

# 비디오 스트림에서 이벤트 검출에 관한 알고리즘 연구

이승열\*, 이조원\*, 김상민\*, 오미경\*, 임영환\*\*

\*송실대학교 대학원 컴퓨터학과

\*\*송실대학교 정보과학대학 컴퓨터학부

e-mail:sylee@archi.soongsil.ac.kr

## A Study on Modeling of Event in video stream for Description

Seungyoul Lee\*, Jowon Yi\*, Sangmin Kim\*, Mikyung Oh\*,  
Younghwan Lim\*\*

\*Dept. of Computing Soongsil Univ.

\*\*School of Computing, Soongsil Univ.

### 요약

본 논문에서는 H.261로 압축된 비디오 데이터에서 움직임 검출과 장면 변환과 같은 이벤트를 압축된 정보인 움직임벡터를 이용하여 검출하는 새로운 알고리즘을 제안하고 구현하였다. 본 논문에서 제시하는 알고리즘은 압축된 데이터를 이용함으로써 성능 향상은 물론 움직임 검출, 움직임 기간 검출, I 프레임 검출 등의 다양한 이벤트의 실시간 검출을 가능하게 한다.

### 제1장 서론

멀티미디어 데이터를 이용할 때 사용하는 압축 기술은 데이터의 효율적 사용이 가능하게 했다. 그러나, 압축기술의 사용은 데이터의 압축과 복원과정에서 오는 시간적 지연을 피할 수 없으며, 이런 지연은 실시간 처리를 요구하는 응용프로그램에 있어서 압축된 데이터가 갖는 장점보다 더 큰 문제점을 초래할 수 있다. 이러한 문제점의 한가지 해결방법으로 복원된 영상에서 얻을 수 있는 정보를 압축된 영상에서 얻는 방법을 통해 영상의 복원과정에서 오는 시간적 지연은 없애는 방법이 있을 수 있다. 본 논문에서는 압축된 데이터의 실시간 처리를 위하여 압축된 영상에서 압축정보를 이용하여 얻을 수 있는 몇 가지 의미정보의 검출방법을 제안하고 구현하였다.

### 제2장 관련 연구

#### 2.1 이벤트(Event)의 정의

동영상 압축 부호화의 표준인 H.261이나 MPEG

에서는 동영상의 시공간적인 정보의 중복성을 제거하여 고압축을 실현하기 위하여 움직임 추정 및 보상과 같은 기술을 적용하고 있다. 따라서 이러한 기술을 적용하여 얻어지는 움직임벡터는 압축된 동영상에서 의미정보를 추출하기 위한 가장 중요한 정보라 할 수 있다. 본 논문에서는 움직임벡터와 같은 압축정보를 이용하여 검출할 수 있는 압축된 비디오 스트림의 의미정보를 이벤트(Event)라 정의하고 정의된 이벤트를 검출하기 위한 알고리즘을 제안한다.

#### 2.2 기존의 비디오에서 의미정보 추출 연구

##### 2.2.1 영역 분할을 통한 움직임 검출

분할된 영역을 이용한 움직임 검출 방법은 전체 프레임을 비교하는 방법에 비해 빠르고 효율적인 성능을 보인다. 일반적으로 동영상에서 서로 다른 특징을 분할하기 위해 다음의 것을 이용한 영역분할 방법들이 주로 사용된다.

- ▶ 밝기 값의 분포
- ▶ 칼라 정보
- ▶ 윤곽
- ▶ 혼합 기법

▶ 균집화

영역 분할이 이루어지면 이동영역에서 추출된 차이 값들에 의해 이동물체를 추출하게 된다. 비교 방법으로는 다음과 같은 방법들이 사용된다.

- ▶ 차영상 분석에 의한 방법
- ▶ 광류(Optical Flow) 측정 및 분석에 의한 방법
- ▶ 블록 정합 알고리즘(Block Matching Algorithm)

이들 기법들은 동영상의 영역을 분할하는 효율적인 방법들이지만 밝기, 칼라, 윤곽 등의 정보는 압축되지 않은 비디오 데이터에서 얻을 수 있는 정보이다. 따라서, 압축되어 전송되어온 데이터는 복원과정이 필요하게 된다. 이때 발생하는 불필요한 시간적 지연은 멀티미디어의 큰 이슈인 실시간성을 구현할 수 없게 한다.

2.2.2 장면 검출

복원된 비디오 데이터로부터 장면전환을 추출하는 방법으로 화소 단위 비교법과 유사율 측정법 등이 있다. 이들 방법은 연산 시간이 적게 걸린다는 장점이 있으나 비디오 데이터들 사이에 섬광이나 선명한 조명이 있을 경우 오류를 발생하는 단점과 저장공간을 많이 필요로 한다는 문제점이 있다.

2.3 문제점

2.3.1 기존 연구의 문제점

압축된 데이터를 앞에서 언급한 비디오에서 이벤트를 추출하는 기법들을 적용하기 위해서는 영상을 복원하는 과정이 필요하고, 이에 많은 문제점이 존재한다. 따라서 이 문제를 극복하기 위하여 본 논문에서는 이벤트를 검출하기 위해 필요한 정보를 추출하는 방법과 이 추출된 정보를 이용하여 압축된 데이터 속에서 이벤트를 검출하는 방법들을 제안한다.

2.3.2 압축된 영상에서 이벤트 검출 시 문제점

움직임검출과 같은 이벤트에 있어서 단순히 동영상에서 움직임만을 검출하는 방법은 움직임벡터를 이용하여 쉽게 구할 수 있고 압축된 비디오영역의 분할과 분할된 영역을 대표하는 움직임벡터를 추출하여 수행시간을 줄임으로 향상된 움직임검출도 가능하다. 이벤트를 검출하는 과정에서 발생 가능한 또 다른 문제점은 동영상에는 잡음이 포함되어 있다는 것이다. 잡음은 움직임벡터나 DCT계수에 영향을 미치게 된다.

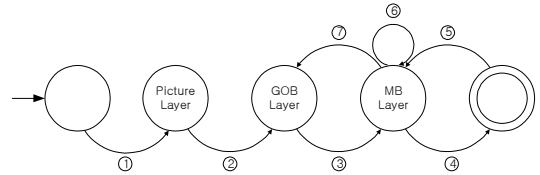
제3장 H.261의 비디오에서 압축정보 추출 오토마타

3.1 H.261의 Data 구조

H.261은 다음과 같은 4개의 계층(layer)을 갖는다.

- ▶ Picture: 픽처(Picture)헤더와 GOB들로 구성.
- ▶ Group Of Block(GOB): GOB 헤더와 MB들
- ▶ Macroblock(MB): MB 헤더와 블럭(4개의 luminance block과 2개의 color difference blocks)들
- ▶ Block: 실제 데이터를 갖는 계층

3.2 움직임벡터 추출 오토마타

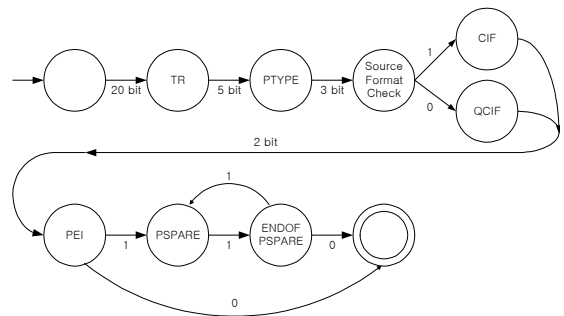


[그림 3-1] 움직임 벡터 추출 오토마타

[그림 3-1]은 움직임벡터 추출을 위한 과정을 간략히 오토마타로 나타낸 것으로 각각의 상태는 자신의 계층 헤더를 처리하고 전이(transition)가 일어난다. 전이가 일어나는 경우는 다음과 같다.

- ① Picture Start Code(PSC)를 찾은 경우 Picture 계층으로 전이가 일어난다.
- ② GOB Start Code(GBSC)를 찾은 경우 GOB 계층으로 전이가 일어난다.
- ③ MB address(MBA)를 찾은 경우 MB 계층으로 전이가 일어난다.
- ④ 모션(Motion) 데이터를 찾은 경우 마지막 상태로 전이가 일어난다.
- ⑤ 다음 MB를 수행하기 위해 MB 계층으로 전이가 일어난다.
- ⑥ 모션 데이터가 없는 MB일 경우 다음 MB가 수행되기 위해 MB 계층 전이가 일어난다.
- ⑦ MB의 수행이 완료되면 다음 GOB가 수행되기 위해 GOB 계층으로 전이가 일어난다.

3.2.1 픽처 계층

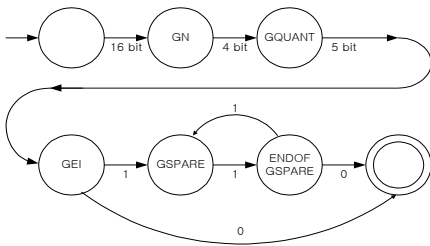


[ 그림 3-2 ] Picture 계층 오토마타

- ▶ TR(Temporal reference): 전송되지 않은 픽처수
  - ▶ PTYPE: 픽처 종류에 대한 정보
  - ▶ PEI: Extra insertion information
  - ▶ PSPARE: Spare information
- 픽처 계층의 초기조건은 PSC이며 GBSC의 값이 나오면 Picture 계층으로 종료한다. 픽처 계층에서 사용되는 거의 모든 헤더 값은 사용되지 않으며 단지 소스의 형식만을 조사하여 GOB와 MB의 개수를 결정한다.

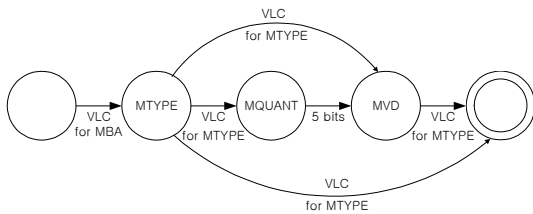
### 3.2.2 GOB 계층

- ▶ GN(Group Number): GOB 위치
  - ▶ GQUANT: Quantizer information
  - ▶ GEI: Extra insertion information
- GOB계층의 초기조건은 GBSC며 MB Address가 나오면 종료한다. GOB의 모든 헤더는 사용되지 않는다.



[ 그림 3-3 ] GOB 계층 오토마타

### 3.2.3 매크로블럭(macroblock) 계층



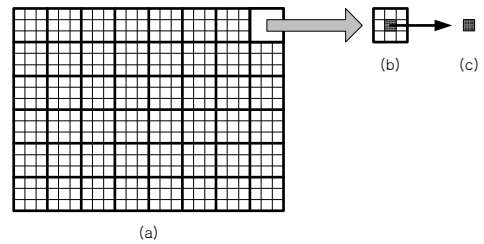
[ 그림 3-4 ] Macroblock 계층 오토마타

- ▶ MTYPE: Type information. MB에 대한 정보
  - ▶ MQQUANT: Quantizer information
  - ▶ MVD(Motion Vector Data): MC(Motion Compensation)를 포함하는 모든 MB에 포함
- MB Address Table에 해당하는 값을 조사하여 해당되는 코드값을 초기조건으로 가지며 움직임 벡터를 찾으면 종료한다. MB의 type을 나타내는 MTYPE에 의해 움직임 벡터의 유무가 결정된다.

## 제4장 움직임 검출

### 4.1 영역 분할을 이용한 움직임 검출

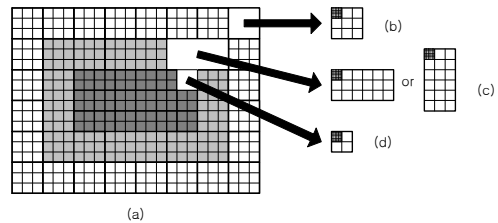
4.1.1 인접 매크로블럭의 상관관계 이용 영역분할  
 동영상의 인접한 MB들은 서로 상관관계가 높으므로 비슷한 물체의 경계를 제외한 인접한 MB들은 비슷한 압축정보를 갖게 될 것이다. 따라서 MB들을 일정한 수로 그룹을 지어 그 그룹을 대표하는 움직임 값을 추출할 수 있다면 움직임을 추출하는 횟수는 그룹의 수만큼만 수행하면 되므로 수행 횟수를 줄일 수 있다.



[ 그림 4-1 ] 상관성에 의한 영역분할 예  
 (a)비디오의 한 프레임  
 (b)Macroblock의 그룹  
 (c)그룹을 대표하는 Macroblock

움직임벡터가 그룹을 대표할 수 있도록 하는 적절한 개수의 MB를 결정하는 것이 이 방법에서 가장 중요하며 실험을 통하여 그 값을 구한다.

### 4.1.2 가중치에 의한 영역 분할



[ 그림 4-2 ] 공간적 특성에 의해 가중치를 부여한 영역분할 예

동영상은 중앙과 테두리 부분에 움직임이 많이 발생하므로 각 부분에 가중치를 두어 영역을 설정하면 움직임벡터 추출 횟수를 효율적으로 줄일 수 있다. 동영상의 공간적 특성에 따라 MB의 개수가 다른 그룹을 결정하고 각 그룹의 대표 움직임 벡터만을 추출함으로써 움직임벡터 추출횟수를 줄일 수 있다.

### 4.1.3 영역분할을 이용한 움직임 검출 알고리즘

먼저 동영상의 한 프레임이 전송되면 그 프레임의 움직임벡터를 추출하고 기존에 정해진 임계값과 비교한다. 움직임이 검출되면 이전 프레임에서 움직임이 있었는지 확인하여 움직임이 없었을 경우만 움직임의 시작을 알리고 이전 프레임에서 움직임이 있었을 경우는 다른 처리를 하지 않는다. 현재 프레임에

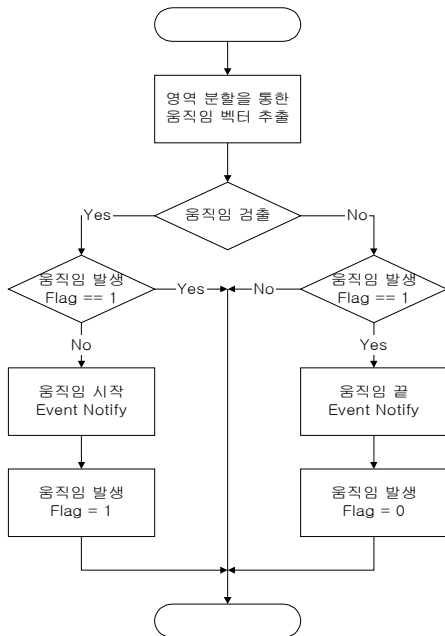
서 움직임이 검출되지 않으면 이전 프레임에서 움직임이 있을 경우만 조사하여 움직임의 끝을 알리게 된다. 이 방법은 구현이 간단하고 빠르다는 장점이 있지만 이동 물체의 속도가 느리거나 움직임의 정지가 자주 발생하면 이벤트가 빈번히 발생한다는 문제점을 가지고 있다.

4.2 움직임 영역 추출을 통한 움직임 검출

4.2.1 움직임 영역 결정

동영상의 프레임에서 움직임 영역을 결정하는 방법은 다음과 같은 과정으로 이루어진다.

- ① 움직임벡터(Motion Vector Data) 추출
- ② 움직임벡터의 잡음 제거
- ③ 움직임 영역 결정



[ 그림 4-3 ] 움직임 검출 알고리즘

[그림 4-4]의 (a)와 같이 먼저 한 프레임 내에 존재하는 모든 움직임벡터를 추출한다. 움직임벡터의 잡음을 제거하는 과정에서는 추출된 움직임벡터의 평균값을 결정하고 이 값의 일정 영역에 해당하는 벡터만을 필요한 값으로 취하여 잡음을 제거하고 움직임 영역을 축소한다. 잡음을 제거하고 얻어진 움직임벡터를 이용하여 움직임 영역을 결정한다.

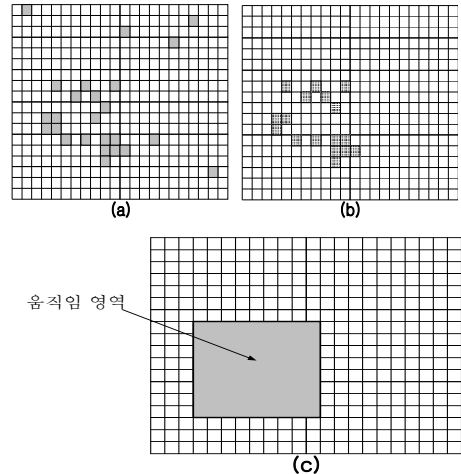
4.2.2 움직임 영역 비교

연속된 프레임에서 이전 프레임에서 움직임 영역이 결정된 경우 현재 프레임의 움직임을 비교하는 방법은 다음과 같은 두 방법이 있다.

- (1) 현재프레임에서 움직임이 없는 경우: 현재프레임에서 움직이던 물체가 존재하고 물체의 움직임이 없

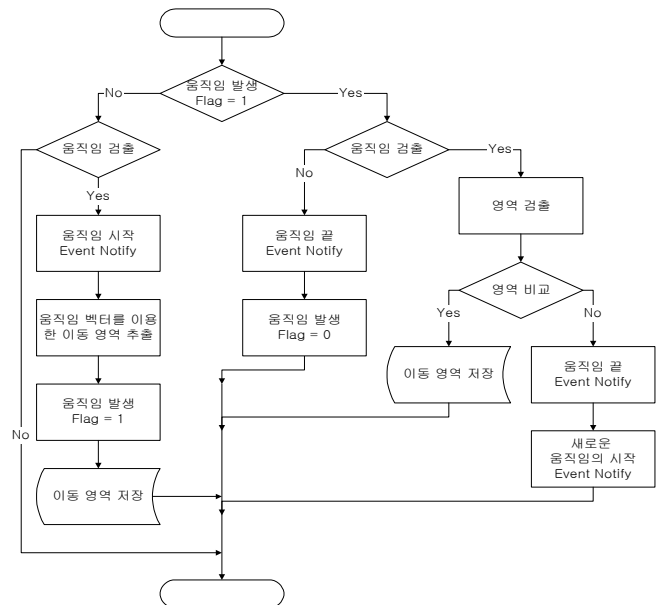
으면 결정된 움직임 영역과 같은 위치의 현재프레임 영역의 MB들은 움직임벡터가 없는 Inter로 부호화되므로 현재프레임의 비교영역이 Intra인지를 확인하여 움직임을 결정 할 수 있다.

- (2) 현재프레임에서 움직임이 있는 경우: 현재프레임에서 움직임이 검출되면 현재프레임에서 비교영역을 결정하고 비교하여 이전 프레임 물체의 지속적인 움직임인지 새로운 물체의 움직임인지를 판단 할 수 있다.



[ 그림 4-4 ] 움직임 영역 추출의 예  
 (a) 움직임벡터 추출  
 (b) 움직임벡터 잡음 제거  
 (c) 움직임 영역 결정

4.2.3 움직임 영역을 이용한 움직임 검출 알고리즘



[ 그림 4-5 ] 움직임 검출 알고리즘

이전에 움직임이 진행 중인지를 조사하여 움직임이 없었을 경우는 현재 프레임에서 움직임을 판단하게 된다. 현재 프레임에서 움직임 있다면 현재 프레임

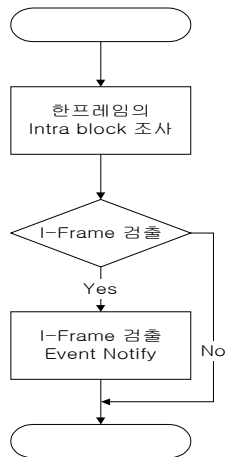
에서 움직임이 시작되었음을 알리고 움직임 영역을 결정하게 된다. 결정된 움직임 영역은 다음 프레임과 비교되기 위하여 저장된다. 이전에 움직임이 진행 중이었다면 현재 프레임에서 움직임이 있는지를 조사하고 없으면 움직임의 끝을 알리고 움직임이 있으면 움직임 영역을 결정하고 움직임을 비교하여 같은 물체인지를 판단하게 된다. 같은 물체로 결정되면 이 영역을 저장하고 다른 물체로 결정되면 움직임의 끝과 새로운 움직임의 시작을 알리게 된다.

제5장 장면 검출 알고리즘

장면 변화에 관한 검출은 H.261 자료구조에서 MB 계층의 MTYPE의 정보를 조사하여 알 수 있다. 비디오에서 영상이 변화되면 변화된 블럭은 이전 프레임 값으로 복원이 불가능하다. 따라서 압축시 변화된 블럭은 Intra로 부호화되며 이 정보로 장면의 변화를 추측할 수 있다. 장면 변화의 정의는 두 가지가 있을 수 있다.

- ▶ I 프레임
- ▶ 새로운 물체의 등장

I 프레임검출은 [그림5-1]의 알고리즘을 이용하여 추출할 수 있다. I 프레임검출과정에서 프레임이 입력으로 들어오면 움직임벡터 추출오토마타를 적용하여 MTYPE을 조사하는 것으로 MB가 Intra인지 검사할 수 있으며 모든 MB가 Intra이면 I프레임을 검출한 것이다.



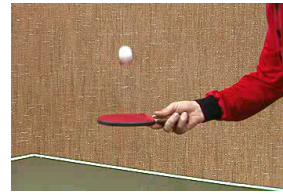
[ 그림 5-1 ] I 프레임검출 알고리즘

제6장 구현 및 결과

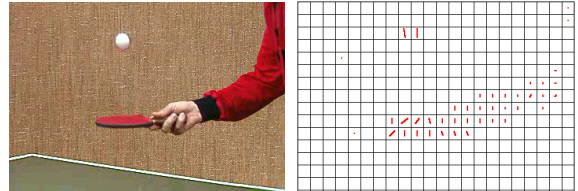
6.1 영역분할을 이용한 움직임 검출

영역 분할하는 첫 번째 방법으로 MB를 일정 수의 그룹으로 묶고 그 MB 그룹을 대표하는 움직임벡터를 추출하여 움직임을 검출하는 방법은 [그림 6-1]의 그림과 같이 잡음 적은 깨끗한 영상에서는 그룹

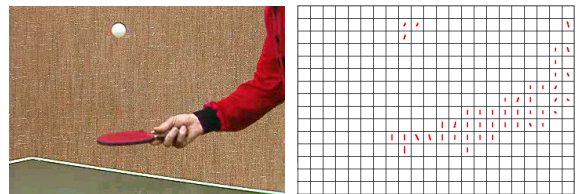
의 수만큼의 속도 향상을 볼 수 있었다.



[ 그림 6-1 ] Intra-Frame

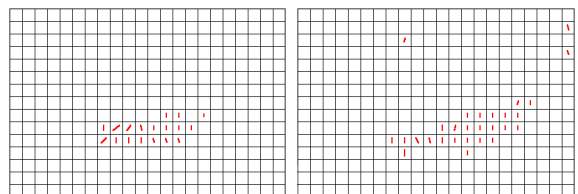


[ 그림 6-2 ] 첫 번째 프레임과 그 움직임벡터



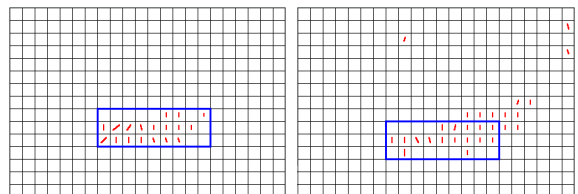
[ 그림 6-3 ] 두 번째 프레임과 그 움직임 벡터

그러나 일반적으로 카메라에서 들어오는 일반영상에서는 많은 양의 잡음이 발생할 수 있고 그 잡음이 대표 움직임벡터로 결정되는 문제로 인해 움직임을 검출하는 성능은 모든 움직임벡터를 조사하여 결정하는 것보다 낮음을 알 수 있었다. 두 번째 영역 분할 방법으로 한 프레임을 영역으로 나누고 가중치를 두어 움직임을 결정하는 방법은 전제 조건으로 카메라의 움직임을 두고 있다. 따라서 실험 데이터로 사용된 깨끗한 영상에 대해서는 낮은 성능을 볼 수 있었다.



(a) 첫 번째 프레임 (b) 두 번째 프레임

[ 그림 6-4 ] 잡음 후의 움직임 벡터



(a) 첫 번째 프레임 (b) 두 번째 프레임

[ 그림 6-5 ] 움직임 영역 결정



## 6.2 움직임 영역 추출을 통한 움직임 검출

움직임 영역을 추출해서 비교하는 방법에 전제 조건은 움직임이 하나여야 한다는 것과 움직임은 평행적으로 움직여야 한다. 실험 데이터로 사용한 [그림 6-1]은 사람의 손과 탁구공 2개의 움직임이 있으나 탁구공과 같은 작은 움직임은 잡음의 제거 과정이나 움직임 영역 결정과정에서 제외될 수 있다. 잡음의 제거 과정으로 중간 값을 사용할 수 있는데 이 방법은 연속된 프레임에서 발생할 수 없는 큰 움직임 벡터와 아주 미세한 움직임 벡터를 잡음으로 결정하고 움직임 영역 계산에서 제외시키는 것이다. 이 방법을 사용해서 [그림 6-2]와 [그림 6-3]의 잡음을 제거 하면 [그림 6-4]와 같은 결과를 얻는다. [그림 6-4]의 (a)의 경우는 움직임벡터가 균집해 있지만 (b)의 경우는 아직도 분산된 움직임벡터가 존재함을 알 수 있다. 이 벡터들은 잡음이 아닌 실제로 움직임이 있는 MB일 수도 있지만 이러한 분산된 벡터들은 움직임 영역 결정에서 제외되어야 한다.

잡음 제거되면 움직임 영역을 결정해야 한다. 이 논문에서 구현된 움직임 영역 결정은 그 이전 프레임에서 결정된 움직임 영역의 위치부터 같은 크기의 움직임 영역을 현재 프레임에서 결정하는 것이다. [그림 6-4]의 각각의 프레임에서 결정된 움직임 영역은 [그림 6-5]와 같다. [그림 3-5]와 같이 영역이 결정되면 이 영역이 물체의 지속적인 움직임인지를 결정해야 한다. 결정된 움직임영역으로 물체의 이동 방향을 계산하면 X축으로 0만큼 Y축으로 -1만큼 이동했음을 알 수 있으며 실제 두 번째 프레임에서 결정된 움직임벡터의 평균값은 X는 0.02, Y는 -1이다. 따라서 이 두 움직임영역은 같은 움직임을 물체의 영역을 나타낸다고 결정 할 수 있다. 움직임의 영역을 결정해서 움직임을 결정하는 방법은 정확한 움직임 물체를 추출하지는 않지만 대략적인 움직임 물체의 이동을 추적할 수 있으며 움직임의 시종을 결정하는 이벤트추출이 가능하고 또한 필요한 블럭만 계산하므로 이벤트추출을 빠르게 할 수 있다는 장점이 있다. 그러나 이 방법이 가지는 전제조건으로 인하여 카메라에서 전송되어 오는 일반적인 데이터에 적용하는데는 어려움을 가지고 있다.

## 제7장 결론

이 논문에서는 압축된 동영상을 복원과정을 거치지 않고 압축정보인 움직임벡터를 사용하여 이벤트를 검출하는 알고리즘을 제안하고 구현하였다. 이벤트

의 하나인 I 프레임 검출은 간단하게 움직임벡터의 유무를 조사하여 검출할 수 있으며 그보다 더 좋은 성능을 기대하는 알고리즘은 없다고 할 수 있다. 모든 움직임벡터를 조사하는 방법을 단순히 움직임을 검출에 이용하면 간단하고 비교적 우수한 성능을 얻을 수 있으나 움직임 기간의 판단과 같은 다른 이벤트 검출에 적용하기에는 어려움이 있다. 따라서 이 논문에서는 움직임 영역을 결정하여 이벤트를 검출하는 방법을 제시하였고 구현하여 가능성을 증명하였다. 움직임벡터는 압축된 영상의 복원을 위한 필수 정보로 사용된다. 따라서 압축된 영상의 의미를 이해하기 위해서 움직임벡터는 반드시 필요하며 여러 물체의 이동 검출과 같은 다양한 이벤트 검출을 위해서는 움직임벡터와 다른 압축정보의 더 깊은 연구가 필요하다.

## 참고 문헌

- [1] 임영환, "ComBiStation : 분산 멀티미디어 컴퓨팅 환경을 위한 컴퓨팅 플랫폼," 정보과학회논문지, 제2권, 제 1호, pp. 6-8.
- [2] Earl Rennison, Kate Finn, D. H. Kim and Y. H. Lim, "Multimedia Processing Model for a Distributed Multimedia I/O System," Proceedings of 3rd International Workshop on Network and Operating Systems for Digital Audio/Video, 1993, pp. 233-239.
- [3] E. Rennison, R. Baker, D. H. Kim and Y. H. Lim, "MuX : An X Co-Existant Time-Based Multimedia I/O Server," The X Resource, Issue 1, 1992, pp. 213-233.
- [4] ITU-T Recommendation H.261
- [5] Farshid Arman, Arding Hsu, and Ming-Yee Chiu Siemens Corporate Research, Inc. Image Processing on Compressed Data for Large Video Databases, ACM Multimedia 1993, pp. 267-272.
- [6] Vikrant Kobla, David Doermann, and King-Ip(David) Lin, Archiving, Indexing, and retrieval of video in the compressed domain, SPIE Vol2916 1996, pp. 78-89.
- [7] 장상현, 유성현, 조형제, 압축된 비디오 데이터에서 내용 기반 색인을 위한 향상된 장면전환 추출방법, 한국정보과학회 춘계학술발표논문집, 1998, Vol. 25, No.1, pp. 616-618.