고는 아주 작은 비용으로 탐색순서를 결정할 수 있다.

III. 제안하는 알고리즘

1. 실시간성 보장

실시간성이란 해당 시간 내에 연산을 처리해야 한다는 것을 의미한다. 특히 하드웨어 연산기를 설계하여 모든 경우에 대한 실시간성을 보장하기 위해서는 최악의 조건(worst case)에 대한 보장을 수행해야 한다. 하지만, 우리가 사용하는 대부분의 알고리즘은 단순히 연산수를 감소시키는 일반적인 방법을 사용하고 있거나, 임계값을 설정하여 임계값 초과시에는 계산을 중지하는 방식으로 최악의 조건까지 연산을 수행하지 않는 형태를 가진다.

움직임 추정기의 경우 단독수행이 아닌 다른 하드웨어들과의 연관 연산을 수행하기 때문에 실제로 특정 시간 내에 연산결과를 출력하는 것이 바람직하다. 하지만, 실시간 움직임 추정기는 최악의 조건을 가정하여서 설계해야 하므로 실제 동작 시에는 연산기가 가지는 성능의 일부만을 사용하는 경우가 발생한다. 특히 움직임 추정기의 경우 최악 연산량과 평균 연산량의 차이가 커 실시간성을 보장하기 위한 하드웨어를 설계한다면 실제 하드웨어 구현시 평균 연산량보다 훨씬 큰 연산을 수행할 수 있어야 하기 때문에 연산기 수가 증가할 수 있다.

표 1은 4CIF급(704×576)의 4개의 영상 시퀀스(ice, soccer, harbour, crew)에 대해서 탐색 영역을 64×64로, 비트 전송률(bitrate)을 768kbps로 설정하였을 때, 다양한 움직임 추정 기법에 조기 중단 기법을 적용하였을 때 한 프레임을 처리하기 위한 연산수를 비교한 것이다. 사용된 동영상 압축 규약은 국제적으로 널리 사용되는 H.264/AVC이며 기준 소프트웨어 (reference software)는 JM15.0이다. 최악 연산량은 조기 중단이 일어나지 않았을 때의 연산량이며, 평균 연산량은 실제로 조기 중단이 일어났을 때의 연산량이고, A/W

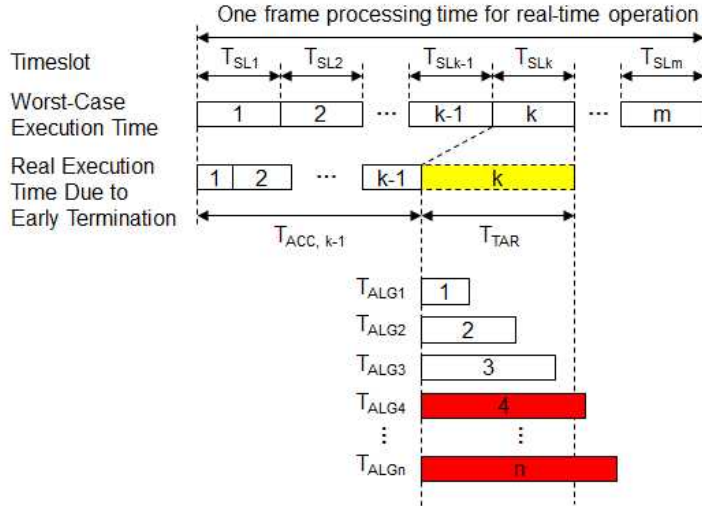


Fig. 3. Adaptive workload control method [12]
 그림 3. 적응적 작업량 제어 기법 [12]

비율은 최악 연산량 대비 평균 연산량의 비율이다.

일반적으로 움직임 추정 기법에서는 영상의 형태, 움직임 벡터 예측값, SAD의 값의 크기에 따라서 조기 중단 기법에 따라 절약할 수 있는 연산량이 매우 크다. 표 1에서 보듯이 각 기법에 조기 중단 기법을 적용했을 때 최악 연산량은 평균 연산량에 비해 4배에서 25배까지 높다는 것을 알 수 있다.

Table 1. Worst-case and average-case calculation counts for various algorithms with early termination

표 1. 조기 중단 기법을 적용했을 때 각 알고리즘별 평균 연산수와 최악 연산수 비교

알고리즘	PSNR (db)	Average-case	Worst-case	A/W 비율
FS[3]	32.66	1,368,521,215	13,598,982,144	0.10
UMHEX[10]	32.54	42,637,606	849,936,384	0.05
EPZS[11]	32.59	32,070,703	849,936,384	0.04
5SS[7]	32.24	31,299,498	136,122,624	0.23

이때, 만약 평균 연산량에 맞추어 움직임 추정기를 설계한다면 평균보다 높은 연산량을 가지는 프레임에 대한 처리가 불가능 하므로 실시간 움직임 추정기 하드웨어의 사양을 결정할 때는 최악 연산량에 맞춰 설계할 수밖에 없다. 하지만 실제로는 조기 중단 기법에 따라 연산량이 가변적으로 변하므로 실시간성을

보장하기 위한 최악 연산량과 평균 연산량의 차에 해당하는 시간 간격이 잉여시간으로 남게 된다. 이와 같은 잉여시간을 이용하기 위해 동적 전압 제어 (dynamic voltage scaling)[12]와 같은 방법을 사용하기도 하지만 별도의 복잡한 회로가 필요하며, 연산시간이 남는다는 것은 연산기의 연산효율이 높지 못하다는 것을 의미한다.

2. 적응적 작업량 제어 기법

실시간성을 보장하기 위해서는 가변적인 연산에 대응할 수 있어야 하며 연산량의 변화에 따라서 연산을 조절할 수 있어야 한다. 본 논문에서는 이를 위해 여러 알고리즘 중에서 주어진 시간 내에 연산을 끝낼 수 있는 알고리즘을 선택하여 수행하는 적응적 작업량 제어 기법[12]을 사용한다.

적응적 작업량 제어 기법은 하드웨어가 선택할 수 있는 다수의 알고리즘이 존재할 때, 실시간으로 처리해야 하는 시간 구간을 몇 개의 시간단위(timeslot: TS)로 나눈 다음에 각 시간 단위마다 실시간 조건을 만족하면서 최대한 성능이 우수한 알고리즘을 선택하여 수행하는 방법이다.

그림 3과 같이 프레임 단위로 실시간 수행 조건을 만족하는 움직임 추정 기법에서 하나의 프레임을 수행하는 시간을 m개의 TS로 나누고, 각 TS마다 n개의 움직임 추정 알고리즘인 ALG₁, ALG₂... ALG_n을 선택할 수 있다고 가정하자. 일반적으로 성능이 우수

한 알고리즘일수록 많은 연산량을 필요로 하므로, i 가 클수록 i 번째 알고리즘 ALG_i 의 성능과 연산량도 높아진다고 가정한다.

각 TS마다 선택된 알고리즘이 최악연산량을 가진다면 $k-1$ 번째 TS까지 실행된 수행 시간을 합친 누적

$$\text{수행시간 } T_{ACC,k-1} = \sum_{i=1}^{k-1} T_{SLi} \text{와 같아야 하지만, 실제}$$

로는 조기 중단에 의해서 상당한 시간이 남게 된다. 따라서 그림 3에서 k 번째 TS가 움직임 추정을 수행할 수 있는 시간 여유인 잉여시간은 T_{TAR} 와 같게 된다. 이때, k 번째 TS에서 선택할 수 있는 i 번째 알고리즘의 최악연산시간이 T_{ALGi} 라고 한다면 그림 3에서처럼 k 번째 TS에서 선택할 수 있는 알고리즘은 ALG_1, ALG_2, ALG_3 뿐이며, 이 중에서 성능이 가장 우수한 ALG_3 을 선택하게 된다.

즉, 적응적 작업량 제어 기법은 이전 시간단위까지 조기 종료에 의해 발생된 잉여시간 내에 현재 시간단위의 연산을 종료할 수 있는 알고리즘 중에서 가장 높은 성능을 가진 알고리즘을 선택하는 방법이다.

이 기법에서는 프레임 단위로 실시간 동작이 확실하게 보장된다. 또한, 잉여시간이 많이 남을수록 선택되는 알고리즘의 성능이 높아지고 현재 시간단위를 처리하는데 필요한 연산시간이 길어지기 때문에 전체적으로 보면 잉여시간은 거의 남지 않고, 조기 종료로 인해 얻어지는 잉여시간이 거의 모두 알고리즘의 성능을 향상하는데 사용되게 된다.

3. 경성 실시간 적응적 탐색 순서 결정 기법

적응적 작업량 제어 기법은 기본적으로 앞서 기술한 잉여 시간을 이용하는 방법으로 이를 위하여서는 기존의 방법보다 좀 더 많은 잉여시간을 가지는 알고리즘이 필요하게 되며, 해당 부분을 만족하기 위한 많은 알고리즘들이 존재한다. 하지만 여러가지 알고리즘을 사용하는 경우 하드웨어관점에서는 해당 알고리즘을 처리하기 위한 또 다른 하드웨어 블록들을 설계하여야 한다는 부담감을 가지게 된다. 본 논문에서는 이러한 문제점을 해소하기 위해 II.4절의 N단계 탐색 방법에 II.4절의 적응적 탐색 순서 결정 기법을 다음과 같이 결합한 후 III.2절의 적응적 작업량 제어 기법을 적용하여 경성 실시간 적응적 탐색 순서 결정 기법을 제안하였다.

경성 실시간 적응적 탐색 순서 결정 기법에서 재정의된 연산순서는 예측 움직임 벡터로부터의 거리, 즉, 움직임벡터가 존재할 확률이 높은 순서대로 연산을 수행하게 된다. 왜냐하면, 일반적으로 예측지점에 움

직임 추정벡터가 존재할 확률이 높고, 그 지점에서 거리가 멀수록 존재 확률이 적어진다. 그러므로 존재 확률이 높은 부분 근방만을 계산하는 것이 적은 연산으로 좋은 결과값을 낼 수 있다고 볼 수 있다.

본 논문에서는 이러한 가정 하에 적응적 탐색 순서 결정 기법에서 사용하는 탐색 위치의 개수만을 다르게 하여 표 2와 같이 4개의 알고리즘인 L0, L1, L2, L3를 생성하였다. 이때 사용된 동영상 압축 규약은 H.264/AVC이며 기준 소프트웨어는 JM15.0이다. 사용 영상은 4CIF급(704×576)의 4개의 영상 시퀀스(ice, soccer, harbour, crew)이며, 탐색 영역 크기는 64×64이고, 비트 전송률은 768kbps로 설정하였다.

경성 실시간 적응적 탐색 순서 결정 기법에서는 적응적 탐색 순서 결정 기법에서 그림 2와 같이 결정된 8개의 탐색 위치 중에서 L0는 ①~⑧의 8개, L1은 ①~⑤의 5개, L2는 ①~③의 3개, L3는 ①의 1개 탐색 위치만을 사용한다. 이때, L3에서 L0로 갈수록 성과 연산량이 높아지므로, L0, L1, L2, L3 알고리즘은 각각 그림 3의 ALG4, ALG3, ALG2, ALG1 알고리즘에 해당한다고 볼 수 있다. L0는 표 1의 5SS과 동일한 탐색 위치를 순서만 바꾸어 탐색하기 때문에 5SS와 성과와 최악 연산량이 거의 동일하다.

L0~L3는 파라미터만 바꾸어 하나의 하드웨어 블록에서 공통으로 처리가 가능하기 때문에 다수의 알고리즘을 사용할 때 발생하는 오버헤드가 거의 없고, N단계 탐색 기법을 기반으로 하기 때문에 하드웨어 구현이 비교적 간단하며 비교적 높은 움직임 추정 성능을 보인다.

Table 2. Comparison of 4 algorithms in the proposed hard-realtime adaptive search order determination (including overhead of early termination)

표 2. 제안된 경성 실시간 적응적 탐색 순서 결정 기법에 사용된 4개 알고리즘의 비교 (조기 중단 기법의 오버헤드 포함)

레벨	첫 Step 연산수	다른 Step 연산수	매크로블록당 최악 연산수
L0	9	8	85,936
L1	6	5	54,496
L2	4	3	33,536
L3	2	1	12,576

4. 하드웨어 크기 비교

실시간으로 동영상을 압축해야 하는 움직임 추정기 하드웨어는 실시간성을 보장하기 위해 최악 연산량에 맞추어 최악 연산량에 맞추어 하드웨어 크기를 결정해야 한다. 즉, 표 1에서 FS를 실시간으로 수행할 수 있는 하드웨어는 5SS를 실시간으로 수행할 수 있는 하드웨어의 약 100배 크기를 가져야 한다.

본 논문에서는 이러한 점을 고려하여 다양한 알고리즘의 하드웨어 크기를 비교하기 위해 최악 연산량에 기반한 비교 척도인 정규화된 하드웨어 크기(normalized hardware size: NHS)를 도입하였다. NHS는 5SS 알고리즘의 최악 연산량의 배수로 표현되며, NHS가 2.0인 경우는 제어 회로를 제외하고 화소끼리 비교하는 연산부의 크기가 5SS에 비해 2배가 된다는 것을 의미한다.

제어 회로를 제외하는 이유는 화소를 비교하는 횟수나 연산량은 하드웨어를 설계하지 않아도 수작업이나 모의 실험에 의해 손쉽게 계산할 수 있으나 제어 회로의 경우 실제로 하드웨어를 설계하기 전에는 알 수 없기 때문이다. [13]에 따르면 대부분의 디지털 회로에서 비스를 포함한 제어 회로는 연산 회로 크기의 약 20%에 해당하며, 이러한 점을 적용하면 제어 회로의 크기를 반영하지 않아도 상대적인 크기는 NHS와 큰 차이가 없다고 볼 수 있다.

경성 실시간 적응적 탐색 순서 결정 기법에서는 L0, L1, L2, L3 알고리즘 중에서 실시간 동작을 보장하는 알고리즘만 선택하여 수행하므로 이론상으로는 사용 알고리즘 중에서 가장 최악 연산량이 작은 L3 알고리즘의 실시간 수행에 필요한 하드웨어 크기만 가져도 된다. 이때 움직임 추정 동작 중에 조기 중단에 의한 잉여 시간이 발생하면 이를 이용하여 실시간 동작을 보장하는 범위 내에서 L0, L1, L2 순으로 적용 알고리즘을 바꾸어 성능을 높일 수 있다. 본 논문에서는 NHS=0.25, 즉 5SS 알고리즘의 실시간 수행에 필요한 하드웨어의 1/4에 해당되는 하드웨어 크기를 기준으로 하였다.

IV. 모의 실험 결과

제안하는 기법의 성능을 확인하기 위해 국제적으로 널리 사용되는 동영상 압축 규약인 H.264/AVC에 4CIF급(704×576)의 4개의 영상 시퀀스(ice, soccer, harbour, crew), 64×64 크기의 탐색 영역, 768kbps의 비트 전송률로 5단계 탐색 방법을 사용하여 JM15.0

기준 소프트웨어를 수행하였다.

모의 실험 결과는 표 3과 같다. 평균연산량은 프레임 당 평균적으로 수행하는 연산 숫자이며 최악연산량은 전혀 조기 중단이 일어나지 않았을 때, 즉 이론적으로 최악인 경우가 일어났을 때 한 프레임의 처리하기 위해 필요한 연산 숫자이다. 각 연산 숫자는 덧셈, 뺄셈, 절대값, 누산, 비교 등의 기본 연산을 각각 하나로 간주하였다. NHS는 III장 4절에서 설명된 정규화된 하드웨어 크기이며, 표 3의 모든 알고리즘이 거의 비슷한 하드웨어 구조를 가지기 때문에 NHS의 비율은 최악연산량의 비율과 동일하다.

표 3에 의하면 제안하는 경성 실시간 적응적 탐색 순서 결정 기법은 기존 움직임 추정 기법에 비해 0.02~0.44 dB의 PSNR 저하를 보이는 대신에 하드웨어 크기를 1/4~1/400까지 줄일 수 있음을 알 수 있다.

움직임 추정 기법이 어느 정도 하드웨어를 효율적으로 사용하는지는 A/W 비율을 보면 어느 정도 알 수 있다. 하드웨어를 설계할 때에는 최악의 경우가 일어났을 때에도 실시간 동작을 수행해야 하므로 최악 연산량에 대비하여 하드웨어 연산부의 크기를 결정해야 하지만, 실제로 움직임 기법을 수행할 때에는 평균 연산량에 해당하는 하드웨어 연산부만 동작하고 나머지는 실제 활용되지 못한다. 따라서 A/W 비율이 낮을수록 최악 연산량을 대비해서 설계된 하드웨어 연산부를 충분히 활용하지 못한다는 것을 의미한다. 제안하는 기법은 기존 기법에 비해 A/W 비율이 3.7~21.5배 정도 높은데, 이는 제안하는 기법이 하드웨어를 매우 효율적으로 사용하였기 때문에 기본 기법에 비해 하드웨어 크기가 월등히 작으면서도 성능 저하를 최소화했다는 것을 보여준다.

Table 3. Simulation results
표 3. 모의 실험 결과

알고리즘	PSNR (db)	평균연산량	최악연산량	NHS	A/W 비율
FS	32.66	1,368,521,215	13,598,982,144	99.90	0.10
UMHEX	32.54	42,637,606	849,936,384	6.24	0.05
EPZS	32.59	32,070,703	849,936,384	6.24	0.04
5SS	32.24	31,299,498	136,122,624	1.00	0.23
HRASOD	32.22	29,433,096	34,030,656	0.25	0.86

V. 결론

본 논문은 동영상 코덱에서 높은 연산량을 차지하고 있는 움직임 추정기에 대해서 실시간 처리가 가능하고 하드웨어 크기를 크게 줄이면서도 성능 저하를 최소화하는 기법을 제안하였다. 이를 위해 먼저 조기 중단을 효과적으로 일으키기 위해 탐색 순서를 바꾸는 방법인 적응적 탐색 순서 결정 기법을 N단계 탐색 방법에 적용하여 연산량을 감소시켰고, 적응적 작업량 제어 기법을 적용하여 실시간 동작을 보장하였다. 제안하는 기법은 하드웨어 크기를 기존 기법의 1/4~1/400까지 줄이면서도 PSNR 저하는 0.02~0.44 dB에 불과하며, 하드웨어의 사용 효율도 기존 기법의 3.7~21.5배에 달하였다.

References

- [1] C. Lee, "Design of High Performance Dual Channel Pipelined Interpolators for H.264 Decoder", Journal of IKEEE, vol. 13, no. 4, pp. 110-115, Dec. 2009.
- [2] D. Kim, M. Cho, S. Moon, J. Lee, J. Jun, and T. Chang, "SVC Based Multi-channel Transmission of High Definition Multimedia and Its Improved Service Efficiency", Journal of IKEEE, vol. 15, no. 2, pp. 179-189, Jun. 2011.
- [3] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding", IEEE Transactions on Communications, vol. COM-29, pp. 1799-1808, Dec. 1981.
- [4] D. Guevorkian, A. Launianen, P. Liuha, and V. Lappalainen, "Architecture for the sum of absolute differences operation", Proceedings of IEEE Workshop on Signal Processing Systems, pp. 57-62, Oct. 2002.
- [5] H. Yang and S. Lee, "Modified 3-step Search Motion Estimation Algorithm for Effective Early Termination", Journal of IEEK, vol. 47, no. 7, pp. 70-77, Jul. 2010.
- [6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion compensation inter frame coding for video conferencing", Proceedings of National Telecommunication Conference, pp. G.5.3.1-G.5.3.5, Nov. 1981.
- [7] I. E. G. Richardson, *Video Codec Design*, Wiley, Apr. 2004.
- [8] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 313-317, Jun. 1996.
- [9] K. Nam, J. Kim, R. Park and Y. Shim, "A fast hierarchical motion vector estimation algorithm using mean pyramid", IEEE Transactions on Circuits and Systems for Video Technology, vol. 5, no. 4, pp. 344-351, Aug. 1995.
- [10] C. Zhu, X. Lin, and L. Chau, "Hexagon-based search pattern for fast block motion estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 12 no. 5, pp. 349-355, May 2002.
- [11] A. M. Tourapis, O. C. Au, and M. L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 10, pp. 934-947, Oct. 2002.
- [12] S. Lee and T. Sakurai, "Run-time Power Control Scheme Using Software Feedback Loop for Low-Power Real-time Applications", Proceedings of Asia-Pacific Design Automation Conference, pp.381-386, Jan. 2000.
- [13] M. Flynn and W. Luk, *Computer System Design: System-on-Chip*, Wiley, Oct. 2011.

BIOGRAPHY

Hyeon-Cheol Yang (Member)



2004 : BS degree in Electronic Engineering, Soongsil University.

2006 : MS degree in Electronic Engineering, Soongsil University.

2006~Now : PhD candidate in Electronic Engineering, Soongsil University
<Main Interest> Low-Power SoC Design, Multimedia SoC Design

Seongsoo Lee (Life Member)



1991 : BS degree in Electronic Engineering, Seoul National University.

1993 : MS degree in Electronic Engineering, Seoul National University.

1998 : PhD degree in Electrical Engineering, Seoul National University.

1998~2000 : Research Associate, University of Tokyo

2000~2002 : Research Professor, Ewha Womans University

2002~Now : Associate Professor in School of Electronic Engineering, Soongsil University

<Main Interest> Low-Power SoC Design, Multimedia SoC Design, Battery Management