

# 이동 단말상에서의 움직임 검출 알고리즘의 설계 및 구현

김선태\*, 마평수

한국전자통신연구원 임베디드소프트웨어 센터

## Design of Motion Estimation algorithm on Mobile device

SeonTae Kim, PyeongSoo Mah

Embedded software Center, ETRI.

### 요 약

본 논문에서는 적은 메모리 공간을 가진 이동 단말 장치와 같은 임베디드 시스템에서 많은 메모리를 필요로 하는 비디오 디지털 신호처리 알고리즘이 동작될 수 있도록 움직임 검출을 효율적으로 구현하는 방법을 제시한다. 즉, 움직임 검출 알고리즘을 구현할 때 현재의 매크로 블록과 다음의 매크로 블록간의 참조 탐색 영역의 중복성이 존재함을 인지하여, 필요 메모리의 데이터 이동을 최소화시켜 내부 메모리를 효율적으로 이용하려 한다. 이렇게 움직임 검출에서 적은 내부 메모리를 이용함으로써 내부 메모리를 다른 용도로 사용할 수 있는 가용성을 증대시키고, 내부 메모리에서 보다 많은 연산을 수행하게 함으로써 빠른 성능으로 복잡한 MPEG-4 비디오 부호화 알고리즘의 구현이 가능하도록 한다.

### 1. 서론

임베디드 시스템에서 현재 사용되고 있는 저 비트율 비디오 압축 알고리즘은 표준 MPEG-4로 기본 구조는 일반 비디오에서 사용되는 사각형의 텍스처 부호화 부분과 임의의 형상을 압축하는 형상 부호화 부분으로 구성되어 있어서 다양한 응용에 대처할 수 있도록 되어있다. 이 중 텍스처 부호화는 공간정보와 시간 정보에 대한 불필요한 데이터를 압축하는 것으로, 현재 다양한 표준의 공통 알고리즘으로 사용된다.

다.

이 알고리즘은 공간과 시간에 대한 정보를 압축하는 것으로, 공간 정보를 압축하기 위해서는 이산 여현 변환/역변환(DCT/IDCT)과 양자화 과정이 요구되며, 시간 공간에 대한 압축을 수행하기 위해서는 움직임 검출과 움직임 보상, 이전 화면의 저장을 위한 버퍼링 과정이 필요하다. 가변장 부호화(VLC)는 공간정보와 시간정보에 대해서 양자화된 값을 정보의 확률에 따라 정보 압축을 수행한다[3].

MPEG-4에서 움직임 검출 알고리즘은

상당히 많은 메모리 접근 및 메모리 크기를 필요로 하여 MPEG-4 비디오 부호기 알고리즘의 성능에 지대한 영향을 미친다. 움직임 검출은 프레임의 방향에 따라 P, B프레임 형식으로 나뉘지만, 메모리 접근에 의한 움직임 검출방식은 비슷하다. P프레임의 움직임 검출은 그림1에서 보듯이, 이전 참조 프레임(t-1)과 현재의 프레임(t)을 가지고 수행하는데, 현재 프레임의 매크로 블록에 대해서 이전 프레임의 탐색 영역 내에서 픽셀단위로 움직여 가장 유사한 위치를 찾아내는 것이다. 유사도를 찾는 방법은 MSE(Mean Square Error), SAD(Sum of Absolute Difference)나 MAD(Mean Absolute Difference) 등으로 수행된다.

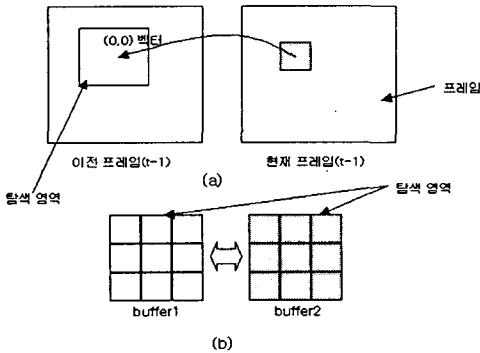


그림1. 기존의 움직임 검출 방법

## 2. 기존의 연구 및 문제점

그림1은 기존의 움직임 검출하는 방법을 보여주는 것으로, 그림1의 (a)의 방식은 참조 프레임과 현재의 프레임을 외부 메모리에 저장하여 유사도를 찾는 것으로 비디오의 프레임 해상도가 클수록 상당히 많은 메모리를 요구한다. 따라서 적은 내부 메모리에는 저장할 수 없어 대부분 외부 메모리에 놓고 수행해야 한다. 외부 메모리를 이용하게 되는 경우, 동작 속도가 빠른 프로세서 비해 메모리의 접근 속도가 느려 시스템 속도에 저하가 일어난다. 뿐만 아니라, 외부 메모리의 잦은 접근은 많은 전력 소모를

일으키므로 전력 소모를 중요시하는 이동단말에서는 구현이 어렵다.

그림1의 (b)는 그림1의 (a) 방법의 단점을 보완하기 위해 이중 버퍼링(double buffering)을 이용하는 것으로, 이를 효과적으로 사용하기 위해서는 메모리간의 데이터 이동에 프로세서의 간섭이 없는 DMA가 필요하다. 즉, 하나의 매크로 블록이 수행될 때 그림1-(b)의 buffer1에 저장한 참조 프레임을 사용하고 동시에 DMA를 이용하여 그림1-(b)의 buffer2에 프로세서의 간섭 없이 다음 참조 프레임을 저장한다. 원하는 매크로의 블록에 대해 움직임 검출이 수행되면, 다음 매크로 블록에 대한 움직임 검출은 이미 DMA를 이용해서 이동된 다음 참조 프레임을 가지고 움직임 벡터를 찾게 된다[2].

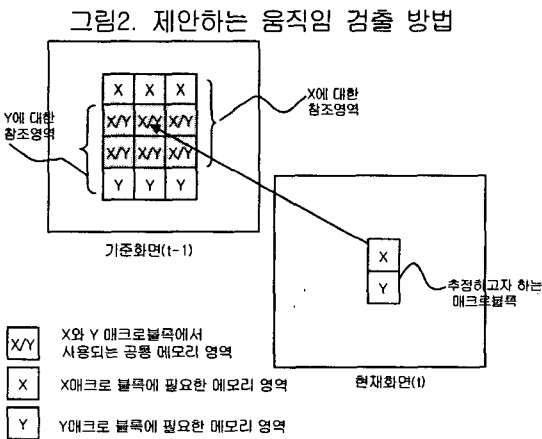
## 3. 설계 내용

본 논문에서 제안한 방법은 위에서 제시한 기존의 움직임 검출의 문제점을 극복하고 보다 효율적으로 메모리를 사용하는 방법을 제시함으로써 메모리가 적은 이동 단말상에서 효과적으로 MPEG-4 비디오 부호기가 작동하도록 한다.

### 3.1 기본 방법

그림2는 기본적인 움직임 검출시의 메모리의 사용법을 제시하는 것으로, 움직임 검출을 위한 연속적인 매크로 블록 X와 Y에 대해서 각각의 매크로 블록에 따른 참조 탐색 영역을 보여준다. 그림2에서 제시한 것처럼 현재 프레임의 연속하는 매크로 블록에 대해서 과거 프레임의 참조 탐색 영역에서 중복된 메모리의 블록이 발생한다. 이때 기존의 방법에서는 참조 탐색 영역에 중복된 메모리 블록(X/Y)을 폐기하고 다음 매크로 블록 움직임 검출시 폐기된 데이터를 갱신하게 된다. 하지만, 본 논문에서는 중복된 메모리 블록을 재사용하고 필요한 메모리 블록

만 버퍼에 저장하여 메모리의 접근을 줄인다. 또한, 그림2와 같이 버퍼의 크기를 줄이고, 버퍼의 개수를 늘려서 기존의 이중 버퍼링에서 사용되는 내부 메모리의 크기를 줄여 작은 내부 메모리를 보다 효과적으로 사용하게 한다. 즉, 그림1의 (b)에서 제시한 방법은 이중 버퍼를 이용한 것으로, 내부 메모리로의 참조 탐색 영역을 보면, 하나의 매크로 블록에 대한 움직임 검출마다 9개의 매크로 블록을 사용하고 폐기한다. 하지만, 그림2에서 제안하는 방식에서는 3개의 매크로 블록으로 구성된 버퍼를 하나 더 두어 순환 버퍼형식을 취했으며, 메모리간의 데이터 이동은 단지 3개의 매크로 블록만을 메모리 이동시켜 움직임 검출이 될 수 있도록 하였다.



4개의 매크로 블록에 대한 움직임 검출이 수행될 때, 데이터의 이동크기를 비교해 보면, 기존의 이중 버퍼를 사용했을 때는 buffer1과 buffer2를 각각 두 번씩 사용하므로 9개의 매크로 블록에 4를 곱한 36개의 매크로 블록의 데이터 이동이 발생한다. 하지만 본 논문에서 제안하는 순환버퍼의 경우에는 18개의 매크로 블록의 데이터 이동이 있어, 기존의 반만을 이동해서 움직임 검출을 수행할 수 있다. 매크로 블록의 개수가 커지면 이 둘의 데이터 이동비는 더 1/3의 값으로 수렴한다. 즉, 간단히 매크로 블

록 개수를 N이라 하면, 이중 버퍼의 경우에는  $N*9$ 이 되고, 순환버퍼의 경우에는  $(N+2)*3$ 이 된다.

### 3.2 배열 형식 및 탐색 구간

버퍼에 저장될 데이터는 3차원 배열형식으로 저장되는데, 그림2의 세로방향 움직임 추정인 경우에는 3개의 매크로 블록이  $48*16$ 의 배열을 만들어 하나의 버퍼를 형성하고 각 움직임 검출이 일어날 때마다 버퍼 하나의 데이터 이동으로 하나의 움직임 벡터를 구할 수 있다.

그림 3. 배열형식 및 탐색영역에서의 매크로 블록 위치

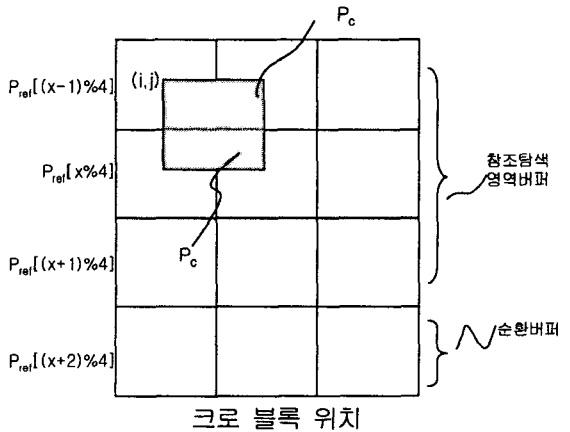


그림3에서는 간단히 탐색 구간을  $[-16, 15]$  설정했을 때의 원하는 매크로 블록( $P_c$ )에 대한 참조 탐색 영역( $P_{ref}[x-1], P_{ref}[x], P_{ref}[x+1]$ )을 나타내는 것으로, 두 매크로 블록의 유사도 SAD를 구하려면 두 개의 버퍼에 걸쳐있는 경우가 있으므로 이에 대한 고려가 있어야 한다. 즉, 움직임 좌표 (i,j)에 대해서 유사도 SAD는  $P_c$ 의  $SAD_1$ 과  $P_c$ 의  $SAD_2$ 의 합으로 구해진다. 여기서 i의 움직임은 하나의 버퍼( $P[]$ )에 속하므로 i의 증가에 따른 식의 변화가 없다. 하지만, j의 경우에는 다른 버퍼 영역에 속할 수 있으므로 두 가지 범위를 두어야 하며, 매크로 블록( $P_c$ )에 대한 움직임 검출로 움직임 벡터 값이 구해지면  $P[(x+2)\%4]$ 의 순환버퍼에

다음 매크로 블록에 대한 움직임 검출을 위해 하나의 버퍼 데이터가 이동되어 이전의 중복된 버퍼와 하나의 참조 탐색 영역이 형성되어진다.

탐색구간의 변화에 따라 내부 메모리의 버퍼 할당이 달라지는데, 탐색구간의 변화에도 버퍼의 할당을 일정하게 유지하면서 움직임 추정이 가능하다. 그림3에서는 탐색 구간이 [-16,15]로 설정하고 있지만, 위와 같은 버퍼의 구성에서 [-8,7]의 탐색구간으로 움직임 검출을 하고자 한다면, 초기에는 움직임 검출을 4번을 하고, 버퍼의 이동이 있을 때 하나의 움직임 검출을 하는 것이 아니라, 두개의 움직임 검출 (X,Y)을 바로 수행하면 된다.

위에서는 세로방향으로 움직임 검출을 수행할 때 움직임 방향, 버퍼의 배열 방식, SAD 구하는 방법 등을 나열하였는데, 가로방향을 먼저 움직임 검출할 때도 세로방향과 유사하게 진행하면 된다.

#### 4. 결론

본 논문에서는 적은 메모리 공간을 가진 이동 단말 장치와 같은 임베디드 시스템에서 많은 메모리를 필요로 하는 비디오 디지털 신호처리 알고리즘이 동작될 수 있도록 움직임 검출을 효율적으로 구현하는 방법을 제시하였다. 즉, 움직임 검출 알고리즘을 구현할 때 참조 탐색 영역의 필요 메모리 데이터를 최소화시켜 내부 메모리를 효율적으로 이용하였다. 이렇게 움직임 검출에서 적은 내부 메모리를 이용함으로써 내부 메모리를 다른 용도로 사용할 수 있는 가용성을 증대시켰고, 내부 메모리에서 보다 많은 연산을 수행하게 함으로써 빠른 성능으로 복잡한 MPEG-4 비디오 부호화 알고리즘의 구현이 가능하도록 했다.

뿐만 아니라, 전력을 많이 소비하는 외부 메모리의 접근을 낮추고 적은 전력을 소모하는 칩 내부 메모리를 사용하므로 이동형 배터리를 사용을 장기간 동안 유지할 수 있으므로 이

동장치에 보다 좋은 휴대 편리성을 증대 시킨다

#### [참고문헌]

- [1] Seongmo Park, Miyoung Lee etc, A .MPEG-4 Video Codec chip with Low power scheme for mobile Application , IEICE Trans. Fundamentals, vol.E86-A, No.6, June. 2003
- [2] Nuno Roma and Leonel Sousa, Efficient and Configurable Full-Search Block-Matching Procerrors , IEEE Transactions on Citcuit and systems for Video technology, vol.12,No.12,Dec.2002
- [3] ISO/IEC 14496-2, Information technology coding of audio-visual objects part2 visual , Dec.2001